Amini A, Wah TY, Saboohi H. On density-based data streams clustering algorithms: A survey. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 29(1): 116–141 Jan. 2014. DOI 10.1007/s11390-013-1416-3

On Density-Based Data Streams Clustering Algorithms: A Survey

Amineh Amini, *Member*, *IEEE*, Teh Ying Wah, and Hadi Saboohi, *Member*, *ACM*, *IEEE*

Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya Kuala Lumpur 50603, Malaysia

E-mail: amini@siswa.um.edu.my; tehyw@um.edu.my; saboohi@siswa.um.edu.my

Received April 11, 2013; revised November 13, 2013.

Abstract Clustering data streams has drawn lots of attention in the last few years due to their ever-growing presence. Data streams put additional challenges on clustering such as limited time and memory and one pass clustering. Furthermore, discovering clusters with arbitrary shapes is very important in data stream applications. Data streams are infinite and evolving over time, and we do not have any knowledge about the number of clusters. In a data stream environment due to various factors, some noise appears occasionally. Density-based method is a remarkable class in clustering data streams, which has the ability to discover arbitrary shape clusters and to detect noise. Furthermore, it does not need the number of clusters in advance. Due to data stream characteristics, the traditional density-based clustering is not applicable. Recently, a lot of density-based clustering algorithms are extended for data streams. The main idea in these algorithms is using density-based methods in the clustering process and at the same time overcoming the constraints, which are put out by data stream's nature. The purpose of this paper is to shed light on some algorithms in the literature on density-based clustering over data streams. We not only summarize the main density-based clustering algorithms on data streams, discuss their uniqueness and limitations, but also explain how they address the challenges in clustering data streams. Moreover, we investigate the evaluation metrics used in validating cluster quality and measuring algorithms' performance. It is hoped that this survey will serve as a steppingstone for researchers studying data streams clustering, particularly density-based algorithms.

Keywords data stream, density-based clustering, grid-based clustering, micro-clustering

1 Introduction

Every day, we create 2.5 quintillion bytes of data; 90 percent of current data in the world has been created in the last two years alone. This data overtakes our capability to store and to process. In 2007, the amount of information created exceeded available storage for the first time. For example, in 1998 Google indexed 26 million pages, by 2000 it reached one billion, and in 2012 Google indexed over 30 trillion Web pages. This dramatic expansion can be attributed to social networking applications, such as Facebook and Twitter.

In fact, we have a huge amount of data generated continuously as data streams from different applications. Valuable information must be discovered from these data to help improve the quality of life and make our world a better place. Mining data streams is related to extracting knowledge structure represented in streams information. The research of mining data streams has attracted a considerable amount of researchers due to the importance of its application and the increasing generation of data streams[1-6].

Clustering is a significant class in mining data streams[5,7-11]. The goal of clustering is to group the streaming data into meaningful classes. Clustering data streams puts additional challenges to traditional data clustering such as limited time and memory, and further one pass clustering.

It is desirable for clustering data streams to have an algorithm which is able to, first discover clusters of arbitrary shapes, second handle noise, and third cluster without prior knowledge of number of clusters. There are various kinds of clustering algorithms for data streams. Among them, density-based clustering has emerged as a worthwhile class for data streams due to the following characteristics:

Firstly, it can discover clusters with arbitrary shapes. Partitioning-based methods are restricted to clusters structured on a convex-shaped. Discovery of clusters with a broad variety of shapes is very important for many data stream applications. For example, in the environment observations the layout of an area with similar environment conditions can be any shape.

Regular Paper
The work is supported by the University of Malaya Research under Grant No. RG097-12ICT.

Secondly, it has no assumption on the number of clusters. Most of the methods require previous knowledge of the domain to determine the best input parameters. Nevertheless, there is not a priori knowledge in a large amount of real life data.

Finally, it has the ability to handle outliers. For instance, due to the influence of different factors such as temporary failure of sensors in data stream scenario, some random noises appear occasionally. Detecting noise is one of the important issues specifically in evolving data streams in which the role of real data changes to noise over time.

There are different surveys recently been published in the literature for mining data streams. A number of them survey the theoretical foundations and mining techniques in data streams[2,12-14] as well as clustering as a significant class of mining data streams. Some of them review the well-known clustering methods in datasets[15-16]. Five clustering algorithms in data streams are reviewed and compared based on different characteristics of the algorithms in [17]. Furthermore, [18-20] review papers on different approaches in clustering data streams based on density. The work presented in [21] surveys existing clustering methods on data stream and gives a brief review on density-based methods. Different from them, this paper is a thorough survey of state-of-the-art density-based clustering algorithms over data streams.

Motivation. In real world applications, naturally occurring clusters are typically not spherical in shape and there are large amounts of noise or outliers in some of them. Density-based clustering can be applicable in any real world application. They can reflect the real distribution of data, can handle noise or outliers effectively, and do not make any assumptions on the number of clusters. Therefore, they are more appropriate than other clustering methods for data stream environments. Density-based method is an important data stream clustering topic, which to the best of our knowledge, has not yet been given a comprehensive coverage. This work is a comprehensive survey on the density-based clustering algorithms on data stream. We decouple density-based clustering algorithms in two different categories based on the techniques they use, which help the reader understand the methods clearly. In each category, we explain the algorithms in detail, including their merits and limitations. The reader will then understand how the algorithms overcome challenging issues. Moreover, it addresses an important issue of the clustering process regarding the quality assessment of the clustering results.

The remainder of this paper is organized as follows. In the next section, we discuss about the basic and challenges of clustering data streams as well as density-based clustering validation. Section 3 overviews the density-based clustering algorithms for data streams. Section 4 examines how the algorithms overcome the challenging issues and also compares them based on evaluation metrics. Finally, Section 5 concludes our study and introduces some open issues in density-based clustering for data streams.

2 Clustering Data Streams

Clustering is a key data mining task[5,7-11] which classifies a given dataset into groups (clusters) such that the data points in a cluster are more similar to each other than the points in different clusters.

Unlike clustering static datasets, clustering data streams poses many new challenges. Data stream comes continuously and the amount of data is unbounded. Therefore it is impossible to keep the entire data stream in main memory. Data stream passes only once, so multiple scans are infeasible. Moreover data stream requires fast and real time processing to keep up with the high rate of data arrival and mining results are expected to be available within short response time.

There are an extensive number of clustering algorithms for static datasets[15-16] where some of them have been extended for data streams. Generally, clustering methods are classified into five major categories[22]: partitioning, hierarchical, density-based, grid-based, and model-based methods (Fig.1).

A partitioning-based clustering algorithm organizes the objects into some number of partitions, where each partition represents a cluster. The clusters are formed based on a distance function like k-means algorithm[23-24] which leads to finding only spherical clusters and the clustering results are usually influenced by

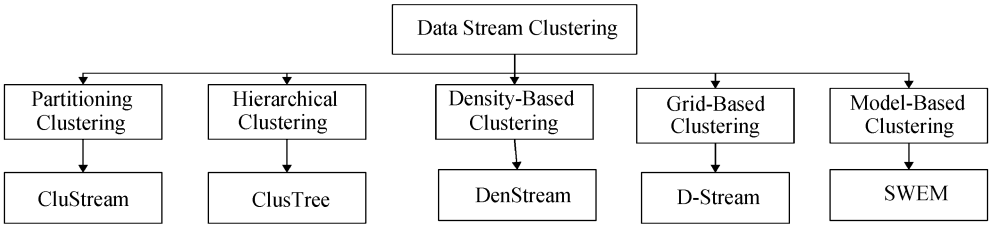

Fig.1. Data stream clustering algorithms[22].

noise. Two of the well-known extensions of k-means on data streams are presented in [25] where k-means algorithm clusters the entire data stream and in STREAM[7,9] which has LOCALSERACH algorithm based on k-median for data streams. Aggarwal *et al.* proposed an algorithm called CluStream[10] based on k-means for clustering evolving data streams. CluStream introduces an online-offline framework for clustering data streams which has been adopted for the majority of data stream clustering algorithms.

A hierarchical clustering method groups the given data into a tree of clusters which is useful for data summarization and visualization. In hierarchical clustering once a step (merge or split) is done, it can never be undone. Methods for improving the quality of hierarchical clustering have been proposed such as integrating hierarchical clustering with other clustering techniques, resulting in multiple-phase clustering such as BIRCH[26] and Chameleon[27]. BIRCH is extended for data stream as microcluster in [10]. Furthermore, ClusTree[28] is a hierarchical index for maintaining cluster feature. In fact, ClusTree builds a hierarchy of micro-clusters at different levels.

Grid-based clustering is independent of distribution of data objects. In fact, it partitions the data space into a number of cells which form the grids. Grid-based clustering has fast processing time since it is not dependent on the number of data objects. Some examples of the grid-based approach include: STING[29], which explores statistical information stored in the grid cells; WaveCluster[30], which clusters objects using a wavelet transform method; and CLIQUE[31], which represents a grid-based and density-based approach. Grid-based methods are integrated with density-based methods for clustering data streams which are referred to as density grid-based. In density grid-based clustering methods data points are mapped into the grids. Then, the grids are clustered based on their density. Some of the density grid based clustering algorithms are D-Stream[4,32] and MR-Stream[33].

Model-based clustering methods attempt to optimize the fit between the given data and some mathematical model like EM (Expectation Maximization) algorithm [34]. EM algorithm can be viewed as an extension of the k-means. However, EM assigns the objects to a cluster based on a weight representing the membership probability. In [35], SWEM (clustering data streams in a time-based sliding window with expectation maximization technique) is proposed which is a clustering data stream method using EM algorithm.

Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty in discovering clusters of arbitrary shapes such as the "S" shape and oval clusters. Given such data, they would likely inaccurately identify convex regions, where noise or outliers are included in the clusters.

Density-based methods have been developed based on the notion of density. The clusters are formed as dense areas which are separated from sparse regions. The main idea is to continuously grow a given cluster as long as the density (number of objects or data points) in the neighborhood exceeds some threshold. Such a method can be used to filter out noise or outliers and to discover clusters of arbitrary shape. The main density-based algorithms include: 1) DBSCAN[36] which grows clusters according to a density-based connectivity analysis, 2) OPTICS[37] which extends DBSCAN to produce a cluster ordering obtained from a wide range of parameter settings, 3) DENCLUE[38] which clusters objects based on a set of density distribution functions. Extensions of density-based algorithms are proposed as well and are discussed in Section 3 in details.

2.1 Basic of Clustering Data Streams

In clustering data streams, an important issue is how to process this infinite data which is evolving over time or how to keep the huge amount of data for later processing. There are some methods such as processing in one-pass, evolving and in online-offline manner as well as different methods for summarization of data streams. A short description of these methods is described as follows.

1) Processing.

One-Pass. In the one-pass, data streams are clustered by scanning data streams only once with the assumption that data objects arriving in chunks like k-means which was extended to be used for data streams[9,25,42]. Another well-known algorithm is STREAM[7,9], which partitions the input stream into chunks and computes (for each chunk) a cluster using a local search algorithm from [25]. DUCstream[43] is a one-pass grid-based clustering algorithm which assumes the arrival of data in chunks.

Evolving. In the one-pass approaches the clusters are computed over the entire data streams; however, data streams are infinite and they continuously evolve with time. Hence, the clustering results may change considerably over time. In the evolving approaches, the behaviors of streams are considered as an evolving process over time and processed in different forms of window model. Different clustering algorithms such as [10, 32-33, 40, 44-46] were developed based on this approach. In the window model, the data is separated into several basic windows and these basic windows are used as updating units. Three kinds of window models are as follows[47]:

• *Landmark Window Model.* The window is determined by a specific time point called landmark and the present. It is used for mining over the entire history of the data streams (Fig.2(a)).

• *Sliding Window Model.* Data is considered from a certain range in the past to a present time. The idea behind "sliding window" is to perform detailed analysis over both the most recent data points and the summarized version of the old ones (Fig.2(b)).

• *Fading* (*Damped*) *Window Model.* A weight is given for each data stream based on a fading function[44], and more weights are given to recent data compared with outdated data. The use of a damped window model is to diminish the effect of the old data on the mining result (Fig.2(c)).

The summarization of the window models with some example(s) of the clustering algorithm as well as their advantages and disadvantages are presented in Table 1. All the models have been considered in clustering data streams. Choice of the window model depends on the applications' needs[47].

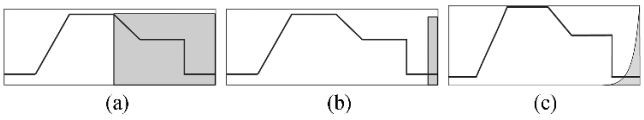

Fig.2. Window models[39]. (a) Landmark window model. (b) Sliding window model. (c) Fading window model.

Online-Offline. Sometimes a data stream clustering algorithm needs to investigate the clusters over different parts of the stream. A different time window model is used for tracing evolving behavior of data streams. However, we cannot perform dynamic clustering over all possible time horizons of data streams. Therefore, online-offline approach was introduced by Aggarwal *et al.* in [10]. The online component keeps summary information (overcoming real-time and memory constraints) about fast data streams and the offline component gives an understanding of the clusters. The majority of data stream clustering developed for evolving data streams use CluStream's two-phase framework[4,10,32-33,40,44-45].

2) Summarization. The large volume of data streams put space and time constraints on the computation process. Data streams are massive and infinite, so it is impossible to record the entire data. Therefore, synopsis information can be constructed from data items in the streams. The design and choice of a particular synopsis method depends on the problem being solved. A brief description about different methods of summarization is as follows[5,22]:

Sampling Methods. Instead of recording the entire data streams which seems impossible, we can make a sampling from the data stream. Reservoir sampling[48] is a technique which is used to select an unbiased random sample of data streams and it is useful for data streams.

Histograms. Histogram-based methods are used for static datasets; however, their extension for data streams is a challenging task. Some of the methods are discussed in [49] for data streams. One of the recent algorithms, called SWClustering[40], keeps summary information of data streams in the form of histogram.

Wavelets. Wavelets are popular multi-resolution techniques for data streams' summarization. Wavelets are traditionally used for image and signal processing. They are used for multi-resolution hierarchy structures over an input signal, in this case, the stream data. Furthermore, wavelet-based histograms can be dynamically kept over time[50-52].

Sketches. Sketch is a probabilistic summary technique for analyzing data streams. Sketch-based methods can be considered as a randomized version of wavelets technique. While other methods emphasize on small part of data, sketches summarize the entire dataset at multiple levels of details[5].

Table 1. Window Models in Clustering Data Streams

Window Model	Definition	Advantages	Disadvantages	Example (s)
Landmark window model	Analyze the entire history of data stream	Suitable for one-pass clustering algorithms[9]	All the data are equally important and the amount of data inside the window would quickly grow to unprocessable sizes	[9]
Sliding window model	Analyze the most recent data points	Suitable for applications where interest exists only in the most recent information like stock marketing	Ignoring part of streams	[40-41]
Fading (damped) window model	Assign different weights to data points	Suitable for applications where old data has an effect on the mining results, and the effect decreases as time goes on diminishing the effect of the old data	Unbounded time window (the window captures all historical data, and its size keeps growing as time elapses)	[3-4, 33]

Microcluster. Microcluster[10] is a method to keep statistical information about the data locality. It can adjust well with the evolution of underlying data streams. We will elaborate on microcluster further in Subsection 3.2.

Grid. In this method, the data space is partitioned into some small segments called grids and the data points in streams are mapped to them. Each grid has a characteristic vector which keeps a summary about all the data points mapped to it [4].

According to the reviewed papers, the most applicable summarization methods for density-based clustering algorithms are micro-clustering and grid-based. Therefore, we categorize the reviewed algorithms based on these two summarization methods[21].

2.2 Challenges in Clustering Data Streams

Considering their dynamic behavior, clustering over data streams should address the following challenges[1,22,25,28,33,53]:

• Handling noisy data. Any clustering algorithm must be able to deal with random noises present in the data since outliers have great influence on the formation of clusters.

• Handling evolving data. The algorithm has to consider that the data streams considerably evolve over time.

• Limited time. Data streams arrive continuously, which requires fast and real-time response. Therefore, the clustering algorithm needs to handle the speed of data streams in the limited time.

• Limited memory. A huge amount of data streams are generated rapidly, which needs an unlimited memory. However, the clustering algorithm must operate within memory constraints.

• Handling high-dimensional data. Some of data streams are high dimensional in their nature such as gene expression or clustering text documents. Therefore, the clustering algorithm has to overcome this challenge in case of its data being high dimensional.

We will discuss how different density-based clustering algorithms over data streams address aforementioned challenges in Subsection 4.1.

2.3 Density-Based Clustering Validation

One of the important issues of clustering algorithms is evaluating (validating) the goodness of the clustering results. There are some metrics for evaluating the quality of clustering results and testing the performance of the algorithms. The performance of the algorithms is tested with synthetic as well as real datasets. In the following parts, we introduce the most applicable real datasets and the evaluation metrics. Furthermore, we discuss new tools and models that have been developed recently for evaluating data stream clustering algorithms.

2.3.1 Datasets in Clustering Data Streams

There are some well-known real datasets for measuring the performance of clustering algorithms including: *charitable donation*, *network intrusion detection*, and *forest cover type* which is explained as follows:

• KDD Cup98 Charitable Donation. The dataset contains 95 412 records with 481 fields, which has information about people who have made charitable donations in response to direct mailing requests. Clustering can be used to group donors showing similar donation behaviors. The dataset was also used for predicting users who are more likely to donate to charity[10].

• KDD Cup99 Network Intrusion Detection. The goal of the dataset is to build a network intrusion detector capable of distinguishing attack, intrusion and other types of connection[54]. It has 494 020 connection records and each connection has 42 continuous and categorical attributes. Each record can correspond either to a normal connection or to an intrusion, which is classified into 22 types.

• Forest Cover Type. The Forest Cover Type real world dataset, is obtained from the UCI machine learning repository①. The dataset is comprised of 581 012 observations of 54 attributes, where each observation is labeled as one of the seven forest cover classes.

2.3.2 Evaluation Metrics

• Cluster Quality. A multitude of evaluation metrics were introduced in the literature for measuring cluster quality. Evaluation quality metrics can be categorized into two main classes, internal and external measures. The main difference is whether the external information is used for the cluster evaluation. Some of the internal and external evaluation measures are: C-index[55], sum of squared distance (SSQ)[22], silhouette coefficient[56], Rand Index[57-58], purity[59], van Dongen[60], B Cubed precision[1], V-measure[61], variation of information[62], F-measure[63], precision[63], and recall[63]. The complete list can be found in [64] for the internal and the external measures. However, the most often evaluation quality metrics used in clustering data streams in the reviewed algorithms are SSQ, purity, and Rand Index.

– SSQ measures how closely related are the objects in the cluster. In fact, it defines the compactness of the spherical clusters in convex approaches.

①Frank A, Asuncion A. UCI machine learning repository, 2010. http://archive.ics.uci.edu/ml/, Nov. 2013.

– Purity is defined as the average percentage of the dominant class labels in each cluster. The higher percentage of the dominant class labels in each cluster, the higher the cluster purity. In fact, the purity of the clusters is defined with respect to the true cluster (class) labels that are known for the datasets.

– Rand index (RI) measures the similarity between two data clusterings, having the highest value 1 when the clusters are exactly the same. In fact, it shows how the clustering results are close to the original classes.

SSQ and cluster purity are the two performance metrics used extensively in density-based data stream clustering. SSQ is more applicable in the convex shapes to define how the points are near the center. If we have the class labels of data, the better choice is to use the purity in the density-based clustering rather than SSQ[44].

In [65], the authors showed that if we use the other evaluation methods likes NMI (Normalized Mutual Information)[66], we would get better results rather than purity which is sensitive to the number of the clusters. They proved their results especially for MR-Stream. The NMI is a measure that evaluates how similar two clusterings are[67].

• Algorithm Performance.

– Scalability. The scalability test shows how the algorithm is scalable with both dimensionality and the number of the clusters. The scalability of the algorithms is defined in terms of execution time and memory usage.

Execution time is defined based on the total clock time used by an algorithm to process all the data. It is evaluated with various dimensionalities and different numbers of natural clusters. Synthetic datasets are used for these evaluations, because any combination of dimensionalities and any number of natural clusters could be used in the generation of datasets.

Memory usage is the amount of memory used by the algorithm. It depends on the data structure used for saving summary information of data streams such as: the number of micro-clusters, the number of grids in the hash table or the number of nodes in the tree. The memory usage is defined based on the real and synthetic datasets.

– Sensitivity. The sensitivity analysis evaluates the algorithms based on the analysis of the important algorithm parameters. It shows how the algorithm's parameters affect the clustering quality and what the best ranges for the algorithm's parameters are.

2.3.3 Tools and Software

The MOA (Massive On-line Analysis) framework②[68] is an open source benchmarking software for data streams that is built on the work of WEKA③[69]. MOA has a set of stream clustering algorithms and a collection of evaluation measures. MOA has considered stream classification algorithms; what is more, recently a stream clustering evaluation tool was added[70]. Furthermore, another evaluation measure called cluster mapping measure (CMM)[71] was integrated to MOA for evolving data streams. CMM has a mapping component which can handle emerging and disappearing clusters correctly. Kremer *et al.*[71], showed that the proposed measure can reflect the errors in data stream context effectively. SAMOA (Scalable Advanced Massive Online Analysis)[72] is another upcoming tool for mining big data streams. The goal of SAMOA is to provide a framework for mining data streams using a cluster/cloud environment.

3 Density-Based Clustering over Data Streams

Based on a comprehensive review on existing density-based clustering algorithms on data stream, these algorithms are categorized in two broad groups called density micro-clustering algorithms and density grid-based clustering algorithms[21] (Fig.3).

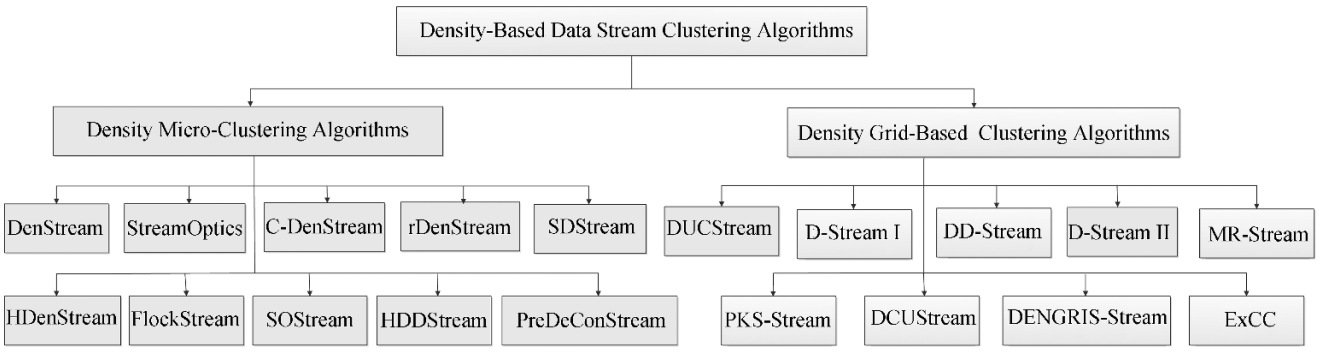

Fig.3. Density-based data stream clustering algorithms's categorization.

②http://mloss.org/software/view/258/, Nov. 2013.

③http://www.cs.waikato.ac.nz/ml/weka/, Nov. 2013.

In density micro-clustering algorithms, micro-clusters keep summary information about data and clustering is performed on the synopsis information. The reviewed algorithms in this category include: DenStream[3], StreamOptics[73], C-DenStream[74], rDenStream[75], SDStream[41], HDenStream[76], FlockStream[67], SOStream[77], HDDStream[78], and PreDeConStream[79].

In the density grid-based clustering algorithms group, the data space is divided into grids, data points are mapped to these grids, and the clusters are formed based on the density of grids. The reviewed algorithms in this category include: DUC-stream[43], D-Stream I[4], DD-Stream[80], D-Stream II[32], MR-Stream[33], PKS-Stream[81], DCUStream[82], DENGRIS-Stream[83], and ExCC[84].

In the following subsections, we will discuss in details about the algorithms, their advantages and disadvantages as well as evaluation metrics used. Additionally, we examine how they address the challenging issues in clustering data streams.

3.1 Density-Based Clustering

Density-based clustering has the ability to discover arbitrary-shape clusters and to handle noises. In density-based clustering methods, clusters are formed based on the dense areas that are separated by sparse areas. DBSCAN is one of the density-based clustering algorithms, which is adopted for data stream algorithms, described in details as follows.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)[36] is developed for clustering large spatial databases with noise, based on connected regions with high density. The density of each point is defined based on the number of points close to that particular point called the point's neighborhood. The dense neighborhood is defined based on two user-specified parameters: the radius (ϵ) of the neighborhood (ϵ-neighborhood), and the number of the objects in the neighborhood (*MinPts*). The basic definitions in DBSCAN are introduced in the following, where D is a current set of data points:

• ϵ-neighborhood of a point: the neighborhood within a radius of ϵ. Neighborhood of a point p is denoted by $N_\epsilon(p)$:

$$N_\epsilon(p) = \{q \in D | dist(p, q) \leqslant \epsilon\},$$

where $dist(p, q)$ denotes the Euclidean distance between points p and q;

• *MinPts*: the minimum number of points around a data point in the ϵ-neighborhood;

• core point: a point the cardinality of whose ϵ-neighborhood is at least *MinPts*;

• border point: a point is a border point if the cardinality of its ϵ-neighborhood is less than *MinPts* and at least one of its ϵ-neighbors is a core point;

• noise point: a point is a noise point if the cardinality of its ϵ-neighborhood is less than *MinPts* and none of its neighbors is a core point;

• directly density reachable: a point p is directly density reachable from point q, if p is in the ϵ-neighborhood of q and q is a core point;

• density reachable: a point p is density reachable from point q, if p is in the ϵ-neighborhood of q and q is not a core point but they are reachable through chains of directly density reachable points;

• density-connected: if two points p and q are density reachable from a core point o, p and q are density-connected;

• cluster: a maximal set of density-connected points.

Core, border and noise points are shown in Fig.4.

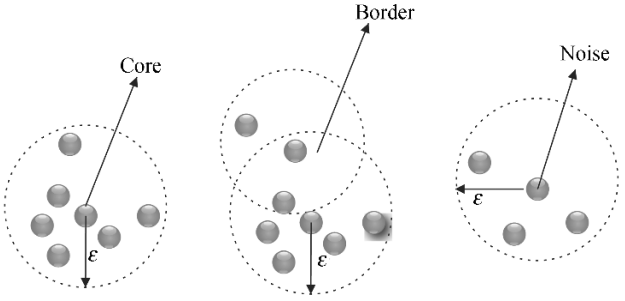

Fig.4. DBSCAN: core, border, and noise points.

DBSCAN starts by randomly selecting a point and checking whether the ϵ-neighborhood of the point contains at least *MinPts* points. If not, it is considered as a noise point, otherwise it is considered as a core point and a new cluster is created. DBSCAN iteratively adds the data points, which do not belong to any cluster and are directly density reachable[36] from the core points of a new cluster. If the new cluster can no longer be expanded, the new cluster is completed. In order to find the next cluster, DBSCAN randomly selects the unvisited data points and the clustering process continues until all the points are visited and no new point is added to any cluster.

Therefore, a density-based cluster is a set of density-connected data objects with respect to density reachability. The points that are not placed in any cluster are considered as noise. Fig.5 shows DBSCAN algorithm performing on a small synthetic dataset. Figs. 5(a), 5(b), and 5(c) are the steps of the clustering and Fig.5(d) is the final clustering results.

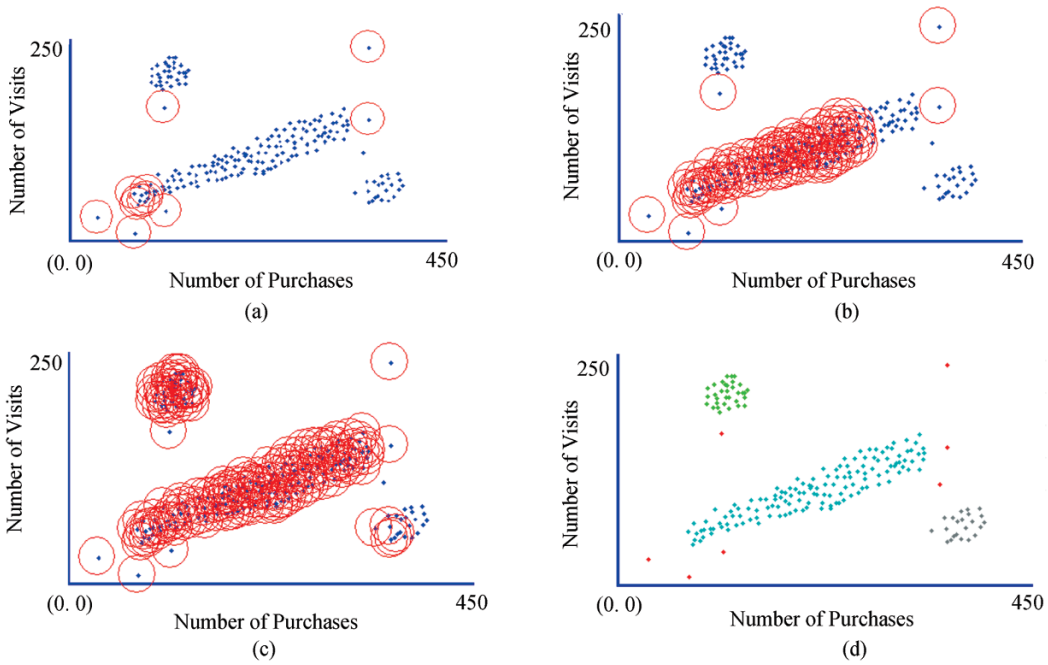

Fig.5. DBSCAN algorithm on a small synthetic dataset: $\epsilon = 20$, $MinPts = 5$.

3.2 Density-Based Micro-Clustering Algorithms on Data Streams

Micro-clustering is a remarkable method in stream clustering to compress data streams effectively and to record the temporal locality of data[5]. The micro-cluster concept was first proposed in [26] for large datasets, and subsequently adapted in [10] for data streams. The micro-cluster concept is described as follows (Fig.6):

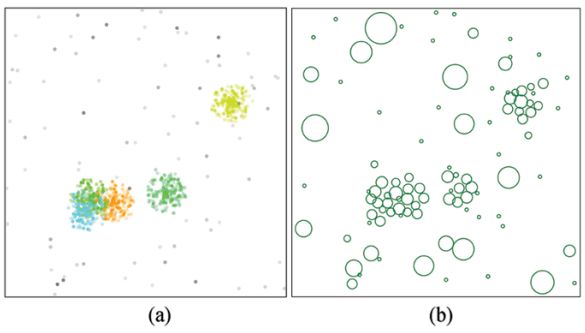

Fig.6. Micro-clusters in density-based clustering generated by MOA.

Micro-cluster is a temporal extension of cluster feature (CF)[26], that is a summarization triple maintained about a cluster. The triple vector comprises the number of data points, the linear sum of data points, and their squared sum. Therefore, a micro-cluster for a set of d-dimensional points $p_{i_1} \dots p_{i_n}$ is defined as the $(2 \times d + 3)$ tuple $(\boldsymbol{CF}2^x, \boldsymbol{CF}1^x, CF2^t, CF1^t, n)$, wherein $\boldsymbol{CF}2^x$ and $\boldsymbol{CF}1^x$ each correspond to a vector of d entries respectively.

- $\boldsymbol{CF}2^x$: for each dimension, the sum of squares of data values is maintained in $\boldsymbol{CF}2^x$. Therefore, $\boldsymbol{CF}2^x$ contains d values. The p-th entry of $\boldsymbol{CF}2^x$ is equal to $\sum_{j=1}^{n} = (x_{i_j}^p)^2$.
- $\boldsymbol{CF}1^x$: for each dimension, the sum of the data values is maintained in $\boldsymbol{CF}1^x$. Therefore, $\boldsymbol{CF}1^x$ contains d values. The p-th entry of $\boldsymbol{CF}1^x$ is equal to $\sum_{j=1}^{n} = x_{i_j}^p$.
- $CF2^t$: the sum of squares of timestamps $T_{i1} \dots T_{in}$.
- $CF1^t$: the sum of timestamps $T_{i1} \dots T_{in}$.
- n: the number of data points.

The micro-cluster for a set of points C is denoted by $\boldsymbol{CFT}(C)$.

Micro-clustering method uses micro-clusters to save summary information about the data streams, and performs the clustering on these micro-clusters.

3.2.1 *DenStream*

Feng *et al.*[3] proposed a clustering algorithm, termed as DenStream, for evolving data stream, which has the ability to handle noises as well. The algorithm extends the micro-cluster concept as core micro-cluster, potential micro-cluster, and outlier micro-cluster in order to distinguish real data and outliers. The core-micro-cluster synopsis is designed to summarize the clusters with arbitrary shape in data streams. Potential and outlier micro-clusters are kept in separate memo-

ries since they need different processing. DenStream is based on the online-offline framework. In the online phase it keeps micro-clusters with real data and removes micro-clusters with noises. In the offline phase, density-based clustering is performed on the potential micro-clusters which have the real data.

The algorithm uses the fading window model to cluster data streams. In this model the weight of each data point decreases exponentially with time t via a fading function $f(t) = 2^{-\lambda t}$, where $\lambda > 0$. Historical data diminishes its importance when λ assumes higher values.

DenStream extends the micro-cluster concepts to core-micro-clusters, potential-micro-clusters, and outlier-micro-clusters (Fig.7) which are described for a group of close points $p_{i1} \dots p_{in}$ with timestamps $T_{i1} \dots T_{in}$, as follows:

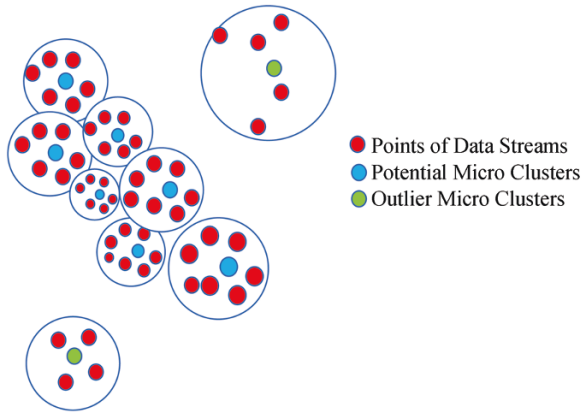

Fig.7. Potential and outlier microclusters.

Core-micro-cluster is defined as $CMC(w, c, r)$:

- $w = \sum_{j=1}^{n} = f(t - T_{ij})$, is the weight and $w \geqslant \mu$,
- $c = \frac{\sum_{j=1}^{n} f(t-T_{ij})p_{ij}}{w}$ is the center,
- $r = \frac{\sum_{j=1}^{n} f(t-T_{ij})dist(p_{ij},c)}{w}$, $r \leqslant \epsilon$ is the radius.

$dist(p_{ij}, c)$ is the Euclidean distance between point p_{ij} and the center c.

Note that the weight of a micro-cluster must be above a predefined threshold μ in order to be considered as a core.

Potential micro-cluster at the time t is defined as $(\boldsymbol{CF}^1, \boldsymbol{CF}^2, w)$.

- $w = \sum_{j=1}^{n} f(t - T_{ij})$, is the weight and $w \geqslant \beta\mu$. β is the parameter to determine the threshold of the outlier relative to core-micro-clusters $(0 < \beta < 1)$,
- $\boldsymbol{CF}^1 = \sum_{j=1}^{n} f(t - T_{ij})p_{ij}$, is the weighted linear sum of the points,
- $\boldsymbol{CF}^2 = \sum_{j=1}^{n} f(t-T_{ij})p_{ij}^2$, is the weighted squared sum of the points.

The center of potential micro-cluster is $\boldsymbol{c} = \frac{\boldsymbol{CF}^1}{w}$. And the radius of potential micro-cluster is $r = \sqrt{\frac{|\boldsymbol{CF}^2|}{w} - (\frac{|\boldsymbol{CF}^1|}{w})^2}$ $(r \leqslant \epsilon)$.

Outlier micro-cluster is defined as $(\boldsymbol{CF}^1, \boldsymbol{CF}^2, w, t_0)$. The definition of w, CF^1, CF^2, center, and radius are the same as in the potential-micro-cluster. $t_0 = T_{i1}$ denotes the creation time of the outlier micro-cluster. In an outlier micro-cluster the weight w must be below the fixed threshold, thus $w < \beta\mu$. However, it could grow into a potential micro-cluster when, by adding new points, its weight exceeds the threshold.

Weights of micro-clusters are periodically calculated and decision about removing or keeping them is made based on the weight threshold.

Online Phase. For initialization of the online phase, DenStream uses the DBSCAN algorithm on the first initial points, and forms the initial potential micro-clusters. In fact, for each data point, if the aggregate of the weights of the data points in the neighborhood radius is above the weight threshold, then a potential micro-cluster is created. When a new data point arrives, it is added to either the nearest existing potential micro-cluster or outlier micro-cluster. The Euclidean distance between the new data point and the center of the nearest potential or outlier micro-cluster is measured. A micro-cluster is chosen with the distance less than or equal to the radius threshold. If it does not belong to any of them, a new outlier micro-cluster is created and it is placed in the outlier buffer.

Offline Phase. It adopts DBSCAN to determine the final clusters on the recorded potential micro-clusters.

DenStream has a pruning method in which it frequently checks the weights of the outlier-micro-clusters in the outlier buffer to guarantee the recognition of the real outliers. The algorithm defines a density threshold function, which calculates the lower limit of density threshold. If the outlier micro-cluster weights below the lower limit, it is a real outlier and can be omitted from the outlier buffer.

Merits and Limitations. DenStream handles the evolving data stream effectively by recognizing the potential clusters from the real outliers. DenStream creates a new micro-cluster if the arriving records are incorporated into existing micro-clusters. However, the algorithm does not release any memory space by either deleting a micro-cluster or merging two old micro-clusters. Furthermore, the storage for the new micro-cluster is repeatedly allocated until it is eliminated in the pruning phase. Nevertheless, the pruning phase for removing outliers is a time consuming process in the algorithm.

3.2.2 StreamOptics

In [73], Tasoulis *et al.* developed a streaming cluster framework that graphically represents the cluster structure of data stream. It addresses visualiza-

tion challenges in clustering data streams. The algorithm is called StreamOptics that extends the OPTICS (Ordering Points to Identify the Clustering Structure) algorithm[37] for data streams using micro-cluster concept. Core-distance and reachability distance from OPTICS algorithm are changed in the form of micro-cluster as follows.

Definition 1. *Micro-cluster core-distance is equal to micro-cluster radius. In OPTICS, the core distance for a data point is the smallest value of ϵ (neighboring radius) that makes the data point as a core object. If the data point is not a core object, its core-distance is undefined.*

Definition 2. *The reachability-distance is the same as that in OPTICS. Reachability-distance of an object p_1 with respect to another object p_2 is chosen based on the maximum value between Euclidean distance of p_1, p_2 and the core distance of p_2. If p_2 is not a core object, the reachability-distance between p_1 and p_2 is undefined. However, in StreamOptics the distance is calculated between the potential micro-clusters. Reachability-distance between micro-clusters mc_1 and mc_2 is chosen based on the maximum value between the Euclidean distance of mc_1 and mc_2 and the core distance of mc_2. If mc_2 is not a core object, the reachability-distance between mc_1 and mc_2 is undefined.*

StreamOptics also uses potential micro-cluster and outlier micro-cluster from DenStream. StreamOptics keeps an ordered list from potential micro-clusters and discards outlier micro-clusters. Therefore, micro-cluster neighborhood and cluster ordering are defined based on the potential micro-clusters as follows.

Definition 3. *Micro-cluster neighborhood is defined based on the Euclidean distance between two potential micro-clusters.*

Definition 4. *Cluster ordering orders the potential micro-clusters based on their reachability distance.*

In StreamOptics, firstly the neighborhood of each potential micro-cluster is determined, and an ordered list of potential micro-clusters is made based on their reachability distance. StreamOptics produces a reachability plot that represents the micro-cluster structure using OPTICS algorithm.

Since data streams are changed by time, in StreamOptics, time is considered as the third dimension which is added to the two-dimensional plots of OPTICS. The StreamOptics plot allows the user to recognize the changes in cluster structure in terms of emerging and fading clusters.

Merits and Limitations. StreamOptics is based on micro-clustering framework, which uses OPTICS algorithm to provide the three-dimensional plot that shows the evolution of the cluster structure over the time. However, it is not a supervised method for cluster extraction; it needs manual checking of the generated three-dimensional plot.

3.2.3 C-DenStream

Ruiz *et al.* in [74] developed a density-based clustering algorithm with constraints for data streams. The algorithm is referred to as C-DenStream, which extends the concept of instance-level constraints from static data to stream data. Instance-level constraints are a particular form of background knowledge, which refer to the instances that must belong to the same cluster (Must-Link constraints) and those that must be assigned to different clusters (Cannot-Link constraints)[74]. In C-DenStream, instance level constraints are converted to potential micro-clusters level constraint (Fig.8) and final clusters are generated on the potential micro-clusters using C-DBSCAN[85].

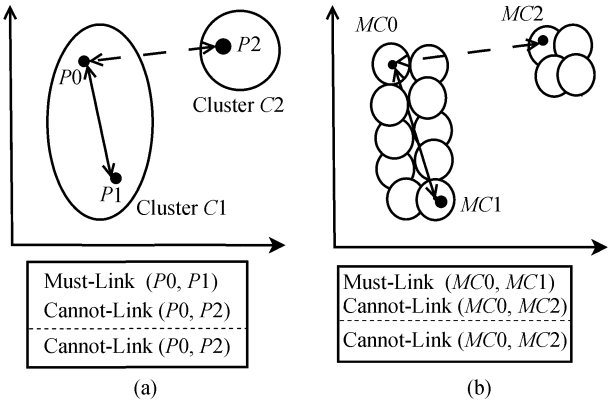

Fig.8. Micro-cluster constraint[85].

Merits and Limitations. C-DenStream includes domain information in the form of constraints by adding the constraints to the micro-clusters. The algorithm is very useful in the applications, which have a priori knowledge on the group membership of some records. It prevents the formation of the clusters that is included in the applications' semantics. However, the algorithm needs an expert to define its constraints. Moreover, the algorithm has DenStream limitations as well.

3.2.4 rDenStream

In [75] the authors developed a density-based clustering algorithm for applications with a large amount of outliers. The algorithm is a three-step clustering algorithm based on DenStream, which is referred to as rDenStream (DenStream with retrospect). rDenStream improves the accuracy of the clustering algorithm by forming a classifier from the clustering result. In the retrospect step of the algorithm, the misinterpreted dis-

carded data points get a new chance to be re-learned and to improve the robustness of the clustering.

In rDenStream the potential and the outlier micro-clusters are determined as in DenStream. However, instead of discarding the outlier micro-cluster, which cannot be converted to a potential-micro-cluster or to satisfy the density requirements, they are placed in a historical outlier buffer. In the retrospect phase, final clusters from performing DBSCAN on potential micro-clusters, are used to form a classifier. This classifier is applied to re-learn the outlier micro-cluster in the historical outlier buffer. In this phase, the micro-clusters that were chosen wrongly as outliers are modified to improve the clustering accuracy.

Merits and Limitations. rDenStream is useful for extracting knowledge pattern from the initial arriving data streams. However, the memory usage and the time complexity are high since it retains and processes the historical buffer. rDenStream is only applicable in the applications with a large amount of outliers, which are worthwhile to spend time and memory to gain better accuracy. The space complexity of rDenStream is similar to that of DenStream; however, it needs extra memory for keeping the historical outlier buffer.

3.2.5 SDStream

The SDStream algorithm[41] has the ability to discover the clusters with arbitrary shapes over sliding window[47]. In the algorithm, the distribution of the most recent data stream is considered and the data points that are not accommodated in sliding window length are discarded. It uses potential and outlier micro-clusters; however, they are stored in the form of exponential histogram. It is also an offline-online phase algorithm.

In the online phase, the new data points are added to the nearest micro-cluster. The nearest micro-cluster is either a potential-micro-cluster or an outlier-micro-cluster. The updated radius of the micro-cluster is less than its respective threshold radius. Otherwise, a new micro-cluster is created. Since the number of micro-clusters is limited, either a micro-cluster has to be deleted or two clusters be merged. For deleting a micro-cluster, the outdated micro-cluster is chosen according to its time value: if the time value does not belong to the length of sliding window. In the merging case, the two nearest micro-clusters, which are density-reachable[36], are merged[40]. In the offline phase, the final clusters of arbitrary shape are generated on potential micro-clusters using a modified DBSCAN.

Merits and Limitations. SDStream uses the sliding window model, processing the most recent data and summarizing the old data. In the real applications, users are interested in the distribution characteristics of the most recent data points. The authors of SDStream did not clarify the main usage of exponential histogram for their algorithm.

3.2.6 HDenStream

HDenStream[76] is a density-based clustering over evolving heterogeneous data stream. It adopts potential and outlier micro-cluster concepts from DenStream algorithm and uses distance method for categorical data from HCluStream[86]. HDenStream adds another entry to potential and outlier micro-cluster concept which is a two-dimensional (2D) array keeping the frequency of categorical data. In fact, for measuring distance between two micro-clusters with categorical data, the distances between two categorical attributes and continuous attributes are calculated separately. The algorithm has online and offline phases and the pruning phase is similar to that of DenStream as well.

Merits and Limitations. The algorithm can cover categorical and continuous data, which makes it more useful since in the real world applications we have any kind of data. However, the algorithm does not discuss how to save categorical features in an efficient way for data stream environment.

3.2.7 SOStream

SOStream (Self Organizing Density-Based Clustering Over Data Stream)[77] detects structure within fast evolving data streams by automatically adapting the threshold for density-based clustering. The algorithm has only online phase in which all mergings and updatings are performed. SOStream uses competitive learning as introduced for SOMs (Self Organizing Maps)[87] where a winner influences its immediate neighborhood. When a new data point arrives, a winner cluster is defined based on Euclidean distance of existing micro-clusters. If the calculated distance is less than a dynamically defined threshold, the micro-cluster is considered as a winner micro-cluster and the new data point will be added to it. It also affects the micro-cluster neighbors of the winner cluster. The neighbors are defined based on *MinPts* parameter of DBSCAN algorithm. The algorithm finds all the clusters overlapping with the winner. For each overlapping cluster its distance to the winner cluster is calculated. Any cluster with a distance less than that of the merge-threshold will be merged with the winner. If the new point is not added to any existing micro-cluster, a new micro-cluster is created for it. SOStream dynamically creates, merges, and removes clusters in an online manner.

Merits and Limitations. SOStream is a density-based clustering algorithm that can adapt its thresh-

old to the data stream. SOM is a time consuming method, which is not suitable for clustering data streams. SOStream is a micro-cluster based algorithm; however, its authors compared its result with two grid-based methods.

3.2.8 HDDStream

HDDStream[78] is a density-based algorithm for clustering high-dimensional data streams. It has online and offline phases. The online phase keeps summarization of both points and dimensions and the offline phase generates the final clusters based on a projected clustering algorithm called PreDeCon[88]. The algorithm uses DenStream concepts; however, it introduces prefer vector for each micro-cluster which is related to prefer dimension in high-dimensional data. A prefer dimension is defined based on variance along this dimension in micro-cluster. A micro-cluster prefers a dimension if data points of micro-clusters are more dense along this dimension. The micro-cluster with preferred vector is called a projected micro-cluster. Projected term shows that the micro-cluster is based on a subspace of feature space and not the whole feature space. Based on this concept, the algorithm changes the potential and outlier micro-clusters to projected potential micro-clusters and projected outlier micro-clusters respectively. HDDStream has pruning time similar to DenStream in which the weights of the micro-clusters are periodically checked.

Merits and Limitations. HDDStream can cluster high-dimensional data stream; however, in the pruning time it only checks micro-cluster weights. Since the micro-cluster fades over time the prefer vector should be checked as well because it may change over time.

3.2.9 PreDeConStream

PreDeConStream[79] is similar to HDDStream; however, PreDeConStream improves the efficiency of the HDDStream by working on the offline phase. This algorithm also introduces a subspace prefer vector which is defined based on the variance of micro-clusters and their neighbors. The algorithm keeps two lists including potential and outlier micro-clusters.

In the pruning time, the neighbors of newly inserted potential micro-clusters as well as deleted potential micro-clusters are checked. The subspace prefer vectors of these neighboring micro-clusters are updated and put in a list as affected micro-clusters. The affected micro-cluster list is used in the offline phase as expanding clusters to improve the efficiency of the offline phase.

Merits and Limitations. The algorithm can cluster high-dimensional data stream based on the density method. However, searching the affected neighboring clusters is a time consuming process.

3.2.10 FlockStream

FlockStream[67] is a density-based clustering algorithm based on a bio-inspired model. It is based on flocking model[89] in which agents are micro-clusters and they work independently but form clusters together. It considers an agent for each data point, which is mapped in the virtual space. Agents move in their predefined visibility range for a fixed time. If they visit another agent, they join to form a cluster in case they are similar to each other. It merges online and offline phases since the agents form the clusters at any time. In fact, it does not need to perform offline clustering to get the clustering results.

Since, FlockStream only compares each new point with the other agents in its agent visibility distance, it reduces the number of comparisons in the neighborhood of each point. The visibility distance has a threshold which is defined by the users. The agents have some rules in order to move in the virtual space such as cohesion, separation and alignment[67]. These rules are executed for each agent over the time. FlockStream has three kinds of agents: basic representative agents for new data point and p-representative, and o-representative agents which are based on potential- and outlier-micro-clusters respectively. Actually, when the similar basic agents merge to each other, they form a p-representative or an o-representative agent based on their weights.

Merits and Limitations. FlockStream is more efficient compared with DenStream since the number of comparisons is so limited. In DenStream, for each new data point, the distances to all existing potential and outlier micro-clusters have to be calculated. Furthermore, it does not perform offline phase frequently. Although the algorithm forms an outlier agent to handle noise, there is not any clear strategy to show when and how to remove the outliers from the agents list.

Table 2 summarizes some of the main characteristics of the reviewed density micro-clustering algorithms.

3.3 Density Grid-Based Clustering Algorithms on Data Streams

Using density-based and grid-based methods, researchers developed several hybrid clustering algorithms for data streams referred to as density grid-based clustering algorithms[4,32-33]. In these algorithms, the data space is partitioned into small segments called grids. Each data point in data streams is mapped into a grid and then the grids are clustered based on their

Table 2. Main Characteristics of Density Micro-Clustering Algorithms

Name	Year	Type of Data	Input Parameters	Results	Objective
DenStream[3]	2006	Continuous	Cluster radius, cluster weight, outlier threshold, decay factor	Arbitrary shape clusters	Clustering evolving data streams
StreamOptics[73]	2007	Continuous	Potential micro-cluster list, core distance, reachability distance	Cluster structure plot over time	Cluster visualization
C-DenStream[74]	2009	Continuous	Cluster radius, minimum number of points in the neighborhood, outlier radius, decay factor, a stream of instance level constraint	Arbitrary shape clusters with constraint	Applying constraint in clustering
rDenStream[75]	2009	Continuous	Cluster radius, cluster weight, outlier threshold, decay factor	Arbitrary shape clusters	Improving accuracy
SDStream[41]	2009	Continuous	sliding window size, cluster radius, cluster weight	Arbitrary shape clusters over sliding window	Clustering over sliding window
HDenstream[76]	2009	Continuous, categorical	Cluster radius, cluster weight, outlier threshold, decay factor	Arbitrary shape clusters	Improving quality
SOStream[77]	2012	Continuous	Cluster radius	Clustering threshold	Automating clustering threshold selection
HDDStream[78]	2012	Continuous	Cluster radius, cluster weight, outlier threshold, decay factor	Arbitrary shape clusters	Clustering high-dimensional data
PreDeCon-Stream[79]	2012	Continuous	Cluster radius, cluster weight, outlier threshold, decay factor	Arbitrary shape clusters	Clustering high-dimensional data
FlockStream[67]	2013	Continuous	Cluster radius, cluster weight, outlier threshold, decay factor	Arbitrary shape clusters	Density-based clustering using flocking model

density. Density grid-based algorithms not only can discover arbitrary shape clusters and detect the outliers, but also have fast processing time which only depends on the number of cells (Fig.9).

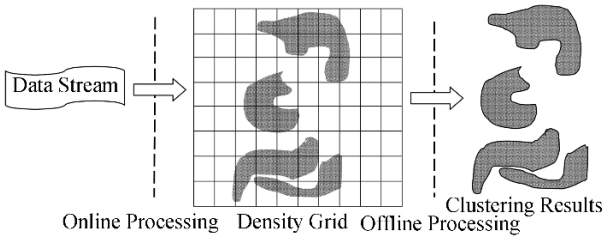

Fig.9. Density grid-based clustering framework[32].

According to the reviewed algorithms, some definitions form the basis of the density grid-based algorithms. In these algorithms, the data space is partitioned into density grids and each data point $x = \{x_1, x_2, \ldots, x_d\}$ is mapped to a density grid $g(x)$. Based on this assumption the following concepts are described:

• *Density Coefficient.* For each data point, a density coefficient is considered to capture the dynamic changes of the clusters. The density of each grid is associated with a decay factor, which is decreased over time. In fact, the grids are processed in the form of fading window model.

• *Grid Density.* The density of each grid is defined based on the aggregation of density coefficient of all the data points in that grid[4]. However, in an algorithm called DUC-Stream[43], the density of the grid is defined based on its number of data points.

• *Dense, Sparse and Transitional Grid.* Density grid-based algorithms consider a threshold for the density of each grid. This density threshold categorizes the grid as dense, sparse, and transitional. A grid is considered as dense if its density is higher than a special threshold. If the grid density is lower than another special threshold, the grid is a sparse grid. The grid with density between the dense and sparse density thresholds is considered as a transitional grid.

• *Characteristic Vector.* Keeps some information about the data points, which are mapped to the grid, such as grid density, update time, creation time, and grid type.

• *Grid Cluster.* A group of dense neighboring grids, which has higher density than the surrounding grids, form a grid cluster[4].

In the following subsections, we explain the density grid-based algorithms in details and discuss their advantages and disadvantages.

3.3.1 DUCstream

Gao *et al.*[43] have proposed an incremental single pass clustering algorithm for data streams using dense unit, which is referred to as DUCstream. DUCstream assumes the arrival of data in chunks, which contain some points. The density of each unit is its number of points and if it is higher than a density threshold, it is considered as a dense units. The algorithm introduces the local dense unit in order to keep only the units, which are most probably converted to dense unit. In

DUCstream, the clusters are identified as a connected component of a graph in which the vertices show the dense units and edges show their relation. Therefore, when a dense unit is added, if there is no related cluster, a new cluster is created; otherwise, the new dense unit is absorbed to the existing clusters.

Furthermore, DUCstream keeps the clustering results in bits, which are called clustering bits, to retain little amount of memory. The clustering bit is a bit string, which keeps the number of dense units. In fact, the clustering result is created in an incremental manner. The time complexity and the memory space of DUCstream are claimed to be low due to utilizing the bitwise clustering.

Merits and Limitations. DUCstream checks the density of each unit. If the unit does not receive enough data points over time, its density is decreased so it is not considered for clustering. Since DUCstream processes the data in chunks, it relies on the user to determine the size of the chunks of data.

3.3.2 D-Stream I

Chen and Tu[4] proposed a density-based clustering framework for clustering data streams in the real time which is termed as D-Stream I. D-Stream I has online and offline phases.

The online phase reads a new data point, maps it into the grid, and updates the characteristic vector of the grid.

The offline phase adjusts the clusters in each time interval gap. The time interval gap is defined based on the minimum conversion time between different kinds of grids. In the first time interval, each dense grid is assigned to a distinct cluster. After that, in each time interval, clusters are adjusted by determining dense and sparse grids. A threshold is considered for the grid density. If the grid density is higher than the special threshold, it is a dense grid otherwise it is considered as a sparse grid. If the grid is dense, it is merged with neighboring grids with higher density and they form a cluster. Otherwise, if it is sparse, the grid is removed from the cluster. In fact, D-Stream I firstly updates the density of the grids and then performs the clustering based on a standard method of density-based clustering.

An important motivation behind this framework is handling the outliers by considering them as sporadic grids. Sporadic grid is a kind of sparse grid, which has very few data and does not have any chance to be converted to a dense grid. D-Stream I defines a lower limit for density threshold based on density threshold function. If a sparse grid density is less than the lower limit of density threshold, it is considered as a sporadic grid. It has also a pruning phase, which happens in each time interval gap. In this phase, the clusters are adjusted and the sporadic grids are removed from the grid list. D-Stream I uses a hash table for keeping the grid list.

Merits and Limitations. D-Stream I clusters data streams in real time based on the grid and the density. It also proposes a density decaying to adjust the clusters in real time and captures the evolving behavior of data streams and has techniques for handling the outliers. However, for determining the time interval gap, the algorithm considers the minimum time for a dense grid to be converted to sparse and vice versa. Therefore, the gap depends on many parameters. In fact, it could be better that the algorithm would define the time gap based on only the conversion of dense grids to sparse ones, since the conversion of sparse grid to dense one has already been considered in the weight of the grid. Furthermore, it cannot handle the high-dimensional data because it assumes that the majority of the grids are empty in the high-dimensional situation.

3.3.3 DD-Stream

DD-Stream algorithm[80] is an extension of D-Stream I, which improves the cluster quality by detecting the border points in the grids. The boundary points are extracted before performing any adjustment on the grids. The online phase performs merely like D-Stream I. The offline component runs in each time interval gap (defined like D-Stream I) and extracts boundary points, detects dense and sparse grids, and clusters the dense grids using density-based methods. DD-Stream assigns the points on the borders based on their distances from the center of the neighboring grids. If the distances are equal, the neighboring grid with higher density is chosen. The information about the center of the grids is kept in the characteristic vector of the grid.

Merits and Limitations. DD-Stream extracts the boundary points from the grids to improve the quality of the clustering. However, the border points are extracted whenever the data is mapped to the grids, which is a time consuming process. It is better to detect the border point in each time interval gap before merging the grids rather than at the arrival time of the data points. Furthermore, the algorithm recognizes the sparse and dense grids based on their density, but it does not have any clear strategy for removing the sporadic grids.

3.3.4 D-Stream II

Tu and Chen[32] proposed an algorithm for clustering data streams based on grid density and attraction. The algorithm is based on the observation that many

density-based clustering algorithms do not consider the positional information of data in the grid. The idea is based on using grid attraction for the grids. Grid attraction[32] shows that to what extent the data in one neighbor is closer to that of another neighbor.

In fact, the algorithm is an extension of D-Stream I, and we refer to it as D-Stream II. The clustering procedure of D-Stream II is similar to D-Stream I; however, in D-Stream II, two dense grids are merged in case that they are strongly correlated. Two girds are called strongly correlated if their grid attractions are higher than a pre-defined threshold. D-Stream II has pruning techniques, like D-Stream I, to adjust the clusters in each time interval gap and to remove the sporadic grids mapped by the outliers.

Merits and Limitations. D-Stream II improves the quality of clustering to some extent by considering the position of the data in the grids for clustering. However, the algorithm still has the problems that are already mentioned in D-Stream I. Nevertheless, it keeps the grid list in a tree rather than a table that makes the processing of the grid list faster and it reduces the memory space.

3.3.5 MR-Stream

Wan *et al.*[33] developed an algorithm for density-based clustering of data streams at multiple resolutions, termed as MR-Stream. The algorithm improves the performance of density-based data stream clustering algorithm by running the offline component at constant times. The algorithm determines the right time for the users to generate the clusters.

MR-Stream partitions the data space in cells and a tree-like data structure, which keeps the space partitioning. Each time a dimension is divided in two, and a cell can be further divided in 2^d parts where d is the dataset dimensionality. The tree data structure keeps the data clustering in different resolutions. Each node has the summary information about its parent and children.

MR-Stream has online and offline phases. In the online phase, when a new data point is arrived, it is mapped to its related grid cell. In the tree structure, if there is not any sub-node, a new sub-node is created for the new data point, and the weight of the new sub-node's parent is updated. The update of weight continues up to the root of the tree. In each time interval gap, the tree is pruned in two ways: from the root to the maximum height and vice versa. In pruning from leaf to root, the sparse grids are detected and the densities of dense grids are added to their parents. In the pruning from root to the maximum height, the dense grids are detected and sparse grids are merged to form noise clusters. The sporadic grid cell is also removed by comparing its density with lower limit of density threshold function.

The offline phase generates clusters at a user-defined height. It determines all the reachable dense cells at a special distance and marks them as one cluster. The noise clusters are removed by checking their size and density with size and density thresholds respectively.

The authors of MR-Stream proposed a memory sampling method to recognize the right time to trigger the offline component. In this method, the algorithm makes a relation between nodes in the tree and evolution of clusters.

Merits and Limitations. MR-Stream introduces a memory sampling method in order to define the right time for running the offline component, which improves the performance of the clustering. However, MR-Stream keeps the sparse grids and merges them for consideration as a noise cluster. It is better not to let the noise cluster to be formed by checking the density of the sparse grids. Furthermore, the algorithm cannot work properly in high-dimensional data.

3.3.6 PKS-Stream

Ren *et al.*[81] proposed an algorithm for clustering data streams based on the grid density for high-dimensional data streams referred to as PKS-Stream. The algorithm is based on the observation that in grid-based clustering, there are a lot of empty cells specially for the high-dimensional data. The idea is based on using PKS-tree for recording non-empty grids and their relations as well. For keeping the non-empty cells, PKS-Stream introduces the k-cover grid cell concept. A grid is a k-cover, if it has the minimum density threshold and it is not covered by any other grid. In fact, k-cover shows the non-empty grids in the neighboring of the leaf node grids.

PKS-Stream has online and offline phases. The online phase maps the data records to the related grid cells in the PKS-tree, if there is a grid cell for the data record. Otherwise, a new grid cell is created. The offline phase forms the clusters based on the dense neighboring grids. In each time interval gap, the PKS-tree is adjusted and the sparse grids are removed from the tree.

Merits and Limitations. PKS-Stream is a density grid-based clustering, which handles the high-dimensional data stream. However, it does not have any pruning on the tree after adding a new data point to any of the cells of the tree. PKS-Stream depends on k, which affects the clustering result. It also affects the k-cover, which defines the resolution of the cluster.

3.3.7 DCUStream

DCUStream[82] is a density-based clustering algorithm over uncertain data stream. For each data point in the stream a tuple which includes data point, existence probability of the data point and its arrival time are considered. Each data point is mapped into a grid. The algorithm considers an uncertain tense weight for each data point which is calculated based on the temporal feature of data stream and its existence probability. By aggregation of uncertain tense weight, the algorithm defines the uncertain data density. DCUStream introduces the core dense grid, which is a dense grid with sparse neighbors. By considering the threshold for uncertain data density, dense and sparse grids are defined. For clustering, DCUStream examines all the grids to find core dense grid. It uses depth first search algorithm to find neighbor grids. The process continues for all unlabeled dense grids. All sparse grids are considered as noise.

Merits and Limitations. DCUStream algorithm improves density-based clustering algorithm for uncertain data stream environment. However, searching the core dense grids and finding their neighbors are time-consuming processes.

3.3.8 DENGRIS-Stream

DENGRIS-Stream[83] is a density grid-based clustering for stream data over sliding window. The algorithm maps each input data into a grid, computes the density of each grid, and clusters the grids using density concepts within time window units. DENGRIS-Stream can capture the distribution of recent records precisely using sliding window model, which is more preferable in data stream applications. It introduces the expired grid concept for detecting and removing the grids whose time stamps are not in the sliding window. Furthermore, DENGRIS-Stream removes the expired grids before any processing on the grid list that leads to save time and memory.

Merits and Limitations. DENGRIS-Stream is the first density grid-based clustering algorithm for evolving data streams over sliding window model. However, there is no evaluation to show its effectiveness compared with other state-of-the-art algorithms.

3.3.9 ExCC

ExCC (Exclusive and Complete Clustering)[84] is an exclusive and complete clustering algorithm for heterogeneous data stream. It is an online-offline algorithm. Online phase keeps synopsis in the grids and offline phase forms the final clusters on demand. The algorithm maps the numerical attributes to the grid and the categorical attributes are assigned granularities according to distinct values in respective domain sets. ExCC is a complete algorithm since it uses pruning based on the speed of data stream rather than a window model such as fading one. ExCC introduces fast or slow stream based on the average arrival time of the data points in the data stream. Furthermore, it is an exclusive clustering algorithm since it uses grid for the distribution of data. The algorithm detects noise in the offline phase using wait and watch policy. For detecting real outliers, it keeps the data points in the hold queue, which is kept separately for each dimension. ExCC uses a user specified threshold for detecting dense and sparse grids. ExCC can filter out noise using cell density and cluster density threshold which is specified by the user. However, the algorithm estimates the threshold based on the granularity of the grid, the data dimension, and the average number of points in each grid. In order to generate the clusters, it considers a pool for dense and recent grids. The dense neighboring grids are chosen from this pool by considering neighboring of each grid. For categorical data the equality of the attributes are considered.

Merits and Limitations. ExCC can cover data stream with mixed attributes (numeric and categorical). Furthermore, the algorithm compares the results with micro-clustering methods. However, since it is a grid-based algorithm the results have to be compared with grid-based algorithms to be fair. The hold queue strategy needs more memory and processing time since it is defined for each dimension. Moreover, using pool for keeping dense grids requires more memory to keep and more time to process.

We summarize the main characteristics of the density grid-based clustering algorithms in Table 3.

4 Discussion

Fig.10 depicts the distribution of the reviewed papers for density-based data stream clustering algorithms over years. There are two peaks in 2009 and 2012 for both categories. However, it can be observed that micro-clustering methods are more popular than grid methods.

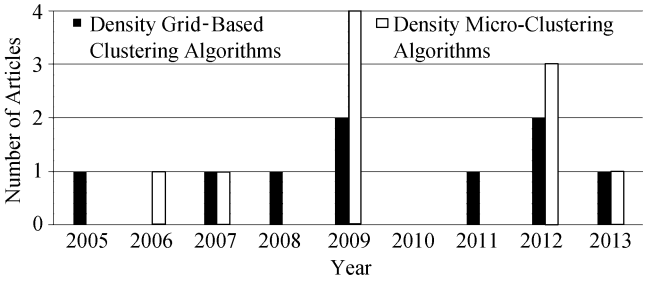

Fig.10. Distribution of the reviewed papers.

Table 3. Main Characteristics of Density Grid-Based Clustering Algorithms

Name	Year	Type of Data	Input Parameters	Results	Objective
DUCstream[43]	2005	Undefined	Chunks of data streams	Clusters as the connected components of the graph	One-scan clustering algorithm
D-Stream I[4]	2007	Continuous	Data stream, decay factor, dense grid threshold, sparse grid threshold	Arbitrary shape clusters	Real-time clustering
DD-Stream[80]	2008	Continuous	Data stream, decay factor, dense grid threshold, sparse grid threshold	Arbitrary shape clusters	Improving quality
D-Stream II[32]	2009	Continuous	Data stream, decay factor, dense grid threshold, sparse grid threshold	Arbitrary shape clusters	Improving quality
MR-Stream[33]	2009	Continuous	Data stream, decay factor, dense cell threshold, sparse cell threshold	Clusters in multiple resolutions	Improving performance
PKS-Stream[81]	2011	Continuous	PKS-tree, density threshold	Arbitrary shape clusters	Clustering high-dimensional data
DCUStream[82]	2012	Continuous	Data stream dimension, density threshold	Arbitrary shape clusters	Clustering uncertain data
DENGRIS-Stream[83]	2012	Continuous	Data stream, sliding window size	Arbitrary shape clusters	Clustering over sliding window
ExCC[84]	2013	Continuous, categorical	Grid granularity	Arbitrary shape clusters	Clustering heterogeneous data streams

In Fig.11 (motivated from [90]), we show the chronological order of the reviewed algorithms as well as how the algorithms relate to each other. It can be observed from the figure that the most remarkable algorithms are DenStream and D-Stream I in micro-clustering and the grid group respectively. Other algorithms in each of the categories try to improve the two mentioned algorithms in different aspects such as improving efficiency or quality or handling different kinds of data by adding some features which are listed in Table 4.

4.1 Algorithms and Challenging Issues

In this subsection, we briefly describe how the algorithms overcome the challenges.

• *Handling Noisy Data.* In micro-clustering algorithms outlier micro-cluster is introduced. The outlier and the real data are retained in different forms of micro-clusters, which help to distinguish the seeds of the new clusters from those of the outliers. In the grid methods, sporadic grid is introduced which has a limited number of data points mapped by outliers.

• *Handling Evolving Data.* Both density-based micro-clustering and grid-based clustering algorithms have the ability to handle evolving data streams using different kinds of window models such as fading and sliding window models. DUCstream does not handle evolving data because it considers the behavior of data streams as the data points arriving in chunks.

• *Limited Time.* D-Stream II has the lowest time complexity, which enables the processing of data stream in limited time. Other algorithms' time complexity grows linearly as data streams are generated. However, the algorithms such as rDenStream and C-DenStream need more time for processing historical buffer and constraints respectively. SOStream has the highest time complexity compared with other algorithms.

• *Limited Memory.* The aforementioned algorithms use micro-clusters or grid to keep summary about the data stream to process data points. However, the algorithms such as rDenStream, C-DenStream, FlockStream and ExCC need more memory.

• *Handling High-Dimensional Data.* If the algorithms are used for the high dimensional data the time complexity would be high which is not acceptable in data stream clustering. In the grid methods, in this situation the number of grids becomes large. PKS-Stream, HDDStream, and PreDeConStream are the algorithms with the ability to handle high-dimensional data streams.

Table 5 summarizes how the algorithms address the mentioned challenging issues.

4.2 Algorithms Evaluation

We compare the algorithms based on the evaluation metrics. The algorithms with same metrics are compared together, for example, algorithms using purity (DenStream, rDenstream, SDStream, PKS-Stream, MR-Stream, FlockStream, HDenStream, SOStream, PreDeConStream) (Fig.12(a)) and algorithms using SSQ (D-Stream I, D-Stream II) (Fig.12(b)). However, C-DenStream is the only algorithm which uses rand index and it is compared with DenStream (Fig.12(c)). FlockStream also uses NMI (normalized mutual infor-

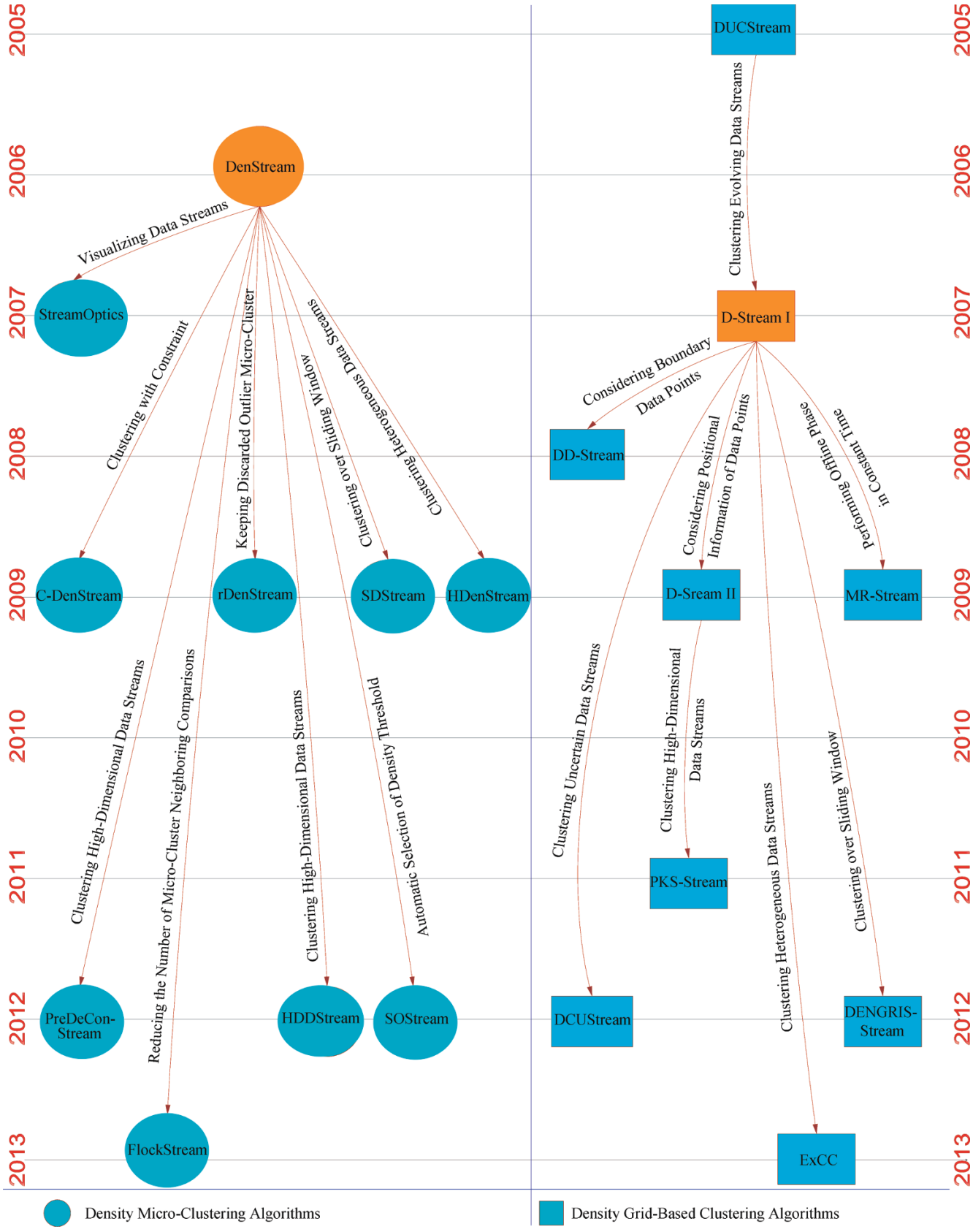

Fig.11. Chronological order of the reviewed density-based data stream clustering algorithms.

Table 4. Algorithms' Relations

Name	Added Feature	Objective
DenStream[3]	Main algorithm	Density micro data stream clustering
StreamOptics[73]	DenStream + visualization	Graphically representing the cluster structure of the data stream
C-DenStream[74]	DenStream + constraint	Guiding clustering process using domain information
rDenStream[75]	DenStream + retrospect phase	Using discarded micro-cluster to improve accuracy
SDStream[41]	DenStream + sliding window	Clustering more recent data
HDenstream[76]	DenStream + categorical data	Achieving higher cluster purity
SOStream[77]	Automating DenStream parameters	Removing difficulties in choosing unsuitable parameters
HDDStream[78]	DenStream + high-dimensional data	Density-based projected clustering over high-dimensional data streams
PreDeConStream[79]	DenStream + high-dimensional data	Improving efficiency of offline phase in density-based projected clustering over high-dimensional data streams
FlockStream[67]	DenStream + bio-inspired model	Avoiding the computing demanding offline cluster computation
DUCstream[43]	Clustering data stream in chunks	Density grid-based single pass clustering
D-Stream I[4]	Main algorithm	Density grid-based data stream clustering
DD-Stream[80]	D-Stream I + considering boundary points	Improving quality
D-Stream II[32]	D-Stream I + grid attraction	Considering positional information of the data in that grid to improve quality
MR-Stream[33]	D-Stream I + removing offline phase	Improving quality
PKS-Stream[81]	D-Stream II + high-dimensional data	Clustering high-dimensional data streams
DCUStream[82]	D-Stream I + uncertain data	Improving density-based clustering algorithm for uncertain data stream environment
DENGRIS-Stream[83]	D-Stream I + sliding window	Clustering more recent data streams
ExCC[84]	D-Stream I + categorical data	Exclusive and complete clustering for mixed attributes data streams

Table 5. Density-Based Clustering Algorithms and Challenging Issues

Density-Based Clustering Algorithms	Handling Noisy Data	Handling Evolving Data	Limited Time	Limited Memory	Handling High-Dimensional Data
DenStream	√	√	√	√	-
StreamOptics	√	√	-	-	-
C-DenStream	√	√	-	-	-
rDenStream	√	√	-	-	-
SDStream	√	√	-	√	-
HDenStream	√	√	√	√	-
SOStream	√	√	-	√	-
HDDStream	√	√	-	√	√
PreDeConStream	√	√	-	√	√
FlockStream	√	√	√	-	-
DUCstream	√	-	√	√	-
D-Stream I	√	√	-	-	-
DD-Stream	√	√	-	-	-
D-Stream II	√	√	√	√	-
MR-Stream	√	√	-	-	-
PKS-Stream	√	√	-	-	√
DCUStream	√	√	-	√	-
DENGRIS-Stream	√	√	-	√	-
ExCC	√	√	-	-	-

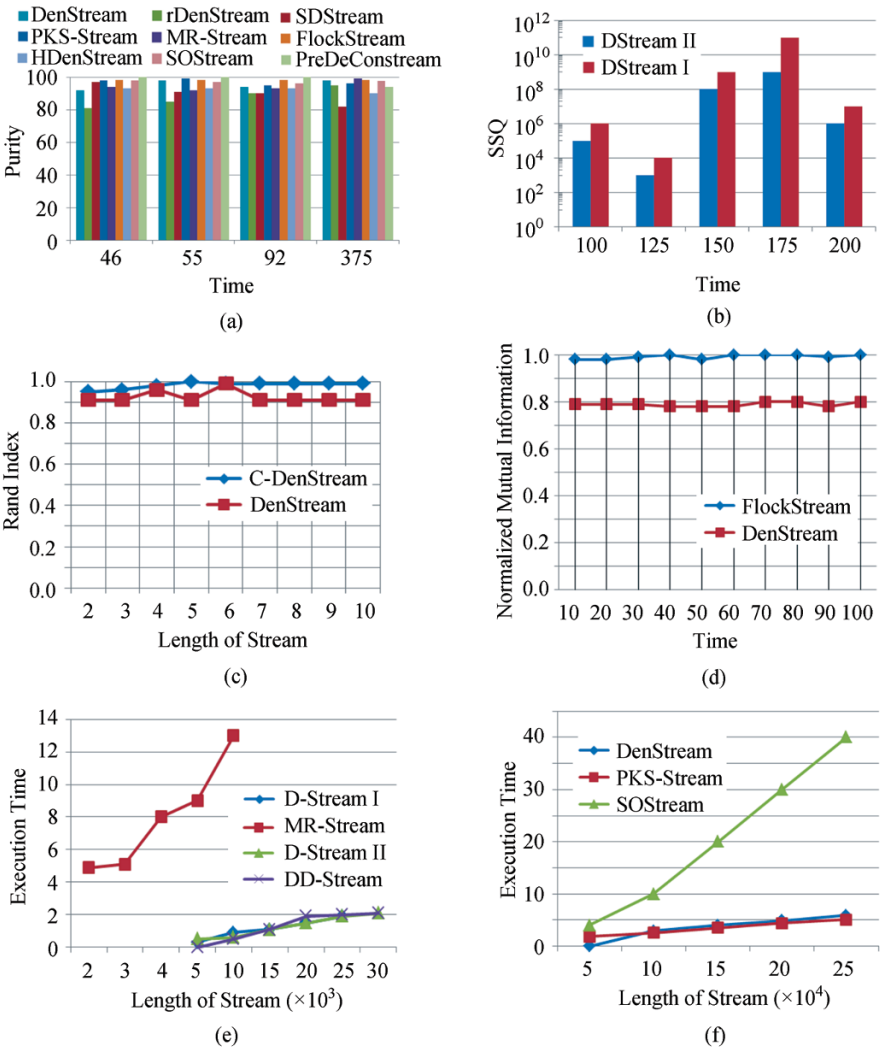

Fig.12. Algorithm evaluation. (a) Quality comparison – purity. (b) Quality comparison – SSQ. (c) Quality comparison – RI. (d) Quality comparison – NMI. (e) Execution time comparison – length of stream $< 50\,\mathrm{K}$. (f) Execution time comparison – length of stream $> 50\,\mathrm{K}$.

mation)[66] and is compared with DenStream to measure quality (Fig.12(d)). NMI is measured based on different time units, which is chosen by FlockStream. All the comparisons are based on the real dataset KDD CUP99. Purity is measured based on various time units in which at least an attack exists.

The high quality of DenStream and MR-Stream benefits from their similar effective pruning strategies, which promptly get rid of the outliers while keep the potential clusters to form final clusters. In terms of high-dimensional data, PreDeConStream has better quality than PKS-Stream since it has a method to improve the offline phase of the algorithm. SDStream has acceptable quality in the initial time unit; however, the quality reduces specifically in time 375, when more attacks should be detected. The quality of rDenStream gradually improves since it makes classifier from clustering result. C-DenStream has a quality better than DenStream which shows that using the background knowledge for guiding the clustering improves the clustering quality. Even though FlockStream uses approximate nearest neighbor, it has higher quality compared with DenStream in terms of purity and normalized mutual information. D-Stream II has better quality compared with D-Stream I, since it considers the positional information about the data points inside grid. HDenStream has quality poorer than DenStream, which shows that it cannot improve DenStream to be used for data streams with categorical attributes.

We compare the algorithms' performance as well. Execution time is measured based on the number of data points (length of the stream) with respect to the time in seconds. We divide algorithms comparison based on the length of the stream: less than 50 000

($< 50\,\mathrm{K}$) data points and more than 50 000 ($\geqslant 50\,\mathrm{K}$) (shown in Figs. 12(e) and 12(f)) respectively to make fairly comparison. The comparison is based on the real dataset, KDD Cup99 Network Intrusion Detection. The algorithms, which are not in Fig.12 use another dataset or they are measured only on synthetic datasets or do not have any evaluation on their execution time.

It can be observed that SOStream has the longest execution time since finding the winner micro-cluster is time consuming. MR-Stream also has a long execution time even in smaller length of streams since the pruning method is time consuming. D-Stream I, DD-Stream, and D-Stream II have almost the same execution time; however D-Stream II has a better time performance than the others. D-Stream II uses tree structure for keeping the grid list that makes the algorithm faster. DenStream's execution time is similar to that of PKS-stream. It shows that PKS-Stream clusters high-dimensional data with acceptable execution time.

Table 6 compares the quality metrics, memory usage, time complexity, and application domain of the reviewed algorithms which will be discussed in the following subsection.

4.3 Density-Based Data Stream Clustering Algorithms' Applications

The literature on density-based clustering for data streams is usually centered around concrete methods rather than application contexts. Nevertheless, in this subsection, we would like to bring examples of several possible scenarios where density-based clustering can be used.

The density-based method has been used for earth environments for a long time[91]. Recently it has been utilized for medical purposes such as a pre-processing phase for prediction of Alzheimer's disease[92] and for skin cancer[93].

Real world applications may have any shape clusters and generate noisy data in some situations. Further-

Table 6. Evaluation on Density-Based Data Stream Clustering Algorithms

Name	Quality Metric	Memory Usage	Time Complexity	Application Domain
DenStream[3]	Purity	m	$O(m)$	Network intrusion detection system
StreamOptics[73]	-	m	$O(m \times \log(m))$	Environment monitoring
C-DenStream[74]	Rand Index	$m + m_c$	$O(m + m_c)$	Environment monitoring
rDenStream[75]	Purity	$m + S_{hb}$	$O(m) + T_h$	Network intrusion detection system
SDStream[41]	Purity	n_{sw}	N/A	Network intrusion detection system
HDenstream[76]	Purity	m	$O(m)$	Network intrusion detection system
SOStream[77]	Purity	m	$O(n^2 \log n)$	Network intrusion detection system
HDDStream[78]	Purity	m	$O(m) + O(m_p)$	Environment monitoring, network intrusion detection system
PreDeConStream[79]	Purity	m	$O(m) + O(m_{ip}) + O(m_{dp})$	Network intrusion detection system
FlockStream[67]	Purity, NMI	$m + n_{\text{agent}}$	$O(m) + O(n_{\text{agent}})$	Network intrusion detection system
DUC-Stream[43]	SSQ	n_d	$O(c_b)$	Network intrusion detection system
D-Stream I[4]	SSQ	g	$O(1) + O(g)$	Network intrusion detection system
DD-Stream[80]	N/A	g	$O(g^2)$	Network intrusion detection system
D-Stream II[32]	SSQ	$\log_{\frac{1}{\lambda}} g$	$O(\log \log_{\frac{1}{\lambda}} g)$	Network intrusion detection system
MR-Stream[33]	Purity	$g \times H$	$O(g \times H) + O(2^g \times H) + O(g \times \log(N))$	Network intrusion detection system
PKS-Stream[81]	Purity	$\log_k^g$	$O(\log k)$, $O(k)$	Network intrusion detection system
DCUStream[82]	Average quality of clusters	g	$O(g)$	Environment monitoring
DENGRIS-Stream[83]	N/A	g	$O(g)$	N/A
ExCC[84]	Purity	$g + S_{\text{Pool}} + S_{HQ}$	$O(g^{xk})$	Network intrusion detection system

Note: n: number of data points, m: number of micro-clusters in main memory, m_c: number of micro-cluster constraints, S_{hb}: size of historical buffer, T_h: time for processing historical buffer, n_{SW}: sliding window length, n_{agent}: number of agents, $O(m_p)$: number of potential micro-clusters, m_{ip}: number of inserted potential micro-clusters, m_{dp}: number of deleted potential micro-cluster, n_d: number of dense units, c_b: clustering bits, g: number of grids in the grid list, λ: decay factor, H: level of clustering, k: PKS-tree degree, S_{Pool}: size of pool for dense grids, S_{HQ}: size of hold queue for noise, xk: number of discovered clusters.

more, they do not require the number of clusters in advance. Since density-based clusterings have some abilities in their nature, they are applicable in different applications, such as:

• Network intrusion detection system: in this system, sensors capture all network traffic and the system analyzes the content of individual packets for malicious traffic[3].

• Environment observations: for example, in applications that are used to monitor flood, hurricane, tsunami, earthquake and forest fire detection[74].

• Medical systems: clustering medical data streams such as anatomical and physiological sensors, incidence records, health information systems, and patient monitoring system[92-93].

• Stock trade analysis: for example, clustering one million transaction records throughout the trading hours of a day[94].

• Social network analysis: clustering micro-blogging text streams (e.g., Twitter), in order to obtain temporal and geo-spatial features of real world events[95].

• Moving object applications: such as animal migration analysis, and vehicle traffic management[96].

Applications for patient monitoring and sensor networks in seismic studies, for example, work in bounded data space. Therefore, it is more preferable to use density grid-based methods. In these applications, a data point is either a member of a cluster or an outlier. Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All the clustering operations are performed on the grid structure, i.e., on the quantized space. The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space. As for these methods, if quality is the most important factor and time and memory are second and third factors respectively, MR-Stream is the best choice. In the case of the importance of execution time such as environmental observations, for example, the best choice is D-Stream II for Tsunami detection since it has the lowest execution time. However, the quality of grid-based methods is highly dependent on the granularity of the grid and further, defining the grid granularity to get the proper result is challenging.

Another important class of density-based algorithms over data streams is the density micro-clustering group. The quality of these algorithms is better than the grid-based methods. In the grid-based method, if we want to get more accurate results we have to fine the grids that lead to high time complexity. Density-based micro-clustering has better quality with reasonable time complexity. The micro-clustering method has limited memory usage, which depends on the number of micro-clusters. In the micro-clustering method, when the data points arrive they are assigned to the related micro-clusters and at the same time the outlier micro-clusters are removed based on the density threshold. Therefore, at any time the method can generate the clustering result by performing a clustering method. However, it has some limitations; finding the proper micro-cluster is time consuming. In some cases, because of the limitations in the memory usage, some real data is removed due to the appearance of an outlier.

However, choosing a proper density micro-clustering algorithm depends on the type of application. For example, in clustering GIS applications the best choice is C-DenStream because it considers the real world constraints such as the city, rivers, and highway networks. If the application needs limited processing time with good quality, FlockStream is a better choice rather than DenStream since it decreases the number of micro-clustering comparisons. If quality is the first priority rDenStream is the best choice; however, it needs more memory usage and execution time compared with the other algorithms. If there is any application with threshold settings (like similarity threshold or grid size) that are difficult to be manually done, SOStream is the best choice because it automatically adapts the thresholds. For detecting clusters in the recent data, such as identifying malicious attacks (clusters) in the current network traffic or recent stock trades on the stock exchange, SDStream and DENGRIS-Stream are more applicable since they cluster within the most recent portion of the stream.

Another aspect of choosing an algorithm is the type of data generated by the application, such as uncertain, high-dimensional or heterogeneous. Most of the algorithms in micro-cluster and grid groups only cover the continuous data. Therefore, if we have for example biomedical data with the categorical attributes, we have only ExCC in the grid group and HDenStream in the micro-cluster group. Furthermore, in some sensor-based applications the output of sensor networks is uncertain because of the noise in the sensor inputs or errors in wireless transmission. In this case, the algorithm has to cover the uncertain data as well. In this situation, the best choice is DCUStream. Moreover, if the data is high dimensional in its nature, we can choose between HDDStream and PreDeConStream in the micro-clustering group and PKS-Stream in the grid-based algorithms.

In summary, the task of choosing a proper density-based clustering algorithm depends on the kind of data produced as well as the application requirements such as limited time, high quality, high accuracy, handling

high noisy data and many other requirements, which are defined based on the application's objectives.

5 Conclusions and Open Issues

The density-based clustering method has attracted researchers due to its special characteristics, which has the ability to detect arbitrary shape clusters and to handle noise. Therefore, an extensive number of clustering algorithms on data stream adopt density method. In this paper, we surveyed a number of representative state-of-the-art algorithms on the density-based clustering algorithms over data streams. The main advantage of this paper is that it gives a comprehensive overview of the density-based data stream clustering algorithms and the evaluation metrics; further, the algorithms were divided into two basic categorizations, micro-cluster and grid algorithms that makes the investigation of the density-based clustering algorithms easier.

From the above detailed discussions of different types of density-based clustering algorithms, it can be easily claimed that the field of clustering data streams is wide open for researchers. Some of the possible research directions in this area are listed as follows.

• The performance of the algorithms are evaluated on the datasets by simulating the data streams. In the further research, the algorithm should be evaluated on real life datasets.

• As discussed earlier, evaluating the clustering quality and the algorithm's performance is an important issue, therefore developing a specific metric for evaluating evolving data streams on clustering algorithm is needed.

• We observed that, the clustering algorithms cannot deal with high-dimensional data well. The number of grids will be increased as the space dimensionality grows. They have low performance on very high-dimensional data. The biggest dimensionality is 40 in DenStream. Therefore, further research may involve handling the high-dimensional data in density-based data stream clustering and at the same time handling the other challenges.

• We noted that all the algorithms use DBSCAN in their offline phases, which in turn needs to set various parameters. As a result, using another type of density-based method for clustering data stream is a further research topic.

• The micro-clustering and grid-based clustering have their own advantages. Hence, developing algorithms using a hybrid method of micro-clustering and grid is an area of further research.

• These algorithms have been developed to handle stream data containing clusters of different shapes, sizes and densities. Nevertheless, only a small number of them can handle difficult clustering tasks (explained in [16]) without supervision.

• Only a limited number of algorithms can handle other kinds of data streams such as categorical or uncertain data. Extending the density-based clustering algorithms to handle all kinds of data is another point of research.

• We observed that most of the algorithms need a lot of parameters to be set. For instance, MR-Stream needs seven parameters, which is a difficult task to be done and it requires some experiences. Therefore, developing an algorithm with a fewer number of parameters to be set is another issue of research.

• With the emergence of big data, which is evolving and changing, data stream is a specific approach to deal with it. Therefore, proposing algorithms for evolving data streams becomes more important.

• Recently, the data stream clusterings prefer to use bio-inspired models. Therefore, proposing a hybrid clustering algorithm using a bio-inspired model and a density-based method, could be another issue of research.

References

[1] Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques (3rd edition). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[2] Gaber M, Zaslavsky A, Krishnaswamy S. Mining data streams: A review. *ACM SIGMOD Record*, 2005, 34(2): 18-26.

[3] Cao F, Ester M, Qian W, Zhou A. Density-based clustering over an evolving data stream with noise. In *Proc. the 2006 SIAM Conference on Data Mining*, April 2006, pp.328-339.

[4] Chen Y, Tu L. Density-based clustering for real-time stream data. In *Proc. the 13th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Aug. 2007, pp.133-142.

[5] Aggarwal C C (ed.). Data Streams: Models and Algorithms. New York, USA: Springer, 2007.

[6] Hahsler M, Dunham M H. Temporal structure learning for clustering massive data streams in real-time. In *Proc. the 11th SIAM Conference on Data Mining*, April 2011, pp.664-675.

[7] OĆallaghan L, Mishra N, Meyerson A *et al.* Streaming-data algorithms for high-quality clustering. In *Proc. the 18th Int. Conf. Data Engineering*, Feb. 26-Mar. 1, 2002, pp.685-694.

[8] Barbará D. Requirements for clustering data streams. *SIGKDD Explorations Newsletter*, 2002, 3(2): 23-27.

[9] Guha S, Meyerson A, Mishra N *et al.* Clustering data streams: Theory and practice. *IEEE Trans. Knowledge and Data Engineering*, 2003, 15(3): 515-528.

[10] Aggarwal C C, Han J, Wang J, Yu P S. A framework for clustering evolving data streams. In *Proc. the 29th International Conference on Very Large Data Bases*, Sept. 2003, pp.81-92.

[11] Ackermann M R, Lammersen C, Märtens M, Raupach C, Sohler C, Swierkot K. StreamKM++: A clustering algorithm for data streams. In *Proc. the 12th Workshop on Algorithm Engineering and Experiments*, Jan. 2010, pp.173-187.

[12] Ikonomovska E, Loskovska S, Gjorgjevik D. A survey of stream data mining. In *Proc. the 8th National Conference*

with International Participation, Sept. 2007, pp.19-25.

[13] Gaber M, Zaslavsky A, Krishnaswamy S. Data stream mining. *Data Mining and Knowledge Discovery Handbook*, 2010, pp.759-787.

[14] Babcock B, Babu S, Datar M, Motwani R, Widom J. Models and issues in data stream systems. In *Proc. the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, June 2002, pp.1-16.

[15] Jain A K, Dubes R C. Algorithms for Clustering Data. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[16] Jain A K. Data clustering: 50 years beyond K-means. *Pattern Recognition Letter*, 2010, 31(8): 651-666.

[17] Mahdiraji A. Clustering data stream: A survey of algorithms. *Int. J. Knowledge-Based and Intelligent Engineering Systems*, 2009, 13(2): 39-44.

[18] Amini A, Wah T, Saybani M *et al.* A study of density-grid based clustering algorithms on data streams. In *Proc. the 8th Int. Conf. Fuzzy Systems and Knowledge Discovery*, July 2011, pp.1652-1656.

[19] Amini A, Wah T Y. Density micro-clustering algorithms on data streams: A review. In *Proc. Int. Multiconf. Data Mining and Applications*, March 2011, pp.410-414.

[20] Amini A, Wah T Y. A comparative study of density-based clustering algorithms on data streams: Micro-clustering approaches. In *Lecture Notes in Electrical Engineering 110*, Ao S, Castillo O, Huang X (eds.), Springer, 2012, pp.275-287.

[21] Aggarwal C C. A survey of stream clustering algorithms. In *Data Clustering*: *Algorithms and Applications*, Aggarwal C C, Reddy C (eds.), CRC Press, 2013, pp.457-482.

[22] Han J, Kamber M. Data Mining: Concepts and Techniques (2nd edition). Morgan Kaufmann, 2006.

[23] MacQueen J. Some methods for classification and analysis of multivariate observations. In *Proc. the 5th Berkeley Symposium on Mathematical Statistics and Probability*, June 21-July 18, 1967, pp.281-297.

[24] Lloyd S P. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 1982, 28(2): 129-137.

[25] Guha S, Mishra N, Motwani R, O'Callaghan L. Clustering data streams. In *Proc. the 41st Annual Symposium on Foundations of Computer Science*, Nov. 2000, pp.359-366.

[26] Zhang T, Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases. In *Proc. the 1996 ACM SIGMOD International Conference on Management of Data*, June 1996, pp.103-114.

[27] Karypis G, Han E, Kumar V. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 1999, 32(8): 68-75.

[28] Kranen P, Assent I, Baldauf C, Seidl T. The clustree: Indexing micro-clusters for anytime stream mining. *Knowl. Inf. Syst.*, 2011, 29(2): 249-272.

[29] Wang W, Yang J, Muntz R R. STING: A statistical information grid approach to spatial data mining. In *Proc. the 23rd Int. Conf. Very Large Data Bases*, Aug. 1997, pp.186-195.

[30] Sheikholeslami G, Chatterjee S, Zhang A. Wavecluster: A wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal*, 2000, 8(3/4): 289-304.

[31] Agrawal R, Gehrke J, Gunopulos D, Raghavan P. Automatic subspace clustering of high dimensional data for data mining applications. *ACM SIGMOD Record*, 1998, 27(2): 94-105.

[32] Tu L, Chen Y. Stream data clustering based on grid density and attraction. *ACM Transactions on Knowledge Discovery Data*, 2009, 3(3): Article No. 12.

[33] Wan L, Ng W K, Dang X H *et al.* Density-based clustering of data streams at multiple resolutions. *ACM Trans. Knowledge Discovery from Data*, 2009, 3(3): Article No. 14.

[34] Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 1977, 39(1): 1-38.

[35] Dang X, Lee V, Ng W K *et al.* An EM-based algorithm for clustering data streams in sliding windows. In *Proc. the Int. Conf. Database Systems for Advanced Applications*, Apr. 2009, pp.230-235.

[36] Ester M, Kriegel H, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. the 2nd International Conference on Knowledge Discovery and Data Mining*, Aug. 1996, pp.226-231.

[37] Ankerst M, Breunig M M, Kriegel H P, Sander J. Optics: Ordering points to identify the clustering structure. *ACM SIGMOD Record*, 1999, 28(2): 49-60.

[38] Hinneburg A, Keim D A. An efficient approach to clustering in large multimedia databases with noise. In *Proc. the 4th KDD*, Sept. 1998, pp.58-65.

[39] Matysiak M. Data stream mining: Basic methods and techniques. Technical Report, RWTH Aachen University, 2012.

[40] Zhou A, Cao F, Qian W, Jin C. Tracking clusters in evolving data streams over sliding windows. *Knowledge and Information Systems*, 2008, 15(2): 181-214.

[41] Ren J, Ma R. Density-based data streams clustering over sliding windows. In *Proc. the 6th Int. Conf. Fuzzy systems and Knowledge Discovery* , Aug. 2009, pp.248-252.

[42] Charikar M, O'Callaghan L, Panigrahy R. Better streaming algorithms for clustering problems. In *Proc. the 35th Annual ACM Symp. Theory of Computing*, June 2003, pp.30-39.

[43] Gao J, Li J, Zhang Z, Tan P N. An incremental data stream clustering algorithm based on dense units detection. In *Proc. the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, May 2005, pp.420-425.

[44] Aggarwal C C, Han J, Wang J, Yu P S. A framework for projected clustering of high dimensional data streams. In *Proc. the 30th International Conference on Very Large Data Bases, Volume 30*, Aug. 29-Sept. 3, 2004, pp.852-863.

[45] Aggarwal C C, Han J, Wang J, Yu P S. On high dimensional projected clustering of data streams. *Data Mining and Knowledge Discovery*, 2005, 10(3): 251-273.

[46] Babcock B, Datar M, Motwani R, O'Callaghan L. Maintaining variance and k-medians over data stream windows. In *Proc. the 22nd ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems*, June 2003, pp.234-243.

[47] Ng W, Dash M. Discovery of frequent patterns in transactional data streams. In *Lecture Notes in Computer Science 6380*, Hameurlain A, Küng J, Wagner R *et al.* (eds.), Springer Berlin/Heidelberg, 2010, pp.1-30.

[48] Vitter J S. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 1985, 11(1): 37-57.

[49] Garofalakis M, Gehrke J, Rastogi R. Querying and mining data streams: You only get one look: A tutorial. In *Proc. the 2002 ACM SIGMOD Int. Conf. Management of Data*, June 2002, pp.635-635.

[50] Aggarwal C C, Yu P S. A survey of synopsis construction in data streams. In *Advances in Database Systems 31*, Aggarwal C C (ed.), Springer, 2007, pp.169-207.

[51] Garofalakis M N. Wavelets on streams. In *Encyclopedia of Database Systems*, Springer US, 2009, pp.3446-3451.

[52] Gilbert A C, Kotidis Y, Muthukrishnan S, Strauss M J. One-pass wavelet decompositions of data streams. *IEEE Trans. Knowl. and Data Eng.*, 2003, 15(3): 541-554.

[53] Gama J, Gaber M M (eds.). Learning from Data Streams – Processing Techniques in Sensor Networks. Springer, 2007.

[54] Rosset S, Inger A. KDD-cup 99: Knowledge discovery in a charitable organization's donor database. *SIGKDD Explorations Newsletter*, 2000, 1(2): 85-90.

[55] Hubert L J, Levin J R. A general statistical framework for assessing categorical clustering in free recall. *Psychological Bulletin*, 1976, 83(6): 1072-1080.

[56] Kaufman L, Rousseeuw P J. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley-Interscience, 2005.

[57] Wu J, Xiong H, Chen J. Adapting the right measures for K-means clustering. In *Proc. the 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, June 2009, pp.877-886.

[58] Rand W M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 1971, 66(336): 846-850.

[59] Zhao Y, Karypis G. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 2004, 55(3): 311-331.

[60] Dongen S. Performance criteria for graph clustering and Markov cluster experiments. Technical Report, National Research Institute for Mathematics and Computer Science, Stichting Mathematisch Centrum, Netherlands, 2000.

[61] Rosenberg A, Hirschberg J. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proc. the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, June 2007, pp.410-420.

[62] Meilă M. Comparing clusterings: An axiomatic view. In *Proc. the 22nd Int. Conf. Machine Learning*, Aug. 2005, pp.577-584.

[63] Rijsbergen C J V. Information Retrieval. Newton, MA, USA: Butterworth-Heinemann, 1979.

[64] Milligan G. A Monte Carlo study of thirty internal criterion measures for cluster analysis. *Psychometrika*, 1981, 46(2): 187-199.

[65] Pereira C M M, de Mello R F. A comparison of clustering algorithms for data streams. In *Proc. the 1st Int. Conf. Integrated Comp. Tech.*, May 31-June 2, 2011, pp.59-74.

[66] Manning C D, Raghavan P, Schtze H. Introduction to Information Retrieval. New York, NY, USA: Cambridge University Press, 2008.

[67] Forestiero A, Pizzuti C, Spezzano G. A single pass algorithm for clustering evolving data streams based on swarm intelligence. *Data Mining and Knowledge Discovery*, 2013, 26(1): 1-26.

[68] Bifet A, Holmes G, Pfahringer B *et al.* MOA: Massive online analysis, a framework for stream classification and clustering. *Journal of Machine Learning Research*, 2010, 11: 44-50.

[69] Holmes G, Donkin A, Witten I H. WEKA: A machine learning workbench. In *Proc. the 2nd Australian and New Zealand Conference on Intelligent Information Systems*, Nov. 29-Dec. 3, 1994, pp.357-361.

[70] Kranen P, Kremer H, Jansen T *et al.* Clustering performance on evolving data streams: Assessing algorithms and evaluation measures within MOA. In *Proc. the IEEE Int. Conf. Data Mining Workshops*, Dec. 2010, pp.1400-1403.

[71] Kremer H, Kranen P, Jansen T *et al.* An effective evaluation measure for clustering on evolving data streams. In *Proc. the 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, July 2011, pp.868-876.

[72] De Francisci Morales G. SAMOA: A platform for mining big data streams. In *Proc. the 22nd Int. Conf. World Wide Web Companion*, May 2013, pp.777-778.

[73] Tasoulis D K, Ross G, Adams N M. Visualising the cluster structure of data streams. In *Proc. the 7th International Conference on Intelligent Data Analysis*, Sept. 2007, pp.81-92.

[74] Ruiz C, Menasalvas E, Spiliopoulou M. C-DenStream: Using domain knowledge on a data stream. In *Proc. the 12th International Conference on Discovery Science*, Oct. 2009, pp.287-301.

[75] Liu L, Jing K, Guo Y *et al.* A three-step clustering algorithm over an evolving data stream. In *Proc. the IEEE Int. Conf. Intelligent Computing and Intelligent Systems*, Nov. 2009, pp.160-164.

[76] Lin J, Lin H. A density-based clustering over evolving heterogeneous data stream. In *Proc. the 2nd Int. Colloquium on Computing, Communication, Control, and Management*, Aug. 2009, pp.275-277.

[77] Isaksson C, Dunham M, Hahsler M. SOStream: Self organizing density-based clustering over data stream. In *Lecture Notes in Computer Science 7376*, Perner P (ed.), Springer Berlin Heidelberg, 2012, pp.264-278.

[78] Ntoutsi I, Zimek A, Palpanas T *et al.* Density-based projected clustering over high dimensional data streams. In *Proc. the 12th SIAM Int. Conf. Data Mining*, April 2012, pp.987-998.

[79] Hassani M, Spaus P, Gaber M M, Seidl T. Density-based projected clustering of data streams. In *Proc. the 6th Int. Conf. Scalable Uncertainty Management*, Sept. 2012, pp.311-324.

[80] Jia C, Tan C, Yong A. A grid and density-based clustering algorithm for processing data stream. In *Proc. the 2nd Int. Conf. Genetic and Evolutionary Computing*, Sept. 2008, pp.517-521.

[81] Ren J, Cai B, Hu C. Clustering over data streams based on grid density and index tree. *Journal of Convergence Information Technology*, 2011, 6(1): 83-93.

[82] Yang Y, Liu Z, Zhang J *et al.* Dynamic density-based clustering algorithm over uncertain data streams. In *Proc. the 9th Int. Conf. Fuzzy Systems and Knowledge Discovery*, May 2012, pp.2664-2670.

[83] Amini A, Teh Ying W. DENGRIS-Stream: A density-grid based clustering algorithm for evolving data streams over sliding window. In *Proc. International Conference on Data Mining and Computer Engineering*, Dec. 2012, pp.206-210.

[84] Bhatnagar V, Kaur S, Chakravarthy S. Clustering data streams using grid-based synopsis. *Knowledge and Information Systems*, June 2013.

[85] Ruiz C, Spiliopoulou M, Menasalvas E. C-DBSCAN: Density-based clustering with constraints. In *Proc. the 11th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, May 2007, pp.216-223.

[86] Yang C, Zhou J. HClustream: A novel approach for clustering evolving heterogeneous data stream. In *Proc. the 6th IEEE Int. Conf. Data Mining Workshops*, Dec. 2006, pp.682-688.

[87] Kohonen T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 1982, 43(1): 59-69.

[88] Bohm C, Kailing K, Kriegel H P, Kroger P. Density connected clustering with local subspace preferences. In *Proc. the 4th IEEE Int. Conf. Data Mining*, Nov. 2004, pp.27-34.

[89] Kennedy J F, Kennedy J, Eberhart R C. Swarm Intelligence. Morgan Kaufmann Pub, 2001.

[90] Shamshirband S, Anuar N, Kiah M *et al.* An appraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique. *Engineering Applications of Artificial Intelligence*, 2013, 26(9): 2105-2127.

[91] Sander J, Ester M, Kriegel H P, Xu X. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery*, 1998, 2(2): 169-194.

[92] Plant C, Teipel S J, Oswald A *et al.* Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease. *NeuroImage*, 2010, 50(1): 162-174.

[93] Mete M, Kockara S, Aydin K. Fast density-based lesion detection in dermoscopy images. *Computerized Medical Imaging and Graphics*, 2011, 35(2): 128-136.

[94] Yang D, Rundensteiner E A, Ward M O. Summarization and matching of density-based clusters in streaming environments. *Proc. VLDB Endow.*, 2011, 5(2): 121-132.

[95] Lee C H. Mining spatio-temporal information on microblogging streams using a density-based online clustering method. *Expert Systems with Applications*, 2012, 39(10): 9623-9641.
[96] Yu Y, Wang Q, Wang X, Wang H, He J. Online clustering for trajectory data stream of moving objects. *Computer Science and Information Systems*, 2013, 10(3): 1293-1317.

Amineh Amini received her B.Sc. degree in software engineering from Mashhad Azad University in 2001. She obtained her M.Sc. degree in the same field from Najafabad Azad University in 2005. She is a faculty member of Karaj Azad University from 2007. She is currently a Ph.D. candidate at Information Systems Department, Faculty of Computer Science and Information Technology, University of Malaya. Her main research interests include data stream mining and big data.

Teh Ying Wah received his B.Sc. and M.Sc. degrees from Oklahoma City University and Ph.D. degree in data mining from University of Malaya. He is currently an associate professor at Information Systems Department, Faculty of Computer Science and Information Technology, University of Malaya. His research focuses on data mining, text mining, document mining, big data, cloud computing and data stream.

Hadi Saboohi received his B.Sc. and M.Sc. degrees in software engineering from Iran in 2001 and 2005 respectively, and a Ph.D. degree in computer science from University of Malaya in 2013. His main research interests include Web intelligence, semantic Web services and data mining.