A Study of Density-Grid based Clustering Algorithms on Data Streams

Amineh Amini, Teh Ying Wah, Mahmoud Reza Saybani, Saeed Reza Aghabozorgi Sahaf Yazdi
Department of Information Science
Faculty of Computer Science and Information Technology
University of Malaya (UM)
50603 Kuala Lumpur, Malaysia
amini@siswa.um.edu.my

***Abstract*—Clustering data streams attracted many researchers since the applications that generate data streams have become more popular. Several clustering algorithms have been introduced for data streams based on distance which are incompetent to find clusters of arbitrary shapes and cannot handle the outliers. Density-based clustering algorithms are remarkable not only to find arbitrarily shaped clusters but also to deal with noise in data. In density-based clustering algorithms, dense areas of objects in the data space are considered as clusters which are segregated by low-density area. Another group of the clustering methods for data streams is grid-based clustering where the data space is quantized into finite number of cells which form the grid structure and perform clustering on the grids. Grid-based clustering maps the infinite number of data records in data streams to finite numbers of grids. In this paper we review the grid based clustering algorithms that use density-based algorithms or density concept for the clustering. We called them density-grid clustering algorithms. We explore the algorithms in details and the merits and limitations of them. The algorithms are also summarized in a table based on the important features. Besides that, we discuss about how well the algorithms address the challenging issues in the clustering data streams.**

***Index Terms*—Data streams, Density-based clustering, Grid-based clustering, Density-grid clustering**

I. Introduction

A data stream is an enormous amount of data which is generated nonstop at a rapid rate from sensors and mobile applications, log records, click-streams in web exploring, call detail records, email, blogging, twitter posts and etc. Therefore, in recent years mining of data stream has attracted researchers. Mining data streams is a real time process of extracting interesting patterns from high-speed data streams. Clustering is an important class in data stream mining in which analogous objects are categorized in one cluster. The clustering of data streams has the following characteristics:[1]

- Stream data may only read once.
- Any clustering algorithm must operate within resource constraints, as streaming data is infinite.
- The number and size of clusters are not known in advance.
- The data evolve by time therefore the underlying clusters can change over the time.
- 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.

Traditional clustering algorithms are not applicable in data streams. Several clustering algorithms are developed recently for clustering data streams [2], [3], [4], [5]. Some of these clustering algorithms apply a distance function for determining similarity between objects in a cluster. These algorithms can only discover spherical clusters. However, non-convex and interwoven clusters are seen in many applications.

Density-based algorithms are another major clustering algorithm that has been long proposed [6]. It can find arbitrarily shaped clusters and handles noises and yet is an one-scan algorithm that needs to examine the raw data only once. In density-based clustering algorithms, dense areas of objects in the data space are considered as clusters, which are segregated by low-density area (noise). Therefore, density-based method is an attractive basic clustering algorithm for data streams.

Another group of the clustering methods are grid-based clustering. The grid-based clustering method uses a multi resolution grid data structure. It forms the grid data structure by dividing the data space into a number of cells and perform the clustering on the grids. Clustering depends on the number of grid cells and independent of the number of data objects [7]. Gird-based method could be natural choice for data stream in which the infinite data streams map to finite grid cells. The synopsis information for data streams is contained in the grid cells.

Using density-based and grid-based methods, researchers have developed several hybrid clustering algorithms for data streams. In these algorithms, each data record in data stream maps to a grid and grids are clustered based on their density. Therefore, this paper intends to overview the grid and density based clustering algorithms on data streams. We called them the density-grid based clustering algorithms. The density-grid based algorithms are inspected. In addition, in the discussion part we mention the prominent challenging issues and discuss how the algorithms handle it.

The rest of the paper is organized as follows. Section II surveys related work. Section III provides an overview of of the density-grid based clustering algorithms. Section IV presents discussion, and Section V summarizes our study.

978-1-61284-181-6/11/$26.00 ©2011 IEEE

II. Related Work

In the literature, there have been several clustering algorithms proposed for data streams [2], [3], [4], [5]. Earlier clustering algorithms for data stream used a single-phase model that treated data stream clustering as a continuous version of static data clustering. These algorithms used divide and conquer schemes that partitioned data streams into segments and discovered clusters in data streams based on a k-means algorithm in finite space [2], [3] . CluStream [5] is another recent data stream clustering algorithm. It uses a two-phase scheme, which consists of an online component that processes raw data stream and produces summary statistics and an offline component that uses the summary data to generate clusters. Many recent data stream clustering algorithms are based on CluStream two-phase framework. One of limitations of CluStream and other similar algorithms that use k-means algorithm in their offline component is that it can neither reveal the cluster in arbitrary shape nor detect the noise and outliers.

Density-based clustering regard clusters as dense areas that are separated by low density area. Traditional density-based methods are DBSCAN, OPTICS and DENCLUE. DBSCAN[1][6] and its extension, OPTICS[2] [8], are both typical density-based methods that grow clusters according to a density-based connectivity analysis in spatial data set. DENCLUE[3] [9] is a method that clusters objects based on the analysis of the value distributions of density functions.

Grid-based clustering algorithms divide up the data space into finite number of cells that form a grid structure and perform clustering on the grid structure. The main advantages of grid-based clustering is fast processing time, since it process the grids and not all data points. Famous grid-based clustering approaches include STING [10], WaveCluster [11], OptiGrid [12] and CLIQUE[4] [13]. The main advantages of grid-based clustering is fast processing time, since it process the grids and not all data points. Famous grid-based clustering approaches include STING [10], WaveCluster [11], OptiGrid [12] and CLIQUE[5] [13].

By combining the density concepts or density-based clustering algorithms with other types of clustering, researchers propose different kinds of algorithms such as density micro-clustering algorithms [14] and density-grid based clustering algorithms. In density-grid algorithms the data are mapped to a grid and the grids are clustered based on density such as CLIQUE, WaveCluster, and DENCLUE, however, they are developed for large data sets. In this paper, we investigate the remarkable density-grid clustering algorithms which is developed specially for data streams including *DUCStream* [15], *D-Stream I* [16], *DD-Stream* [17], *D-Stream II* [18], and *PKS-Stream* [19].

[1]Density-Based Spatial Clustering of Applications with Noise
[2]Ordering Points To Identify the Clustering Structure
[3]DENsity-based CLUstEring
[4]CLustering InQUEst
[5]CLustering InQUEst

III. Density-grid Clustering Algorithms

In this section, we review the density-grid based clustering algorithms on data streams. In the density-grid based clustering algorithms, the data records are mapped into grid structure and cluster the grid based on density. Fig. 1 shows a framework for density-grid based clustering.

One of the common characteristics of reviewed algorithms is that the majority of them are based on CluStream framework which is developed by Aggrawal [5]. The algorithms have online and offline phase for clustering. In the online phase, the algorithm record summary information about the data records and the offline phase perform clustering on synopsis information. Algorithms on clustering data streams can be categorized into two groups: one-pass approach and evolving approach. The one-pass approach clusters data stream by scanning the data stream only once, and under assumption that the data objects arrived in chunks such as *DUCStream* [15] in this paper . In the evolving approach the data streams are considered to be changing over time. There are three kinds of window models in evolving approach include landmark window, sliding window, and fading window [20]. Another common characteristics of reviewed algorithms is that they are based on fading model which is the relevance of the data diminish over time.

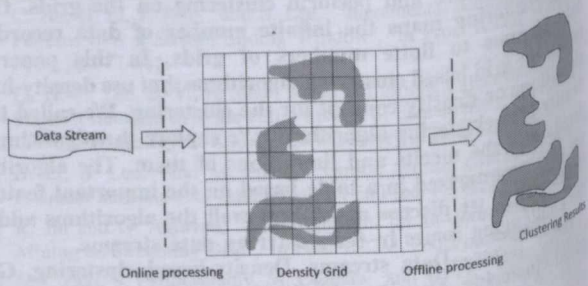

Fig. 1. Density-Grid based clustering Framework (Adapted from [18])

A. DUCStream

Gao et al. [15] developed an incremental single pass clustering algorithm for data streams which is referred to as *DUCStream*. It has the ability to detect evolving clusters in limited memory and time. *DUCStream* is a single pass clustering algorithms in which the data objects arrive in chunks. Each chunk fits in the main memory and contains a number of data points. It partitions the data space into units and keeps only the units with large number of data points. The density of a grid is defined by the number of the data points in the grid and if it is higher than density threshold, it is considered as a dense unit.

Local dense unit concept is introduced for determining which unit should be maintained. The local dense unit is a candidate for dense unit which may become a dense unit. Therefore, *DUCStream* keeps the entire local dense unit and chooses the dense unit between them to do the clustering. The clustering results are shown by bits to reduce the memory

requirements. "Clustering Bits" of a cluster shows the number of dense unit in the cluster by one and zero for dense and non dense unit. For clustering data stream, *DUCStream* identifies the clusters as a connected component of a graph in which vertices represent the dense units and the edges are related to common attributes between two vertices. It uses depth first search algorithm in graph. The time complexity and memory space of the *DUCStream* is low due to utilizing the bitwise clustering.

B. D-Stream I

Chen et al. [16] proposed a framework for clustering data streams, which is referred to as *D-Stream I*. The framework is based on the observation that many clustering algorithms on data streams cannot find clusters of arbitrary shape or handle the outliers. The idea is using density-grid approach for clustering data streams.

The algorithm procedure could be described as follows:

D-Stream I has online-offline components. The online component reads a new data record, maps each input data record into a density grid and update the characteristic vector which records summary information about the grid. The offline component clusters the density grids by merging two dense neighboring grids. A grid cluster is a connected grid group which has higher density than the surrounding grids. For removing outliers, *D-Stream I* periodically detect sporadic grids mapped to by outliers. It also run the offline component occasionally in order to adjust the clusters.

As *D-Stream I* is based on fading model [21] of data stream, it considers weight for each data record which is decreased as data record ages. The density of the grid is defined as sum of the weight of all data records in the grids. If no data record is added to this grid, the density of grid decrease over the time. Based on grid density, dense and sparse grid are introduced. Their differences referred to their density. Dense and sparse grid are defined as follows:

Definition 1: **Dense Grid** at time t, for a grid g, is defined as follows: $D(g,t) \geq \frac{C_m}{N(1-\lambda)} = D_m, C_m > 1$

Definition 2: **Sparse Grid** at time t, for a grid g, is defined as follows: $D(g,t) \geq \frac{C_l}{N(1-\lambda)} = D_m, 0 < C_l < 1$

Where C_m and C_l controls the threshold because the density value could not be more than $\frac{1}{(1-\lambda)}$ according to [16]. The $D(g,t)$ is the density of the grid, it is defined as $D(g,t) = \sum_{x \in E(g,t)} D(x,t)$ and $D(x,t) = \lambda^{t-T(x)} = \lambda^{t-T_c}$, where $\lambda \in (0,1)$ is a constant called the *decay factor* (capture the dynamic changes of a data stream). N is the number of grids.

D-Stream I puts the grids under consideration on the grid list as hash table and checks the list in special time intervals. If the density value of a grid become lower than special density threshold, it will be removed from the grid list.

Chen et al. showed that *D-Stream I* improves the time complexity and quality of clustering compared to CluStream. When a new data record arrives, *D-Stream I* needs to update the summary information of the grid which is the new data is mapped to it. Hence, the time complexity is $O(1)$.

C. DD-Stream

Jia et al. in [17] proposed a framework called *DD-Stream* for density-based clustering of data streams in grids. They developed an algorithm, DCQ-means, for improving quality of clustering by considering the border points of the grids. The framework is online-offline phase in which the online phase reads the new data records and maps to the grids and the offline phase perform the clustering on the grids using DCQ-means algorithm. DCQ-means algorithm extracts the data points on the border of the grid and joins boundary data points in the grid before adjusting the cluster. In DCQ-algorithm if the data is located in the border of two or more grids, it uses the most direct distance from the center of these grids to determine which data point belongs to which grid. In order to determine the distance between the data points and the neighboring grid, the *eigenvector* of the grid is defined for keeping a record of the central grid. If the distance of the grid is the same for more than one neighboring grids, the data point will be added to the grid with the higher density. If the neighboring grids have same density, the data point is added to the grids with the latest updates.

Jia et al. in [17] show that by extracting the border points, their algorithms has lower time complexity in comparison to CluStream and yet it has better scalability.

D. D-Stream II

In [18], Tu et al. improved the *D-Stream II* by considering the positional information about the data. They address this issue by introducing the grid attraction concept which shows to what extend the data in one neighbor is closer to another neighbor. It has attraction based mechanism to generate cluster boundaries. The clustering procedure of *D-Stream II* is the same as the *D-Stream I*, the only difference is that before merging two grids *D-Stream II* checks the grid attraction of two grids. If the grid attraction is higher than threshold, they are strongly correlated, and the grids will be merged. The grid attraction is defined as follows:

Definition 3: **Grid Attraction** The density attraction for a d-dimensional data record $x = (x_1 \dots x_d)$, if x map to a grid g centered at $c_1 \dots c_d$ and $g^{'} \in NB(g)$ be a neighboring grid of the g centered at $(g_1^{'} \dots g_d^{'})$, $c_k \neq g_k^{'}$ and $c_i \neq g_i^{'}$, $i = 1, \dots, d, i \neq k$. The attraction between x and $g^{'}$ is $attr_{ini}(x, g^{'}) = \prod_{i=1}^{d} b_i(x, g^{'})$ which denotes the attraction between x and h in the ith dimension is:

$$b_i(x, g^{'}) = \begin{cases} \frac{1+w}{2} & \text{if } |x_i - c_i| < r_i - \epsilon_i, \\ \frac{1}{2} + w(\frac{r_i}{2\epsilon_i} - \Delta_i) & \text{otherwise.} \end{cases}$$

$w = -1$ if $i \neq k$ and $w = 1$ if $i = k$ and $\Delta_i = \frac{x_i - c_i}{2 * \epsilon_i}$ $NB(g)$ is the set of neighboring grids whose center differs from g in at most one dimension.

By considering both density and attraction they generate better result for clustering. *D-Stream II* keeps the grid list in black red tree which improves the running time for lookup and update. The space complexity is $O(\log_{\frac{1}{\lambda}} N)$, and yet time complexity is $O(\log \log_{\frac{1}{\lambda}} N)$ for looking up in the grid list (N the total number of grids and λ is the decay factor).

TABLE I
SUMMARIZATION OF DENSITY-GRID CLUSTERING ALGORITHMS

Algorithm Name	Year	Objective	Data structure	Synopsis data	Time Complexity	Space Complexity
DUC-Stream	2005	Clustering data streams based on dense unit detection	Graph	Chunks in main memory	Size of cluster bits	a bit string with size of dense unit
D-stream I	2007	Real-time Clustering data streams	Hash Table	Summarized vector of data records in the grid	$O(1)$	$O(N)$
DD-stream	2008	Cluster border points based on distance function	Hash Table	Summarized vector of data records in the grid	$O(N)$	$O(N)$
D-stream II	2009	Handling positional information	Red black Tree	Summarized vector of data records in the grid	$O(\log\log_{\frac{1}{\lambda}} N)$	$O(\log_{\frac{1}{\lambda}} N)$
PKS-Stream	2011	Handling high dimensional data in grid	Pks-Tree	Summarized vector of data records in the grid	$O(\log N)$, worst case $O(N)$	Depend on size of tree

E. PKS-Stream

Ren et al. in [19] proposed an algorithm for clustering data streams based on grid density for high dimensional data streams. Most of existing density-grid clustering algorithms cannot handle high dimensional data stream efficiently, therefore Ren et al. in [19] proposed *PKS-Stream* algorithm based on grid density and Pks-tree. By using Pks-tree for clustering, the efficiency of storage and indexing are improved.

In the grid-based clustering approach, there are a lot of empty cells especially for high dimensional data. If all the grids are saved, it has an easy computation with high time complexity. If only non empty grids are saved, the algorithm looses the relation between grids. So Pks-tree is used for recording not only the non-empty cells, but also the relation between grids.

It is online-offline algorithm. In the online phase of *PKS-Stream* algorithm, the new data record in the data stream are continuously read and mapped to the related grid cells in the Pks-tree at all levels. If there is a grid cell for the data record, the data record is inserted. Otherwise, a new grid cell is created in the tree. In the offline phase, the clustering is started with non-empty cells in the leaf node level of Pks-tree. Firstly, it checks the density of the grid, if it is higher than a threshold, a new cluster is created. After that, the neighboring grids are checked if their density is higher than the threshold, the grids will put in the same cluster as the first grid

The sporadic grid is omitted in two situations: if the grid receive a few records over the time, or if many data records mapped to it but the density is reduced and is less than threshold. For improving the efficiency of the algorithms the empty grid cell is omitted using K-cover concept periodically. K-cover shows that the number of non empty grids in the neighboring of leaf node grids. The average computational complexity of *PKS-Stream* is $O(\log N)$ and in the worst case complexity is $O(N)$.

IV. DISCUSSION

We reviewed several important algorithms that use density-grid methods for clustering data streams. These algorithms mapped the input data into the grids and then applied density-based clustering algorithms or density concepts for clustering data streams. Table I summarizes the density-grid algorithms discussed in this paper.

The major challenging issues in the data stream clustering are represented in handling noise, limited time, limited space, handling evolving data and high dimensional data. While many methods have been proposed to address some of these issues, they are often unable to address these issues simultaneously.

Table II presents the reviewed algorithms in terms of addressing the above challenges. The algorithms have following solutions for addressing the challenges:

- Handling noisy data : Noisy data has great influence in formation of clusters , so each algorithms must be able to handle the noisy data. Most of density-grid reviewed algorithms has a technique which is developed to detect and remove sporadic grids mapped by outliers. The algorithms check the density of a sporadic grid periodically. If the density is lower than the special threshold it is considered as noise and will be removed from the grid list.
- Evolving data: It is not desirable to treat the data stream as a long sequence of static data since we are interested in the evolving temporal feature of the data stream. In the aforementioned algorithms of this paper, except the *DUCStream* algorithm, all the other algorithms consider the behavior of data streams as an evolving process over the time using fading window model. In fading window model, the density of each data record associates with a decay factor. The decay factor palaces more weights on the most recent data without discarding the historical information. *DUCStream* considers the behavior of data streams as the data objects arrive in chunks.
- High dimensional data: In high-dimensional data, the number of grids can be large. Therefore, how to handle

TABLE II
DENSITY-GRID CLUSTERING ALGORITHMS WITH CHALLENGING ISSUES

Density-Grid Algorithms	Limited Memory	Limited Time	Evolving Data	Handling Noisy Data	High Dimensional Data
DUCStream	×	×		×	
D-Stream I			×	×	
DD-Stream			×	×	
D-Stream II	×	×	×	×	
PKS-Stream			×	×	×

high dimensionality and improve scalability is a critical issue. Among the reviewed algorithms, only *PKS-Stream* is developed specially for high dimensional data. Since there are many empty grids in high dimensional data in grid clustering, the index structure Pks-tree is used to store the non-empty grid cells, which improves the efficiency of storage and indexing. Other algorithms assume that most grids are empty or contain few records so they do not have special method for handling the high dimensional data or they suffer from high time complexity.

- Limited Time: One of the prominent challenges in data stream clustering is forming the clusters in limited time to handle the high speed data stream. *D-Stream II* is the lowest time complexity compared to the other algorithms. It uses red black tree data structure for retaining the grid list which makes it faster to update and lookup.
- Limited Space: Due to the large volume of stream data, it is impossible to retain the information for every data record. Therefore, the reviewed algorithms partition the data space into discretized fine grids and map new data records into the corresponding grid. The raw data are not recorded since the algorithms only operate on the grids.They record the synopsis information about the data records in each grids. The density-grid algorithms reviewed in this paper use different summarization method and data structure for recording the synopsis data.

V. SUMMARY

Clustering data streams have several important applications in business, industry, and science. This paper reviewed the outstanding density-grid clustering algorithms for data stream. In density-grid algorithms, the data records are mapped into a grid and then the grids are clustered based on density. The summarization of reviewed density-grid clustering algorithms with their characteristic are shown in Table I.

Furthermore, important research challenges in data stream clustering were presented and discussed how the algorithms address these issues. Selection of the approaches is depend on the quality of addressing the research challenges. As it illustrated in Table II, the algorithms cannot handle the all challenging issue at the same time. The area of data stream clustering is still in its infancy. A number of open challenges still remain in density-grid algorithms particularly in handling high dimensional data and low time complexity.

REFERENCES

[1] L. Wan, W. K. Ng, X. H. Dang, P. S. Yu, and K. Zhang, "Density-based clustering of data streams at multiple resolutions," *ACM Transactions Knowledge Discovery Data*, vol. 3, no. 3, pp. 1–28, 2009.

[2] L. O'Callaghan, A. Meyerson, R. Motwani, N. Mishra, and S. Guha, "Streaming-data algorithms for high-quality clustering." Los Alamitos, CA, USA: IEEE Computer Society, 2002, p. 685.

[3] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams: Theory and practice," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 3, pp. 515 – 528, June 2003.

[4] D. Barbará, "Requirements for clustering data streams," *SIGKDD Explor. Newsl.*, vol. 3, pp. 23–27, January 2002.

[5] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proceedings of the 29th international conference on Very large data bases*. VLDB Endowment, 2003, pp. 81–92.

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*. AAAI Press, 1996, pp. 226–231.

[7] J. Han, *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[8] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," *SIGMOD Rec.*, vol. 28, pp. 49–60, June 1999.

[9] A. Hinneburg and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *KDD*, 1998, pp. 58–65.

[10] W. Wang, J. Yang, and R. R. Muntz, "Sting: A statistical information grid approach to spatial data mining," in *Proceedings of the 23rd International Conference on Very Large Data Bases*, ser. VLDB '97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 186–195.

[11] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wavecluster: a wavelet-based clustering approach for spatial data in very large databases," *The VLDB Journal*, vol. 8, pp. 289–304, February 2000.

[12] A. Hinneburg and D. A. Keim, "Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering," in *Proceedings of the 25th International Conference on Very Large Data Bases*, ser. VLDB '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 506–517.

[13] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," *SIGMOD Rec.*, vol. 27, pp. 94–105, June 1998.

[14] A. Amini and Y. W. Teh, "Density micro-clustering algorithms on data streams: A review," in *International Conference on Data Mining and Applications (ICDMA)*, Hong Kong, March 2011, pp. 410–414.

[15] J. Gao, J. Li, Z. Zhang, and P.-N. Tan, "An incremental data stream clustering algorithm based on dense units detection," *Lecture Notes in Computer Science*, vol. 3518, 2005.

[16] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '07. New York, NY, USA: ACM, 2007, pp. 133–142.

[17] C. Jia, C. Tan, and A. Yong, "A grid and density-based clustering algorithm for processing data stream," in *Proceedings of the Second International Conference on Genetic and Evolutionary Computing*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 517–521.

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

[19] C. H. Jiadong Ren, Binlei Cai, "Clustering over data streams based on grid density and index tree," *Journal of Convergence Information Technology*, vol. 6, pp. 83 – 93, 2011.

[20] A. Zhou, F. Cao, W. Qian, and C. Jin, "Tracking clusters in evolving data streams over sliding windows," *Knowledge and Information Systems*, vol. 15, pp. 181–214, May 2008.

[21] W. Ng and M. Dash, "Discovery of frequent patterns in transactional data streams," in *Transactions on Large-Scale Data- and Knowledge-Centered Systems II*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6380, pp. 1–30.