ONLINE CLUSTERING OF EVOLVING DATA STREAM BASED ON ADAPTIVE CHEBYCHEV DISTANCE

[1]MUSTAFA TAREQ, [2]ELANKOVAN A SUNDARARAJAN, [3]MASNIZAH MOHD

[1,2]Centre of Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, UKM Bangi, 43600 Selangor, Malaysia
[3]Center of Articial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, UKM Bangi, 43600 Selangor, Malaysia
E-mail: [1]P89127@siswa.ukm.edu.my, [2]elan@ukm.edu.my, [3]masnizah.mohd@ukm.edu.my

Abstract—Density-based methods have appeared as a valuable category for the clustering of evolving data streams. Although a number of density-based algorithms have recently been developed for the clustering of data streams, these algorithms are not without their issues. The quality of the clustering is dramatically reduced when the distance function is used. This paper presents an online clustering algorithm based on density called the Clustering of Evolving data streams based on the adaptive Chebychev distance (CEC). This algorithm keeps a summary of evolving data streams in the form of Core Micro-Clusters (CMCs), and also introduces a Chebychev distance to calculate the distance between the incoming data point and the centre of the CMCs in an online manner. The experimental results were compared with another online density-based clustering method called the CEDAS algorithm. The recommended method was then implemented to both real and artificial data streams using different quality metrics. The proposed method provides an effective solution for improving the quality of clusters.

Keywords - Chebyshev, Density-based clustering, Core Micro-Clusters, Evolving data stream.

I. INTRODUCTION

In recent years, the rapid growth of computer intelligence and the quarrying of data streams have made them noticeable as instruments, and the different devices that produce data streams are being utilized broadly [1]. Quite a number of data streams with excessive volumes and real-time high-speed features have been produced, making them relatively different from traditional static data [2-6]. Data streams are supposed to be treated as they arrive. Consequently, the excavation of data streams and real-time analyses have triggered a considerable number of researches [7-11]. Due to the use of data streams, clustering algorithms are receiving more consideration in data mining research.

The process of data stream mining involves the extraction of valuable patterns in real time from dynamic streaming data in a single scan alone, which can be very challenging [11-13]. However, the process of data stream clustering has been the subject of much attention due to its effectiveness in data mining [9, 12, 14]. Clustering involves the processing of data and the partitioning of the information or objects containedwithin it into subsets known as clusters [15-20]. The aim of this process is to classify similar objects into the same cluster, while objects in various clusters are dissimilar [21-26]. However, traditional clustering algorithms are unsutable due to the nature of data streams that are massive and evolve over time. Thus, improving and developing new clustering techniques has become crucial.

As far as is known, Existing online clustering algorithms based on density, such as CEDAS [22] use the Euclidean distance function to calculate the distance between an incoming data point and the centre of the CMCs or outliers, and the distance between a modified CMC centre and another CMC centre. The Euclidian distance cost function that is widely used in the literature does not capture the quality of the clustering, making it an inappropriate metric to be applied for maximizing both the homogeneity between different cluster as well as the heterogeneity within each clusters.

This paper proposed the clustering of evolving data streams, which is able to operate in a fully online way, based on the adaptive Chebychev distance (CEC). The CEC encapsulates the clustering information as CMCs, where clusters are produced. The recomended method can detect outliers and extract arbitrary-shaped clusters. Unlike the CEDAS algorithm, the proposed CEC algorithm uses the Chebychev distance function to design a reliable and accurate function to solve the clustering problem.

Various sample speeds were relied on to analyse the efficiency of the CEC algorithm. The results demonstrated that the proposed method significantly improves the clustering results compared to the CEDAS algorithm. The proposed algorithm has better clustering quality than the existing method.

The remainder of this paper is organized as follows. Section 2 presents the state-of-the-art models of previous studies, Section 3 defines the methodology associated with the CEC, Section 4 describes the experimental results and discussions, and Section 5 provides the conclusions drawn from this study.

II. RELATED WORK

A comparative study of K-means and Fuzzy clustering algorithm (FCM) with different distance measures such as Chebyshev, σ-distance and Chi-

square was proposed [27]. On the four data sets these algorithms are tested such as Disabetes, Contraceptive Method Choice, Statlog (Heart), and Liver Disorders. The FCM algorithm tested with cluster validity indices such as partition entropy and partition coefficient. The results show that the Chebyshev distance measurement is reported as a minimum partition entropy and a maximum partition coefficient compared to the other distance measurements. However, this algorithm needs hybrid evolutionary algorithms to enhance clustering efficiency.

For IoT streams, a density-based algorithm that may used is the hybrid density-based clustering for data stream (HDC-Stream) [26]. Rapid processing time is associated with the algorithm versus its predecessors, thus making it appropriate for a real-time application that pertains to IoT devices. The benefits of micro-clustering and density grid-based methods are applied to this method which can handle outliers and detect arbitrary-shaped clusters. This method is derived from offline and online phases. A new data record is constantly read by the online phase of HDC-Stream, and it also allows adding to an existing mini-cluster optionally or mapping with a grid. Outliers in the pruning step are periodically removed via HDC-Stream. The final clusters are produced on request via the offline phase by the user. Studies have shown that high-quality clustering results along with low time complexity are achieved when working with the data stream; however, clustering of multi-density data is infeasible and is associated with low memory efficiency [26].

Clustering online data streams into arbitrary shapes (CODAS) was proposed by [25] to allow the formation of arbitrary-shaped clusters via online clustering of data streams. Two key stages are included in this technique. The first stage involves generating a new micro-cluster when a specific value is reached by the number of data samples. In the second stage, micro-clusters are integrated into a global cluster. In this manner, arbitrary shapes, including shapes similar to concave clusters that are considered traditionally difficult, are created. The calculations required to combine the micro-clusters remain limited to hyperspheres. CODAS creates a cluster of high quality and can be scaled to multidimensional data streams. however, in CODAS the generated clusters do not evolve.Recently, the CODAS was improved in CEDAS [22] by introducing a simple linear aging process to handle the evolving data streams characteristics. This algorithm is the first fully online clustering for evolving data streams.This technique also includes two stages. In the first stage, micro-clusters are produced, or data are added to the current micro-clusters, followed by information adjustment. The micro-clusters' r_0 radius is fixed and must be kept as small as possible. Here, a simple linear ageing determination method is adopted by the CEDAS, thereby decreasing a micro-cluster's life, i.e. the micro-cluster is left to vanish entirely as an unused micro-cluster. We cannot only apply alternative ageing techniques, including those exponential types which leave insignificant energy lying with the micro-clusters, but also allow the likelihoods of being 'reborn' to become pertinent in the future by adding further data. Whenever a micro-cluster receives new data, its life is renewed. When no data are available at the receiving end, gradual fading of micro-clusters occurs. If this phenomenon persists extensively, then the life of the micro-clusters gradually reaches zero. Thus, they are discarded. Intersecting micro-clusters is the second stage, in which the micro-clusters are grouped into the kernel and shell regions. The grouping of micro-clusters by accounting for the kernel regions that intersect a micro-cluster shell allows automatic determination of edge micro-clusters. Micro-clusters that do not possess the minimum threshold persist as outlier. In this technique, each macro-cluster includes a graph that demonstrates micro-clusters intersection; for every micro-cluster, storing of the adjacency relations is performed as a characteristic of that micro-cluster. By applying the graph structure, this technique allows minimising the calculations needed to separate clusters when a cluster breaks or eventually dies. However, the CEDAS algorithm entails considerable computational time.

III. METHODOLOGY

The aim of the recomended CEC algorithm is to provide a cluster of high quality and changing data point features in a data stream. CEC summarizes the data points clustering information to produce the CMCs. Furthermore, the algorithm proposes Chebyshev distance function to calculate the distance between an incoming data point and the center of CMCs or outliers, and the distances between modified CMC center with another CMCs center.

A. Distance Measures

Clustering methods are based on evaluating data vector resemblance by calculating the distance between pairs. There is no prevalent distance metric that can best suit all applications for clustering. The following points are some significant distance measurement features: (i) The distance from point a to itself is always zero; (ii) Distance is always positive; (iii)There is always the same distance between a and b as between b and a; (iv) The distance from point a to point b cannot exceed the sum of the distance between a and another point c and the distance between c and b.

The measurement of Chebychev distance is based on the maximum difference in attributes. It is also known as the distance of Chessboard, or $L\infty$ norm, or the norm of Chebychev. It calculates the absolute size of the differences between data point and the CMC

center. The distance between vector x and centroid v of Chebyshev is determined by:

$$d(x, c) = \max|x_1 - x_2|, |y_1 - y_2| \quad (1)$$

B. Data Structures in CEC Algorithm

The recommended CEC algorithm is a totally online clustering algorithm based on density. It encapsulatesthe clustering information as CMCs, much like other density clustering algorithms. Macro clusters are produced on the basis of the interconnectionbetween the CMCs in a clustering graph. Two linked CMCs are associated with the same macro-cluster. The CEC algorithm outlines the "minimum density threshold" for distinguishing the outliers from the CMCs along with the "decay" parameter for determining the data's evolving property. A CMC possessing a density lower than the minimum level is an outlier. Every time, a new data point falls into CMC or helps create a fresh CMC. The CEC algorithm has two types of micro-clusters core and outlier. These micro-clusters are described in the following definitions.

Definition 1 Core micro-cluster (CMC): A CMC, at time t, is described as a set of close data points $X_1, X_2, X_3, \dots, X_{N_t}$ in a high density area wherein the local density N_t equals or exceeds the minimum limit $N_t \geq TH$.

Definition 2 Outlier micro-cluster: An outlier, at time t, is described as a set of one or higher number of data points $X_1, X_2, X_3, \dots, X_{N_t}$ in a low density area wherein the local density N_t is less than the minimum limit $N_t < TH$.

On the basis of the two definitions, it can be said that a CMC entails the local density to be above the user stated minimum benchmark for being a CMC, whereas the outlier micro-cluster scenario is reversed. The CMC aggressively partakes in the cluster graph and boasts of a positive macro-cluster ID. The outlier micro-clusters, however, do not partake in the cluster graph and lack a macro-cluster ID. Outside CMCs are categorised as noise and eliminated post detection.

C. Description of the Proposed Algorithm

There are four essential steps for the full algorithm, the description of each step is provided. For each data point, the algorithm executes these steps if needed: (i) Parameter Selection, (ii) Assign Core Micro-Cluster, (iii) Kill Cluster, (iv) Update Cluster Graph.

1. Parameter Selection

To perform the CEC algorithm, several parameters are required. Its parametric values depend on applications as follows:

a) Decay: This parameter denotes how many last samples we consider for processing at the current time t. If decay is set to N and we are at moment t, then we consider $t, t-1, t-2, \dots, t-N+1$ sample. It is used to update CMCs energy.

b) Radius: is defined as the minimum allowed distance for a sample from the center of CMC to belong to it. Otherwise, it belongs to either an outlier or it creates a new outlier.

c) Minimum Threshold: This parameter is the minimum number of data points that is required to form CMC or convert an existing outlier to CMC.

2. Assign Core Micro Cluster

This part of the algorithm used to assign CMC when arrive new data point. This algorithm use Chebychev distance function to calculate the distance between an incoming data point and the center of CMC and outliers. If the (d)distance of the data point and the nearest outlier or CMCs is less than radius is expressed by Eq. 2, then the CMC or outlier is updated. Otherwise, a new outlier is created. The algorithm check to determine if the update unit is an outlier and the locale density is equal to or exceeds the minimum threshold. Then, the algorithm will promote the outlier to a CMC. The new CMC must be assigned to the cluster of the nearest CMC in the shell, or a new cluster number is assigned to it.

$$d(X_i, C) < R \quad (2)$$

3. Kill Cluster

This part of the algorithm minimizes the life of the CMCs (Section 3.2.2) and removes them when their life is below zero. The life of CMCs minimizes by the $\frac{1}{\text{Decay}}$ amount. All edges that refer to CMC will also be removed, and the total number of CMCs will be decreased.

4. Update Cluster Graph

Like other density-based such as CEDAS and CODAS, a clustering graph is maintained in order to create an online macro-cluster. The clustering graph make any change if either:

Case1. A CMC center location has been shifted.

Case2. A CMC is moved or when a new CMC arrives.

Case3. A CMC has died and been removed.

The edge list of CMCs can be changed in the above cases; therefore, the number of macro-clusters need to be updated accordingly. First calculate the distances between modified CMC and all others using Chebychev distance function. the changes are made to any CMC that has been modified as a CMC that has recently reached the threshold or moved its center location. In either case, the graph edges for this CMC may have changed. If the edge list has changed then the new graph has its macro-cluster number set to a new value.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A series of experiments has been analyses and compares the performance of the CEC algorithm with that of the CEDAS algorithm [22]. CEC algorithm is implemented using MATLAB R18a, and their

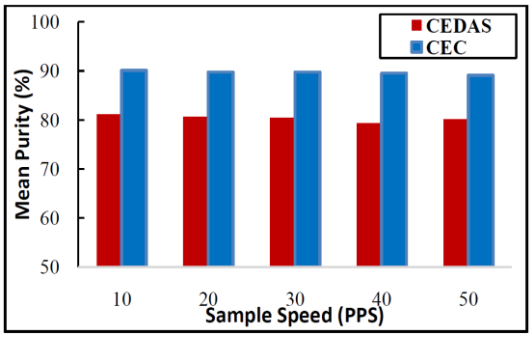
performances are evaluated on a PC with an intel i5 CPU with 16GB RAM. Cluster accuracy and purity parameters is used to defined the cluster quality [22, 25]. In addition to purity and accuracy parameters, the percentage of data points assigned to clusters is an important measure of cluster quality, considering the possibility of generating highly accurate and pure clusters from a few data points. A data streams called KDDCUP'99 [22, 24, 28-31]and DS1 [32-34] has been used to measure cluster quality. The clustering parameters are set as Minimum Threshold 4 data point, Radius = 0.05, Decay = 1000 data points. This experiment aims to verify the purity and accuracy across sample speed and time.

A. Cluster Purity

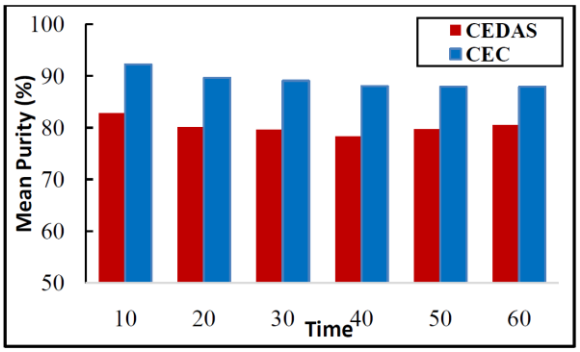
Is the class most frequently divided by the number of data points in that cluster. Purity may be used for clustering analysis of evolving data stream in various studies and is defined as:

$$\text{Purity} = \frac{\sum_{i=1}^{N} n_i^d}{n_i} \times 100\% \qquad (3)$$

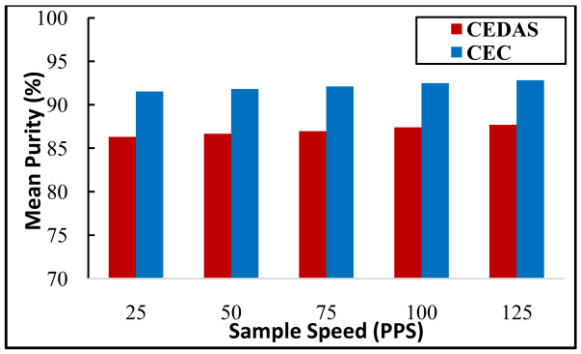
where n_i^d is the dominant class sample, n_i holds the sample numbers that a cluster contains, and N is the number of clusters. The comparison between the CEC and CEDAS algorithms in terms of average clustering purity is depicted in Figure 1. We test them on the same actual KDD CUP'99 dataset which is exactly the same test dataset used in [22]. Eq (3) is used for measuring the purity of KDDCUP'99 and DS1 clustering using the proposed CEC algorithm for a number of sample speeds and time periods. The CEC algorithm show the maximum purity at most of the sample speed and time. The following reasons contribute to the improved performance of the CEC algorithm is the online using Chebychev distance function to calculate the distance from an incoming data point and CMC centre, based on the maximum difference in attributes. This outcome shows that the predominant clusters are appropriately empowered by nearly all samples. The purity of CEC algorithm in DS1 data set is above 87% at all sample speed and time period. While, the purity of the KDDCUP'99data set is above 90% at all sample speed, in time period 10 CEC reaches 75% and increases respectively. Figure 1 emphasizes the improvement in purity of CEC algorithm.



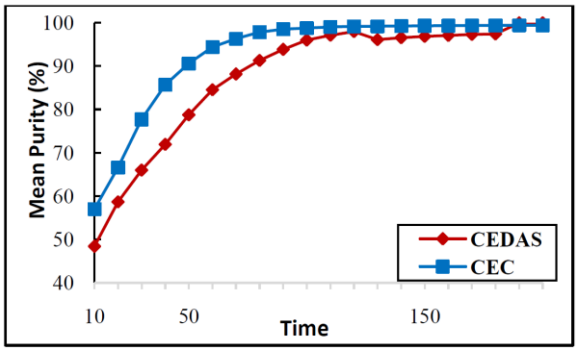
(a) DS1 dataset evaluation purity with sample speed.



(b) DS1 dataset evaluation purity with time.



(c) KDDCUP'99 dataset evaluation purity with sample speed.



(c) KDDCUP'99 dataset evaluation purity with time.

Figure 1. Clustering purity evaluations (a, b) DS1 dataset purity with sample speed and time, (c, d) KDDCUP'99 dataset purity with sample speed and time.

B. Cluster Accuracy

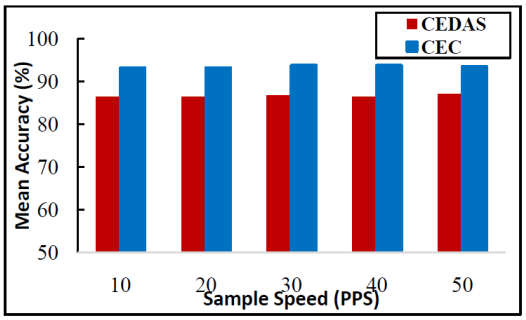
For data stream-clustering evaluation, accuracy is used and can be defined as the number of samples in a cluster that belongs to that cluster and does not belong to any other cluster.

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} n_i^d}{\sum_{i=1}^{N} n_i} \times 100\% \qquad (4)$$

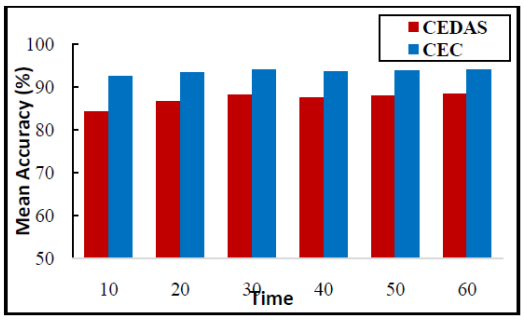
where n_i^d is the dominant class sample, n_i holds the sample numbers that a cluster contains, and N is the number of clusters.

Eq (4) is used for measuring the purity of KDDCUP'99 and DS1 clustering using the proposed CEC algorithm for a number of sample speeds and time periods. The comparison between the CEC and CEDAS algorithms in terms of average clustering

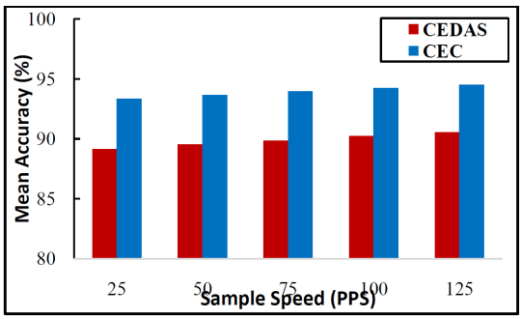
accuracy is depicted in Figure 2. CEC algorithm shows the highest clustering accuracy than CEDAS because of the same improvement in cluster purity (Section 4.1). The accuracy of CEC algorithm in DS1 data set is above 90% at all sample speed and time period. While, the accuracy of the KDDCUP'99data set is above 90% at all sample speed, in time period 10 CEC reaches 66% and increases respectively. This outcome shows that the data points in CEC algorithm are more accurate than in CEDAS.



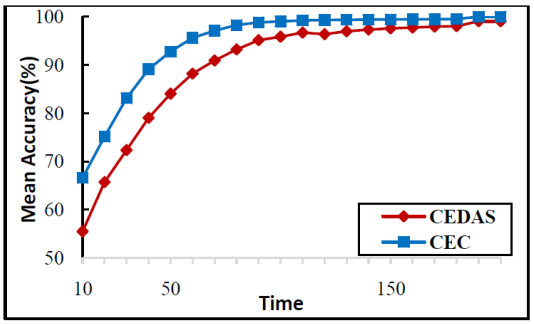
(a) DS1 dataset evaluation accuracy with sample speed.



(b) DS1 dataset evaluation accuracy with time.



(c) KDDCUP'99 dataset evaluation accuracy with sample speed.



(c) KDDCUP'99 dataset evaluation accuracy with time.

Figure 2. Clustering accuracy evaluations (a, b) DS1 data set accuracy with sample speed and time, (c, d) KDDCUP'99 data set accuracy with sample speed and time.

V. CONCLUSION

This article introduces a fully online clusterig algorithm based on density called CEC, which produces arbitrary shaped clusters in online mannar. The main concept of our proposed algorithm is using Chebychev distance function, based on the maximum difference in attributes.Chebychev is used to calculate the distance between arriving data point and the CMC data center. CEC demonstrates its ability to produce high-quality clusters in practical network attacks in the KDDCUP'99 and DS1 data sets. The outcome indicates that the clusters produced in CEC are purer and more accurate. Based on a extensive evaluation of real and artificial data set using quality metrics of purity and accuracy.

In this study, the proposed clustering algorithm based on density is a fully online clustering algorithm for evolving data streams into arbitrary shaped clusters with high purity and accuracy. This algorithm is scalable to handle a high-dimensional data stream. Our future work will focus on the improvement of the computational complexity of CEC in evolving data stream for high-dimensional datasets.

ACKNOWLEDGMENTS

This work is partly supported by the Ministry of Higher Education Malaysia, grant number "FRGS/1/2018/ICT04/UKM/02/1".

REFERENCES

[1] Toshniwal, Durga. "Clustering techniques for streaming data-a survey." In 2013 3rd IEEE International Advance Computing Conference (IACC), pp. 951-956. IEEE, 2013.

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

[3] Y. Tao, X. Xiao, and R. Cheng, "Range search on multidimensional uncertain data," ACM Transactions on Database Systems (TODS), vol. 32, no. 3, p. 15, 2007.

[4] A. Amini, T. Y. Wah, and H. Saboohi, "On density-based data streams clustering algorithms: A survey," Journal of Computer Science and Technology, vol. 29, no. 1, pp. 116-141, 2014.

[5] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," Information Fusion, vol. 37, pp. 132-156, 2017.

[6] K. Wankhade, T. Hasan, and R. Thool, "A survey: Approaches for handling evolving data streams," in 2013 International Conference on Communication Systems and Network Technologies, 2013, pp. 621-625: IEEE.

[7] L. O'callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani, "Streaming-data algorithms for high-quality clustering," in Proceedings 18th International Conference on Data Engineering, 2002, pp. 685-694: IEEE.

[8] 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, 2003.

[9] D. S. Suresh and W. Vinod, "Survey Paper on Clustering Data Streams Based on Shared Density between Micro-Clusters," 2017.
[10] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," Annals of Data Science, vol. 2, no. 2, pp. 165-193, 2015.
[11] D. Dell'Aglio, T. Eiter, F. Heintz, and D. Le Phuoc, "Ensemble learning for data stream analysis: Asurvey," Semantic Web, vol. 10, no. 3, pp. 453-455, 2019.
[12] S. Ding, F. Wu, J. Qian, H. Jia, and F. Jin, "Research on data stream clustering algorithms," Artificial Intelligence Review, vol. 43, no. 4, pp. 593-600, 2015.
[13] A. Amini, "An adaptive density-based method for clustering evolving data streams," University of Malaya, 2014.
[14] H.-L. Nguyen, Y.-K. Woon, and W.-K. Ng, "A survey on data stream clustering and classification," Knowledge and information systems, vol. 45, no. 3, pp. 535-569, 2015.
[15] D. B. Setyohadi, A. A. Bakar, and Z. A. Othman, "Optimization overlap clustering based on the hybrid rough discernibility concept and rough K-Means," Intelligent Data Analysis, vol. 19, no. 4, pp. 795-823, 2015.
[16] F. Boobord, Z. Othman, and A. A. Bakar, "A WK-means Approach for Clustering," Int. Arab J. Inf. Technol., vol. 12, no. 5, pp. 489-493, 2015.
[17] M. Mousavi and A. A. Bakar, "Improved density based algorithm for data stream clustering," Jurnal Teknologi, vol. 77, no. 18, 2015.
[18] B. Aubaidan, M. Mohd, and M. Albared, "Comparative study of k-means and k-means++ clustering algorithms on crime domain," Journal of Computer Science, vol. 10, no. 7, pp. 1197-1206, 2014.
[19] M. Mohd et al., "Optimal initial centroid in k-means for crime topic," Journal of Theoretical and Applied Information Technology, vol. 45, no. 1, pp. 19-26, 2012.
[20] M. Mousavi, A. A. Bakar, and M. Vakilian, "Data stream clustering algorithms: A review," Int J Adv Soft Comput Appl, vol. 7, no. 3, p. 13, 2015.
[21] A. Madraky, Z. A. Othman, and A. R. Hamdan, "Analytic methods for spatio-temporal data in a nature-inspired data model," International Review on Computers and Software, vol. 9, no. 3, pp. 547-556, 2014.
[22] R. Hyde, P. Angelov, and A. R. MacKenzie, "Fully online clustering of evolving data streams into arbitrarily shaped clusters," Information Sciences, vol. 382, pp. 96-114, 2017.
[23] B. Wu and B. M. Wilamowski, "A fast density and grid based clustering method for data with arbitrary shapes and noise," IEEE Transactions on Industrial Informatics, vol. 13, no. 4, pp. 1620-1628, 2016.
[24] A. Amini, H. Saboohi, T. Herawan, and T. Y. Wah, "MuDi-Stream: A multi density clustering algorithm for evolving data stream," Journal of Network and Computer Applications, vol. 59, pp. 370-385, 2016.
[25] R. Hyde and P. Angelov, "A new online clustering approach for data in arbitrary shaped clusters," in Cybernetics (CYBCONF), 2015 IEEE 2nd International Conference on, 2015, pp. 228-233: IEEE.
[26] A. Amini, H. Saboohi, T. Ying Wah, and T. Herawan, "A fast density-based clustering algorithm for real-time internet of things stream," The Scientific World Journal, vol. 2014, 2014 b.
[27] O. M. Jafar and R. Sivakumar, "A comparative study of hard and fuzzy data clustering algorithms with cluster validity indices," in Proceedings of International Conference on Emerging Research in Computing, Information, Communication and Applications, 2013, pp. 775-782.
[28] A. Forestiero, C. Pizzuti, and G. Spezzano, "A single pass algorithm for clustering evolving data streams based on swarm intelligence," Data Mining and Knowledge Discovery, vol. 26, no. 1, pp. 1-26, 2013.
[29] C. Isaksson, M. H. Dunham, and M. Hahsler, "SOStream: Self organizing density-based clustering over data stream," in International Workshop on Machine Learning and Data Mining in Pattern Recognition, 2012, pp. 264-278: Springer.
[30] L. Tu and Y. Chen, "Stream data clustering based on grid density and attraction," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 3, no. 3, p. 12, 2009.
[31] F. Cao, M. Estert, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in Proceedings of the 2006 SIAM international conference on data mining, 2006, pp. 328-339: SIAM.
[32] M. Hahsler and M. Bolaños, "Clustering data streams based on shared density between micro-clusters," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 6, pp. 1449-1461, 2016.
[33] C. F. Eick, B. Vaezian, D. Jiang, and J. Wang, "Discovery of interesting regions in spatial data sets using supervised clustering," in European Conference on Principles of Data Mining and Knowledge Discovery, 2006, pp. 127-138: Springer.
[34] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," in Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on, 2004, pp. 576-584: IEEE.

★★★