DATA STREAM CLUSTERING BASED ON A COMPLETE BINARY TREE

K. Shyam Sunder Reddy
Department of IT, Vasavi College of Engineering, Hyderabad.

***Abstract*:** Data stream clustering is a significant method for handling huge quantity of dynamic data. In recent days data streams are generated through numerous applications which increases the complexity of handling huge data. To reduce such kind of overhead several techniques and algorithms are proposed. This paper proposes an algorithm called Complete Binary Tree Mapping Micro Clusters (CBTMMC) which converts the complete dataset into the complete binary tree, in which the left sub tree contains high density data points and the right sub tree contains low density data points. The maximum of the outliers are obtained by right sub tree leaf node. So clustering and detecting outlier through proposed approach is simple in less time and space consuming. The experiment results shows that the proposed system provides high performance than the existing systems on the real-time datasets.

Keywords: Data stream, Density-based clustering, Micro-clustering, Complete Binary Tree.

1. INTRODUCTION

In the recent years, data analyzing techniques are growing enormously due to massive amount of data. Potentially unlimited quantity of data is the cause of a fresh field of research called data stream mining [22] [23] [24] [25]. The term data stream describes the continuous data which is produced from numerous applications. In the field of data stream mining [17], clustering active data streams [18] [21] have additional constrains to static batch clustering. Data Streams can possibly be real time and infinite, so clustering requires to be performed quickly in a single pass of data and the discovered clusters need to be summarized in an expressive way. Data is usually clustered based on different criteria; e.g. similarity and homogeneity. Data stream clustering is done through two phase process: (1) online part that converts data into micro clusters, (2) offline part which re-cluster the micro-cluster. The data stream clustering algorithm must have the following needs [1]:

- Algorithm must be computed with less memory and CPU usage.
- The clusters should be changed based on the data.

The novel Complete Binary Tree Mapping Micro Clusters (CBTMMC) algorithm is used to convert data into complete binary tree, and the micro-clusters are formed from the tree. The micro-clusters are re-clustered during offline part through the density among the micro-clusters and density information between the micro-clusters, which provides the efficient clusters.

The major challenge in stream based clustering is the huge amount of data that cannot be saved or processed in memory and the speed of stream which is so fast, but nowadays the technologies do not have adequate time to process the incoming data. To handle these challenges, an online/offline component based structure was introduced in CluStream [1] algorithm which was very effective, and most of the stream clustering algorithms after CluStream follow the same structure.

Partition based clustering techniques [1] [2] were proposed for clustering stream data, and these methods cannot detect arbitrary shape clusters. DBSCAN [3] estimates the density around each data point by calculating the density of points in an eps-neighborhood and applies Minimum Point (MinPts) thresholds to detect the core, border and noise points. In the next stage, core points are combined and form a cluster if they are density-reachable. It groups neighboring data points into clusters depending on the local density estimates rather than proximity among objects. Density-based methods contain noise tolerance, and can determine non-convex clusters. Alike hierarchical and partitioning methods, density-based methods encounter complications in high dimensional spaces due to inherent sparse feature space, which in turn, drops any clustering affinity.

Some illustrative density-based clustering algorithms are: Clustream [1], DBSCAN [3], OPTICS [4], DENCLUE [5], DenStream [6], CDenstream [7], rDenStream [8], D-Stream [9], Incremental DBSCAN [10], MRStream [11] and strAP [12], C-DBSCAN[13]. The above specified methods are efficiently cluster the data through the density. But density between micro clusters during re-clustering was not considered in the above mentioned techniques. This technique was proposed in the DBSTREAM [14] algorithm that uses shared density graph for performing clustering. The density related details in this graph is then utilized for re-clustering depends on exact density among nearby micro-clusters. But the performance measure of this approach is not efficient and also this technique leaves lots of outlier, which discards the maximum of data stream. So to overcome such kinds of difficulties the proposed method, uses density based clustering through complete binary tree.

2. MICRO-CLUSTER FORMATION

The micro-clusters contain numerous data points which are obtained from the data streams. The data stream techniques cannot store all the incoming points (data objects) because they have infinite size also the range and memory is restricted. The micro-clusters are formed through Complete Binary Tree Mapping Micro Clusters (CBTMMC) algorithm which convert data points into the micro-cluster through calculating the density among each data points, Building complete binary tree, obtain the core, and noise points from the data streams. The figure 1 describes the exact scenario of micro cluster formation. The micro-clusters are formed depends on the eps and MinPts values.

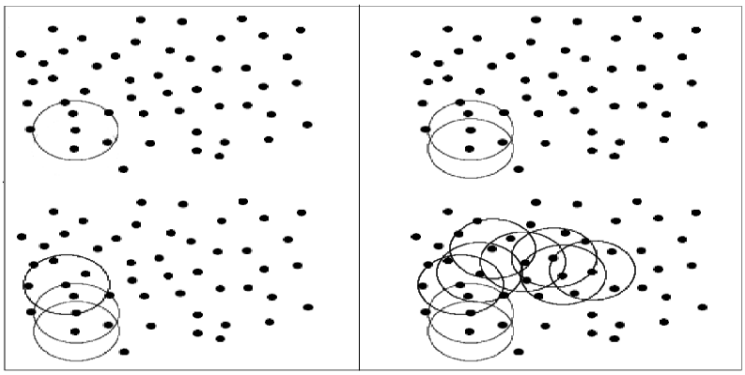

Figure 1 Micro Clusters (Khan et al., 2009)

2.1 Building Complete Binary Tree

Complete Binary tree is a finite set of n nodes, which may be an empty set, or composed of one root node and two sub-trees, known as the left and right sub-tree of the root, respectively. The data stream generated from time t is converted into the complete binary tree. The random data point (A_1) from the data stream is utilized as a root node, depends on the eps (radius) a circle is generated in the data stream. It checks, the obtained root node density within the circle. Then the other random data points (A_2) are chosen, again the circle is drawn around data points, if the obtained weight of A_2 is greater than the A_1, then the A_1 is replaced through A_2 from the position of root. Meanwhile, the A_1 data point is moved to left sub tree of the complete binary tree. If the other Random point A_3 occupies the place of A_2, then the A_1 moved to right sub tree and A_2 moved to left sub tree.

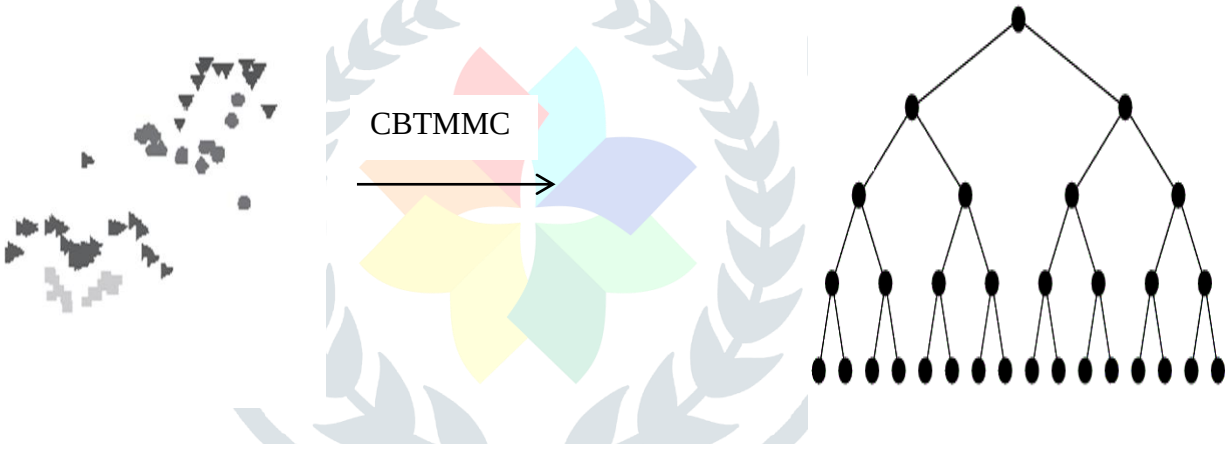

Figure 2 Building Complete Binary Tree

Definition 2.1: The complete binary tree T_r is formed by $T_r = (W_{DP} > WR_{DP,}\ WR_{DP} < WN_{DP} => N_{DP}=R_{DP})$, where W_{DP} is the weight of data points, WR_{DP} is the weight of root data point, WN_{DP} is the weight of new data point, R_{DP} is the root data point and N_{DP} is the new data point.

2.2 Micro-Cluster Formation from Complete Binary Tree

Several tree based clustering techniques have been proposed for data streams [16] [19]. Forming micro-cluster from complete binary tree is a simple technique; it can easily cluster the data stream depending on the high and low density values. The high density data points are placed at left sub tree of the complete binary tree. Also the low density data points are placed at the right side of completed binary tree. The outliers are obtained from the right sub tree leaf nodes. From the above mentioned procedures it is clear that left sub tree contains high density data. But when depth of the tree increases there is a chance to get more low density data in left sub tree also. To overcome such kind of issue the proposed technique used the super head based clustering technique, which cluster the data depends on the information obtainable in super head.

Definition 2.2 Micro cluster from complete binary tree is formed through $MC_i = L_{ST}$, where the L_{ST} is left sub tree of the complete binary tree and MC_i is the i^{th} micro-cluster.

2.2.1 Super Head Based Micro-Cluster Formation

Leader-based clustering technique was presented in Hardigan [15]. Clustering through left sub tree is an efficient technique in complete binary tree based clustering. But when depth of the tree increase and may cause improper result. So clustering the entire left sub tree into the single micro-cluster is not an efficient process. To overcome such kind of difficulty the super heads based clustering techniques are used. Super heads are the parent node of left and right sub tree. Initially, parent node and its child nodes are clustered into the micro-cluster. The obtained micro-cluster is then compared with other parent node, if they have equivalent density

neighbor (density information of the each data points are stored in the root) then parent node and its sub tree are added into the existing micro-cluster. If they don't have equivalent density neighbor then the specific sub tree form separate cluster. In the same way the low density right sub tree based micro-cluster is formed. But it does not cluster the leaf nodes because of their low density. They also cluster the low dense data point through the parent node (super heads).

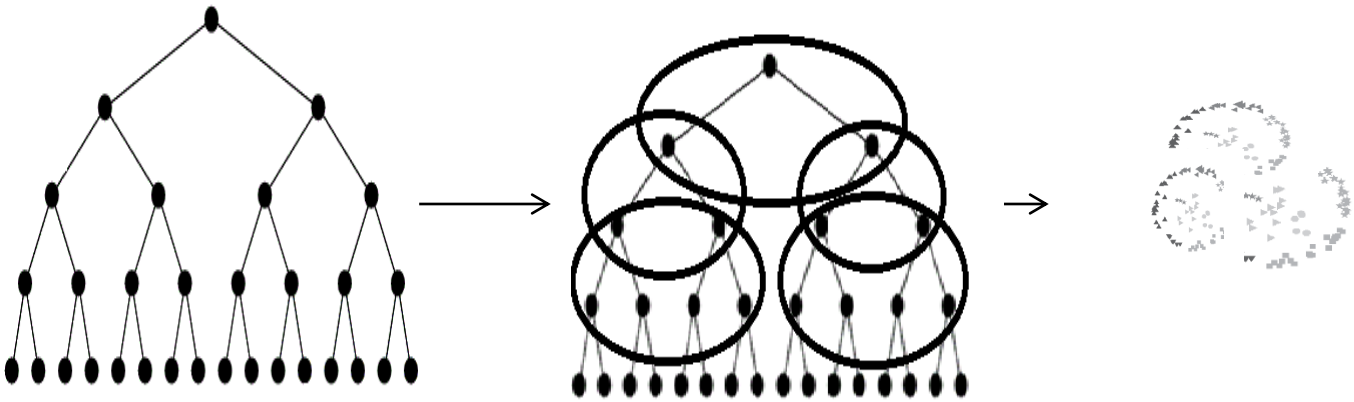

Figure 3 Super Head based Clustering

The above figure 3 depicts the super head based clustering technique. The complete binary trees are clustered, depending on super head, which is parent node. Based on the number of parents the density between the micro-clusters is varied.

Definition 2.2.1: In the initial stage the complete binary trees are clustered by, $C_{LSTi} = P_{DPi} + (C_{DPR+} C_{DPL})$, then if $WC_{LSTi} >= WP_{DPi+1}$. Where C_{LST} is the obtained micro-cluster value, $_{DPi} + (C_{DPR+} C_{DPL})$ is the parent and child data points (left and right). WC_{LSTi} is the weight of previous micro cluster. WP_{DPi+1} is the weight of parent data point.

2.2.2 Algorithm 1: Micro Cluster Formation

W_{DP} ⟵ weight of data point
R_{DP} ⟵ root data point
WR_{DP} ⟵ density of R_{DP}
L_{ST} ⟵ Left sub tree
L_{STDP} ⟵ Left sub tree data point
R_{STDP} ⟵ Right sub tree data point
$P_{DP} + C_{DPS}$ ⟵ parent and child data points
W_{STi} ⟵ weight of the sub tree
D_{Mci} ⟵ Density between the micro-cluster i and i+1
User Specified parameters:
r - Fixed radius
TR_{STLF} - cleanup interval
Function: CBTMMC (C)
Check Data Point (DP) in a pass,
Create circle with radius r,
R_{DP} = random (DP)
For each DP
If $W_{DP} >= WR_{DP}$
$R_{DP} = N_{DP}i$
$L_{STDP} = N_{DPi-1}$ // Complete binary tree formation
$R_{STDP} = N_{DPi-2}$
End if
End for
For each L_{ST}
$C_{LSTi} = P_{DPi}+C_{DPSi}$
If $WSTi \approx P_{DPi+1} + C_{DPSi+1}$
$C_{LSTi}[w] = C_{LSTi}[w] + 1$
$C_{LSTi+1} = (P_{Dpi} + C_{DPSi})$ // Micro cluster formation
$D_{Mc} = DP (C_{LSTi} \cap C_{LSTi+1})$ // capturing density between the cluster
End if
End for
For each R_{ST}
$C_{RSTi} = P_{DPi}+C_{DPSi}$
If $WSTi \approx P_{DPi+1} + C_{DPSi+1}$

$C_{RSTi}[w] = C_{RSTi}[w]\ 2^{-\lambda\ (t-C_{RSTi}[t])} + 1$
If C_{DPSi} = = L_{DPSi}
Remove DPS_i
Else
$C_{LST} = (P_{Dpi} + C_{DPSi})$ // Micro cluster formation
$D_{Mci} = DP\ (C_{LSTi} \cap C_{LSTi+1})$ // capturing density between the cluster
End else
End If
End if
End for

The algorithm 1 describe the complete online part of micro-cluster generation through complete binary tree. Initially it describes the method of complete binary tree formation, then the micro cluster formation through complete binary tree is described. The above algorithm depicts the left and right sub tree based clustering. It also calculates the density between clusters by using intersection function. The obtained result from this approach provides efficient clusters and also reduces the time and memory space used for density based data stream clustering technique. The leaf nodes of right sub tree represents the outlier, which is less density data points. The outlier from the complete binary tree is stored in the Weak (w) variable. It is checked with the other micro clusters weight. If the obtained value is less than Weak (w) then this micro-cluster is removed from this process.

3. RE-CLUSTERING MICRO-CLUSTERS

Re-clustering based on micro-clusters are carried out in the offline part. The offline part of proposed technique, re-cluster the micro-clusters using the density among the micro-cluster (density information is stored in the root node database). Based on the density information, the proposed technique re-clusters the micro-cluster with high density. The low density micro-clusters are also clustered with other low density micro-cluster. This approach leaves low quantity of data points as an outlier, because the complete binary tree is a straight forward technique.

Definition 3.1 The micro-clusters are re-clustered by, $R_C = D\ (C_{LSTi+1}$ and $C_{LSTi})$, where the C_{LST} is the micro cluster. D (C_{LSTi+1} and C_{LSTi}) is the density among the C_{LSTi+1} and C_{LSTi}.

3.1 Algorithm 2 Re-clustering Micro-Clusters

D_{Mci} ⟵ density between micro cluster i and i+1
MA_C ⟵ Macro cluster
D (C_{i+1} and C_i) ⟵ density between the cluster i+1 and i (it signifies both right and left sub tree)
For each D_{MC}
$temp_i = D\ (C_{LSTi}$ and $C_{LSTi+1})$
If $temp_i > D_{Mci}$
$MA_C = MA_C + (C_{LSTi}$ and $C_{LSTi+1})$
End if
End for

Here, the C_{LSTi} is the micro cluster. The re-clustering is performed by checking density between C_{LSTi} and C_{LSTi+1}, if they have higher density, both C_{LSTi} and C_{LSTi+1} forms a macro-cluster. The re-clustering algorithm describes the process of re-clustering depends on density information between the obtained micro clusters. The re-clustering process is only depicted for left sub tree based micro cluster, it is not described about right sub tree based micro-clusters. The right sub tree based micro-clusters are also formed in the same manner of left sub tree based micro- cluster. But they can check the high density value of right sub tree they do not compare its high density of left sub tree values.

3.2 Outliers

Outliers for the complete binary tree can be easily predicted be the leaf node data points of right sub tree. Because the leaf node does not have any further child node so removing such kind of outliers does not make huge effect on data stream..

Definition 3.2: The outliers are obtained by: $O_L = R_{LF}$ (CBT). Where the O_L is defined as outlier, R_{LF} depicted the right most leaf, CBT is complete binary tree.

4. Computational Complexity

Space complexity depends on the quantity of data points stored on the complete binary tree. The complete binary tree removes the outlier through the leaf node of right sub tree. So the space complexity purely depends on the leaf node of right sub tree which is O (TR_{STLF}) where the R_{STLF} describes the leaf node of right sub tree and $TR_{STLF,}$ time required to clean the leaf node. The time complexity can be developed as O (d n log (k)) using k-d tree [23]. During the clustering the outlier cleanup process time is also added to which, the worst case O (n $|TR_{STLF}|$). So complete time complexity for clustering is O (d n log (k)) + O (n $|R_{STLF}|$). The time complexity for clean-up process depends on the quantity of neighbor which is O ($\frac{N}{TRSTLF}$) for cleanup. T_{RSTLF} is the time for removing leaf outlier of left sub tree. Time complexity for re-clustering depends on minimum amount of micro-clusters. O (MC_{min}). The below table depicts the space and time complexity of data point based on high and low dimensionality.

Table 1: Time and Space Complexity

	Low Dimensionality	High Dimensionality
Space Complexity	O (TR_{STLF})	O $(TR_{STLF})^2$
Time Complexity 1. clustering 2. clean up 3. re-clustering	 O (n log ($2TR_{STLF}$)) O ($\frac{N}{TRSTLF}$) O (MC_{min})	 O (nT^2_{RSTLF}) O ($\frac{N2}{TRSTLF}$) O (MC^2_{min})

5. EXPERIMENTAL EVALUATION

The performance of this proposed approach is evaluated using various data sets. Initially, the static datasets are used for evaluation which are Chameleon dataset DS3, and Chameleon dataset DS4. These datasets are clustered and re-clustered through this proposed approach which can provide better outcome than the existing techniques. It also provides less time and also space complexity for this clustering and re-clustering process. The below table depicts the overview of datasets used in this approach.

Datasets	Number of samples	Dimensions	Characteristics
Chameleon dataset DS3	7196	2	noises, different shape, size and density
Chameleon dataset DS4	1748	2	noises, different shape, overlaps

The purity of the obtained clusters is also compared with different kinds of density based clustering algorithms. We generate an evolving data stream EDS by randomly choosing the synthetic data sets (DS3, DS4) in 10 times. Figure 4 shows the cluster purity results.

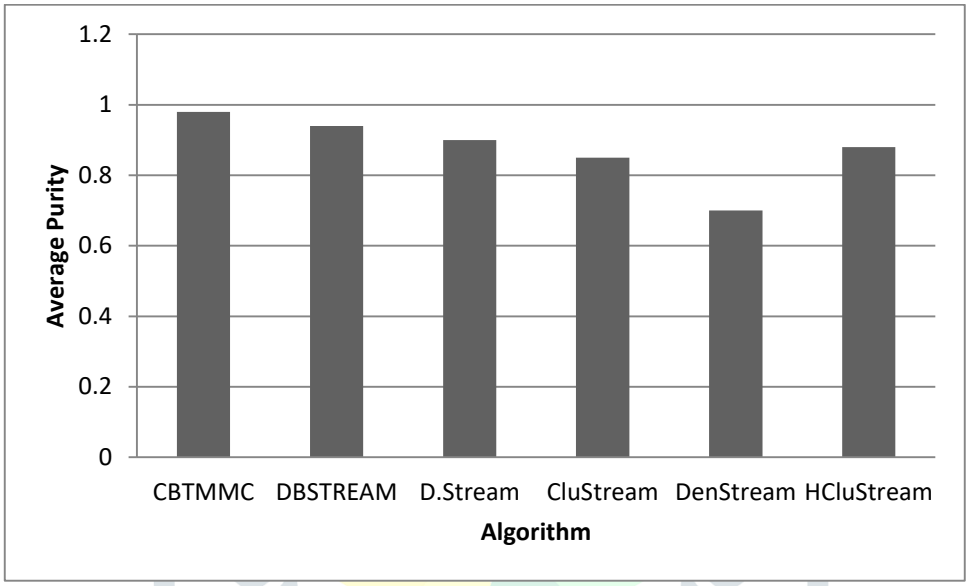

Figure 4 Comparing clustering purity on EDS data set

The above figure depicts the comparison of clustering purity with different kinds of datasets. From the comparison it is clear that the purity level of the proposed CBTMMC algorithm is high (0.98) compared to other kinds of density based clustering algorithms. In the same way the execution time for the EDS dataset is compared with the different density based clustering algorithm:

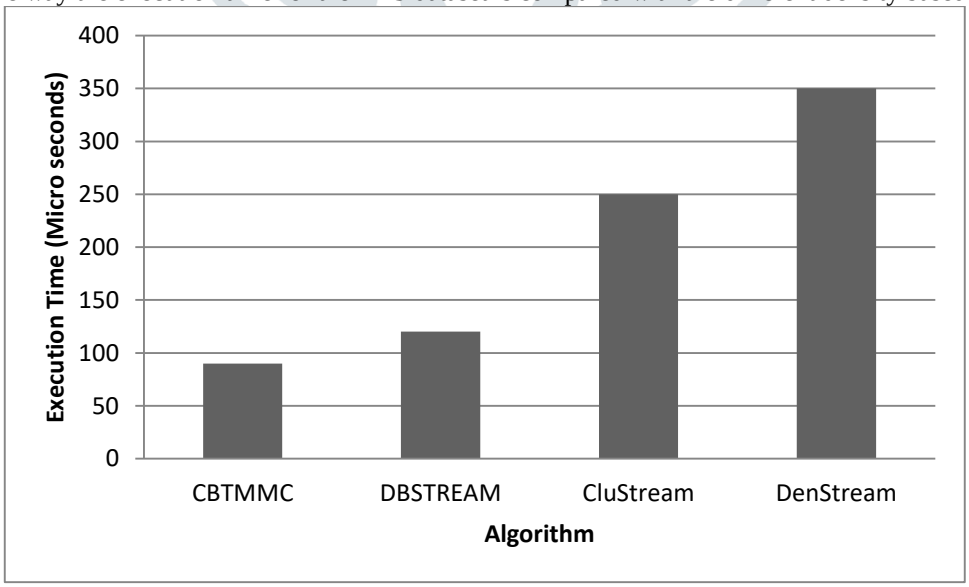

Figure 5 Comparing execution time with different algorithm

From the above result it is clear that the CBTMMC has low time complexity (90 Micro seconds) when compared to the existing algorithms [27]. The space complexity of the proposed technique is compared below, with other kind of density based clustering algorithms.

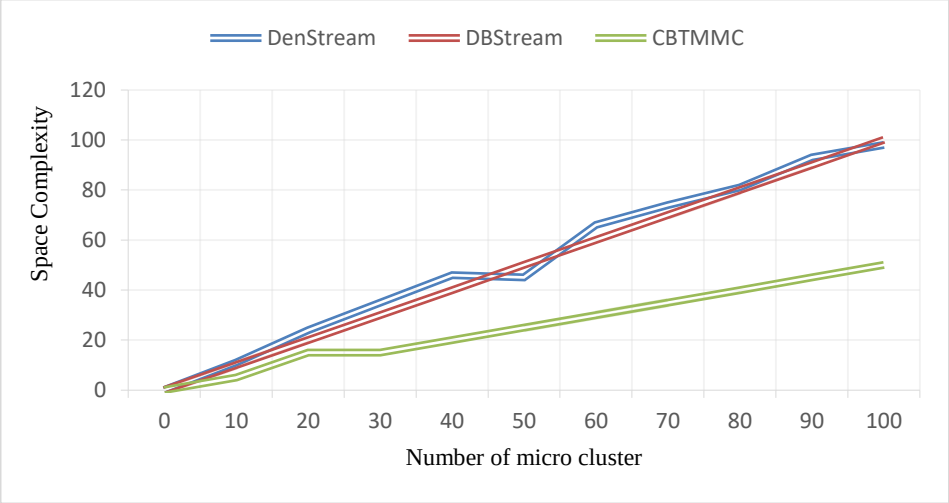

Figure 6 Space Complexity comparison

The space complexity is calculated by number of micro-clusters formed. The space complexity of the DBstream and DenStream [14] [27] is high due to the large space used for storing micro-clusters. But the proposed technique contains less amount of space complexity compared to others.

6. Conclusion

Data stream clustering is the most valuable technique in today's situation, because of the numerous data generated from various applications. Many applications are developed, for this data stream clustering technique, but still the efficient data stream clustering technique is not obtained due to less performance of existing approaches. To overcome this issue, the proposed technique converts the data stream into the complete binary tree, clustering from complete binary tree is simple in less time and space consuming. It is also compared with various datasets which results in better clustering than existing.

REFERENCES

1. Aggarwal, C. C., Han, J., Wang, J., & Yu, P. S. (2003, September). A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases-Volume 29* (pp. 81-92). VLDB Endowment.
2. Barbará, D. (2002). Requirements for clustering data streams. *ACM sIGKDD Explorations Newsletter*, *3*(2), 23-27.
3. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, No. 34, pp. 226-231).
4. Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999, June). OPTICS: ordering points to identify the clustering structure. In *ACM Sigmod record* (Vol. 28, No. 2, pp. 49-60). ACM.
5. Hinneburg, A., & Keim, D. A. (1998, August). An efficient approach to clustering in large multimedia databases with noise. In *KDD* (Vol. 98, pp. 58-65).
6. Cao, F., Estert, M., Qian, W., & Zhou, A. (2006, April). Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining* (pp. 328-339). Society for Industrial and Applied Mathematics.
7. Ruiz, C., Menasalvas, E., & Spiliopoulou, M. (2009). C-DenStream: Using domain knowledge on a data stream. In *Discovery Science* (pp. 287-301). Springer Berlin/Heidelberg.
8. Liu, L. X., Huang, H., Guo, Y. F., & Chen, F. C. (2009, December). rDenStream, A Clustering Algorithm over an Evolving Data Stream. In *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on* (pp. 1-4). IEEE.
9. Chen, Y., & Tu, L. (2007, August). Density-based clustering for real-time stream data. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*(pp. 133-142). ACM.
10. Ester, M., Kriegel, H. P., Sander, J., Wimmer, M., & Xu, X. (1998, August). Incremental clustering for mining in a data warehousing environment. In *VLDB* (Vol. 98, pp. 323-333).
11. Wan, L., Ng, W. K., Dang, X. H., Yu, P. S., & Zhang, K. (2009). Density-based clustering of data streams at multiple resolutions. *ACM Transactions on Knowledge discovery from Data (TKDD)*, *3*(3), 14.
12. Zhang, X., Furtlehner, C., & Sebag, M. (2008). Data streaming with affinity propagation. *Machine learning and knowledge discovery in databases*, 628-643.
13. Ruiz, C., Spiliopoulou, M., & Menasalvas, E. (2007, May). C-dbscan: Density-based clustering with constraints. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing* (pp. 216-223). Springer, Berlin, Heidelberg.
14. Hahsler, M., & Bolaños, M. (2016). Clustering Data Streams Based on Shared Density Between Micro-Clusters. *IEEE Transactions on Knowledge and Data Engineering*, *28*(6), 1449-1461.
15. Hartigan, J. A., & Hartigan, J. A. (1975). *Clustering algorithms*(Vol. 209). New York: Wiley.

16. Sun, H., Huang, J., Han, J., Deng, H., Zhao, P., & Feng, B. (2010, December). gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on* (pp. 481-490). IEEE.
17. Domingos, P., & Richardson, M. (2001, August). Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 57-66). ACM.
18. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, No. 34, pp. 226-231).
19. Bortner, D., & Han, J. (2010, March). Progressive clustering of networks using structure-connected order of traversal. In *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*(pp. 653-656). IEEE.
20. Leicht, E. A., Holme, P., & Newman, M. E. (2006). Vertex similarity in networks. *Physical Review E*, *73*(2), 026120.
21. Nutakki, G. C., & Nasraoui, O. (2017, June). Clustering Data Streams with Adaptive Forgetting. In *Big Data (BigData Congress), 2017 IEEE International Congress on* (pp. 494-497). IEEE.
22. Gaber, M. M., Zaslavsky, A., & Krishnaswamy, S. (2009). Data stream mining. In *Data Mining and Knowledge Discovery Handbook* (pp. 759-787). Springer US.
23. Bifet, A., & Kirkby, R. (2009). Data stream mining a practical approach.
24. Matysiak, M. (2012). Data Stream Mining.
25. Joseph, J. (2011). *Data Stream Mining*.
26. Chen, J. Y., & He, H. H. (2016). A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data. *Information Sciences*, *345*, 271-293.
27. Cao, F., Estert, M., Qian, W., & Zhou, A. (2006, April). Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining* (pp. 328-339). Society for Industrial and Applied Mathematics.