Proceedings of the 4th International Conference on Computing and Data Science (CONF-CDS 2022)
DOI: 10.54254/2755-2721/2/20220605

Improving Performance Parameters of Clusters Using Density-Based Algorithm

Twinkle Keshvani[1], Madhu Shukla[2, *], Meghnesh Jayswal[3], Kishan Makadiya[4]

[1] Department of Computer Engineering, Marwadi University, Rajkot-360003, India
[2] Department of Computer Engineering - AI, Marwadi University, Rajkot-360003, India
[3] Department of Computer Engineering, Marwadi University, Rajkot-360003, India
[4] Department of Computer Engineering, Marwadi University, Rajkot-360003, India

madhu.ce@gmail.com

Abstract. With the advancement in technology, data generated by non-stationary in day-to-day life is massive, continuous and rapid. Many applications such as IoT, transaction systems, network sensors, video surveillance systems, and network intrusion detection systems generate a massive amount of real-time data. The data used in traditional data mining is static in nature, and it can be revised for processing and Analysis. While data in data stream mining is dynamic in nature and it never stops. Besides, the data generated may have a change imbibed in its characteristics over a long/short period of time which is called concept drift. So, analysing such data has huge inbuilt challenges that deal with the dynamism of the characteristics of data itself. This dynamic nature is because of fast and continuous changing data and its enormity. To overcome this limitation, we can use modified clustering techniques that could help us in proper data analysis. Clustering is an effective method used in data mining; but clustering data streams may add some additional challenges such as storage capacity, limited time, one pass and rate of arrival. Furthermore, data streams are fickle in nature and because of this behaviour it needs to be processed as and when it arrives. In addition to that, knowledge about the number of clusters like in K-means clustering is unknown. In view of these characteristics of the data stream, the information or the data generated in the data stream are non-deterministic. Such non-deterministic information contains noise points or outliers, so developing an effective clustering algorithm in a data stream is a crucial task. These methods can work with labelled data in data stream clustering, which has the potential to identify clusters of any shape and noise. The motivation for this research work is using the said algorithm to address and overcome the constraints of the data stream and to dig out the best knowledge from it.

Keywords: datastream, clustering, massive data, concept drift, density-based, arbitrary, micro-clusters, pruning.

1. Introduction

Recent advancement in technologies is the sole source to generate and store huge amounts of data that lead, to the motivation to work with data streams. Almost all the non-stationary systems or IOT enabled systems, do give rise to endless data that is generated at varied speed and has characteristics to change over time [1][2]. Data of such a huge amount is called data stream. Mining a data stream is a

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

process to analyse and identify useful patterns from non-stationary systems. Data gets generated in various applications and could be of any shape such as sensor data in-network sensor applications, a number of transactions in business processes, data generated in video surveillance systems, weather forecast information, stock trading, commodity sales, and so on. There are certain requirements for handling such data generated from real-time applications. For example, data generated from network sensor applications could be of any arbitrary shape due to irregularity of connections similarly for weather forecast applications due to environmental conditions data generated could be of any arbitrary shape [2][3]. Data generation across the globe is a continuous process and it changes over the time as it is also affected by the surrounding/ environment in which they lie. Such continuously changing data is termed as evolutionary data[4]. There are several challenges and issues in the data stream [5]. The real challenge is how to extract valuable information from this evolutionary and enormous data.

Clustering is an Effective tool that can lead in discovering hidden information. Clustering is a process of segregating data objects into one or more similar groups [6]. Function used to measure the distance between two similar objects is Euclidean distance[7]. But this method can find clusters of spherical shape only. So, this method is vulnerable when shapes are arbitrary or undefined. Data generated from data streams can be of any shape and size based on the application. Considering such a case, we need to have a method which address this issue and find clusters of any shape. There are five distinct types of data stream clustering: Partitioning method, Hierarchical method, Density-based method, Grid-based method and Model-based method. Among these Partitioning and Hierarchical clustering algorithms generate clusters which are spherical while Density based methods are known to generate clusters of arbitrary shapes even [5][8]. Adding to this Grid-based is another method that generates clusters of arbitrary shape and has an add-on of fast processing speed. These clustering algorithms do not depend on the number of the data points [7]. Model based clustering is used when there is calculation of standard statistics.

There are several clustering algorithms based on data stream such as [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] discussed in this mentioned paper in detail. Among them density-based clustering algorithm [19] gives insight on clusters which are of arbitrary shape and aiding outliers detection tools. These algorithms are designed to work on data points in space which are scattered. Additionally, these algorithms also help in detecting noise, thus protecting the system from malfunction and detecting it in advance if such patterns are seen to occur. This algorithm is designed not only to identify clusters of arbitrary shape but it also gives protection against noise points [20]. Density based clustering algorithm works in two modes 1. Micro-based clustering 2. Grid-based clustering.

2. Density based clustering algorithm

DENSTREAM, which is a data stream algorithm for clustering, has several features which satisfies the requirement of the evolving data stream. It presents the idea of core micro-clusters, P-micro-clusters and O-micro-clusters. In order to identify core micro-clusters i.e. o-micro-clusters, the algorithm processes in two phase 1. Online phase: For detection of macro-clusters and 2. Offline Phase: For detection of micro-clusters. These methods use a sliding window model specifically fading window type where the data points are faded as time passes and the noise points are pruned. However the limitation of this algorithm is, it does not release any memory space. New memory is assigned to the newly arrived data points and it will not be free until it is pruned. SDSTREAM which is one of the density based clustering method also has ability to identify clusters of arbitrary shape. This method focuses on the most up-to-date information by using the sliding window model and helps in generation of potential micro-clusters and noise. It considers the most recent time frame to generate the potential micro-clusters. The data points which are not allocated in the sliding window frame are discarded over a period of time after it reaches a threshold count.

When it comes to mining data streams, clustering is crucial. The aim of clustering is to collect meaningful information from the streaming data. There are various clustering algorithms introduced based on distance, which can find clusters of only spherical shape. Density-based have a tendency to look for clusters, that aren't spherical in shape. and have that ability to handle noise points that appear

DOI: 10.54254/2755-2721/2/20220605

in data[21]. With the help of a density-based method, it generates the clusters of arbitrary shape and for that; It searches for dense places with sparse spaces between them. [19].Density-based clustering is known for detecting clusters of any shape and size as well they are known to manage broad variety of cluster which is an important characteristic for many real time applications even with traditional data. These algorithms also supports when we don't have to assume no. of clusters. There are certain algorithms which require prior information about the no. of clusters to identify the best input parameters [22][23]. In density based there is no prerequisite about the number of clusters, as it has a huge amount of real-life data.One of the important aspect for these algorithms is to deal with Outliers[24]. Ability to handle Outliers: There are certain factors which affect the results of clustering in an evolving data stream. Considering a network intrusion-detection system fails temporarily and some random noises occur. As data evolves over the time in the data stream, after sometime this real data changes to noise [25]. So, detecting noise in moving data streams is a critical challenge

Based on a research of density-based data stream clustering techniques [5][26], it is found that it, is classified into two broad categories 1. Density micro based clustering which works on micro-clusters [16] [20] and 2. Density Grid based clustering which maps the data points on the grid [14][27][28]. The Density micro based clustering algorithm will be discussed in this study.

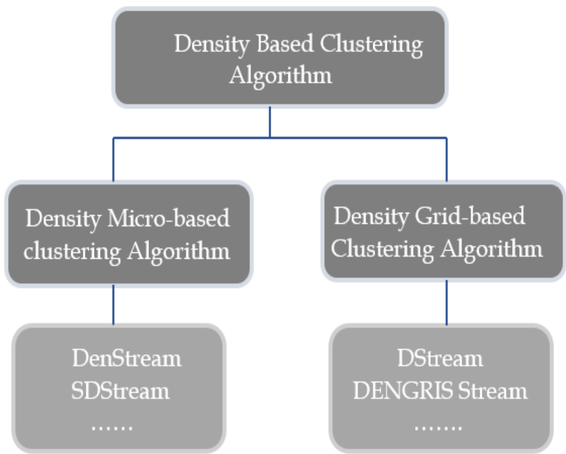

Figure 1. Classification of density based clustering.

2.1. Density micro based clustering's core principle

In this section we have introduced a remarkable method of density-based clustering based on micro-clusters. The idea of micro-cluster was first introduced in [29] for the datasets which were massive and temporal and later it was implemented in Aggarwal et al [29] for data streams. One of the methods to handle the data stream comprises of techniques which works in two phases 1. Online phase: Detection of macro-clusters 2. Offline phase: Detection of micro-clusters [19]. In the first phase it captures the synopsis of information and stores summary statistics of data points from the data stream and in the offline phase it uses this summary statistics to generate the clusters. Basically, micro-clusters and macro-clusters are generated in the online and offline phase respectively. Micro-clusters are a well-known technique in data stream clustering. It makes a compact representation of data and also overcomes one of the challenges of DataStream [30]. Micro-clusters maintain a Temporal locality of data, which is also defined as a time-based extension of the cluster feature vector [29]. Cluster feature vector consist of three fields, so it is also called as triplet vector [20]. It is represented as follows:

The abbreviations and annotations used are mentioned in detail herewith.

CF = (N, LS, SS)

N: Number of data point in the cluster

DOI: 10.54254/2755-2721/2/20220605

LS: Linear sum of N data points [12]

SS: Squared sum of the data points

By Using an additive property of cluster feature it is represented [9] as

CF1 = (N1, LS1, SS1) and CF2 = (N2, LS2, SS2). So, by summing these two vectors it will be:

CF1 + CF2 = (N1 + N2, LS1 + LS2, SS1 + SS2)

This additive property makes it efficient for analyzing data stream clusters. Micro-cluster concept is enhanced by using a timestamp for a collection of d-dimensional objects

It is defined as (CF2x,CF1x,CF2t, CF1t, n) where,

CF2x and CF1x are the same as SS and LS respectively.

CF1t: Sum of timestamps Ti1…….Ti n.

CF2t: Sum of squares of timestamps Ti1……. Ti n.

In [31], statistics information generated can be used in additive ways on different sets of data. This makes it efficient for use in data stream algorithms. Summary information of the dominant micro-clusters in the data stream algorithm [32] is maintained at some moment.

Table 1. Observed parameters.

Sr. no.	Methods	Purity	Memory	Execution Time
1	Density-Based Clustering over an Evolving Data Stream with Noise.[09]	✓	✓	✓
2	A Grid and Density-based Clustering Algorithm for Processing Data Stream.[27]			✓
3	A Modified Approach of optics Algorithm for Data Streams.[19]	✓	✓	✓
4	An efficient hybrid CluStream algorithm for stream mining.[31]	✓		
5	Density-Based Clustering for Real-Time Stream Data			✓
6	Density-based Data Streams Clustering over Sliding Windows	✓		
7	rDenStream, A Clustering Algorithm over an Evolving Data Stream		✓	✓
8	Empirical analysis & improvement of Density-Based Clustering Algorithm in Data Streams.[16]	✓		✓

3. Proposed work

Proposed clustering algorithm works in two modes 1. Online Phase: Which is used for detection of macro-clusters and 2. Offline phase: used for detection of micro-clusters. Online phase is for the maintenance of micro-clusters and offline phase is for generating final clusters.

DOI: 10.54254/2755-2721/2/20220605

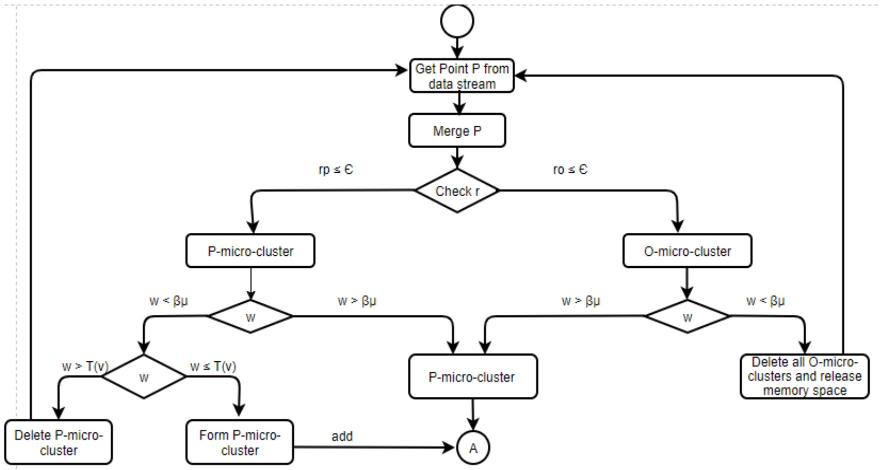

Figure 2. Proposed work.

In order to find arbitrary form clusters in an evolving data stream, it is necessary to maintain clusters in two different groups: Potential micro-clusters and Outliers micro-clusters are detected in an online process while data is continuously arriving. This is stored in a separate memory which is a buffer that stores the outliers. With the arrival of the most recent data points from the data stream [32], the first step is to merge it with the closest P-micro-cluster C_p. If the new radius of C_p, that is an r_p is below or equal to Є than merge it with P-micro-cluster. Else, algorithm tries to merge it with the nearest O-micro-cluster. This can be done by checking radius of O-micro-cluster C_o that is r_o is less than or which equates to Є than merge it with the O-micro-clusterC_o. If there are no new points in a data stream that means the weight of the p-micro-clusters gradually decreases and after some time it gets converted into o-micro-cluster [12]Therefore, it requires checking the weight of p-micro-clusters intermittently. As minimal time span T_p of the P-micro-clusterconverts to O-micro-clusters are

$$T_p = \frac{1}{\lambda} \log \left(\frac{\beta\mu}{\beta\mu - 1}\right)$$

Which is derived from the equation $2^{-\lambda T_p}\beta\mu + 1 = \beta$.As the data stream proceeds, the number of data points arriving increases or decreases as per velocity of the non-stationary system’s data generation. Initially the weight of the micro-cluster is less compared with the existing clusters. So, it does not fall into the group of P-micro-cluster.

Outliers are becoming more common as the quantity of data objects grows, the situation worsens. Also, it is required to keep records of O-micro-clusters, which can turn into potential micro-clusters [20]. Now, by checking the weight of the O-micro-cluster, we get to know the growth of the O-micro-clusters to convert it into the P-micro-clusters. In order to check the weight of O-micro-clusters, first it compares it with the lower limit, which is denoted as ξ. If the weight exceeds the βμ that it means it initiates to convert into a potential micro-cluster by C_o. If the weight does not exceed the lower limit that means it cannot convert into the P-micro-cluster and it requires to be deleted from the outlier buffer space. The Lower bound of the weight is defined as

$$\xi(t_c\ ,\ t_o) = \frac{2^{-\lambda(t_c - t_o + t_p)} + 1}{2^{-\lambda t_p} - 1}$$

where $t_c$is the most recent time and $t_o$is the time at which the O-micro-cluster was created. Based on the value of w, βμ, p-micro-clusters are decided. The weight and validity of p-micro-clusters is decided from w > βμ. When w is less than βμ it goes to check threshold value, whether this microcluster is a real outlier or it has the ability to convert into a potential micro-cluster. At a certain threshold value, it is found that this microcluster will not be able to develop into a potential microcluster. and it is deleted and memory space is released.

DOI: 10.54254/2755-2721/2/20220605

There might be a possibility of error while pruning micro-clusters (O-micro-cluster or P-micro-cluster), but there is guarantee that when checking the current weight of P-micro-clusters C_p, if it is greater βμ, than it must fall into the group of P-micro-cluster or in the Outlier buffer where there are micro-clusters which can grow and convert into potential micro-clusters and if the most recent weight of P-micro-cluster C_p is greater than 2βμ, than it definitely falls in the category of P-micro-cluster.

Assuming w_e exact weight of C_o or C_p. Whenever any micro-clusters are pruned, there are certain conditions. Whenever any O-micro-clusters are deleted, their weight is $w_e \leq \beta\mu$, where βμ is the threshold of outliers. Therefore, the maximum weight for an O-micro-cluster before pruning is βμ. Similarly, whenever any P-micro-cluster C_p, which is generated at the very first when a stream starts to flow and whose weight is at most βμ. After some time, it fades into 2^(-λ(t_c-t_o)) βμ. So, at last w_e is less than or equal to weight maintained by P-micro-cluster plus 2^(-λ(t_c-t_o)) βμ.

The proposed method's pseudocode is as said below:

Step: 1 Input data points in form of chunks
Step: 2 Online phase()
Step: 3 Merging(p)
Step: 4 If $r_p \leq \in$
Step: 5 P- Micro cluster
Step: 6 Else if $r_o \leq \in$
Step: 7 o- Micro-cluster
Step: 8 Check weight of P-micro-cluster
Step: 9 If W<βμ
Step: 10 If check threshold $t_v > w$
Step: 11 Delete P-microcluster
Step: 12 Else
Step: 13 Form P-microcluster and add in potential microcluster
Step: 14 Else
Step: 15 Form Potential microcluster
Step: 16 Check weight of o-microcluster
Step: 17 If W $\geq \beta\mu$
Step: 18 Add into P-microcluster
Step: 19 Else
Step: 20 Delete all o-microcluster with w ≤ βμ and release memory space.

4. Experimental results

While applying clustering on different dataset like Electric Dataset, Sensor dataset etc we did empirical analysis of time consumed, memory space utilized and purity wherein we can conclude that it took all the parameter observed took optimum time when this method was used.

Pruning was applied to these dataset with an intention to improve the time taken; memory space utilized and improves purity resulting into follow scenarios:

It is observed from the experimental table of sensor dataset that the purity of the potential cluster before pruning at threshold value 10 is 93.85 and after applying the concept of pruning the purity gets increased to 94.04 with decrease in memory head too. Implying by the fact that pruning helps to decrease the number of elements which were not suited for clustering. Also, discussing on time taken to cluster the results show more inclination with less time consumed when the process of pruning is employed on data.

Likewise as observed about purity, the weak candidates in terms of cluster are removed due to pruning, henceforth giving good purity in less time.

Similarly looking at the electric dataset it is observed that at threshold value 7 the purity of potential clusters after applying the concept of pruning is highest i.e, 94.04%. Having a look at the experimental results obtained below, it is proved that by removing the outliers from the clusters we get the better cluster quality. We get the best case by checking at different threshold values.

DOI: 10.54254/2755-2721/2/20220605

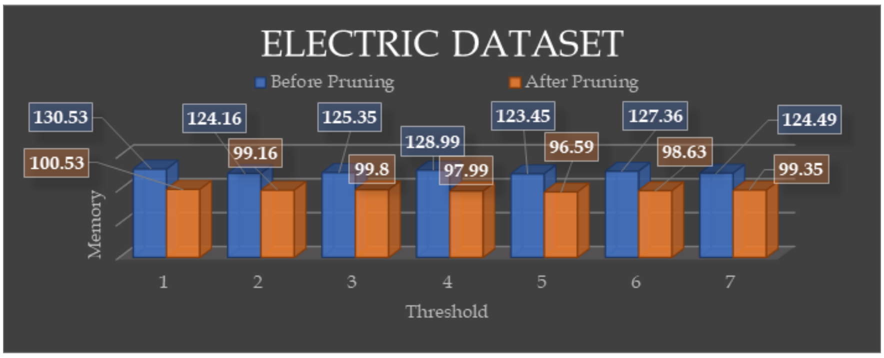

Figure 3. Memory required in electric dataset.

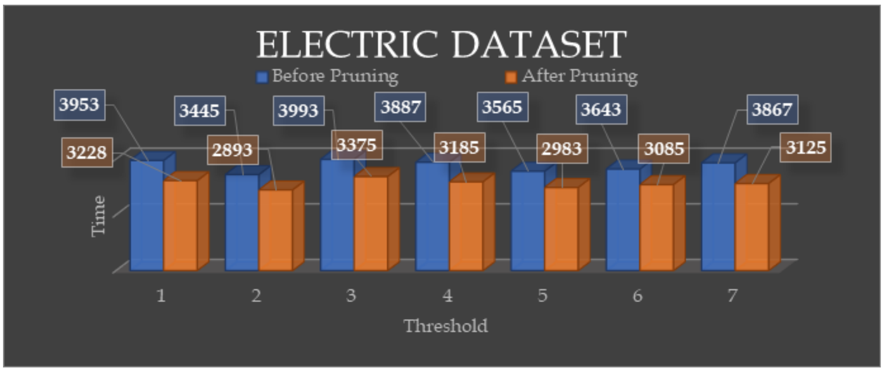

Figure 4. Time required in electric dataset.

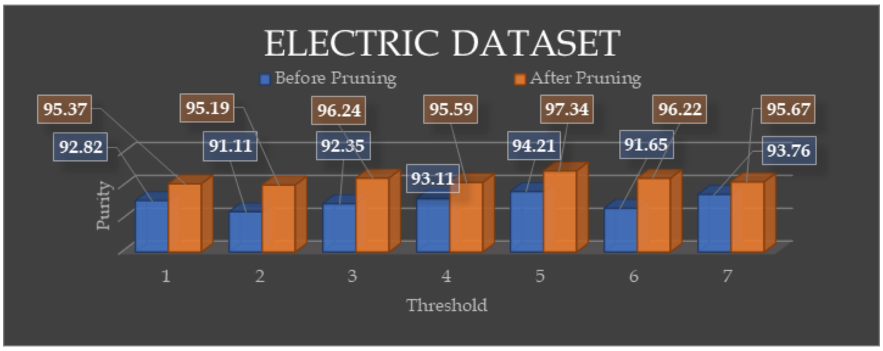

Figure 5. Purity in electric dataset.

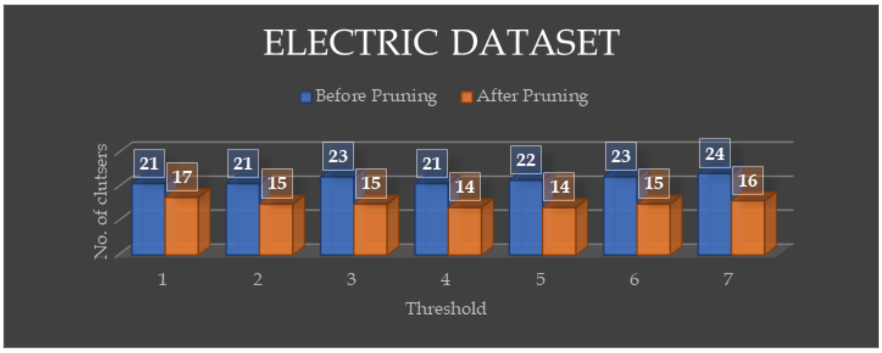

Figure 6. No. of clusters in electric dataset.

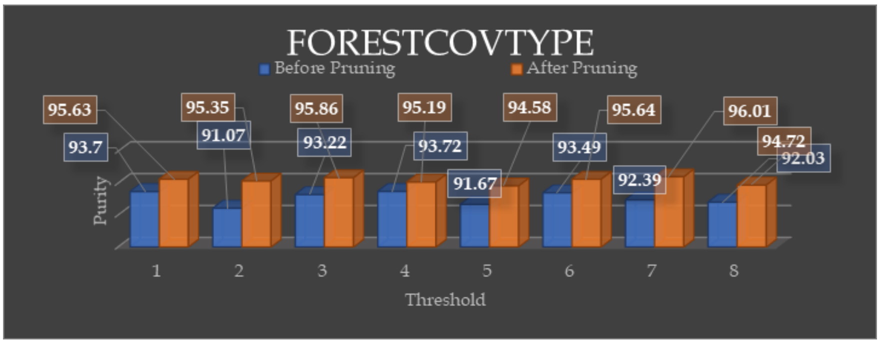

Figure 7. Purity of forestcovtype dataset.

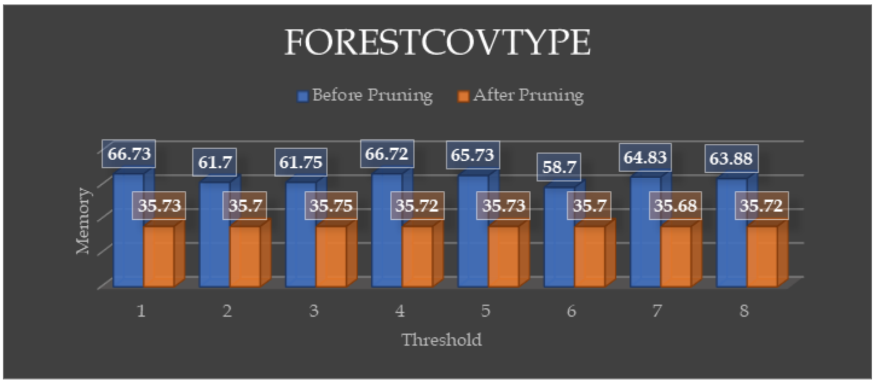

Figure 8. Memory utilization forestcovtype in electric dataset.

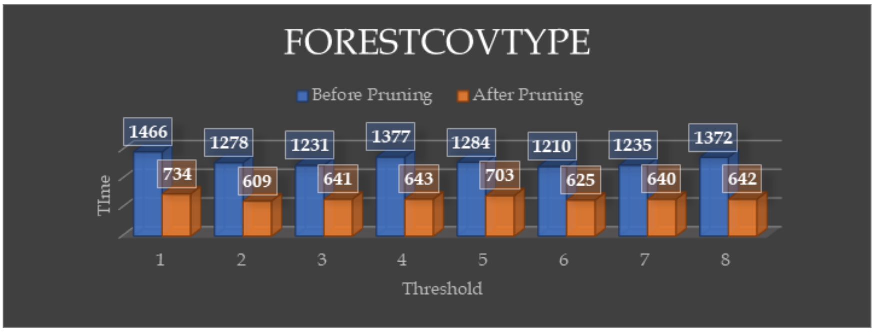

Figure 9. Time required in forestcovtype dataset.

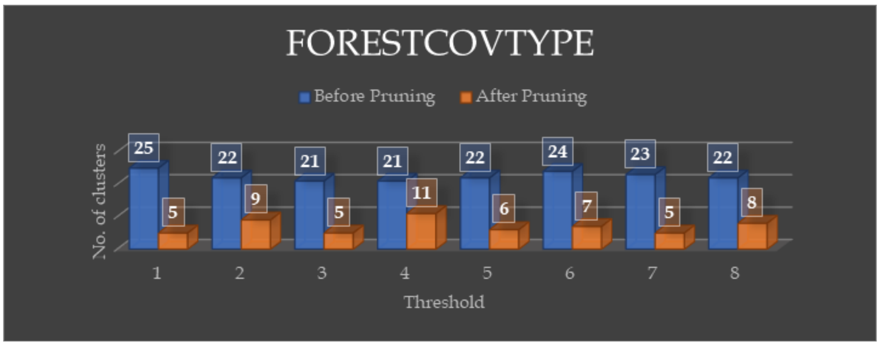

Figure 10. No. of clusters in forestcovtype dataset.

DOI: 10.54254/2755-2721/2/20220605

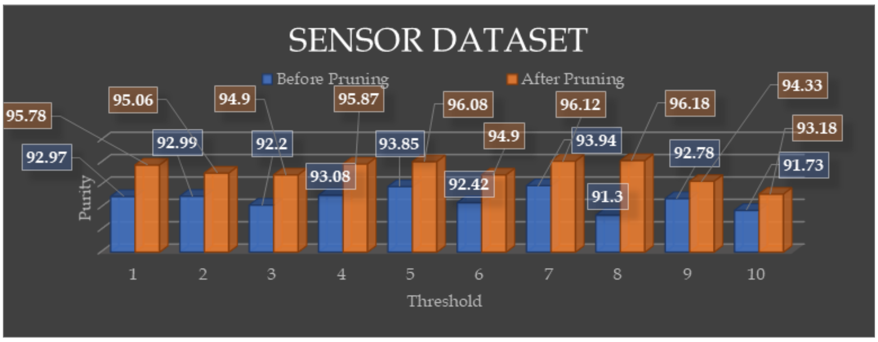

Figure 11. Purity of sensor dataset.

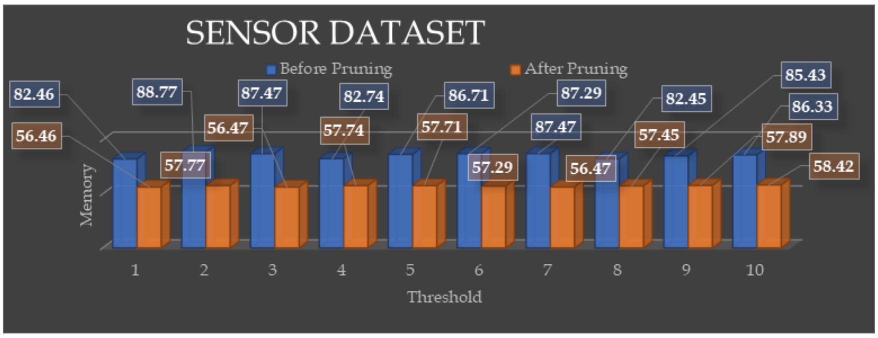

Figure 12. Memory utilization sensor in electric dataset.

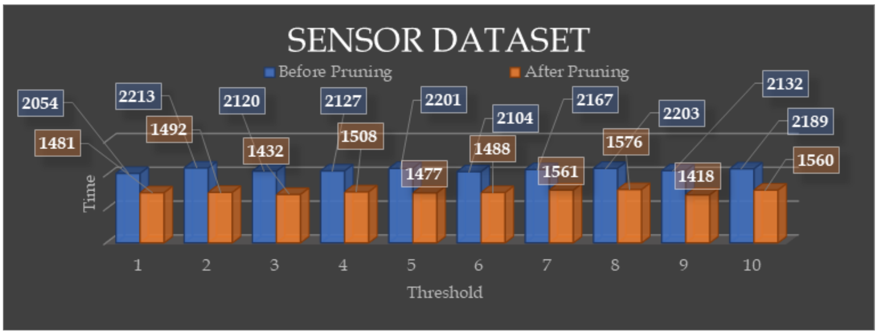

Figure 13. Time required in sensor dataset.

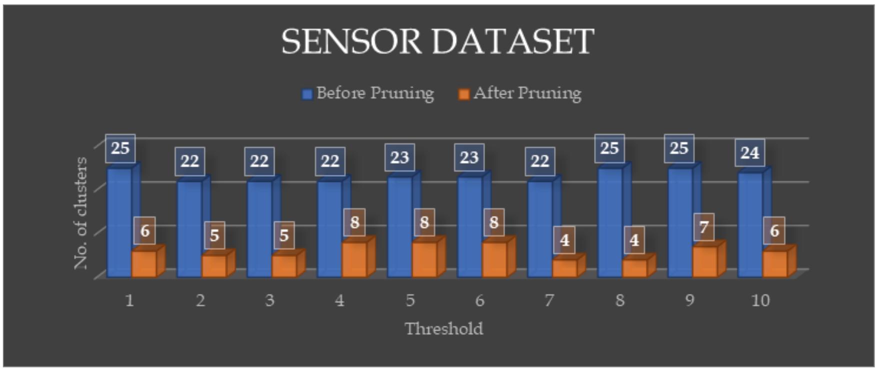

Figure 14. No. of clusters in sensor dataset.

DOI: 10.54254/2755-2721/2/20220605

5. Conclusion

The Density based clustering algorithms has attracted a lot of interest due to its special properties to tackle stream data and the challenges they behold. Therefore, algorithms for grouping a data stream adopt the density-based approach, which supplements the identification of clusters of any shape and size and also aid in outlier detection. The presented research work focuses on the performance parameters of the clustering algorithm before and after pruning with the before and after effects on the results. The performance is evaluated by taking the synthetic as well as real datasets. The results show pruning has an effective role in reducing the CPU timing and also improves the type of clusters found. Empirical evidence of the same is presented in the paper. We got the improvised quality of clusters by applying the proposed algorithm on that.

References

[1] U. Kokate, A. Deshpande, P. Mahalle, and P. Patil, “Data Stream Clustering Techniques, Applications, and Models: Comparative Analysis and Discussion,” Big Data Cogn. Comput., vol. 2, no. 4, p. 32, 2018, doi: 10.3390/bdcc2040032.

[2] Y. Wu, “Network Big Data: A Literature Survey on Stream Data Mining,” J. Softw., vol. 9, no. 9, pp. 2427–2434, 2014, doi: 10.4304/jsw.9.9.2427-2434.

[3] F. Cao, M. Ester, W. Qian, and A. Zhou, “Density-based clustering over an evolving data stream with noise,” Proc. Sixth SIAM Int. Conf. Data Min., vol. 2006, pp. 328–339, 2006, doi: 10.1137/1.9781611972764.29.

[4] I. Khan, J. Z. Huang, and K. Ivanov, “Incremental density-based ensemble clustering over evolving data streams,” Neurocomputing, vol. 191, pp. 34–43, 2016, doi: 10.1016/j.neucom.2016.01.009.

[5] T. Keshvani and M. Shukla, “A Comparative Study on Data Stream Clustering Algorithms,” Lect. Notes Data Eng. Commun. Technol., vol. 31, pp. 219–230, 2020, doi: 10.1007/978-3-030-24643-3_27.

[6] J. Chandrika, “A Novel Approach for Clustering Categorical Data Streams,” Int. J. Innov. Manag. Technol., vol. 4, no. 5, 2013, doi: 10.7763/ijimt.2013.v4.447.

[7] P. Chauhan and M. Shukla, "A review on outlier detection techniques on data stream by using different approaches of K-Means algorithm," 2015 International Conference on Advances in Computer Engineering and Applications, 2015, pp. 580-585, doi: 10.1109/ICACEA.2015.7164758.

[8] M. Jayswal and M. Shukla, "Consolidated study and Analysis of different clustering Techniques for Data streams,” 3rd International Conference on Computing for Sustainable Global Development (INDIACom) pp. 3541–3547, 2016.

[9] F. Cao, M. Ester, W. Qian, and A. Zhou, “Density-based clustering over an evolving data stream with noise,” Proc. Sixth SIAM Int. Conf. Data Min., vol. 2006, pp. 328–339, 2006, doi: 10.1137/1.9781611972764.29.

[10] Kaneriya, A., & Shukla, M. (2015, March). A novel approach for clustering data streams using granularity technique. In 2015 International Conference on Advances in Computer Engineering and Applications (pp. 586-590). IEEE.

[11] J. Gao, J. Li, Z. Zhang, and P. Tan, “Ducstream,An incremental data stream clustering algorithm based on Dense units Detection” pp. 420–425, 2005.

[12] Amini, Amineh, and Teh Ying Wah. "DENGRIS-Stream: A density-grid based clustering algorithm for evolving data streams over sliding window." Proc. International Conference on Data Mining and Computer Engineering. 2012.

[13] A. Zhou, F. Cao, W. Qian, and C. Jin, “Tracking clusters in evolving data streams over sliding windows,” Knowl. Inf. Syst., vol. 15, no. 2, pp. 181–214, 2008, doi: 10.1007/s10115-007-0070-x.

DOI: 10.54254/2755-2721/2/20220605

[14] Shukla, M., Kosta, Y. P.,& Chauhan, P. "Analysis and evaluation of outlier detection algorithms in data streams." In 2015 International Conference on Computer, Communication and Control (IC4) (pp. 1-8). IEEE.
[15] P. Viswanath and R. Pinkesh, "L-DBSCAN: A fast hybrid density based clustering method," Proc. - Int. Conf. Pattern Recognit., vol. 1, pp. 912–915, 2006, doi: 10.1109/ICPR.2006.741.
[16] M. Shukla and Y. P. Kosta, "Empirical analysis and improvement of density based clustering algorithm in data streams," Proc. Int. Conf. Inven. Comput. Technol. ICICT 2016, vol. 2016, pp. 1–4, 2016, doi: 10.1109/INVENTIVE.2016.7830200.
[17] P. Pei, D. Zhang, and F. Guo, "A Density-based Clustering Algorithm Using Adaptive Parameter K-Reverse Nearest Neighbor," 2019 IEEE Int. Conf. Power, Intell. Comput. Syst. ICPICS 2019, pp. 455–458, 2019.
[18] C. Ruiz, E. Menasalvas, and M. Spiliopoulou, "C-DenStream: Using domain knowledge on a data stream," in Proceedings of the 12thInternational Conference on Discovery Science, ser. DS '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 287–301.
[19] Shukla, M., Y. P. Kosta, and M. Jayswal. "A modified approach of optics algorithm for data streams." Engineering, Technology & Applied Science Research 7.2 (2017): 1478-1481.
[20] A. Amini and T. Y. Wah, "IMECS2011-Density Micro-Clustering Algorithms on Data Streams.pdf," vol. I, pp. 14–18, 2011.
[21] D. Barbará, "Requirements for clustering data streams," ACM SIGKDD Explor. Newsl., vol. 3, no. 2, p. 23, 2002, doi: 10.1145/507515.507519.
[22] A. Amini and T. Y. Wah, "Requirements for Clustering Evolving Data Stream," 2nd Int. Conf. Soft Comput. its Appl., pp. 47–50, 2013.
[23] S. Ding, F. Wu, J. Qian, H. Jia, and F. Jin, "Research on data stream clustering algorithms," Artif. Intell. Rev., vol. 43, no. 4, pp. 593–600, 2015, doi: 10.1007/s10462-013-9398-7.
[24] J. Tamboli and M. Shukla, "A survey of outlier detection algorithms for data streams," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 3535-3540.
[25] Lu, J. Feng, J. Zhang, P. Xia, and X. Xiao, "A Parallel Approach on Clustering Traffic Data Stream Based on the Density," Proc. - 2018 6th Int. Conf. Adv. Cloud Big Data, CBD 2018, pp. 281–286, 2018, doi: 10.1109/CBD.2018.00057.
[26] A. Amini, T. Y. Wah, and H. Saboohi, "Streams Clustering Algorithms " J. Comput. Sci. Technol., vol. 29, no. 1, pp. 116–141, 2014, doi: 10.1007/s11390-013-1416-3.
[27] C. Jia, C. Tan, and A. Yong, "A grid and density-based clustering algorithm for processing data stream," Proc. - 2nd Int. Conf. Genet. Evol. Comput. WGEC 2008, pp. 517–521, 2008, doi: 10.1109/WGEC.2008.32.
[28] A. Amini, T. Y. Wah, M. R. Saybani, and S. R. A. S. Yazdi, "A study of density-grid based clustering algorithms on data streams," Proc. - 2011 8th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2011, vol. 3, pp. 1652–1656, 2011, doi: 10.1109/FSKD.2011.6019867.
[29] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A Framework for Clustering Evolving Data Streams - Proceedings of the 29th international conference.pdf," {VLDB} 2003, Proc. 29th Int. Conf. Very Large Data Bases, Sept. 9-12, 2003, Berlin, Ger., pp. 81–92, 2003, [Online]. Available: http://www.vldb.org/conf/2003/papers/S04P02.pdf.
[30] M. Khalilian and N. Mustapha, "Data stream clustering: Challenges and issues," Proc. Int. MultiConference Eng. Comput. Sci. 2010, IMECS 2010, vol. I, pp. 566–569, 2010.
[31] A. Kumar, A. Singh, and R. Singh, "An efficient hybrid-clustream algorithm for stream mining," Proc. - 13th Int. Conf. Signal-Image Technol. Internet-Based Syst. SITIS 2017, vol. 2018-Janua, pp. 430–437, 2018, doi: 10.1109/SITIS.2017.77.
[32] R. Mythily, A. Banu, and S. Raghunathan, "Clustering models for data stream mining," Procedia Comput. Sci., vol. 46, no. Icict 2014, pp. 619–626, 2015, doi: 10.1016/j.procs.2015.02.107.