Data Stream Clustering for Big Data Sets: A comparative Analysis

Ankit Kumar Dubey¹, Rajendra Gupta² and Satanand Mishra³

¹Department of Computer Science & Application, St. Aloysius College (Autonomous), Jabalpur, MP, India
²Rabindranath Tagore University, Bhopal, MP, India.
³CSIR-Advanced Materials and Processes Research Institute (AMPRI), Bhopal, MP, India

E-mail: ankitkdubey@yahoo.com

Abstract. The world is growing rapidly with constantly increasing the data. There are innumerable people around the world who use different types of applications, whether it is for reservation, marketing, shopping or knowledge in the form of text, image, audio and video. Only data is being generated everywhere and this growing data which is large and high dimensional is in nature is generally known as “big data”. For the organizations, it is a big task to cluster streaming big data successfully. In this paper, we are presenting a survey of data stream clustering algorithms applied over big data and big datasets. The paper shows the comparative analysis of all the studied methods and also review the evolution and progression of data stream clustering algorithm for big datasets. The paper also analyses the proposed and implemented algorithms in recent years.

1. Introduction

In any organizations, automation is in almost every domain, and the departments generated number of transactions for each business process. These processes are in sequences of data objects that are continuously streaming. Dealing with this volume and amount of streaming data is a biggest challenge for researchers. The challenge here is to deal with the high dimensional, large volume of big data sets which is changing rapidly with the time. These massive, unbounded stream of data which in and out continuously and the data not available to visit and treatment for the next time, the so-called data streams. Streaming of data $S$ as a continues sequence of data objects $S = \{o_1, o_2, \ldots, o_t, \ldots\}$, where $t$ is the time at data object arrived. Each data object is a continuous sequence of arrival. Social networking is the principal source of knowledge in today's age of big data. Big Data is huge and rapidly growing. In short, since it can be so big and complex, none of the conventional information management tools can store and process large data effectively [1]. For many applications, such as web traffic flow, WSN, network traffic control, the study of vast datasets and extraction patterns is useful.

The data classification into clusters is unsupervised and can be denoted as

$$D = \{K_1 \cup \ldots \cup K_c\}$$

$D$ denotes original data set. $K_a$ are set of clusters in $D$ and $c$ denotes the number of clusters set [2][3].

1.1 Big Dataset Formats

Big Dataset refers to large growing data sets that include formats like: structured, unstructured
and semi-structured data. (Refer Figure 1 and Table 1).

**Figure 1.** Formats of Big Datasets

| PROPERTIES          | STRUCTURED DATA | SEMI-STRUCTURED DATA | STRUCTURED DATA |
|---------------------|-----------------|----------------------|-----------------|
| Data Based On       | Relational database table | XML/RDF           | character and binary data |
| Data Organization   | Well Organized  | Organized up to some limit only | Non Organized |
| Data Concurrency    | Yes             | Not Present          | Not Present     |
| Versioning          | Over Tuples, Row, Tables | Over Tuples or Graph is Possible | As whole Data |
| Flexibility         | Less Flexible   | It is more flexible than structured data but less flexible than unstructured data | More Flexible |
| Scalability         | Less Scalable   | Scalable as compare to Unstructured Data. | More Scalable |
| Query Performance   | High            | Less than Structured Data but more than Unstructured Data | Very Less |

### 1.2 The Sources of Big Data

The maximum size of a big data that can be created consists of social media platform, sensor data, public information and transaction data. Other datasets, such as public data, are included, but corporate data is generated by internal and external companies and must be entered into the system. One important factor is whether the data is unstructured or structured. Non-structural data does not have a predefined data model and therefore requires more resources to understand.

- **Social media datasets** comes from the various different social media platforms Likes, Twitter Tweets & Retweets, Facebook Comments, Video Uploads, and other media that are uploaded and shared via these social media platforms. This kind of data provides the insights into user sentiment and behavior.

- **Sensor datasets** is provides information which is generated by equipment like, mobile phones,
medical devices, traffic monitoring system and weather prediction equipment etc. Sensors that are installed in this kind of machinery provides big datasets, which track user behaviour.

- **Transactional datasets** is generated from the daily transactions that take place through internet (Online). It also includes offline transactions. Transactional data include invoices, payment record, storage records, delivery records etc. and with all these transaction datasets which are almost meaningless, organizations struggle to make sense of the data that they are generating and how it can be put to good use.

- **Public dataset** is the information which can be used and reused without any local and global legal restrictions. It can be freely access, used, modified, distributed and shared by anyone without any permission. Examples of public data includes job openings information, press releases, and marketing materials etc.

It is hoped that this type of data will grow more rapidly over time and spread around the world. The main characteristics of this type of data are high speed, value, size and diversity. (Refer Figure 2).

![Sources of Big Datasets](image)

**Figure 2. Sources of Big Datasets**

1.3 Clustering Methods
There are mainly five types of clustering methods are available.

- **Hierarchical Clustering**: Similar objects in this group of algorithms are divided into groups called clusters. An endpoint is a set of clusters, each cluster is different, and the objects in each cluster are similar. There are two levels of clustering: dominant (up and down) and agglomerative (bottom up). In the split or top-down approach, we split all observations into one “cluster” and then divide the cluster into two clusters with the least similarity.

- **Partitioning Clustering**: This clustering method creates separate clusters or groups so that the members within the cluster are as similar as possible and the members of the other clusters are as different as possible. Algorithms require an expert to specify the number of clusters to create. The most commonly used methods are k-mean, k-medoids and k-Mode Algorithm.

- **Density-Based**: This unsupervised learning clustering method is capable of identifying uncommon clusters in the data, based on the characteristics that a data cluster is a continuous area with a high density of points, the closer the data, the greater the density. They are separated form cluster of low density.
Grid-Based: In this method, the object space is measured into defined number of cells that form a grid formation. The major power of this method is fast processing time. It is dependent only on the number of cells in each dimension in the assigned space.

Model-Based: Model-based clustering claims that the model is generating the data and tries to retrieve from the data the original model. The model that is then extracted from the information defines how the text is transmitted to the clusters and clusters. Probability is the most widely used criterion for testing model parameters.

Clustering Methods Advantages and Limitations are provided in Table 2.

| SN | Clustering Method | Advantages | Limitations |
|----|-------------------|------------|-------------|
| 1  | Hierarchical      | Efficient when dealing with similarity or distance. | Indefinite in case of termination criteria. |
| 2  | Partitioning      | Easy to implement and execute. In this creation of clusters is in iterative manner. | User has to predetermine the number of clusters. Works only on spherical shaped clusters. |
| 3  | Grid-Based        | Easily handle noises. Fast Execution. | A predefined grid size is compulsory. Not able to work on high dimensional data. |
| 4  | Density-Based     | Easily detect arbitrary shaped clusters. Easily handle noises. | Prior parameters settings are needed. Not able to work well in multi density data. |
| 5  | Model-Based       | Automatically defined Number of clusters. Able to deal with noises effectively. | Depends on hypothesized structure or model. |

2. Motivation
After reviewing the research papers (references given), what we understand is A big dataset stream is a massive sequence of transaction and we have few constraints,
1. Big dataset streams are everlasting behavior, so it is not possible to keep these entire data streams in memory.
2. Once Big dataset stream has been processed, it will not available for processed again.
3. No system can detect, manage and control in which order big datasets data streams will arrive.
4. Due to continue in nature, big datasets streams should be cluster in specific time limit.
5. Noisy and outlier transaction are also there in big dataset streams, and should also be detected.

We use Big data stream analysis to gain useful knowledge from current events in an efficient and fast way and to detect new trends to improve their performance [5].

3. Related Work
Most of the previous and latest research that associate to big data stream clustering surveys give an overview on Big Data clustering methods and how they can help to improve performance, scalability and results accuracy. A number of survey papers began to summarize and structure the field.

The work of [6] presented studies from traditional data analysis to the existing big data analysis. This
paper offers a brief introduction to algorithms for data and big data mining that consist of mining technologies for clustering, classification, and recurrent trends. Similarly, [7] offers numerous Big Data instruments, approaches and technologies in this field of study. The authors analyse the characteristics of three programming models and their development. Three types of analytics are compared in the data processing category, with definitions of analytics, predictive analytics, and instructional analytics. The authors found two kinds of experiments in the information system experiment type: part tests and system tests. Finally, the classification of large data processing software is discussed. Authors in [8] presented the detailed and comprehensive survey of algorithms for data streams clustering. They categorized clustering approaches according to their nature and given overview of opensource streaming platform. Authors in [9] focuses on anomaly detection and conducted a survey on big data technologies and machine learning algorithms. Authors had worked with use cases and investigated anomaly detection on real time data. They showed the limitation of existing approaches in identifying anomalous behaviour of data. In this work they also discuss about the Application of big data, Machine Learning algorithms and big data processing technology for identifying anomalies. The work of authors [10] presented evaluation of the performance of recent clustering algorithms on the different data sets having different parameters. The work of [11] discussed about data streams hidden pattern and trends. The algorithm and methods has been categorized and the main observation was combination of clustering method works and show better performance for evolving data streams. The Author’s [3] present a comprehensive survey of tools and technologies, methods and techniques and major issues in big data stream analysis. In this work [12], Authors focus in finding tendencies in big data and the streaming data. They proposed an algorithm which can help finding frequent item sets.

4. Research Method

In this paper we studied systematic literature review of clustering methods and algorithm used in analyzing and clustering of big dataset streams. The main objective of this research review is to study and analysis what research has been carried out for the clustering of big dataset streams and what all efforts done by the researcher’s and practitioner [13]. This study tries to review the following challenges of data stream clustering for big datasets:

- What methods and algorithms are developed for memory space optimization?
- What methods and algorithms are developed for increasing processing capability (time complexity)?
- What methods and algorithms are developed as Mechanism for identify outliers/noise?
- What methods and algorithms are developed for handling high dimensional data?

Table 3. Depicts a Comparison of Different Stream Clustering Methods Comparing their Capabilities and Characteristics in Context with Big Dataset Stream Analysis

| Year | Algorithm | Clustering Method | memory space optimization | processing capability (time Complexity) | Mechanism for identify outliers/noise | Handling high dimensional data |
|------|-----------|-------------------|---------------------------|------------------------------------------|---------------------------------------|-------------------------------|
| 2015 | TS-Stream | Hierarchical      | Limited                   | Limited                                  | Limited                               | -                             |
| 2015 | pcStream  | Model Based       | Limited                   | Limited                                  | Limited                               | -                             |
| 2015 | StreamXM  | Partitioning      | Limited                   | Limited                                  | Limited                               | -                             |
| 2015 | SNCSStream| Model Based       | Limited                   | Limited                                  | -                                     | -                             |
| 2016 | SNCSStream+| Model Based       | Limited                   | Limited                                  | -                                     | -                             |
| 2016 | DBSTREAM  | Density Based     | Limited                   | Limited                                  | Limited                               | Limited                       |
| 2017 | EDDS      | Density Based     | Limited                   | Limited                                  | Limited                               | -                             |
| Year | Algorithm          | Type                  | Limited   | Slow     | Sensitive | Accuracy | Yes  |
|------|--------------------|-----------------------|-----------|----------|-----------|----------|------|
| 2017 | A-BIRCH            | hierarchical          | Limited   | Limited  | -         | -        | -    |
| 2018 | evoStream          | Evolutionary algorithm| Optimised | High     | -         | Yes      |      |
| 2018 | Fuzzy-CSar-AFP     | Fuzzy Clustering      | Restricted| High     | -         | Yes      |      |
| 2019 | SNDC               | Density Based         | Limited   | Slow     | Sensitive | High     | Yes  |
| 2019 | DGStream           | density               | Yes       | low time complexity | High | Accuracy | Yes  |

From the analysis of the stream clustering algorithms (Table (3)), we have seen that over the past five years, several studies have been conducted to analyze big data stream. Although we have focused mainly on the 12 algorithm which had been developed between 2015 to 2019. This paper compares these algorithms based on the big-data challenges that were discussed in research method section as questions.

The TS-Stream algorithm is a clustering system based on a hierarchy [14]. TS-Stream uses sluggish, fixed-length memory that is washed and replenished at each iteration, so it runs slowly and, as with the batch algorithm, does not expect all streams to be instantly accessible. The spatial complexity is $O(m)$ in the worst case (divided into lists), since the distance needed for the tree is restricted by a set of flow of times.

pcStream [15], is a model based algorithm for dynamically detecting and managing sequential temporal contexts. This algorithm is capable to work with high velocity data-streams.

StreamXM [16], is following portioning method for clustering data stream. A streamlined clustering technique that does not require random selection, iteration, or extensive knowledge of the data to create a cluster with data-related information. This allows the clustering method to convert several classes into classes with atomic distribution.

SNCStrean [17], an online clustering algorithm, is capable of identifying clusters of non-hyperspherical results. In order to create social media and find clusters using variable mode, this approach only uses one stage of computation. To track the evolution of clusters during data flow, it utilises a scaleless hemophilic technique.

SNCStream+ [18] outperforms a number of clustering consistency data stream clustering algorithms and works on both synthetic and real data issues under restricted computing time and memory space.

DBSTREAM [19], for the application of data processing modules, offers a new language for flow processing, as well as the ability to combine, filter and archive for further data analysis. Priority restrictions are immediately overcome by the DBStream scheduler, so it can run individual questions in parallel to speed up processing time.

In the input dataset, the EDDS algorithm identifies clusters and discrimination, merges new clusters with existing clusters, and filters new outputs for the next step [20]. This altered the standard DBSCAN algorithm and summed up the main surface points for each cluster. As a convergence technique, the algorithm uses the DBSCAN principle of reaching densities and reduces internal key points with a meta heuristic change. Based on the decay function, the algorithm eliminates key points and ageing boundaries.

A-BIRCH [21], a tool used by Gap Statistic to automatically measure the threshold of a BIRCH cluster algorithm. This approach removes the need for the global clustering phase of BIRCH and does not require the number of projected clusters to be estimated. This is achieved to break down properties such as the cluster radius and the minimum distance by evaluating short, representative subsections of
the results. To measure the high likelihood threshold for joining the right cluster of objects, these properties are then used. During the online process downtime, EvoStream [22] performs evolutionary optimization and eventually generates and improves the final clusters. EvoStream will substitute individual parts entirely, boosting computing speeds by up to 75%. Additional measures for re-clustering will further boost the efficiency of the cluster. Calculating the flow is very quick and a good number of new generations is beneficial. We may observe that there is no influence on the consistency of clusters through the use of evolutionary algorithms in upgrading microclusters. This enables the algorithm to run without flow, thus significantly reducing the additional expense of the individual part measurement. Fuzzy-CSar-AFP [23], an online genetic fuzzy scheme designed to provide data sources with fascinating fuzzy association rules. It has the ability to control multiple sections of a variable, enabling the algorithm to conform to each rule variable's precision requirements. Without getting to know the domain of the properties, it is easy to deal with data streams, since it requires a process to change them in real time. In the first real-world psychophysiology problem, which analyses the relationships between different EEG signals of subjects receiving different stimuli, the efficacy of Fuzzy-CSar-AFP was verified. SNDC [24], For SNDC, it is possible to equate category characteristics with numeric characteristics; then a process of nonlinear dimension reduction dependent on adjacent comparisons is applied to and the database size. To evaluate the similarity of data points, a group method calculates a neighbor's distance. The SNDC is able to change the focal points according to the effects of the cluster dynamically and can distinguish logical variations. SNDC can boost the efficiency of algorithms and have more reliable outcomes. DGStream [25], It functions faster than multiple algorithms. Without prior knowledge of parameters like aggregation of some form, various densities, and number of clusters, it is possible to locate data sets. It is an effective and efficient noise and outlier reduction algorithm; only one transmission is needed to process the transmitted data; see data modification using a decomposition function that reduces the weight of the data in period of time.

5. Discussion
In this paper we have selected 12 common data stream clustering algorithms, and then the proposed algorithms are variations or extensions of these algorithms. Study results are now presented in conjunction with research questions aimed at conducting systematic reviews of research.

- What methods and algorithms are developed for memory space optimization?
  We find that almost all algorithms are using limited memory space and it depends on their parameters. We found evoStream, Fuzzy-CSar-AFP and DGStream is comparatively better from rest of the surveyed clustering algorithms.
  In the idle periods of the online process, evoStream performs evolutionary optimization to construct and optimise the final clusters incrementally. Since the online phase will otherwise be inactive, our solution does not decrease the loading speed while essentially reducing the offline phase computing overhead.

- What methods and algorithms are developed for increasing processing capability (time complexity)?
  evoStream completely replaces individual components, increasing computing speeds by up to 75%. You can further improve the quality of the cluster by taking additional steps to re-cluster. It is useful to calculate the flow is very fast and a sufficient number of new generations. Fuzzy-CSar-AFP requires the development of a wide range of rules to increase processing time. In addition, we can observe that both Fuzzy-CSar and Fuzzy-CSar-AFP record less, depending on the growing and usage characteristics of these sub-algorithms. Initially, the association rules supported by these algorithms throughout the entire process are empty, then grow as the algorithm's runtime, and eventually begin to
stabilize. DGStream saves execution time because it only has to deal with these representative points, not all of the data it represents.

- What methods and algorithms are developed as Mechanism for identify outliers/noise?
evoStream allows for more efficient use of available computing resources for competitive results, especially for spherical clusters. Because it is based on an evolutionary algorithm, it cannot detect noise. It can be observed that both the minimum and maximum values are updated, regardless of whether the Fuzzy-CSar-AFP has a greater or lesser value than most data. As you can see from the pictures above, the compatibility isn't perfect, but it's pretty good, and in any case it's much better than the option to process the data and use the absolute attribute domain. DGStream density-based algorithm for aggregating large, noisy spatial datasets. It also uses noise canceling and external communication mechanisms. DGStream has its own strategy for working with external groups. He discovers them and marks these points as excessive. The algorithm then continues to read the stream data points and update them to a new microcluster if some of the data weights close to each other form such a dense network that it exceeds the set threshold. A more efficient approach is a very important step in completing the clustering process on a dataflow cluster to save system time and space.

- What methods and algorithms are developed for handling high dimensional data?
The evoStream method can be useful when the stream is very fast and there is not enough time for many generations. Overall, the proposed upward trend appears to have already yielded good results. If we need to receive real-time updates on the status of the rule population every time we receive new data, this will be possible using Fuzzy-CSar or Fuzzy-CSar-AFP. DGStream divides the multidimensional space of the input data into a density grid; We used this method because it is impractical to store all the raw data. These small networks are high resolution and are associated with numbered data logs.

Finally, in our study, DGStream outperformed many common stream clustering algorithms based on density. Without prior knowledge of parameters such as aggregation of any form, multiple density, number of clusters, it is possible to find datasets. The algorithms are the fastest comparable algorithms; the optimum output of full time efficiency is obtained. It is also ideal for real-time applications, where old information declines over time, for example in stock marketing, and the most recent information is of greatest importance.

6. Conclusion
In this paper, we have investigated the latest developments since last five years (2015-2019) in the area of big data stream Clustering. This review provides a comprehensive and systematic overview of the newly developed Big Data Stream Clustering algorithm to facilitate research in this area. The algorithm under consideration provides the necessary training to suggest future research directions and to solve existing problems when using big data stream clusters. It is expected that many new big data stream clustering methods will be developed, especially for data streams, such as sensor network, time-series streams, and video and audio data streams. When scanning a data stream a single piece of data, a large stream of data cannot be loaded into memory. This is a big issue and questions for future research.

7. References
[1] N. AlNuaimi, M. M. Masud, M. A. Serhani and N. Zaki 2019 Applied Computing and Informatics.
[2] M. Khalilian, N. Mustapha and N. Sulaiman 2016 Journal of Big Data
[3] A. Jain, M. Murty and P. Flynn 1999 ACM Computing Surveys 31.
[4] G. Sunitha and C. Jaswitha, 2019, International Journal of Innovative Technology and Exploring Engineering (IJITEE) 8.
[5] T. Kolajo, O. Daramola and A. Adebiyi 2019 *Journal of Big Data*.
[6] C.-W. Tsai, C.-F. Lai, H.-C. Chao and A. V. Vasilakos 2015 *Journal of Big Data* volume.
[7] D. Chong and H. Shi 2015 *Journal of Management Analytics* 2.
[8] M. Gheshmoune, M. Lebbah and H. Azzag 2016 *Big Data Analytics*.
[9] A. H. R. Ahamed, F. Nasaruddin, A. Gani, H. I. AbakerTargio, E. Ahmed and M. Imrand 2019 *International Journal of Information Management*, 45 pp. 289-307.
[10] S. Mansalis, E. Ntoutsi, N. Pelekis and Y. Theodoridis 2018 *Statistical Analysis and Data Mining: The ASA Data Science Journal* pp. 167-187.
[11] U. Kokate, A. Deshpande, P. Mahalle and P. Patil 2018 *Big Data Cognitive Computer*.
[12] C. Fernandez-Basso, A. J.Francisco-Agra, M. J.Martín-Bautista and M. Ruiz, 2019 *Knowledge-Based Systems* 163 pp. 666-674.
[13] M. Höst and A. Oručević-Alagić 2011 *Information and Software Technology*. 53, pp. 616-624.
[14] C. M. M. Pereira and R. F. d. Mello 2014 *Journal of Intelligent Information Systems* pp. 531-566.
[15] Y. Mirsky, B. Shapira, L. Rokach and Y. Elovici 2015 *Advances in Knowledge Discovery and Data Mining*.
[16] R. Anderson and Y. S. Koh 2015 *Big Data Analytics and Knowledge Discovery* pp. 270-282.
[17] J. P. Barddal, H. M. Gomes and F. Enembreck 2015 (Association for Computing Machinery New York NY United States, Salamanca Spain).
[18] J. P. Barddal, H. M. Gomes, F. Enembreck and J.-P. Barthès 2016 *Information Systems*, 62 pp. 60-73.
[19] M. H. a. M. Bolaños 2016 *IEEE Transactions on Knowledge and Data Engineering*, 28 pp. 1449-61.
[20] A. A. A. Alazeez, S. Jassim and H. Du 2017 (*IEEE*, Bristol).
[21] B. Lorbeer, A. Kosareva, B. Deva, D. Softić, P. Ruppel and A. Küpper 2016 *Advances in Big Data* pp. 169-178.
[22] M. Carnein and H. Trautmann, 2018 *Big Data Research*, 14 pp. 101-111.
[23] E. Ruizab and J. Casillas 2018 *International Journal of Approximate Reasoning* 93 pp. 463-486.
[24] S. Xu, L. Feng, S. Liu and H. Qiao 2020 *Engineering Applications of Artificial Intelligence* 89.
[25] R. Ahmed, G. Dalkılıç and Y. Ertenc 2020 *Expert Systems with Applications* 141.