Iraqi Journal of Statistical Science (33) 2021
pp**[72-88]**

ODED: Outlier Detection in Educational Data

Dr. Ammar Thaher Yaseen Al Abd Alazeez*
ammarthaher@uomosul.edu.iq

Abstract

Clustering data streams is one of the prominent tasks of discoverin g hidden patterns in data streams. It refers to the process of clustering newly arrived data into continuously and dynamically changing segmentation patterns. The current data stream clustering algorithms are lacking general clear steps for analysing new incoming data chunks. However, the majority of existing data stream solutions are adapting the clustering methods of static data to work with data stream setting. The main issue of concern is to propose a solution can improve the performance of existing approaches and present correct clusters and outliers. Data arriving in streams often contain outliers, which may have equal importance as clusters. Thus, it is desirable for data stream clustering algorithms to be able to detect the outliers as well as the clusters. The data stream clustering algorithms should be able to minimise the effects of noise and outliers data in a given dataset. This article presents a stream mining algorithm to cluster the data stream and monitor its evolution. Even though outlier detection is expected to be present in data streams, explicit outlier detection is rarely done in stream clustering algorithms. The proposed method is capable of explicit outlier detection and cluster evolution analysis. Relationship between outlier detection and the occurrence of physical events has been studied by applying the algorithm on the education data stream. Experiments led to the conclusion that the outlier detection accompanied by a change in the number of clusters indicates a significant education event. This kind of online monitoring and its results can be utilized in education systems in various ways. Viber education data streams produced by Viber groups are used to conduct this study.

Keywords: Big Data; Data Stream Clustering Algorithms; Clustering Educational Data

الملخص

عنقدة البيانات المستمرة هي واحدة من المهام المميزة لاكتشاف الانماط المخفية في البيانات المستمرة. فهي تشير الى عملية اكتشاف مجاميع جديدة في البيانات مستمرة الوصول والتي تغير من انماطها باستمرار. خوارزميات عنقدة البيانات الحالية تفتقد الخطوات العامة الواضحة لتحليل مجاميع البيانات الجديدة. ولكن غالبية الحلول الموجودة الآن هي تحديث الخوارزميات التقليدية مع البيانات المستمرة. المسالة المهمة قيد الدراسة هي اقتراح حل يمكنه من تحسين كفاءة الطرق الموجودة وتقديم عناقيد صحيحة فضلا عن الحالات الشاذة. البيانات الواردة في سيل البيانات غالبا ما تحتوي على حالات شاذة، والتي من الممكن ان تحمل الاهمية نفسها التي تملكها العناقيد. وهكذا، انه من المرغوب لخوارزميات عنقدة البيانات المستمرة انها تكون قادرة على اكتشاف الحالات الشاذة فضلا عن عناقيد البيانات.

Received date: 9/12/2020 **Accepted date:** 2/2/2021 **Published data:** 1/6/2021

Computer Science and Mathmatics College-University of Mosul

خوارزميات عنقدة البيانات يجب ان تكون قادرة على تقليل تاثير البيانات العشوائية والحالات الشاذة في البيانات المعطاة. هذه المقالة تقدم خوارزمية تنقيب جديدة لعنقدة البيانات المستمرة ومراقبة تطورها. على الرغم من ان اكتشاف الحالات الشاذة هو متوقع في البيانات المستمرة، التعريف الصريح لاكتشاف هذه الحالات غير واضح. الخوارزمية المقترحة قادرة على اكتشاف البيانات الشاذة وتحليل تطور العناقيد. العلاقة بين تحديد البيانات الشاذة وتوالي الاحداث الحقيقية تم دراستها بواسطة تطبيق الخوارزمية المقترحة في هذا البحث على البيانات التعليمية المستمرة. التجارب التي تم اجرائها على هذه البيانات قادت الى استنتاج ان اكتشاف الحالات الشاذة مرتبط مع التغييرات في عدد العناقيد والتي تشير الى احداث مهمة في عملية التعليم. هذا النوع من المراقبة الآنية ونتائجه ممكن ان يستخدم في انظمة التعليم في مختلف الطرق. البيانات التعليمية المستمرة لجروبات برنامج الفايبر تم استخدامها في هذه الدراسة.

Introduction

Data stream mining has as of late tremendous measure of consideration [1]. An data stream is an arrangement of persistently arrive data which forces a solitary pass limitation where random access to the data isn't attainable [2]. The speed of data appearance, just as the speed of data handling, may shift from application to application. For certain applications, the appearance and handling of data can be acted in a disconnected cluster examination design, others require constant and on-going investigations; here and there it requires quick activity upon the preparing of approaching data streams, for example, dynamic administration of data centres [3]. Data stream mining can be characterized as the way toward finding hidden patterns inside a huge volume of unbounded data streams. In such cases, mining methods must notice and acknowledge existence requirements and must have the option to find right hidden clusters inside the imperative limits. Data stream clustering methods intend to find grouping designs (clusters) fundamental the stream data as per similitudes between their highlights [4].

Data arrive in streams regularly contains outliers, which may have equivalent significance as clusters. Accordingly, it is desirable for data stream clustering methods to have the option to distinguish the anomalies just as the groups [5]. Simultaneously, the entirety of the data stream properties in addition to the realities that data stream is non-deterministic and consistently contains outliers and noises represent a critical and genuine specialized test to data stream clustering [6].

A significant test in data streams analytics is outlier detection, where the example encoded in the stream changes over time. Anomaly identification exists, in actuality, issues, for example, occasional climate changes, stock market downturns, rallies, PC network traffic, remote sensor data, telephone discussions, web-based media, marketing data, ATM exchanges, web pages, power utilization traces, online sentiment analysis, intrusion discovery, and fraud detection, etc. Then again, certain groups may decrease over the long time because of a portion of its

individuals are matured; such a group may ultimately turn into an outlier. An data mining model ought to consistently distinguish anomalies and along these lines needs to adjust rapidly [7].

The three successive phases of activities inside the clustering data stream outline work are as per the following:

1. *Build Clusters*: This stage takes the association set of the approaching data chunk and the maintained existing outliers as the info dataset and apply a traditional clustering method to cluster the data in the data dataset into groups. The output groups of this progression are then put away in memory for different activities. Ideally, this cycle should be finished in one pass of scanning the data chunk.
2. *Merge*: This phase of activity inspects the connections between the recently framed groups from the data chunk and existing groups from the past round, and afterward utilizes a group of reasonable consolidating procedures to choose which new groups are to be joined with certain current groups.
3. *Prune*: This phase of activity takes the refreshed groups and remaining anomalies from the past activity and rejects the outliers from the refreshed groups. A fading function is then conveyed to inspect matured groups and outliers and eliminate them from the data repository. The excess groups and the outliers are considered as the last outputs of the cycle, fit to be utilized as contributions for the following round of clustering.

The third fundamental part of data stream clustering is with respect to the discovery and support of anomaly data objects, for example those data protests that don't have a place with any of the current and recently shaped groups. For static data clustering, the outliers are at last external the last groups and can't be of the interest. In any case, for data stream clustering, regardless of it is finished by utilizing the incremental methodology or two-phase learning approach [8], the current outliers can become group individuals in the following round of clustering considering another approaching data chunk. Overlooking outlier data objects at each round of a continuous clustering measure implies losing numerous conceivable group individuals over time, seriously influencing the fulfilment of clustering. Because of the seriousness of the completeness issue, this part of data stream clustering must be appropriately tended to. As such, data stream clustering methods must have the option to distinguish and look after anomalies.

This paper explores broadly the current writing in the field of data stream clustering and distinguishes the basic preparing units supporting different existing methods. The paper then proposes a strategy to find outlier in educational data streams toward providing a response to evolving environments in near-real time. Practically, our research results can benefit a range of real-time big data applications such as sensor network monitoring, social media data analysis, etc.

Related Work

Numerous real-world data mining applications need to manage unlabelled streaming data. They are unlabelled in light of the fact that the sheer volume of the stream makes it unreasonable to name a significant portion of the data. Simultaneously, all of the data stream contains outliers and noises pose a significant and serious technical challenge to data stream clustering [6].

2.1 Data Streams Characteristics

The principle attributes of the data streams involve [9]:

- New data points arrive constantly at various velocities.
- Existing data points may get old and might be eliminated after they are processed.
- The size of data streams is enormous and likely unbounded.
- The data creating process may not be known and non-fixed, i.e. its likelihood distribution (patterns underlying the data) may change over time.
- Although there is a period succession of the data streams, there is no influence over the sequence in which the data points must be handled inside a similar chunk of data.

2.2 Static Data Versus Stream Data Clustering Techniques

The crucial thought of conventional clustering methods is to essentially utilize them on the static dataset. As such, the principle impediment of customary clustering methods is the point at which they are utilized for clustering progressively evolving data. All the more correctly, numerous distinctions exist between conventional data clustering methods and data stream clustering methods [8]: Firstly, customary clustering methods for static data are frequently iterative and filtering the dataset on various occasions. For instance, the K-Means method allocates and unassigns data objects to model groups ordinarily until the best-fit K groups are found. Then again, data stream clustering methods in a perfect world should filter the approaching data only for one time to catch the consistent appearance stream data. Furthermore, after executing the customary clustering methods on static data, the outcomes created are incremental and last while executing data stream clustering methods on the dynamic data produces estimated and temporal results that will be changed with regards to recently arrived data. At last, the execution time of conventional clustering methods is ordinarily not constrained, for example there might be no great crucial time direness for creating the clustering results. Nonetheless, data stream clustering methods are frequently very time basic, for example the comparing changes to the group model must occur

progressively. Therefore, conventional clustering methods are not, at this point ready to meet the necessities of data stream clustering and should be improved.

2.3 Requirements of Data Stream Clustering Algorithms

It is hard for the research community to concede to a set of requirements for data stream clustering methods with the goal that the necessities can be utilized as a benchmark to assess the adequacy of any new methods. Nonetheless, a few attractive necessities seem to have been agreed [10] which will likewise be deliberately viewed as in this paper:

- Iteratively refreshing clustering results. The subsequent groups should be consistently and more than once refreshed to accommodate changes brought about by the recently arrived data.
- Building and refreshing clustering models effectively. The transient nature of data streams proposes that data arrive at a quick movement and clustering task must be finished inside an exacting time limit to synchronize with the data changes.
- Being ready to handle cluster evolutions and the concept drift considering the fresh arrive and the decay of obsolete data objects and groups.
- Making the group model accessible whenever, either utilizing the incremental methodology or the two-phase learning approach.
- Detecting the presence of anomalies. This specific necessity might be adequate however not fundamental. Many existing techniques don't restore groups just as anomalies.
- Providing a compact model representation which develops all the more gradually with the quantity of data points handled. A portion of the significant qualities of the data representation include: (a) computationally simple to update, (b) having the option to adjust changes of the underlying of the data streams creating process, (c) having the option to store chronicled data and forget them on the off chance that they become old, and (d) supporting statistical analysis of the resulting groups [11].

All the above prerequisites have been considered in this research into data stream clustering. More subtleties will be given in the following sections.

2.4 Data Stream Clustering: Promises and Challenges

Data stream mining is empowered by arising applications including immense datasets. Coming up next are a portion of these applications:

1. Stock market examination: Prices of stocks are expanding and decreasing after some time. Price data is ceaselessly produced progressively during exchanging. In any case, some stock costs increas and fall simultaneously in

certain time spans. Such stocks can be gathered utilizing stream clustering methods. This data will be useful for speculators to arrange their stock portfolios and choose when it is the best possible time for selling or purchasing stocks [9].

2. Sensor networks: One of the well-known sensor networks is in the medical care framework. For example, modernized emergency clinics are furnished with a patient reconnaissance framework to upgrade medical care quality and staff efficiency. Since sensor gear simply store changed data and the natural eyes can't distinguish these signs, the framework ought to break down medical care data streams in an on-going and concentrate helpful data for restorative experts to recognize significant occasions [8].
3. Water distribution networks: The assessment of the drinking water-quality is in a perfect world extraordinary scale and an on-going reconnaissance application. Water-quality is a proportion of the water's condition and refers to the physical, organic, and compound features. Water-quality estimations produce enormous measure of stream data that should be handled. Li *et al* [12] executed tests with two distinctive dissemination networks. The principal network is a real water dissemination framework with 129 hubs. The subsequent organization incorporates 920 stations, arrives at the middle for water frameworks at the University of Exeter. The creators introduced a clustering method that constantly removed delegates out of colossal data streams. To persistently distinguish the delegates in a proficient manner, the creators applied online agent change measures just when significant clustering advancement happens.

Data streams have fundamental properties, for example, endless size, consecutive request, and dynamical updates. Thusly, creating compelling data stream clustering methods is basic for the investigation of such data. Simultaneously, the entirety of the data stream properties in addition to the realities that data stream is non-deterministic and consistently contains anomalies and noises represent a critical and genuine specialized test to data stream clustering [6].

2.5 Outliers Detection

There were explicit endeavours to distinguish the anomaly protests in the data streams. Thakran and Toshniwal [13] introduced a method that joined the K-Means standards and the DBSCAN standards to decide an anomaly. The method further uses a weighted K-Means method for weighting properties in deciding anomalies. Koupaie *et al* [14] created two answers for identifying anomalies in data streams. The principal arrangement [15] misuses both clustering and characterization strategies: the data points are first gathered into groups utilizing the K-Means

strategy. Anomalies that are a long way from the centroids (contingent upon an edge) are additionally recognized. At that point both group individuals and outliers are marked. From that point onward, a SVM clustering model is worked to arrange anomalies. The second arrangement [14] applies a clustering method with two equal stages. The main online stage applies the K-Means method for clustering data in the current chunk. From these groups, clusters of little sizes and data points far away from others are considered as anomalies and put away for additional utilization in the subsequent stage. The second disconnected stage consolidates recently distinguished outliers with anomalies of the current window chunk and present last anomalies. Natchial *et al* [16] introduced a hybrid two-stage arrangement by joining two distinct methods. The main stage is to assemble the data into Clusters-inliers and Clusters-outliers utilizing the K-Means method. The subsequent stage is to build a divergence framework to discover the disparity degree to additionally distinguish the genuine global outliers.

As of late, Kontaki *et al* [17] proposed a method name AMCOD to distinguish anomalies in data streams. It is a sliding-window technique focused in on distance-based outliers. The thought behind this method is that an article x is considered as an anomaly if there are not as much as n neighbour objects lying a ways off R from x. Notwithstanding, fixing the two limits (n, R) are difficulties in this method. Furthermore, there are a ton of competitor outliers that should be kept until concluding they are genuine anomalies or not. Another on-going method named SAIC was proposed by Zheng *et al* [18] for clustering discretionary shapes dynamic datasets. It incorporates learning and post-handling stages. The learning stage constantly distinguishes the up-and-comer groups and the post-preparing stage eliminates the anomalies by using the counter estimation of each group and the quantity of emphases. All in all, eliminates the limit points relying upon an edge.

The techniques looked into above have some helpful thoughts, for example, the two-stage method introduced in [14] for deciding little size groups and distant data objects as outliers. However, the strategies revealed are hybrid arrangements with high computational expense, and thus may not be material in adapting to the genuine enormous data streams. Truth be told, the group inliers and group outliers are different sides of a similar issue. The disclosure of both can be accomplished inside a solitary method, which can be a more ideal way to deal with take care of the two issues. Furthermore, we have contended for outlier discovery as a vital advance for data stream clustering.

The arrangement must be as finished as could be expected under the circumstances, for example every data point in a group or being an anomaly. Naming, each new data point ought to have a place with one of the current groups or it should be considered as outliers (noises). Certain groups may decrease over

the long time because of a portion of its individuals are matured; such a group may at last turn into an anomaly. As an important advance, the anomaly discovery issue must be tended to by an data stream clustering arrangement.

2.6 Existing EINCKM and EDDS Algorithms for Clustering Data Streams

We start by extensively researching the present status of-the-art in data stream grouping, looking over different kinds of existing methods announced in the writing, and dissecting the shared characteristics behind these sorts of methods just as their impediments and deficiencies.

- **EINCKM Algorithm**

EINCKM [19] is an incremental strategy for grouping model data streams. It relies upon a nonexclusive system for data stream clustering that includes three fundamental measured advances [20] Build Clusters (BC), Merge, and Prune. Build Clusters incorporates the clustering method that used to discover the groups from input data chunk, Merge (stage 2) is utilized to coordinate the new and existing arrangement of groups, and Prune (stage 3) is to identify outliers and check the fading cycle. The strategy utilizes a heuristic way to deal with anticipate the *K* (number of groupings), a radius figuring to consolidate covered groupings and a fluctuation way to deal with recognize the anomalies. The strategy is adaptable and prepared for additional upgrade. Be that as it may, the strategy created to introduce curved shape groups. Naming, it doesn't distinguish right groups in the event that they framed discretionary shapes.

- **EDDS Algorithm**

EDDS [20] is incremental power based technique for grouping data streams. It seeks after a comparative structure for data stream grouping that incorporates three rule steps Build Clusters (BC), Merge, and Prune. The strategy recognizes groupings and abnormalities in a moving toward data chunk. It changed the ordinary DBSCAN technique to gather each grouping with respect to a number of surface-point data records. The strategy executes the intensity reachable thought of DBSCAN as its consolidating method and prunes within focus using a heuristic course of action. The technique similarly deletes the old communities and abnormalities depending upon a fading cycle. In any case, this method has high method time contrasting to EINCKM. In addition, it doesn't separate the output shape groups, for example it doesn't recognize arched or non-raised shape groups.

Proposed ODED Algorithm

The adapted procedural structure of clustering data streams that includes three principle successive advances, for example Build Clusters, Merge, and Prune. Build Clusters is a stage to recognize groups from an approaching data chunk. Merge thinks about a potential blend of new and existing groups. Prune is a stage to discover remaining anomaly protests in the wake of blending groups. A great deal of the current methods (for example Inc.KM) are fragmented in light of the fact that they overlook so numerous data really busy handling data streams considered them as anomalies which likely could be individuals from a recently arisen group later.

Blending basis and keeping the anomalies independently not grouping them into various constant groups contrasted and ground truth. Given the need of anomaly ID in clustering data streams for completeness, methods following this system will have the additional favourable position over those that don't. The method is intended to identify new groups in an approaching data chunk, combine new groups and existing anomalies to present existing groups, create changed groups and outliers prepared for the following round, and inserting concept drift. Figure 1 shows the pseudo-code of the ODED method. The data sources are a data chunk of size *M*, a pool of anomalies, the base number of data points per chunk, and a group of existing groups rundown. Each group outline is a tuple (*N*, *LS*, *LSS*, μ, *R*), where *N* is various data points, *LS* is the straight amount of the data points, LSS is the amount of squared data points, μ is the centroid, *R* is the radius. The outputs are *K* groups and anomalies.

ODED Algorithm:

Inputs:

CH : Data chunk of size M. Initially *CH = {}*

CF: Set of clusters summary (N, LS, LSS, μ, R); // Previous clustering summary of *T* clusters. Initially, *CF = {}*

Po: Pool of outliers; // Previous Pool of *W* outliers. Initially, *Po = {}*.

$MinPts$: Minimum number of data points per cluster.

Outputs:

CF`: Modified CF;

Po`: Modified Po;

Algorithm Steps:

1. $CH = CH \cup Po$;
2. $cf = EINCKM(CH, K, Ini)$ *or* $cf = EDDS\ (CH, Eps, MinPts)$;// cf is a structure contains cf.member as cluster members & cf.summary as cluster summary.
3. Calculate cluster summary for S_i// i.e. *N*, *LS*, *L SS*.
4. $CF = Merge(CF, cf)$; // Merge overlapping clusters.

5. $\langle CF, Po\rangle = Prune(CF, cf, MinPts)$; // Filter outliers.
6. *Detecting the presence of outliers.*

Figure 1: Pseudo-code of the proposed algorithm

The Build Clusters step in our method comprises of two capacities EINCKM and EDDS. The Merge step in the overall structure is actualized by a solitary Merge process. The pruning step of the overall structure is actualized by a solitary Prune process as appeared in Figure 2. The function utilizes the difference of each group and MinPts limit to separate the anomalies from the groups delivered by the Merge process. The function examines the data points in every last group and contrasts their distance to the centroid and the range of that group. On the off chance that the distance is more noteworthy than the radius, the data point is considered as an outlier.

Prune Function:

$\langle CF, Po\rangle = Prune\ (CF, cf, MinPts, \lambda)$
$For\ i = 1\ to\ size(cf)$
{
$For\ j = 1\ to\ size(S_i)$ // cf.member
{
$D = dist(p(j), \mu_i)$; // p is a data point in S_i
$If\ D > R_i\ \ then\ \ Po = Po \cup p(j)$;
}
$If\ size\ (S_i) < MinPts\ \ then\ \ Po = Po \cup S_i$;
Update cluster summary for S_i
}
$For\ i = 1\ to\ size(CF)$ // Check the aged core points
$If\ (t_c - t_o) = F(t)\ \ then\ \ CF(i) = \{\}$;
$For\ i = 1\ to\ size(Po)$ // Check the aged outliers
$If\ (t_c - t_o) = F(t)\ \ then\ \ Po(i) = \{\}$;

Figure 2: Pseudo-code for the prune function

Implementation ODED Algorithm in Clustering Data Streams

4.1 Application dataset

Various certifiable applications can be discovered like climate, design, and consumer habits and so forth where the ideas repeat when the relating settings rehash. Concept repeat is a typical situation in some certifiable applications. At the point when ideas are identified with concealed settings, re-appearance of settings prompts concept repeat. The adjustments in time, financial conditions, style and so

on are instances of settings that drive related ideas in true streams. In this exploration we will rely upon instructive data streams as contextual analysis and spotlights on Viber Group in our department. Viber stream data of our department is utilized for this investigation. Data gathered at a half year span contains nine boundaries; Event_No., Event_Release, Event_Type, Start_Date, Start_Time, End_Date, End_Time, Attach, and Response. The stream begins 1/1/2020 and completes in 1/7/2020. The disconnected clustering is performed on the underlying 50 examples to make the main model. The accompanying Table 1 incorporates data about properities that sums up from the subjects and reaction messages that have been made.

Table 1: Sample of Viber Group data stream

Event_ No.	Event_Rel ease	Event_Ty pe	Start_D ate	Start_Ti me	End_D ate	End_Ti me	Atta ch	Respo nse
1	Head	Congratula tion	1/1/202 0	11:19am	1/1/202 0	12:00p m	1	31
2	Participant	Intimation	1/1/202 0	11:45am	1/1/202 0	2:20pm	0	6
3	Participant	Intimation	1/1/202 0	3:34pm	1/1/202 0	7:08pm	1	15
…	…	…	…	…	…	…	…	…
228	Participant	Intimation	30/6/20 20	11:41pm	30/6/20 20	11:49p m	0	17

4.2 Embedding the Outlier Detection into the Prototype-based Clustering Algorithm

To accommodate outlier detection in the prototype_based methods (for instance EINCKM method), data is maintained about when each group was made and when it was last changed. In the Prune process, a fading function is presented utilizing a fade window model in which the weight of every data object decease dramatically with time t by means of a fading function F: $F(t) = 2^{-\lambda.t}$…(1), where $0 < \lambda < 1$ characterizes the pace of decay of the weight over the long run and $t = (t_c - t_o)$, where t_c indicate the current time and t_o is the make/alter time of the data point[21].

With the fading function, the accompanying advances are taken to deal with outlier recognition issue with regards to a lifetime L (a user characterized limit, for example L=10 min.) and a fading boundary λ (we set λ to 0.3 in our analyses):

1. Makes and keeps up two timestamp tables:

   A. CTS (group timestamps) containing sections [centroid, timestamp] for each group.
   B. OTS (anomaly timestamps) containing sections [outlier, timestamp] for every outlier.
2. For every cycle of handling new approaching data chunk do:
   A. Figures the estimation of the fading function *F* utilizing the equation (1);
   B. Reduce the timestamp by *F* sum for CTS and OTS;
   C. Sets the timestamp for new centroids and anomalies to an edge *L*;
   D. For consolidated groups, refreshes all influenced centroids by setting the timestamp to *L*;
   E. Erases all centroids and outliers which have terminated (for example *L*=0).

4.3 Embedding the Outlier Detection into the Density-based Clustering Algorithm

To accommodate outlier identification in the density_based method (for instance EDDS method), data is maintained about when each group was made and when it was last altered. In the Prune process, a fading function is presented utilizing the equation (1). To deal with anomaly discovery issue, like the methodology taken in Section 4.2, in the Prune step, we present a fading function that:

1. Sets a fading boundary λ to 0.3 in our examinations.
   A. Makes and keeps up two tables for timestamp observing:
   B. CTS containing sections [surface-centers, timestamp] for each group.
   C. OTS containing passages [outlier, timestamp] for every outlier.
2. For every cycle of handling new approaching data chunk do:
   A. Figures the estimation of the fading function *F* utilizing the equation (1);
   B. Decrease the timestamp by F sum for CTS and OTS;
   C. Sets the timestamp for new surface-centers and outliers to *L* (for example *L*=10);
   D. For consolidated groups, set the timestamp to *L* for all influenced surface-centers;
   E. Erases every surface-center and anomalies which have lapsed (for example *L*=0)

4.4 EVALUATION of ODED METHOD

Various ways to deal with assess clustering method execution exist in the writing [22]. To assess the accuracy of the proposed method, we have chosen to utilize the

assessment by the reference strategy, for example to discover if the method can restore the known groups in a given ground truth. To assess rightness, we utilized three generally utilized evaluators: purity, entropy, and the sum of squared errors (SSE). Purity was utilized in [23], entropy in [24], and SSE in [4]. Purity refers to the extent of the data guides having a place toward a referred to group that are assigned as individuals from a group by the method. The higher the extent of purity (between [0, 1]) is, the more sure that the method has discovered the first groups and the better the method is [25]. Entropy mirrors the quantity of the data points from various known groups in the original dataset that are relegated to a group by the method. The estimation of this measure is between [0, $Log_2 N$] where N is the quantity of realized groups included. The more modest estimation of the entropy is, the less individuals from the realized groups are blended in the groups found by the method, and the better the clustering method is [8]. SSE is a generally utilized group quality measure. It assesses the conservativeness of the subsequent groups. Low scores of SSE demonstrates better clustering results as the groups contain less inner varieties [25]. The proficiency of a method was estimated by the measure of time in seconds taken for the method in finishing the clustering task.

MATLAB was utilized to assemble an execution of the EINCKM, EDDS, and ODED methods and the examination structure. We split a given dataset into two sections: the dynamic arrive data chunks of a specific size and the underlying dataset before the appearance of the primary powerful data chunk. We randomly chose an underlying assortment of 50 data points as the underlying dataset and the leftover 178 were arbitrarily chosen as data points in the dynamic chunks. Our proposed method doesn't treat the underlying dataset and later arrived chunks in an unexpected way, and thus an empty arrangement of existing groups and an empty arrangement of outliers were expected as the sources of info when the primary chunk is handled. The thought behind the irregular determination of the data points is to research the conduct of the methods when there is no control on the clustering of data points, for example we didn't choose explicit data points from explicit groupings in the original datasets. No supposition that was made that the underlying data chunk represents to the whole data area. To limit the impact of arbitrary decision of data points, the investigations were rehashed multiple times, and the normal is determined.

- ***Purity***

Figure 3 shows the subtleties of assessment results between the known groups and the output groups from EINCKM, EDDS, and ODED procedures independently. ODED has the most significant purity. This is considering the way that it keeps all the agent data centres and stringent favourable to centre. EDDS in like way has a decent flawlessness by pruning and sparing the surface-centers data objects, at any

rate this methodology excuses non-raised shape groups. EINCKM has a reasonable flawlessness also and slights a lot of focus data records which may not affect the last groupings.

- ***Entropy***

As appeared in Figure 4, ODED has the base entropy. EDDS has progressively raised proportion of entropy. EINCKM has the most peculiar proportion of entropy among the three techniques. The outcomes exhibit that the pruning inside focus data centres influences group accuracy. However, this outcome should be inspected along with the purity assessment results to have a sensible view on clustering accuracy.

- ***Sum of Square Error (SSE)***

As appeared in Figure 5, ODED has the less SSE, followed by EDDS which in this way is followed by EINCKM, again showing the expense of pruning inside focus data centres. Of course, EDDS EINCKM still have the most extremely low SSE score, illustrating that blending incorrectly data centre into found groups does in like way impact group quality. It should be referred to that SSE may not be the ideal evaluator for nature of non-raised shape groupings.

Efficiency Evaluation

- ***Execution Time***

Execution time is the degree of the extent of time in seconds that proceed for the technique in finishing the clustering task. Concerning use time, the ODED method has the base execution time searched after by EINCKM, by then EDDS (see Figure 6).

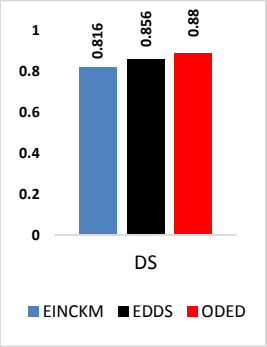

Figure 3: Purity

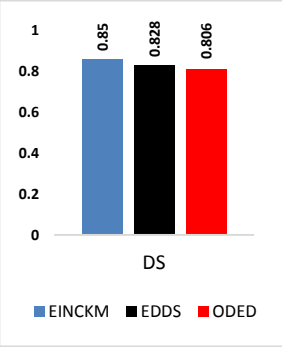

Figure 4: Entropy

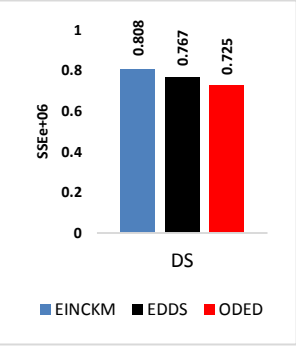

Figure 5: SSE

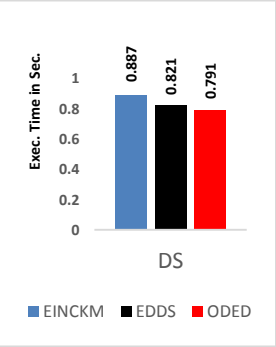

Figure 6: Efficiency

Adaptation Prototype-based and Density-based Methods to the Outlier Detection

Anomaly discovery has been perceived as one significant issue in data stream clustering [8]. In clustering, outlier identification refers to the developmental changes to group models over the long time. Static data clustering just has one concept: the general model of groups while data stream clustering may have different ideas that develop over the long time. We distinguished anomaly identification at two levels: the transformation level and change checking level. At the transformation level, our method, by following the incremental methodology of data stream clustering, consistently refines the current model of groups considering the recently arrived data chunk, and thus consistently adjusts to the progressions reflected by new groups added into the model or change to the current groups by means of consolidating. At the change checking level, the proposed method itself doesn't keep a set of experiences trail of the changed group models. Truth be told, execution of the method may utilize variable boundaries to keep a solitary duplicate of the new model of groups, over-composing the past model.

Conclusion

In this paper we concentrate how the clusterings delivered by various data streams clustering methods change, comparative with the ground truth, as quantitatively various sorts of outlier location are experienced. This paper makes two commitments to the writing. In the first place, we propose a strategy for creating genuine value data streams with exact quantitative outlier location. Second, we lead an exploratory investigation to give quantitative examinations of data stream clustering strategies execution with genuine value data streams and tell the best way to apply this data to real data streams. Future work will focus on refreshing the technique. Due to the technique is adaptable; those refreshing ideas can manage the significant elements of the strategy. In the beginning, we will examine the topology method to introduce more modern adaptation of the embedding outlier discovery step. Besides, we will examine hybridizing diverse fading functions, as neural network based, swarm-based, and genetic based functions to test the measured quality of ODED technique.

Reference

[1] M. Hassani and T. Seidl, “Clustering Big Data streams: recent challenges and contributions,” *It-Information Technol.*, vol. 58, no. 4, pp. 206–213, 2016.

[2] S. Guha, N. Mishra, R. Motwani, and L. O’Callaghan, “Clustering Data Streams,” *0-7695-0850-2/00 $10.00 0 2000 IEEE*, pp. 359–366, 2000.

[3] D. A. Marcos, N. C. Rodrigo, B. Silvia, A. S. N. Marco, and B. Rajkumar,

"Big Data Computing and Clouds: Trends and Future Directions," *J. Parallel Distrib. Comput.*, no. arXiv:1312.4722v2, pp. 1–44, 2014.

[4] C. Aggarwal, J. Han, J. Wang, and P. Yu, "A Framework for Clustering Evolving Data Streams," *Proc. 29th VLDB Conf. Ger.*, 2003.

[5] C. Isaksson, "New Outlier Detection Techniques For Data Streams," Thesis, Southern Methodist University, Bobby B. Lyle School of Engineering, 2016.

[6] S. Ding, F. Wu, J. Qian, and H. Jia, "Research on data stream clustering algorithms," *Springer*, vol. Artif Inte, pp. 593–600, 2013.

[7] F. Stahl, A. Badii, M. Oldenburg, and F. Theodorstahldfkide, "Building Adaptive Data Mining Models on Streaming Data in Real-Time," *Comput. Intell.*, vol. 3, no. 2, p. 12, 2020.

[8] H. L. Nguyen, Y. K. Woon, and W. K. Ng, "A survey on data stream clustering and classification," *Knowl. Inf. Syst. Springer*, pp. 535–569, 2015.

[9] Yogita and D. Toshniwal, "Clustering Techniques for Streaming Data – A Survey," *3rd IEEE Int. Adv. Comput. Conf.*, pp. 951–956, 2012.

[10] A. Amini, "An Adaptive Density-Based Method for Clustering Evolving Data Streams," Thesis, University of Malaya Kuala Lumpur, Department of Computer Science and Information Technology, 2014.

[11] R. N. Davies, "Efficient Analysis of Data Streams," Thesis, Lancaster University, Department of Computing and Communications, 2017.

[12] Q. Li, X. Ma, S. Tang, and S. Xie, "Continuously identifying representatives out of massive streams," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7120 LNAI, no. PART 1, pp. 229–242, 2011.

[13] Y. Thakran and D. Toshniwal, "Unsupervised outlier detection in streaming data using weighted clustering," *2012 12th Int. Conf. Intell. Syst. Des. Appl.*, pp. 947–952, 2012.

[14] H. M. Koupaie, S. Ibrahim, and J. Hosseinkhani, "Outlier Detection in Stream Data by Clustering Method," *Int. J. Adv. Comput. Sci. Inf. Technol.*, vol. 2, no. 3, pp. 25–34, 2013.

[15] H. M. Koupaie, S. Ibrahim, and J. Hosseinkhani, "Outlier Detection in Stream Data by Machine Learning and Feature Selection Methods," *Int. J. Adv. Comput. Sci. Inf. Technol.*, vol. 2, no. 3, pp. 25–34, 2013.

[16] J. Natchial F., E. P., and T. B., "Hybridizing Clustering and Dissimilarity Based Approach for Outlier Detection in Data Streams," *Int. Sci. Press*, vol. 9, no. 3, pp. 127–131, 2016.

[17] M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsichlas, and Y. Manolopoulos, "Efficient and flexible algorithms for monitoring distance-based outliers over data streams," *Inf. Syst.*, vol. 55, pp. 37–53, ISBN 9781424489589, 2016.

[18] L. Zheng, H. Huo, Y. Guo, and T. Fang, "Supervised Adaptive Incremental Clustering for data stream of chunks," *Neurocomputing*, vol. 219, no. September 2016, pp. 502–517, ISBN 18728286, 2017.

[19] A. Al Abd Alazeez, S. Jassim, and H. Du, "EINCKM: An Enhanced Prototype-based Method for Clustering Evolving Data Streams in Big Data," *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods*, no. Icpram, pp. 173–183, 2017.

[20] A. Al Abd Alazeez, S. Jassim, and H. Du, "EDDS: An Enhanced Density-Based Method for Clustering Data Streams," *2017 46th Int. Conf. Parallel Process. Work.*, pp. 103–112, 2017.

[21] K. Udommanetanakit, T. Rakthanmanon, and K. Waiyamai, "E-Stream: Evolution-Based Technique for Stream Clustering," *Springer-Verlag Berlin*, vol. 4093, no. March 2014, pp. 42–55, 2007.

[22] H. Kremer *et al.*, "An effective evaluation measure for clustering on evolving data streams," *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '11*, pp. 868–876, 2011.

[23] 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.

[24] Y. Zhao and G. Karypis, "Technical Report Criterion Functions for Document Clustering: Experiments and Analysis," *Univ. Minnesota, Dep. Comput. Sci. / Army HPC Res. Center/ Tech. Rep.*, pp. 1–30, 2001.

[25] J. Silva, E. Faria, R. Barros, E. Hruschka, and A. Carvalho, "Data Stream Clustering : A Survey," *ACM Comput. Surv.*, pp. 1–37, 2013.