HPPD: A Hybrid Parallel Framework of Partition-based and Density-based Clustering Algorithms in Data Streams

Ammar Th. Al Abd Alazeez

ammarthaher@yahoo.com

Department of Computer Science
College of Computer Science and Mathmatics
University of Mosul, Mosul, Iraq

Received on: 26/05/2019 **Accepted on: 18/06/2019**

ABSTRACT

Data stream clustering refers to the process of grouping continuously arriving new data chunks into continuously changing groups to enable dynamic analysis of segmentation patterns. However, the main attention of research on clustering methods till now has been concerned with alteration of the methods updated for static datasets and changes of the available modified methods. Such methods presented only one type of final output clusters, i.e. convex or non-convex shape clusters. This paper presents a novel two-phase parallel hybrid clustering (HPPD) algorithm that identify convex and non-convex groups in online stage and mixed groups in offline stage from data stream. In this work, we first receive the data stream and apply pre-processing step to identify convex and non-convex clusters. Secondly, apply modified EINCKM to present online output convex clusters and modified EDDS to present online output non-convex clusters in parallel scheme. Thirdly, apply adaptive merging strategy in offline stage to give last composed output groups. The method is assessed on a synthetic dataset. The output results of the experiments have authenticate the activeness and effectiveness of the method.

Keywords: Big Data; Hybrid Clustering Algorithms; Data Stream Clustering Algorithms.

نموذج متوازي هجين لخوارزميات العنقدة للبيانات المستمرة بالاعتماد على التقسيم والكثافة للبيانات

عمار ظاهر ياسين ال عبد العزيز

قسم علوم الحاسوب

كلية علوم الحاسوب والرياضيات

جامعة الموصل، الموصل، العراق

تاريخ استلام البحث: ٢٦/٠٥/٢٠١٩ **تاريخ قبول البحث: ١٨/٠٦/٢٠١٩**

الملخص

المصطلح عنقدة البيانات المستمرة يشير الى عملية توزيع مستمرة للبيانات الجديدة والمتولدة بشكل مستمر الى مجاميع قابلة للتغيير بشكل مستمر لتمكين عملية التحليل المتزامنة للانماط الجديدة. على اية حال، توجه البحوث في مجال خوارزميات العنقدة الى وقتنا هذا متركزة على تحديث هذه الخوارزميات والتي تعمل مع البيانات

الثابتة الى بيئة البيانات المستمرة او تطوير خوارزميات البيانات المستمرة. هذه الخوارزميات تقدم فقط نوع واحد من العناقد المخرجة والتي تكون اما عناقيد كروية او عناقيد غير منتظمة الشكل. هذا البحث يقدم خوارزمية متوازية هجينة جديدة تدعى HPPD والتي تميز العناقيد الكروية والعناقيد غير الكروية في الطور المباشر وكذلك تميز العناقيد المشتركة في الطور غير المباشر. في هذا البحث، اولا نقوم باستلام البيانات المستمرة ونطبق عليها عمليات تهيئة استباقية لاكتشاف العناقيد الكروية وغير الكروية. ثانيا، نقوم بتطبيق نسخة محدثة من خوارزمية EINCKM على العناقيد الكروية وكذلك نطبق نسخة محدثة من خوارزمية EDDS على العناقيد غير الكروية وهذا يتم في الطور المباشر. ثالثا، نطبق ستراتجية دمج جديدة للحصول على العناقيد المختلطة النهائية. هذه الخوارزمية تم فحصها على بيانات افتراضية لغرض معرفة مدى فاعليتها. النتائج النهائية للتجارب وثقت فاعلية وفائدة الخوارزمية المقترحة ومدى فرقها عن سابقاتها.

الكلمات المفتاحية: البيانات الكبيرة؛ خوارزميات العنقدة الهجينة؛ خوارزميات العنقدة للبيانات المستمر.

1. Introduction

Late technologies in data and systems administration innovations and their applications in pretty much every area of life [1]have prompted a quickly developing transition of huge measure of information known as Big Data. Such big data might be put away at different areas with various configurations. Big Data need opportune investigation for upgrading intensity and improving the exhibition of organizations [2]. A standout amongst the most significant attributes of big data is its speed (velocity), which implies that information may arrive and require handling at various paces. Some systems, the entry and preparing of information can be performed in a bunch handling style, others need consistent and continuous investigation of approaching information stream blocks [3][4]. Grouping stream data is characterized as the gathering of information in light of much of the time arriving new information in pieces for increasing comprehension about basic gathering patterns that may change after some time in the stream data [5].

There is no one method is reasonable for a wide range of data records, nor all methods suitable for all issues. Conventional grouping methods have surely understood weaknesses, for example, reliance on the underlying state, assembly to nearby optima, worldwide arrangements of enormous issues can't have existed with sensible measure of calculation exertion and so forth.

Up to now, combination of two or three different clustering algorithms to identify both convex and non-convex clusters is new to clustering data stream. Hybrid method may conceive full utilize of advantages of different optimization approaches. Our believe is that the hybridization of clustering data stream methods could give more benefits and advantages. Therefore, the goal of this research is to give a novel way of combining data stream grouping methods.

Methods of clustering data streams present sequence of clustering views periodically (incremental learning approaches) [6] or depend on user query point (two-phase learning approaches) [7]. However, these algorithms presented only one type of final output clusters, i.e. convex or non-convex shape clusters. In this research, we dispute that methods for big data stream grouping must present convex, non-convex, and mixed output clusters. To do so, this paper presents a novel two-phase parallel hybrid clustering (HPPD) algorithm that identifies convex and non-convex groups in online stage and mixed groups in offline stage from data stream. In this work, we first receive the data stream and apply pre-processing step to identify convex and non-convex clusters. Secondly, apply modified EINCKM (Enhanced INCremental K-Means) algorithm to present online output convex clusters and modified EDDS (Enhanced Density-based Data Stream) algorithm to present online output non-convex clusters in a parallel scheme. Thirdly, apply adaptive merging strategy in offline stage to produce eventual mixed output groups. The method is assessed on a synthetic dataset.

The hybridization of methods is a general acceptance idea via their abilities in managing nowdays challenges that contain complicated, inaccurate and ambiguity (uncertinty). Many applications could take advantages from designing hybrid system (a two-phase parallel model of finding convex, non-convex, and mix groups) in stream data records. Act tracking of some applications like social networking, cars, rockes and animals tracking from videos, crime controling using CCTV camera in discovering curious entities such as unexpected cars, and patient advisor in healthcare are some of potential cases.

This paper presents a novel hybrid (HPPD) algorithm that detects convex and non-convex shape clusters separately in an incremental learning scheme and mix clusters in a two-phase learning scheme depending on the user demand from streaming data. In this work, we first modify our recent EINCKM [8] algorithm and EDDS [9] algorithm. After that, designed a parallel model to apply the two modified algorithms.

The method was assessed on an elected datasets utilizing different qualifications. The output test state that the suggested method enhances grouping accuracy. The outline of the method is created to be flexible for more enhancements of further modifications and paralelise the method.

The remain of this research is calssified as follows. Section 2 states the related work on hybrid clustering data stream methods. Section 3 illustrates our published two algorithms EINCKM and EDDS. Section 4 describes the problem in more details. Section 5 states the suggested HPPD method. Section 6 explains the assist of the effectivness of the method and practical tests utilizing a selected dataset. Section 7 summarizes the research and states the future thoughts of this work.

2. Related Work

All the current grouping methods have their own qualities, yet additionally remain imperfect [10]. As a sort of partitional grouping, K-Means method is basic and highly effective, yet it can just find convex shape groups. The partitional grouping, for example, K-Means, is delicate to the anomalies and centers. A good center we pick the better outcomes we gain. For the most part, the computational multifaceted nature of the hierarchical grouping is O(n^2), where n is the all-out number of data records. In this way, they are typically used to examine little datasets and can't repudiate the earlier grouping output. The hierarchical methods can discover non-curved groups, yet they are delicate to anomalies and are not appropriate for enormous databases. The grid-based grouping methods are not reasonable for high-dimensional datasets. Density-based (for example DBSCAN) can find groups of non-convex shape. In any case, it is delicate to the information parameters, particularly when the intensity of information is non-uniform. Then again, DBSCAN experiences issues with high-dimensional datasets. Hence, the hybrid solution is the best way to take advantages from different algorithms and methods and get rid of their limitations.

In the coming sections, we shall produce five concepts that are associated to have more understanding of the hybridization of data stream grouping methods. First of all, we shall sum up the researches in hybrid conventional grouping methods. Secondly, we will explain works of hybrid outlier detection algorithms. That is because detecting outliers in unsupervised learning algorithm demand finding clusters as well as outliers. Thirdly, since our general idea is to design a hybrid clustering algorithm from the other clustering methods, we will summarize some works in clustering ensemble methods. Finally, we shall state the researches that presented in hybrid data stream grouping methods.

A. Traditional Clustering Algorithms

Jain [11] described a hybrid clustering algorithm based on K-Means and K-Harmonic Means (KHM). It takes advantages of both algorithms to present method which is not affected by initial clusters and converge to a global minimum under certain conditions.

Sangam and Om [12] presented a hybrid algorithm integrates K-Modes and K-Means algorithms to allow clustering data points described by mixed data type, i.e. categorical and numerical attributes by using a combined dissimilarity measure.

Viswanath and Pinkesh [13] presented a novel scalable hybrid clustering method (L-DBSCAN) to get fast arbitrary shaped clusters by combining density-based clustering DBSCAN (which can discover consistent arbitrary shaped clusters along with detection of noisy outliers) with K-Means (which can find compact and spherical shaped clusters). First two types of prototypes are derived using Leaders Clustering method. These prototypes are carefully used by the DBSCAN to present clusters faster than DBSCAN using the entire dataset.

Dhiman et al. [14] used a hybrid system of clustering and classification algorithms to search for the data record of tax bills and find the useful knowledge regarding the tax magnitude. Naming, they hybridize three main algorithms, i.e. K-Means, SOM, and HAC from grouping and CHAID and C4.5 methods of classification. It presented good output results than the conventional methods.

Conventional algorithms are elite via assurance of discovering convex shape groups (such as K-Means) or arbitrary shape clusters (such as DBSCAN). However, the traditional methods also have their limitations, like find clusters in static data (they did not consider concept drift) and identify only one cluster type, i.e. spherical or arbitrary clusters. The goal of our paper is to evolve a dynamic grouping algorithm that discovers spherical and arbitrary shape clusters in online fashion.

B. Outlier Detection Algorithms

Jiang et al. [15] proposed a two-phase clustering algorithm for outlier’s detection. They first modified the traditional K-Means algorithm in phase one by using a heuristic technique (if new data point is far from existing clusters assign it as centre). And then they constructed minimum spanning tree MST in phase two and remove the longest edge. The small clusters in the tree with less number of nodes are selected as outlier.

Thakran & Toshniwal [16] presented an algorithm addressed outlier detection in data streams. For identifying anomalies two classes of grouping methods, density based, and partitioning are mixed. Then weighted K-Mean method is utilised for weighting properties based on their association, which lead to minimise the consequence of the inapplicable features.

Koupaie et al. [17] proposed combined classification and clustering techniques to detect outliers in data streams. More detailed, they used K-Means algorithm to cluster data points and add labels to each of them. After that, apply SVM algorithm to classify outliers.

Koupaie et al. [17] displayed a hybrid method of two stage grouping method to recognize anomalies in information streams. The two stages are kept running in parallel with one another. First stage is online stage in which K-Means method is utilized for grouping window information. From these groups, a few groups are little in size and a few groups are far from different groups these are called as anomalies and put away for further use in second stage. Second stage is offline stage in which all the recently identified anomalies are joined with anomalies of current window. At that point this information is grouped utilizing K-Means method and groups are created.

Although these methods identify clusters and outliers, they cannot identify clusters with different shape clusters.

C. Clustering Ensemble

Ensembles grouping is an important approach for enhancing authentication, efficiency, and activity of unsupervised classification methods [18][19]. There is a rich body of literature in this field:

Zhang *et al.* [20] proposed an ensemble method which combines both classifiers and clusters together for mining data streams through a weighted average mechanism. Fathzadeh and Mokhtari [21] introduced Stream Ensemble Fuzzy C-Means (SEFCM) method. It is separate and-vanquish technique contained three phases; 1) isolate information stream to littler chunks; 2) group each chunk utilizing ensemble grouping (EFCM) method; and 3) consolidate the finishing up segments utilizing single linkage and concentrate a flat out segment. Mutazinda et al. [22] exhibited a ensemble grouping strategy to group blended information (category and numerical). It executes Chameleon method for the numerical data and Squeezer method for the categorical data and joins the yields to display last groups.

Ensemble group is increasingly about the agreement of different grouping models, and not by any stretch of the imagination about recognizing convex and non-convex shapes.

D. Data Stream Clustering Algorithms

Aghabozorgi et al. [23] proposed a hybrid grouping method is presented depending on the likelihood in shape of the time series records. Time series objects are firstly clustered as sub-clusters depending on likelihood in time data record. The sub-groups are then consolidating utilizing the K-Medoids method depending on likelihood in shape.

Sree & Sowjanya presented [24] an algorithm which combined hierarchical and partitioning clustering methods. It applied hierarchical clustering first to decide location and number of clusters in the first round and ran the K-Means clustering in another round to identify output clusters.

Chen & He [25] proposed a hybrid distance evaluation method Str-FSFDP to cluster mixed data stream. This method was inspired by the density-based clustering and self-adaptive peak density clustering algorithms for data with mixed attributes.

Again these algorithms are focusing on one type of output clusters, i.e. spherical or arbitrary shaped clusters.

Lklklk

E. Data Stream Clustering Algorithms EINCKMand EDDS

• EINCKM Algorithm

EINCKM [8] is an incremental method for grouping prototype data streams. It depends on a generic framework for data stream clustering that involves three main modular steps [8] [9] *Build Clusters* (*BC*), *Merge*, and *Prune* (Fig. 1). *Build Clusters* includes the clustering algorithm that used to find the clusters from input data chunk, *Merge* (step 2) is used to integrate the new and existing set of clusters, and *Prune* (step 3) is to detect outliers and check the fading process. The method uses a heuristic approach to predict the *K* (number of groups), a radius calculation to combine overlapped groups and a variance approach to identify the anomalies. The method is flexible and ready for further enhancement. However, the method developed to present convex shape clusters. In other words, it does not identify correct clusters if they formed arbitrary shapes.

• EDDS Algorithm

EDDS [9] is incremental intensity-based method for grouping information streams. It pursues a similar framework for information stream grouping that includes three principle steps [8] [9] Build Clusters (BC), Merge, and Prune (Fig. 1). The method identifies groups and anomalies in an approaching information piece. It adjusted the conventional DBSCAN method to condense each group regarding a lot of surface-center data records. The method execute the intensity reachable idea of DBSCAN as its combining technique and prunes the inside center utilizing a heuristic arrangement. The method likewise expels the old centers and anomalies relying upon a fading process. However, this algorithm has high computation time comparing with EINCKM. Besides, it does not separate the output shape clusters, i.e. it does not distinguish between convex or non-convex shape clusters.

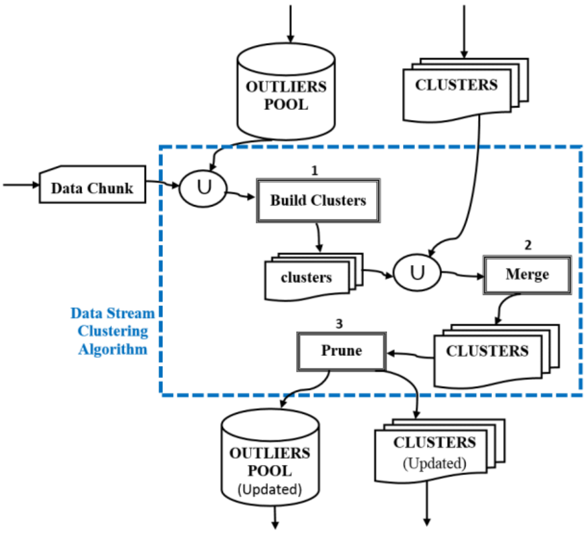

Fig. 1. The general framework of Data Stream Clustering Algorithms [8][9].

3. Problem Description

In this work, we are trying to identify different types of clusters, i.e. convex and non-convex shape clusters over time in the data stream. The basic idea is this (Please, see Fig. 2);

1. Get the stream data and split it into many data blocks (step 1).
2. Do “Pre-processing” to divide the data chunk into two types of clusters (convex clusters) and (non-convex clusters).
3. On one hand, the convex clusters go to the "Modified EINCKM" to always present convex output clusters (step 3-Left). On the other hand, the non-convex clusters go to the "Modified EDDS" to always present non-convex output clusters (step 3-Right). Step 3 we could call it “Online Phase” or “Incremental Learning” because they present output clusters incrementally (convex shape clusters from left hand side (Modified EINCKM) and non-convex shape clusters from right hand side (Modified EDDS)).
4. If there is a query from the user about the whole clusters (convex and non-convex or mix) we will apply “Adaptive Merger” (step 4). This merger will give us mixed clusters. We could call this step as “Offline Phase” or “Two-phase learning”.

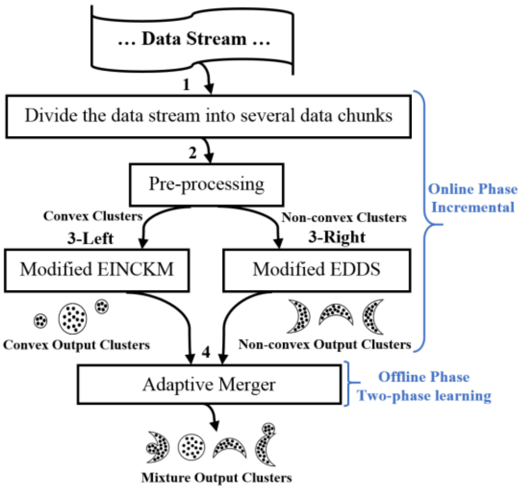

Fig. 2. General Idea

4. HPPD Algorithm

Recently, many researchers have been proposed algorithms using a hybrid scheme to solve the data steam clustering problem. This study is involved in building a clustering model that combine the advantages of our two-recent data stream clustering algorithms EINCKM and EDDS named HPPD.

Before officially elaborating the HPPD algorithm-level framework, we would like to introduce some preliminary knowledge to better facilitate its understandability. More precisely, we start with illustrating the outline of general system framework (Sec. 5.A), and the HPPD algorithm details (Sec. 5.B).

A. The outline of the suggested framework

The overall framework of the suggested HPPD method is illustrated in Fig. 3. It composes of the coming major stages:

1. Receive the data streams and divide it into several chunks (step 1).

2. Do “*Pre-processing*” (step 2). The output of this step is two types of clusters (*convex clusters*) and (*non-convex clusters*). This step includes:
   - Apply DBSCAN to the incoming data chunk (step 2.A).
   - For each cluster repeat (step 2.B): Find the boundary (BN) using our developed function in EDDS (step 2.C.Left). And find the convex-hull (CH) using Qhull algorithm (step 2.C.Right).
   - If BN = CH the cluster is a convex, otherwise it is a non-convex (step 2.D).
3. The convex clusters go to the "*Modified EINCKM*" (step 3.Left) to always present convex output clusters. This step includes:
   - Compute the centroids and radius (step 3.Left.A).
   - Do merging using Merge function in EINCKM (step 3.Left.B).
   - Do pruning using Prune function in EINCKM (step 3.Left.C).

On the other hand, the non-convex clusters go to the "*Modified EDDS*" (step 3.Right) to always present non-convex output clusters. This step includes:

- Compute the surface-cores (step 3.Right.A).
- Do merging using Merge function in EDDS (step 3.Right.B).
- Do pruning using Prune function in EDDS (step 3.Right.C).

4. If there is a query from the user about the whole clusters (*convex* and *non-convex* or *mix*) we will apply “*Adaptive Merger*” (step 4). This merger will give us mixed clusters. This step includes:
   - For all the convex clusters that come from “Modified EINCKM”, add artificial surface-cores which are data points far from the centroids of 2STD and the distance between the data points themselves is less than or equal *EPS* (This criterion helps merging strategy as “density reachable”) (step 4.A).
   - Do merging using Merge function in EDDS (step 4.B).
   - Remove the artificial surface-cores if there were no merging (step 4.C) and present the final output clusters.

Step 1, 2 and 3 we could call them “Online Phase” or “Incremental Learning” because they present output clusters incrementally (convex shape clusters from left hand side (Modified EINCKM) and non-convex shape clusters from right hand side (Modified EDDS)). We could call step 4 as “*Offline Phase*” or “*Two-phase learning*”.

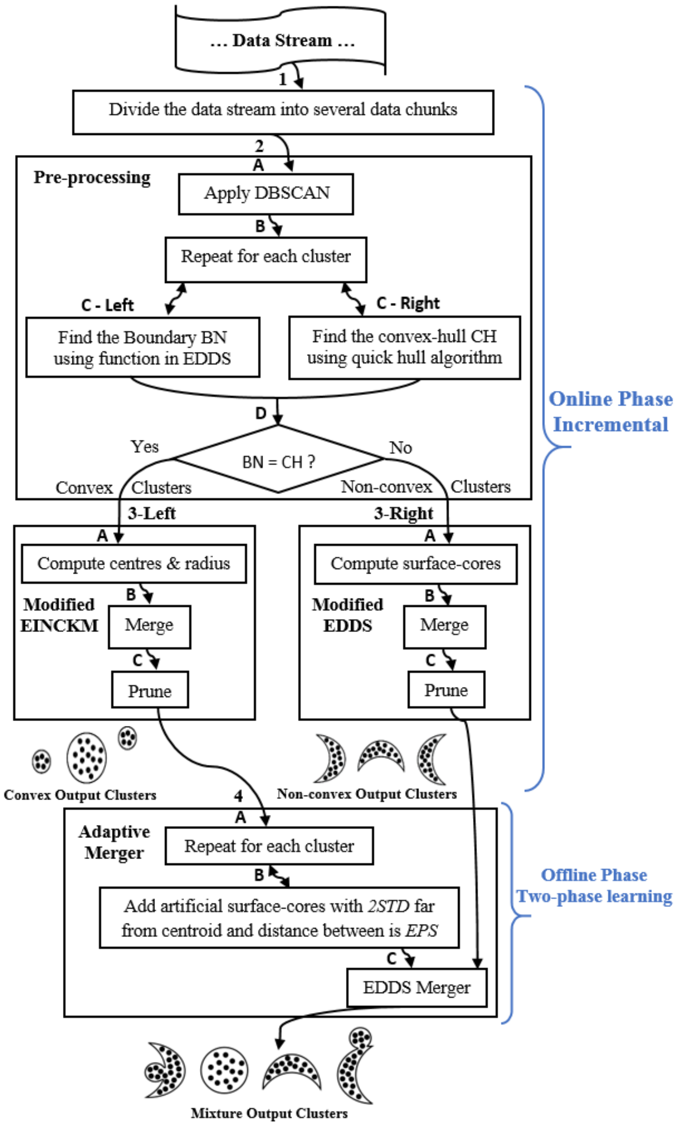

Fig. 3. General framework of the HPPD algorithm

B. Suggested HPPD Algorithm

After introducing the main preliminary steps of the algorithm, we are going to present the suggested Method. Fig. 4 summarizes the code explanation of the essential HPPD method. The inputs are a block of data *CH* of *M* capacity, the minimum number of data points per cluster *MinPts*, and epsilon *Eps*. The outputs are K spherical clusters and R arbitrary shape clusters.

```
Hybrid Algorithm:
Inputs:
- $DH$: Block of data of $M$.
- $MinPts$: The smallest amount of data records for each cluster.
- $Eps$: The radius of the minimum group.
Outputs:
- $K$ Convex Clusters;
- $R$ Non-convex Clusters;
Algorithm Steps:
  - Repeat for each data chunk
1. $\langle cf \rangle = DBSCAN(DH, Eps, MinPts)$;//$cf$ is a data record contain a summary group.
2. $\langle BN \rangle = Bound(CF, Eps, MinPts)$
3. $\langle CH \rangle = Qhull(CF)$
   $If\ BN = CH\ then$
      Calculate $\mu_i\ and\ \sigma_i$
       $\langle CF \rangle = MergeEINCKM(CF)$
      $\langle ConvexCF \rangle = PruneEINCKM(CF)$
   $else$
      Calculate SurfaceCores
       $\langle CF \rangle = MergeEDDS(CF)$
       $\langle NonConvexCF \rangle = PruneEDDS(CF)$
   $end$
4. $If\ there\ is\ a\ query\ for\ convex\ and\ nonconvex\ clusters\ then$
       $Add\ ArtificialSurfaceCores\ into\ ConvexCF$
       $\langle CF \rangle = MergeEDDS(CF)$
      $Remove\ ArtificialSurfaceCores\ from\ ConvexCF$
 $end$
```

Fig. 4. Code of the method.

5. Evaluation of the Suggested Algorithm

Here, we shall assess the effectiveness of the proposed method from the accuracy and efficiency concepts of the approach. Firstly, we will introduce some correctness criteria (Sec. 6.A). Then, we will present the synthesized datasets that will use for testing (Sec. 6.B). After that, explain the evaluation framework (Sec. 6.C). Finally, we will introduce the experimental results and discussion (Sec. 6.D).

A. Evaluation Criteria

Correctness of the clustering methods can be evaluated via various approaches which can be found in the literature [26]. This research choses the external (supervised) methods to assess the accuracy of the proposed method. We build a synthetic datasets which contain known groups as the ground truth for testing the "*closeness*" of the persistent clustering results produced by the algorithm to the ground truth. The closer the resulting persistent clusters to the known clusters, the more accurate is. Those criteria including entropy, purity, and the sum of squared errors (SSE).

Purity was utilized in [27], entropy in [28], and SSE in [7]. Purity alludes to the extent of the information directs having a place toward a referred to group that are allocated as individuals from a determined group by the method. The higher the extent of purity ([0, 1]) is, the more sure that the method has discovered the first group and the better the method is [29]. Entropy mirrors the quantity of the information focuses from various known groups in the first dataset that are doled out to a determined group by the method. The estimation of this measure is $[0, Log_2 N]$ where N is the quantity of realized groups included. The littler estimation of the entropy is, the less individuals from the realized groups are blended in the determined groups found by the method, and the better the grouping method is [30].. SSE is a regularly utilized group quality measure. It assesses the conservativeness of the subsequent persevering groups. Low scores of SSE demonstrates better persevering grouping results as the groups contain less inside varieties [29].

The productivity of a method was estimated by the measure of time in seconds taken for the method in finishing the grouping task. Therefore, we conducted all the experiments with a collection of synthetic datasets.

B. Dataset Used

All the experiments are done with a synthetic dataset. The same dataset is non-uniformly distributed over different nodes. In details, we created a synthetic dataset (DS) of 800000 data records for 2D. The DS include 4 groups; 3 concave shape groups and convex shape group. Groups in DS have various data records and different variations. The main idea behind selecting this dataset is that to test the effectiveness of suggested method of identifying convex and non-convex shape clusters. Fig. 5 illustrates the dataset. We agreed that there are some drawbacks of the selected dataset. It does not have many features that the real-life datasets have. In future work, this method will evaluated on more complicated datasets.

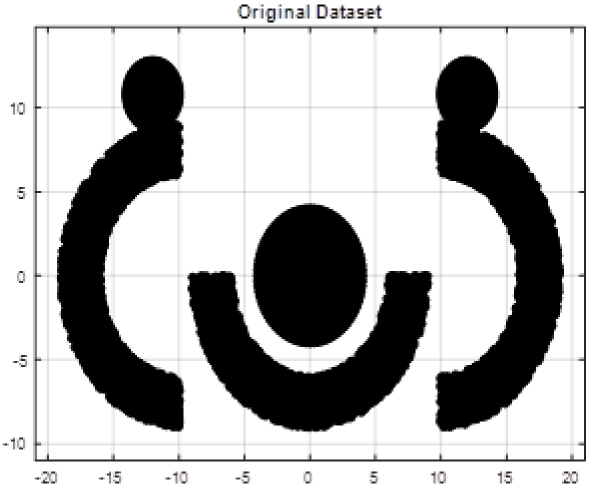

Fig. 5. Scatterplot of DS

C. Experimental Results

MATLAB 2017b was utilized to assemble an execution of the HPPD method and the examination structure. The suggested method does not deal with the first chunk of the dataset and later arrived chunks in an unexpected way, and consequently an unfilled arrangement of existing groups and a vacant arrangement of anomalies were accepted as input when the primary piece is handled. The thought behind the irregular choice of the information is to explore the conduct of the method when there is no control on the arrangement of information focuses, for example we didn't choose explicit information focuses from explicit groups in the ground truth dataset. No supposition that was agreed that the first information piece correspond to the whole information space. So as to limit the impact of the irregular selection of information focuses, the examinations were rehashed multiple times, and the average is determined.

Every one of the examinations were kept running on a machine furnished with 2.30 GHz 4 cores Intel(R) Core(TM) i5-4590 CPU and 16 GB memory. The working framework was Windows7. All the programs were developed using MATLAB.

- ***Purity***

Fig. 6 demonstrates the subtleties of examination results between the known groups and the yield groups from EINCKM, EDDS, and HPPD methods separately. HPPD has the most noteworthy purity. This is on the grounds that it keeps all the delegate information focuses and stringent merge technique. EINCKM likewise has a decent purity by pruning and reserving the surface-center data objects, however this procedure disregards non-convex shape groups. EDDS has a decent purity also and disregards a great deal of core data records which may not influence the last groups.

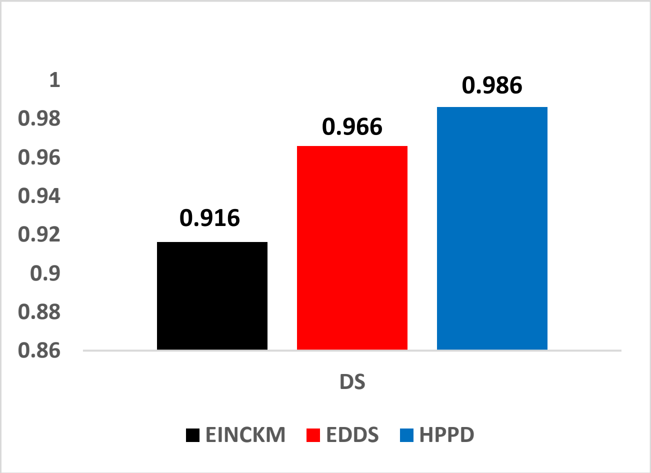

Fig. 6. The purity measurement.

- ***Entropy***

As appeared in Fig. 7, HPPD has the least entropy. EDDS has more elevated amount of entropy. EINCKM has the most abnormal amount of entropy among the three methods. The outcomes show that the pruning internal core data points affects group accuracy. Be that as it may, this outcome should be perused together with the purity estimation results to have a fair view on group accuracy.

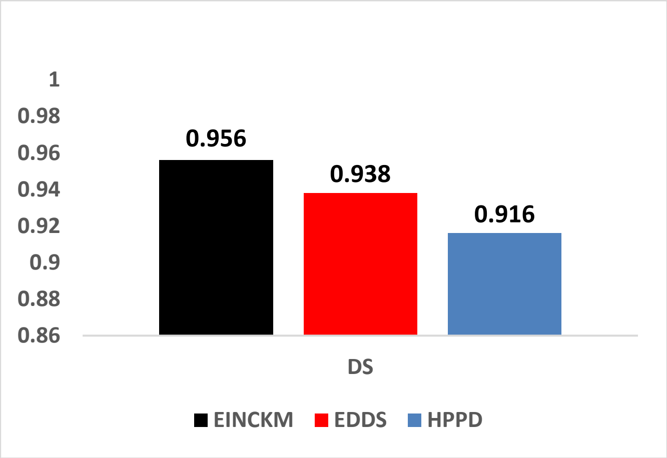

Fig. 7. The entropy measurement.

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

As appeared in Fig. 8, HPPD has the most minimal SSE, trailed by EDDS which thus is trailed by EINCKM, again demonstrating the expense of pruning internal core data points. On the other hand, EDDS and EINCKM still have the most exceedingly low SSE score, showing that blending incorrectly information focuses into found groups does likewise influence group quality. It should be mentioned that SSE may not be the perfect evaluator for nature of non-convex shape groups.

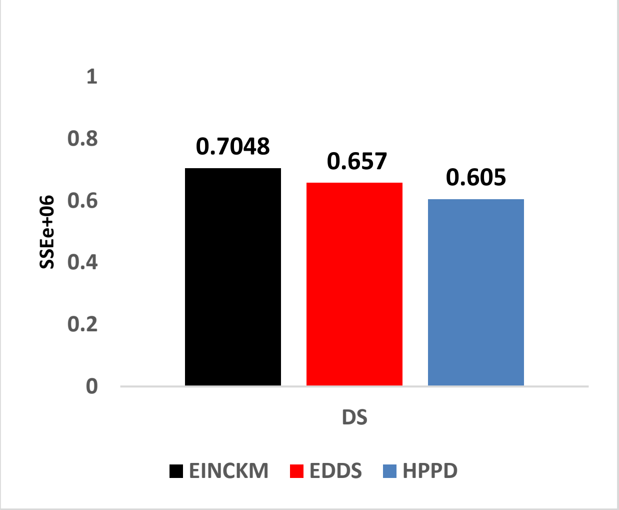

Fig. 8. The SSE measurement.

Efficiency Evaluation

- ***Execution Time***

Execution time is the proportion of the measure of time in seconds that proceed for the method in finishing the grouping task. Concerning implementation time, the EINCKM method has the base execution time pursued by EDDS, at that point HPPD (see Fig. 9).

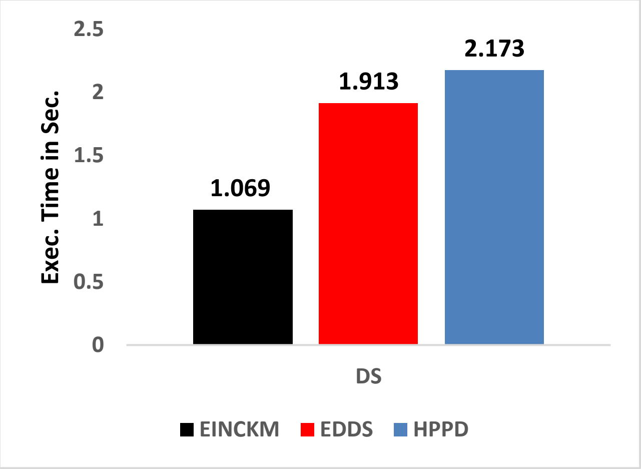

Fig. 9. The efficiency measurement

After getting these output results we could approve that EINCKM and EDDS methods are faster than HPPD.

D. Discussion

The major promise of the HPPD method is the easiness and high flexible structure. This consist of the main functions of the method: *pre-prcessing* to decide convex and non-convex clusters, modified EINCKM to perform convex clusters, modified EDDS to perform non-convex clusters, *adaptive merger* to prenesnt mixed output clusters. All process were allocated as functions which means we could simultaniosly enhance every function apart without modifying the overall main structure of the method.

There were propostions of more enhancements of every operation inside the method. Firstly, we know that utilising three separate functions (DBSCAN, EDDS function, and Qhull algorithm) cause slow process to deside convex and non-convex clusters. Therefore, we could replace those processes by one function, such as using computational topology to identify the cluster shape. Besides, the merge strategy we used in both modified algorithms is now very straight to reduce the magintude of execution time. In fact, the merging approche could be more enhanced via more advance solutions like a Bayesian theory. Lastly, machine learning techniques and fuzzy-based approches could be consider to produce more effecint merger function to produce mixed final groups.

6. Conclusion and Future Work

This paper presented a novel hybrid parallel framework by adopting our lastly produced data stream grouping methods, Modified EINCKM and Modified EDDS to present not only the outliers but also the convex, non-convex, and mixed clusters. The evidence shows that the convex, non-convex, and mix clusters are very useful on the application side of data streams. The hybrid algorithm HPPD emphasizes easiness, modularity, and flexibility.

The main thoughts of the presented method are to maintain the spherical and non-spherical groups concurrently divide and produce the last output groups as requested. The method stated one important issue of data stream grouping methods: Presenting convex and non-convex shape clusters separately at the same time and mixed clusters when there is request from the user in data stream. The evaluation of some synthesized datasets has explained that the method presents right and high accurate groups with less execution time.

Future work will concentrate on updating the method. Because of the method is flexible, those updating thoughts can deals with the major functions of the method. Firstly, we will investigate the topology computation to present more sophisticated version of the pre-processing step. Secondly, we will investigate hybridizing different clustering algorithm, like graph-based, hierarchal-based, and model-based methods to test the modularity of our framework. Thirdly, we will investigate utilizing learning strategy as feedback to improve the pre-processing step and improve clustering algorithms that have been used. Finally, distributed approaches may have encapsulated to produce more accurate, correct, and authentic new form of the HPPD method.

REFERENCES

[1] C. Liu, R. Ranjan, X. Zhang, C. Yang, D. Georgakopoulos, and J. Chen, "Public Auditing for Big Data Storage in Cloud Computing -- A Survey," *2013 IEEE 16th Int. Conf. Comput. Sci. Eng.*, pp. 1128–1135, Dec. 2013.

[2] E. Olshannikova, A. Ometov, and Y. Koucheryavy, "Towards Big Data Visualization for Augmented Reality," *2014 IEEE 16th Conf. Bus. Informatics*, pp. 33–37, Jul. 2014.

[3] M. Z. Islam, "A Cloud Based Platform for Big Data Science," *Dep. Comput. Inf. Sci. Linköping Univ. Master's Final Thesis*, pp. 1–57, 2013.

[4] N. Kaur and S. K. Sood, "Efficient Resource Management System Based on 4Vs of Big Data Streams," *J. Big Data Res.*, vol. 9, pp. 98–106, 2017.

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

[6] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering Data Streams," *IEEE FOCS Conf.*, pp. 359–366, 2000.

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

[8] 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.

[9] 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.

[10] H. Jiang, J. Li, S. Yi, X. Wang, and X. Hu, "A new hybrid method based on partitioning-based DBSCAN and ant clustering," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 9373–9381, 2011.

[11] R. Jain, "A Hybrid Clustering Algorithm for Data Mining," *UGC-SAP Scheme, DRS, Devi Ahilya Univ. Indore, India,* pp. 45–48, 2010.

[12] R. S. Sangam and H. Om, "Hybrid data labeling algorithm for clustering large mixed type data," *J. Intell. Inf. Syst.*, vol. 45, no. 2, pp. 273–293, 2015.

[13] P. Viswanath and R. Pinkesh, "l -DBSCAN : A Fast Hybrid Density Based Clustering Method," pp. 18–21, 2006.

[14] R. Dhiman, S. Vashisht, and K. Sharma, "A Cluster Analysis and Decision Tree Hybrid Approach in Data Mining to Describing Tax Audit," *Int. J. Comput. Technol.*, vol. 4, no. 1, pp. 114–119, 2013.

[15] M. F. Jiang, S. S. Tseng, and C. M. Su, "Two-phasee clustering process for outliers detection," *Pattern Recognit. Lett.*, vol. 22, no. 6–7, pp. 691–700, 2001.

[16] 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.

[17] 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.

[18] R. Ghaemi, N. Sulaiman, H. Ibrahim, and N. Mustapha, “A Survey : Clustering Ensembles Techniques,” *Eng. Technol.*, vol. 38, no. February, pp. 636–645, 2009.

[19] P. Hore, L. O. Hall, and D. B. Goldgof, “A scalable framework for cluster ensembles,” *Pattern Recognit.*, vol. 42, no. 5, pp. 676–688, 2009.

[20] P. Zhang, X. Zhu, J. Tan, and L. Guo, “Classifier and cluster ensembles for mining concept drifting data streams,” *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 1175–1180, 2010.

[21] R. Fathzadeh and V. Mokhtari, “An ensemble learning approach for data stream clustering,” *2013 21st Iran. Conf. Electr. Eng.*, pp. 1–6, 2013.

[22] H. Mutazinda, M. Sowjanya, O. Mrudula, and (M Tech, “Cluster Ensemble Approach for Clustering Mixed Data,” *Int. J. Comput. Tech. -–*, vol. 2, no. 5, pp. 43–51, 2015.

[23] S. Aghabozorgi, T. Ying Wah, T. Herawan, H. A. Jalab, M. A. Shaygan, and A. Jalali, “A hybrid algorithm for clustering of time series data based on affinity search technique.,” *ScientificWorldJournal.*, vol. 2014, p. 562194, 2014.

[24] A. Sree and A. M. Sowjanya, “A HYBRID CLUSTERING ALGORITHM FOR DATA STREAMS,” *IJAICT*, vol. 2, no. 5, pp. 3255–3269, 2015.

[25] J. Y. Chen and H. H. He, “A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data,” *Inf. Sci. (Ny).*, vol. 345, pp. 271–293, 2016.

[26] H. Kremer, P. Kranen, T. Jansen, T. Seidl, A. Bifet, G. Holmes, and B. Pfahringer, “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.

[27] 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.

[28] 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.

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

[30] 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.