Improve BIRCH algorithm for big data clustering

Fanny Ramadhani[1,*], Muhammad Zarlis[1], Saib Suwilo[2]

[1]Department of Information Technology, Universitas Sumatera Utara, Medan 20155, Indonesia
[2]Department of Mathematics, Universitas Sumatera Utara, Medan 20155, Indonesia

[*]fannyramadhani58@yahoo.com

Abstract. Big Data is a collection of data with super large data volumes, has a very high diversity of data sources, so needs to be managed with methods and devices that help perform accordingly. Clustering is one of the effective techniques for dealing with Big Data. The hierarchical method with the BIRCH algorithm is able to produce a short time in data execution. The BIRCH algorithm is a matching grouping algorithm for very large data sets. In an algorithm, a CF-tree is built in which all entries in each leaf node must meet same T threshold, and the CF-tree is rebuilt at each stage with a different threshold. But using a static (fixed) threshold produces poor cluster quality, in this paper proposes a solution to this deficiency by modifying the Threshold value to dynamic so that it can produce good cluster quality and be validated using silhouette coefficient (SC). There is a very clear difference between the standard BIRCH algorithm and the BIRCH algorithm on the modified T parameter (BIRCH (CF-Leaf (modif)). The CF-Node result, the total CF-Entries and Total CF-Leaf Entries produced 60% less than CF-Node, the total CF-Entries and Total CF-Leaf Entries in the standard BIRCH algorithm.

1. Introduction

Big Data is a term that describes a very large amount of data. Not only big in size, big data also has a large volume and variety of data as well as a gigantic data quantity, on both structured data and unstructured data. Big data is closely related to large data that must be analyzed in an executive way that is related to patterns, trends and associations especially with human habits. Big data is also very related to the high speed of data transfer to be able to store, process and analyze data [1].

The big data challenge is rooted in three important characteristics, they are: Volume, Velocity, Variety [2]. Big Data Clustering is a technique to group large data objects into several clusters. Clustering is one of the most effective ways to analyze problems found in Big Data. Recently several clustering techniques have attracted much attention because this technique is more flexible in scalability, reducing the amount of memory capacity that need to be used and offering faster response times to users. The problem that often arises in big data clustering is that the worst quality of cluster.

A research conducted by Pedregosa found that the CF-Tree performance on BIRCH was very fast, he was able to make 100,000 data point clusters to 1000 clusters in 4 seconds, at 2.9 GHz Intel Core i7, using scikit-learning. While the k-means implementation of the same data takes more than two minutes to complete the same task on the same architecture.

Furthermore, tree-BIRCH does not require the number of clusters as input, which is fully BIRCH only needed for the global clustering phase [3].

In [4], Praveen et.al mentions BIRCH found a significant difference between execution time in clustering the world document done by K-Means and BIRCH. The K-means for the first document with a size of 591 bytes requires 124 milliseconds while BIRCH requires 18 miliseconds. The second document that has 423 bytes of size requires 90 milliseconds while BIRCH requires 12 miliseconds. As well as the third document with 501 bytes of size, requires 106 milliseconds, while BIRCH requires only 15 milliseconds.

In [5], Lober introduced A-BIRCH: a variant parameter that is owned by BIRCH. Deciding the right and suitable parameters / attributes for grouping algorithms frequently found difficult because it requires information about unavailable data. This also applies to BIRCH, which requires the number of clusters k and also the threshold T to calculate the cluster correctly. BIRCH has a CF-tree that can make execution time shorter. But BIRCH produces poor quality clusters. For this reason, they removed the global phase of clustering, so that the rendering of the number of clusters was not needed, and proposed a method that automatically estimated the T threshold that achieved using Gap Statistics to determine the cluster property. The evaluation proved the application of the researcher's approach in a very powerful way to a two-dimensional gaussian isotropic distribution with approximately the same variance, regardless of the number of clusters or elements. By using Gap Statistic, it can produce clusters with good quality and shorter execution time.

In [6], Nidal mention BIRCH algorithm is a clustering algorithm suitable for very large data sets. In the algorithm, a CF-tree is built whose all entries in each leaf node must satisfy a uniform threshold T, and the CF-tree is rebuilt at each stage by different threshold. But using a single threshold cause many shortcomings in the birch algorithm, in their paper to propose a solution to this shortcoming by using multiple thresholds instead of a single threshold. Experimental results demonstrate that the medications appear to give good performance and overcome many of the shortcomings in basic birch algorithm. Nidal found that the cluster quality in the BIRCH algorithm can be improved by making changes to the threshold value. The threshold value that is worth default 0 [9] can be changed by creating a dynamic threshold so that it can improve the quality of the cluster.

Praveen and Pedregosa have proven that the clustering technique with the hierarchical method and using the BIRCH algorithm are able to reduce the execution time on Big data to be shorter. But Lober and Nidal found weaknesses in BIRCH. BIRCH produces poor cluster quality. So it is necessary to improve the quality of clusters to be better.

BIRCH can produce shorter execution times to solve big data problems, but the quality of clusters produced by BIRCH is very poor. So that it needs to be done in improving the quality of clusters. In the research we will do, this research will make the threshold value dynamic. Because according to Nidal [6], changing the threshold value to dynamic can produce better cluster quality. In this study, we tried to create a dynamic threshold by enlarging the scale on the leaf entry. so that it can accommodate similar data points. By changing the threshold value to dynamic, the threshold is able to follow the scala of the data point. Reduce the high level of the tree, because it does not split the parent if CF-leaf does not meet the requirements. So that it will still produce a short time and a good cluster.

2. Research background

2.1 Big data clustering

Clustering is the process of grouping data sets into several groups so that objects in one

group have many similarities and have many differences with objects in other groups. Clustering is one of the important methods to be able to know the similarity of the set of objects. Based on the similarity of these features, classes will be formed and get a pattern from a data set that is not labeled (unsupervised). Generally, Big Data grouping techniques can be classified into two categories [7]: Single-machine clustering and multiple machine clusters techniques that become very attractive and more interesting recently because it is faster and more adaptable to the new challenges of Big Data.

2.2. The hierarchy method

The hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified such as agglomerative or divisive, based on the way hierarchical decomposition is formed. The agglomerative approach has two approaches, namely bottom-up and top-down. The bottom-up approach takes place as follows, each object forms a separate group initially. Consistently combine objects or groups that are close to each other, until all groups are combined into one (the top level of the hierarchy), or until the condition of the termination of the relationship occurs. While the top down approach, begins with all objects in the same cluster divided into smaller groups, until each object in a cluster finally or until a condition of termination occurs. The hierarchical method contains the fact that after the merger or split step is carried out, the divisive process cannot be canceled. There are two approaches to improve the quality of hierarchical grouping: (1) carry out careful analysis of the "linkage" object on each partition hierarchy, as in Chameleon, or (2) integrates hierarchical agglomeration and other approaches by first using the agglomerative hierarchy of group object algorithms into micro clusters, and then macro clustering micro clusters using other grouping methods such as repeated relocation. One algorithm that belongs to the hierarchical method is BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies). BIRCH is an integrated hierarchy grouping algorithm. BIRCH introduces two concepts, clustering features and clustering feature trees (CF trees), which are used to describe cluster summaries [2].

3. Proposed method

3.1 BIRCH

BIRCH algorithm is an integrated hierarchical clustering algorithm. It uses the clustering features (CF) and cluster feature tree (CF Tree) two concepts for the general cluster description. Clustering feature tree outlines the clustering of useful information, and space is much smaller than the meta-data collection can be stored in memory, which can improve the algorithm in clustering large data sets on the speed and scalability. And is very suitable for handling discrete and continuous attribute data clustering problem

Objects in the dataset are arranged into a sub clustering CF form. This CF then clustered into k-groups using the traditional hierarchy clustering procedure. CF is triple of information that contains CF = (N, LS, SS), where N is the number of data points, LS is the result of adding the value of X (attribute value), and SS which is the result of adding the value of X squared. If there are 2 CFs merged, then the theoremare:

CF12 = (N1 + N2, $\overrightarrow{LS}_1$ + $\overrightarrow{LS}_2$, SS1 + SS2)

BIRCH incrementally calculates a summary of CF sub cluster. Clusters are represented by Vector CF and only Vector CF is stored in memory. This CF value is enough to calculate information about sub cluster such as centroid, radius and diameter and also an efficient storage method by summarizing information about sub cluster rather than saving all points

[8]. To find the location of a cluster feature that is suitable to be combined, we use the distance formula. we use the formula D2.

$$D2=\frac{\sqrt{(N_1 SS_2)+(N_2 SS_1)-2LS_1 LS_2}}{N_1 N_2}$$

To calculate the radius for CF-leaf, we use the formula:

$$R=\frac{\sqrt{SS-(LS)^2/n}}{n}$$

Shown in Figure 1, BIRCH phase includes the following four stages：

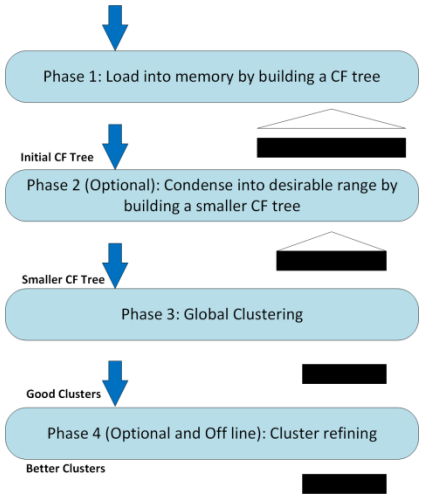

Figure 1.BIRCH Phase (Shirkhorshidi et al [2])

3.2 BIRCH standard algorithm

As for standard BIRCH algorithms are as follows [9]:

1 All data points are converted into CF form using the formula CF = (N, LS, SS).
2 When all the data has been changed in the form of CF, then CF-Tree starts working to bring together several formed CFs. In this section, you will be asked to enter the number B (Brancing).
3 Before scanning any data points from the database, we must initialize the initial CF tree threshold, this threshold will be used as the initial threshold value for each new CF entry that will not be changed during the grouping process. (static)
4 In a standard BIRCH, we will be asked to initialize L (number of leaf). For help the calculations, we add 2 parameters, namely m and b. parameter b is used to count the number of branches on CF-non leaf and m is used to count the number of leaf branches on CF-leaf.
5 For each record given, BIRCH compares the location of the record with the location of each CF at the root node, using a linear number or average CF. BIRCH continues the entry to the CF root node closest to the entry record.
6 Node then descends to the non-leaf child node of the CF nodes selected in step 5. BIRCH compares the location of records with the location of each non-leaf CF. BIRCH continues the note that goes to the non-leaf CF node closest to the entry.

7 Node then descends to the leaf child node of the non-leaf CF node selected in step 6. BIRCH compares the record location with the location of each leaf. BIRCH temporarily passes the entry to the closest leaf with the entry node.

8 Do one (a) or (b):

a If the leaf radius (R) selected includes a new node no does not exceed T Threshold, then the entry entered is assigned to that leaf. Leaves and all parent CFs are updated to take into account new data points.

b If the leaf radius selected including the new record exceeds the Threshold T, then a new leaf is formed, consisting of incoming notes only. CF parent updated to account for new data points.

9 If the leave (m) branch has exceeded the specified Leave (L) limit, there will be an additional branch (B).

10 If B has exceeded, there will be a split parrent process on CF and then it will be combined again with the new hight CF formed. CF parent updated to account for new data points

Shown in Figure 2, Flowchart of BIRCH Standard：

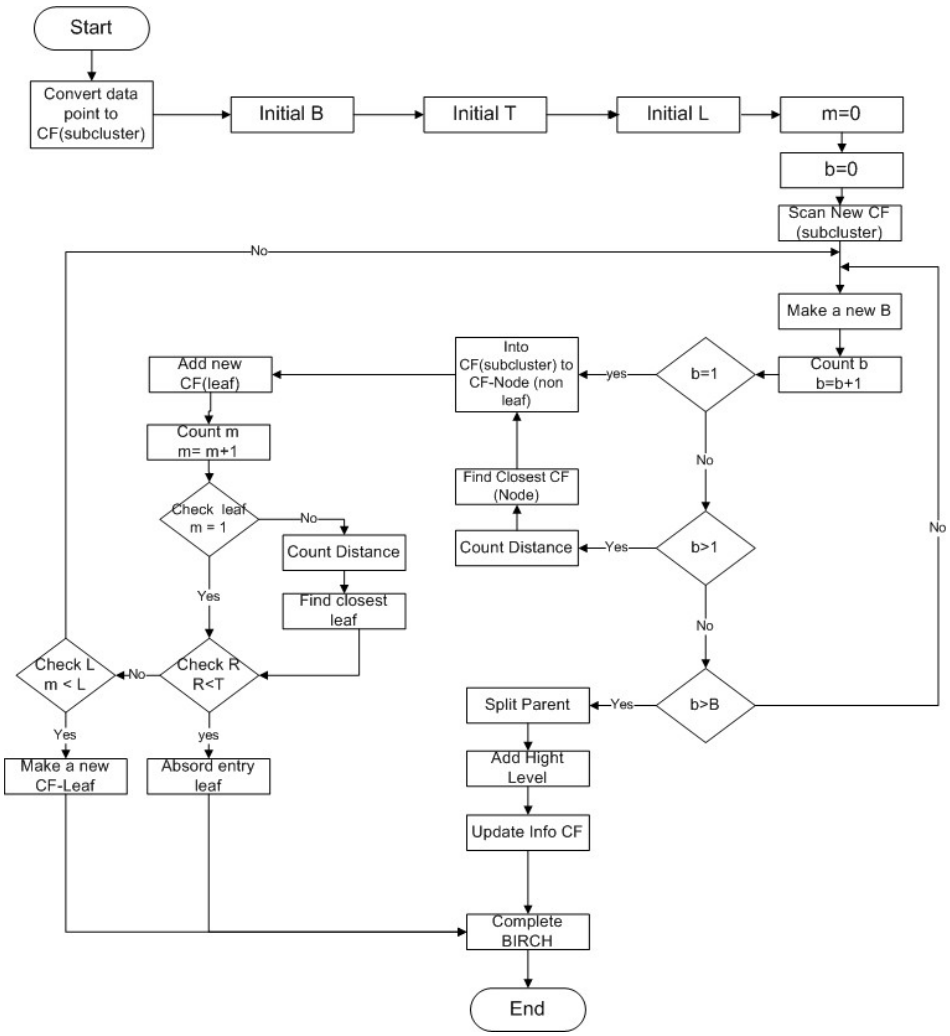

Figure 2. BIRCH Standard

3.3 Improved BIRCH algorithm

Based on previous research, some researchers mentioned that by modifying BIRCH at the Threshold value, it was able to improve cluster quality. In this study, the threshold value will be modified in CF-entry. While discussing this study, the author tries to improve the quality of the cluster by changing the threshold value to dynamic. In standard BIRCH, the CF formula used is CF = (N, LS, and SS). This study will use changes to the CF-Leaf value. The CF-leaf modif formula to be used is CF-Leaf (modif) = (N, LS, SS, T). Addition of the T parameter to CF-Leaf (modif) is used to store the latest changes from the threshold used. The T Addition parameter is only used for information about CF-Leaf while the CF-Node still uses the formula CF = (N, LS, and SS).

In the standard BIRCH when the data point has found the CF-node through the calculation of the closest distance. Then the data point will enter the CF-leaf if the radius on the leaf does not exceed the threshold (T). But if it exceeds the threshold value, a new leaf will be built and if it exceeds the leaf limit, there will be a split parent. Whereas in BIRCH- modif, the new data point that goes beyond the threshold will be modified at the threshold value. That is by enlarging the scale on the leaf radius so that it can reduce split parent in BIRCH.Shown in Figure 3, Flowchart of BIRCH (CF-Leaf (modif)):

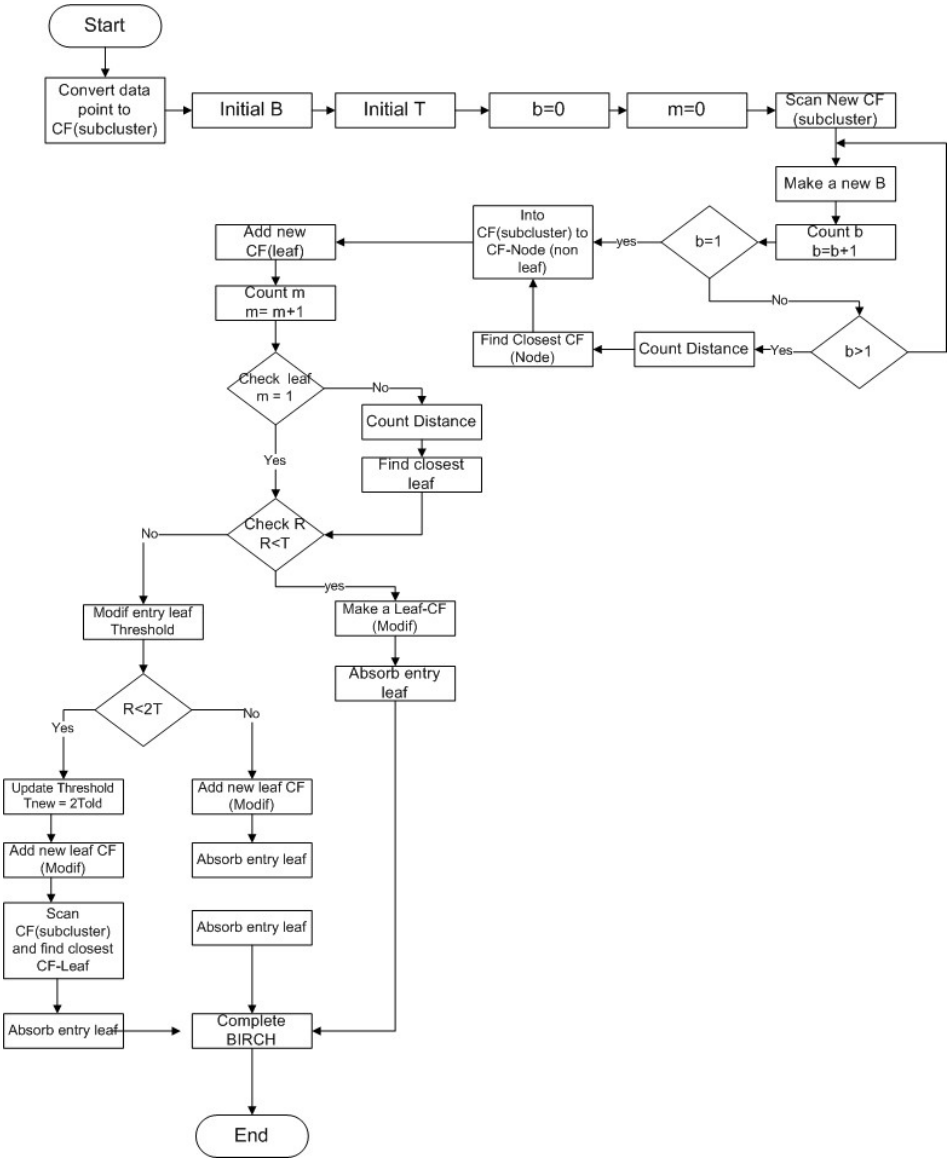

Figure 3. BIRCH (CF-Leaf (modif))

The algorithm from improve BIRCH is as follows:

1. All data points are converted into CF form using the formula CF = (N, LS, SS).
2. When all the data has been changed in the form of CF, then CF-Tree starts working to bring together several formed CFs. In this section, you will be asked to enter the number B (Brancing).
3. Before scanning any data points from the database, we must initialize the initial CF tree threshold; this threshold will be used as the initial threshold value for each new CF entry that will not be changed during the grouping process.
4. For Improve BIRCH we don't need L, because L will be add when the cluster need.
5. For help the calculations, we add 2 parameters, namely m and b. Parameter b is used to count the number of branches on CF-non leaf and m is used to count the number of leaf branches on CF-leaf.
6. For each CF sub cluster that enters, BIRCH will compare the location of each CF that has

been formed at the root node, using the euclidean distance. BIRCH continues the CF sub cluster into the CF root node closest to the CF sub cluster.

7. Sub cluster then descends to the non-leaf child node from the chosen CF node. BIRCH compares the location of the sub cluster with the location of each non-leaf CF. BIRCH continued the sub cluster that entered the non-leaf CF node which was closest to the incoming sub cluster.
8. Sub cluster then descends to the child leaf node from the non-leaf CF node chosen closest to it. BIRCH compares the location of the sub cluster with the sub cluster of each leaf. BIRCH temporarily passed Sub cluster which entered the leaf closest to the incoming CF sub cluster.
9. After finding the closest leaf, the sub cluster will check, if the radius of the leaf selected includes the new sub cluster does not exceed the Threshold T, then the sub cluster will enter the leaf (leaf-CF (modif)).
10. If the leaf radius selected including the CF sub cluster exceeds the Threshold T, the system will enlarge the cluster scale. Then check again. If the radius does not exceed the new threshold value, the change in the threshold value of T will be updated and the sub cluster will enter the leaf (leaf-CF (modif)).
11. But if the change in the threshold value of T keeps making the radius value exceed the T threshold, then leaf (leaf-CF (modif)) is formed, consisting of incoming sub cluster only. CF parent updated to root for new data points.
12. Then BIRCH will do phase 3, namely global clustering.

4. Result and analysis

In this section grouping steps will be carried out using the BIRCH algorithm with dynamic threshold values. In this study, we will adopt a bit of research [6]. In his research discussing how to change threshold values can improve cluster quality. To validate cluster quality, this study uses the silhouette coeffient (SC) method. SC value is in the range of -1 to 1. If, the SC value is close to 1, the better the grouping of data in one cluster. [10]

In this study using online retail datasets taken from UCI machine learning, and normalization of the data. The following are the results of testing the modified BIRCH algorithm with dynamic threshold values.

Table 1. Birch test results with dynamic threshold values 1.0

Algorithm	*Total CF-Node*	*Total CF-Entries*	*Total CF-Leaf-Entries*	*SC value*
BIRCH (Standard)	*11530*	*31380*	*109600*	*0.45*
BIRCH (CF-Leaf(modif))	*3000*	*11670*	*41760*	*0.97*

Table 2. Birch test results with dynamic threshold values 2.0

Algorithm	*Total CF-Node*	*Total CF-Entries*	*Total CF-Leaf-Entries*	*SC value*
BIRCH (Standard)	*11460*	*31310*	*109530*	*0.35*
BIRCH (CF-Leaf(modif))	*2300*	*11600*	*41690*	*0.93*

Table 3. Birch test results with dynamic threshold values 3.0

Algorithm	*Total CF-Node*	*Total CF-Entries*	*Total CF-Leaf-Entries*	*SC value*
BIRCH (Standard)	*11390*	*31240*	*109460*	*0.33*
BIRCH (CF-Leaf(modif))	*1600*	*11530*	*41620*	*0.91*

In Table 1, 2 and 3, we can see the effect of the threshold value on CF-Node results, total CF-Entries, Total CF-Leaf Entries and (SC) value. In the table it can be seen that, the greater the threshold value (T), the CF-Node value, the total CF-Entries, the Total CF-Leaf Entries and the resulting SC value also decrease.

From Table 1,2 and 3, it can be seen clearly the difference between the standard BIRCH algorithm and the BIRCH algorithm on the modified T parameter. The CF-Node result, the total CF-Entries and Total CF-Leaf Entries produced 65% less than CF-Node, the total CF-Entries and Total CF-Leaf Entries in the standard BIRCH algorithm. This research is in accordance with research conducted by [6]. The researcher stated that the change in Threshold (T) parameters to be dynamic can reduce the formation of CF-Tree in clustering, the execution time that is generated is also shorter and can make clusters more accurate. The difference is also seen in the SC value. The SC value is in the range of -1 to 1. The more the SC value approaches 1, the better the grouping of data in one cluster. Conversely, if SC approaches the value of -1, the worse was grouping of data in one cluster [10].

5. Conclusion

There is a very clear difference between the standard BIRCH algorithm and the BIRCH algorithm on the modified T parameter (BIRCH (CF-Leaf (modif)). The CF-Node result, the total CF-Entries and Total CF-Leaf Entries produced 60% less than CF-Node, the total CF-Entries and Total CF-Leaf Entries in the standard BIRCH algorithm. Modified BIRCH can make clusters more accurate and better can be validated with the SC method. So that it can be concluded that the BIRCH algorithm on the modified T parameters results in a much better cluster quality compared to the standard BIRCH algorithm.

References

[1] Mohanty H, Bhuyan P and Chenthati D 2015 *Big Data : A Primer* (New Delhi : Springer)

[2] Suganya R, Pavithra M and Nandhini P 2018 *International Journal of Engineering and Techniques* **4** 40

[3] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V,et all 2011 J. Mach. Learn. Res. **12**, 2825

[4] Praveen K, Sunita N and Chaudhari 2016 *International Conference for research in Applied Science & Engineering Technology (IJRASET).*

[5] Lorbeer B and Kosareva A 2017 *Advances in Big Data, Advances in Intelligent Systems and Computing*. Springer International Publishing, 160.

[6] Nidal I 2014 *International Journal of Artificial intelligence and Application for Smart Devices* **2**(1):1-10.

[7] Shirkhorshidi A S, Aghabozorgi S, Wah T Y and Herawan T 2016 *ICCSA.* Springer International Publishing 707
[8] Nithya P and Kalpara, A M 2017 *International Journal on Recent and Innovation Trends in Computing and Communication* **5** 1387
[9] Zhang T, Ramakrishnan, R and Livny, M 1996 *International Conference in Management of Data,Montreal, Canada* 103.
[10] L. Kaufman and P. J. Rousseuw 1990 *Finding Groups in Data* (New York: John Wiley & Sons).