Efficacy Boost Clustering on Big Data for Association Rule Mining

K. Mani
Associate Professor
P.G & Research Department of Computer Science
Nehru Memorial College, Puthanampatti-621 007.
Tiruchirappalli District, Tamilnadu, India.

R. Akila
Assistant Professor
P.G & Research Department of Computer Science
Nehru Memorial College, Puthanampatti-621 007.
Tiruchirappalli District, Tamilnadu, India.

***Abstract*— Extremely large data sets are analyzed computationally to reveal patterns, trends and associations, especially relating to human behavior and interactions. Since the databases are extremely large in their volume, large numbers of frequent itemsets are constructed for the generation of association rules and many more rules are generated. Parallelism with the distribution of tasks into clusters to different processors in a distributed environment with local and global minimum support thresholds may be helpful to minimize the number of frequent itemsets to be used for the construction of association rules. Thus, it is worthwhile to group the transactions into clusters and discover frequent itemsets using local and global minimum supports in order to minimize the number of frequent itemsets to be considered in turn to generate moderate number of strong association rules. Thus, this proposed paper focuses on clustering of the transactions among multiple parallel processors with local as well as global minimum supports are used to find frequent itemsets resulting in minimization of frequent itemsets and association rules. It speeds up the discovery of associations by minimizing the processing time and reduces the required memory space.**

Keywords*— Clusters; Association Rule Mining; Distribution; Parallelism, Priority Scheduling; Local and Global minimum supports

I. INTRODUCTION

The application of data mining techniques in any domain mainly employs algorithms such as Artificial Neural Network, Naive Bayes, Support Vector Machines, and other Machine Learning algorithms that are linked to data mining in classification, clustering, association rules mining, sequence pattern mining, and prediction tasks [24]. Association rules represent relationships among attributes of databases. They are descriptive models explores the properties of the data being examined [22]. It does not predict new values of the properties like predictive models. They are used to make important decisions [24] in decision support systems. Since transaction databases are voluminous, mining associations among itemsets requires more memory space and also is a time consuming process. Apriori is one of the popular rule mining algorithms. Apriori generates association rules by discovering frequent itemsets[16] [18] [23] following finding of candidate itemsets [21] to discover the relationships among itemsets. In order to find frequent itemsets, candidate itemsets are constructed [17] [20] from the itemsets of the database and database is scanned [19] for 1, 2, 3, …, n itemsets to generate the frequent itemsets. It leads to too many scans [18] [19] [20]. Another popular algorithm for association rule mining is FP-Growth which uses divide and conquer method [21]. It requires only two scans and no candidate itemset is generated. But it requires complicated pruning strategies. When the database is too large and the transactions have many items, voluminous numbers of frequent itemsets are to be taken into account for the generation of rules. Besides, many rules are generated which may or may not be relevant. It results in the requirement of more processing time as well as more memory space for the mining process. Thus it is necessary to minimize the processing time and memory space requirement for which distributed file system may be used. In order to distribute, transactions may be grouped into clusters and these clusters are distributed among multiple processors. Clustering is an unsupervised learning technique [24] to classify and group a set of data points [22]. The classification and grouping are made based on some constraints like mean, mediod, neighbourhood links [1] [2] [3] [6] and so on. Many clustering algorithms exist. They are
1) Partitioning Method: Each group contains at least one object. Each object must belong to exactly one group [6] [7]. CLARA, CLARANS and PAM are of this kind.
2) Hierarchical Method: Only one group is formed with all objects.
i) Agglomerative Approach: It is bottom-up approach [6]. CURE, BIRCH [13], CHAMELON [5] and ROCK [4] [8] are some agglomerative hierarchical methods [1] [2].
ii) Divisive Approach: It is top-down approach [9][10]. DIANA is one of the divisive approaches,
3) Density Based Method: This method is based on the notion of density. Some of these methods are DBSCAN, OPTICS and DENCLUE.
4) Grid Based Method: The major advantage of this method is fast processing time. It is dependent only on the number of cells in each dimension in the quantized space. STING and CLIQUE are of this kind.
5) Model Based Method: In this method, a model is hypothesized for each cluster to find the best fit of data. This method locates the clusters by clustering the density function. K-MEANS [11] [12] [14] [15] and EM are these methods.
6) Constraint Based Method: In this method, the clustering is performed by the incorporation of user or application-oriented constraints.

All of the existing clustering methods perform clustering using mean, median or links as distance measures. Those are immaterial for clustering transactions and these clustering methods are inconsistent while considering transactions consisting of itemsets. Since transactions consist of itemsets, it is fruitful to cluster the transactions based on the number of items. But, the items cannot be separated. Thus, the transactions may alternatively be clustered based on their timestamp i.e. occurrences in the database. While considering scheduling of these clusters, it is normally performed using

First-Come, First-Served Scheduling, Shortest Job Next Scheduling, Priority Scheduling, Shortest Remaining Time, Round Robin Scheduling and Multiple-Level Queues Scheduling. As transactions are strongly connected with itemsets and cannot be separated, it is prolific to apply priority scheduling on clusters based on the timestamp i.e. their occurrences in the database. It provides a systematic way to assign clusters to processors. As the generation of association rule is based on the itemsets occurring in the transactions as well as frequent itemsets, filtering of frequent itemsets plays a major role in the minimization of the number of irrelevant associations. Instead of using one level filtering with a minimum support threshold, two levels filtering with two minimum support thresholds may be used to double filter many infrequent itemsets. Thus, this paper focuses on partitioning the database by constructing clusters of transactions with one minimum support named as local minimum support at the lowest level for the clusters and another minimum support named global minimum support at the highest level for the database. Local and global minimum supports may determine the number of frequent itemsets to be used for the generation of association rules. The total numbers of transactions of the database as well as the number of available processors determine the number of transactions to be organized as clusters. The order in which the transactions appear in the database depends on the time when it happened. This order determines the priority to be provided to clusters. Thus, the transactions most recently take place considered as higher priority cluster and the transactions next recently take place considered as cluster 2 and so on. The formed clusters are distributed based on the priorities.

The methodology that is used in this proposed paper relies on load balancing strategy on transactions in a distributed environment. The number of transactions to be assigned to each cluster is calculated. The number of clusters to be constructed is decided on the basis of the number of processors to be used and also the number of transactions existing. The transactions are considered as they occur. The first appearing calculated numbers of transactions are grouped as first cluster and so on. Thus, the last appearing calculated numbers of transactions are grouped as last cluster. As priorities are to be given to clusters, the first cluster is designated as highest priority process, the second is designated as the next higher priority and the last cluster is designated as the lowest priority. If outlier occurs, the transactions of outlier are assigned to processes using round robin mechanism. Then, the clusters are distributed to processors based on priorities. Each processor involves in the task of discovering local frequent itemsets for which single itemsets are extracted from the transactions of clusters. Permutations are constructed from these extracted single itemsets. Frequent itemsets are found from all of the transactions of each cluster. Then, the frequent itemsets of all clusters are accumulated together and frequent itemsets for the accumulated itemsets i.e. for the database are found. In order to find frequent itemsets, instead of using only one minimum support, two minimum support thresholds local and global are used. Local minimum support is used on clusters and the global minimum support is used on the accumulated itemsets which are received from all the clusters. The association rules are then generated from these frequent itemsets using minimum confidence.

The rest of the paper is organized as follows. The related work based on Clustering and Association rule mining is presented in section 2. Section 3 focuses on the proposed methodology. Proposed work with an illustration is discussed in section 4. Results and discussions are presented in section 5. Finally, section 6 ends with conclusion.

II. RELATED WORK

Yogita Rani and Harish Rohil [1] described about clustering and its types namely hierarchical and partitioning methods. The authors extended their work by presenting an overview and a detailed discussion on some improved hierarchical clustering algorithms. In addition to this, they gave some criteria on the basis of which one can also determine the best among these mentioned algorithms. They suggested that the quality of hierarchical clustering methods can be improved by integrating hierarchical clustering with other techniques for multiple phase clustering. [2] Rahmat Widia Sembiring et al. have performed a comparative study on clustering technique Agglomerative Hierarchical using Euclidean single linkage and complete linkage and stated that different method will create a different number of clusters. Dr. Sankar Rajagopal [3] has performed the identification of high-profit, high-value and low-risk customers via the data mining technique-customer clustering has been studied using IBM Intelligent Miner and have used demographic clustering technique for customer clustering by identifying the high-value low-risk customers.

Sudipto Guha [4] studied clustering algorithms for data with boolean and categorical attributes. Regarding this, they proposed a concept of links to measure the similarity/proximity between a pair of data points and developed a robust hierarchical clustering algorithm ROCK that employs links and not distances when merging clusters. It exhibits good scalability properties. [5]George Karypis et al., discussed about various clustering algorithms namely, CURE, ROCK, K-Means CLARANS, DBScan, KNN and agglomerative. They compared Chameleon's performance against that of CURE and DBScan on four different data sets. These data sets had from 6,000 to 10,000 points in two dimensions; the points form geometric shapes and these data sets represent some difficult clustering instances because they contain clusters of arbitrary shape, proximity, orientation, and varying densities.

Yiling Yang et al. [6], have studied the problem of categorical data clustering, especially for transactional data characterized by high dimensionality and large volume. They developed an algorithm named CLOPE based on the intuitive idea of increasing the height-to-width ratio of the cluster histogram. The idea is generalized with a repulsion parameter that controls tightness of transactions in a cluster and thus the resulting number of clusters. The simple idea behind CLOPE makes it fast, scalable, and memory saving in clustering large, sparse transactional databases with high dimensions. In [7], Preeti Baser, Dr.Jatinderkumar and R. Saini described clustering techniques and several applications where clustering technique is used. The authors also described about various pros and cons of these techniques and performed a comparative analysis of various clustering techniques. Ashwina Tyagi and Sheetal Sharma [8] have described that the major problem is to identify heterogeneous subject areas where frequent queries are asked and addressed the problem with agglomerative clustering algorithms is that they make use of distance measures to calculate similarity. The authors target was to maximize the

criterion function so that the intra cluster similarity can be maximized and inter cluster similarity can be minimized. Thus they have implemented Robust Clustering using Links (ROCK) algorithm which uses Jaccard coefficient to find the similarity between the data or documents to classify the clusters. This technique actually reduces the searching time of documents from the database. Vera Marinova–Boncheva [9] has shown how a hierarchical clustering method can support an investor decision to choose stocks which can pretend to be participants in an investment portfolio by using a data mining tool and the identification of clusters of companies of a given stock market can be exploited in the portfolio optimization strategies. Then, they differentiated two of them like classification and clustering as supervised and unsupervised learning from data.

K.Sasirekha and P.Baby [10] have reviewed agglomerative clustering and have further stated that agglomerative hierarchical clustering starts with every single object (gene or sample) in a single cluster. Then, it agglomerates (merges) the closest pair of clusters in each successive iteration by satisfying some similarity criteria, until all of the data is in one cluster. The clusters have sub-clusters, which in turn have sub-clusters, etc. They concluded with that it can produce an ordering of the objects, which may be informative for data display. Smaller clusters are generated, which may be helpful for discovery and determine the similarity between prototypes and data points, and it performs well only in. Anil K. Jain [11] has provided a brief overview of clustering, summarized well known clustering methods, discussed the major challenges and key issues in designing clustering algorithms and point out some of the emerging and useful research directions, including semi-supervised clustering, ensemble clustering, simultaneous feature selection during data clustering, and large scale data clustering. Archana Singh et al. [12] have said that the power of k-means algorithm is due to its computational efficiency and the nature of ease at which it can be used and have implemented K-Means algorithm using three different metrics Euclidean, Manhattan and Minkowski distance metrics along with the discussion on the comparative study of results for two dimensional data. The results are displayed with the help of histograms. Tian Zhang et al. [13] evaluated BIRCH'S time/space efficiency, data input order sensitivity, and clustering quality through several experiments and also presented performance comparisons of BIRCH versus CLARANS, a clustering method proposed recently for large datasets, and showed that BIRCH is consistently superior.

Navjot Kaur [14] proposed a work representing ranking based method that improved K-means clustering algorithm performance and accuracy. An analysis has also been on K-means clustering algorithm by applying two methods, one is the existing K-means clustering approach which is incorporated with some threshold value and second one is ranking method applied on K-means algorithm and also compared the performance of both the methods by using graphs. The experimental results demonstrated that the proposed ranking based K-means algorithm produces better results than that of the existing k-means algorithm. Pranoti P. Jagtap [15] used K-means Algorithm and Apriori Algorithm for clustering and aggregation of the data which improves the effectiveness of the system. Further they stated that reducing clustering time enables the use of larger sampling percentages to improve clustering accuracy and gives the researcher greater flexibility when interactively exploring data archives and to solve the problem of data storage and efficiency the best solution is web server which provides with the web services to the employees and the owner. Krutika. K .Jain and Anjali, B. Raut [16] have made an attempt to use data mining as a tool used to find the hidden pattern of the frequently used item-sets and used Apriori Algorithm for finding these patterns from large databases so that various sectors can make better business decisions especially in the retail sector and it may find the tendency of a customer on the basis of frequently purchased item-sets. They changed the value of minimum confidence that gives different association rules and stated that if the value of minimum confidence is high, then rules filtered more accurately. They concluded that there are wide range of industries have deployed successful applications of data mining. Data mining in retail industry can be deployed for market campaigns to target profitable customers using reward based points. The retail industry will gain, sustain and will be more successful in this competitive market if adopted data mining technology for market campaigns. In [17], Priyanka Asthana et al. presented many improved Apriori algorithm to increase the efficiency of generating association rules and included hash-based technique, partitioning, sampling and using vertical data format. They stated that association mining rules are very useful in applications going beyond the standard market basket analysis and Hash-based technique can minimize the size of candidate itemsets. Further they mentioned hash based methods can be combined with Apriori algorithm to reduce time and space complexity. In [18], Mohammed Al Maolegi and Bassam Arkok have improved Apriori is proposed through reducing the time consumed in transactions scanning for candidate itemsets by reducing the number of transactions to be scanned. Whenever the k of k-itemset increases, the gap between the improved Apriori and the original Apriori increases from view of time consumed, and whenever the value of minimum support increases, the gap between the improved Apriori and the original Apriori decreases from view of time consumed. The time consumed to generate candidate support count in the improved Apriori is less than the time consumed in the original Apriori the improved Apriori reduces the time consuming by 67.38%. Sukhjit Kaur and Monica Goyal [19] have proposed a novel web data association rule mining based hybrid algorithm called HPSO-TS-ARM. They stated that these algorithms are based three well known high-level procedures: Particle Swarm Optimization, Tabu Search and Apriori Algorithm for Association Rule Mining. In their proposed work, PSO fetches the web search data in its optimized form, which is further computed by Tabu Search to prepare balance data arrangement followed by Association rule mining on processed web search data and concluded that the proposed algorithms have outperformed HPSO-TS and BSO-ARM on the basis of elapsed time and fitness function. In [20], Shruti Aggarwal and Ranveer Kaur discussed various classical algorithms like AIS, Apriori, Direct Hashing and Pruning and Partitioning algorithm. Then methods to improve the Apriori Algorithm are mentioned and improved approaches have been discussed also. A comparative study showed the benefits of different approaches and technique used by these algorithms/approaches.

Ramratan Ahirwal et al. [21] stated that Apriori is based on generating and testing and FP-growth is based on dividing and conquering. They do not cope with the new requirements of data mining. These previous approaches applied to generate frequent set generally adopt candidate generation and pruning techniques for the satisfaction of the desired objective.They

presented an algorithm which is useful in data mining task and knowledge discovery without candidate generation and this approach reduced the disk access time and directly found the frequent itemset by using support count table. It worked well with static dataset by using support count table as well as for mining streams requires fast, real-time processing in order to keep up with the high data arrival rate and mining results are expected to be available within short response time. In [22], Sayali Rajesh Suyal and Mohini Mukund Mohod explored the potential usefulness of data mining techniques in enhancing the quality of student performance. They have used a descriptive data mining technique called association rules mining to describe the student's current performance and a predictive technique called classification is used to predict student's future performance. They left the detection of the outliers in the educational database for more accurate predictions about the student's performance as future work. This study helped to identify those students which need special attention to reduce failure rate. They have also worked on investigation of similar patterns in student's withdrawal which also includes student's socio- economic status parallel with the academics from any course and help students as well as the institute in student retention along with the upgraded performance.

In [23], Divya Bansal et al. have used apriori Algorithm to discover and understand the underlying patterns involved in the court's records from their data contains in various sections. They stated that pathetic crimes against women are an alarming public issue not only in one or the other area but of worldwide issue. Hence, there is a need present for accurate, timely information to react to changing pathetic condition of women, identifying who are mostly involved i.e. age group of accused, stranger or known to the victim, and basically which age groups girls are the main target of victims are analyzed to improve the deteriorating condition of women. Their work answered all the questions as age group of men is 20- 24 ,age group of girls who are on their target is 16-22 and mostly accused are well known by the victim. This is helpful for the government, society and police that they will take certain actions towards the male society so that this appalling situation of women will improved and women can go freely anywhere. Their future work has been focused on to identify the states where crime rate is very much and what type of crime is faced by respective states such as murder, stealing, etc. Jamilu Awwalu et al.[24] showed that Multilayer Perceptron of Artificial Neural Network (ANN) takes longer to build and test a model compared to Decision Tree, Naive Bayesian, and the 10-Folds Cross Validation. However, in terms of accuracy, the Multilayer Perceptron seem be the best to cut across dataset percentage split and cross validation algorithms. Also, it was observed in this study that the smaller the number of the dimension of class of a dataset, the higher the accuracy of the model would be.

III. PROPOSED WORK

This section explains the tasks related with the proposed methodology. Clustering of transactions is required when distributed environment is used for accomplishing parallelism. The number of clusters is determined by the number of available processors. It is noted that all the existing clustering techniques only deal with closeness of data points, but not the number of items of transactions. The data points are different from transactions which consisting of itemsets. The methods viz., K-Means, K-Medoid, ROCK, CLARANS, DIANA and BIRCH determine the closeness of data points and they may not be used to determine the closeness of transactions. Thus, it is required to form clusters based on their occurrences. As a first step, the number of transactions to be organized into each cluster, say NOTC is calculated. Then, clusters are formed with transactions based on the number of processors say NOP. Clusters are formed for as many as processors. The first NOTC transactions of the database are grouped as first cluster. The second NOTC transactions of the database are grouped as second cluster and so on. Thus, the last NOTC transactions of the database are grouped as last cluster. Then, the first cluster is designated as highest priority process, the second is designated as the next higher priority and the last cluster is designated as the lowest priority. Assigning priority is to assign the clusters to processors and also to implement the same algorithm using multithreading in a single processor system to get the effect of distributed environment in a single processor. If outlier occurs, the transactions of outlier are assigned to clusters using round robin mechanism. These clusters are distributed to different processors. A scan on the transactions of clusters extracts only single itemsets. 2, 3, …, n itemsets are generated from these extracted single itemets. This kind of a way to generate itemsets reduces multiple scans to one scan and also eliminates the generation of candidate itemsets. Then, local Min_Support is used by each processor locally on the itemsets formed from the clusters of themselves. It leads to the generation of local frequent itemsets.

Normally, when partitions are used in a distributed environment, message passing technique is used. That makes network traffic. But, in the proposed methodology, no message passing technique takes place. Instead the same Min_Support is used by all processors. The processors are working autonomic without relying on another. As non overlapping clusters are used by the processors, messages need not be transferred from one processor to another processor. Thus, there is no need of collaboration among them. So, they can work at great speed. Then, the local frequent itemsets of all clusters are accumulated together to examine them against global Min_Support. It results in the generation of global frequent itemsets for the whole transaction database. Min_Confidence threshold is used to discover strong associations among frequent itemsets of all clusters. The steps involved in the proposed methodology are as follows and it is shown in Algorithm 1.

i. Calculate the number of transactions to be assigned to each cluster using (1)

$$NOTC = n/p \quad (1)$$

where n is the total number of transactions and p is the total number of processors.

ii. Consider the records of the database as they take place.

iii. Partition the transactions of database into p numbers of cluster say C_l, $1 \leq l \leq p$ where C_l is defined as

$$C1 = \{T1, \ldots, Tn/p\} \quad C2 = \{Tn/p + 1, \ldots, T2n/p\} \ldots$$
$$Cp = \{T(p-1)n/p + 1 \ldots Tpn\} \quad (2)$$

iv. Assign the priorities to the clusters. It is determined by the timestamp of the transactions. The transactions most recently occur occupies first pace and the transactions occur earliest takes lowest priority.

v. Assign outliers if any, to clusters based on round robin fashion from cluster 1 to cluster p-1.
vi. Distribute the clusters to processors on priorities.
vii. Extract only single itemsets presenting in each transaction of each cluster.
viii. Construct permutations from the single itemsets.
ix. Calculate support count for the permutations.
x. Examine against local Min_Support to find local frequent itemsets.
xi. Steps (vii) to (x) are performed by the processors separately on the clusters of themselves.
xii. Accumulate the itemsets and Calculate support count.
xiii. Examine against global Min_Support to find global frequent itemsets
xiv. Generate association rules from the frequent itemsets using Min_Confidence.

Algorithm1:EFFICACY_BOOST_CLUSTERING_ ASSOCIATION RULE MINING (*TDB, n,p*)

// TDB:Transaction DataBase; n : Number of Transactions,
//p: Number of Processors; C :Clusters ;
//LF :Local Frequent Itemsets; LFC:Support Count of LF
//GF: Global Frequent Itemsets GFC: Support Count of GF
EFFICACY_BOOST_CLUSTERING_ASSOCIATIONRULE MINING() // runs on master node
Input : TDB, n, p
Output : Clusters C_i and Strong Association rules *SAR*
Begin

```
NOTC=n/p;
i←1; f←NOTC; k←1;
for each Ti ∈ TDB do
while (i≤n)
 begin
      for j←i to f do
      Ck ={ null };
      begin
        Ck←{Ck} U Tj
      end
          k←k+1;i←i+NOTC;f←f+NOTC;
  end
end
  for i←1 to p   Ci ← i;
```

End
EFFICACY_BOOST_CLUSTERING()//Individual Process
Begin

```
  for each cluster Cj do
  NDi ← NDi + LFj ; NDCi ← NDCi + LFCj
  for each itemset i in NDi do
 begin
  If NDCi ≥ global Min_Support then
  GFi←Retain  itemset i in NDi & GFCi ← its count in NDCi
  else
  Eliminate itemset i from NDi & its count in NDCi
 end
 for each itemset in GFi do
 begin
      Generate antecedent and consequent
      Rule : antecedent ⟵ consequent
      CC=Min_Support(Rule) /Min_Support(antecedent)
      If  CC (Rule) <= Min_Conf then rejection
     else accepted as strong association rule:
         SAR←Rule
  end
End.
```

EFFICACY_BOOST_CLUSTERING()// runs on distributed nodes
Begin // cluster process

```
      for each Ii ∈ Tl do
     begin
       SISi← { Ii }     //   Add to single item set
       for k←2 to powerset(SISi) do
      begin
          is = SISi || SISi;
        if  is∈NDi then NDCi++;
       else { NDi ←is; NDCi ← 1;}
      end// for k
    end//for each Ii
for each itemset i in NDi do
begin
 if NDCi ≥ local Min_Support then
  LFi←Retain  itemset i in NDi &  LFCi←its Count in NDCi
 else
  Eliminate itemset i from NDi & its Count in NDCi
end
```

End // cluster process

The proposed methodology is shown in Fig. 1.

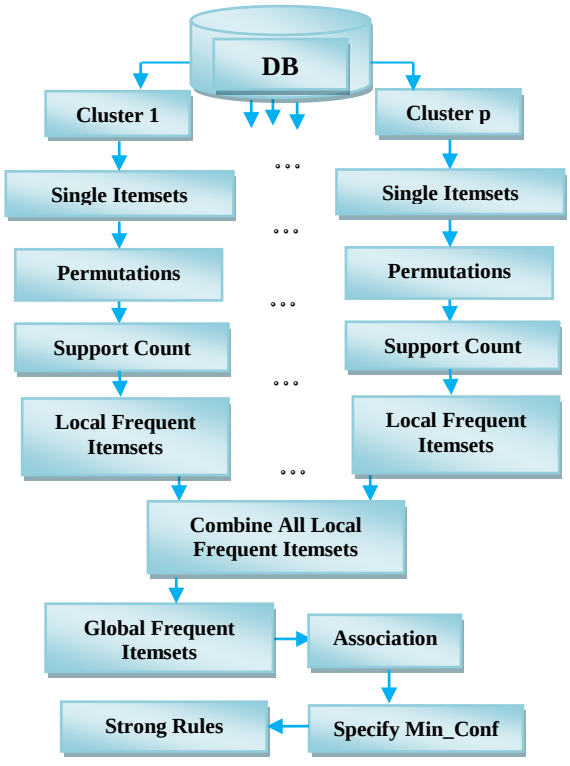

Fig. 1. Efficacy Boost Clustering for Big Data For Association Rule Mining

IV. PROPOSED WORK-AN ILLUSTRATION

To show the relevance of the proposed methodology, let there are p=5 processors and t=100000 transactions. Table1 shows the transactions with their itemsets.

TABLE 1. Transactional Table

Transactions	Itemsets	Transactions	Itemsets
T_1	I_1,I_2,I_3,I_4	T_6	I_1,I_2,I_3,I_4,I_5
T_2	I_2,I_4	T_7	I_2,I_3,I_5
T_3	I_1,I_4	…	…
T_4	I_3	…	….
T_5	I_1,I_2,I_4,I_5	T_{100000}	I_2,I_3,I_4,I_5

As NOP = p = 5, the sorted transactions are classified into five clusters. Since there are 5 processors and the number of transactions assigned to each cluster is 20000 which is calculated using (1). Then, the first 20000 transactions i.e. the transactions from T_1 to T_{20000} are assigned to first cluster C_1 which is designated as first process, the transactions from T_{20001} to T_{40000} are assigned to second cluster C_2 which is designated as second process, the transactions T_{40001} to T_{60000} are assigned to third cluster C_3 which is designated as third process, the transactions T_{60001} to T_{80000} are assigned to fourth cluster C_4 which is designated as fourth process, the transactions T_{80001} to T_{100000} are assigned to fifth cluster C_5 which is designated as fifth process. i.e. $C_1=\{T_1,\ldots,T_{20000}\}$ $C_2=\{T_{20001},\ldots,T_{40000}\}$ $C_3=\{T_{40001},\ldots,T_{60000}\}C_4=\{T60001,\ldots,T_{80000}\}$ $C_5=\{T_{80001},\ldots,T_{100000}\}$ are formed using (2).

These clusters are distributed to processors in a distributed environment to discover frequent itemsets. Single itemsets are extracted from these clusters. Single itemsets I_1,I_2,I_3,I_4 are extracted from the first cluster T_1 of C_1 and I_2,I_4 from T_2 of C_1. Similar process of extracting single itemsets is performed on the remaining transactions of cluster 1 and permutations of these itemsets for the cluster C_1 are constructed. The itemsets $\{I_1\}$ $\{I_2\}$ $\{I_3\}$ $\{I_4\}$ $\{I_1,I_2\}$ $\{I_1,I_3\}$ $\{I_1,I_4\}$ $\{I_2,I_3\}$ $\{I_2,I_4\}$ $\{I_3,I_4\}$ $\{I_1,I_2,I_3\}$ $\{I_1,I_2,I_4\}$ $\{I_1,I_3,I_4\}$ $\{I_2,I_3,I_4\}$ and $\{I_1,I_2,I_3,I_4\}$ are the permutations formed from T_1 of cluster 1. The occurrences of these itemsets are counted. After the support count of all transactions of cluster 1 is calculated, it is examined with the local Min_Support to discover local frequent itemsets. Then, the local frequent itemsets of all clusters are accumulated and examined with global Min_Support to discover global frequent itemsets. Associations among these frequent itemsets are constructed and examined with Min_Confidence to discover efficient strong association rules. Some of strong association rules satisfying Confidence ≥ 60% which are generated from frequent itemsets having Global Min_Support=3 are shown in Table 2.

TABLE 2 : Rules satisfying Global Min-Support and Min-Confidence

Rules	Confidence	Rules	Confidence
$I_1 \rightarrow I_4$	100%	$I_5 \rightarrow I_4$	80%
$I_4 \rightarrow I_1$	62.5%	$I_5 \rightarrow I_2I_3$	60%
$I_2 \rightarrow I_3$	66.7%	$I_2I_3 \rightarrow I_5$	75%
$I_3 \rightarrow I_2$	66.7%	$I_3I_5 \rightarrow I_2$	100%
$I_2 \rightarrow I_4$	83.3%	$I_2I_5 \rightarrow I_3$	75%
$I_5 \rightarrow I_2$	80%	$I_2 I_5 \rightarrow I_4$	75%

V. RESULTS AND DISCUSSIONS

It is observed that Efficacy Boost clustering on big data for association rule mining generates specific distributive patterns which speeds up the generation of frequent itemsets, in turn speeds up the generation of association rules, since the transactions are distributed and also the distribution is made on transactions. As the construction of frequent itemsets is based on the itemsets present in the transactions of clusters, it requires only one scan on the transaction database and also candidate itemsets are not generated.

It is also evident from Fig. 2, the proposed algorithm processes lesser number of transactions and also lesser number of itemsets than the same in a single processor environment. In a non distributed single processor environment, all transactions and all items of all transactions in the entire database are processed whereas in a distributed environment, since the transaction database is scattered among many processors, only the itemsets in the corresponding clusters which is normally very lesser in numbers are processed. It leads to fast response. Moreover, local Min_Support and global Min_Support are used where local Min_Support eliminates some of the infrequent itemsets at the lowest level. Double filtering using local and globlal thresholds are not only minimize the number of frequent itemsets, but also the processing time to construct association rules. Besides the space to retain the frequent itemsets is also minimized. Local Min_Support complies with the antimonotone property of Apriori algorithm that if a set cannot pass in a test, all of its supersets will fail the same test as well.

Apart, there is no network traffic as no collaboration among the processors is entitled. Message passing takes place only between the master and working processors, but not among parallel processors. All the parallel processors work without relying on others. After the distribution of transactions, the processors work independently. It is also observed that clustering of transactions and the distribution of them may efficiently utilize the processors.

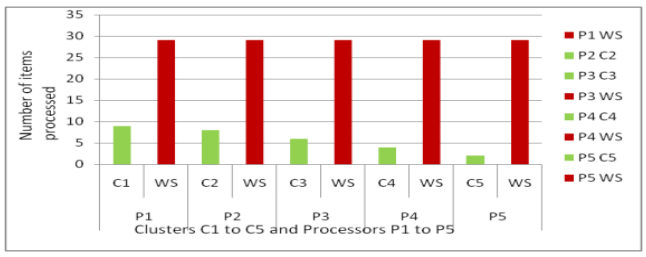

Fig. 1. Comparison of processing of the number of itemsets between Non-Distributed and Distributed Environment

VI. CONCLUSION

A novel Efficacy Boost Clustering on Big Data for Association rule Mining has been proposed in this paper and it generates association rules from voluminous number of transactions with lesser processing time and memory space. This proposed paper partitions the transactions into clusters as well as uses the local and global minimum supports. It works in two scenarios where one relies on clustering to accomplish parallelism to speed up the processing and another on local and global minimum supports resulting in the minimization of the number of frequent itemsets in turn minimize the space to

retain frequent itemsets as well as time to generate association rules. Efficacy Boost Clustering on Big Data for Association Rule Mining distributes the transactions into clusters which work in parallel among multiple processors with two support thresholds. The result clearly shows that the proposed Efficacy Boost Clustering on Big Data for Association Rule Mining outperforms well in terms of memory and speed than the classical apriori. The idea is unique and innovative.

REFERENCES

[1] Yogita Rani and Harish Rohil, "A Study of Hierarchical Clustering Algorithm", International Journal of Information and Computation Technology, Vol. 3, 2013.

[2] Rahmat Widia Sembiring, Jasni Mohamad Zain and Abdullah Embong, "A Comparative Agglomerative Hierarchical Clustering Method to Cluster Implemented Course", Journal Of Computing, Vol 2, Issue 12, December 2010.

[3] Dr. Sankar Rajagopal, "Customer Data Clustering Using Data Mining Technique", International Journal of Database Management Systems, Vol. 3, No.4, November 2011.

[4] Sudipto Guha, RajeevRastogi and Kyuseok Shim, "ROCK:A Robust Clustering Algorithm for categorical Attributes", this work is a part of .Serendip data mining project at Bell Lab IEEE 1999

[5] George Karypis, Eui-Hong (Sam) Han and Vipin Kumar University of Minnesota, "Chameleon: Hierarchical Clustering Using Dynamic Modeling", IEEE 1999.

[6] Yiling Yang, Xudong Guan and Jinyuan You, " CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data" SIGKDD '02, July 23-26, 2002, Edmonton, Alberta, Canada.

[7] Preeti Baser, Dr. Jatinderkumar and R. Saini, "A Comparative Analysis of Various Clustering Techniques used for Very Large Datasets", International Journal of Computer Science & Communication Networks, Vol 3(4).

[8] Ashwina Tyagi and Sheetal Sharma, "Implementation of ROCK Clustering Algorithm For The Optimization Of Query Searching Time", International Journal on Computer Science and Engineering, Vol. 4 No. 05 May 2012.

[9] Vera Marinova–Boncheva, "Using The Agglomerative Method Of Hierarchical Clustering As A Data Mining Tool In Capital Market", International Journal Information Theories & Applications" Vol.15, 2008.

[10] K.Sasirekha and P.Baby, "Agglomerative Hierarchical Clustering Algorithm- A Review", International Journal of Scientific and Research Publications, Vol. 3, Issue 3, March 2013.

[11] Anil K. Jain, "Data clustering: 50 years beyond K-means", International Conference on Pattern Recognition, Dec 2008, Elsevier B.V., Sep 2009.

[12] Archana Singh, Avantika Yadav and Ajay Rana" K-means with Three different Distance Metrics", International Journal of Computer Applications (0975 – 8887), Vol. 67, No.10, April 2013.

[13] Tian Zhang, Raghu Ramakrishnan and Miron Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases", This research has been supported by NSF Grant IRI-9057562and NASA (Grant 144-EC 78.SIGMOD '96 6/96 Montreal, Canada IQ 1996 ACM 0-89791 -794-4/96/0006.

[14] Navjot Kaur, Jaspreet Kaur Sahiwal, Navneet Kaur, "Efficient K-Means Clustering Algorithm Using Ranking Method In Data Mining", International Journal of Advanced Research in Computer Engineering & Technology Vol. 1, Issue 3, May2012.

[15] Pranoti P. Jagtap, Jyoti J. Danawale, Nidhi A. Chitambare, Sangita M. Jaiswal,"Conceptual Model of ERP with Web Server and Android Application Using K-means Clustering Based on Data Mining", International Journal of Advance Research in Computer Science and Management Studies, Vol. 2, Issue 5, May 2014.

[16] Krutika. K .Jain, Anjali . B. Raut, "Review paper on finding Association rule using Apriori Algorithm in Data mining for finding frequent pattern", International Journal of Engineering Research and General Science Vol. 3, Issue 1, January-February 2015.

[17] Priyanka Asthana, Anju Singu and Diwakar Singh , "A Survey on Association Rule Mining Using Apriori Based Algorithm and Hash Based Methods", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue 7, July 2013.

[18] Mohammed Al-Maolegi, Bassam Arkok, "An Improved Apriori Algorithm For Association Rules", International Journal on Natural Language Computing, Vol. 3, No.1, Feb 2014.

[19] Sukhjit Kaur and Monica Goyal, "Fast and robust Hybrid Particle Swarm Optimization Tabu Search Association Rule Mining (HPSO-ARM) algorithm for Web Data Association Rule Mining (WDARM)", International Journal of Advance Research in Computer Science and Management Studies, Vol. 2, Issue 9, September 2014.

[20] Shruti Aggarwal and Ranveer Kaur, "Comparative Study of Various Improved Versions of Apriori Algorithm", International Journal of Engineering Trends and Technology,Vol. 4, Issue 4, April 2013.

[21] Ramratan Ahirwal, Neelesh Kumar Kori and Dr.Y.K. Jain. "Improved Data Mining Approach To Find Frequent Itemset Using Support Count Table", International Journal of Emerging Trends & Technology in Vol. 1, Issue 2, July – August 2012.

[22] Sayali Rajesh Suyal and Mohini Mukund Mohod, " Quality Improvisation Of Student Performance Using Data Mining Techniques", International Journal of Scientific and Research Publications, Vol. 4, Issue 4, April 2014.

[23] Divya Bansal and Lekha Bhambhu, "Execution of APRIORI Algorithm of Data Mining Directed Towards Tumultuous Crimes Concerning Women", International Journal of Advanced Research in Computer Science and Software Engineering 3(9), September – 2013.

[24] Jamilu Awwalu, Anahita Ghazvini, and Azuraliza Abu Bakar, "Performance Comparison of Data Mining Algorithms: A Case Study on Car Evaluation Dataset", International Journal of Computer Trends and Technology (IJCTT) , Vol. 13, No. 2, Jul 2014.