ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Frequent Itemset Mining and Association Rule Generation using Enhanced Apriori and Enhanced Eclat Algorithms

S.Sharmila[1], Dr. S.Vijayarani[2]

Ph.D. Research Scholar, Dept. of Computer Science, School of Computer Science and Engineering, Bharathiar University, Coimbatore, India[1]

Assistant Professor, Dept. of Computer Science, School of Computer Science and Engineering, Bharathiar University, Coimbatore, India[2]

ABSTRACT: The main objective of this research work is to find the frequent items and association rule generation by using the Enhanced-Apriori and Enhanced-Eclat algorithms. In data mining, normally association rule generation process consists of two steps; first step is finding the frequent items based on the minimum support threshold which is assigned commonly to all the items and the second step is the association rule generation. This research work also used the same steps with small modification i.e. in the first step, instead of assigning common minimum support threshold, this work has assigned an individual minimum support threshold to each and every item in the database, from this frequent items are found and association rules are generated. Performance factors used are execution time, memory space, number of frequent items and number of rules generated. Different sizes of datasets and threshold are used for experimentation. From the results, we observed that the Enhanced-Apriorialgorithm has produced good results than Enhanced- Eclat algorithm.

KEYWORDS: Frequent item mining, Association Rule Mining, Enhanced Apriori, Enhanced Eclat.

I. Introduction

Association rule mining is one of the most important techniques in of data mining, It was first introduced by Agrawal et al. 1993 [1]. Association rules are used for examining the relationship between objects/ items in the databases [2].Popular association rule mining algorithms are Apriori, partition, pincer-search, dynamic item set counting, FP-tree growth, H-Mine, FIN and Relim etc. [3]. An association rule has two factors, an antecedent (if) and a consequent (then). An antecedent is an item found in the data [4].A consequent is an item that is found in sequence of antecedent.

Association rules are useful for marketing, commodity management and advertising etc. It is predetermined to identify strong rules discovered in databases using two important measures i.e. support and confidence [5].The support of the rule X -> Y is the percentage of transactions in T that contain $X \cup Y$. It determines how frequent the rule is applicable to the transaction set T. The confidence of a rule describes the percentage of transactions containing X which also contain Y. The main objective of this research work is to find the frequent items using individual item threshold and to generate the association rules. Two popular existing association rule algorithms are enhanced they are termed as Enhanced-Apriori and Enhanced-Eclat.

Remaining section of the paper is organized as follows. Section 2 describes the related works. Section 3 illustrates the methodology and discusses the enhanced algorithms. Section 4 gives related work results and finally conclusion is given in Section 5.

DOI: 10.15680/IJIRCCE.2017. 0504031

II. Related work

In [1] presented a theoretical survey on existing algorithms. The concepts behind association rules were discussed and followed by an overview to some of the previous research works done on this area. The advantages and limitations were also discussed. Author had analysed that adequate attention was not given to the quality of the rule generated. The enhanced algorithm had reduced the execution time, complexity and improved the accuracy.

In [2] author has analyzed the performances of Apriori algorithm in minimizing candidate generation. This research scenario has introduced a new way in which the apriori algorithm can be improved. The modified algorithm introduced factors such as set size and set size frequency which in turn are being used to eliminate non-significant candidate keys. With the use of these factors, the modified algorithm introduced more efficient and effective way of minimizing candidate keys.

In [3] discussed different algorithms for association rule mining on different size of database First he had improved Apriori algorithm which took less time for frequent item set generation. Second he had focused on Feature Based Association Rule Mining algorithms. Finally author had focused in Optimized Distributed Association Rule Mining Algorithm in distributed database. The classical Apriori algorithm had some disadvantages therefore, in this paper, authors have studied different algorithms from which they concluded that Feature Based Association Rule Mining Algorithm was best and efficient than other algorithms

In [7] author discussed about mining frequent itemset in transactional database. The main objective of this comparative analysis is to reduce the number of scans and improve the efficiency. The strength and weakness of Apriori, DHP, Partitioning, Sampling, DIC, H-mine, FP-growth, and Eclat algorithms were analyzed. Finally it had observed that FP-growth algorithm works better than all other algorithms.

In [8] described an Improved Apriori Algorithm based on Matrix Data Structure for mining regular item set, the proposed algorithm used bit matrix, it needs only single scan for whole transactional database. It has constructed compressed data structure which has reduced I/O cost and does not generated the irregular item set. Improved algorithm decreased the temporal complexity and spatial complexity and also it have higher efficiency as compared to classical apriori algorithm.

In [9] author had proposed an Improved Apriori Algorithm for Mining Association rules. The proposed algorithm has decreased pruning operations of candidate 2-itemsets, thereby it consumed time and increased efficiency. It optimized the subset operation, through the transaction tag to speed up support calculations.

In [10] reviewed about different frequent mining algorithms like Apriori, FPgrowth and DIC. A brief description of each technique has been provided. Different frequent pattern mining techniques are compared based on various parameters and real time dataset. From this work it is observed that FP-growth algorithm gives better results than other algorithms.

III. METHODOLOGY

Figure 1 shows the architecture of proposed work. . The dataset is generated synthetically, it consists of three different sizes, i.e. 5K, 10K and 15K transactions .Average length in the transaction of a dataset are 15 and 20 items

DOI: 10.15680/IJIRCCE.2017. 0504031

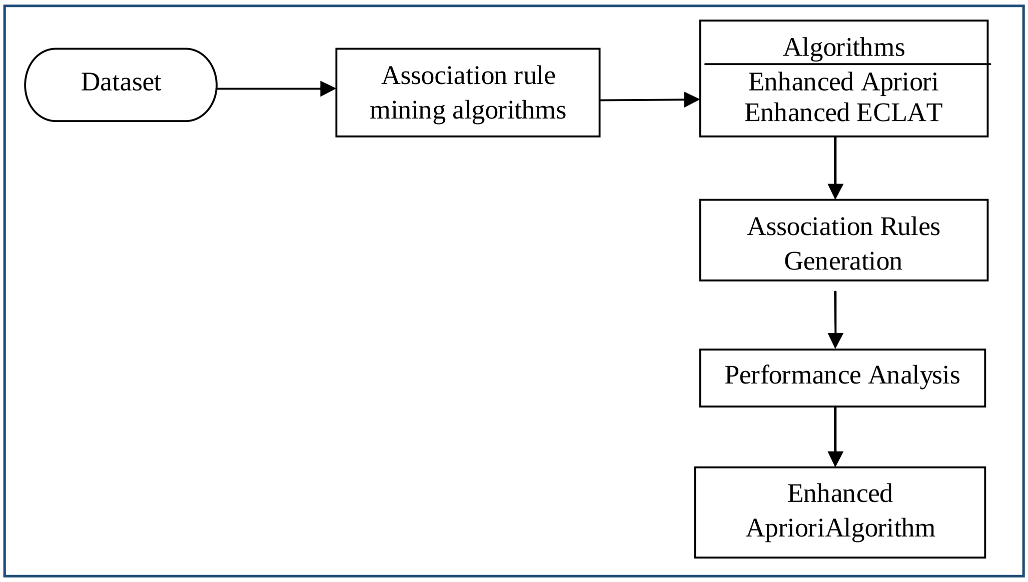
Dataset
Association rule mining algorithms
Algorithms
Enhanced Apriori
Enhanced ECLAT
Association Rules Generation
Performance Analysis
Enhanced AprioriAlgorithm

Figure. 1 System Architecture

A. *Enhanced Apriori :*

Apriori Algorithm was first introduced by R. Agrawal. This algorithm is used to discover frequent item set. The main principle of Apriori algorithm is, it is level-wise search, mining frequent itemsets from transactional database7. In this algorithm, frequent subsets are extended one item at a time and this step is known as candidate generation process 8. Then groups of candidates are tested with the data. Apriori uses breadth-first search method (level-wise search) and a hash tree structure to count candidate item sets efficiently in the search space. There are several key concepts used in Apriori algorithm such as Frequent Itemsets, Apriori Property and Join Operation 11. Table 1 explains the pseudo code of Apriori algorithm [19].

IV. PSEUDO CODE

PSEUDO CODE FOR APRIORI ALGORITHM

Pseudo code for Apriori algorithm

```
Initialize: K: = 1, C1 = all the 1- item sets;
Read the database to count the support of C1 to determine L1.
 L1:= {frequent 1- item sets};
K:=2; //k represents the pass number//
While (Lk-1 ≠ ∅) do
 Begin
 Ck := gen_candidate_itemsets with the given Lk-1
 prune (Ck)
 For all transactions t ∈ T do
 Increment the count of all candidates in CK that are contained in t;
Lk: = All candidates in Ck with minimum support;
k := k + 1;
End
Answer: = ck Lk;
```

In the Enhanced Apriori algorithm, an individual threshold value is given to each item in a transactional database. In the first candidate generation, all the items are compared with the respective threshold value and number of occurrence. If the occurrence is equal to or greater than the threshold value, the items are selected for the candidate generation. During the candidate generation for e.g. finding the frequent two-itemset, the threshold value for these two items are compared and then the minimum threshold value is chosen, then the algorithm finds the number of occurrences of these two-items with the chosen minimum threshold value. If the number of occurrences is greater than minimum threshold then these items are selected for next iteration. This process is continued for all the items as well as all the candidate generation. The pseudo code of Enhanced Apriori algorithm is given below

PSEUDO CODE FOR ENHANCED APRIORI ALGORITHM

```
Pseudo code for Enhanced Apriori algorithm
Input:
1. Set of items and transactions
2. Threshold values for each item
3. Support and confidence threshold values for strong association rules
Output:
1. Frequent items
2.  Association Rules
Algorithm
Step: 1 consider a database D, Which Tn transactions
        D = {$T_1$, $T_2$, $T_3$, $T_4$.........$T_n$} // D contains set of all transactions
        $T_1$= {$I_1$ $I_2$ $I_3$ $I_4$................ $I_n$} where each T1 has set of items
        $ITI_J$ = {$ITI_1$, $ITI_2$, $ITI_3$........ $ITI_n$}where j=1,2....l each Ij has individual threshold
        //Assign individual threshold values to all items in the database
Step; 2 frequent item generation
        If $I_1$> = assigned min_support then
        For (m=1 to l)
        {
                    Count the occurrence of $I_m$
        }
        Check if the count of $I_m$> $ITI_m$ then
        $L_k$ = $I_m$ // $L_k$ frequent itemset
        Repeat the same for all the items
        End for
        End if
Candidate generation
        $L_1$:= {frequent 1- item sets};
        K: =2; //k represents the pass number//
        While ($L_k$-1 $\neq\emptyset$) do
        Begin
                    $C_k$: = gen_candidate_itemsets with the given $L_{k-1}$
                    Min (IT (Ij)), (IT ($I_k$)).......... (IT (In))
                    //where j,k…n are the items in the transaction
                    // select the minimum threshold item from the combination of items
                     // repeat the same for all the items in the transaction
                    Prune ($C_k$)
                    $L_k$: = All candidates in $C_k$ with Individual minimum support;
                    For all transactions t $\in$ T do
                    //Increment the count of all candidates in $C_K$ that are contained in t;
                    k := k + 1;
        End
        Return $C_k$ $L_k$;
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

B. Enhanced Eclat

Eclat algorithm primarily depends on depth-first search algorithm which uses the set intersection method to find the frequent itemset from the database [12]. It uses a vertical database layout i.e. alternative of explicitly listing all transactions; each item is stored together and uses the intersection based approach to compute the support of an itemset. Hence, the support of an itemset can be easily computed by intersecting any two subsets i.e. ($Z \subseteq X$, such that YU $Z = X$.) [13]. When the database is stored in the vertical layout [15], the support of a set can be counted much easier by simply intersecting subsets 18. In this algorithm each frequent items are added in the output set and new database is created. This is done by first finding every item that frequently occurs together with in itemset [19]. This algorithm is also called recursive algorithm. Table 3 explains the pseudo code of Eclat algorithm [20].

PSEUDO CODE FOR ECLAT ALGORITHM

```
Pseudo code for Eclat algorithm
Input: $E((i_1, t_1), \ldots (i_n, t_n))|P), s_{min}$
Output: $F(E, s_{min})$
for all $i_j$ occuring in E do
    $P := P \cup i_j$ // add $i_j$to create a new prefix
    init(E′) // initialize a new equivalence class with the new prefix P
    for all $i_k$ occuring in E such that $k > j$ do
        $t_{tmp} = t_j \cap t_k$
        if $|t_{tmp}| \geq s_{min}$ then
            $E' := E \cup (i_k, t_{tmp})$
            $F = F \cup (i_k \cup P)$
        end if,
    end for
    if $E' \neq \{\}$ then, Eclat$(E', s_{min})$
    end if, end for
```

In the Enhanced Eclat algorithm, an individual threshold value is given to each item in a transactional database. In the first candidate generation, all the items are compared with the respective threshold value and number of occurrence. If the occurrence is equal to or greater than the threshold value, the items are selected for the candidate generation. During the candidate generation for e.g. finding the frequent two-itemset, the threshold value for these two items are compared and then the minimum threshold value is chosen. Then find the number of occurrences of these two-items and is compared with the chosen minimum threshold value. If the number of occurrences is greater than minimum threshold then these items are selected for next iteration. This process is continued for all the items as well as all the candidate generation. The pseudo of Eclat is explained. The pseudo code of Enhanced Eclat algorithm is given below.

DOI: 10.15680/IJIRCCE.2017. 0504031

PSEUDO CODE FOR ENHANCED ECLAT ALGORITHM

```
    Pseudo code for Enhanced Eclat algorithm
Input:
    1. Set of items and transactions
    2. Threshold values for each item
    3. Support and confidence threshold values for strong association rules
Output:
    1. Frequent items
    2.  Association Rules
    D = {T_1, T_2, T_3, T_4.........T_n}
            // D contains set of all transactions
    T_k = {I_1 I_2 I_3 I_4................ I_n}
            //Tk contains set of all items
    T_1 = {IT (i_1), IT (i_2), IT (i_3) IT (i_4) ..... IT (i_n)}
           //Assign individual threshold values to all items in the database
1:  for all $i_j$ occuring in D do
2:  $P := P \cup i_j$ // add $i_j$ to create a new prefix
3:   init(D′) // initialize a new equivalence class with the new prefix P
4:   for all $i_k$ occuring in D such that $k > j$ do
5:   $T_{tmp} = t_j \cap t_k$
6:   if $|T_{tmp}| \geq s_{individual\ support}$ then
7:   $D' := D \cup (i_k, T_{tmp})$
8:   $F = F \cup (i_k \cup P)$// Candidate generation
9.  Candidate Generation ($i_J$, $i_{J+1}$…$i_n$ )
10:  $s_{individual\ support}$ =Min (IT (Ij)), (IT (Ik))……… (IT (In))
           //where j,k…n are the items in the transaction
           // select the minimum threshold item from the combination of
           items
           // repeat the same for all the items in the transaction
11: Go To Step 5, // repeat the same for all the items in the transaction
12:  end if, 13:  end for
14   if $D' \neq \{\}$ then, : Eclat($D', s_{individual\ support}$)
15end if, end for
```

V. Results And Discussion

Three different synthetic datasets are generated for experimentation whose sizes are 5K, 10K and 15K of transactions. Average length of the items of these transactions is 15 and 20. Performance factors used for comparison are number of frequent items, number of rules, strong rules, execution time and memory usage. This work is done in Intel Core i5 processor running at 3.30 GHz, 4 GB RAM and 32 bit Windows 8.Table 1 gives the number of frequent items generated by the two algorithms for different size of data sets.

Table.1 Number of Frequent Items

DATASET SIZE	15 Items		20 Items	
	ENHANCED APRIORI	ENHANCED ECLAT	ENHANCED APRIORI	ENHANCED ECLAT
5000	1940	1908	2195	1991
10000	5441	4933	7327	7299
15000	9136	8131	11345	11207

Figure 2 depicts number of frequent items generated by the two algorithms. It is observed that Enhanced-Apriori algorithm gives best results than Enhanced-Eclat algorithm

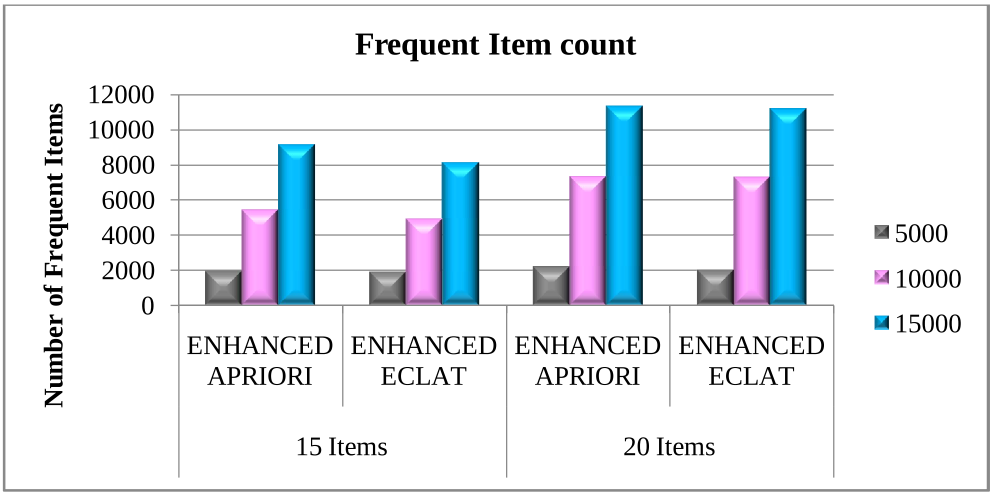

Fig.1 Number of frequent items

Tables 2 shows the generation of association rules using Enhanced-Apriori and Enhanced Eclat for different size of datasets.

Table.2Number of Rules Generation

DATASET SIZE	15 Items		20 Items	
	ENHANCED APRIORI	ENHANCED ECLAT	ENHANCED APRIORI	ENHANCED ECLAT
5000	2573	2180	3285	2138
10000	8049	7012	9430	9221
15000	10763	10581	14450	14079

Figure 2 describes the analysis of number of rules. It is observed that Enhanced-Apriori algorithm gives best results than Enhanced-Eclat algorithm.

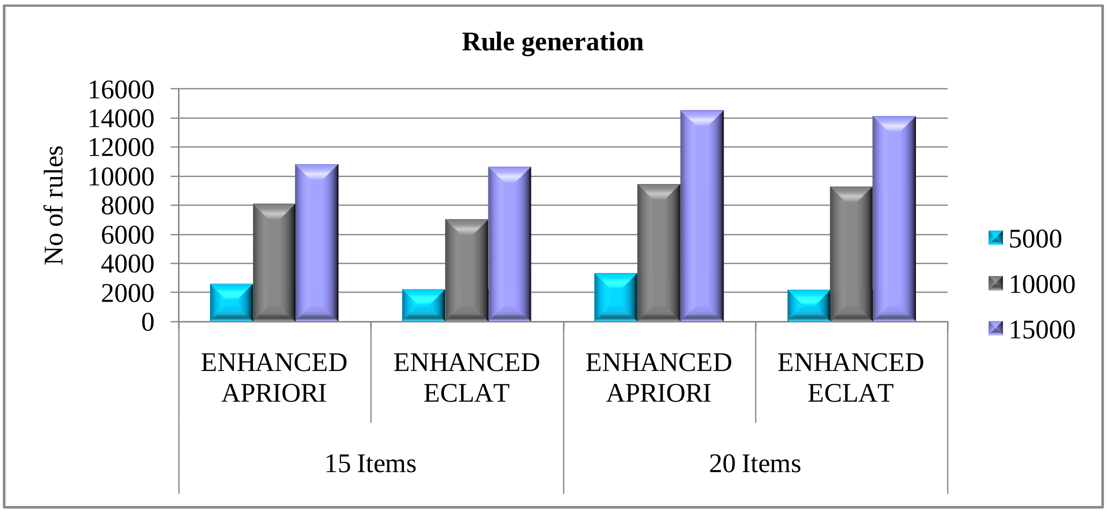

Fig.2 Analysis ofRules Generation

Tables 3 gives Generation of strong Rule using Enhanced-Apriori and Enhanced Eclat for different size of datasets with support =30 and confidence =70.

Table.3Strong Rules

DATASET SIZE	15 Items		20 Items	
	ENHANCED APRIORI	ENHANCED ECLAT	ENHANCED APRIORI	ENHANCED ECLAT
5000	1983	1813	2118	1945
10000	5936	5810	6865	5990
15000	9845	9798	12567	12498

Figure 3 illustrates the analysis of strong rules. It is observed that Enhanced-Apriori algorithm gives best results than Enhanced-Eclat algorithm.

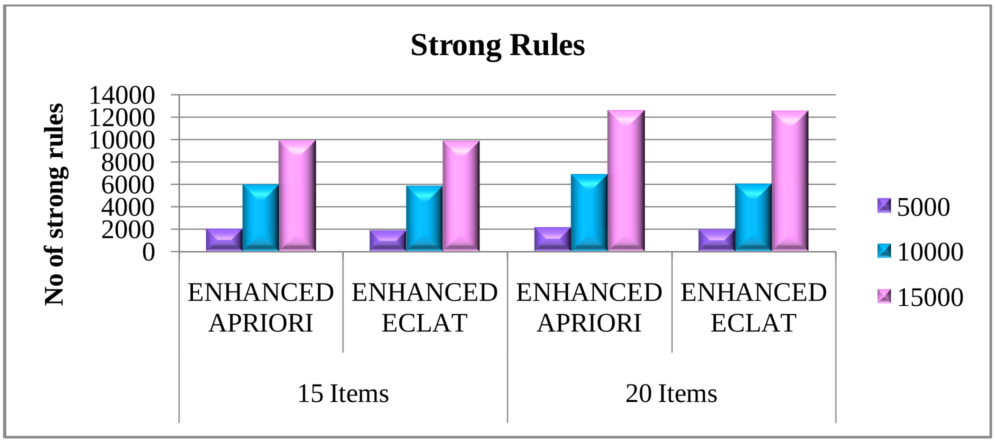

Fig.3 Number of strong rules

Tables 3 givesthe result of execution time in milliseconds using Enhanced-Apriori and Enhanced Eclat for different size of datasets.

Table.4Execution Time in Milliseconds

DATASET SIZE	15 Items		20 Items	
	ENHANCED APRIORI	ENHANCED ECLAT	ENHANCED APRIORI	ENHANCED ECLAT
5000	2328	2813	10390	10985
10000	3234	3968	16721	16856
15000	9907	9969	21543	22937

Figure 4 analysis the performance of execution time. Comparison is done between Enhanced-Apriorialgorithmand Enhanced-Eclat algorithm.it has observed that Enhanced-Apriori algorithm gives best results than Enhanced-Eclat algorithm.

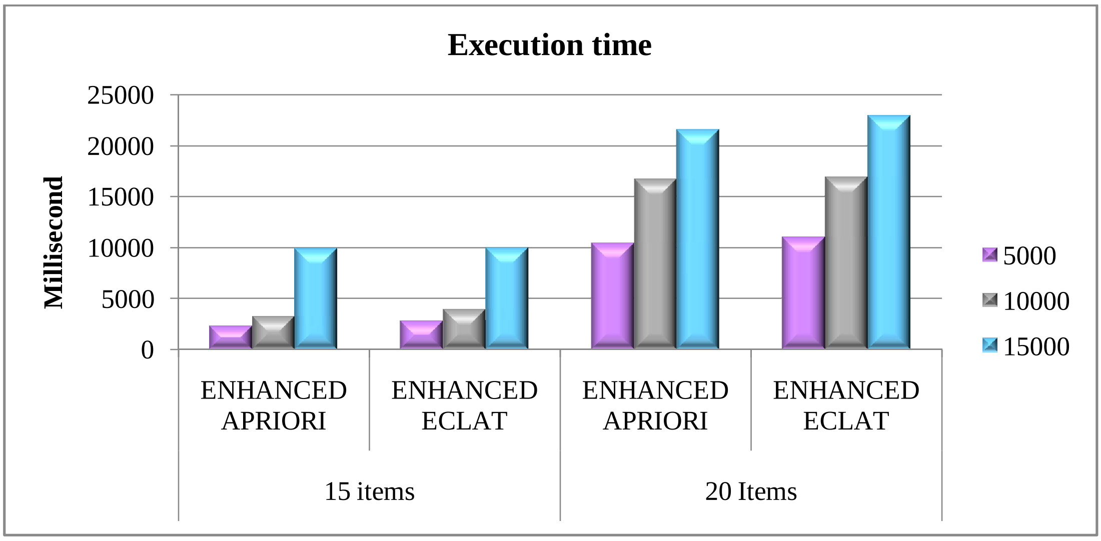

Fig.4Execution time in milliseconds

Tables 5 gives the results of memory usage in kilo bytes using Enhanced-Apriori and Enhanced Eclat for different size of datasets.

Table.5 Memory usage in kilobytes

DATASET SIZE	**15 Items**		**20 Items**	
	ENHANCED APRIORI	**ENHANCED ECLAT**	**ENHANCED APRIORI**	**ENHANCED ECLAT**
5000	1015	1873	4158	4305
10000	7195	7964	7809	7901
15000	9245	10139	10230	10507

Figure 5 shows the performance of memory usage. Comparison is done between Enhanced-Apriori algorithmand Enhanced-Eclat algorithm.it has observed that Enhanced-Apriori algorithm gives best results than Enhanced-Eclat algorithm.

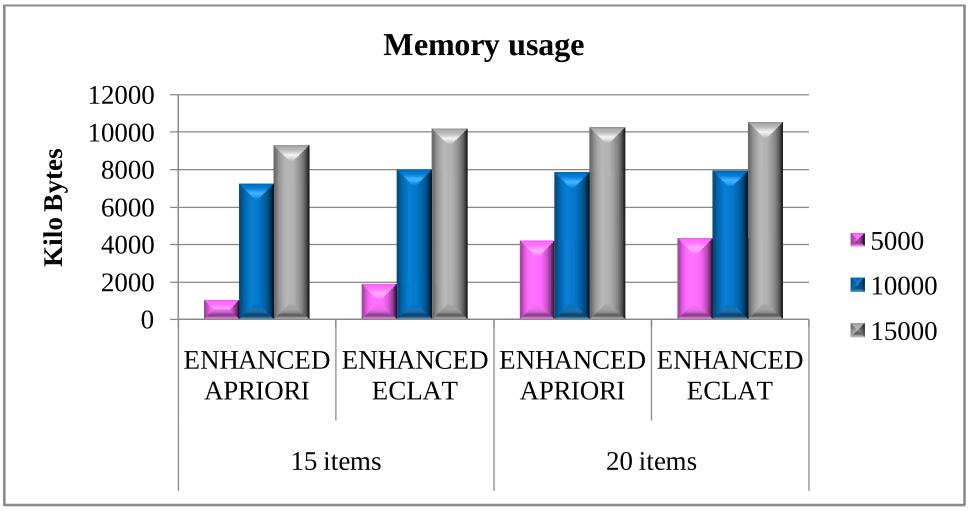

Fig.5Memory usage

VI. Conclusion and Future Work

Association rule mining is the most effective data mining technique to discover hidden pattern from large volume of data. This research work is mainly focused on, to find the more number of frequent itemset, generate association rules and identifies strong rules. In the research work two algorithms namely Apriori and Eclat are enhanced. Individual threshold is set for all items and candidate generation is done with minimum threshold value. From the performance metrics it has observed that Enhanced-Apriori algorithm is best comparing to Enhanced-Eclat Algorithm. In future, this work will be implemented with different size and types of dataset like medical, bioinformatics, CRM, telecommunication etc.

References

1. Karthikeyan.T and Ravikumar.N “A Survey on Association Rule Mining” International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 1, January 2014 ISSN (Print) : 2319-5940 ISSN (Online) : 2278-1021
2. Abaya, Sheila A. "Association Rule Mining based on Apriori algorithm in minimizing candidate generation" International Journal of Scientific & Engineering Research 3.7 (2012): 1-4
3. Malli, K. Mohana Siva Naga, Sumanasri Bezawada, and Vathira Vijayan. "Analysis of Association Mining through Enhanced Apriori Algorithm." ISSN 2278-3091.
4. Ashok, Abhang Swati, and S. JoreSandeep. "The Apriori algorithm: Data Mining Approaches is to find frequent item sets from a transaction dataset."International Journal of Innovative Research in Science, Engineering and Technology 3 (2014).
5. https://www.linkedin.com/pulse/what-heck-association-rules-analytics-jeffrey-strickland-ph-d-cmsp
6. V.Moustakides a, Vassilios S. Verykios b,* Rio, Greece, “A MaxMin approach for hiding frequent itemsets Data & Knowledge Engineering “65 (2008) 75–89 George e Received 5 June 2007; accepted 5 June 2007 Available online 18 July 2007
7. Amit Mittal1, Ashutosh Nagar2, Kartik Gupta3, Rishi Nahar ,” Comparative Study of Various Frequent Pattern Mining Algorithms “ International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 4, April 2015 Copyright to IJARCC DOI 10.17148/IJARCCE.2015.44127 550
8. Dutt, Shalini, Naveen Choudhary, and Dharm Singh. "An Improved Apriori Algorithm based on Matrix Data Structure." Global Journal of Computer Science and Technology 14.5 (2014).
9. Patel Tushar S.1, Panchal Mayur2, Ladumor Dhara2, Kapadiya Jahnvi2, Desai Piyusha2, Prajapati Ashish3 and Prajapati Reecha4, Association.” An Analytical Study of Various Frequent Itemset Mining Algorithms”, Research Journal of Computer and Information Technology Sciences Vol. 1(1), 6-9, February (2013) Res. J. Computer & IT Sci. International Science Congress’
10. Manasa G* , Mrs. Kulkarni Varsha, “Integration of Apriori and FP-Growth Techniques to Personalize Data in Web” , International Journal of Scientific and Research Publications, Volume 5, Issue 7, July 2015 1 ISSN 2250-3153 www.ijsrp.org IAFP:
11. Zhi-Hong Deng , Sheng-Long Lv, Fast mining frequent itemsets using Node sets, Expert Systems with Applications 41 (2014) 4505–4512
12. Jian pei1, Jiawei Han, Hongjun lu, shojiro nishio, shiwei tang and dongqing yang “.h-mine: fast and space-preserving frequent pattern mining large database”

13. Vivek Badhe, Parul Richharia"A Survey on Association Rule Mining for Finding Frequent Item Pattern "2016 IJSRSET | Volume 2 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN: 2394-4099 Themed Section: Engineering and Technology.
14. Jeetesh Kumar Jain, Nirupama Tiwari, Manoj Ramaiya A Survey: On Association Rule Mining, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 3, Issue 1, January -February 2013, pp.2065-2069
15. Manoj Ganjir1 , Jharna Chopra2 "Combining Apriori and FP Growth algorithms with Simulated Annealing for Optimized Association Rule Mining" International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 12, December 2015.
16. J.R.Jeba,Kumaracoil Dr.S.P.Victor, Comparison of Frequent Item Set Mining Algorithms, International Journal of Computer Science and Information Technologies, Vol. 2 (6) , 2011, 2838-2841
17. Pramod S. O.P. Vyas Survey on Frequent Item set Mining Algorithms International Journal of Computer Applications (0975 - 8887) Volume 1 – No. 15 86
18. Kamani Gautam, J., Y. R. Ghodasara, and Vaishali S. Parsania. "Mining Frequent Itemset Using Parallel Computing Apriori Algorithm."
19. Dr. S. Vijayarani and R. Prasannalakshmi Comparative analysis of association rule generation algorithms in data streams. International Journal on Cybernetics & Informatics (IJCI) Vol. 4, No. 1, February 2015
20. Trieu Anh Tuan A Vertical Representation for Parallel Dclat Algorithm in Frequent Itemset Mining Ritsumeikan University 2012

BIOGRAPHY

Dr. S.Vijayarani has completed MCA, M.Phil and Ph.D. Working as Assistant Professor in the School of Computer Science and Engineering, Bharathiar University, Coimbatore. Her fields of research interest are data mining, privacy and security issues, text mining, web mining information retrieval, data streams and big data. She has authored a book and published more than 70 research papers in the international journals and also presented research papers in international and national conferences.

Ms. S. Sharmila has completed MCA in Computer Science. She is currently pursuing her Ph.D. in Computer Science in the School of Computer Science and Engineering, Bharathiar University, Coimbatore. Her fields of interest are Data Mining, Association Rule Mining and Association Rule Hiding.