American Journal of Engineering Research (AJER) 2015
American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-4, Issue-12, pp-126-133
www.ajer.org
Research Paper Open Access

Adaption of Fast Modified Frequent Pattern Growth approach for frequent item sets mining in Telecommunication Industry

Sanjib Kumar Routray[1] , Sasmita Mishra[2] ,Laxman Sahoo[3]
[1] *Research Scholar, Utkal University, Bhubaneswar, Odisha, India*
[2] *Asst. Professor, IGIT, Saranga, Dhenkanal,Odisha,India*
[3] *Professor, Computer Science &Engineering, KIIT, Bhubaneswar, Odisha, India*

ABSTRACT: *A Fast Modified Frequent Pattern Growth approach namely (F-MFPG) is presented to mine the frequent item sets through adaption of frequent growth method. From experimental analysis on CRM real datasets in special reference to Telecommunication Industry, this approach improved the mining efficiency of Association rule. In this paper modified FP-tree algorithm with reduced header table and Auxiliary tree and FFIM algorithm for association rule mining is proposed. The advantage of F-MFPG approach is finding association rules without candidate set as well as CP-tree generation, which saves the execution time.*

Keywords**:** *Association Rules, Customer Relationship Management, Frequent Item sets, Reduced Header Table, Main Modified FP-Tree, Auxiliary Tree*

I. INTRODUCTION

In recent years, handling of large database from several sources is quite difficult. In Telecommunication Industry, conversion of large amount of data into useful and interesting information through knowledge discovery process can be achieved by various data mining functionalities like Association, Correlation analysis, Classification, Cluster analysis[1]. Out of which rules of Association are one of the most important prime research methods. Customer relationship management is the active module of Telecommunication Industry, where Association rules may due to several reasons. (1)Market Basket Analysis : Process of observing customer buying habits (2)Competitive Market : Successfully tailoring the marketing strategy through understanding customer's personal & demographic characteristics (3)High Churn rate : Predicting whether customer will churn with reason (4)Big data collection : Predicting the customer behavior in future that helps the management for making effective decisions. Association rules mining is for finding strong association, which can be divided into two parts. (1)Determining frequent item sets by using two interesting measures, support and confidence (2) Generating Association rules from frequent item sets. An Association rule in the form X =>Y where X,Y are finite set of items x_1,x_2,x_3……..x_n and y_1,y_2,y_3…….y_n represents that if the set of items X in a transactions exists, then set of items Y occurs with high probability in same transactions[13]. In this regard Apriori heuristic (Agarwal & Srikant 1994) have two important drawbacks. First repetitive scanning, which needs lot of I/O head. Second it requires huge candidate sets[14]. For example, if any length 'a' pattern is not frequent, its length (a + 1), super pattern can never be frequent. Hence it is required to generate iteratively huge set of candidate patterns of length (a + 1) from the frequent pattern of length 'a'(for 'a' greater than equal to 1). For 10^4 frequent item sets, Apriori approach needs to generate $10^7 - 2$ candidates. Frequent pattern of size 100 (x_1,x_2…….x_{100}), it must generate $2^{100} - 2 = 10^{30}$ candidates(say), which is very costly[8]. In order to avoid those repeated scanning and checking of large candidates, a new novel research developed called frequent pattern growth mining. The advantage of FP-Growth algorithm is, scanning database two times, compressing database into frequent pattern tree to ensure the data structure compact and informative without using a candidate key and generates various Association rule. However in FP-Growth mining, there are again two drawbacks. First in FP-Tree, conditional FP-Tree and its traversal requires more time in recursive digging, that affects the efficiency of algorithm with respect to time and space. Second for this traditional algorithm, construction of huge FP-Tree is essential that consume long processing time for some amount of processing. For this problem, this paper proposes a novel version of efficient association mining techniques by using three data structure (1) Modified FP-Tree (2) Reduced Header Table (3) Auxiliary Tree to discover valuable knowledge model from large Telecommunication data sets for customer relationship management in a fast and efficient way. The outline of the remaining paper is as follows. Sec-II discusses the literature review. The main idea of theory and methodology in detail is discussed in Sec-III. Sec-IV illustrates the experimental results and implementation. Finally conclusion describes in Sec-V.

II. REVIEW OF LITERATURE

Data mining play a crucial role in Telecommunication Industry due to availability of huge data sets for Customer relationship management [1] and big data under Telecommunication Industry implemented frequent pattern mining by interpreting the relationships between characteristics of input data as proposed by [2]. On research it has shown by [3], that a good number of Telecommunication companies using data mining models for improving a CRM strategy for keeping customer happy. A general survey has been done by [4] on Association rule mining with different merits and demerits of different data mining methods like Apriori, FP-Growth with positive and negative association rules. The Association rule mining is to find all association rules above minimum user defined support threshold and confidence and it is done by two steps (1) Finding all frequent pattern (2) generating all association rules from frequent pattern. Every frequent pattern mining techniques is a unique as suggested by [5] and can be applied depending on input data in many applications of real life. For finding frequent pattern, Apriori algorithm uses prior knowledge of frequent item set properties but it suffers several drawbacks like generation of candidate sets and multiple scanning, which creates lot of memory and time complexities [6]. To avoid generating many candidate sets and multiple scanning, [7] proposes Rapid Association rule mining (RARM), which shows better performance. Finally [8] proposed a mining of complete set of frequent item sets without candidate generation on divide-and-conquer principle. First scanning the database once for deriving the list of frequent items in order of descending frequency and then compressing the database into FP-Tree for retaining item set association information. This frequent pattern mining has ample scope of data analysis with deep impact on pattern mining applications presently and in future for extracting the new information that helps for more guidance as per research study by [9] [10]. Also [11] suggested that for handling massive small files, an improved parallel FP-Growth algorithm required for frequent item sets mining, which increases good speed-up and higher mining efficiency. Similarly for achieving better performance on frequent item set mining, new algorithm proposed by [12] namely Header Table Recursion (HTR) , where no. of new FP-Tree generation decreases on each recursive call of classical FP-Growth algorithm. As traditional FP-Tree, generating many candidate sets and CP-Tree, its efficiency decreases. In order to avoid those drawbacks, an improved version of FP-Tree algorithm with a modified header table along with a spare table and mining frequent item sets algorithm can be developed as Improved FP-Growth techniques decreases space and time complexities by [13]. It is concluded that FP-Growth techniques further can be modified by using different novel data structure to solve the time and space problems.

III. THEORY AND METHODOLOGY

Traditional Apriori algorithm involves expensive candidate generation process for mining the complete set of frequent item set. Hence an interesting method attempted called frequent pattern growth approach or simply FP-Growth, where divide-and-conquer strategy adopted instead of generate-and-test of Apriori. In this method, first database will be compressed for representing frequent items into frequent pattern tree, which retains the item set association information. Second it divides the compressed database into conditional database, each associated with one pattern fragment that mines each database separately.

Advantages: (1) Faster execution time than Apriori. (2) Only two scanning over complete datasets. First scan involves collection and sorting the frequent items and second for constructing FP-Tree. (3) Compresses complete data sets. (4) Construction without candidate sets.

Disadvantages: (1) More expensive for building complete FP-Tree by using Telecommunication Data sets. (2) May not fit in memory space. (3) More time needs to build complete FP-Tree. (4) If user defined support threshold is high, time will be wasted unnecessarily. (5) Support of all individual items calculated once the entire data-set is added to FP-Tree. (6) Increasing of space and time complexity, as it involves recursive call for tree traversing.

3.1 Proposed Fast Modified FP-Growth Approach (F-MFPG)

3.1.1 Idea behind proposed method

Out of different methods of mining association rules, FP-Tree is the latest method, where entire transactional datasets converted into a compact prefix tree structure. The path/branch of a FP-Tree is the representation of individual transaction and FP-Tree removes various shortcomings of traditional association rule mining methods. But fast proposed modified FP-Tree is highly condensed, which removes all shortcomings of traditional FP-Tree (1) expensive multiple scanning (2) Unnecessarily traversal of complete tree, while checking the particular node in the tree.

3.1.2 Reduced Header Table

Proposed reduced header table only keeps the items with their corresponding frequencies, which are already present in the FP-Tree according to descending order of support. The main advantage of reduced header table is all the items in the transactional data sets may not present always in the header table, in comparison to traditional to traditional header table of FP-Tree.

3.1.3 Auxiliary Tree

Auxiliary Tree is the third type of data structure, where existence of node item name and their node count of same item in the proposed modified FP-Tree. Initially auxiliary tree will be constructed, where 2nd most frequent item will be the root node and other node, whose frequencies on descending order will be child node correspondingly. Starting from root node to end node, node-count will be initialized to zero initially. There are only two cases, when items in the individual transaction will be added to auxiliary tree.

(1) When the most frequent item is not present as first element of individual transaction, all transactional node item will be added in auxiliary tree as a result node-count will be incremented.
(2) When there is no direct link between the current root and item in consideration and also exist in the FP-Tree, then all the following items in the transaction will be added to auxiliary tree as a result node count will be incremented.

3.2 Proposed Main Modified FP-Tree Generation Algorithm

First transactional data sets are read initially for finding the frequency of each item. Then most frequent item is identified. Declare user defined minimum support threshold and remove items, which do not have minimum support threshold. Then arrange all transactions in descending order of frequency of items. Initially created Root node initialized to NULL. Let first item in each transaction be 'X' & remaining be 'Y'. First check whether 'X' is the most frequent item & child of root, if not create a node corresponding to 'X' as the child of root node. If exists increment the node-count in the reduced header table & root will be shifted. In the auxiliary tree, create a root node as 2nd most frequent item and other child items exists in order of descending support correspondingly and initialize node counts of all items to zero. If X is not the most frequent item, then all the items in that transactions are moved to auxiliary tree and node count of added items will be incremented. For each item Y in the transaction, check direct link between current root and node-item, if exists, then increment the count in the reduced header table and root will be moved. If not exists, then again check, whether the node corresponding to that item exists in the FP-Tree through reduced header table. If exists, then move the items to the auxiliary tree and node-count of auxiliary tree will be incremented. If not, node is not available in modified FP-Tree, then child current root create corresponding to that item and move to this node. If an item added to auxiliary tree, then irrespective of all conditions, all the following items in those transactions are also moved. Repeat the procedure until all transactional datasets are read.

INPUT: Transaction Database
OUTPUT: Main Modified FP-Tree, Reduced Header Table & Auxiliary Tree
STEP-1.List out all items in the transaction dataset & find its Support for each item.
STEP-2.Declare the minimum support threshold and remove the item from list, which do not have minimum threshold.
STEP-3.Identify the most frequent item in the transaction dataset & for each transaction in the dataset sort the individual transaction based on the support in a descending order.
STEP-4.Create a root node which is referred to as Original root of a main modified FP-Tree.
STEP-5.Set the original root as current root for each transaction of main-modified FP-Tree.
STEP-6.Create a root node as 2nd most frequent item from list of the items of Auxiliary Tree. Other child items exist in order of decreasing support correspondingly & initialize node counts of all items to zero.
Let the first item in each transaction be 'X' and remaining be 'Y'.
If
'X' is the most frequent item
If
'X' is not child of current root of main modified FP-Tree
Create 'X' as the child of current root
Make count of the first item in reduced header table as 1.
Make the newly created node as current root
Else
Make 'X's node as the current root
Increment the count of 'X' in reduced header table
For all frequent items 'Y' when 'X' is the most frequent item
If 'y' exists as a child of current node
Increment the count in the reduced header table
Move the current root to the child node
Else If 'Y' is not present in the reduced header table
Create a new node for 'Y' as the child of current root
Make the count of corresponding item in the reduced header table as 1.
Make the newly created node as current root

Else
Increment the node count of remaining item in Auxiliary tree
Else
Increment the node count of all the items in the Auxiliary tree.

3.3 Proposed Main Modified FP-Tree Generation Algorithm description with CRM-Dataset example

In the CRM master database, of one Telecommunication Company, following example transactional CRM-data set is considered for describing proposed main modified FP-Tree on which, 32 no. of attributes are present. Out of which 9 (nine) attributes are selected for forming a transactional data sets of 10 (Ten) transactions on 6 (six) various types of data items as per Table-I.

Table 1: Sample Transactional CRM-data set for main modified FP-Tree

Transactions	Items
100	a,b
200	b,c,d
300	a,c,d,e,f
400	a.d,e
500	a,b,c
600	a,b,c,d,f
700	a,f
800	a,b,e
900	a,b,d
1000	b,c,e

On scanning of above transactional data sets, we find the support for each item on descending order. User defined minimum threshold support assumed in this case is 4. Identify the most frequent item as 'a' as the frequency is 8. Identify the item which does not meet the minimum user defined threshold support as 4. Accordingly Item 'f' removed because its frequency is less than minimum support as per Table-II.

Table 2: Transactions Item and their Support

Item	Support
A	8
B	7
C	5
D	5
E	4

Identify the most frequent item in the transaction dataset & for each transaction in the dataset sort the individual transactions based on the support in a descending order and create a root node which is referred to as "original root" and initialize it to "NULL" in case of main modified FP-Tree. Create a root node as 2nd most frequent item from list of the items of auxiliary Tree. Other child items exist in order of decreasing support correspondingly & initialize node counts of all items to zero.

(1)Transaction 100: [a,b]

As [a], is the most frequent item and not the child of original root, new node will be created corresponding to [a] and child of original root will be made. Similarly [b] is not a child of [a] and [b] is not in the tree so node corresponding to [b] will be created and represented as child of node corresponding to [a]. There will be no change in auxiliary tree.(M-Tree stands for Main Modified FP-Tree and A-Tree stands for Auxiliary Tree for all figures)

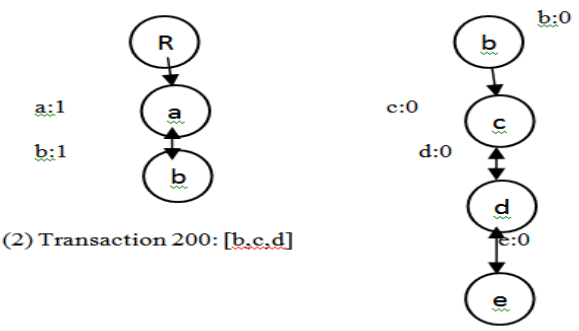

Fig. 1: M-Tree and A-Tree representation of Transaction [a,b]

Since [b] is not the most frequent items in this transaction, so all items present in the transaction are sent to auxiliary tree and node count of above transactional items in the auxiliary tree will be incremented.

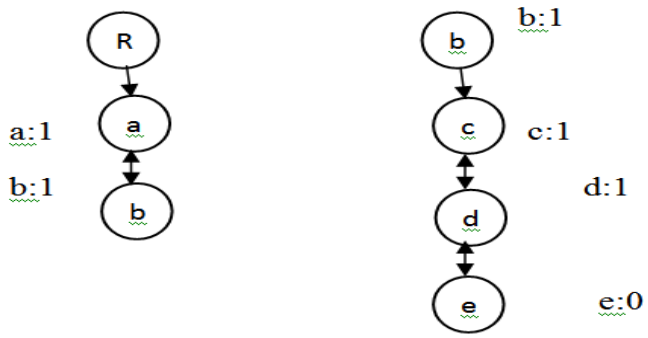

Fig. 2:M-Tree and A-Tree representation of Transaction [b,c,d]

(3) Transaction 300: [a,c,d,e,f]
[a] already exists in the main modified FP-Tree. Hence count of [a] will be incremented in the reduced header table. In this transaction as [a] is not direct link from [c], all remaining items including [c] are added to auxiliary tree. Node count of items in the auxiliary tree will be incremented accordingly. [f] is not considered because it is removed initially as its frequency is less than user defined minimum support threshold.

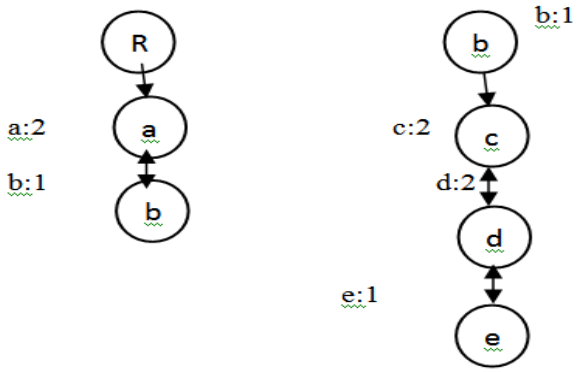

Fig. 3:M-Tree and A-Tree representation of Transaction [a,c,d,e,f]

(4) Transaction 400: [a,d,e]
[a] already exists. Hence count of [a] will be incremented in reduced header table. Since [d] is not a child of [a] and is not available in the main modified FP-Tree, a node corresponding to [d] will be created as a child node representing [a]. Similarly [e] will be created as a child node representing [d]. There will be no change in the auxiliary tree.

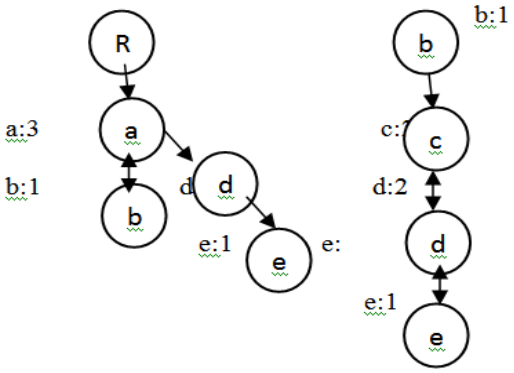

Fig.4:M-Tree and A-Tree representation of Transaction [a,d,e]

(5) Transaction 1000: [b,c,e]

Since [b] is not the most frequent items in this transaction, so all items present in the transaction are sent to auxiliary tree and node count of above transactional items in the auxiliary tree will be incremented.

The final main modified FP-Tree and auxiliary tree obtained as per figure-10.

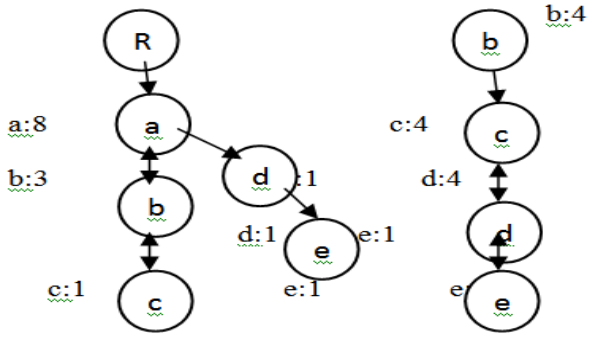

Fig.5:M-Tree and A-Tree representation of Transaction [b,c,e]

3.4 Fast Frequent Item sets Mining (FFIM) Algorithm

The proposed main modified FP-Tree along with the reduced header table and auxiliary tree are given as input to the FFIM algorithm that reduces the complexities of unnecessary generation of CP-Trees. In this algorithm, by traversing reduced header table FP-Tree items are accessed one by one as items in the modified FP-Tree are same as items in the reduced header table. First of all frequency of each item is compared with user defined minimum support threshold, if frequency is less than minimum support, all the items in the modified FP-Tree connected to that most frequent items are taken into account. If frequency is equal to user defined minimum support threshold, the item sets at lower index than the current item's index in reduced header table are taken into account as initially on descending order. Accordingly frequent item sets are generated by using these items with support is equal to sum of the frequency and auxiliary tree node-count. If frequency is more than user defined minimum support threshold, frequency is frequency of that item as present in reduced header table considered as above in same manner. FFIM Algorithm is as follows.

INPUT: Modified FP-Tree, Reduced Header Table, Auxiliary Tree

Output: Items set 'A' containing all frequent item set with their corresponding frequencies

Initially 'A' is empty

For all items in the reduced header Table

S=User declared minimum support threshold

F=Frequency of the item in reduced header table

Auxiliary Node count as AN Count = count in the node item of Auxiliary tree

If

F is not equal to S

If

F is greater than S

Frequency of frequent item set is FF = F

Else

Frequency of frequent item set is FF = F + AN Count

Generate all maximum possible combination of the current item and all the nodes up to most frequent item node in modified FP-tree and than to 'A' with their frequencies as FF

Else

Frequency of frequent item set is FF = F

Generate all possible combination of the current item and all the elements present at the lower index than the current item's index in the reduced header table

IV. EXPERIMENTAL RESULTS AND IMPLEMENTATION

The fast frequent item sets with their frequencies are generated by inputting main modified FP-Tree, reduced header table and auxiliary tree to the fast frequent item sets mining (FFIM) algorithm. Fast of all items will be accessed in the reduced header table in the evaluation as items are already stored in sorted manner. Initially we compare with user defined minimum support threshold with frequency of item in main modified FP-Tree. In above example 4 is the minimum support and frequency of first item [a] present in reduced header table is 8, which is greater. So frequency of frequent item sets generated using this item will be – 8. Hence by considering all possible combination of current item with all the nodes up to most frequent item node present in main

modified FP-Tree, frequent items generated. Frequent items will be {{ a : 8 }}. The second item in reduced header table is [b],whose frequency is 3, which is less than minimum support frequency. So the frequency of frequent items generated will be addition of main modified FP-Tree frequency (3) and auxiliary node count of that item [b]. Hence by considering all possible combination of items up to most frequent item node present in main modified FP-Tree, frequent items generated. Frequent item will be {{b : 7},{a,b : 7}}. Similarly for third item in the reduced header table is [c], frequent item will be {{c : 5},{a,c : 5},{b,c : 5},{a,b,c : 5}}.
Similarly frequent item sets will be generated for all the items.

4.1 Generating Association Rule

If we consider minimum confidence threshold is 50%, the frequent item set {{a,b,c : 5)) is taken to mine the association rules accordingly

1. a,b => c
   Frequency {{a,b,c}} = 2
   Frequency {{a,b,}} = 5
   Hence confidence is 2/5 = 0.4 = 40% (Not selected)
2. b,c => a
   Frequency {{a,b,c}} = 2
   Frequency {{b,c}} = 4
   Hence confidence is 2/4 = 0.5 = 50% **(Selected)**
3. a,c => b
   Frequency {{a,b,c}} = 2
   Frequency {{a,c}} = 3
   Hence confidence is 2/3 = 0.66 = 66% **(Selected)**
4. a => b,c
   Frequency {{a,b,c}} = 2
   Frequency {{a}} = 8
   Hence confidence is 2/8 = 0.25 = 25% (Not selected)
5. c => a,b
   Frequency {{a,b,c}} = 2
   Frequency {{c}} = 5
   Hence confidence is 2/5 = 0.4 = 40% (Not selected)
6. b => a,c
   Frequency {{a,b,c}} = 2
   Frequency {{b}} = 7
   Hence confidence is 2/7 = 0.28 = 28% (Not selected)

Accordingly we find the association rules by using all frequent item sets generated by FFIM algorithm.

4.2 Challenges

In the main modified FP-Tree, each item may appear at most once for making very condensed. Reduced header table contains pointer to each node in the main modified FP-Tree with their frequencies. As there is no recursive calls during tree traversal in FFIM algorithm, time complexity and space complexity increases. As no. of database scans is less than some traditional association rule mining, saving of execution times is the main challenging issue in this case. Further no conditional Pattern Tree and candidate set generation required for reducing complexities.

4.3 Implementation

Experimental analysis on CRM data sets of Telecommunication Industry containing 5000 transactions and 32 items between the traditional FP-Tree and proposed Fast modified FP-Tree. The implementation of both the algorithms was done in matlab 7.12(2011A) version in windows platform. The analysis of simulation result shows that the proposed algorithm has a better detection as well as the faster rate of execution as compared to the existing one. The used data sets in the analysis were obtained from one CRM-Master database of one Telecommunication Industry. During experimental analysis on implementation phase, it is observed that, proposed algorithm takes less time to find association rules in comparison to Traditional FP-Tree algorithm.

V. CONCLUSION & SCOPE FOR FUTURE WORK

5.1 Conclusion

The proposed fast modified FP-Growth approach (F-MFPG) can be represented in condensed form of large CRM-data sets in Telecommunication Industry. As reduced header table avoided many recursive calls during tree traversal of main modified FP-Tree during implementation phase, time complexity is optimized and saving of execution time.

5.2 Scope for future work

Using F-MFPG pattern, Telecommunication Industry may find various association rules of billing & network modules.

As time consumption & requirement of space is very less, proposed approach may apply on other data mining techniques like classification for finding effective decision in Telecommunication Industry.

REFERENCES

[1] Marwah,Ranju.: *"Data mining techniques and applications in Telecommunication Industry". IJARCSE, Vol 4 issue9 :* pp.430-433, 2014

[2] Akioka ,Sayaka.: *"Data performance characterization of frequent pattern mining algorithms", IJDKP ,Vol 5 issue 1 :* pp.51-70, 2015

[3] Camilovic,D.: *"Data mining and CRM in Telecommunication, Serbian Journal of Management" ,Vol 3 issue1 :*pp.61-72, 2008

[4] Jain, J K., Tiwari, N., Ramaiya, M.*:" A survey on Association Rule mining", IJERA,Vol 3 issue 1 :* pp.2065-2069, 2013

[5] Patil, V S., Deshpande, N A.*:" Pattern mining techniques of data mining", IJETAE,Vol 4 issue3* :pp. 523-529, 2014

[6] Das, N., Ghosh, A., Das, P.:" Mining Association Rules to evaluate consumer perception": A new FP-Tree Approach, *Proc 5th Int Conf Centre for Quality:* pp. 855-872, 2011

[7] Vijayarani, S., Sathya, P.: *"An efficient algorithm for mining frequent items in data stream" ,IJIRCC , Vol 1 issue3 :*pp. 742-747, 2013

[8] Han, J., Pei, J., Yin, Y., Mao, R.: *"Mining frequent patterns without candidate generation:A FP Tree Approach". Proc Int Conf on Management of Data (SIGMODOO), Dalton, TX:* PP. 1-12, 2000

[9] Jiawei , H., Hong, C., Dong, X., Xifeng, Y.: *"Data mining knowledge description", Springer Science, Business Media:* pp. 55-86, 2007

[10] Abdullah, Saad Almalaise Alghamdi. *:"Efficient implementation of FP-Growth algorithm : Data mining on Medical data", IJCSNS,Vol 11 issue 12* :pp. 7-15, 2011

[11] Dawen, X., Yanhui, Z., Zhuobo, R., Zili, Z.: *"An improved parallel FP-Growth algorithm for frequent item set mining", Proc 59th ISI World Statistics Congress:* pp. 25-30, 2013

[12] Tianjun, Lu., Tian,Si., Wang, Shal.*:" Header table recursion algorithm for mining frequent patterns", AISS Vol 5 issue 2 :*pp. 769-775, 2013

[13] Agarwal , V., Kushal, M., Kumar, P.*: "An improvised frequent pattern tree based Association Rule mining technique with mining frequent item sets algorithm and a modified header table", IJDKP , Vol 5 issue (2)* :pp. 39-51, 2015

[14] Zhou, Lijuan., Wang, X.: *"Research of FP Growth Algorithm based on Cloud Environments", Journal of Software, Vol-9 issue(3)*:pp .676-683, 2014