© 2014 IJIRT | Volume 1 Issue 11 | ISSN: 2349-6002

Survey or Analysis of Centralized and Distributed Association Rule Mining Algorithm

Viral Jethava[1], Prof. Risha Pandey[2]
[1]*Student of M.E CE, India.*
[2]*Assistant Professor, Computer Engg Dept., India.*

Abstract*– Among the available data mining methods Association rule mining popular one. However, mining association rules often provides very large number of rules, leaving the analyst with the task to go through all the rules and find out suitableones. Most of the historical studies use an Apriori-like candidate set generation-and-test approach. However, candidate set generation is still not cheaper, especially when we have large & long patterns. Moreover, dynamic itemset counting algorithms, which are an extension of Apriori algorithms, will help to decrease the number of scans forthe chosen dataset. This can be termed as alternative approach to Apriori Itemset Generation. In which, itemsets are dynamically added and deleted as transactions are read.It depends on the fact that for an itemset to be frequent, all of its subsets must also be frequent, so we have to monitor only those itemsets whose subsets are all frequent. This paper presents the comparison of the performances of Apriori, Frequent Pattern Growth and DIC algorithm. Here, the performance analysis done, based on the time taken for the execution for different number of instances and confidence in different datasets. The performance study shows that the fp-growth method is more efficient and scalable, is about an order of magnitude faster than the Apriori algorithm.

***Index Terms*- Association rules, Apriori, Dynamic itemset counting, Frequent pattern growth,Support and Confidence.**

I. INTRODUCTION

Data mining [1] can be seen as an outcome of natural evolution of information technology. This is all about utilizing the information by acquiring the hidden but useful underlying knowledge. Another way to interpret this is a practice of searching over huge data for discovering patterns and trends automatically.Thus, this is not bound with the limitations of simple analysis. Among with the different available methods for mining data, association rule mining is very famous one. In this technique, an interrelation among different items in data is dig up by determining frequent large itemsets which are repeated more than a threshold number of times in the database. Data mining process couldbe defined as centralized or distributed based on the location of data. While considering centralized data mining process, data is located into a single site but in distributed process data is resided atmultiple sites. Data Mining Tasks are mainly categorized into two typesas shown byFig-1.

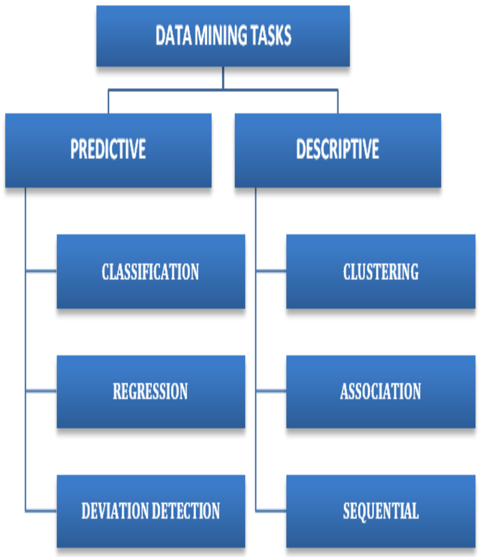

Fig-1: Different Task of Data Mining

II. OVERVIEW OF ASSOCIATION RULE MINING

Let usintroduce association rule mining in detail. In this different Association Rule Mining (ARM) Algorithms[2] will be expounded together. Problem of Association Rule was first stated by Agrawal [2], that the conventional statement of association rule mining problem was discovering the interesting association or correlation relationships among a large set of data items.

With increasing applications on iternset data mining has been applied to the distributed environment of large amount of data. Since data are located in different sites,an efficient algorithm is needed to mine the large amount of data[3].This paper proposes distributed database algorithm (DD) for mining the association rules by finding the local frequent itemset and furthergenerate global frequent itemset. The efficiency of this algorithm is compared with the standard FP-Growth algorithm[4].

Association rule mining (ARM) [2] has become one of the basic data mining tasks,whichhas attracted extent interest among data mining researchers. Association rule mining is one of the major tasks of the Data Mining. Association rule mining could be helpfulfor extracting knowledge in various applications like advertisements, bioinformatics, database marketing, fraud detection, E-commerce, health care, security, sports, telecommunication, web, weather forecasting, financial forecasting, etc. ARM is an unsupervised technique which works on variable length ofdata, and leads to clear and understandable results. Association rule mining finds the interesting or correlation relationship among a large set of data items. The typical example of association rule mining is the market basket analysis. Frequent itemset mining leads to the discovery of associations and correlation among items which are in large transactional or relational data sets. In brief, an association rule is an expression A ⟹B, where A and B are sets of large items. Generally association rule mining is done in two steps.

• **Finding all frequent itemsets:** The items which are frequently or habitually purchased together in one transaction are called the frequent itemsets.

• **Generate strong association rules from the frequent itemsets:** the rules which are generated must need to satisfy minimum support as well as minimum confidence.

Typically, association rules can be considered interesting if they could satisfy minimum support threshold and a minimum confidence threshold.

Let us take **I =** ***i1, i2, ….im*** as a set of items. Take D, the data relevant to task, as a set of transactions where in each transaction T is a set of items such that T⊆I. Here, for each and every transaction there is a unique identifier, TID. Let assume A to be a set of items available. A transaction T is said to contain A if and only if A⊆T. An association rule is an implication of the form A⟹B, where A⊂T,B⊆T and A⋂B=∅. Here, the rule A⟹B holds in the transaction set D with support **s**, where **s** is the percentage of transactions in D that contain A∪B. The rule A⟹B is said to have confidence **c** in the transaction set D, if there are c percentage of transactions in **D** that contains **A** and also contains **B**,

- Support:Support of a particular itemset is the percentage of the total transactions having that itemset.
  Support(A⟹B) = prob{A ∪ B}
- Confidence:Confidence is the probability of occurring (such as buying) items together.
  Confidence(A⟹B)=prob{B/A}

The methodsby which we can improve efficiency of association rule mining algorithms:

- Reduce database Scan
- By sampling the database
- By adding Some extra constraints on the structure of patterns
- Using parallelization

For the generating association rules many algorithms were presented. Numbers of association rule mining algorithms have been developed with different mining efficiencies. Any algorithm should find the same set of rules though their computational efficiencies and heaving different memory requirements. Let us discuss some well-known algorithms like Apriori, FP-Growth and DIC in brief.

A. Apriori:

The Apriori algorithms has been developed by RakeshAgrawal[1] and colleagues which is one of the best ARM algorithms. Which serves as the base algorithm for most of the parallelalgorithms[5]. Apriori works in bottom-up fashion with a horizontal layout for searching and enumerates all the frequent itemsets. This is an iterative algorithm, which counts itemsets of specific length in a particular database pass. The process startsby scanning all transactions in the database and computing the frequent items.As a scan result, a set of potentially frequent C*andidate* 2-

item sets is formed from the frequent items&next database scan obtains their supports. The frequent 2-itemsets are retained for the next pass, and the process is repeated until all frequent item sets have been enumerated. The algorithm has three main steps:

- By doing self-join on $F_{k-1,}$ generate candidates of length *k* from the frequent *(k-1)* length ofitemsets
- Prune the candidates which are having atleast one infrequent subset.
- Scan all the transaction to obtain support of candidates.

Apriori uses the hash tree for storing candidates which supports fast support counting. Here, in the hash tree, itemsets forms leaves and internal nodes are formed by hash tables(hashed by items), which directs the candidate searching.

Limitations of this algorithm

- The algorithm is heavinglow efficiency, because it requires repetitive database scan,which spends much in I/O.
- It creates a large number of 2-candidate itemsets during outputting frequent 2-itemsets.
- It doesn't exclude the useless itemsets during outputting frequent k- itemsets.
- Finally it takes more time, space and memory for candidate generation process.

Methods to Improve Apriori's Efficiency

- Hash-based itemset counting: Any of the *k*-itemset which is having hashing bucket count less than the threshold is not frequent.
- Transaction reduction: Reduce transaction scan that does not contain any frequent k-itemset is useless for next scans.
- Partitioning: Any potential frequent itemset of database must be frequent in atleast one of the partitions of database.
- Sampling: It is a process of mining particularly on given data subset. Which lowers support threshold and is a method to determine the completeness.
- Dynamic itemset counting:Adding a new candidate itemset, only after the estimation of their subsets to be frequent.

Dynamic itemset counting(DIC):

Generalization of Apriori called asDIC [5] algorithm proposed by Sergey Brin [6]. Basically it's an extension of Apriori algorithm which decreases the number of scans on the dataset. DIC Algorithm is strictly separate counting and generating candidates[7].

(Some heading should be here to show efficiency mentioned by points)

- Alternative to Apriori Itemset Generation
- Itemsets are dynamically added and deleted as transactions are read.
- Relies on the fact that for an itemset to be frequent, all of subsets also be frequent, so we have to examine only those itemsets whose subsets are all frequent.

A dynamic itemset counting technique was proposed in which the database is divided into blocks marked by start points. In this variation, new candidate itemsets can be added at any point of scan, unlike in Apriori, which determines new candidate itemsets only immediately prior to each complete database scan. This is one of the dynamictechniques which estimates support of all of the itemsets, counted so far and adds new candidates& all of its subsets are estimated to be frequent.

Here, Itemsets marked in four different ways as they are counted:

- Solid box: confirmed frequent itemset - an itemset which has finished counting and exceeds the support threshold minsupp
- Solid circle: confirmed infrequent itemset – which has finished counting and it is below minsupp
- Dashed box: suspected frequent itemset - an itemset which are still counting that exceeds minsupp
- Dashed circle: suspected infrequent itemset - an itemset which are still counting that is below minsupp

FP-Growth:

FP-growth from JiaweiHan's[8] research group at Simon Fraser University, is the algorithm , which propose generation of frequent itemsets for generating association rules. The final version of this algorithm was released on February 5, 2001. This was an improvement to earlier versions available.

FP Tree algorithm is used to find the frequent itemset without the candidate itemset generation.

Two step approach:
Step 1: Build a compact data structure called the FP-tree
Step 2: Extracts frequent itemsets directly from the FP-tree

FP-Tree Construction
FP-Tree is constructed byusing 2 passes over the data-set:

Pass 1:
(i)Scan data and find support for each item.
(ii)Discard infrequent items.
(iii) Sort frequent items in decreasing order based on their support. Use this order while building the FP-Tree, so common prefixes can be shared.

Pass 2:
Nodes correspond to items and have a counter
(i) FP-Growth reads 1st transaction at a time and maps it to a path.
(ii)Fixed order is used, so paths can overlap when transactions share items In this case, counters are incremented
(iii)Pointers are maintained between nodes containing the same item, creating singly linked lists (dotted lines)
(iv)Frequent item sets extracted from the FP-Tree.

Frequent Itemset Generation
(i)Each prefix path sub-tree is processed recursively to extract the frequent itemsets. Solutions are then merged.
Conditional FP-Tree
The FP-Tree that would be built if we only consider transactions containing a particular itemset (and then removing that itemset from all transactions).

Advantages of FP-Growth
–Takes only two passes over data-set
– "Compresses" the data-set
– No candidate generation
– Much faster than the Apriori

Disadvantages of FP-Growth
– FP-Tree may not be able to fit in memory
– FP-Tree is expensive to build

For storing the database in compressed manner, FP-growth (frequent pattern growth) uses a special structure which is called extended prefix-tree (FP-tree).This technique follows divide-and-conquer approach for decomposing the mining tasks and database and use of pattern fragment growth technique to have relief from costly candidate generation and testing, which is used by Apriori approach.

FP-Growthperforms in following steps:

- Compress a large database into a compact, Frequent-Pattern tree (FP-tree) structure
- A divide-and-conquer methodology: decompose mining tasks into smaller ones
- Avoid candidate generation: sub-database test done.

```
Input: FP-tree
Method: Call FP-growth (FP-tree, null).
ProcedureFP-growth (Tree, α)
{
1) if(single path P) do
2) for each combination= minimum support of nodes in β.
3) Else For each header aido
{
5) Construct β.s conditional pattern base and then β.s conditional FP-tree Tree β
6) If Tree β = null
7) Then call FP-growth (Tree β, β)}
}
    Output: complete set of frequent patterns
```

Figure 2:FP Tree algorithm pseudo code [9]

III. LITERATURE SURVEY

(Tannuet al 2011)Provided the detailed study on two different type of association rule mining algorithm which are DIC and FP Growth, and proposednew Dynamic FP which contains the best features of both thealgorithms. They performed their experiment on a single dataset and carryout the performance analysis by varying confidence level and evaluated computational time. Concluded that

Dynamic FP algorithm gives the better performance for large dataset than the DIC and FP-Growth.[10]

Dominic and Abdullah dealt with FP-growth's Variation algorithms[11]. The paper considers to the "Classic" frequent itemsets problem, which is the mining of all frequent itemsets that exist in market basket-like data with respect to support thresholds. The execution time and the memory usage were recorded to see which algorithm is the best one. For the time consumption AFOPT(A Frequent Pattern Tree) algorithm took advantage for most of the data set even though it suffers from segmentation fault in the low support values on connect4 data set.

AFOPT Algorithm

Liu et al[12] investigated the algorithmic performancespace of the FP-growth algorithm. AFOPT algorithm uses

dynamic ascending frequency order for both the searchspace exploration and prefix-tree construction, it uses the

top-down traversal strategy. AFOPT algorithm utilizesdynamic ascending frequency for the item search space

,adaptive representation for the conditional databaseformat, physical construction for the conditional database

construction, and top-down traversal strategy for the treetraversal. The dynamic ascending frequency search ordercan make the subsequent conditional databases shrinkrapidly. As a result, it is useful to use the physical

construction strategy with the dynamic ascendingfrequency order.

NONORDFP Algorithm

The computational time and space complexity of the FPGrowthalgorithm was improved by NONORDFP

algorithm. Racz[13] proposed thecompact representation ofFP-tree which allows faster allocation, traversal, and optionallyprojection. It contains less administrative informationabout the items in the database and allows more recursivesteps to be carried out on the same data structure, without rebuild it.

FP-Growth Algorithm*

Grahne et al [14], found that 80% of CPU was used fortraversing the FP-trees. FP Growth* algorithm uses FP-treedata structure in combination with the array-based andincorporates various optimization techniques. Array-basedtechnique is used to reduce the traversal time of FP-tree. Itreduces the memory consumption compared to FP-growthAlgorithm.

Broglet's FP-Growth

Improvement in FP-Growth algorithm can be done by Broglet'sFPgrowthalgorithm[14] . It scans the frequencies ofthe items and all infrequent items, then all items that'sappear in transactions which are heaving minimum number specified by user are discarded from the transactions, which could be never be a part of frequent item set.Computational cost can be reduced by theitems in each transaction are sorted, so that they are indescending order with respect to their frequency in thedatabase.*G. DynFP-growth Algorithm*

The main drawback of the Aprioi-like methods is at thecandidate set generation and test. This problem was

considered by introducing a novel, compact datastructure, named frequent pattern tree[16], or FP-tree, is

not unique for the same logical database. This approachcan provide a very quick response to any queries even on

databases that are being continuously updated. Here in the dynamic reordering process, Gyorodi C et al [15]

proposedthe original structuresmodification, In which the linking of tree nodes to header & addition to a master table with same header is done by replacing the single linked list with the duly linked list.

Enhanced FP-Growth Algorithm

Enhanced-FP[16], which works without any prefixtree and any other complex data structure& processes thetransactions directly, so main strength it's is simplicity.It initially scans the supports of the items and is calculated.The items whose support count is less than minimumsupport are discarded and identify as infrequent items.Then the database itemsare sorted in ascendingorder with respect to their support. And the initialtransaction ofdatabase is converted in to a set of transactionlist, with one list for each item. These lists are stored inarray, each of which contains a pointer to the head of thelist. And the Transaction lists are traversed from left toright for searching all the frequent item set that contain theitem which corresponds to the list. Before a transaction list isprocessed, its support count is checked, if it exceeds thanminimum support count than there must be a frequent itemset.

IV. EVALUATION AND RESULTS

Experiment carried out with the machine heaving dual Core 1GHz processors and 2 GB of RAM and Windows7 was the operating system. For making the time measurement more reliable, no other process or an application was running at the same time on the machine. Above mentioned all the algorithms were evaluated on the single system.

The Apriori and FP-Growth association rule mining algorithms were tested in WEKA Data Mining tool of version 3.6.1. WEKA Data Mining Tool is a collection of open source of many data mining and machine learning algorithms, including pre-processing on data, Classification, clustering and association rule extraction. And DIC Algorithm was tested in NetbeansIDE 6.8.The performance of Apriori, FP-Growth and DIC were evaluated based on execution time. Here, the execution time is measured for different number of instances and Confidence level on Different datasets. We have analyzed algorithms for different datasets.

To evaluate the performance of the association rule mining algorithmswe performed an experiment on two different data sets one is of supermarket[18] and the another one isSPECT heart[19] data taken from UCI Repository.Data set Information is shown below.

TABLE 1. INFORMATION OF DATASET

Dataset Name	Data type	Instances	Attributes
Super market	Numerical	4627	217
Heart Spect	Binary	267	22

First dataset Super Market uses for evaluate Apriori and FP-Growth Algorithms and performed experiment on weka and Tanagra tools.Wehave imported the data set in ARFF format.

Second dataset Heart Diagnosis uses to evaluate Apriori and DIC Algorithmand performed experiment on Tanagra and netbeans tools.We have imported the data set in binary data format in text file.

Association rule mining algorithm is design to improve efficiency by some parametersas mentioned below.

- Number of Scan:
  Number of database scan is affecting the memory I/O operation. If number of scan is less then algorithm is provide better efficiency.
- DataStructure:
  Used to store frequent item and their count during execution of algorithm.Available data structure aretrie tree,FP-Tree prefix tree and Hash tree.
- Data Layout:
  Data Layout are two type:vertical and horizontal. Algorithm uses that layout for scanning database.
- Optimization Techniques:
  Many algorithm is use to improve efficiency optimize by any technique like FP-Growth is used FP-Tree to improve efficiency.DIC algorithm which reducingnumber of database scan.

According to Survey and evaluation of an algorithm done.Here, we have provided some features of various algorithms as shown in table 2.

TABLE 2.FEATURES OF VARIOUS ALGORITHMS

Algorithm	Scan	Data Structure	Database Layout	Optimizations
Apriori	K	Hash Tree	Horizontal	C_2 array
FP-Growth	2	FP-Tree	Horizontal	Data structure, Does not generate Catedate Set
DIC	<=k	Prefix Tree	Horizontal	Count multiple lengths per scan

Here, chart shown in Figure 2 and 3 are the performance chart of algorithms that required time in execution with minimumsupports ranging from 10% to 90%. Those three algorithmsthat generates frequent item sets, namely Apriori, FP-growth and DIC. The first chart shown by figure 2 indicatesthat FP-Growth is faster than Apriori& the second chart shown by figure 3indicates that DIC is faster than Apriori. But in DIC case, lower minimum supports apriori is better than DIC as shown in figure 3 at min support of 10%. DIC takes more execution time than Apriori. In second case DIC takes less execution time than Apriori.(need to check statement because only Apriori is common in both comparisons)

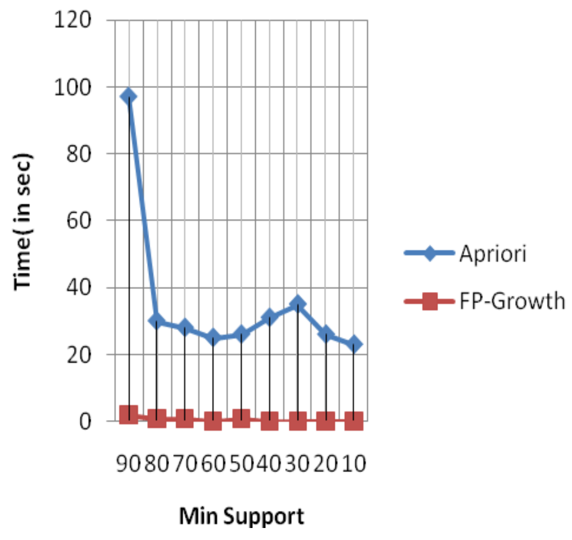

Fig 2 : Excuction time of Apriori And FP-Growth

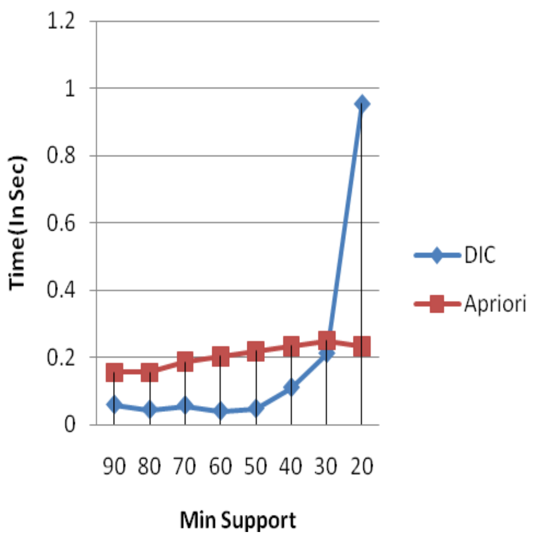

Fig 3 : Excuction time of DIC and Apriori

V.CONCLUSION

The association rules play a vital role in many data mining applications, trying to find impressive patterns in databases. This paper provides an overview of three different association rule mining algorithms like Apriori, FP-Growth and DIC algorithms. The performance analysis was done by varying number of instances and confidence level. The efficiency of all algorithms is defined based upon time required to generate the association rules. From the experimental data presented,it could be concluded that the DIC algorithm and FP-Growth behaves better than the Apriori algorithm based on execution time required to generate rules.

VI.FUTURE WORK

As a future work, we can focus on Mining of association rules in distributed environment, which could be helpful in reducing the memory input/output operation and communication overhead between sites.

REFERENCE

[1].RakeshAgrawal and RamakrishnanSrikant, 1994. “Fast Algorithms for Mining Association Rules”, In Proceedings of the 20th Int. Conf. Very Large Data Bases, pp. 487-499.

[2]R.Agrawal, T.Imielinski, and A.Swami, 1993. “Mining association rules between sets of items in

large databases", in proceedings of the ACM SIGMOD Int'l Conf. on Management of data, pp. 207-216.

[3]JyotiArora, NidhiBhalla and SanjeevRao,"A Review On Association Rule Mining Algorithms",*International Journal of Innovative Research in Computer and Communication Engineering*,Vol. 1, Issue 5,pp1246-1251 July 2013

[4]JochenHipp, Ulrich Guntzer and GholamrezaNakhaeizadeh,"Algorithms for Association Rule Mining – A General Survey and Comparison",*sigkdd Explorations*,vol2 pp58-64,july2000

[5]Mohammed J. Zaki.,"Parallel and Distributed Association Mining:A Survey" ," *IEEE Concurrency*",pp14-25, October–December 1999

[6]S. Brin, R. Motwani, J.D. Ullman, and S. Tsur, 1997. "Dynamic itemset counting and implication rules for market basket data".In Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, vol. 26(2), pp. 255–264.

[7]JochenHipp, Ulrich Guntzer and GholamrezaNakhaeizadeh,"Algorithms for Association Rule Mining – A General Survey and Comparison",*sigkdd Explorations*,vol2 pp58-64,july2000

[8]Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao, 2004. "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", Data Mining and Knowledge Discovery, vol. 8, issue 1, pp. 53 – 87.

[9]]PROXY DRIVEN FP GROWTH BASED PREFETCHING

[10] TannuArora, and Rahul Yadav,"Improved Association Mining Algorithm for Large Dataset",*International Journal of Computational Engineering & Management*,vol13,pp33-39,July2011

[11]Grahne G. and Zhu J., "Efficiently Using Prefix-Trees in miningFrequent Item sets," Proc. ICDM 2003Workshop Frequent Item setMining Implementations, (2003).

[12] Liu,G. , Lu ,H. , Yu ,J. X., Wang, W., & Xiao, X.. "AFOPT:AnEfficient Implementation of Pattern Growth Approach", In Proc.IEEE ICDM'03 Workshop FIMI'03, 2003.

[13] BalazesRacz," nonordfp: An FP-Growth Variation withoutRebuilding the FP-Tree**",** 2nd Int'l Workshop on Frequent ItemsetMining Implementations FIMI2004

[14] Grahne O. and Zhu J. "Efficiently Using Prefix-trees in MiningFrequent Itemsets", In Proc. of the IEEE ICDM Workshop onFrequent Itemset Mining, 2004.

[15]Cornelia Gyorodi, Robert Gyorodi, T. Cofeey& S. Holban –"Mining association rules using Dynamic FP-trees" – înProceedings of The Irish Signal and Systems Conference,University of Limerick, Limerick, Ireland, 30th June-2nd July2003, ISBN 0-9542973-1-8, pag. 76-82.

[16]Luca Cagliero and Paolo Garza "Infrequent Weighted ItemsetMining using Frequent Pattern Growth", IEEE Transactions onKnowledge and Data Engineering, pp. 1- 14, 2013.

[17].A.M. Said, P.D.D. Dominic and A. B. Abdullah ,(2009) "A Comparative Study of FP-growth Variations", International Journal of Computer Science and Network Security(IJCSNS), vol.9, issue.5,pp 266-272,2009.

[18]http://archive.ics.uci.edu/ml/datasets/SPECT+Heart

[19]http://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/supermarket.arff