An Experimental Study of Pattern Mining Technique to improve the Business Strategy

S.Megala
Research Scholar (C.S),
Karpagam University,
Coimbatore,
Tamilnadu – 641021.

Dr.M.Hemalatha
Head, Software System,
Karpagam University
Coimbatore,
Tamilnadu – 641021.

Dr.T.Christopher
Head, Computer Science,
Government Arts College,
Udumalpet,
Tamilnadu-642126.

P.Soundar Rajan
Head/EEE Dept,
Professional Group of Institutions , Tirupur
Tamilnadu-641662

ABSTRACT
Pragmatism of pattern mining system is to study the data and determines a model that is closets to characteristics of the data being examine. This necessitates identifying interesting association patterns idea can be described as a recursive eradication method in a preprocessing step, that remove all items from the transactions that are not regular individually, i.e., do not appear in a user-specified least number of transactions. Then choose all transactions that have the least regular item (least frequent along with those that are frequent) and delete this item from them. Recursive to procedure the obtained reduced (also known as projected) database, evoke that the entry sets construct in the recursion split the deleted item as a prefix. On revisit, eliminate the processed item also from the folder of all transactions and start over, process the second frequent item etc. In these dispensation steps the prefix tree, which is improved by associations between the branches, is demoralized to quickly discover the transactions containing a specific entry and also to eliminate this entry starting the business after it has been processed.

Keywords
Data mining, Association Rule, FP-growth, Frequent Pattern, Prefix Tree

1. INTRODUCTION
Frequent entry position mining is solitary of the majority significant and general issue of revision designed for association rule mining in data mining research area. While an association rule mining is distinct as the relation among diverse item sets. Association rule mining takes part in pattern discovery techniques in knowledge discovery and data mining (KDD). As performance of association rule mining is depends upon the frequent item sets mining, thus is essential to mine frequent item set resourcefully.

Association Rule Mining and formative the correlation of the matter near in the transaction report be the most main points of spotlight with regard to Data Mining on transactional databases. Discovering interesting association rules, which are used to convalesce the business helpfulness of an activity, have extended been predictable in data mining part. This necessitates identifying interesting association patterns that are both statistically and semantically important to the business utility.

Even still quite a lot of algorithms are existing in the literature for association rule mining, [1] a good number of them deal with efficient implementations rather than the production of effective rules [2]. The techniques to facilitate aid in the confiscation of suitable and valid association patterns are frequently quantitative in nature [3]. In tidy to absolutely manage the inherent quality of a dataset the qualitative attributes are decisive. In the Mining techniques effort is focused on normal item set in an existent time database. The majority significant target of our loom is to categorize a recursive removal method in a preprocessing step that delete all items from the transaction to facilitate are not frequent individually, i.e., do not show in a user-specified least number of contact. The planned approach to decide on all business that contains the least frequent item (least frequent among those that are frequent) and delete this item from them. Recursive to process the obtained reduced (also known as projected) database, identification with the purpose of the item sets originate in the recursion distribute the deleted item as a prefix. In these dispensation steps the prefix tree, which is enhanced by associations flanked by the branches, is demoralized to quickly discover the communication containing a given item and also to remove this item from the transactions after it has been process.

The rest of the paper is organized as follows: Section 2 present a brief review of association rule mining. The FP Tree mining in a real time system is discussed in detail in Section 3.The investigational outcome are accessible in Section 4 and conclusions are summed up in Section 5.

2. RELATED WORK
A related of earlier works in the field of Association Rule Mining inspiration behind the future methodology. Charu C. Aggarwal et al. [4] have provided a survey of research on association rule generation. They have discussed a number of variations of the association rule problem which have been projected in the literature and their convenient applications. Cai et al., [5] have proposed a technique of mining weighted association rule. They have provided two different definition of weighted support: without normalization and with normalization and proposed a new algorithm based on the support bounds.

Chuan Wang and Christos Tjortjis [6] have proposed an efficient algorithm for mining association rules, which first identifies all large itemsets and then generates association rules. Their advance has reduced large itemset production time, known to be the mainly protracted step by scanning the database only

once and using logical operations in the process. Cornelia Gyorodi, Robert Gyorodi and Stefan Holban have proposed a comparative study on association rule mining algorithm. The compared algorithm is presented together with a experimental data.

C. Borgelt [7] gave a C implementation of the FP-Growth algorithm, by means of his individual specialized memory allocation supervision module. The initial FP-tree is built as a simple list of integer arrays. This list is sorted lexicographically and can be turned into an FP-tree with a recursive procedure. The proposed 2 projecting approaches do not need parent-too-child pointers, so the structure of tree node can be more compact. Despite this implementation's technical merits, its C coding style and complex data structures make it not easy to be used as educational purpose or fast application prototype building.

D.Sujatha, B.L.Deekshatulu[8] have analyze the past transaction data to discover customer behaviors such that quality of business decision can be improved. The Input of Association mining is large set of transactions each consisting of a list of items a customer has purchased at a supermarket. The list of frequent itemsets that may change in time as the purchasing habits get affected by season, fashion and introduction of new products. The output of this algorithm is to get the frequent items which occur after a particular item or item sets.

Girish K. Palshikar et al. [9] have proposed the concept of heavy itemset, which compactly represents an exponential number of rules for an efficient theoretical characterization of a heavy item set. It also presented an efficient covetous algorithm to generate an assortment of disjoint heavy itemsets in a given transaction database.

3. FREQUENT ITEMSET MINING IN ASSOCIATION RULE

In this section the principal objective of the research is to find association pattern from the transaction data item of a enterprise to improve the business. An FP-tree is fundamentally a prefix tree for the production. That is, every corridor represents a set of transactions that divide the same prefix; each node corresponds to one item. In addition, all nodes referring to the corresponding item are associated equally in a list, so that all statement containing an explicit item can simply be found and counted by traversing this list. The list can be accessed during a head element, which besides states the entire number of occurrences of the item in the database.

The sorted record can easily be bowed into an FP-tree with a straightforward recursive method: at recursion depth k, the k-th entry in the entire operation is used to divide the record into sections, one for each item [10,11]. For every operation a node of the FP-tree is formed and labeled with the item corresponding to the section. Every segment is then processed recursively, split into subsections, a new layer of nodes (one per subsection) is created etc. While single individual operation is route at a time, barely the FP-tree depiction along with one new operation are in core memory. This generally saves space, because an FP-tree is regularly a much more packed together illustration of an maneuver database. Persistent drawbacks that when inserting connections hooked on the FP-tree, one such directory has to be searched for every item of the operation in regulate to get the child to go to a possibly fairly costly operation.

In distinguish to this, initial loading the transaction record as a easy list of integer arrays, sorting it, and construction the FP-tree with a recursive, makes it achievable to do without parent-to-child pointers exclusively. Beginning the instant while the FP-tree is built pinnacle down, the parent is formerly renowned when the children are formed. Hence it can be standard along in the recursion, where the close relative pointers of the offspring are set straight. If two interactions distribute a familiar prefix, according to some sorted group of frequent items, the mutual parts can be grouping using one prefix structure as long as the *count* is registered properly. If the persistent items are sorted within their *occurrence downward order*, there are well all over again probability that more prefix strings can be mutual [12,13]. Among the beyond observations, one may erect a frequent-pattern tree technique and they felt that this procedure will be useful to sponsor their business in coordination with associated product manufactures[14].

First, a scrutinize of *DB* derives a *list* of frequent items, (Sintex tank :4), (pump:4), (heater:3), (toilet fitting:3), (wash basin:3), (shower:3) in which objects are structured in frequency descending order. This ordering is main seeing as each conduit of a tree will follow this order.

Second, the source of a tree is formed and labeled with "*null*". The FP-tree is constructed as follows by scanning the business database. The scan of the first transaction leads to the construction of the first branch (Fig.1) of the tree (Sintex tank:1), (pump:1),after reading the TID2 it construct the second branch with (Pump:1),(Heater:1),(Wash Basin:1) as shown below. Likewise the remaining TID are created and finally FP-Tree is constructed.

TID	Items
1	{Sintextank,pump}
2	{Pump,Heater,wash basin}
3	{Sintextank, Heater,wash basin,Shower}
4	{Sintextank,wash basin,Shower}
5	{ Sintextank,Pump,Heater,wash basin}
6	{ Sintextank,Pump,Heater,wash basin}
7	{Pump,Heater}
8	{Sintextank,pump Heater,}
9	{Sintextank,pump, wash basin }
10	{Pump,Heater,Shower}

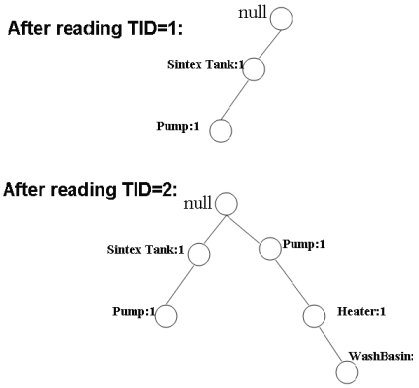

Fig.1.FP-Tree Construction for TID

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Transaction Database

Header table

Item	Pointer
A	SintexTank
B	Pump
C	Heater
D	Wash Basin
E	Shower

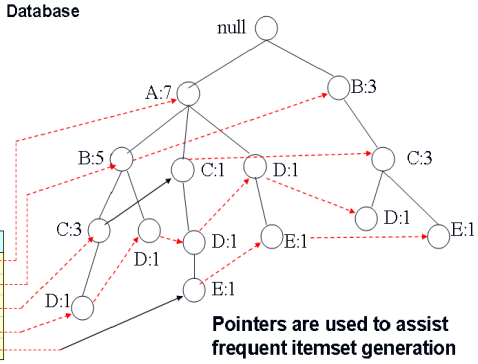

Fig.2 Frequent-Pattern Tree construction for DB

A Frequent-Pattern Tree (FP-tree in short) is a tree construction define below

1. It consists of one root label as "null", a position of item prefix junior trees as the offspring of the root, and a frequent-item-header table.
2. Each join in the item-prefix subtree consists of three fields: item-name, add up, and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree[15].
3. Every admission in the frequent-item-header table consists of two fields, (1) item name and (2) head of node-link (a pointer pointing to the first node in the FP tree carrying the item-name)

Algorithm of (FP-tree constructio

Input: A transaction database *DB* and a minimum support threshold ξ.

Output: FP-tree, the frequent-pattern tree of *DB*.

Method: The FP-tree is constructed as follows.

1. Scan the transaction database DB once. Collect F, the set of frequent items, and the support of each frequent item. Sort F in support-descending order as F List, the list of frequent items.

2. Create the root of an FP-tree, T, and label it as "null". For each transaction Trans in DB do the following.

Select the frequent items in *Trans and* sort them according to the order of *FList*. Let the sorted frequent-item list in *Trans* be [*p* | *P*], where *p* is the first element and *P* is the remaining list. Call insert tree ([*p* | *P*], *T)*. The function *insert tree* ([*p* | *P*], *T*) is performed as follows. If *T* has a child *N* such that N.item-name = p.item-name, then increment N's count by 1; else create a new node N, with its count initialized to 1, its parent link linked to T , and its node-link linked to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree (P, N) recursively.

4. EXPERIMENTAL RESULTS

The proposed tree construction has been designed and implemented using Visual Basic programming under windows operating system. An FP-tree node contains fields for (1) an item identifier, (2) a counter, (3) a pointer to the parent node, (4) a pointer to the successor node (referring to the same item) and (5) an auxiliary pointer that is used for a frequent itemset in the business strategy.

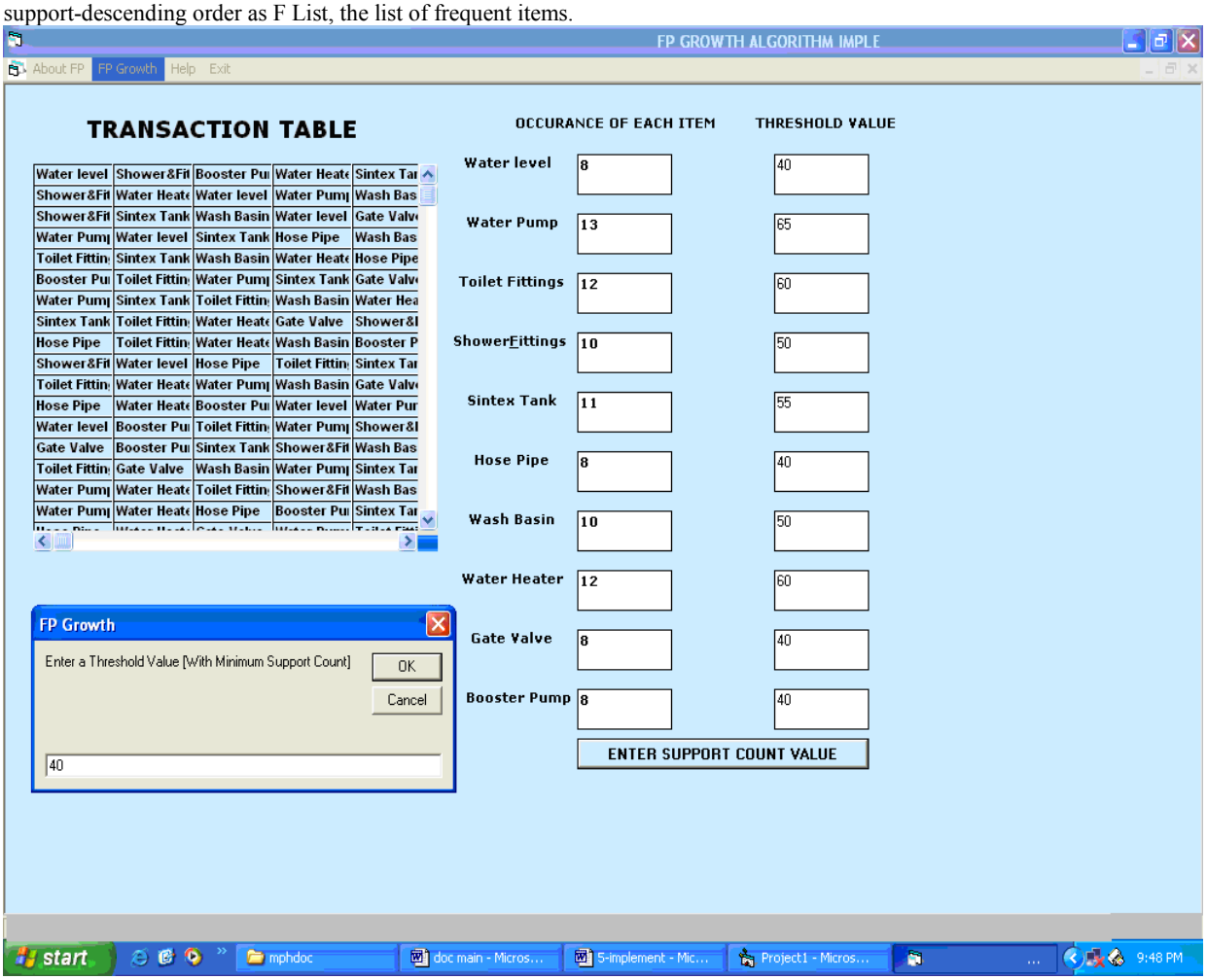

Fig.3 Transaction of Each Item

The transaction table contains the list of records, which are the itemsets purchased by the customer in a single visit. The Fig.3 shows the first step of the process. The occurrences of each itemsets in the transaction and threshold value of an item are found by using the minimum user defined support count.

$$\text{Support count } (X \rightarrow Y) = (X \cup Y) / N$$

Where X and Y are items and N is total number of transactions in the database. The fig.4 shows the second step of the process in which the uninterested items are eliminated based on the user defined support value in the each transaction in the data base. A new database is formed with the interestingness items. The interestingness items are said to be frequent items and are more needed for business analysis.

The next step process is to sort the frequent items in descending order in the each transaction based on the support value of the item. This process will minimize the complicity of the tree construction. The goal of this stage is to build the efficient pattern mining tree. After the sorting process is over the frequent pattern tree is constructed as shown in Fig.5. The tree construction started with full I/O scan of the dataset. The root node of a tree is created and labeled with "null". Scan the DB the second time for building the tree. The scan of the first transactions leads to the construction of the first level of the tree which are child's of root node of the tree. To facilitate tree traversal, an item header table is built, in which each item points, via a head of node-link, to its first occurrence in the tree. Nodes with the same item-name are linked in sequence via such node-links.

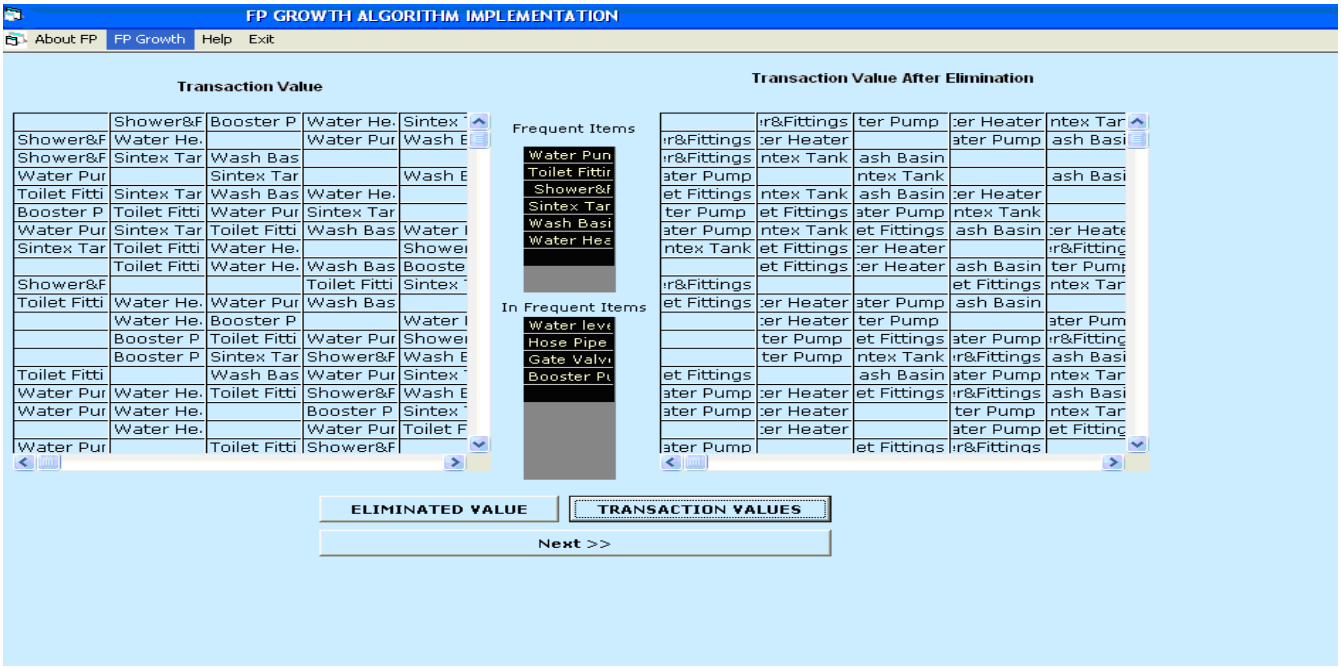

Fig .4 Frequent and Discarded Items

In this study, the frequent pattern mining approach is applied in real time marketing domain. The mining of the profitable itemsets is obtained by the scanning the frequent pattern tree for N most interesting itemsets.

For example, the first level frequent items in the tree are Water_heater, Water_Pump, Sintex_tank, and Toilet_fittings. With the initial frequent pattern tree, the frequent item water heater has linked with the item sintex_tank in two paths. The first path is water_heater(4) → sintex_tank(1) → toilet_ fittings(3) and second path is water_heater(1) → sintex_ tank(1) → shower&fittings(1)

The frequent item water_heater linked with the item toilet_fittings in two ways. (i).water_heater (1) → toilet_ fittings(1) → sintex_tank(1) → wash_basin(1).

(ii).water_
heater(1) → toilet_fittings(1) → sintex_tank(1) → shower&Fitt ings(1).

The frequent item sintex _tank have one path: Sintex_tank (2) → Wash_basin(2) and Sintex_tank(2) → Wash_basin(2) → Shower&Fittings(2)

The item toilet_fittings have the following branch: toilet_fittings(1) → sintex_tank(1), and toilet_fittings(1) → sintex_tank(1) → shower &fittings(1)

The item water_pump have linked with the items toilet_fillting and water_heater as in the following ways: toilet _fitting (4) → water_heater(8),
water_pump(13) → toilet_fittings(4) → shower&Fittings(2),
water_pump(13) → water_heater(4) → shower&Fittings(1),
water_pump(13) → water_heater(4) → wash_basin(1) → show er&Fittings(1),
water_pump(13) → water_heater(4) → sintex_tank(1),
water_pump(13) → water_
heater(4) → toilet_fittings(4) → wash-basin(2) → shower& fittings(1).

This frequent pattern tree construction takes two scanning of the database. Since the items in the most frequent itemsets are ordered with respect to support value of each time, the itemsets are gets the focus in the real time market analysis. The patterns derived from the conditional tree can be used to improve the business strategy.

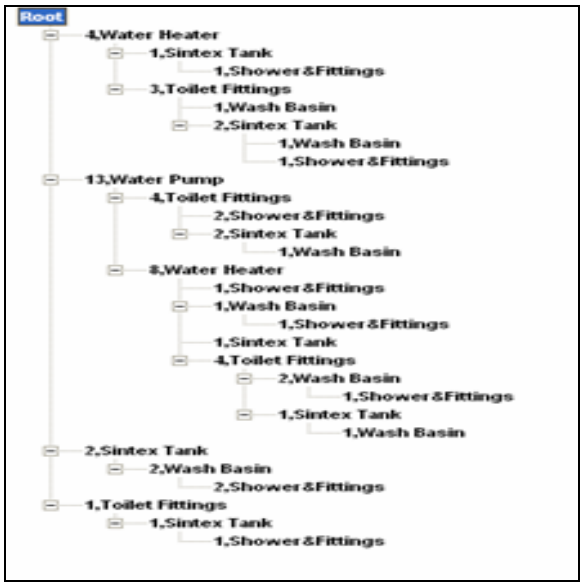

Fig:5 Tree Specification

5. CONCLUSION

Association rule have been widely used to determine customer buying pattern from market basket data. Discovering interesting association rules, used to improve business utility of an enterprise has long been recognized in data mining community. The basic idea of this paper can described as recursive eradication design in a preprocessing step and prune all items from the transactions that are not commonly occur, i.e., do not materialize in a user-specified least number of business. Then select all communication that hold the least repeated item and delete this entry from them. Eradicate the processed item also from the list of all transactions and start over, i.e., process the second common thing etc. In these dispensation steps the prefix tree, which is superior by links between the undergrowth, is demoralized to quickly stumble on the transactions containing a known entry and also to get rid of this item from the business after it has been process.

6. REFERENCES

[1]Jiawei Han and Micheline Kamber. Data Mining: Concepts and Techniques.Morgan Kaufmann Publishers, 2001.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In Proceedings of the 20th International Conference on Very Large Data Bases, pages 487–499. Morgan Kaufmann Publishers Inc., 1994.

[3] Herb Edelsten. Mining large databases - a case study. Technical report.

[4] C.C.Aggrawal and P.S.Yu. “Mining large itemsets for association rules”. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 21(1): 23-31, March 1998.

[5] Cai, C.H., Fu, A.W-C., Cheng, C. H., Kwong, W.W. “Mining Association Rules with Weighted Items”. In: Proceedings of 1998 Intl. Database Engineering and Applications Symposium (IDEAS'98), pages 68--77, Cardiff, Wales, UK, July 1998.

[6] Wang, C., Tjortjis, C., “PRICES: An Efficient Algorithm for Mining Association Rules,” Lecture Notes in Computer Science, Volume 3177, Jan 2004, Pages 352 – 358.

[7] C. Borgelt. “An implementation of the FP-growth algorithm”. In Proceeding of OSDM 2005, pp.1-5, 2005.

[8] D.Sujatha, B.L.Deekshatulu,”Agorithm for mining time varying frequent itemsets” Journal of Theoretical and Applied Information Technology,2009Vol 6 (2) pp165-170.

[9] GK Palshikar, MS Kale, MM Apte, “Association rules mining using heavy itemsets”, Data and Knowledge Engineering, Vol. 61, No. 1, pp. 93-113, 2007.

[10] Agarwal, C. Agarwal, and V.V.V.Prasad, “ATree Projection Algorithm for Generation of Itemsets,”Journal on Puraflel Distributed Computing, 2000, Vol.61, pp, 350.

[11] Leung, C. K.-S., and Khan, Q. I. 2006. DSTree: A treestructure for the mining of frequent sets from data streams. InProc. of the 6th Int. Conf. on Data Mining (ICDM). 928-932.

[12] Tanbeer, S. K., Ahmed, C. F., Jeong, B.-S., and Lee, Y.-K2008. CP-tree: a tree structure for single-pass frequent pattern mining. In Proc. of PAKDD, Lect Notes Artif Int, 1022-1027.

[13] Lilin FAN, “Research on Classification Mining Method of Frequent Itemset”, JCIT: Journal of Convergence Information Technology, vol. 5, no. 8, pp. 71-77, 2010.

[14] Jyothi Pillai, O.P.Vyas, "Overview of Itemset Utility Mining and its Applications",International Journal of Computer Applications, Vol: 5, No.11, August 2010.

[15] Bac Le, Huy Nguyen, Tung Anh Cao, Bay Vo, "A Novel Algorithm for Mining High Utility Itemsets", *First Asian Conference on Intelligent Information and Database Systems*, Dong Hoi, pp: 13 - 17, 2009.