iJET EXPLORE International Journal of Engineering & Technology
CENTRE OF PROFESSIONAL RESEARCH PUBLICATIONS
ISSN 2049-3444

International Journal of Engineering and Technology Volume 4 No. 3, March, 2014

Implementation of Efficient Algorithm for Mining High Utility Itemsets in Distributed and Dynamic Database

G. Saranya[1], A.Deepakkumar[2]
[1]M.E Student, Computer Science and Engineering, Sathyabama University, Chennai.
[2]Assistant professor, Computer Science and Engineering Department, St.Joseph's Institute of Technology, Chennai.

ABSTRACT

Association Rule Mining (ARM) is finding out the frequent itemsets or patterns among the existing items from the given database. High Utility Pattern Mining has become the recent research with respect to data mining. The proposed work is High Utility Pattern for distributed and dynamic database. The traditional method of mining frequent itemset mining embrace that the data is astride and sedentary, which impose extreme communication overhead when the data is distributed, and they waste calculation resources when the data is dynamic. To overcome this, Utility Pattern Mining Algorithm is proposed, in which itemsets are maintained in a tree based data structure, called as Utility Pattern Tree, and it generates the itemset without stare the entire database, and has sparse communication overhead when mining with respect to distributed and dynamic databases. A quick update incremental algorithm is used which scans only the incremental database as well as collects only the support count of newly generated frequent itemsets. Incremental Mining Algorithm not only includes new itemset into a tree but also discard the infrequent itemset from a utility pattern tree structure. Hence it provides faster execution, minimal communication and cost when compared to the existing methods.

Keywords- *Association Rule Mining, High Utility Pattern Mining, Distributed and Dynamic Database, Incremental pattern Mining.*

1. INTRODUCTION

Data mining [1] [2] is the process of some extracts information from very large datasets. Association Rule Mining (ARM) [3], [4], [5] is an important data mining technique which is used to discover the trends/patterns among items in a large database. The goal of ARM is to identify group of items that occur together, for example in a market basket analysis [6]. A transaction database consists of two aspects such as internal and external utility. Quantity of a product in a particular transaction is called the internal utility and the profit value of a product is called external utility. The utility of itemset is defined as the product of external utility and internal utility. The high utility itemset is that the itemset with a utility not less than a user specified minimum support threshold value; else that itemset is treated as unassuming utility itemset.

Incremental algorithm is a process where only the incremental part will be mined further for discovering frequent patterns. Main goal of incremental mining algorithm is that, the data is regularly added to the original transactions databse, hence the database size becomes larger and mining the entire lot would take high computation time, hence it would be better if we mine only the updated portion. Hence it can provide faster execution when compared to the existing methods.

The Remainder of this paper is organized as follows. Related researches are reviewed in section 2. Comparison of proposed mechanism with existing approaches in section 3. Proposed methods are described in section 4. Results and the discussion are quoted in section 5. Experimental results have been discussed in section 6. Conclusions are presented finally in Section 7.

2. RELATED RESEARCHES

Association rule mining is considered to be an interesting research area and studied widely [1], [5], [7], [8], [9], [10] by many researchers. In the recent years, some relevant methods have been proposed for mining high utility itemsets from transactional databases.

Y.Liu, W-K.Liao, A. Choudhary proposed a two phase algorithm [11] which was developed to find high utility itemsets, using the download closure property of apriori. The algorithms have defined the transaction weighted utilization (twu) while maintaining the download closure property. In this paper they defined two database scans. In the first database scan, the algorithm finds all the one-element transaction-weighted utilization itemsets and its results form the basis for two element transaction weighted utilization itemsets. In the second database scan, the algorithm finds all the two element transaction-weighted utilization itemsets and it results in three element transaction weighted utilization itemsets. The drawback of this algorithm is that it suffers from level wise candidate generation and test methodology. CTU-mine [12]

proposed an algorithm that is more efficient than the two phase method only in dense databases when the minimum utility threshold is very low.

J Hu et al developed an algorithm for frequent item set mining that identify high utility item combinations [13]. The goal of this algorithm is to find segments of data, defined through combinations of some items (rules), which satisfy certain conditions as a group and maximize a predefined objective function. The high utility pattern mining problem considered is different from former approaches, as it conducts rule discovery with respect to individual attributes as well as with respect to the overall criterion for the mined set, attempting to find groups of such patterns that together contributes to the most to a predefined objective function.

Liu Jian-ping, Wang Ying, Yang Fan-ding et al proposed an algorithm called tree based incremental association rule mining algorithm (Pre-Fp) [14]. It is based on a FUFP (fast update frequent pattern) mining method. The major goal of FUFP is the re-use of previously mined frequent items while moving onto incremental mining. The advantage of FUFP is that it reduces the number of candidate set in the updating procedure. In FUFP, all links are bidirectional whereas in FP-tree, links are only unidirectional. The advantage of bidirectional is that it is easy to add, remove the child node without much reconstruction. The FUFP structure is used as a input to the pre-large tree which gives positive count difference whenever small data is added to original database. It deals with few changes in database in case of inserting new transaction. In this paper the algorithm classifies the items into three categories: frequent, infrequent and pre-large. Pre-large [15] itemsets has two supports threshold value i.e. high and low threshold.

Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu proposed an efficient algorithm for mining high utility itemsets from transactional databases [16]. In this paper, they discovered two algorithms named as UP-Growth and UP-Growth+ for mining high utility itemsets from transactional databases. In this technique they are totally dependent on the candidate length; it scans the database twice to construct the UP-Tree. They used efficient utility mining algorithm to generate huge number itemsets called potential high utility itemsets (PHUIs). In this technique they achieved a better performance than all previous high utility pattern mining techniques. However these algorithms still endure with the problem of search space, level wise candidate generation and wide memory usage.

Y-C. Li, J-S. Yeh and C-C. Chang proposed an [17] isolated item discarding strategy (IIDS). In this paper, they discovered high utility itemsets and also reduced the number of candidates in every database scan. IIDS shows that itemset share mining [18]. They retrieved efficient high utility itemsets using the mining algorithm called FUM and DCG+. In this technique they showed a better performance than all the previous high utility pattern mining technique. However, their algorithms still suffer with the problem of level wise generation and test problem of apriori and it require multiple database scans.

Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, Young-Koo Lee, Ho-Jin Choi et al proposed a Single-pass incremental and interactive mining for finding weighted frequent patterns [19]. The existing weighted frequent pattern (WFP) mining cannot be applied for incremental and interactive WFP mining and also for stream data mining because they are based on a static database and its require multiple database scans. To overcome this, they proposed two novel tree structures IWFPTWA (Incremental WFP tree based on weight ascending order) and IWFPTFD (Incremental WFP tree based on descending order) and two new algorithms IWFPWA and IWFPFD for incremental and interactive mining using a single database scan. IWFPFD ensures that any non-candidate item cannot appear before candidate items in any branch of IWFPTFD and thus speeds up the prefix tree. The main advantage of this algorithm is prefix sharing node [20] with compact structure of the tree. Numbers of nodes are less as compared to the previous method which saves memory space.

3. COMPARISON

Table 1 summarizes the comparison of proposed mechanism with existing approaches

Method	No. Of Databases Scans required	No. Of Passes Required	Approach Used	Relative CPU Utilization	Dependency on Main Memory	Parallel Processing
Two –Phase	2	Multiple	Level wise	Medium	High	No
IIDS	2	Multiple	Novel tree	Medium	High	No
HUC-PRUNE	3	Multiple	Pattern growth	High	High	No
TIARM	1	Multiple	Divide &conquer	High	High	No
Incremental Updating using Apriori- Tid List	1	Multiple	Intersection of link list	Low	High	No
PRE-FUFP	2	Multiple	Divide &conquer	Medium	High	No
Proposed Method	1	Single	Incremental Profit based	Low	Low	Yes

Most of the algorithms are mainly dependent on the main memory and it makes them low efficient. Apriori-Tid it uses like link list data structure; it may not fit into main memory because the size of database is very large in size. The other methods use UP-tree, memory requirement is too high. In HUC-Prune based algorithm it as assumption that utility pattern are fit into memory, is not possible to use very large databases.

TIARM, Pre-FUFP tree scans the database occur twice but it requires multiple passes of data processing, it increases communication overhead in terms of the time and CPU requirements. All above algorithm it prunes the irrelevant data items before creation a tree structure. It leads to the reduction of CPU requirement during processing of incremental data.

Major limitation of all previously discussed algorithms is, they do not support parallel processing as UP-tree. The proposed method can be implemented in distributed and dynamic database which increases the efficiently and efficiency.

4. PROPOSED METHOD

The proposed high utility pattern mining is a conceptual model built for distributed and dynamic database. Mining high utility itemsets from databases refers to finding the itemsets with both high profit and purchased quantities. The high utility itemset means that if its utility is no less than a user-specified minimum utility threshold, otherwise it is called a low level-utility itemset.

Administrator can enter the Specified Minimum Utility Threshold value. The time taken to complete the entire computation is called the processing time and it is the difference between the starting time and ending time of a computation.

The System employs one master node and number of slave nodes. Partition the database and send it to every slave node for computation. The slave node scans the database once and counts the occurrence of each item. These data's are stored in their local table. The entire table from the slave nodes is transmitted to the master node. The Master Node prepares a global table. The global table has the number of occurrence of the all data items in the given database. Based on the minimum utility threshold value it calculates the auspicious and inauspicious itemsets. Auspicious items are those for which the threshold values are greater than the admin specified minimum utility. Inauspicious items are those items whose threshold values are lesser than the admin specified minimum utility and hence remove the infrequent items.

Incremental Mining Algorithm is used where regularly updating goes on appearing in a database. Incremental Mining Algorithm not only added new itemset into a tree, but also removes the infrequent itemset from Utility Pattern tree structure. Finally incremental database is reconstructs and the high utility itemsets is discovered.

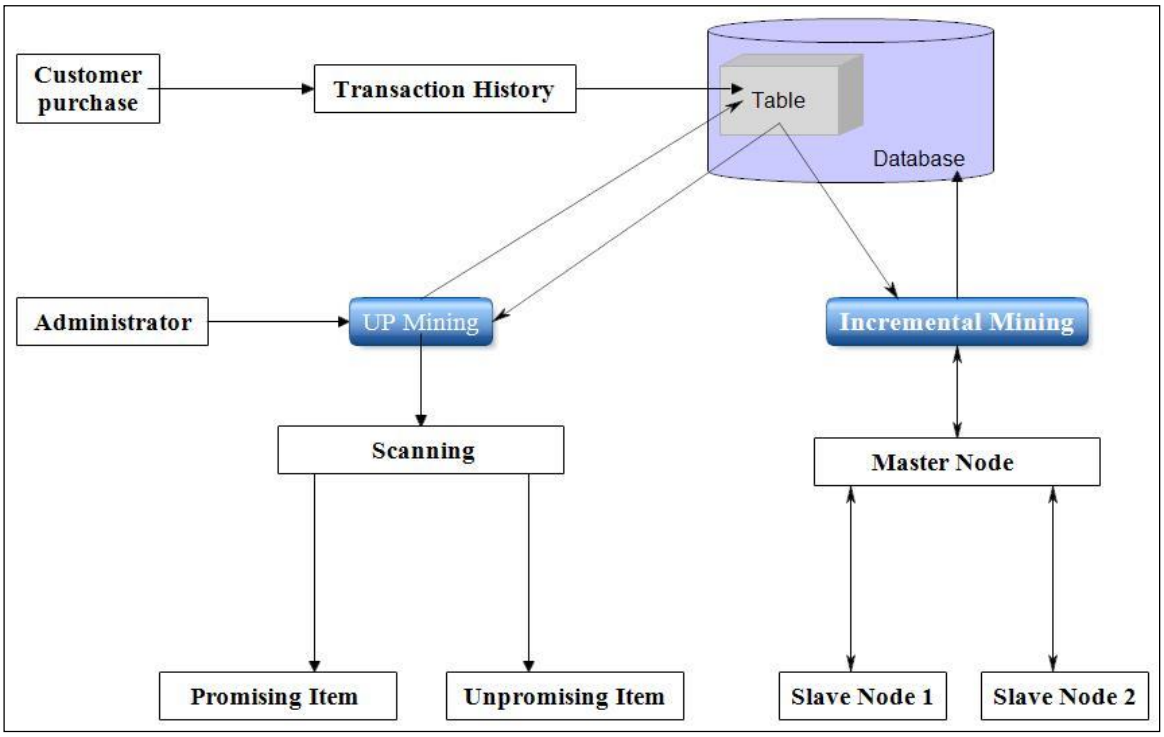

Figure1: Proposed Method

4.1 The Proposed Algorithm Works in The Following Steps:

Input: minimum utility threshold
Output: high utility itemsets
Begin
For each transaction T_i, do
Sort the items inside T according to the current sort order
Insert/delete/modify T into the tree
Update the twu and if in the header table H
End
Let total transaction utility value TU=0
Let L be the list of all candidate patterns
Create a header table H to keep the candidate items in the two value descending order
For each item I in $Trans_i$ in DB
While there is a mining request from user do
Input δ from a user
If previous δ > current δ
For each items δ in the header table H do
If utility (α) ≥ minutil threshold
α is high utility itemset
Create Prefix-tree PT_α with the header table HT_α for item α
Call Mining (PT_α, HT_α, α)
End
End
If this is the first δ value for the IHUP-tree then
Scan the database to find high utility itemsets

End

Procedure Mining (T, H, α)
Begin
For each item β of H do
If utility (β) < minutil then
Discard β from T and H to create conditional tree and header table
End
End
Let CT be the conditional tree of α created from T
Let HC be the Header table of condition tree CT created from H
For each item β in HC do
Add pattern αβ in the candidate pattern list L
Call mining ($PT_{\alpha\beta}$, $HT_{\alpha\beta}$, αβ)
End for
End

Example 1: consider a small example of a transaction database representing the sales data then, the profit associated with the sale of each unit of the items and finally it calculates the auspicious and inauspicious items in TWU.Table 2 Represents the sales figure for seven items and five transactions overall. The entry in the transaction represents the unit of any item sold in that transaction list of items.

Table 2 Transaction database

TID	Transaction
T1	(A,1) (D,1) (F,2)
T2	(A,2) (B.6) (D,2) (G,5)
T3	(A,1) (B,2) (C,1) (D,6) (E,1)(F,5)
T4	(C,4) (D,3) (E,1) (F,2)
T5	(B,2) (C,2) (E,1) (G,2)

Table3 Represents the profit associated with the sales of individual items. The profit of each item displays in column

Table 3 Unit Profit Associated With Items

Item Name	A	B	C	D	E	F	G
Profit	5	1	2	1	2	3	1

Table4 Item And Their Twu's

Item Name	A	B	C	D	E	F	G
TWU	67	58	61	84	61	51	33

INAUSPICIOUS ITEMS

Minimum utility= 55

If we consider minimum support count = 55% then we observe that there are 7 itemsets A, B, C, D, E, F, G which qualify as frequent itemsets because they have support more than minimum support threshold value. But if we consider the TWU, we find that out of the two items most inauspicious itemsets (below the minimum utility threshold value) i.e. F, G only two are infrequent itemsets, so it will disposes the items from the list. Itemsets A, B, C, D, E are itemsets which are frequent itemsets (above the minimum utility threshold value) and it will discovered as auspicious itemsets. The auspicious items are displayed in descending order. Finally, both purchased quantities and profits for items should be considered.

5. RESULTS AND DISCUSSION

A minimum utility threshold values as shown in Figure 2: was implemented using Java (JDK 1.5). Administrator will enter the minimum utility value, and time will calculate for entire process is called the processing time and is the difference between starting and ending time of a processing.

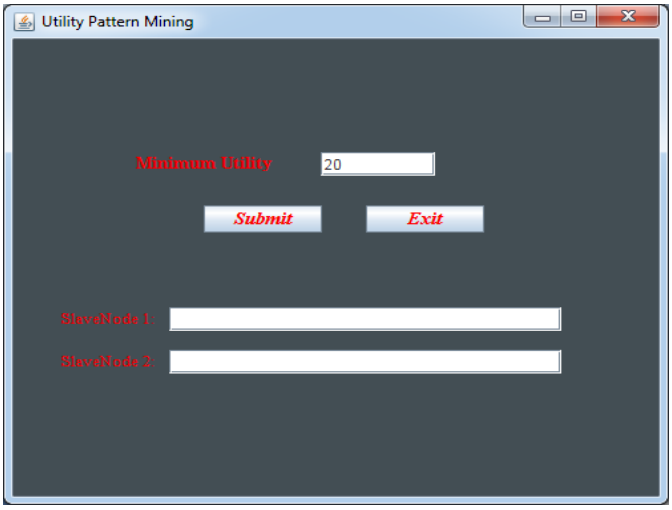

Figure2: Minimum utility threshold value given by administrator.

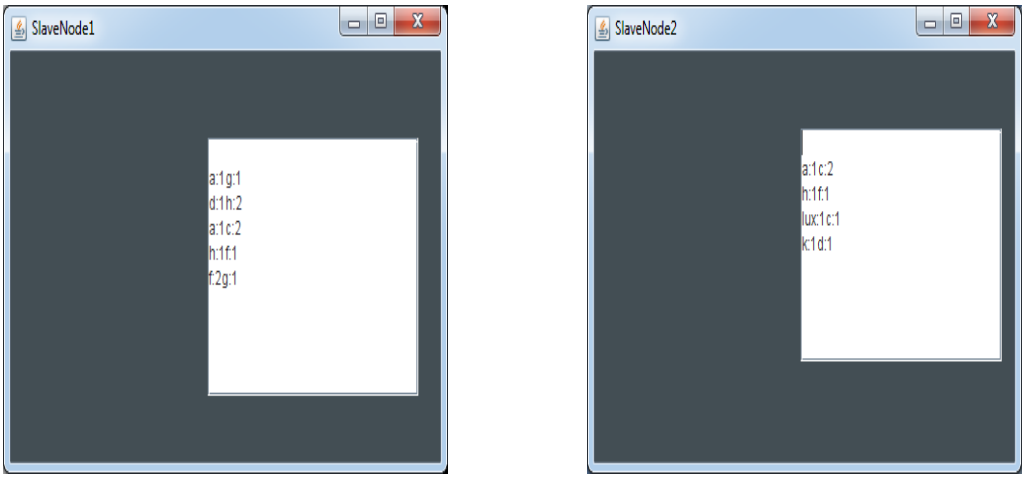

Figure 3: The system ancillary one master node and number of available slaves nodes. Very large distributed database and it sends to all slave nodes for computation.

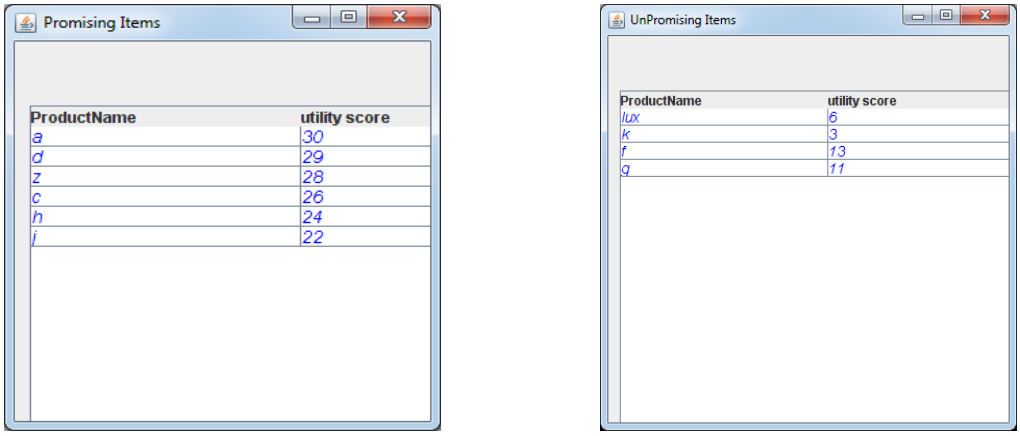

Figure 4: Based on the minimum utility value it discovered the auspicious and inauspicious items (i.e. promising and unpromising items). Auspicious items are those for which the threshold value is higher than the admin specified minimum utility value and hence it disposes the below threshold value.

6. EXPERIMENTAL RESULTS

In this section, experimental results on synthetic and real world datasets [21] are summarized on both high utility and incremental mining algorithm. These experiments were conducted on 2.53 Intel(R) Core(TM) i3 Processor with 2 GB of RAM, and running on windows 7 operating system. In this algorithms were implemented in Java language (JDK 1.5) and applied both real and synthetic datasets to evaluate the performance of the both algorithms.

7. CONCLUSION

In this paper, a high utility and incremental mining method is proposed to generate complete set of high utility itemsets from very large databases. Mining high utility itemsets from databases refer to finding the itemsets with high profit and purchased quantities. In distributed, it inauspicious items based on the minimum utility itemsets from transactions database. This approach creates distributed environment with master node and number of slaves nodes scans the database once and counts occurrence of particular item. The very large database is circulates, number of available slave nodes. The final resultant calculated by global table. Incremental mining algorithm is where regularly adding to the appearing original database. Finally incremental database is layout and the high utility itemsets is generated. Hence, it provides faster execution, reduced time and cost.

REFERENCES

[1] Ruowu Zhong, Huiping Wang,"Research of Commonly Used Association Rules Mining Algorithm in Data Mining", International Conference on Internet Computing and Information Services, pp. 219 – 222, IEEE 2011.

[2] R. Agrawal, T. Imielinski, and A.N. Swami "Mining association rules between sets of items in large databases," Proceedings of ACM SIGMOD International Conference on Management of Data, ACM Press, Washington DC, pp.207-216, May 1993.

[3] R. Agrawal, and R. Srikant, "Fast algorithms for mining association rules," Proceedings of 20th International Conference on Very Large Data Bases, Morgan Kaufmann, pp. 487-499, 1994.

[4] Savasere, A., Omiecinski, E., and Navathe, S. An Efficient Algorithm for Mining Association Rules in Large Databases. Proceedings of the VLDB Conference. 1995.

[5] S.J.Yen and Y.S. Lee: Mining High Utility Quantitative Association Rules, in proc, of 9th int'l conf on data warehousing and knowledge discovery, lecture notes in computer science, pp.283-392, sep., 2007.

[6] Pillai Jyothi, User centric approach to itemset utility mining in Market Basket Analysis, International Journal on Computer Science and Engineering (IJCSE), 2011.

[7] C.F.Ahmed, S.K.Tanbeer, B.S. Jeong, and Y.K. Lee: Efficient Tree Structure for High Utility Pattern Mining in Incremental Databases. In IEEE Transactions on Knowledge and Data Engineering vol.21, issues 12, pp:1708-1721,2009

[8] R. Chan, Q.Yang, and Y. Shen: Mining high utility itemsets. In Proc of Third IEEE Int'l Conf on Data Mining, pp:19-26, 2003.

[9] Jiawei, Han, Jian. Pei, and Y.Yin: Mining frequent patterns without candidate generation: In Proc of the ACM-SIGMOD Int'l Conf. on management of data, pp:1-12, 2000.

[10] H. Yao, H.J. Hamilton, and L. Geng.: A Unified Framework for Utility-Based Measurs for Mining Itemsets. In Proc. Of ACMSIGKDD 2nd Workshop on Utility-Based Data Mining, pp:28-37, USA, Aug.,2006.

[11] Y.Liu, W.K. Liao and A. Choudhary, ―A two phase algorithm for fast discovery of high utility itemset‖, Cheng, D. and Liu. H. PAKDD, LNCS. PP: 689-695, 2005.

[12] Erwin A, Gopalan RP, Achuthan NR (2007) CTU-Mine: an efficient high utility itemset mining algorithm using the pattern growth approach. In: Proceedings of the 7th IEEE international conference on computer and information technology (CIT), 2007, pp 71–76

[13] J.Hu, A. Mojsilovic, ―High utility pattern mining: A method for discovery of high utility itemssets1, in: pattern recognition. PP: 3317-3324, 2007.

[14] Liu Jian-Ping, Wang Ying Fan-Ding, ‖Incremental Mining algorithm Pre-FP in Association Rule Based on FP-tree‖, Networking and Distributed Computing, International Conference, pp: 199-203, 2010.

[15] T. P. Hong and C. Y. Wang, "Maintenance of association rules using pre-larg itemsets," Intelligent Databases: Technologies and Applications, Z. Ma (Ed.), Idea Group Inc., pp. 44-60, 2006.

[16] Vincent S Tseng, Bai-En Shie, Cheng-Wu, Philip S,‖ Efficient algorithms for mining high utility itemsets from transactional databases‖, IEEE Transactions on knowledge and data engineering, 2013.

[17] Y.-C. Li,j,-s. Yeh, and C.-C. Chang, Isolated Items Discarding Strategy for Discovering High Utility Itemsets,‖ Data and Knowledge engg., pp: 198-217, 2008.

[18] Barber B, Hamilton HJ (2003) Extracting share frequent itemsets with infrequent subsets. Data Min Knowl Discov 7:153–185

[19] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, Young-Koo Lee a,Ho-Jin Choi(2012) ―Single-pass incremental and interactive mining for weighted frequent patterns‖, Expert Systems with Applications 39 pp.7976–7994, ELSEVIER 2012.

[20] Tanbeer, S. K., Ahmed, C. F., Jeong, B.-S., Lee, Y.-K., Efficient single-pass frequent pattern mining using a prefix-tree. Information Sciences, 179(5), pp.559–583, 2009.

[21] Frequent itemset mining implementations repository, http:// fimi.cs.helsinki.fi/

ABOUT AUTHORS

G SARANYA completed her Post Graduate degree, M.Sc. Software Engineering from Sathyabama University, Chennai in the year 2012. Her interest towards gaining knowledge made her to successfully complete her second Post Graduate degree, Master of Engineering in department of Computer Science and Engineering from Sathyabama University, Chennai in the year 2014. Her research in the Data Mining domain made her to publish a Survey Paper ”A survey on efficient incremental algorithm for mining high utility itemsets in distributed and dynamic database”. Her area of Interest includes Data Mining and Knowledge Discovery.

A DEEPAK KUMAR completed his Bachelor of Technology degree in department of Information Technology from Sathyabama University, Chennai in the year 2010. He has also completed his Post Graduate degree, Mater of Engineering in department of Computer Science & Engineering from Sathyabama University in the year 2013. In 2013 he joined as an Assistant Professor in the Department of Computer Science & Engineering, St. Joseph’s Institute of Technology, Chennai. His research interest includes Data Mining algorithm and Knowledge Engineering.