AN ENHNACED HIGH UTILITY PATTERN APPROACH FOR MINING ITEMSETS

P.Sharmila[1], Dr. S.Meenakshi[2]

[1]Research Scholar, Department of Computer Science, Gobi Arts & Science College, Tamilnadu, India.

[2]Associate Professor in Computer Science, Gobi Arts & Science College, Tamilnadu, India.

Abstract— **Data mining is used to discover interesting and useful knowledge from massive data. Finding interesting patterns play an important role in knowledge discovery process and are essential for many real life applications. Recently high utility pattern mining is important for mining high utility itemsets which overcomes the limitation of frequent pattern mining. High utility pattern mining is used to identify the itemsets with highest utilities, by considering profit, quantity, cost or other user preferences. This research paper proposes an enhanced high utility pattern approach to mine the high utility itemsets with less computation time and less memory space when larger itemsets are explored for complex datasets.**

Keywords— Data mining, frequent patterns, high utility pattern mining, high utility pattern mining algorithm, high utility itemset mining performance.

I. INTRODUCTION

Data mining is the process of discovering hidden patterns and knowledge with in large amounts of data and also making predictions for behaviors or outcomes [2]. Data mining is commonly used to discover interesting patterns and knowledge discovery from massive data. Interestingness measures play an important role in knowledge discovery process and these measures are intended for selecting and ranking patterns according to their potential interest to the user. Finding interesting patterns is essential for variety of applications such as financial data analysis, retail system, market basket analysis, customer profiling, targeting, inventory prediction, and condition monitoring and genome analysis [9]. The discovered knowledge from data mining is generally classified as association rule, sequential patterns, frequent patterns and utility patterns.

In recent past, utility pattern mining has emerging an important research topic since the quantity and profit factors are both used to discover interesting patterns from huge data. Utility pattern mining addresses the limitation of frequent pattern mining by considering user's expectation as well as the raw data [18]. In frequent pattern mining, the user can easily express their perception regarding the itemset and its attributes and can mine the itemset by defining a threshold value. But in the case of utility pattern mining, these factors take a change. The utility value changes with the preferences of the employed user. The restrictions of frequent or rare itemset mining inspired researchers to conceive a utility based mining approach, which allows a user to conveniently express the perspectives concerning the usefulness of itemsets as utility values and then find itemsets with high utility values higher than a threshold [5]. The various approaches and the process for high utility pattern mining have been reviewed [16]. Among utility mining problems, high utility pattern mining with the itemset framework is more challenging than the other categories of utility mining and frequent pattern mining. This research paper proposes an enhanced high utility pattern approach (EHUPA) for mining high utility itemsets.

The remaining sections of this paper are organized as follows. Section II describes the literature review of existing work for mining high utility itemsets. Section III presents the proposed methodology and algorithm for mining high utility itemsets. Section IV specifies the performance analysis and experimental results of the proposed system and section V presents the conclusion of this research work.

II. LITERTURE SURVEY

The field of high utility pattern mining is gaining more importance in the recent past due to the increase in data generation and the need to get unidentified patterns from the known data sets. Several research works have been proposed to meet the issues of high utility pattern mining. The various proposed algorithms for mining high utility patterns are described as follows.

Liu et al., [13] have proposed pseudo projection algorithm which is fundamentally different from those proposed in the past. This algorithm uses two different structures such as array based and tree-based to represent projected transaction subsets and heuristically decides to build unfiltered pseudo projection to make a filtered copy according to features of the subsets. This work builds tree-based pseudo projections and array-based unfiltered projections has been build for projected transaction subsets which makes algorithm both CPU time efficient and memory saving. This algorithm grows the frequent itemset tree by depth first search, where as breadth first search is used to build the upper portion of the tree if necessary. This algorithm is not only efficient on sparse and dense databases at all levels of support threshold and also highly scalable to very large databases. The disadvantage of this algorithm is, it only support minimum description code length with small number of patterns.

Han et al., [8] have proposed a frequent pattern growth (FP-Growth) algorithm for mining frequent pattern with constraints. In this work the frequent pattern tree (FP-tree) structure which is an extended prefix tree structure developed for storing crucial information about frequent patterns. The pattern fragment growth mines the complete set of frequent patterns using the FP-growth. This algorithm constructs a highly compact FP-tree and applies a pattern growth method for database scans which is usually substantially smaller than the original database by which costly database scans are saved in the subsequent mining processes. The disadvantage of this algorithm is it reduces multi-pass candidate generation process in the first phase by discarding isolated items to reduce the number of candidates. Also this work shrink the database scanned in each pass and it takes more computation time.

Liu et al., [13] have proposed a two-phase algorithm to find high utility itemsets. This algorithm efficiently prunes down the number of candidates and obtains the complete set of high utility itemsets. This work has developed with two phases. Phase one uses transaction-weighted downward closure property which is applied to add high transaction weighted utilization sets during the level wise search. In phase two, any over estimated low utility itemsets are filtered using an extra database scan. This algorithm requires fewer database scans, less memory space and less computational cost for large databases and performs very well in terms of speed and memory cost on both synthetic and real database. The main disadvantage of this algorithm is, the insufficient frequent counts for repeated candidate itemset which can lose interesting patterns.

Li et al., [15] have proposed an isolated items discarding strategy (IIDS) algorithm for utility mining. This algorithm discovered high utility itemset with less number of candidates which improve the performance of the pattern mining. This algorithm shows that itemset share mining problem can be directly converted to utility mining problem by replacing the frequent values of each items in a transaction by its total profit, i.e., multiplying the frequency value by its unit profit. In this work the share frequent set mining scans the database to calculate the share value of each itemset and removes all useless candidate itemsets and remaining candidate to generate. The direct condition generation is a level wise method and it maintains an array for each candidate during each pass. This algorithm provides an efficient way to designed critical operations by using transaction weighted downward closure. However this algorithm still suffers with the problem of level wise generation and test problem of apriori and it requires multiple database scans.

Erwin et al., [6] have proposed a transaction weighted utility (TWU) algorithm which is based on compact utility pattern tree data structures. This work implements the parallel projection scheme to utilize the disk storage. This algorithm first identifies the TWU items from transaction database and the compressed utility pattern tree is constructed for mining complete set of high utility patterns. In this algorithm parallel projection is used to create subdivision for subsequently mining. This algorithm has anti-monotone property which is used to discover the pruning space. In this work the task of high utility itemset mining discovers all the utility which has utility higher than the user specified-utility. Generation of frequent graphs results in high in memory usage and low in accuracy.

Shankar et al., [17] have proposed a fast utility mining (FUM) algorithm that finds all high utility itemset within the given utility constraint threshold. It is faster and simpler than the original UMining algorithm. This algorithm efficiently handles the duplicate itemsets. It checks whether a transaction defined by an itemset purchased in it, repeats its occurrence in a later transaction. If a later transaction also contains same itemset purchased in any of the previous transactions, then that transaction is ignored from processing and duplicate itemset are removed. This reduces the execution time of the algorithm further more. This algorithm provides absolute accuracy and proves to be extremely efficient in finding every possible high utility itemset from the transactions in the database. This algorithm executes transaction datasets exceptionally faster when more itemset are identified as high utility itemset and when the number of distinct items in the database increases.

Ahmed et al., [1] have proposed a tree-based incremental high utility pattern mining (IHUPM) algorithm. In this work a tree based structure called IHUP-Tree which is used to maintain the information about itemsets and their utilities. This work proposes three tree structures to perform incremental and interactive high utility pattern mining efficiently. This reduces the calculations when a minimum threshold is changed or a database is updated. The first tree structure is an incremental high utility pattern lexicographic tree (IHUPLTree) that is arranged according to an item's lexicographic order. It can capture the incremental data without any restructuring operation. The second tree structure is the incremental high utility pattern transaction frequency tree (IHUPTF-Tree) which is simple and easy to construct and handle. In this tree the items are arranged according to their transaction frequency. It does not require any restructuring operation even when the data base is incrementally updated. They have achieved the less memory consumption. The third tree structure is the incremental high utility pattern transaction weighted utilization tree (IHUPTWU-Tree) and this tree is based on the transaction weighted utility value of items in descending order. This algorithm takes insufficient memory usage and outperforms with earlier lexographical approaches.

Tseng et al., [18] have proposed an efficient algorithm called as utility pattern growth plus (UP-Growth+) which is an improved version of utility pattern growth (UP-Growth) mining algorithm. In this work the information of high utility itemset is maintained in a special data structure named utility pattern tree (UP-Tree) and the candidate itemsets are generated with one scans of the database. The four strategies, applied in this algorithm are discarding global unpromising items (DGU), decreasing global node utilities (DGN), discarding local unpromising items (DLU), and decreasing local node utilities (DLN). By these strategies, the estimated utilities of candidates are well reduced, by discarding the utilities of the items which are impossible to be high utility or not involved in the search space. The proposed strategies not only decrease the estimated utilities of the potential high utility itemsets but also reduce the number of candidates. This algorithm outperforms substantially in terms of execution time, especially when the database contains lots of long transactions. However the operation time and search space of high-utility itemset mining can increase the high computation cost.

Liu and Qu., [12] have proposed a high utility itemset miner (HUI-Miner) for high utility itemset mining. This algorithm uses a novel structure called utility-list which is used to store both the utility information about an itemset and the heuristic information for pruning the search space. This algorithm first creates an initial utility list for itemsets of the length 1 for promising items. This algorithm constructs recursively a utility list for each itemset of the length k using a pair of utility lists for itemset of the length k-1 for mining high utility itemset, each utility list for an itemset keeps the information of indicates transaction for all of transactions containing the itemset, utility values of the item set in the transactions, and the sum of utilities of the remaining items that can be included to super itemset of the itemset in the transactions. This algorithm first estimate the utilities of the itemsets and generate the candidate itemsets and then by scanning the database compute the exact utilities of the itemset to generate the high utility itemset. This algorithm mines the high utility itemset without generation of the candidates and the algorithm outperforms in terms of both running time and memory consumption.

Fournier-Viger et al., [7] have proposed an algorithm fast high utility miner (FHM) which extends the high utility itemset miner (HUI-Miner) algorithm. It is a depth-first search algorithm that relies on utility-lists to calculate the exact utility of itemsets. This algorithm consists of discovering frequent itemset that is groups of itemsets appearing frequently in transactions. This work integrates a novel strategy named estimated utility co-occurrence pruning (EUCP) to reduce the number of joins operations when mining high utility itemset using the utility list data structure. The estimated utility co-occurrence pruning structure (EUCP) stores the transaction weighted utility of all itemsets. It built during the initial database scans. The memory footprint of the estimate utility co-occurrence pruning structure is small. This algorithm performs high utility itemset miner based on the analysis of item co-occurrences to reduce the number of join operations that need to be performed. An important limitation of this algorithm is it assumes that each item cannot appear more than once in each transaction and that all items have the same importance.

Junqiang Liu et al., [9] have proposed an algorithm direct discovery high utility pattern (D^2HUP**)** which gains the combination of high utility pattern miner and utility pattern. This algorithm mines utility itemset in share framework. The direct discovery of high utility patterns, which is an integration of the depth-first search of the reverse set enumeration tree. This algorithm addresses the scalability and efficiency issues occurred in the existing systems as it directly extracts the high utility patterns from large transactional databases. This algorithm is based on the powerful pruning approaches. The lookahead strategy tries to find the patterns in recursive enumeration and it utilizes the singleton and closure property to enhance the efficiency of dense data. The linear data structure as chain of accurate utility list is used to show the original information of utility in the unrefined data. This work helps to discover the root causes of prior algorithm which employs to maintain data structure information of original utility.

Most of the high utility pattern mining algorithms reviewed in this section based on the factors include one phase, two phase approaches for generating candidates of high utility itemsets. Also the recent research has focused on high utility mining using anti-monotone measures for pruning the search space. However the challenge is, if the numbers of candidates are huge, then the scalability and efficiency issues still persists for finding high utility patterns. Also the existing algorithms also suffer from poor performance when mining dense datasets and long transactions. In order to overcome the issues, the focus of this research work is to develop an enhanced high utility pattern algorithm for mining itemsets.

III. PROPOSED METHODOLOGY

In this research work an enhanced high utility pattern approach (EHUPA) has been proposed to improve the performance of high utility pattern for mining itemsets. The proposed system contains name of the item as node and after calculating transaction utility and transaction weighted utility, the item sets having less utility than predefined minimum threshold utility are identified. Local unfavorable items are removed using path utility of each item in descending order. The reorganized path is inserted into the utility pattern tree using reduce local node utility strategy. Potential high utility item sets and their utilities are identified by the proposed system. The proposed system eliminates the local unfavorable items and reduces local node utility. The proposed system improves the performance of high utility pattern for mining itemsets in large datasets with several advantages such as less memory space usage and less execution time for mining itemsets.

A. System design

The proposed enhanced high utility pattern approach has been designed to find effective high utility patterns for improving the performance of mining itemsets. The proposed system describes the dataset for set of transactions with profit item as input to system with calculation of transaction utility transaction weighted utility, utility pattern tree construction, high utility pattern algorithm and finally the output as the enumerated patterns for utility itemsets. The transaction utility and transaction weighted utility prune the search space of high utility itemsets. The proposed system design is shown in Figure 3.1

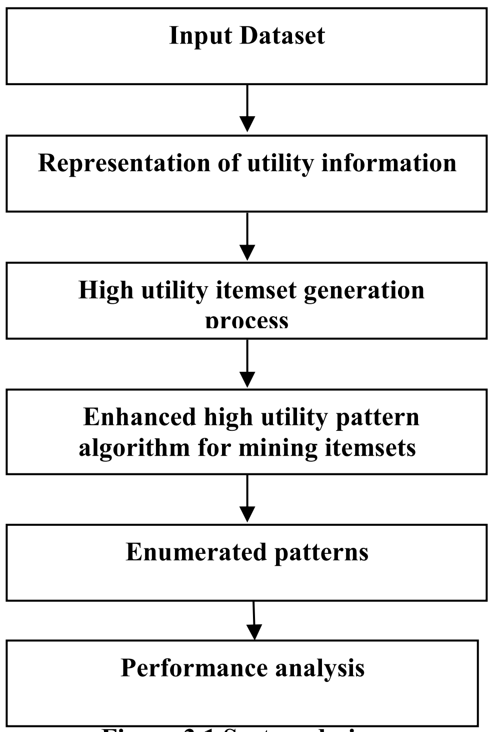

Figure 3.1 System design

The proposed system builds transaction set (TS) by scanning the database D and build the external utility table (XUT)to compute s(i), u(i),uBitem(i), and uBfpe(i) for each item i. The proposed system starts searching high utility patterns from the construction of the utility pattern tree by calling the depth first search (DFS) approach. For the each node N currently being visited, DFS prints pattern (N) as a high utility pattern if its utility is no less than the threshold which makes the set W of relevant items.

B. Enhanced high utility pattern algorithm

A pattern that is of interest of one user may not be interest to another user, since users have different levels of interest in patterns. A pattern is of utility to a person if its use by that person contributes to reach a goal. People may have differing goals concerning the knowledge that can be extracted from a data set. The proposed system allows a user to conveniently express the perceptiveness concerning the usefulness of patterns as utility values higher than a threshold.

The proposed enhanced high utility pattern approach (EHUPA) finds high utility pattern to enumerate each subset of item, and test if subset has a utility over the threshold. The transaction set is build scanning the database and the utility table is used to filter out the irrelevant items. The proposed system starts searching high utility patterns from the root of utility pattern tree using depth first search. The node currently is being visited computing utilities and if it's utility is no less than the threshold, makes the set of relevant item for a high utility for each relevant item belongs to the set.

The proposed algorithm has been used the utilities such as transaction utility and external utility for identifying the items. The transaction utility of an item is obtained from the information stored in the transaction dataset. The external utility of an item is given by the user and is based on information not available in the transaction dataset. In this work external utility has been represented by a utility table or utility function. By combining a transaction dataset and a utility function the proposed algorithm finds the discovered pattern.

The proposed algorithm performs various steps that include representation of utility information, construction of utility pattern tree and the generation of high utility pattern for itemsets. The proposed algorithm provides scalability and efficiency for mining utility itemsets along execution time and memory space on database transactions. The proposed enhanced high utility pattern algorithm for mining high utility itemsets is shown in Figure 3.1

Step 1: Scan D and Build TS ({})
  Scanning the database D of transactions T_d € D
  // where TS= transaction set, D= database
Step 2: Determine transaction utility of T_d in D and TWU of itemset(X)
Step 3: Construct external utility table XUT to compute
  s({i}),u{(i)},uBitem(i{}),uBfpe(i{}) for each item i
Step 4: List the items in the descending order of uBitem
Step 5: Construct HUP Tree using DFS (N,TS(pat(N)),minU)
Step 6: Generation of high utility pattern **if** u (pat(N)) ≥ minU **then** output pat(N)
Step 7: Makes the set W of relevant items W ← {i|i<pat(N) ∧uBitem(i pat(N)) ≥ minU}
Step 8: DFS prints the pat(N) with relevant items if the closure property holds
  if Closure(pat(N),W, minU) is satisfied
  then output nonempty subsets of W ∪ pat(N)
Step 9: DFS prints the pat(N) with relevant items if the singleton property holds
  if Singleton (pat(N),W, minU) is satisfied
  then output nonempty subsets of W ∪ pat(N)
Step 10: foreach relevant items i€ W if the upper bound on the utilities is not less than
  Threshold DFS prepares TS(pat(c)) for child node C for item i
Step 11: else foreach item i€ W
  if uBfpe({i} ∪pat(N)) ≥ minU
  then *C* ←the child node of N for i
Step 12: Generation of high utility pattern for mining itemsets from recursively searches
  the subtree rooted at C TS(pat(C)) ← project(TS(pat(N)),i)
Step 13: End

Figure 3.2.Enhanced high utility pattern mining algorithm

- **Representation of utility information**

The proposed algorithm performs a scan in the database involved in the transaction to find the entire itemset utility information. Then the algorithm determines the transaction utility (TU) and transaction weight utility (TWU) of itemset X. After computing the utility values. The utility function f has been defined to express the significance of the itemset. Based on the utility formulation of an itemset the high profit itemsets are obtained from a dataset.

- **Construction of utility pattern tree**

After obtaining the itemset utility information by scanning the database, utility patterns are mined from the tree structure. The proposed algorithm constructs a tree structure called high utility pattern (HUP) tree. In a tree, each node consists of item name, count, parent node and child node details and node utility which refer the overestimated utility of node. In order to construct the tree structure the proposed algorithm needs to update the node links of transaction. Thus the utility table has been used to facilitate the traversal of tree which maintains the node details. The proposed algorithm discards the low utility items and their utilities from the transaction utilities of the paths in descending order.

- **Generation of high utility pattern for itemsets**

The actual high utility itemsets are identified and extracted from the HUP tree and the proposed algorithm generates the required items in efficient manner. High-utility itemsets has been generated from HUP tree efficiently with only two scans of original databases. The proposed algorithm has been generated the patterns by tracing the paths in the tree. The overestimated utilities of candidates have been reduced by discarding utilities of the items that are not involved in the search space.

IV. EXPERIMENTAL RESULTS

In this work five real-world datasets are used for evaluation. The first dataset is T10I6D1M which contains the items are selected such as milk, bread, butter, jam. The data from used for 1-itemset,2-itemset,3-itemset.The second one is Chess which is a dense dataset used for transaction items. The third dataset is Chain store generated the itemsets. The fourth one is T20I6DIM in mixed dataset, it increases with the transactions. The last dataset is foodmart which contains real utility values generated from high utility values.

For T10I6D1M, Chess, Chain store and T20I6DIM, food mart divide each part generated itemsets in 10,000 items are selected for each transaction. The transaction database that contains items which find effective high utility patterns for improving the performance of mining itemsets. The method provides scalability and efficiency for mining utility itemsets along execution time and memory space on database transactions. The first column is the name of a dataset, the second (|t|) is the average and maximum length of transactions, the third (|I|) is the number of distinct items, the fourth (|D|) is the number of transactions, and the fifth (Type) is a rough categorization based on the number of high utility patterns to be mined, partially depending on the minimum utility threshold. The detailed of the five datasets which include transaction, distinct items, number of transaction and type are shown in Table 4.1.

The proposed system contains name of the item as node and after calculating transaction utility and transaction weighted utility, the item sets having less utility than predefined minimum threshold utility are identified. Local unfavorable items are removed using path utility of each item in descending order. The reorganized path is inserted into the utility pattern tree using reduce local node utility strategy. Potential high utility item sets and their utilities are identified by the proposed system. The proposed system eliminates the local unfavorable items and reduces local node utility.The proposed enhanced high utility pattern approach has been designed to find effective high utility patterns for improving the performance of mining itemsets. The proposed system describes the dataset for set of transactions with profit item as input to system with calculation of transaction utility transaction weighted utility, utility pattern tree construction, high utility pattern algorithm and finally the output as the enumerated patterns for utility itemsets. The transaction utility and transaction weighted utility prune the search space of high utility itemsets. The proposed algorithm has been executed on same minimum utility value as per the datasets to generate itemsets. Experiments are performed to evaluate the performance of the proposed EHUPA with the existing d^{2}HUP algorithm based on five datasets.

Table 4.1 data sets

Data Set	Transaction $\|t\|$	Distinct Items $\|I\|$	Number of Transaction $\|D\|$	Type
T10I6D1M	10:33	1000	933,493	Mixed
Chess	37:37	76	3,197	Dense
Chain-Store	7.2:170	46,086	1,112,949	Sparse
T2016DIM	20:49	1,000	999,287	Mixed
Foodmart	4.8:27	1,559	34,015	Dense

B. Performance analysis and results

Experiments are performed to evaluate the performance of the proposed EHUPA algorithm based on the five datasets. The performance of EHUPA algorithm has been compared with existing d^{2}HUP algorithm based on the metrics such as memory usage and running time for mining high utility itemsets.

- **Memory usage for high utility pattern**

The proposed EHUPA is compared with existing d^{2}HUP method for memory usage for mining high utility itemsets and the performance graph is shown in Figure 4.1. In the graph, x-axis represents the datasets and y-axis represents the memory space. The graph shows that the proposed EHUPA method provides better high utility pattern mining with less memory space usage than the existing d^{2}HUP method.

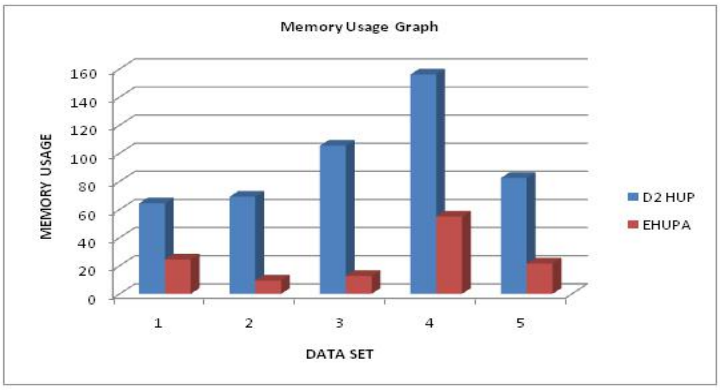

Figure 4.1 Comparison of memory usage for high utility pattern

- **Running time for high utility pattern**

The proposed EHUPA is compared with existing d^{2}HUP method to mine high utility itemsets for running time and the performance graph is shown in Figure 4.2. In the graph, x-axis represents the datasets and y-axis represents the running time of utility itemsets. The graph shows that the proposed EHUPA method provides better high utility pattern mining with less running time than the existing d^{2}HUP method.

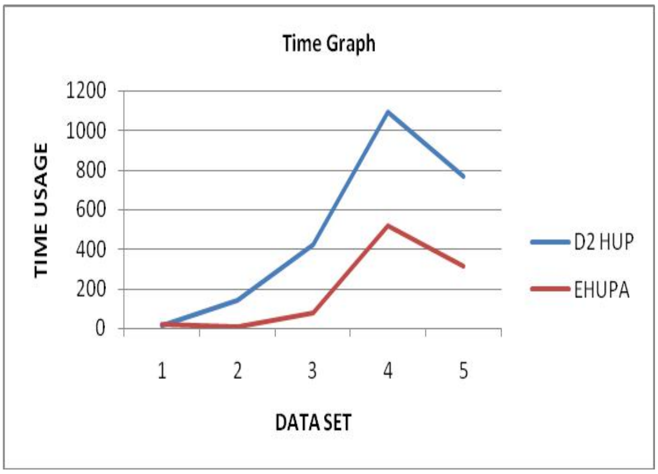

Figure 4.2 Comparison of running time for high utility pattern

V. CONCLUSION

The high utility pattern mining with the itemset share framework is more challenging than the other categories of utility mining such as weighted itemset mining, association rule mining and frequent pattern mining. During the knowledge discovery process, utility based measures are used to find the unidentified patterns to improve the mining efficiency. In this research paper an enhanced high utility pattern approach (EHUPA) has been proposed to mine high utility itemsets. The experimental results show that the proposed system provides better performance than the existing direct discovery high utility pattern algorithm in terms of memory usage and execution time for mining high utility itemsets.

REFERENCES

[1] Ahmed C. F., Tanbeer S. K., Jeong B.-S., and Lee Y. -K., "Efficient tree structures for high utility pattern mining in incremental databases," IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 12, pp. 1708– 1721, 2009.

[2] Arati Borker W., "Utility mining algorithm for high utility itemsets from transactional databases," International Organization of Scientific R Journals (IOSR Journal of Computer Engineering), Vol. 16, No. 2, pp. 34– 40, 2014.

[3] Bonchi F., Giannotti F., Mazzanti A., and Pedreschi D., "Exante: A preprocessing method for frequent-pattern mining," IEEE Intelligent Systems, Vol. 20, No. 3, pp. 25–31, 2005.

[4] Chun-Wei Lin J., Wensheng Gan., Fournier-Viger P., and Yang L., Liu Q., Frnda J., Sevcik L., Voznak M., " High utility itemset-mining and privacy-preserving utility mining," Vol. 7, No. 11, pp. 74–80, 2016.

[5] Dawar S., Goya V. l., "UP - Hist tree: An efficient data structure for mining high utility patterns from transaction databases," In Proceedings of the 19th International Database Engineering & Applications Symposium. Association for Computing Machinery, pp. 56–61, 2015.

[6] Erwin A., Gopalan R. P and. Achuthan N. R., "Efficient mining of high utility itemsets from large datasets," In Proceeding of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 554–561, 2008.

[7] Fournier-Viger P., Wu C.-W., Zida S., and Tseng V.S., "Fhm: Faster high-utility itemset mining using estimated utility Co-occurrence pruning," In Proceedings of the 21th International Symposium on Methodologies for Intelligent Systems. Springer.pp.83-92, 2014.

[8] Han J., Pei J., Yin Y., Mao R., "Mining frequent patterns without candidate generation: a frequent-pattern tree approach," Data Mining Knowledge Discovery in Data.Vol. 8, No. 1, pp. 53–87, 2004.

[9] Junqiang Liu., Ke Wang., Benjamin., Fung C.M.,"Mining High Utility Patterns in One Phase without Generating Candidates", IEEE Transactions on Knowledge and Data Engineering, Vol. 28, No. 5, pp.1–14, 2016.

[10] Kanimozhi R., Saravanan K., "Decisive high-utility itemset mining an innovative algorithm for mining the high utility itemsets," International journal of computer engineering and technology. Vol. 8, No. 4, pp.127–137, 2017.

[11] Komal Surawase., Madhav Ingle., "High Utility Itemset Mining from Transaction Database Using UP-Growth and UP-Growth+ Algorithm", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, No. 4, pp.3262–3270, 2015.

[12] Liu M., Qu J., "Mining high utility itemsets without Candidate generation," Conference on Information and Knowledge Management. Association for Computing Machinery, pp. 55–64, 2012.

[13] Liu V., Liao W., and Choudhary A., "A fast high utility itemsets mining algorithm," in utility – Based Data Mining Workshop in Special Interest Group on Knowledge Discovery in Data. Association for Computing Machinery, pp. 253 – 262, 2005.

[14] Liu J., Wang K., and Fung C. M., "Mining high utility patterns one phase without Candidate generation", IEEE Transactions on Knowledge and Data Engineering, Vol. 28, No. 5, pp. 1–14, 2016.

[15] Li Y.-C., Yeh J.-S., and Chang C.-C., "Isolated items discarding Strategy for discovering high utility itemsets," Data & Knowledge Engineering, Vol. 64, No. 1, pp. 198–217, 2008.

[16] Meenakshi S., Sharmila P., "A Review of Mining high utility itemsets," International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol. 5, No. 8, pp. 14055-14061, 2017.

[17] Shankar S., Purusothoman T.P, Jayanthi S., Babu N., "A fast agorithm for mining high utility itemsets" ,In Proceedings of IEEE International Advance Computing Conference (IACC), Patiala, India, pp.1459-1464, 2009.

[18] Tseng V. S., Shie B.-E., Wu C.-W., and Yu P. S., "Efficient algorithms for mining high utility itemsets from transactional databases," IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No. 8, pp. 1772–1786, 2013.