Volume 8, No. 5, May-June 2017
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Pattern Matching Base Tree Construction for Extracting High Utility Patterns in Data Mining

Bhavya Shukla
Dept. of Computer and Science
Madhav Institute of Technology and Science
Gwalior, M.P., India

Prof.R.K Gupta
Dept. of Computer and Science
Madhav Institute of Technology and Science
Gwalior, M.P., India

***Abstract*:** High utility pattern mining has become a grueling novel analysis topic in data mining. Discovery of patterns with high profit from datasets is thought as High Utility Pattern mining. There square measure three window models used usually in data streams specifically (landmark window model, Damped or Time attenuation window model and Sliding window mechanism model). The traditional ways of sliding window mechanism that uses HUP mining algorithmic rules suffer from a retardant of level wise candidate-and-test-generation which degrades the performance of mining in terms of overall execution time and memory consumption. Attributable to the large characteristics of streamed data which have fast growing arrival rate, real time, unbounded and continuous which need to be scanned only once and as soon as the new data arrives discarding the old information is the crucial challenge. In this paper, we unravel these issues by proposing a algorithm named *Pattern Sliding Window Based High Utility –Growth(PSHU-Growth)*tree algorithm which is economical one pass tree approach for mining patterns matching base tree over data streams. Outcomes show that our algorithm provides higher results than ancient approaches that suffer from level wise candidate-test-and generation increasing the overall execution and efficiency.

Keywords: *Data Mining, HUI, Data stream etc.*

I. INTRODUCTION

High utility patterns visit the sets of items with a high utility like profit in an exceeding info, and economical mining of high utility patterns plays an imperative part in a few reality applications and is a vital analysis issue in data preparing district.

High utility itemsets mining is growing with a lot of innovative mining techniques with wider applications area unit being developed. Mining high utility itemsets from transactional databases is incredibly vital and has wide selection of applications like on-line e-commerce management, website click stream analysis, mobile commerce setting coming up with, and business promotion in chain hypermarkets, cross promoting in retail stores Utility Mining is among one amongst the foremost troublesome data processing activity that is that the mining of high utility itemsets with efficiency. Discovering the itemset with high utilities is known as as Utility Mining. A high utility itemset is an itemset that's used regularly and is a worthwhile itemset, also it's far measured according to user desire software or other expressions [1, 2]. The researchers got here up with the idea of application primarily based mining which entails a user to freely express his or her view for the usefulness of itemsets as software values and among them find the high application values more than threshold due to the limitations of common and uncommon itemsets. The time period application is the quantitative measure of consumer choice this is user's view approximately the software cost of itemset.

Application mining rises as a basic point in data mining subject. Mining high utility itemsets from databases alludes to finding the itemsets with high profit. Here, which methods for itemset utility is intriguing quality, centrality, or benefit of an object to user. Utility of items in a transaction database consists of two aspects:

The significance of discrete items is called external utility whereas internal utility is defined as significance of items in transactions. Mining high utility itemsets from databases refers to coming across itemsets with excessive earnings or excessive software values. The meaning of itemset utility is profitability, characteristics or importance of an item in consumer's factor of view or customers want. A high utility itemset can be expounded as: A pack of itemsets in a transactional database. This itemset in a transactional database consists of two standards: Firstly, Data in single transaction which has a itemset price are referred to as Internal utility and Secondly, Data in multiple transactions which has itemsets values are called External utility.

II. DATA STREAM

A data stream is a infinite ,unbounded and extended sequence of tuples, which is usually ordered, commonly by tuple arrival time or tuple number In data stream facts is added at fast charge and thus, they have non-stop and limitless functions and their sizes are constantly accelerated consistent with the accumulation of transaction data so the patterns generated over data streams also become very large in size, which means spending a huge amount of time in mining the patterns, and thereby it can rebel against the most important requirements for the streamed data in data mining that is immediate processing[3]. Data stream mining has to satisfy the requirements in which each data element required for data stream analysis has to be examined only once and all of the entered data elements have to be processed very quickly and the results of data stream analysis should be available instantly and their quality should also be acceptable whenever users want the results. The old frequent pattern mining techniques do not satisfy these requirements since they need to conduct multiple database scans to mine latest frequent patterns.

Therefore, to resolve these problems apply mining approaches which support for single database scan to import the data and use effective tree structure to find latest. Frequent pattern over data streams effectively. The data streams are utilized in one-of-a-kind industrial fields like network tracking, sensor group

evaluation, cosmological software, and intrusion detection, natural and climate data analysis.

Data stream applications

Data streams are utilized as a part of different applications some critical programs are as follows:

- Network monitoring in data stream
- Intrusion detection in data stream
- Sensor network analysis in data stream
- Cosmological application in data stream
  Environmental and weather data in stream

III. SLIDING WINDOW MECHANISM

There are 3 varieties of data stream process models [4] specifically, Landmark Model, Damped Model or Time attenuation Model, sliding window Model. Landmark Model processes the complete history of stream knowledge over the some specific purpose within the past and in the present. During this model, outline knowledge is to be maintained within the organization. Sliding window Model maintains and processes the part of the stream knowledge within the current window. The result from window model reflects the recent frequent itemsets. The previous transactions are deleted once the new transactions arrived into the present window for process because of infinite, high speed characteristic of data in nature. The size of the window depends on applying values on system resources. Damped Window Model strategies the circulation information based totally at the load assigned to every transaction. Here, the older transactions are allotted by less weight towards the itemset frequencies and better weight for recent information. Damped Window Model is additionally one in every of the categories of window model. In this model, the decay rate is employed to cut back the result of previous transactions from the window. This model brings the recent frequent itemsets within the mining result. Based mostly upon the application and user wants, the model has been chosen for mining process.

IV. LITERATURE SURVEY

Xiaoxuan Wang et al. [2017] Makes a speciality of a trouble of incremental mining excessive application co-locations on spatial databases which might be constantly changed with brought and disappeared records. In a spatial database when changes are made Increment of mining in high utility co locations is a complicated process, because added and disappeared data will produce new spatial relationships and take away the existing spatial relationships. The changed relationships can hold the results of utility based co-location mining. So the efficient update of the utility based co-locations is a big challenge for us. This paper presents an efficient algorithm for incremental mining the high utility patterns and evaluates the method by experiments [5].

Minh Nguyen Quang et al. [2016] have proposed strategies for hiding excessive utility sequential patterns. The traditional approach is first using mining algorithms to discover all high utility sequential patterns in a specific user threshold and then apply hiding algorithms to conceal them. Generally, these algorithms are usually time consuming when performing in the large datasets. To address this issue, this paper presents associate degree integrated algorithmic program named MHHUSP (Mining with activity High Utility serial Patterns) which mixes mining process with hiding process in a very common method. an intensive experimental analysis is conducted on large-scale datasets to gauge the performance of the projected algorithmic program in terms of execution time and memory consumption. Experimental results show that MHHUSP outperforms the state of- the-art HHUSP algorithmic program [6].

Jingyu Shao et al. [2016] In this examination acknowledgment has been on upgrading the effectiveness to make algorithms speedier and more noteworthy strong However, the coupling connections between items in given itemsets area unit unnoticed. For instance, the utility of 1 itemset may well be not up to the manager expected till one extra item takes half in; and the other way around, the utility of associate itemset would possibly drop sharply once another one joins in. What's additional, it's not occasional to search out that quite an ton of redundant itemsets sharing a similar underlying item area unit bestowed supported existing educational HUI mining ways. Store managers wouldn't create expected profits supported such results that make the results not unjust the least bit to the current finish, here we have a tendency to introduce a brand new framework for mining unjust patterns, known as Mining Utility Associated Patterns (MUAP), that aims to search out high utility progressive and powerfully associated item/itemset with combined incorporating criteria. The outputs of this formula area unit convincing on real datasets also as artificial datasets [7]

Morteza Zihayat et al. [2016] In this paper,proposed a new framework for mining HUSPs in big data. A distributed and parallel algorithm called *Big HUSP* is proposed to discover HUSPs efficiently. At its heart, Big HUSP uses multiple Map Reduce-like steps to process data in parallel. We additionally propose some of pruning strategies to limit seek space in a disbursed environment, and for this reason decrease computational and communique charges, whilst nonetheless maintaining correctness. Our trials with genuine life and huge engineered datasets approve the viability of Big HUSP for mining HUSPs from huge sequence datasets [8]… We additionally propose some of pruning strategies to reduce seek area in a distributed surroundings, and consequently decrease computational and communique prices, in the meantime as in any case looking after accuracy. Our trials with genuine and extensive manufactured datasets approve the adequacy of Big HUSP for mining HUSPs from enormous accumulation datasets [8].

Serin Lee et al [2016] in this paper, proposed a new algorithm, TKUL-Miner, to mine top-k high utility itemsets efficiently. It utilizes a new utility-list structure which stores necessary information at each node on the search tree for mining the itemsets. The proposed algorithm has a approach the usage of search order for precise region to elevate the border minimal software threshold swiftly. Moreover, two extra strategies for calculating smaller overvalued utilities are cautioned to prune unpromising itemsets successfully. Exploratory outcomes on different datasets demonstrated that the TKUL-Miner outflanks other late algorithms both in runtime and memory efficiency [9].

Prajakta R. Padhye et al [2016] in his analysis work introduced a system that uses HUPID-Tree structure to take care of

knowledge} regarding the info and patterns and it's updated solely with the incremented data. It reduces the time overhead of rescanning the info from the start. High utility itemsets (HUIs) i.e. the fascinating patterns strip-mined from the HUPID-Tree are used for generating rules. Cross commercialism profit of every rule are calculable with the assistance of associate objective function i.e. the rule utility function. Cross commercialism is that the apply of commercialism among the established customers. It uses things within the sequent a part of a rule for recommendation and provides future profit info with the applying of a rule. Managers will use this cross-selling profit info to maximize the profit and therefore the itemsets which can be sold within the future also will be the high utility item sets [10].

Junqiang Liu et al. [2015] in his analysis work planned a unique algorithmic rule that finds high utility patterns in a very single section while not generating candidates. The curiosities lie a high utility pattern growth approach, a look forward technique, and a straight association. Solidly, our pattern growth approach is to go looking a turn around set list tree and to prune search region by utility higher bouncing. Our conjointly look ahead to spot high utility patterns without enumeration by a closure property and a singleton property. Our linear system allows United States of America to reckon a decent sure for powerful pruning associated to directly establish high utility patterns in an economical and scalable means that targets the basis cause with previous algorithms. in depth experiments on distributed and dense, artificial and world knowledge recommend that our algorithmic rule is up to one to three orders of magnitude additional economical and is additional ascendable than the progressive algorithms [11].

Vincent S. Tseng, et al [2015] during this paper, addressed the higher than drawbacks by proposing a replacement framework for top-k high utility itemset mining, wherever k is that the desired variety of HUIs to be well-mined. Two varieties of cost-effective algorithms named TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One phase) area unit projected for mining such itemsets while not the necessity to line min_util. we give a structural comparison of the two algorithms with discussions on their benefits and limitations. Empirical evaluations on each real and artificial datasets show that the performance of the projected algorithms is near that of the optimum case of progressive utility mining algorithms [12].

V. PROPOSED WORK

In existing technique author create Batches on random basis but for faster accessing higher item sets. we first put transactions whose data set is same, so that we have batches of similar type of transaction, after that we create a window, and one window contain those transaction who have same consecutive three similar data items. By creating these two things we focus on creating a tree on the basis of window and batches. When we put similar transaction together calculating the values of streamed data becomes easy and accessing higher data items also become fast.

We will continue this process till three places say A, B, C, which may appear for transaction T1, T2, T3 then the further process will continue from where we will get the value of streamed data in continuous manner

Increasing the performance and Integrity of transactions that are being calculated. Decreasing the overall processing and generates less candidates within less time. After which hierarchy of trees is created which will contain the parent and child nodes such as A(PARENT) and B(CHILD) where the items with the same frequency will lie on the relative node i.e. child node of the respective parent. To execute the above process we define an algorithm for construction of patterns using *Pattern Sliding Window based High Utility –growth (PSHU-Growth) algorithm.*

Steps for PSHU-Growth algorithm are shown below-

1. Arrange all the transaction according to first 3 items
2. Ws←0
3. T← NULL
4. H← Null
5. While Ws<=window size
6. B no← 0
7. While B no<=batch size
8. scan transaction in a data stream from present location
9. Trans ← scanned transaction
10. U←0
11. R←root node
12. Repeat [x to till all transaction]
13. If R has no child R_x such that R_x =x
14. Create a new child R_x under R
15. $R_{x\ .name=item}$
16. $R_{x.tail}$=null
17. $R_{.ru}$=0
18. Calculate item utility of x
19. Increment u
20. If Header has no entry for x
21. Create a new entry E_x
22. Set E_x .RTWU=0
23. Increment E_x .RTWU by u
24. R←R_x
25. If R_{tail} =null
26. Than allocate an array to $R_{x.tail}$
27. Set value of present batch of Trans in R_{tail} to TRUE
28. Increase B no
29. Increase Ws

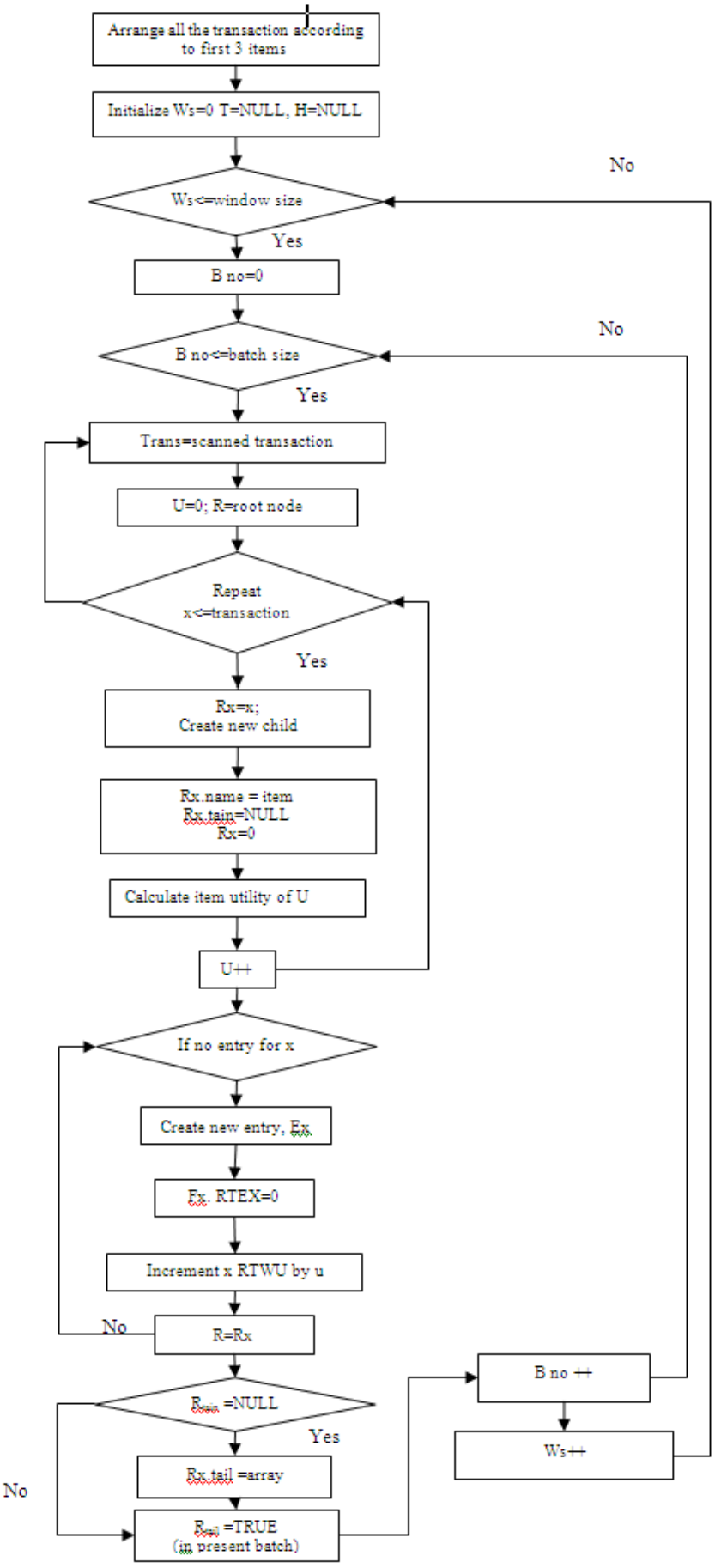

Figure 1. Flow Chart

VI. RESULT ANALYSIS

In this section, we evaluate the performance of our algorithm PSHU-Grow. For the evaluation, various experiments were conducted on a 2.20 GHz Intel Processor with 2 GB main memory. In addition, the experiments were run on the Windows 7 operating system. All algorithms used in the experiments were written in C ++ language. Items in transactions are sorted in a accordingly as of which three transactions have same data values are used for Construction batches before inserting the transactions into the global trees. Next, high transaction weighted utilization patterns (or candidate patterns) are generated. Finally, the algorithms identify high utility patterns from the candidates by an additional database scan.

Below a table is created by evaluating the various values on different datasets and there graphs are plotted.

No. of Transaction	Elapsed time (base)	Elapsed time (proposed)
100	15.373269 sec	14.328968sec
200	50.565146 seconds	44.422810 seconds
300	91.282439 seconds	88.929643 seconds.
400	151.848448 seconds.	141.699010 seconds
500	271.391809 seconds	250.282411 seconds.
1000	1048.148740 seconds	768.522684 seconds.

TABLE.1.

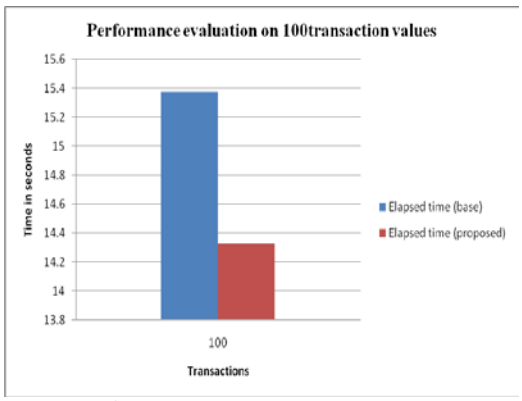

Figure 2. Performance evaluation on 100 transaction values

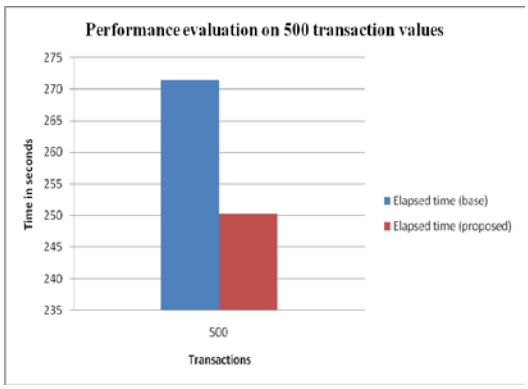

Figure 3 Performance evaluation on 500 transaction values

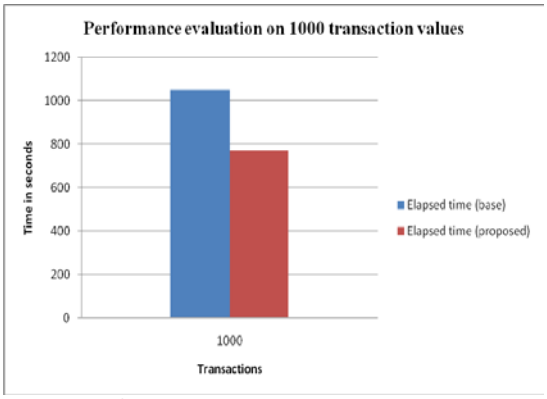

Figure 4. Performance evaluation on 1000 transaction values

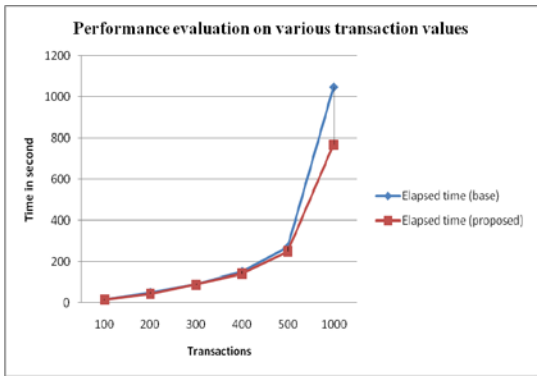

Figure 5. Performance evaluation on various transaction values

From above results shown in figure 2, 3, 4 and 5, we have calculated the reduced overestimation utilities and the overall performance of the system is enhanced by decreasing the running time and space consumption.

Experimental results of planned formula for construction of pattern matching base tree an algorithm was designed and are discussed in this section. It is implemented in R130b MATLAB 2013 on windows seven operating system. For testing patterns with high profits mainly named as patterns with high utility value in streamed data using sliding window mechanism. We used Real Dense Datasets namely *Accidents and Retail* were obtained from the FIMI repository (http://fimi.cs. helsinki.fi)which were obtained from FMI repository. The data sets contain values of various transactions along with their profits on which item utilities are calculated.

- **No of patterns generated**

There are windows created for each three batch size which have the at least three same consecutive values namely W1,W2,W3…..The performance of our algorithms is compared by the two factors particularly execution time and also the range of patterns discovered in every window sliding concept is used in this work. After finding the patterns in W1 the next window W2 automatically slides. Different sizes of transactions 100,200,300,400,500 and 1000 are tested and their results are obtained.

- **Time efficiency**

Another performance factor used for measuring the effectiveness of its execution time.

Execution time is nothing however what proportion time needed for characteristic the patterns with minimum utility in every window.

VII. CONCLUSION

Major reasons for overall performance degradation of overestimation technique-primarily based High Utility Pattern

Mining over sliding window statistics streams are the generation of plenty of candidates and tests because of overrated utilities. By using this approach we analyzed and proposed A algorithm for construction of tree, PSHU-Growth, for mining excessive Utility Patterns Effectively from the reflecting environment. For this purpose, we developed Batch sizes on which windows are created on the basis of each Transaction which has at least three similar Data values. We additionally devised a singular tree structure, PSHU-Tree, and production strategies for the tree so that you can make use of decreased overestimation utilities, to emphasize the latest information, and to apply the sliding window version, which facilitates extra efficient mining of the identical whole set of high application styles without any loss. through the proposed algorithm, it is feasible to discover excessive software styles from streamed data with relatively high speed and much less or similar memory usage as compared to the preceding one with the aid of generating the smaller variety of candidates and search space and by using reduced processing time. Furthermore, the proposed method also can be employed to improve mining performance of the other similar concept, high average utility pattern mining, and accordingly, we are scheduled to develop the above solutions in the future work.

VIII. REFERENCES

[1] Prashant V. Barhate, S. R. Chaudhari and P. C. Gill," Efficient High Utility Itemset Mining using Utility Information Record", International Journal of Computer Applications (0975 – 8887) Volume 120 – No.4, June 2015.

[2] V. Keerthy, Mrs. B. Buvaneswari," Survey of Efficient Algorithms in Data Mining For High Utility", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 11, November 2015.

[3] Rahul Anil Ghatage,'' Frequent Pattern Mining Over Data Stream Using Compact Sliding Window Tree & Sliding Window Model", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 02 Issue: 04 | July-2015.

[4] B. Subbulakshmi.Dr. C. Deisy and A. Periya Nayaki." Survey on Frequent Pattern Mining over Data Streams, International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 12, December - 2013

[5] Xiaoxuan Wang, Lizhen Wang," Incremental Mining of High Utility Co-locations from Spatial Database,"978-1-5090-3015-6/17/$31.00 ©2017 IEEE

[6] Minh Nguyen Quang, Tai Dinh, Ut Huynh, Bac Le," MHHUSP: An Integrated Algorithm for Mining and Hiding High Utility Sequential Patterns" 2016 Eighth International Conference on Knowledge and Systems Engineering (KSE), IEEE

[7] Jingyu Shao, Xiangfu Meng, Longbing Cao" Mining Actionable Combined High Utility Incremental and Associated Patterns", 2016 IEEE/CSAA International Conference on Aircraft Utility Systems (AUS)

[8] Morteza Zihayat, Zane Zhenhua Hu, Aijun An and Yonggang Hu,",Distributed and Parallel High Utility Sequential Pattern Mining, 978-1-4673-9005-7/16/$31.00 ©2016 IEEE.

[9] Serin Lee, Jong Soo Park" Top-k High Utility Itemset Mining Based on Utility-List Structure,, 978-1-4673-8796-5/16/$31.00 2016 IEEE

[10] J Prajakta R. Padhye, R. J. Deshmukh," A marketing solution for cross-selling by high utility itemset mining with dynamic transactional databases" 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), IEEE

[11] Junqiang Liu,, Ke Wang, Benjamin C.M. Fung" Mining High Utility Patterns in One Phase without Generating Candidates, 10.1109/TKDE.2015.2510012, IEEE

[12] Vincent S. Tseng,, Cheng-Wei Wu, Philippe Fournier-Viger, Philip S. Yu, *Fellow"* Efficient Algorithms for Mining Top-K High Utility Itemsets" 1041-4347 (c) 2015 IEEE