International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Novel Data Stream High Utility Itemset Miner with the Batch Transaction Processing Model

[1]Subba Reddy Meruva, [2]Dr. BonduVenkateswarlu

Submitted: 03/02/2024 **Revised**: 11/03/2024 **Accepted**: 17/03/2024

Abstract:High-utility mining techniques play a significant role in effectively finding the high utility itemsets (HUIs). These techniques aim to find the HUIs based on threshold values of minimum utility. Real-time applications, such as dynamic retail store transactions, continuous web data stream, and item updates in sensor network databases, need dynamic HUI mining techniques. Recently, an incremental mining-based high utility itemset (IM-HUM) was developed to handle the dynamic itemsets in the HUI mining process using incremental schedulers. It was primarily focused on processing HUI based on time frame schedulers rather than considering the amounts or size of items processed during the particular scheduler. It becomes tedious when a reasonable number of items cannot be processed in the prescribed schedule. For this reason, in the proposed work, the reasonable number of items in the data stream is defined by fixing the size of the batch of items instead of considering schedulers. The proposed data stream high utility miner is implemented using a batch model, say (DS-HUI-BM). It is superior to other state-of-the-art HUI mining techniques for both sparse and dense datasets, and the same is illustrated in the experimental section.

Keywords: *High Utility Itemset, Association Mining, HUI Mining, Data Stream, Batch Model*

1. Introduction

An essential data mining capability to define relationships between sets of items is called association mining [1]. In order to enhance strategies for marketing, sets of items that are either often used or attractive to customers are determined dependent on their associations. In order to find strong association rules, the mining method considers the frequency of item sets. Within the context of support and confidence, conventional association methods, such as FP-growth [2] and Apriori [3], are helpful for this data extraction approach. In order to do a market analysis, these methods would first identify the most in-demand products, defined as items that customers buy often or groups of products that consumers buy often. In particular situations, marketing applications may not require item sets with a specific frequency. Best practices for market enhancements should consider both the frequency and the profit (or usefulness) of the items or products. In the past few years, utility-based mining has led association mining research. When determining which sets of items were most helpful, these methods took two crucial elements into account: frequency and profit. These collections of items are commonly referred to as high-utility item sets (HUIs) [4]. Mining algorithms such as UP-Growth [5], HUI-Miner [6], and FHM (fastest high utility item set miner) may effectively extract high utility item sets based on analysis of item sets' utility and frequency. Instead of utilizing a support-confidence paradigm, the HUI Miner and the FHM depend on a utility-confidence schema [7]. These tools evaluate how the set of associated rules affects business profitability and the frequency of item sets. The first step of HUI Miner is to calculate utility lists, which are collections of item values.

Processing execution time can be improved by minimizing the search space using heuristic information from HUI Miner [8]. Before building a tree based on utility patterns (UP), another technique called UP-Growth calculates transaction utility (TU) and weighted transaction utility (TWU) [9]. A UP tree describes the final product. Finding groupings of items with high utility is made easy by using UP-tree. In order to extract high utility itemsets from candidate itemsets, the UPgrowth method employs a data structure known as a utility pattern tree. The implementation of utility lists, which comprised a group of objects with high utility, gave rise to the notion of FHM [9]. While it may not work well for dynamic databases, it is quicker at extracting collections of items with high utility. Extracting HUIs from dynamic databases is necessary for real-time applications. Fixing the schedulers in IM-HUI overcomes the dynamic updating problem for data stream databases. However, it was recognized as an effortless technique with an incremental schedule instead of considering the size of the items. At some scheduled times, there is a chance to get a smaller

[1]*Research Scholar, Department of Computer Science and Engineering, School of Engineering*
Dayananda Sagar University, Bangalore, India
E-mail: subbareddy.meruva@gmail.com
[2]*Associate Professor, Department of Computer Science and Engineering, School of Engineering*
Dayananda Sagar University, Bangalore, India
E-mail: bonduvenkat-cse@dsu.edu.in

number of items to get zero HUIs. Such cases impose less efficient HUI mining for the data stream databases. The proposed work is more attentive to addressing the HUIs extraction problem over the dynamic stream or incremental databases with the idea of the batch model. Real-time datasets are processed incrementally in the experimental section to analyze the performance of the proposed DS-HUI-BM method and for the comparative study. Highlights of the proposed work are summarized as follows:

1. Construct the specific utility data structure (uds) to extract HUI over the dynamic data stream. The 'uds' will support adding or removing batches.
2. An efficient mining method, DS-HUI-BM, was developed to use the utility data structure (uds) underlying the assumed batch model.
3. Experimental demonstrations are conducted to illustrate the efficacy of the proposed DS-HUI-BM model and comparative analysis with existing mining techniques.

2. Background Study Of The Work

Association rule mining is driven mainly by market-basket analysis, its primary motive. Mining for frequent itemsets [10] is an essential part of association mining since it allows for extracting frequent patterns or sets of items. Within most real-world applications, some items are found with high quantities (or frequency) yet low profit from investment. These items may not be interesting for the sake of market study. In order to keep the enormous quantity of items and obtain a poor profit with those items, the reason is to raise the investment in those items. The objective of the market strategy is to bring about a rise in the profit of the itemset and to guarantee that the itemset will be frequent and profitable. FIM-related algorithms are advantageous in this scenario, particularly for identifying the itemset that yields the most profits. These days, utility-based mining [11], [12] is a new activity that is being developed to extract the worth of an object, which may be helpful for HUI-based applications that are used in real life. With utility-based mining [13], one of the ultimate goals is to evaluate an object's vital information, including its usefulness and frequency. Utility is often used to allude to either profit or the number of units of the thing. It is possible to ascertain the frequency of an itemset by utilizing the amount of information inside the itemset. During the process of mining high-utility itemsets, low-utility stuff is not taken into consideration. Only during the generating process of high utility itemset mining is it discovered that the utility of the itemset should be met with the minimal utility threshold value. It is the only way that it can be discovered.

In the realm of research, mining high-utility item sets encompasses a wide variety of subjects; several algorithms are making progress in order extracting high-utility item sets. Some proposed algorithms, like UP-Growth, UP-Growth+, and HUI Miner, can do high-utility itemset mining effectively. Both UP-Growth and UP-Growth+ utilized the feature of transaction-weighted downward closure [14], [15] to extract HUIs through a two-step process. The first phase involves the generation of potential HUIs by overestimating utility values. In the second stage, the HUIs are determined based on the outcomes of the previous first step. These methods produce an enormous number of candidate HUIs [16], which increases the amount of processing time and memory required for the HUI identification process. As a result of the low utility threshold value, the performance of these two algorithms is significantly diminished. Extracting HUIs from the database requires them to scan the database several times. Through the construction of the utility list data structure, HUI Miner was able to accomplish this task.

The item or item's utility factor may be determined using this data structure, eliminating the need to scan the database repeatedly. In order to efficiently prune the search space, the tight upper bound of utility is defined as supersets containing the search space. However, excessive HUIs are still needed for some large data mining applications. The search space is home to many HUIs that demand significant processing time and memory usage. EFIM [17] and FHM [18] are two examples of HUIs extraction methods that use the tree data structure. Both of these methods fall under the latter group. In order to enhance the speed at which the join operation of the produced utility lists is performed, the revised successive algorithms mHUIMiner [19] and ULB Miner [20] suggested an effective data structure for the needed utility construction.

The mathematical formalism was an extension of basic Apriori and FP-Growth expansion concepts. When it comes to market analysis, the idea of HUI mining is the one that is suggested the most for improving earnings or smoothly reaching profit margins. The present state of the art indicates that the HUI mining algorithms are produced in two classes: algorithms that rely on creating candidates and other groups of algorithms that do not rely on candidate generations [22]. Both of these classifications are based on the current state of the art. Both of these are designed to be divided into two groups: Group A, which depends on the development of candidates, and Group B, which does not need the generation of candidates. The attribute of Transaction-Weighted-Downward, which was addressed previously in the same section, is adopted by the algorithms that belong to Group A. HUI Miner, HUP Miner, FHM, and EFIM are the algorithms that belong to Group B. These algorithms are responsible for discovering the HUIs without the production of

candidates. The HUI Miner method is the first algorithm in Group B that can detect HUIs without the requirement for candidate creation and repeated database scans. As a result, it is a more effective algorithm for extracting HUIs than the techniques used in Group A. In order to provide utility information for the goods, a specific data structure was used, which was identified as a utility list. Doing a single database scan to generate a utility list is sufficient. It is accomplished by conducting join operations on utility lists of items, which generates other item sets.

Furthermore, the overall time and memory efficiency for big transactional datasets is worsened since HUI Miner requires costly joint operations to resort to utility items with smaller item sets. This is the case for large datasets that contain many transactions. These HUI mining algorithms must be refined to handle the data stream kinds of transactional databases and efficiently extract HUIs. The proposed algorithm DS-HUI-BM describes the solutions well-explained in the following section.

3. Proposed Data Stream High Utility Itemset Using Batch Model (DS-HUI-BM)

Finding the utility of an item or itemset is the pre-determined step, and it can be computed by the composition (or multiplication) of the frequency of an item and the corresponding profit. It shows in Eqn. (1) The utility of an itemset in a particular transaction (T) is computed as per the following Eqn. (2).

$$\text{utility}(\text{item}, T) = \text{frequency}_{\text{item}} * \text{profit}_{\text{item}} \quad (1)$$

$$\text{utility}(\text{itemset}, T) = \sum_{\text{item} \in \text{itemset}} \text{frequency}_{\text{item}} * \text{profit}_{\text{item}} \quad (2)$$

Based on the minimum utility requirements (taking the threshold value, min_util), filter out (or mine) the high utility itemsets. Eqn. (3) and (4) show the computation of transaction utility, TU, and transaction-weighted utility (TWU)

$$\text{Transaction Utility (TU)} = \sum_{\text{item} \in T} \text{utility}(\text{item}, T) \quad (3)$$

$$\text{Transaction Weighted Utility (TWU)} = \sum_{\text{item} \in T \text{ and } T \in \text{Database}} \text{TU}(T) \quad (4)$$

The extraction high TWU is derived based on the satisfaction of the item's TWU with min_util

Our proposed method's main focus is to derive the best - k-high utility itemset from the transactional data stream databases. In the real-time social application, the transactions are updated continuously in such data stream databases. A batch with a fixed-size model is proposed for processing continuous transactions. Each batch defines a fixed number of transactions. Fixed-size batch batch_size is defined as a constant number of transactions processing in particular time frames. A single batch frame is considered for the experimental work in the implementation.

Table 1: Batch Wise Data Stream Transaction Table

Batches	Trans. No.	Frequency of Items in the Transaction	Transaction Utility (TU)
Batch1	TID1	(IA-1), (IC-1), (ID-1)	8
	TID2	(IA-2), (IC-6), (IE-2), (IG-5)	27
Batch2	TID3	(IA-1), (IB-2), (IC-1), (ID-6), (IE-1), (IF-5)	50
	TID4	(IA-1), (IB-4), (IC-3), (ID-3), (IE-2)	28
Batch3	TID5	(IA-1), (IB-2), (IC-2), (IE-1), (IG-2)	16
	TID6	(IB-2), (IC-2), (IE-1), (IG-2)	11

Table 2: Utility (or Profit) Table of Items

Transaction Item	IA	IB	IC	ID	IE	IF	IG
Utility Value	5	2	1	2	3	5	1

Table 3: TWU values in the First Sequence of Batch 1 and Batch 2 and in another sequence of Batch 2 and Batch 3 (here max batches should be 2)

Transaction Item	TWU in the First Sequence of Two Batches- B1 & B2	TWU in the Second Sequence of Batches– B2 & B3
IA	113	94
IB	78	105
IC	113	105
ID	86	78
IE	105	105
IF	50	50
IG	27	27

Table 4: uds data structure for 1-itemset iutil and rutil values of items in the First Sequence of Batch 1 and Batch 2 and in another sequence of Batch 2 and Batch 3 (here max batches should be 2)

Transaction and Batch	**iutil and rutil values in the First Sequence of Two Batches- B1 & B2**		**iutil and rutil values in the Second Sequence of Two Batches – B2 & B3**	
TID1 in Batch1	**iutil**	**rutil**	**iutil**	**rutil**
TID 2 in Batch1	5 (1*5) {IA, IC}	1(1*1)	-	-
TID3 in Batch2 & Batch2	10 (2*5) {IA, IC}	6(6*1)	-	-
TID4 in Batch2 & Batch3	5(1*5) {IA, IC}	1(1*1)	5*1{IA, IB,IC ,IE}	2*2+ 1*1+1*3 =8
	5	3(3*1)	5*1{IA,IB, IC,IE }	4*2+3*1 +2*3=17

For every arrival of the new batch, the existing batch should be eliminated from batch processing. For every new arrival of a batch of transactions, the top HUIs are updated dynamically to address the problem of HUIs extraction in the transactional data stream databases. The algorithm descriptions as follows:

Algorithm DS-HUI-BM

Input: batch_size [ ], uds (utility data structure), Bt_i, DT, UT

// where batch_size[ ] refers to the number of transactions to be processed in which batch_size[0] is the number of transactions in first batch, batch_size[1] is the number of transactions in second batch,…. so on,

// uds is the utility data structure

// Bt_i is the batch number 'i'

// DT is the data stream transaction table

// UT is the utility (or profit) table.

Methodology:

1. Finds the transaction utility using the Eqn. (3) and stores the values in the transaction utility table (TUT)
2. Define the batch model with the array values of batch_size[ ]
3. bn=count(batch_size)
4. Set i=0
5. while (i<bn)
   a. b_value=batch_size[i]
   b. Read the transaction numbers and store the transaction numbers into the set 'Bt_i' // here; the number of reading transactions should be b_value
   c. Find the TWU for the transactions of Bt_i and order the items for the transactions of Bt_i
   d. Construct the 'uds' data structures by computing the initial and rutile values for the specified batches of transactions.
   e. Update the uds for the itemset size from 2,3, ….. until to get the no supersets of items
   f. Eliminate the super itemsets whose (iutil+rutil) value should be less than the threshold value of min_util
6. Derive the HUIs after eliminating the itemsets based on the sum of utility values in the 'uds' for the batches of transactions.

In this algorithm, the construction of uds is essential for the HUIs extraction. The summary of uds consists of transaction number, iutil, and rtutil values. In the first step, it is required to describe the recognized transaction that consists of ordering items in ascending order according to the TWU values for the specified batch of transactions. The iutil denotes the profit or utility of items in the described recognized transaction. Another critical parameter is rutil, and it denotes the remaining part of the utilities of the item in the recognized transaction. Tables 1 to 4 illustrate the proposed algorithm with the corresponding example. It shows the dynamic upgrading of uds data structure using the batch model. Here, three batches are considered, which allows the maximum number of batches at a time is 2. Dynamic updating of uds is performed to compute the sum of {iutil, rutil}, and it should be satisfied with a given minimum utility threshold value. Table 4 shows the iutil and rutil values for the different transactions in the batch model. Based on the satisfaction of the minimum utility threshold value, the 1-itemset high utility itemsets are extracted in the corresponding transaction of batch model execution. The high utility itemsets are extracted using the constructed

novel data structure is uds. The 'uds' data structure is created for the 2-itemsets and 3-itemsets….until the null supersets are found. These ads list values give the util and retail values for the various itemsets that are useful for describing the high utility itemsets.

4. Experimental And Comparative Analysis

The experiments are carried out using the system configurations of the i7 processor, 16GB RAM, and the JDK. Three real-time datasets are taken: retail, foodmart, and bms and collected from the repository of [23]. Three existing methods, namely FHM, HUI Miner, EFIM, and our proposed method, DS-HUI-BM, are experimented with using the retail, foodmart, and bms datasets

Table 5. Computational Time and Memory Analysis of the Proposed and Existing Methods

HUI Mining Method	**Min_Utility Value**	**Runtime in milliSeconds(ms) Memory Requirement in MB High Utility Itemsets Count**	**Memory Requirement in MB**	**High Utility Itemsets Count**
Retail				
FHM	100000	1483 ms	575.666	22
HUI Miner	100000	879 ms	398.637	22
EFIM	100000	1240 ms	297.137	22
DS-HUI-BM	100000	700 ms	85.7156	22
FHM	200000	997 ms	102.866	9
HUIMIner	200000	670 ms	313.736	9
EFIM	200000	1211 ms	198.161	9
DS-HUI-BM	200000	539 ms	26.8237	9
FHM	300000	917 ms	145.076	2
HUIMIner	300000	1064 ms	185.689	2
EFIM	300000	994 ms	89.1134	2
DS-HUI-BM	300000	755 ms	62.3692	2
Foodmart				
FHM	10000	324 ms	26.941894531125	428
HUIMIner	10000	789 ms	21.065711975097656	428
EFIM	10000	624 ms	25.489997863769 53	428
DS-HUI-BM	10000	145 ms	16.432304382322422	428
FHM	15000	219 ms	32.183860778808594	89
HUIMIner	15000	587 ms	47.8653564453125	89
EFIM	15000	482 ms	18.45116424560547	89
DS-HUI-BM	15000	86 ms	15.450599670410156	89
FHM	20000	183 ms	48.90404510498047 MB	12
HUIMIner	20000	584 ms	61.846061706542 97 MB	12
EFIM	20000	331 ms	24.099388122558594	12

DS-HUI-BM	20000	70 ms	17.2105712890625 MB	12

BMS				
FHM	2500000	745 ms	75.978515625	5
HUIMIner	2500000	1240 ms	68.8626708984	5
EFIM	2500000	1314 ms	43.501953125	5
DS-HUI-BM	2500000	340 ms	34.940353393	5
FHM	3000000	215 ms	78.12886047363281 MB	3
HUIMIner	3000000	293 ms	58.5097274780273	3
EFIM	3000000	458 ms	49.000205993652344	3
DS-HUI-BM	3000000	181ms	22.554771423339844	3
FHM	3200000	327 ms	108.09402465820312	2
HUIMIner	3200000	215 ms	70.29513549804675	2
EFIM	3200000	274 ms	52.77223205566406	2
DS-HUI-BM	3200000	195 ms	32	2

The comparative analysis of HUI mining methods is shown in Fig. 1 and Fig. 2 to illustrate the efficiency concerning the parameters of memory and computational time analysis, respectively.

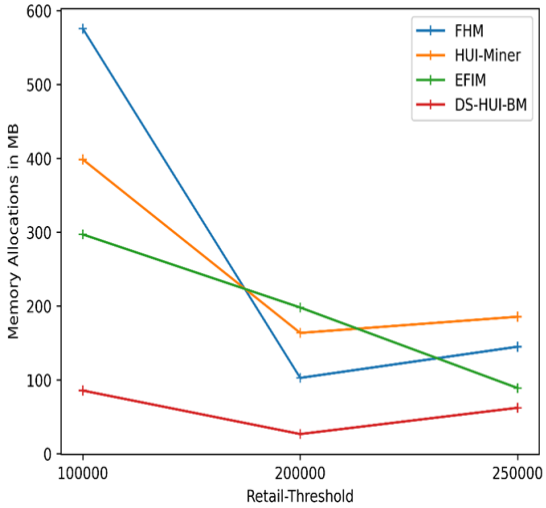

Fig. 1 Memory Comparative Analysis of HUI Mining Methods Using the Retail Datasets

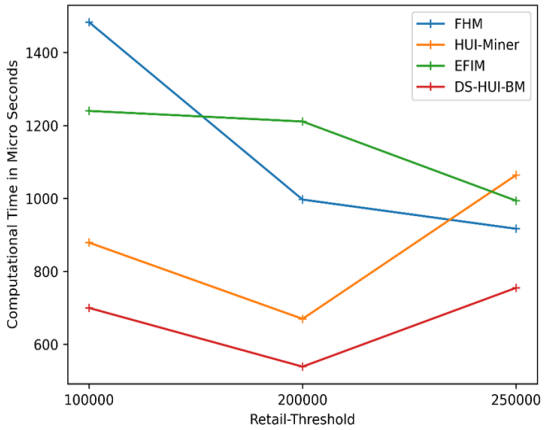

Fig. 2 Computational Time Comparative Analysis of HUI Mining Methods Using the Retail Datasets

The batches of transactions are applied under the assumption of the retail data stream dataset. Two batches of 50 transactions are processed in the experimental stage for the underlying assumption of three datasets: retail, foodmart, and bms. Other two datasets (i.e., foodmart and bms) of computational time and memory analysis are visualized from Fig. 3 to Fig. 6.

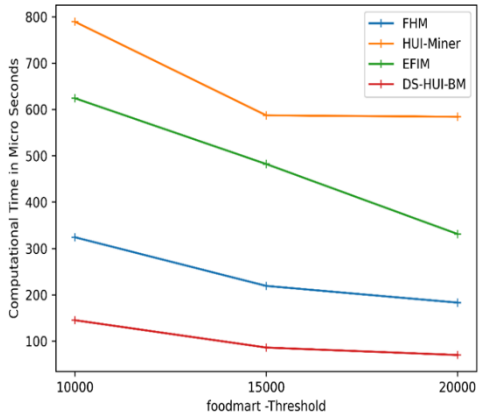

Fig. 3 Computational Time Comparative Analysis of HUI Mining Methods Using the Foodmart Datasets

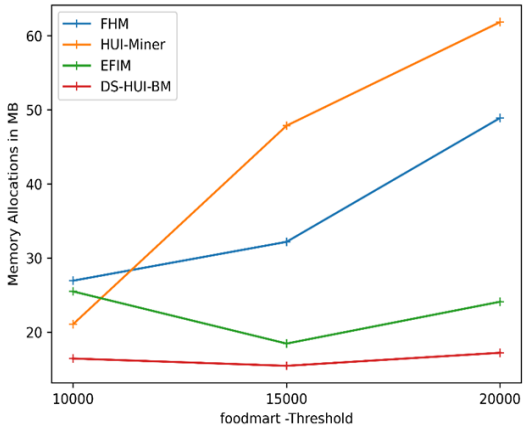

Fig. 4 Memory Comparative Analysis of HUI Mining Methods Using the Foodmart Datasets

Different minimum utility thresholds are taken for each dataset to analyze the computational time and memory analysis, which are shown in the x-axis of line graphs. It was observed that the number of HUI extractions remains the same for any minimum utility threshold.

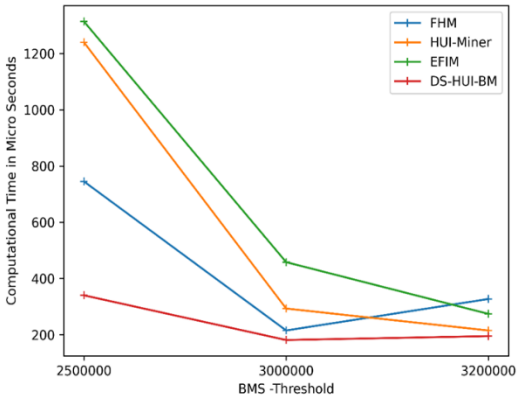

Fig. 5 Computational Time Comparative Analysis of HUI Mining Methods Using the BMS Datasets

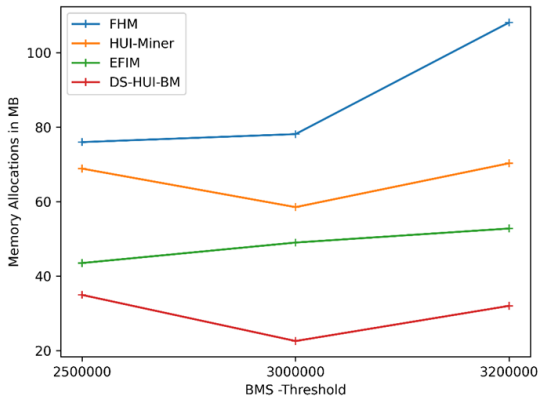

Fig. 6 Memory Comparative Analysis of HUI Mining Methods Using the BMS Datasets

Using the proposed batch models for the dynamic transaction datasets, high utility item sets were achieved using minimal computational time and memory allocations. For the retail datasets, it was observed that the proposed method is 6 to 7 times faster than other HUI mining methods for the extraction of high-utility itemsets. The memory allocation takes nearly 100 units; for others, the memory is between 200 and 600 units; for the Foodmart and BMS datasets, the same experimental observations are made. It indicates that the proposed DS-HUI-BM models outperformed the others in extracting data stream transactional datasets.

5. Conclusion And Future Scope

The batch model of the HUI mining technique is proposed for extracting HUIs for the data stream transactional datasets. In the proposed work, initially, the value for the parameter of batch_size is set. It indicates the number of batches that need to be processed for the transactions. In this experiment, 50 transactions are taken for every batch in the proposed work. Three real-time datasets are used, and the transaction items are processed as per the batches in the DS-HUI-BM. The proposed approach outperformed competing HUI mining techniques by a factor of 6–7 when extracting high-utility itemsets from retail datasets. It also used roughly 100 units of memory for allocation, whereas the others used 200 to 600 units. Identical experimental findings are also obtained for the Foodmart and BMS datasets. It shows that the proposed DS-HUI-BM models performed better for extractions than the others in data stream transactional datasets. The future scope of the work is to address the scalability of a problem for real-time web-related datasets.

References

[1] Chee, CH., Jaafar, J., Aziz, I.A. *et al.* Algorithms for frequent itemset mining: a literature review. *Artif Intell Rev* **52,** 2603–2621 (2019). https://doi.org/10.1007/s10462-018-9629-z

[2] Meruva, Subba Reddy and Venkateswarlu Bondu. "Review of Association Mining Methods for the Extraction of Rules Based on the Frequency and Utility Factors." *IJITPM* vol.12, no.4 2021: pp.1-10. http://doi.org/10.4018/IJITPM.2021100101

[3] D. Chen, S. L. Sain and K. Guo, "Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining", *J. Database Marketing Customer Strategy Manage.*, vol. 19, no. 3, pp. 197-208, Sep. 2012.

[4] J. Han, J. Pei and Y. Yin, "Mining frequent patterns without candidate generation", *Proc. ACM SIGMOD Int. Conf. Manage. Data*, vol. 29, pp. 1-12, 2000.

[5] Nguyen L.T.T., Mai T., Vo B. (2019) High Utility Association Rule Mining. In: Fournier-Viger P., Lin JW., Nkambou R., Vo B., Tseng V. (eds) High-Utility Pattern Mining. Studies in Big Data, vol 51. Springer, Cham. https://doi.org/10.1007/978-3-030-04921-8_6

[6] Tseng et al., 2013, V.S. Tseng, B.E. Shie, C.W. Wu, P.S. Yu**,** Efficient algorithms for mining high utility itemsets from transactional databases, IEEE Trans. Knowl. Data Eng., 25 (2013), pp. 1772-1786

[7] Tseng, V.S., Wu, C.-W., Shie, B.-E., Yu, P.S.: Up-growth: an efficient algorithm for high utility itemset mining. In: ACM SIGKDD, pp. 253–262. ACM (2010)

[8] Tseng VS, Shie BE, Wu CW, Philip SY (2012) Efficient algorithms for mining high utility itemsets from transactional databases. IEEE Trans Knowl Data Eng 25(8):1772–1786

[9] Lin JCW, Gan W, Hong TP (2016) Maintaining the discovered high-utility itemsets with transaction modification. Appl Intell 44(1):166–178

[10] K. Rajendra Prasad (2017), Optimized high-utility itemsets mining for effective association mining paper, International Journal of Electrical and Computer, Vol. 7, Issue. 5, pp: 2911-2918

[11] T.-P. Hong, C.-H. Lee and S.-L. Wang, "Effective utility mining with the measure of average utility", Expert Syst. Appl., vol. 38, no. 7, pp. 8259-8265, Jul. 2011.

[12] B. Vo, L. T. T. Nguyen, N. Bui, T. D. D. Nguyen, V. -N. Huynh and T. -P. Hong, "An Efficient Method for Mining Closed Potential High-Utility Itemsets," in IEEE Access, vol. 8, pp. 31813-31822, 2020, doi: 10.1109/ACCESS.2020.2974104.

[13] R. J. Hilderman, C. L. Carter, H. J. Hamilton and N. Cercone, "Mining market basket data using share measures and characterized itemsets", Proc. PAKDD, pp. 159-173, 1998.

[14] Chan R, Yang Q, and Shen Y-D (2003) Mining high utility itemsets, in IEEE International Conference on Data mining, pp. 19–26

[15] Krishnamoorthy S (2019) A comparative study of top-K high utility itemset mining methods. High-Utility Pattern Mining, pp 47–74

[16] Liu, J., Wang, K., Fung, B.C.: Direct discovery of high utility itemsets without candidate generation. In: Proceedings of the 12th IEEE International Conference on Data Mining, pp. 984–989. IEEE (2012)

[17] Zida, S., Fournier-Viger, P., Lin, J.CW. et al. EFIM: a fast and memory efficient algorithm for high-utility itemset mining. Knowl Inf Syst 51, 595–625 (2017). https://doi.org/10.1007/s10115-016-0986-0

[18] Using Length Upper-Bound Reduction. In: Fujita, H., Ali, M., Selamat, A., Sasaki, J., Kurematsu, M. (eds) Trends in Applied Knowledge-Based Systems and Data Science. IEA/AIE 2016. Lecture Notes in Computer Science(), vol 9799. Springer, Cham. https://doi.org/10.1007/978-3-319-42007-3_11

[19] Li, HF., Huang, HY. & Lee, SY. Fast and memory efficient mining of high-utility itemsets from data streams: with and without negative item profits. Knowl Inf Syst 28, 495–522 (2011). https://doi.org/10.1007/s10115-010-0330-z

[20] Fageeri, S.O., Hossain, S.M.E., Arockiasamy, S., Al-Salmi, T.Y. (2022). High-Utility Pattern Mining Using ULB-Miner. In: Aurelia, S., Hiremath, S.S., Subramanian, K., Biswas, S.K. (eds) Sustainable Advanced Computing. Lecture Notes in Electrical Engineering, vol 840. Springer, Singapore. https://doi.org/10.1007/978-981-16-9012-9_17

[21] Liu Y., Liao W., Choudhary A. (2005) A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets. In: Ho T.B., Cheung D., Liu H. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2005. Lecture Notes in Computer Science, vol 3518. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11430919_79

[22] Subba Reddy Meruva, Dr. Bondu Venkateswarlu, Tree Integrated High Utility Miner for Improving an Efficiency of Association Mining, Vol., Vol. 83, 2020, pp:15938-15946

[23] E. Hikmawati, N. U. Maulidevi and K. Surendro, "Pruning Strategy on Adaptive Rule Model by Sorting Utility Items," in IEEE Access, vol. 10, pp. 91650-91662, 2022,doi: 10.1109/ACCESS.2022.3202307.

[24] S. R. Meruva and B. Venkateswarlu, "A Fast and Effective Tree-based Mining Technique for Extraction of High Utility Itemsets," 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 1393-1399,doi: 10.1109/ICECA55336.2022.10009213