

PAPER • OPEN ACCESS

High Utility Mining of Streaming Itemsets in Data Streams

To cite this article: Abdullah Bokir and V B Narasimha 2021 *J. Phys.: Conf. Ser.* **1962** 012027

View the [article online](#) for updates and enhancements.



240th ECS Meeting

Oct 10-14, 2021, Orlando, Florida

**Register early and save
up to 20% on registration costs**

Early registration deadline Sep 13

REGISTER NOW



High Utility Mining of Streaming Itemsets in Data Streams

ABDULLAH BOKIR^{1,2,*}, Dr V B NARASIMHA³

¹Research Scholar, Dept of Computer Science & Engineering, UCE, OU Hyd-India.

²Hadhramout University, Mukalla-Yemen.

³Assistant Professor, Dept of Computer Science & Engineering, UCE, OU Hyd-India.

*Corresponding author's e-mail: bokir102@gmail.com

Abstract. The traditional models for mining frequent itemsets mainly focus on the frequency of the items listed in the respective dataset. However, market basket analysis and other domains generally prefer utility obtained from items regardless of their frequencies in the transactions. One of the main options of utility in these domains could be profit. Therefore, it is significant to extract items that generate more profit than items that occurs more frequently in the dataset. Thus, mining high utility itemset has emerged recently as a prominent research topic in the field of data mining. Many of the existing researches have been proposed for mining high utility itemset from static data. However, with the recent advanced technologies, streaming data has become a good source for data in many applications. Mining high utility itemset over data streams is a more challenging task because of the uncertainty in data streams, processing time, and many more. Although some works have been proposed for mining high utility itemset over data streams, many of these works require multiple database scans and they require long processing time. In respect to this, we proposed a single-pass fast-search model in which we introduced a utility factor known as utility stream level for tracing the utility value of itemsets from data streams. The simulation study shows that the performance of the proposed model is more significant compared with the contemporary method. The comparison has been performed based on metrics like process-completion time and utilized search space.

Keywords: High Utility Pattern Mining, High utility streaming itemset mining, Data Stream, Stream Level Utility, Window Level Utility.

1. INTRODUCTION

Data mining is the process of extracting useful and hidden information from massive datasets, which is very difficult to be extracted manually. In the past decades, many types of programmatic data mining methods like association rule mining (ARM), classification, and clustering have been proposed.

Association rule mining is one of the most well-known tasks in data mining which is mainly to find the rules that associate items from the given dataset. It has a wide range of application domains such as market basket analysis, recommendation systems, web click analysis, and many more. Association rule mining can be done based on other core tasks such as frequent pattern mining, sequential pattern mining, or high utility pattern mining. Among them, frequent pattern mining has been used extensively to perform association rule mining.

Frequent pattern mining is to find sets of patterns that occur frequently in the dataset. It assumes that the quantity of each item in a transaction can be either zero or one and all items are equally the same.



Due to this frequent itemset mining is not sufficient in many cases and hence the idea of high utility itemset mining came into the picture.

High utility itemset mining takes into consideration the quantity of each item in a transaction and it assumes that items are not equally the same, some items are more important because they contribute more to the respective organization and they can be called high utility itemset.

The works [1], [2], [3], [4] presented that Association Rule Mining (ARM) has extensively utilized for detecting meaningful item-sets from transactions of the given databases.

The ARM has numerous methods: sequential pattern mining (SPM), Utility Pattern Mining (UPM), & Frequent Pattern Mining (FPM). The works [5], [6], [7] presented that FPM is the most common method in the domain of ARM. It easily detects frequently occurring patterns in the databases. Although this method was used in various data mining implementations, it's not appropriate for examining databases with a non-binary quantity of the items, because; this method presumes that the quantity of every item in the transactions could be either 0 or 1.

Contrarily, in UPM, such type of utility information (non-binary data) of the items in the databases could be utilized appropriately to define the pattern's significance. Hence, this UPM has been used by analysts of data in various domains that handle voluminous data with maximum intricacies.

The works [6], [8], [9], [10], [11] presented that the conventional UPM method is High-UPM (HUPM) which assessing the utilities of patterns by the measure of utility that summates the utilities of an item in the patterns. Nevertheless, this method might suffer from a huge amount of pattern generation with extended lengths due to the aggregation of item-set utilities, usually large when the length of the pattern is long. Here, for better utility mining, the high average- UPM (HAUPM) [12], [13] was researched. This method reflects the lengths of patterns into respective patterns utilities to measure the utilities correctly compared to the conventional HUPM method.

Contemporarily, the processing of time-sensitive data stream is a significant problem because current data could be more significant in discovering important patterns than time-elapsing data. For example, in a medical database, the current patient's records are more significant for examining the present medical trends than previous old records. The time-sensitive information is required to deal with the data mining models with the deliberation of time parameters. The former contributions on the HAUPM have basic confines in examining such type of time-sensitive information. Motivated by the above-stated issues, a new algorithm of utility-based itemset mining known as High utility streaming itemset mining (HUSIM) has been proposed in this paper.

2. RELATED WORK

One of the main challenges in utility-based mining models is how to prune the search space because high utility patterns are not following the downward closure property which was used to prune the search space in models based on frequent pattern mining. This property states that if an itemset is frequent then all its supersets will be frequent as well. To address this problem the authors in [14] proposed a 2-phase algorithm for high utility itemset mining. As the superset of a high utility itemset might or might not be a high utility itemset, which means there would be no property of downward closure for high utility itemset. In this work, a new property called Transaction Weighted Utility (TWU) was proposed. In the first phase, the 2-phase algorithm implemented an algorithm called Apriori [15] for detecting entire sets of items that satisfy the new property (TWU) in the form of candidate-itemsets, and later in the second phase, the algorithm will scan database transactions for computing utility towards every candidate itemset to detect which candidates itemsets are high utility itemset. Even though the two-phase algorithm detects overall high utility itemsets from a transactional database, the processing time due to more candidate generation in the first phase is critical.

The work [16] presented that the former methods for frequent itemset mining which are based on Apriori, generate candidate itemsets iteratively from the k-frequent item-sets where the k value is not less than 1 and then verified if these candidate itemsets were frequent or not.

In the case of huge input datasets or a small value of minimum support threshold, algorithms like Apriori might suffer from scanning the database multiple times, and repeatedly searching from a huge amount of candidate itemsets.

To overcome the limitations of Apriori-based approaches which require multiple scans through the original dataset, and generate a huge set of candidate itemsets, the work [17] shows that tree-based algorithms can avoid multiple database scans and generate fewer candidate itemset. They work by converting the original database transaction into an FP tree and then traverse the generated tree to extract frequent itemset.

Though tree-based algorithms can detect frequent itemset from a transaction database effectively, the memory usage and execution time were relatively high.

Hence, some of the researchers have proposed the closed itemsets concept like in [18]. Here, the number of closed-frequent-itemsets (CFI) is often much lower than the complete set of frequent items of the transaction, where entire frequent itemsets could be derived from CFI. Hence, either memory usage or the time of execution for mining CFI is lower than traditional frequent itemset mining.

The researchers in [19], [20] proposed different methods for High-Utility-Itemset-Mining from a static database. These methods could not discover high utility itemsets effectively from data streams, as they require reconstructing the entire set of high utility itemset and rescanning the original database each time a transaction is added to or removed from the data streams. Instead of that, they need to immediately detect which itemsets emerged as high utility itemset and which one has to be removed.

Researchers in [21],[22] designed methods for mining high utility itemset from the data stream. In [21], a tree-based approach was proposed, while Apriori based method was proposed in [22]. Nevertheless, these methods attempted to detect high utility itemset from candidate high transaction weighted utility itemsets(HTWU), they take a huge amount of time searching high utility itemsets and rescanning the original database multiple times because they are not utilizing former information of candidate itemsets. Hence, effective HUI mining in the data stream has been proposed. When transactions were eradicated or added, algorithms might upgrade the HTWU itemsets as per the eradicated or added transactions and compute the HTWU item-sets directly without searching HUI from HTWU item-sets and rescanning database.

Temporal High Utility Itemset(THUI) mining algorithm was proposed in [23] to discover the utility-based itemsets from data streams. The algorithm will first generate HTWU itemset of length 2-items and later it will discover all HTWU itemset by implementing a two-phase approach. When a new transaction is added, the algorithm will check if it contains some new items and it will check if those newly added items satisfy the utility threshold or not. If the items in the newly added transaction already exist before, then the algorithm will check if they still satisfy the utility threshold. Because the THUI mining algorithm utilizes the 2-phase approach, it requires multiple database scans and takes a long processing time. HUP-HUI-DEL algorithm was proposed in [24], which is also based on the two-phase model, it considers the deletion of the transaction, this algorithm generates a huge candidate HUI set and it requires multiple scans of the dataset.

The authors in [25], proposed three tree structures for interactive and incremental HUP mining. The first tree structure sorts the items in lexicographical order, when data incremented the algorithm will extract HUP without rebuilding the tree. The second tree structure arranges items according to their transaction frequency. The third tree structure sorts the items based on their TWU. Although these tree structures are interactive, they still require huge memory space to store the data in the tree and they need to implement FP-Growth [26] to generate the sub-tree structures repeatedly. They also require a minimum of two database scans to generate HUI from the candidate itemset.

SHU-Growth algorithm [27], and HUPID-Growth algorithm [28] were proposed by Yun and Ryang. The proposed HUPID-Growth requires a single database scan for constructing TIList and HUPID-Tree. The SHU-Growth utilizes the IHUP tree structure and stores gathered utility for every node when transactions set are added. The SHU-Growth implements the UP-Growth algorithm [29] for reducing the over-estimated utility value and count of candidate HUIs. The SHU-Growth & UPID-Growth still

implement the FP-Growth algorithm for detecting HTWU itemsets, and HUIs are searched from a huge amount of candidate HUIs.

In [30], the authors proposed a single-phase algorithm for mining high utility itemsets without candidate generation. The algorithm uses an inverted list data structure to store the utility of the items. Although the algorithm was able to avoid the candidate generation phase, it assumes the ordering of items does not change across windows which is not a practical condition in real-time scenarios.

The idea of high utility itemset mining has been applied recently in many real-time applications such as in [31,32]. The authors of [31], applied the concept of high utility pattern mining to extract emerging topics from Twitter, they used a tree-based structure to store the utility information of the words from given tweets. The researchers in [32], combined the idea of high utility pattern mining and Aspect based sentiment analysis for extracting groups of product features that produce a high profit.

The contemporary model HUIStream has been proposed in [33] to discover the High utility itemsets from the given data streams. This contemporary HUIStream model finds the utility itemsets from each transaction in a sequence of streaming data and updates the utility of buffered itemsets after each transaction of the target stream.

3. METHODS AND MATERIALS

This section explores the proposed High utility Streaming Itemset Mining. For each window of the streaming records, the proposed model extracts high utility streaming itemsets of size one and lists them as a set *sis*. Meanwhile, for each itemset *e* in the list *sis* it prepares a list of positive records(*pt_e*) i.e list of records in which itemset *e* exists. The algorithm will use the generated list of high utility itemset of size one to generate itemset of size two and above by performing a union operation $e = \{e_i \cup e_j\}$ for all itemsets e_i, e_j /where $i \neq j$. Further, if the resultant streaming itemset *e* does not exist $\{e \notin sis\}$ in the list *sis*, and the set $pt_e = pt_{e_i} \cap pt_{e_j}$ of positive transactions of the streaming itemset *e* is not empty, add the streaming itemset *e* to the list *sis* and prepare the set $pt_e = pt_{e_i} \cap pt_{e_j}$ representing the positive transactions of the corresponding streaming itemset *e*. This process is recursive until the task of buffering the records of streaming data is completed. The steps performed by the proposed model are explored in the following section.

4. SINGLE-PASS FAST-SEARCH HIGH UTILITY STREAMING ITEMSET MINING

To achieve a single scan of streaming data, in our proposed model we used set operations (union, intersection, minus) over list structure. The steps performed in our model are:

STEP1: Finding itemset of size one and calculate record utility and window utility

For each given window *w*, find all items that exist in each record *r_i* of that window and list them as set *sis*.

$$\begin{aligned} & |w| \\ & \forall \{r_i \exists r_i \in w\} \\ & i = 1 \\ & sis(w) = sis(w) \cup r_i \end{aligned}$$

Meanwhile, calculate the utility of each item *i* in *sis* and utility of each record *r_i* in *w* as following :

$$riu_r(i) = q(i) * g(i) \quad (1)$$

In equation(1) above, the record level utility of item i in a record r ($riu_r(i)$) is the multiplication of its quantity in that record $q(i)$ and its gain $g(i)$.

$$wiu(i) = \frac{1}{|w|} \sum_{j=1}^{|w|} \{riu_j(i) \exists r_i \in w\} \quad (2)$$

From equation(2) above, the window level utility of each item i in sis ($wiu(i)$) is equal to the empirical probability of the record level utility of that item.

$$ru(r) = \sum_{p=1}^{|r|} \{riu_r(i), \forall i \in r\} \quad (3)$$

In the above equation(3), for a given record r , the record utility ($ru(r)$) is the cumulative sum of the utility observed for all items which exist in that record.

$$wu(w) = \sum_{p=1}^{|w|} \{ru(r_p) \forall r_p \in w\} \quad (4)$$

In the above equation(4), for a given window w , the window utility $wu(w)$ is the cumulative sum of the record utility $ru(r)$ observed for all records which exist in that window.

$$u\tau(w) = wu(w) * \tau \quad (5)$$

In the above equation(5), the Utility threshold of a window w ($u\tau(w)$), is the absolute product of the window utility $wu(w)$ of window w and given threshold constant τ .

STEP2: Finding lists of positive transactions for all streaming items of the window.

$pt(sis_i)$: list of all transactions in which item sis_i exists.

STEP3: Finding the streaming itemset of size two and above and buffer the window

Loop1: for each itemset sis_i in $sis(w)$ Begin:

 Loop2: for each item sis_j in $sis(w)$ Begin:

$sis_k \leftarrow (sis_i \cup sis_j)$

 if $sis_k \notin sis(w)$ AND $wiu(sis_k) > u\tau(w)$ then

$pt(sis_k) = pt(sis_i) \cap pt(sis_j)$

$sis \leftarrow sis_k$

 End if

 End Loop2

End Loop1

STEP4: calculate stream utility threshold of all buffered windows

$$wu\tau = \frac{\sum_{i=1}^{|w|} u\tau(w_i)}{|w|} \quad (6)$$

The above equation(6) is to find the empirical probability of the window utility threshold of all buffered windows.

$$d_{wut} = \frac{\sum_{i=1}^{|w|} |wut - ut(w_i)|}{|wut|} \quad (7)$$

Equation(7) is to find the absolute mean deviation of the window utility threshold of all buffered windows.

$$sut = wut - d_{wut} \quad (8)$$

Equation(8) is to calculate the Stream utility threshold of all buffered windows.

STEP5: finding top-k high utility streaming itemset

Loop for each streaming itemset $sis_i \in sis$ Begin:

$$siu(sis_i) = \frac{1}{|w|} \sum_{j=1}^{|sis|} wiu_j(sis_i) \quad (9)$$

$$if (siu(sis_i) \geq sut) \text{ then}$$

$$husi \leftarrow sis_i$$

End Loop.

In the above equation(9), The stream level utility of an itemset sis_i ($siu(sis_i)$) is the empirical probability of the window level utility of the corresponding streaming itemset observed so far for all buffered windows. If the stream level utility of the itemset sis_i is greater than or equal to stream utility threshold sut then move that itemset to the list of high utility streaming itemset $husi$.

Further, sort the list $husi$ by context factor such as the size of the itemset, stream level utility of the itemset ... etc., and then select the top-k itemset of the sorted list.

5. EXPERIMENTAL EVALUATION

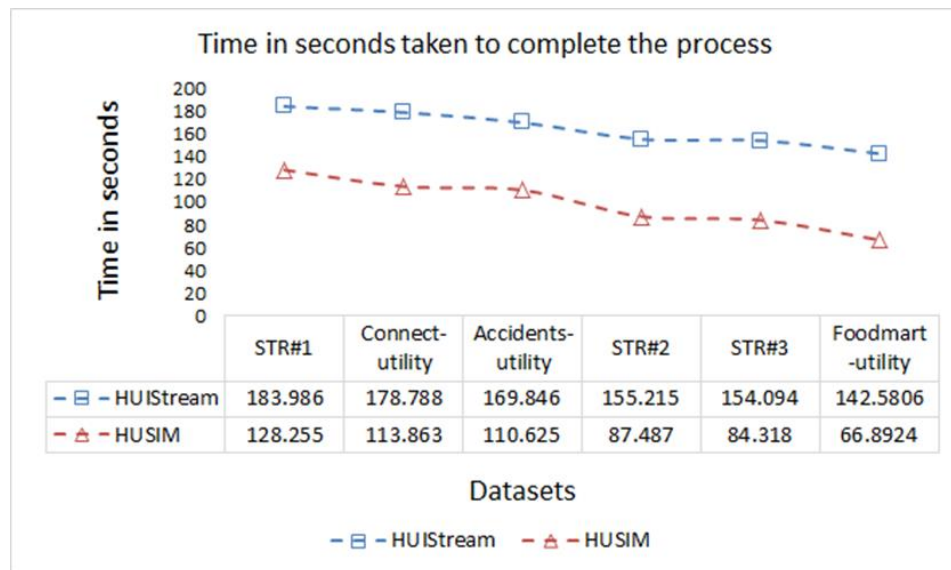
The experimental study was performed on streaming data sourced by the datasets acknowledged and accepted in [34]. Besides, the significance of the proposed method against continuous streaming data was tested using synthetic data generated by the IBMGenerator [35]. The benchmark datasets "connect-utility," "accident-utility," and "food mart-utility," along with the other three data streams labeled as STRM#1, STRM#2, and STRM#3, which have been synthesized using IBMGenerator are considered for experimental study.

The proposed HUSIM is scaled by comparing it with the contemporary model HUIStream [33]. The two critical objectives, namely, the search space and computational overhead regarding utility itemset mining, are considered to assess the performance of the proposed HUSIM & other contemporary model HUIStream.

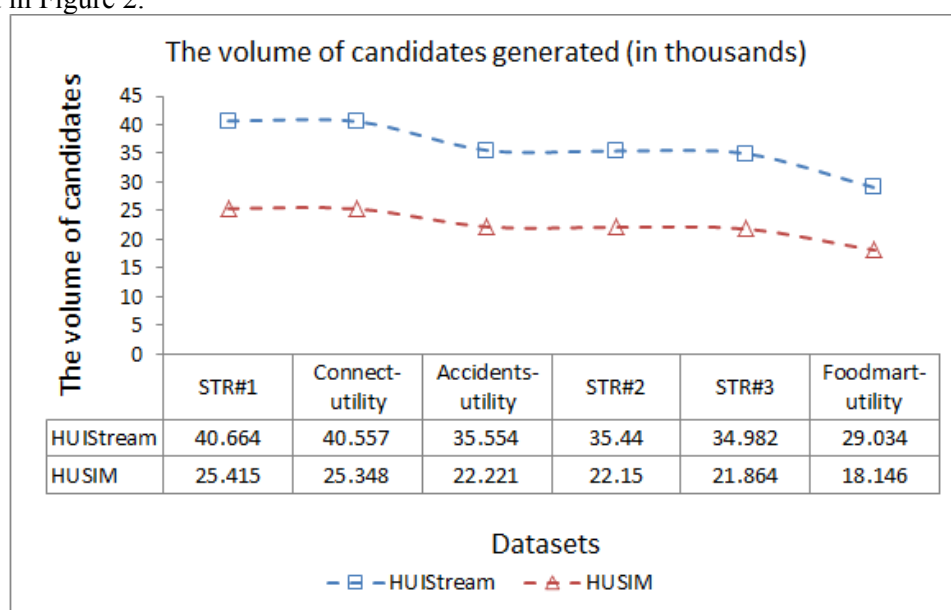
The performance metric Computational overhead has been estimated as the process completion time and candidate generation load. The time is taken to complete the utility mining process of the models HUSIM and HUIStream have listed and visualized in Figure 1. The comparison of the average process completion time of the proposal and contemporary model have been observed over both synthetic and real data sets evincing that HUIStream is reflecting an average of 30% additional process time compared to the processing time taken by the proposed model HUSIM.

Table1: average process completion time of HUSIM and HUIStream.

	STR#1	Connect-utility	Accidents-utility	STR#2	STR#3	Foodmart-utility
HUIStream	183.986	178.788	169.846	155.215	154.094	142.5806
HUSIM	128.255	113.863	110.625	87.487	84.318	66.8924

**Figure 1:** The time in seconds taken to complete the process by HUSIM, and HUIStream on different datasets.

The candidate generation observed for HUSIM is more than 50% fewer than the candidates generated by HUIStream. The candidates count observed from the proposed, and contemporary models have been visualized in Figure 2.

**Figure 2:** The count of candidates generated by HUSIM and HUIStream.

Search space usage about utility itemset mining is the aggregate space required to manage the candidates generated and the count of input records used in this regard. Hence it is obvious that the proposed model HUSIM performs the process under minimal search space compared to other contemporary models. Here, since the count of input records loaded and candidates' count, those generated by HUSIM are significantly fewer than the HUIStream. The empirical study denoting that the average search space utilized by HUIStream is 20% more than the search space utilized by the HUSIM (see Figure 3).

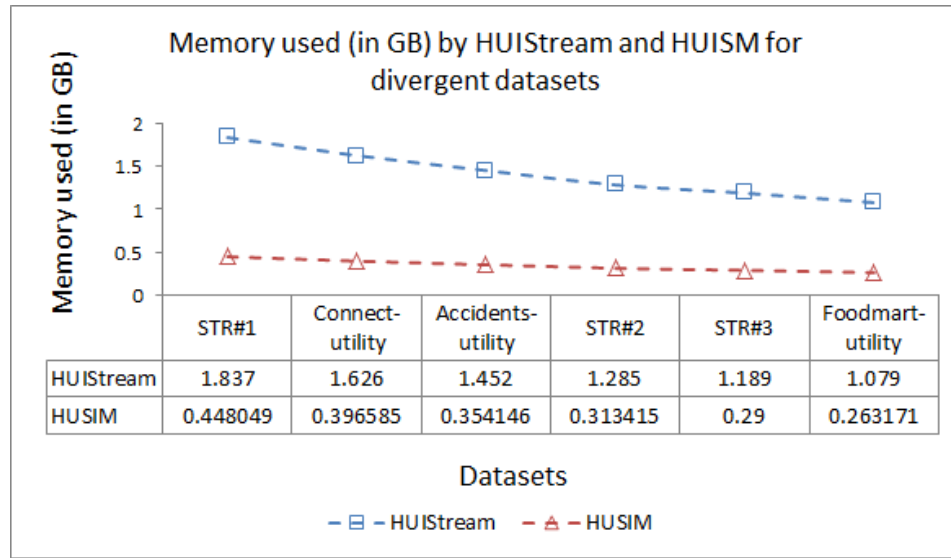


Figure 3: The use of search space observed for HUIStream, and HUSIM.

The pattern detection accuracy is also being estimated by performing cross-validation with the utility itemsets discovered by the annotation process carried by human efforts. The statistics used to estimate the accuracy is as follows.

$$\frac{|n|}{\forall} \left\{ acc[i] \leftarrow \frac{|prd[i] \cap ann[i]|}{|prd[i] \cup ann[i]|} \right\} \quad // \text{ the accuracy } acc[i] \text{ of predicted utility itemsets of size } i \text{ shall be the ratio of itemsets those exists in both predicted } prd[i] \text{ and annotated } ann[i] \text{ itemsets of size } i \text{ against total number of unique utility itemsets exists in both predicted } prd[i] \text{ and annotated } ann[i] \text{ utility itemsets of size } i.$$

The overall accuracy acc of the high utility itemset detection shall be estimated as follows

$$\langle acc \rangle = n^{-1} * \left[\sum_{i=1}^n acc[i] \right] \quad // \text{ the average of the prediction accuracy of diversified utility itemsets of size 1 to } n$$

$$\delta = n^{-1} * \left[\sum_{i=1}^n \left\{ \sqrt{(\langle acc \rangle - acc[i])^2} \right\} \right] \quad // \text{ the deviation of the accuracy}$$

$$acc = \langle acc \rangle \pm \delta \quad // \text{ the high utility itemset detection accuracy}$$

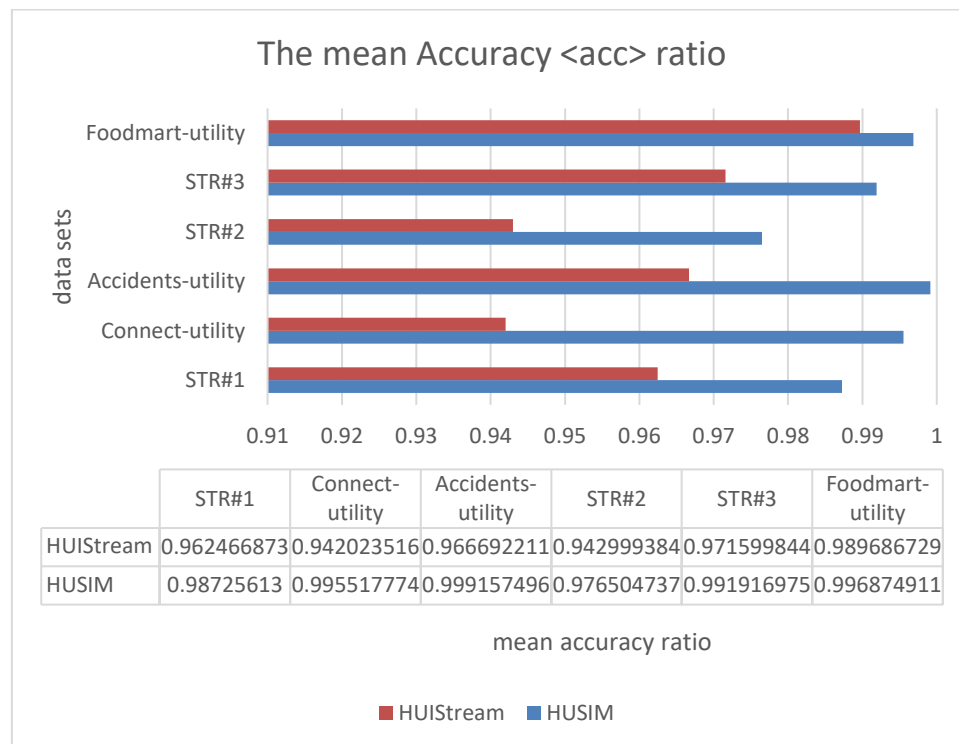


Figure 4(a): Mean accuracy $\langle acc \rangle$ of the high utility itemsets of the different sizes.

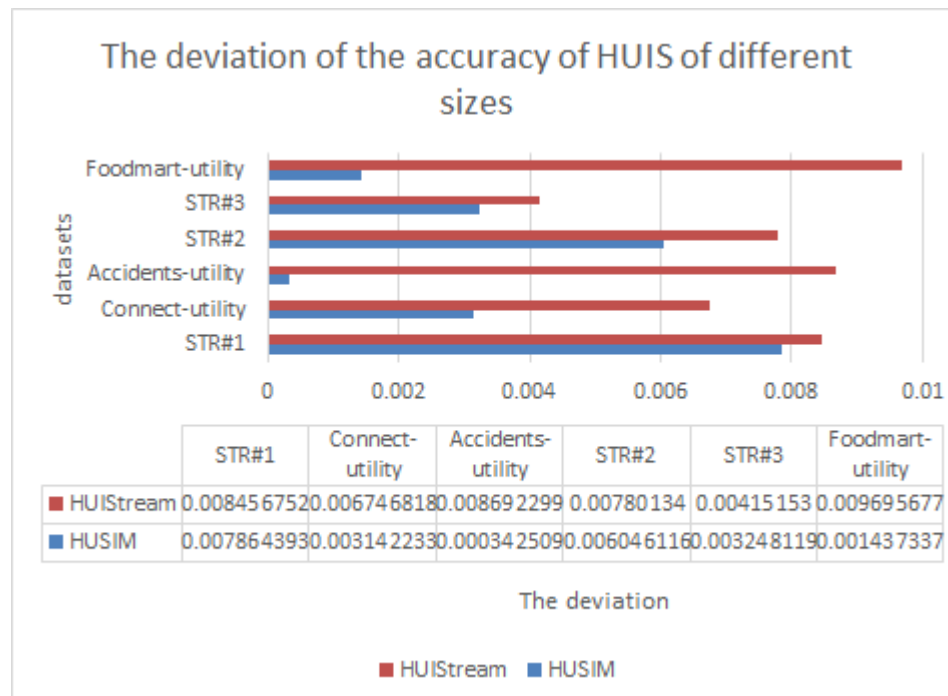


Figure 4(b): The root mean square deviation (δ) of the accuracy ratios of different high utility itemset sizes.

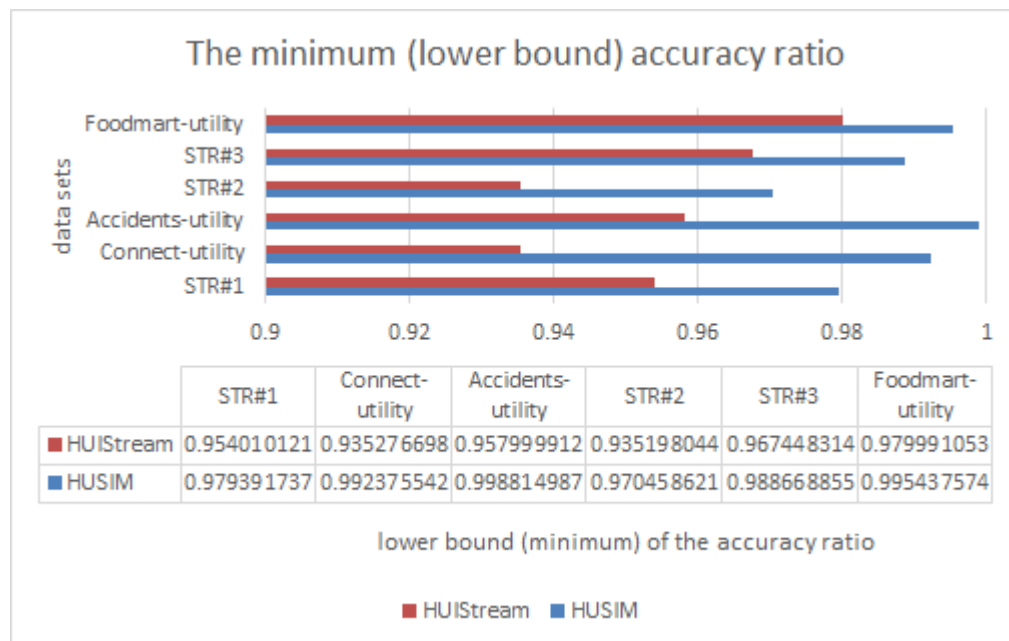


Figure 4(c): The lower bound of the accuracy ratios ($\langle acc \rangle - \delta$) of the different sizes of high utility itemsets.

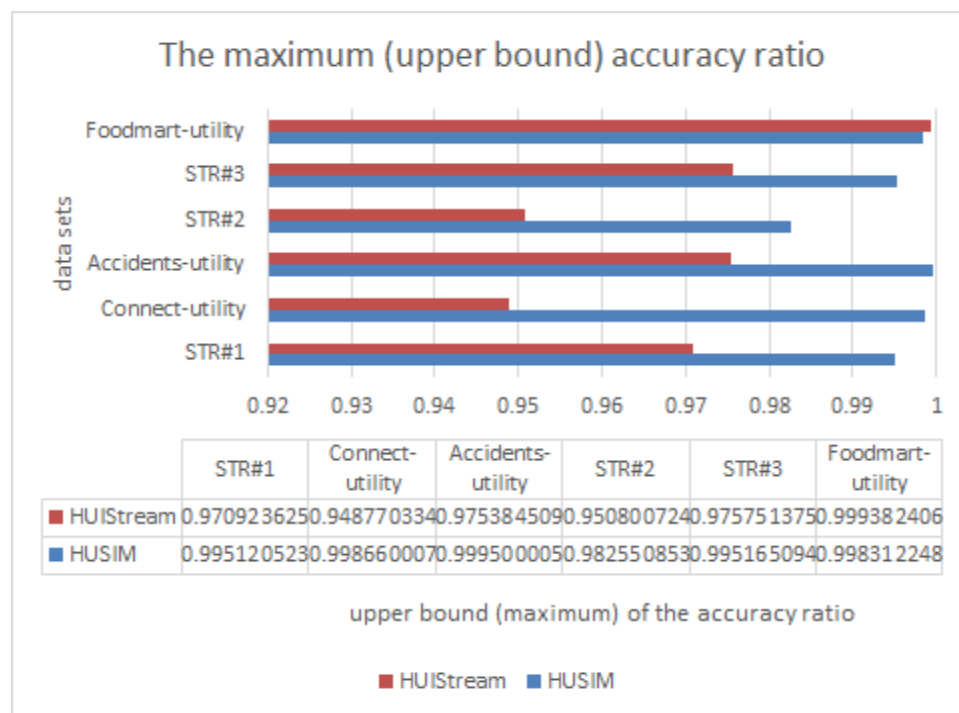


Figure 4(d): The upper bound of the accuracy ratios ($\langle acc \rangle + \delta$) of the different sizes of high utility itemsets.

Figure 4: The accuracy statistics of high utility itemsets discovered by HUSIM and HUIStream.

The accuracy statistics (mean, deviation, lower bound, and upper bound) have been portrayed as figures 4(a), 4(b), 4(c), and 4(d) in respective order. These statistics clearly evincing that the proposed method HUSIM is outperforming the contemporary model HUIStream with the maximum ratio of lower-bound, upper-bound, and mean accuracy ratios, which stated in the following notations.

$$acc_{HUSIM}^{\min} > acc_{HUIStream}^{\min}, acc_{HUSIM}^{\min} > acc_{HUIStream}^{\min}, \text{ and } \langle acc \rangle_{HUSIM} > \langle acc \rangle_{HUIStream}$$

$$acc_{HUSIM}^{\delta} < acc_{HUIStream}^{\delta}$$

In addition, the root-mean-square deviation of the accuracy ratios of the HUISM is evinced as minimum that scaled against the contemporary model HUIStream.

6. CONCLUSION

This paper aimed to devise a fast algorithm to derive high utility itemsets based on tabu search. The proposed tabu search-based fast and high utility itemsets (HUSIM) mining algorithm uses tabu lists for items, item sets, and transactions. This approach evinced that the candidates generated are much less than (almost 50%) compared to other state-of-the-art models, HUIStream. The search space in use for the proposed model is much less, which is more than 20% of the search space used by the HUIStream. The computational complexity of the HUSIM has been observed as linear from the experimental study, whereas the state-of-the-art model evinced the computational complexity is a compliment. This proposal motivates us for future work to extend this strategy for transaction datasets and data streams under multiple utility objectives.

References

- [1] A Lee, Gangin, Unil Yun, and Heungmo Ryang. "Mining weighted erasable patterns by using underestimated constraint-based pruning technique." *Journal of Intelligent & Fuzzy Systems* 28.3 (2015): 1145-1157.
- [2] Ryang, Heungmo, and Unil Yun. "Top-k high utility pattern mining with effective threshold raising strategies." *Knowledge-Based Systems* 76 (2015): 109-126.
- [3] Yun, Unil, Gangin Lee, and Chul-Hong Kim. "The smallest valid extension-based efficient, rare graph pattern mining, considering length-decreasing support constraints and symmetry characteristics of graphs." *Symmetry* 8.5 (2016): 32.
- [4] Zhang, Jingsong, Yinglin Wang, and Dingyu Yang. "CCSpan: Mining closed contiguous sequential patterns." *Knowledge-Based Systems* 89 (2015): 1-13.
- [5] Lee, Gangin, et al. "Approximate maximal frequent pattern mining with weight conditions and error tolerance." *International Journal of Pattern Recognition and Artificial Intelligence* 30.06 (2016): 1650012.
- [6] Mai, Thang, Bay Vo, and Loan TT Nguyen. "A lattice-based approach for mining high utility association rules." *Information Sciences* 399 (2017): 81-97.
- [7] Yun, Unil, Gangin Lee, and Kyung-Min Lee. "Efficient representative pattern mining based on weight and maximality conditions." *Expert Systems* 33.5 (2016): 439-462.
- [8] Duong, Quang-Huy, et al. "An efficient algorithm for mining the top-k high utility itemsets, using novel threshold raising and pruning strategies." *Knowledge-Based Systems* 104 (2016): 106-122.
- [9] Lin, Jerry Chun-Wei, et al. "EHAUPM: Efficient high average-utility pattern mining with tighter upper bounds." *IEEE Access* 5 (2017): 12927-12940.
- [10] Ryang, Heungmo, and Unil Yun. "Indexed list-based high utility pattern mining with utility upper-bound reduction and pattern combination techniques." *Knowledge and Information Systems* 51.2 (2017): 627-659.

- [11] Tseng, Vincent S., et al. "Efficient algorithms for mining top-k high utility itemsets." *IEEE Transactions on Knowledge and data engineering* 28.1 (2015): 54-67.
- [12] Kim, Donggyu, and Unil Yun. "Efficient mining of high utility pattern with considering of rarity and length." *Applied Intelligence* 45.1 (2016): 152-173.
- [13] Yun, Unil, et al. "Mining recent high average utility patterns based on sliding window from stream data." *Journal of Intelligent & Fuzzy Systems* 30.6 (2016): 3605-3617.
- [14] Liu, Ying, Wei-keng Liao, and Alok Choudhary. "A fast-high utility itemsets mining algorithm." *Proceedings of the 1st international workshop on Utility-based data mining. ACM*, 2005, pp. 90–99.
- [15] Agrawal, Rakesh, and Ramakrishnan Srikant. "Fast algorithms for mining association rules." *Proc. 20th int. conf. very large data bases, VLDB. Vol. 1215*. 1994.
- [16] Park, Jong Soo, Ming-Syan Chen, and Philip S. Yu. An effective hash-based algorithm for mining association rules. Vol. 24. No. 2. *ACM*, 1995, 175–186.
- [17] Yen, Show-Jane, Chiu-Kuang Wang, and Liang-Yuh Ouyang. "A search space reduced algorithm for mining frequent patterns." *Journal of information science and engineering* 28.1 (2012): 177-191.
- [18] Yen, Show-Jane, et al. "A fast algorithm for mining frequent closed itemsets over stream sliding window." *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, 996–1002.
- [19] Ahmed, Chowdhury Farhan, et al. "An efficient candidate pruning technique for high utility pattern mining." *Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg*, pp.749–756 (2009).
- [20] Tseng, Vincent S., et al. "Efficient algorithms for mining high utility itemsets from transactional databases." *IEEE transactions on knowledge and data engineering* 25.8 (2012): 1772-1786.
- [21] Zihayat, Morteza, and Aijun An. "Mining top-k high utility patterns over data streams." *Information Sciences* 285 (2014): 138-161.
- [22] Li, Hua-Fu, et al. "Fast and memory efficient mining of high utility itemsets in data streams." *2008 eighth IEEE international conference on data mining. IEEE*, pp.881–886 (2008).
- [23] Tseng, Vincent S., Chun-Jung Chu, and Tyne Liang. "Efficient mining of temporal high utility itemsets from data streams." *Second International Workshop on Utility-Based Data Mining. Vol. 18. Pp.18–27* (2006).
- [24] Lin, C. W., Lan, G. C., Hong, T. P.: Mining high utility itemsets for transaction deletion in a dynamic database. *Intelligent Data Analysis* 19(1), 43–55 (2015).
- [25] Ahmed, Chowdhury Farhan, et al. "Efficient tree structures for high utility pattern mining in incremental databases." *IEEE Transactions on Knowledge and Data Engineering* 21.12 (2009): 1708-1721.
- [26] Han, Jiawei, et al. "Mining frequent patterns without candidate generation: A frequent-pattern tree approach." *Data mining and knowledge discovery* 8.1 (2004): 53-87.
- [27] Ryang, Heungmo, and Unil Yun. "High utility pattern mining over data streams with sliding window technique." *Expert Systems with Applications* 57 (2016): 214-231.
- [28] Yun, Unil, and Heungmo Ryang. "Incremental high utility pattern mining with static and dynamic databases." *Applied Intelligence* 42.2 (2015): 323-352.
- [29] Tseng, Vincent S., et al. "UP-Growth: an efficient algorithm for high utility itemset mining." *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM*, pp.253–262 (2010).
- [30] Dawar, Siddharth, Veronica Sharma, and Vikram Goyal. "Mining top-k high-utility itemsets from a data stream under sliding window model." *Applied Intelligence* 47, no. 4 (2017): 1240-1255.

- [31] Choi, Hyeok-Jun, and Cheong Hee Park. "Emerging topic detection in twitter stream based on high utility pattern mining." *Expert systems with applications* 115 (2019): 27-36.
- [32] Demir, Seyfullah, Oznur Alkan, Firat Cekinel, and Pinar Karagoz. "Extracting Potentially High Profit Product Feature Groups by Using High Utility Pattern Mining and Aspect Based Sentiment Analysis." In *High-Utility Pattern Mining*, pp. 233-260. Springer, Cham, 2019.
- [33] Yen, Show-Jane, and Yue-Shi Lee. "An Efficient Approach for Mining High Utility Itemsets Over Data Streams." *Data Science and Big Data: An Environment of Computational Intelligence*. Springer, Cham, 2017. 141-159.
- [34] <http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php>.
- [35] 32. <https://github.com/zakimjz/IBMGenerator>.