Computation Of High Utility Item sets By Using Range Of Utility Technique

S.J. Vivekanandan[1],S.P. Ammu[2], R. Sripriyadharshini[3], T.R. Preetha[4]

[1]*Assistant Professor, Dhanalakshmi College of Engineering, Chennai.*

[2,3,4]*UG Scholars, Dhanalakshmi College of Engineering, Chennai.*

[1]*vivekanandan.sj@dce.edu.in*

[2]*ammusp.cse2017@dce.edu.in* [3]*sripriyadharshinir.cse2017@dce.edu.in* [4]*preethatr.cse2017@dce.edu.in*

Abstract:*Association rule mining is one of the major fact finding domains in data mining. There are many algorithms to estimate association rule. These algorithms are working with support value and confidence value, which are not sufficient for real time applications. High utility itemset is a new booming field in data mining, which is very useful in real time applications. It gives the significant or more valuable items to the user. In this paper, we are computing, high utility itemsets by using multi-utility range algorithm. This algorithm is very much useful to target the profit and also it gives a chance to modify the range of utility so that we can get the more valuable items.*

Keywords:**Association rule mining, Support and Confidence, High Utility Itemsets, Range of utility**

1. Introduction

Nowadays, a vast of data is generating day by day. So that, an efficient strategy is needed to handle the data. One of the important strategy is data mining. Data mining is the task of finding new patterns from the collection of data. There are various techniques like text mining, classification, spatial mining, web mining, association rule mining, etc. Association rule mining is one of the important booming fields of data mining. There are many algorithms to estimate association rule. The famous algorithms are Apriori algorithm and Frequent Pattern tree algorithm. These algorithms are working with support value and confidence value, which are not sufficient for real time applications. In real time business scenario, they need items which are more profitable.

Association Rules Mining (ARM) is one of the most booming investigation fields among the researchers. The target of ARM is to identify the strong association rules or the relationship between the itemsets from the mass amount of data. It can be understood in a simple phrase, “what goes with what” and “the purchase of one product when we purchase another product”. Association rules are denoted as G→H, where G is antecedent and H is consequent. It means, if G occurs, then H also possible to occur. Association rule mining is mainly used in market basket analysis to identify the customer purchase habits.

Let I = {i1, i2, i3, …, in} be a collection of items and Tran_DB = {t1, t2, t3,…, tn} a collection of transactions where every transaction is also a set of items. Association rules can be measured by support and confidence. Support is the total count of an item that appears in the transaction. Support (i1) = Number of times i1 appears in the database divided by the total number of transactions.

Support (i1i2) = Number of times i1i2 appears together in the database divided by the total count of transactions. Confidence is the probability of A occurs when B also occurs, where A and B are two different itemsets. Confidence (i1→i2) = ratio of Support (i1i2) to the Support (i1). An item or itemsets are not less than minimum support value, then the itemsets are called frequent item or frequent itemsets. An item or itemsets are lesser than the minimum support value, then the item or itemsets are called infrequent item or infrequent itemsets Association mining is a process of finding strong association rules [15] [16]. It can be completed in two steps. The step 1 is Frequent Itemsets Generation, here we compute all itemsets that are greater than or equal to support value. The second step is Association Rule Generation Based on step 1, pull out all the rules that are not lesser than confidence value. Those rules are called strong association rules.

In general, Association rule mining algorithms can be categories into two i.e. with candidate generation approach and without candidate generation approach. Apriori algorithm is the famous algorithm for the first category and frequent pattern tree algorithm is the fasmous algorithm for the second category. Apriori algorithm [15] is a famous algorithm to find the association rules. It works based on apriori property. Property 1, if an itemset is an infrequent itemset, then all its supersets also infrequent itemset. Property 2, if an itemset is a frequent itemset, then all its subsets also frequent itemset. The transactional database can be Boolean transaction i.e. it contains only 0 or 1 entries. 0 represent an item or itemset has not purchased. 1 represent an item or itemset has purchased. These algorithms are the base for many research work in this applications.

2. Related Work

Apriori-TID algorithm [15] was proposed to compute the frequently occurrence itemsets. Instead of scanning the database for candidate itemset support, it scans the database Tk which is smaller than the original database. So that it is better than original algorithm but it can be efficient only when the database is small. DHP algorithm [7][14] is another method to improve the original apriori algorithm. It is effective in trimming the database by discarding itemset from the transactions that do not need to be scanned. This algorithm uses a Hash table and bit vector to decrease the creation of candidate itemsets in the beginning pass. It uses the hash table in the next pass to minimize the count of candidate itemsets. But this algorithm also works well only if the size of the database is small; if the database is large, it takes a large amount of memory for hash table and the time consumption is very high. Dynamic Itemset Counting (DIC) [7] is an approach to enhance the apriori algorithm. This algorithm divides the database into multiple partitions. It scans the first partition for 1-frequent itemset and combines with the next partition until it completes entire partitions. It works well when the database is in same format. If not in the same format, it will not work well. Association rule mining applied in direct marketing application[13][16] to yield profits in their business based on customer purchase history. The association rule used to make a decision to predict the behavior of a customer group. HDO algorithm [12] is used to find association rule in the high-dimensional data. This algorithm uses a new method to remove candidate itemsets with rare itemsets, candidate itemsets can be pruned validly with the rare itemsets with lower dimension. A novel improved algorithm [11] was used to increase the mining efficiency of apriori algorithm. In this algorithm, they introduced some concepts called interest items and frequency threshold which are used to decrease the database searching time. Also dynamic mining used to enhance the efficiency of the algorithm. Another improved algorithm [9] was used to find association rules depends on utility weighted score (UW – score) which are extracted from weightage constraint (W-gain) and utility gain (U-gain). In this algorithm, association rules are computed depends on frequency as well as significant of the itemsets. Another important approach to prune mined association rules were discussed in this novel algorithm [10]. In this algorithm, the post processing method is used to prune association. It uses ontology, taxonomy and rule selection schema. It uses matching operator (M) and user-constraint template (UC) to select the most interesting rule. Another enhanced apriori algorithm for association rules [8] was used to increase the efficiency of the original apriori algorithm. It reduces the number of times the database has been scanned.

In the original apriori algorithm, they have made a slight variation in the logic that gives more efficiency for this improved algorithm. A matrix based apriori algorithm [3] was used to enhance the performance of the algorithm. In this method, transactional Boolean matrix was used to find the different candidate itemset generation. Initially, it uses the original apriori algorithm to find 1-frequent itemset and then it uses a Boolean matrix to find 2-itemsets, 3-itemsets with the transactional weight and Boolean matrix. So it is highly efficient than the original approach. An enhanced apriori algorithm based on time series [6] was used to increase the efficiency of the algorithm. In this method, Boolean matrix as well as different new concepts like sequence association rule, frequent item sequence generation, etc. Those concepts were used to find the time series association rule. Another enhanced apriori algorithm based on support weight matrix [5][3] was used to find association rules. It uses 0-1 transaction matrix and it gives association rules as well as significance to the user. In [17], most of the improvement and research work of the apriori algorithm were accumulated. In [18], most of the improvement and research work of utility mining were accumulated. And also applications of apriori in medical transactions, i.e., efficiently handling the repeated transaction in [4]. In business scenario, they need frequency value items and also need utility value items to have a consistent growth. So, in [1] a study of utility-frequent itemsets which covers most of the UF algorithms and its usage. Like high utility itemset mining (HUIM), a new framework created for low utility itemset mining(LUIM), this mechanism would boost the profit in a collection of low utility itemsets[2]. A traditional utility mining algorithm is discussed in [19], it gives the HUI but it is not handles the repeated transactions efficiently.

3. Materials and Methods

In this section, we have given the basic concept of high utility mining and also our proposed methodology of computing high utility itemsets with range of utility algorithm. Utility mining is a investigating domain in data mining. It is mainly focused in market basket analysis to retrieve a high profit from the different combination of itemsets. Utility is the interestingness of user preferences like profit, cost, etc. Utility mining is the enhanced version of association rule mining.

We discuss the basic terminologies with the help of the Table I and Table II. In Table I, it gives unit profit for each item, so it will be helpful to find the total cost of a transaction and also to find the total cost of an item in the database. In Table II, it has ten transactions; every transaction has unique TID and quantity of items purchased.

Table1. Profit Table

Item	Profit
A	5
B	100
C	40

Table2.Sample Transaction Database

Transaction TID	Quantity of items purchased in transactions		
	A	B	C
T101	2	0	1
T102	4	0	2
T103	4	1	0
T104	0	1	1
T105	5	1	2
T106	10	1	5
T107	4	0	2
T108	1	0	0
T109	3	0	0
T110	5	0	0

Support denotes the total count of an item or itemset present in the transaction. In Table II, Support (A) = 9/10, Support (B) = 4/10, Support (C) = 6/10, Support (AC) = 5/10 and Support (BC) = 3/10. An item or itemsets are not lesser than minimum support value, then the itemsets are called frequent itemsets. An item or itemsets are not greater than the minimum support value, then the itemsets are called infrequent itemsets or rare itemsets. Internal utility refers the quantity of each item in the transaction. It is used to find the utility of an item as well as the utility of a transaction. In Table II, T101 has (A, 2) and (C, 1) are the internal utilities of the itemsets. Similarly, T106 (A, 10), (B, 1) and (C, 5) is the internal utilities of the itemsets. External utility is the profit of each item available in the transaction. In Table I, (A, 5), (B, 100) and (C, 40) denotes its profit associate with it. The Utility of an item or itemset is the multiplying its internal utility and external utility. It can be represented as U = IU * EU. For example, U (A, T101) = 2 * 5 = 10, U (B, T104) = 1*100 = 100, U(C, T106) = 5 * 40 = 200, U (AC, T107) = 4 * 5 + 2 * 40 = 100, U (ABC, T106) = 10 * 5+1 * 100+5 * 40 = 350.

The Utility of an item or itemset for the database is the product of the quantity of the item or itemsets purchased in the database and its corresponding external utility (profit). For example, UD (A) = 38 * 5 = 190, UD (B) = 4 * 100 = 400 and UD (C) = 13 * 40 = 520. Transaction utility is computed by multiplying each item's internal utility with its external utility. For example, TU (T101) = 2 * 5 + 0 * 100 + 1 * 40 = 50, TU (T104) = 0 * 5 + 1 * 100 + 1 * 40 = 140, TU (T105) = 5 * 5 + 1 * 100 + 2 * 40 = 205, TU (T106) = 10 * 5 + 1 * 100 + 5 * 40 = 350. Transaction-weighted utility of an itemset Y is defined as the sum of the transaction utilities (TU) of all the transactions that contains the itemset Y. For example, TWU (B) = TU (T103) + TU (T104) + TU (T105) + TU (T106) = 120 + 140 + 205 + 350 = 815. TWU (A) = TU (101) + TU (102) + TU (103) + TU (105) + TU (106) + TU (107) + TU (108) + TU (109) + TU (110) = 50 + 100 + 120 + 205 + 350 + 100 + 5 + 15 + 25 = 970. TWU (C) = TU (101) + TU (102) + TU (104) + TU (105) + TU (106) + TU (107) = 50 + 100 + 140 + 205 + 350 + 100 = 945. The Utility of an itemset is not less than minimum utility threshold, then the itemset iscalled high utility itemset. An itemset Y is called a high transaction weighted utility itemset, then its transaction-weighted utility (TWU) must be greater than the minimum threshold utility (min_util). For any itemset Y, if Y is not a HTWUI, then any superset of Y is a low utility itemset. For example TWU (A) = 970, TWU (B) = 815 and TWU (C) = 945 if min_util is 900, then item B is not HTWUI. Then U (AB) = 120 + 125 + 150 = 395 i.e. low utility itemset. Range of utility is the technique to give two different utilities High Range (HR) utility value and Low Range (LR) utility value, which are useful to define a target profit.

4. Proposed Methodology

In this section, we have proposed a new algorithm called multi-utility range algorithm. We are computing, high utility itemsets by using multi-utility range algorithm. Thisalgorithm is very much useful to target the profit and also it gives a chance to modify the range of utility so that we can get the more valuable items.

4.1. Multi-Utility Range Algorithm

```
Multi-Utility Range algorithm – Compute the High utility itemsets by using HR and LR
Input: Transaction Database, Profit table, HR util, LR utilR
Output:   High utility Itemset
Steps:
1. Initialize an empty set High Utility Itemset
2. Scan the transaction database and take all available items and compute its utility value in thedatabase.
3. If(repeated transaction) then neglect it
4.Otherwise, generate a combination of available items in that particular transaction and find itsutility.
```

```
5. Select each item from combination
        If(HR>=Utility>=LR) and (Not in HUI) then add items to HUI
6. Repeat steps 3, 4 and 5 until there is no more transaction.
7. Display the High Utility Itemset
8. Stop
```

We discuss our proposed algorithm. We take the transaction table and profit table for our example.

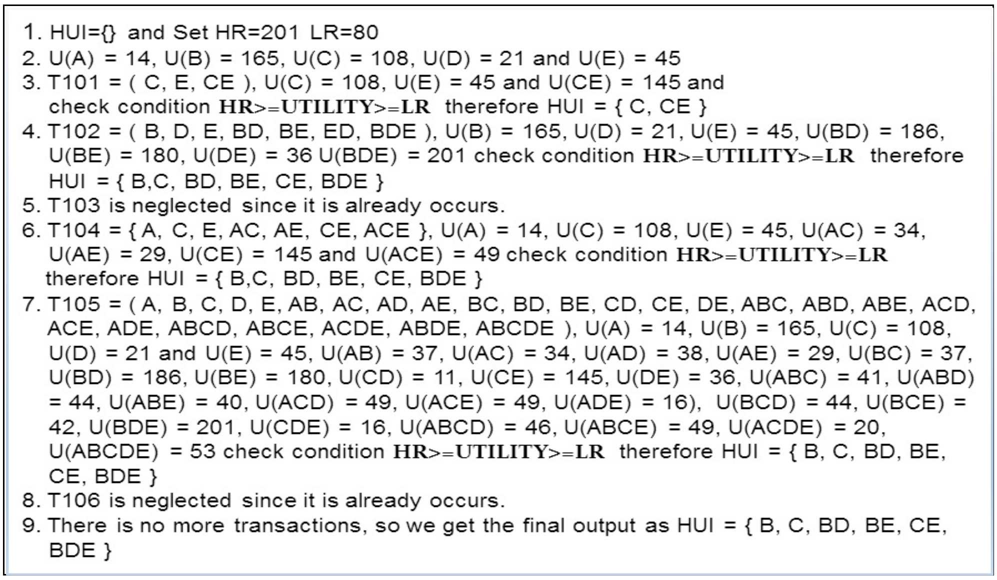
Table 3.Transaction Table

TRANSACTION ID	A	B	C	D	E
T101	0	0	18	0	1
T102	0	6	0	1	1
T103	0	6	0	1	1
T104	5	0	4	0	2
T105	2	3	1	1	1
T106	0	0	4	0	3

Table 4. Profit Table

A	B	C	D	E
2	11	4	7	5

We assume LR = 80 and HR = 201. In step 1, HUI set is initialized to zero i.e., HUI = { } empty set. In step 2, it checks the available items and compute its utility value. In step 3, it checks the transaction duplications. If same transaction occurs, it ignores that transaction. In step 4, if the transaction is new, then it takes the item which are present in the transaction and it generates combination of all present items and compute the utility value for all combinations. In step 5, select each item from combination and check its utility value is in the range between HR util and LR util. If it is in the range and also it is not present in the HUI set, then add that item to HUI set. Repeat the steps 3, 4 and 5 until there is no more transaction in the database. In step 7, we get the final HUI as an output of this algorithm. The example is illustrated in the below Figure 1.



1. HUI={} and Set HR=201 LR=80
2. U(A) = 14, U(B) = 165, U(C) = 108, U(D) = 21 and U(E) = 45
3. T101 = (C, E, CE), U(C) = 108, U(E) = 45 and U(CE) = 145 and check condition **HR>=UTILITY>=LR** therefore HUI = { C, CE }
4. T102 = (B, D, E, BD, BE, ED, BDE), U(B) = 165, U(D) = 21, U(E) = 45, U(BD) = 186, U(BE) = 180, U(DE) = 36 U(BDE) = 201 check condition **HR>=UTILITY>=LR** therefore HUI = { B,C, BD, BE, CE, BDE }
5. T103 is neglected since it is already occurs.
6. T104 = { A, C, E, AC, AE, CE, ACE }, U(A) = 14, U(C) = 108, U(E) = 45, U(AC) = 34, U(AE) = 29, U(CE) = 145 and U(ACE) = 49 check condition **HR>=UTILITY>=LR** therefore HUI = { B,C, BD, BE, CE, BDE }
7. T105 = (A, B, C, D, E, AB, AC, AD, AE, BC, BD, BE, CD, CE, DE, ABC, ABD, ABE, ACD, ACE, ADE, ABCD, ABCE, ACDE, ABDE, ABCDE), U(A) = 14, U(B) = 165, U(C) = 108, U(D) = 21 and U(E) = 45, U(AB) = 37, U(AC) = 34, U(AD) = 38, U(AE) = 29, U(BC) = 37, U(BD) = 186, U(BE) = 180, U(CD) = 11, U(CE) = 145, U(DE) = 36, U(ABC) = 41, U(ABD) = 44, U(ABE) = 40, U(ACD) = 49, U(ACE) = 49, U(ADE) = 16), U(BCD) = 44, U(BCE) = 42, U(BDE) = 201, U(CDE) = 16, U(ABCD) = 46, U(ABCE) = 49, U(ACDE) = 20, U(ABCDE) = 53 check condition **HR>=UTILITY>=LR** therefore HUI = { B, C, BD, BE, CE, BDE }
8. T106 is neglected since it is already occurs.
9. There is no more transactions, so we get the final output as HUI = { B, C, BD, BE, CE, BDE }

Figure1. Example of MUR algorithm

5. Results andDiscussion

We implement our algorithm using python with a different set of transaction size. Our MUR works well than the apriori algorithm. Apriori algorithm is focusing only on frequent itemsets but our algorithm is focusing High Utility Itemsets which are very useful to the user. If we take the same example for our comparison, let's take 40% of support value, then apriori algorithm gives frequent itemset as { B, C, D, E, BD, BE, CE, DE, BDE }. If we compare these frequent items with utility value, all the items are frequent but not high utility itemsets. If we compare our algorithm with a traditional utility mining algorithm, traditional utility mining algorithm will not handle the repeated transactions and it takes more computation time and also more memory. But our proposed approach performs well to handle the repeated transactions.

We compare our proposed algorithm with apriori algorithm, the count of occurring often itemsets generated from apriori is more but its utility value is less. In the small number transaction database, frequent and utility may look equal. But when we take different size of transactions, our proposed algorithm will give more utility items even that items are not a frequent. The below chart shows that, our algorithm performs well than the apriori algorithm.

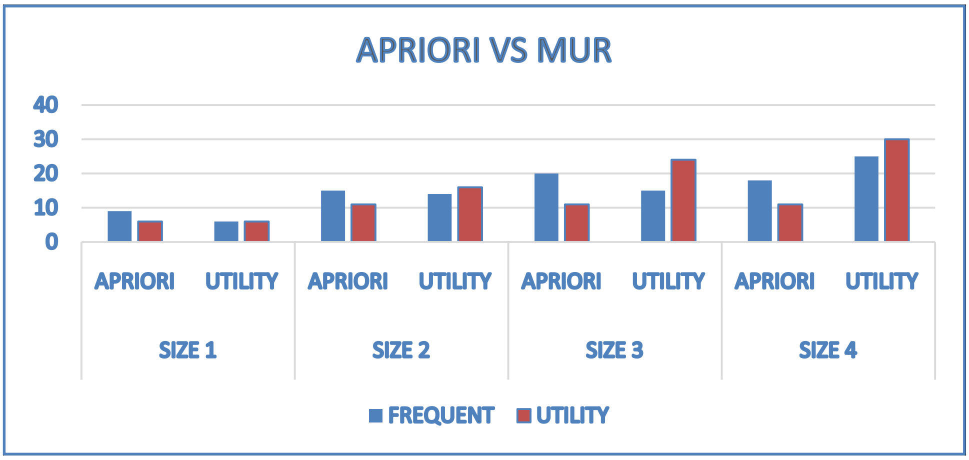


Figure 2.Apriori Vs MUR

We compared our proposed algorithm with traditional utility mining algorithm, the traditional utility mining algorithm is not handling the repeated transactions. But our proposed algorithm is handling the repeated transactions effectively. So the combination size, execution time and memory is getting reduced. The below figure 3 shows that our proposed approach is perform well than the previous utility approach.

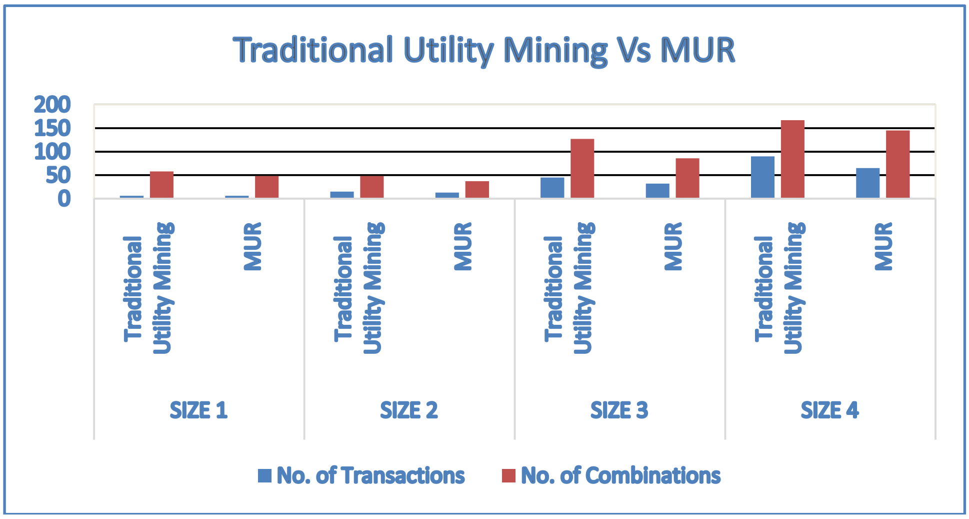


Figure 3. Traditional Utility Vs MUR

6. Conclusion

In this work, we proposed a new way to compute high utility itemsets by using multi utility range algorithm. Experiments results shows that our approach is performs well than the apriori algorithm in terms of utility value. Also, it proves that our algorithm handles the repeated transactions efficiently than the traditional utility algorithm. So we conclude that our approach will be useful for the business community to improve their profit and also it gives a chance to them to modify their targets depends on their daily sales. In the future, we are planning to focus to merge our approach with low utility mining scenario.

REFERENCES

[1] *S J Vivekanandan, G Gunasekaran. A Study on Utility-Frequent Itemset Mining. Journal of Seybold Report. Volume 15 Issue 9 2020; 3573-3581.*

[2] *N. Alhusaini, S. Karmoshi, A. Hawbani, L. Jing, A. Alhusaini and Y. Al-sharabi, "LUIM: New Low-Utility Itemset Mining Framework," in IEEE Access, vol. 7, pp. 100535-100551, 2019, doi: 10.1109/ACCESS.2019.2929082.*

[3] *Q. Yang, Q. Fu, C. Wang and J. Yang, "A Matrix-Based Apriori Algorithm Improvement," 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, China, 2018, pp. 824-828, doi: 10.1109/DSC.2018.00132.*

[4] *S J Vivekanandan, G Gunasekaran. An Improvisation Apriori Algorithm Applied in Medical Transcation. Journal of Green Engineering. Volume 10 Issue 10 October 2020; 8574-8586.*

[5] *Sun, Ln. An improved apriori algorithm based on support weight matrix for data mining in transaction database. J Ambient Intell Human Comput 11, 495–501 (2020). https://doi.org/10.1007/s12652-019-01222-4*

[6] *Wang, C., Zheng, X. Application of improved time series Apriori algorithm by frequent itemsets in association rule data mining based on temporal constraint. Evol. Intel. 13, 39–49 (2020). https://doi.org/10.1007/s12065-019-00234-5*

[7] *G.K.Gupta, "Text Book: Introduction to Data Mining with Case Studies" 3rd edition, pp. 91-151 PHI Learning Pvt. Ltd. 2019*

[8] *Mohammed Al-Maolegi, Bassam Arkok, "An Improved Apriori Algorithm for Association Rules", IJNLC vol. 3, No.1, February 2014.*

[9] *P. S. Sandhu, D. S. Dhaliwal, S. N. Panda and A. Bisht, "An Improvement in Apriori Algorithm Using Profit and Quantity," 2010 Second International Conference on Computer and Network Technology, Bangkok, Thailand, 2010, pp. 3-7, doi: 10.1109/ICCNT.2010.46.*

[10] *D.Narmada, G.Naveen, S.Geetha, " An efficient approach to prune Mined association rules in large databases", IJCSI, Vol.8, Issues 1, jan 2011.*

[11] *L. Wu, K. Gong, Y. He, X. Ge and J. Cui, "A Study of Improving Apriori Algorithm," 2010 2nd International Workshop on Intelligent Systems and Applications, Wuhan, China, 2010, pp. 1-4, doi: 10.1109/IWISA.2010.5473450.*

[12] *L. Ji, B. Zhang and J. Li, "A New Improvement on Apriori Algorithm," 2006 International Conference on Computational Intelligence and Security, Guangzhou, China, 2006, pp. 840-844, doi: 10.1109/ICCIAS.2006.294255.*

[13] *Wang K, Zhou Q, Yeung, "Mining Customer value: From Association Rules to Direct marketing", Data Mining and Knowledge Discovery, Vol. 11, pp. 57-79 2005.*

[14] *J.-S. Park, M.-S Chen, and P.S. Yu, " An Effective Hash based algorithm for mining association rules", Proc. of ACM-SIGMOD, pp. 175-186, May 1995.*

[15] *Agrawal R, Srikant R, " Fast algorithms for mining association rules", Proceedings of 20th International Conf. on Very large Databases, Santiago, Chile, pp 487-499, 1994*

[16] *Agrawal R, Imielinski T., Swami A, "Mining association rules between set of items in large databases", Proceedings of the ACM SIGMOD Intl. Conf. on Management of Data, Washington, D.C.. may 1993, pp 207-216.*

[17] *S J Vivekanandan, G Gunasekaran. A Survey on Association Rules Mining. Asian Resonance. VOL.- 8, ISSUE-1, pp. 1 – 4, January (Part-1) 2019.*

[18] *S J Vivekanandan, G Gunasekaran. A Survey on Utility Mining. Asian Resonance. VOL.- 8, ISSUE-1, pp. 5 – 9, January (Part-1) 2019.*

[19] *S. Shankar, N. Babu, T. Purusothaman and S. Jayanthi, "A Fast Algorithm for Mining High Utility Itemsets," 2009 IEEE International Advance Computing Conference, Patiala, India, 2009, pp. 1459-1464, doi: 10.1109/IADCC.2009.4809232.*