IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Privacy Preserving Techniques with Novel Tree Structure and Table for Utility Mining

G. Kiruthika [1], Dr. V. Umarani [2]

Research Scholar, Department of Computer Science, Sri Ramakrishna College of Arts and Science for Women, Coimbatore, India[1]

Associate Professor, Department of Computer Science, Sri Ramakrishna College of Arts and Science for Women, Coimbatore, India[2]

Abstract: Utility pattern mining is to identify the item sets with highest utilities, by considering profit, quantity, cost or other preferences. Mining high utility item set from a transaction database is finding item sets that have utility above a user-specified threshold. It extracts highly utilized items accurately, but it suffers due to the privacy breaches. Privacy preserving data mining preserves the sensitive data by modifying the structure of the data. A PPUM algorithm requires a significant amount of time to protect the privacy of data holders because they conduct database scanning operations excessively many times until all important information is hidden. The algorithm becomes inefficient in case of larger application. To overcome this drawback FPUTT(Fast Perturbation algorithm using Tree structure and Table) algorithm minimize the scanning to three times, two time for constructing tree and one time for updating On comparing FPUTT with PPUM in terms of time complexity, privacy preservation and the execution time , FPUTT is efficient than HUM-UT. The existing FPUTT algorithm uses perturbation technique for securing sensitive items which offers a low level security. The proposed work HUM-UT (High Utility itemsets Mining based on Utility Tree) algorithm handles the data streams. In order to enhance the security level, HUM-UT uses anonymization technique for protecting sensitive items The MWS (Maximal frequent pattern mining with weight conditions over data streams)Maximal frequent pattern mining rule and closed frequent pattern rule is applied, to extract maximal itemset and closed item set MMCAR(Minimal non redundant Multilevel and Cross-level Association Rules) from the given item sets. The implementation of the above techniques s increased security to item set and also extracts maximal and closed item sets from data stream. The experimental results show the proposed algorithm outperforms the existing in terms of execute time, memory of usage and number of candidates.

Keywords: Fast Perturbation, closed frequent pattern rule, High utility item set, Maximal frequent pattern mining rule.

I. INTRODUCTION

Utility pattern mining [1], one of the interesting pattern mining techniques, can analyze business relationships in market data including utility issues since such utility pattern mining approaches can deal with non-binary data such as quantities of products and consider relative importance of items such as their profits. However, these methods may also cause privacy concerns by violating the privacy of data holders because they can mine sensitive information of data holders. In business environments, privacy breaches (or risks) signify that specific information data holders want to hide is unexpectedly or illegally discovered by data analyzing methods.

An existing system [2] uses fast perturbation method based on a tree structure which done the database perturbation processes more effectively and quickly for preventing sensitive items from being exposed, Perturbation is the process depends on threshold values that decrease the values of the sensitive item set. Even if the values of the sensitive item set are modified, it can be easily identified. To prevent this anonymization technique is implemented in the proposed system.

The proposed system is to increase the privacy of the database for preventing sensitive items from being exposed by implementing the data anonymization algorithm Update-Anonymize-Reorder (UAR). The data stream is handled by implementing (HUM-UT) High Utility itemsets Mining based on UT-Tree.

Finally maximal and closed pattern mining are performed by implementing Maximal frequent pattern mining with Weight conditions over data Streams (MWS) and closed itemset lattice-based approach for mining Minimal non redundant Multilevel and Cross-level Association Rules(MMCAR).

II. BACKGROUND

Ali, N. A., & Arunkumar, M., [3] proposed Miner Algorithm for extracting high utility itemset by focusing on memory and time for high utility mining. A vertical data structure is utilized to store the elements with the utility values. A matrix representation is developed to determine the element co-occurrences and diminish the

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

joint operation for the patterns developed. The advantages of the system are inexpensive scenario and higher performance. The disadvantage of this system is Search space complexity.

Pyun, G., et al [4] proposed a Linear Prefix Tree (LP-tree) used to extract the high utility items without the utilization of pointers. Most of the frequent pattern mining methods utilized tree structures and nodes were connected through the pointers, these pointers occupy lot of memory and create security issues. Such issues were overcome by LP-tree. LP-tree is created in the form of array and reduced the usage of pointers between the nodes. Moreover, it used a small amount of information to mine frequent patterns in the dataset. The accuracy of frequent pattern mining is not reached at expected level.

Navaz, A. S. S., et al [5] proposed approaches to protect sensitive rules from disclosure based on data mining called inference rule hiding. In this approach, initially determine frequently occurring itemset in the dataset and retrieve all possible association rules from the transactions that support the itemset. Then hide the association rules by reducing the value of confidence or support of association rules. The advantages of this approach are high performance, greater security and high reliability. The downside of this approach is in few cases adversary can easily uncover the information.

Duraiswamy, K., et al [6] proposed Sensitive Rule Hiding algorithm which is used in output privacy to hide the sensitive rules that contains sensitive items in the dataset. This approach changed the data in the dataset to preserve sensitive items. By changing the data the sensitive items cannot be assumed through association rule mining. Due to this modification of data in a transaction the confidence level of association rule can be decreased. The benefits of this approach are reduced the attack significantly and provide high quality hiding solutions. The downside of this approach is undesirable side effects will degrade the efficiency of the scenario in few cases.

Li, Y., et al [7] proposed a solution for challenges in Multi-Level Trust Privacy Preserving Data Mining (MLT-PPDM). The solution concentrated on additive perturbation approach. In this approach random Gaussian noise is mixed with arbitrary distribution of original dataset. This solution lets a data owner to create unique perturbed copies of its data based on different trust levels. The advantages of this approach is suitable for high dimensional data, achieves the privacy goal efficiently and high trust level. The disadvantage this approach is takes long time for execution.

Tseng, V. S., et al [8] proposed utility pattern growth (UP-Growth) algorithm for long transaction or database contain high utility itemsets. This algorithm required only two scans of database and it have information about high utility itemset in a tree like data structure called UP-Tree. There are four strategies were developed to prune utility items that are involved in a search space and are impossible to be a high utility itemset. The advantages of this technique is provides importance high utility itemsets and highly scalable approaches are used. The drawback of this technique is mining performance is degraded consequently and it consumes larger memory space.

Tzvetkov, P., et al [9] proposed a mining task for mining top k frequent closed sequential patterns called travelling salesman. This algorithm utilized large number of constraints and characteristics and properties of top k frequent closed patterns to describe the top k frequent closed patterns by dynamic support raising and database pruning. The advantage of this approach is high memory efficiency and less computational complexity. The drawback of this approach is overhead.

Vidyabanu, R., & Nagaveni, N., [10] proposed an approach based on Principal Component Analysis for privacy preserving clustering. This approach is well suited for the database of horizontally partitioned or centralized datasets. This approach hides sensitive information in database with the help of self organizing map and principal component analysis technique by converting multi dimensional data into lower dimensions.

III.PROPOSED WORK

The proposed system implement anonymization for effective results. Anonymization is the process that stores the data in form range of values so that high privacy is achieved for the item set .The method of storing the data in the form group of values, so when trying to access a particular value provides result in the form group of values hence sensitive data are highly protected thereby ensuring privacy for the data. Data anonymization is a technique that will not take away the original field layout (position, size and data type) of the data being anonymized, so the data will still look realistic in test data environments. The Update-Anonymize-Reorder (UAR) algorithm aims at anonymizing a dataset so that it satisfies the specified privacy and utility constraints and incurs minimal information loss. To achieve this, UAR stores the privacy constraints that require protection in a priority queue. The process continues until all privacy constraints are satisfied. The implementation of data streaming deals with effective handling of addition of data items into the database HUM-UT (High Utility itemsets Mining based on UT-Tree), for mining high utility itemsets from data streams based on the sliding window filtering method. HUM-UT can find the high utility item sets without generating candidate item sets, and need only one scan of dataset.

MWS (Maximal frequent pattern mining with Weight conditions over data Streams) guarantees efficient mining performance in the data stream environment by scanning stream databases only once, and prevents overheads of pattern extractions with an abbreviated notation: a maximal frequent pattern form instead of the general one. Furthermore, MWS contributes to enhanced reliability of the mining results by applying weight conditions to each element of the data streams.

Closed item set lattice-based approach for mining. Minimal non redundant Multilevel and Cross-level

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Association Rules (MMCAR) algorithm is implemented for closed pattern mining. It is designed for mining cross-level closed itemsets by traversing the level-wise generated closed itemset lattice in a very efficient manner. The proposed system performs database anonymization processes for preventing sensitive information from being exposed. There are four phases are involved in the proposed system.

Phase 1: Collecting the sensitive items from the transactional data streams.
Phase 2: Applying the maximal property and closed property to find the maximal and closed itemset.
Phase 3: Protecting the sensitive items using anonymization technique.
Phase 4: Secured transactional data streams containing the sensitive items.

3.1 Overview of Proposed System

Fig 4.1 is the flow diagram explains the overall process of existing system. The process of input data, using anonymization techniques after tree construction, perturbation operation, table initialization the process completed final execution the sensitive item set are stored in perturbed database.

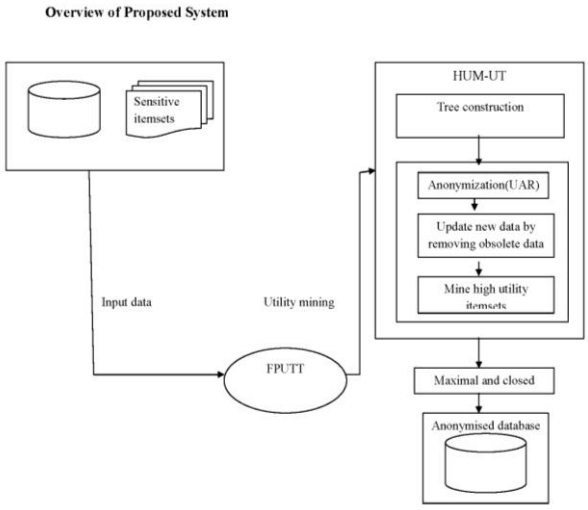

Fig 3.1 Flow diagram for proposed system

Table explains proposed system with examples

Table 3.1: An example database

TID	Transaction
T1	(C, 7), (D, 1), (E, 1)
T2	(A, 1), (C, 2), (E, 2)
T3	(B, 6), (C, 4), (D, 3), (E, 7)
T4	(B, 5), (C, 3), (D, 9)
T5	(A, 3), (C, 10), (D, 3)
T6	(C, 5), (E, 9)
T7	(A, 6), (C, 9), (D, 2), (E, 5)
T8	(A, 1), (B, 6), (C, 2), (D, 5), (E, 3)

Table 3.2: An example external utilities of items

TID	Transaction
A	5
B	3
C	1
D	6
E	2

Table 3.3: An example of database with sensitive items

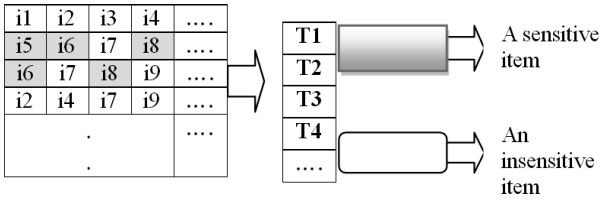

Table 3.4: Items with actual count

Items	Actual count
i1	4
i2	7
i3	3
i4	1
i5	9
i6	8
i7	7
i8	5
i9	2

Table 3.5: Perturbation operation

Items	Actual count
i1	4
i2	7
i3	3
i4	1
i5	6
i6	4
i7	7
i8	3
i9	8

Table 3.6: Anonymization operation

Items	Actual count
i1	1-4
i2	3-7
i3	1-3
i4	0-1
i5	4-6
i6	2-4
i7	4-7
i8	1-3
i9	5-7

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Table 3.7: Output table with high utility item set

Itemset	Utility	Item set	Utility
{ACD}	131	{BDE}	104
{ACDE}	104	{BCDE}	110
{AD}	110	{CD}	173
{BCD}	162	{CDE}	120
{BD}	153	{D}	138

3.2 Phase 1: Collecting the sensitive items by performing utility mining on the transactional data streams

The sensitive items are determined by means of utility mining. Mining High Utility itemsets from a transaction data stream is a method of finding itemsets that have utility above a user-specified threshold, this referred as Utility mining. Utility Mining is an extension of Frequent Itemset mining. HUM-UT (High Utility itemsets Mining based on UT-Tree), to find high utility itemsets from transactional data streams. The HUM-UT algorithm mines high utility itemsets from the UT-Tree without additional database scan.

Steps involved in HUM-UT (High Utility itemsets Mining based on UT-Tree),

1.Construct a UT-Tree

2. Remove obsolete data and update new data

3.mine high utility itemsets from the current window

3.2.1. Construct a UT-Tree

The construction of a UT-Tree is as follows:

1. Arrange the items of each transactional itemset in lexicographic order

2. Add the sorted itemset into a UT-Tree, and the utility values of all items of an itemset are stored as a list on its tail-node.

3.2.2. Remove obsolete data and update new data

The tail node table maintains all tail-nodes of each batch of the current window. When obsolete data are discarded, then can find all tail-nodes of the obsolete batch through searching the tail node table instead of traversing the whole tree, and set the corresponding utility information of these nodes as null. After removing the obsolete data, next step is to append a new batch of data into the tree.

3.2.3 Mine high utility item sets from the current window

Step 1: Calculate the minimum utility value

The total utility of current window is calculated using the global table, and then the minimum utility value for the current window is calculated as the total utility multiplied by the minimum utility threshold.

Step 2: Create a header table for the global tree

To mine high utility itemsets from the global tree, first construct a header table for the global tree by traversing the tree once. The header table keeps records of the items and their corresponding nodes on the tree. All items of the header table are sorted in lexicographical order.

Step 3: Add an attached list to each leaf node

Because the data on the global tree are used not only for the current window, but also for the subsequent windows, the utility information on the global tree cannot be modified when mining HUIs from the current window. Thus attach a list (called Utility cache) to each leaf node, to record the sum of each element of all other lists of utility information on this leaf node, and store this list in the last "{}" on the leaf node .

Step 4: Calculate two and utility value of each item in the header table.

For each item the utility value and the two values are estimated, process each item beginning from the last one of the header table. Utility values can be calculated from the corresponding nodes Utility cache.

Step 5: Create a prefix-tree and a header table for the base-item set

The actual utility of each item can be retrieved from the global tree. Build a prefix tree and a sub header table for a base-item set through scanning the corresponding branches on the global (or parent) tree.

Step 6: Process the prefix tree

Process the prefix tree and the sub header table recursively. When a branch on a prefix tree is retrieved to construct a new prefix tree, the utility of the currently process node is added to the utility value of the old base-itemsetto generate a new utility for the new base-itemseton the corresponding tail-node of the new prefix tree.

Pseudo Code for FP-UT Tree Algorithm:

Input: Original Database DB, Minimum Utility Threshold, A sensitive Item sets SI

Output: A perturbed Database PDB.

Algorithm:

Call Create FPUTT-Tree(Tree,SI,HT,IIT)
Call Create Perturb FPUTT-Tree
For each path T_i in DB do
P_i = Find a Path that has T_i in TiD in Tree
if($P_j \neq \emptyset$) then
PDB=PDB ∪ P_j and items with P_j.TiD in IIT
else PDB = PDB ∪ P_j

Pseudo Code for FP-UT Tree Algorithm:

Input: Original Database DB, Minimum Utility Threshold, A sensitive Item sets SI

Output: A perturbed Database PDB.

Algorithm:

Call Create FPUTT-Tree(Tree,SI,HT,IIT)
Call Create Perturb FPUTT-Tree
For each path T_i in DB do
P_i = Find a Path that has T_i in TiD in Tree
if($P_j \neq \emptyset$) then
PDB=PDB ∪ P_j and items with P_j.TiD in IIT
else PDB = PDB ∪ P_j

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

3.3 Phase 2: Applying the maximal property and closed property to find the maximal and closed item set.

3.3.1 Maximal Property:

An item set is maximal if it is frequent, but none of its proper supersets is frequent. Consider the set of maximal (frequent) item sets:

$$\mathbf{M_T(S_{min}) = \{I \sqsubseteq B | S_T(I) \geq S_{min} \wedge \forall \sqsupset I : s_t(J) < S_{min}\}}$$

Here M_T –maximal itemset, S_{min} minimumsupport ,B - item base, I and J –itemset, frequent item sets $F_T(S_{min}) = \{I \sqsubseteq B | S_T(I) \geq S_{min}\}$,T-transactional databases.

3.3.2 Closed Property:

An item set is closed if it is frequent, but none of its proper supersets has the same support Consider the set of closed (frequent) item sets:

Here M_T –maximal itemset,S_{min} _minimumsupport, B - item base, I and J –itemset,

$$\mathbf{C_T(S_{min}) = \{I \sqsubseteq B | S_T(I) \geq S_{min} \wedge \forall \sqsupset I : s_t(J) < s_t(I)\}}$$

Frequent item sets$F_T(S_{min}) = \{I \sqsubseteq B | S_T(I) \geq S_{min}\}$, T-transactional databases. The maximal and closed itemsets are generated based on the maximal and closed properties.

Pseudo Code for Maximal and Closed Pattern Mining Algorithm

Input: Original Database DB, Minimum Utility Threshold, A sensitive Item sets SI

Output: A perturbed Database PDB.

Algorithm

```
Call Create UT-Tree (Tree,SI,HT,IIT)
Call Create Maximal-Closed Pattern
For each Item in Itemset do
getItemSuperSet
if(!Item Superset is Frequent) then
Itemset=itemset∪ Item
For each Item in MaxItemset do
```

```
getItemSuperSet
if(ItemSuperSet>min_support) then
Itemset=itemset∪ Item
For each path Ti in DB do
Pi = Find a Path that has Ti in TiD in Tree
if(Pj≠ ∅) then
PDB=PDB ∪ Pj and items with Pj.TiD in IIT
else PDB = PDB ∪ Pj
```

3.4 Phase 3: Protecting the sensitive items using anonymization technique

In this module the sensitive items are protected by anonymization process, the algorithm used for anonymization is UAR (Update-Anonymize-Reorder) aims at anonymizing a dataset so that it satisfies the specified privacy and utility constraints and incurs minimal information loss. The four steps are following are Initialize privacy constraint queue

1. Update phase
2. Anonymization phase
3. Reordering phase

3.4.1 Initialize privacy constraint queue:

To achieve this, UAR stores the privacy constraints in a priority queue UAR selects a privacy constraint from the priority queue and processes it in three phases

3.4.2 Update phase:

The items of the selected privacy constraint are updated to reflect the generalizations and/or suppressions that have occurred in previous iterations. This way, the algorithm avoids the computational overhead that updating all privacy and utility constraints after each item transformation would bring.

3.4.3 Anonymization phase:

UAR attempts to generalize a selected item (or generalized item) in the privacy constraint that is being processed, using the set-based anonymization model. If this would violate the utility constraints, UAR applies suppression to satisfy the selected privacy constraint.

3.4.4 Reorder phase:

The auxiliary information about some of the privacy constraints in the priority queue is updated, and the priority queue is reorganized accordingly. This ensures that the selected privacy constraint. Thus Anonymization is the process that stores the item sets in form range of values so that high privacy is achieved for the item set using UAR algorithm.

Pseudo Code for UT-HUM Tree Algorithm

Input: Original Database DB, Minimum Utility Threshold, A sensitive Item sets SI, Profit Item (PI)

Output: A perturbed Database PDB.

Algorithm

```
Call Create UT-Tree(Tree,SI,HT,IIT)
Call Create ItemsetsHUM(Tree,PI)
For each item in Itemset
If(count>PI)
Itemsets=Itemsets ∪ Item.
Else
Tree Remove(Item)
For each path Ti in DB do
Pi = Find a Path that has Ti in TiD in Tree
if(Pj≠ ∅) then
PDB=PDB ∪ Pj and items with Pj.TiD in IIT
else PDB = PDB ∪ Pj
```

3.5 Phase 4: Secured transactional data streams containing the sensitive items

In this module as the result of applying the anonymization technique, the sensitive items in the transactional data streams are highly secured and thus confidential items can be easily protected by using this approach.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Advantages of the Proposed Work

1. In the Proposed System the Anonymization techniques for protecting the Sensitive item sets and handling the datastream.

2. The anonymization algorithm for using to reduce the Scanning time and low memory Space.

3. In the mining techniques find the maximal and closed itemsets.

IV. EXPERIMENTAL EVALUATIONS

4.1 Data set

This experimental result purpose two datasets are used called retail data set and synthetic data set.In the retail data there are various attributes like bread, peanut butter, milk, coffee and jelly with different number of transaction and one more dataset is synthetic data set which contains attributes like bread, jelly, peanut butter, milk and coffee with 1000 transaction. These two data sets are used for experimental purpose in order to prove the effectiveness of proposed techniques.

4.2 Execution time

Execution time is considered as the total time required for processing a dataset. In this research, the execution time is taken in seconds. The execution time of dataset is evaluated with respect to the minimum threshold taken in % while dataset is evaluated in terms of number of transactions. In this two datasets are used for experimental purpose are retail dataset and synthetic dataset.

The experiment Fig 4.1 and Table 4.1 perform the time comparison on the retail data set,in figure the time measurement should mention in retail data set and Average transaction length and table for comparison of time for real data set are included the time taken to execute.

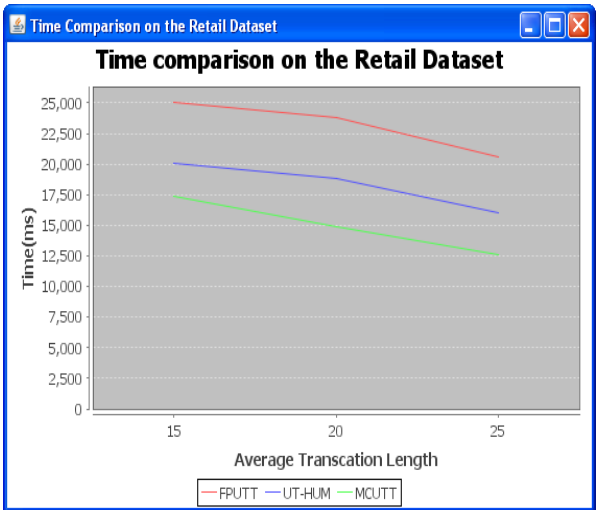

Figure 4.1: Comparison of time vs. Average transaction length on the retail dataset

Table 4.1: Comparison of time vs. Average transaction length on the retail dataset

Average transaction length	Techniques (time)		
	FPUTT	HUM-UT	MCUT
15	25083.1507931 19	20099.602152124342	17383.327401173374
20	23762.414434934246	18777.0284307454	14874.038343544275
25	20571.546444706797	15969.73212532212	12629.274378629058

The experiment Fig 4.2 and table 4.2 perform the time comparison on the synthetic data set,in the figure time measurement should mention in synthetic data set and Average transaction length and table for comparison of time are included the time taken to execute.

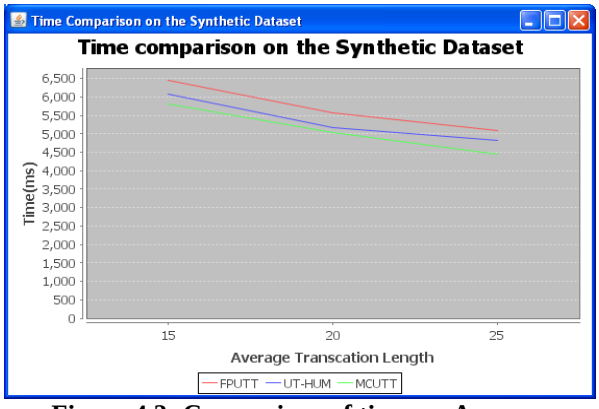

Figure 4.2: Comparison of time vs. Average transaction length on the synthetic dataset

Table 4.2: Comparison of time vs. Average transaction length on the synthetic dataset

Average transaction length	Techniques (time)		
	FPUTT	HUM-UT	MCUT
15	6461.309956444656	6069.565293944594	5810.138573655559
20	5578.256177938939	5178.314474752939	5028.887244119221
25	5082.435726266729	4829.99419263532	4447.83601566722

The above figures and tables show that, the time taken to execute MCUTT algorithm is comparative low when compared to FPUTT and HUMUT.

4.3Memory Usage

Memory usage is an important concern in the mining operations. The larger the datasets results in larger memory consumption. The main aim of efficient pattern mining approaches is to reduce unwanted pattern generation and reducing the overall memory usage. In this

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

research, the memory usage for the mining in the dataset is evaluated against the minimum threshold while the dataset is evaluated against the number of transactions.
The experiment Fig 4.3 nd Table 4.3 perform the memory comparison on the retail data set,in figure the memory measurement should mention in retail data set, average transaction length and table for comparison of time are included the memory taken to execute.

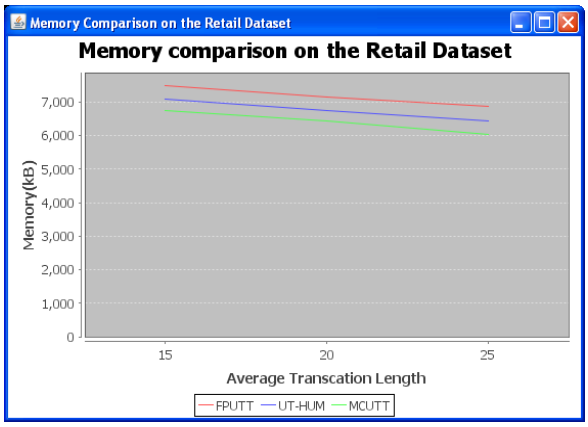

Figure 4.3: Comparison of memory vs Average transaction length on the retail dataset

Table 4.3: Comparison of memory vs Average transaction length on the retail dataset

Average transaction length	Techniques (memory)		
	FPUTT	HUM-UT	MCUT
15	7507.655254515124	7076.317891990475	6765.620053016744
20	7160.376321183585	6736.439958667199	6445.504006906385
25	6876.465123069424	6449.794698318185	6050.009802633491

The experiment fig 4.4 and table 4.4 perform the memory comparison on the synthetic data set,in figure the memory measurement should mention in synthetic data set, average transaction length and table for comparison of time are included the memory taken to execute.

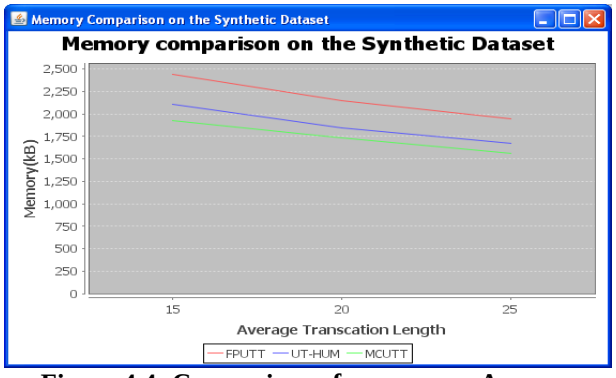

Figure 4.4: Comparison of memory vs Average transaction length on the synthetic dataset

Table 4.4: Comparison of memory vs Average transaction length on the synthetic dataset

Average transaction length	Techniques (memory)		
	FPUTT	HUM-UT	MCUT
15	2445.613389839881	2111.831575402545	1927.7043128714975
20	2150.7807696481646	1846.4631432905287	1735.0811605690617
25	1950.2545404369112	1675.144116787661	1561.6156070645868

The above figures and tables show that, the memory consumption during execution for MCUTT algorithm is comparative low when compared to FPUTT and HUMUT.

4.5 Candidate Comparison

Candidate comparison is the number of candidates involved in the transaction dataset. The experimentFig 4.5 and Table 4.5 perform the number of candidate comparison on the retail data set,in figure the measurement of candidate performance should mention in retail data set ,average transaction length and table for comparison of time are included the number of candidate list are taken to execute.

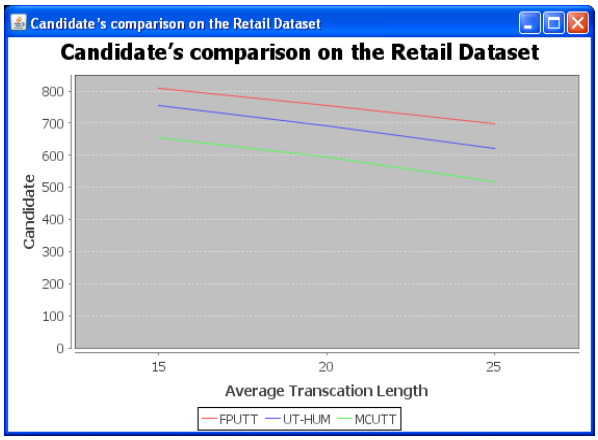

Figure 4.5: Comparison of candidate vs. average transaction length on the retail dataset

Table 4.5: Comparison of candidate vs. average transaction length on the retail dataset

Average transaction length	Techniques (candidate)		
	FPUTT	HUM-UT	MCUT
15	810	755	655
20	756	692	593
25	697	620	519

The experiment Fig 4.6 and table 4.6 perform the number of candidate comparison on the synthetic data set,in figure the measurement of candidate performance should mention in synthetic data set, average transaction length

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

and table for comparison of time are included the number of candidate list are taken to execute.

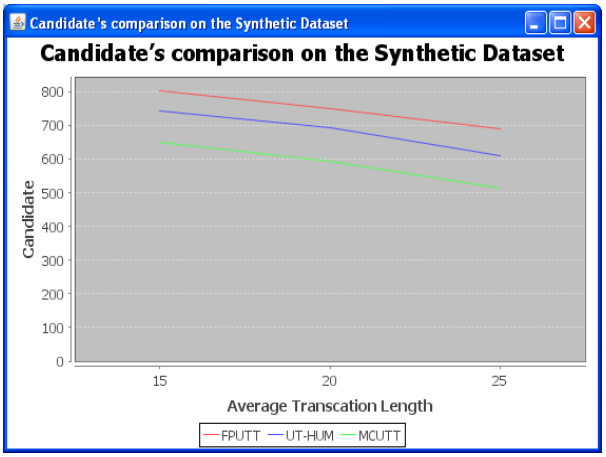

Figure.4.6: Comparison of candidate vs. average transaction length on the synthetic dataset

Table 4.6: Comparison of candidate vs. average transaction length on the synthetic dataset

Average transaction length	Techniques (candidate)		
	FPUTT	HUM-UT	MCUT
15	805	745	652
20	752	694	594
25	692	611	513

The above figures and tables show that, the candidate in the transaction during execution for MCUTT algorithm is comparative low when compared to FPUTT and HUMUT.

V. CONCLUSION

In proposed scenario an utility pattern mining extracts highly utilized items accurately, but it suffers due to the privacy breaches. In order to overcome the drawback of utility mining, PPUM (Privacy Preserving Utility Mining) is utilized which combination of utility and privacy is preserving data mining. Privacy preserving data mining preserves the sensitive data by modifying the structure of the data. Using anonymization the maintaining high utility item into appropriate memory, Maximal and closest item set techniques for used to reduce the memory and time and execute number of Candidates. The proposed system implements anonymization, for providing increased privacy to the database and there by ensures high security to the sensitive itemset and implementing HUM-UT overcomes the drawback of the existing system which does handle data stream, in additional maximal and closed item set extracts from the database and also comparison of real and Synthetic datasets are included. Thus the proposed system is efficient than the existing since it provides increased security and also handles data stream, Maximal and closed items are also identified.

REFERENCES

[1] Lan, G. C., Hong, T. P., Tseng, V. S., & Wang, S. L. (2012, August). An improved approach for sequential utility pattern mining. In Granular Computing (GrC), 2012 IEEE International Conference on (pp. 226-230). IEEE.
[2] Yun, U., & Kim, J. (2015). A fast perturbation algorithm using tree structure for privacy preserving utility mining. Expert Systems with Applications,42(3), 1149-1165.
[3] Arunkumar, M. Mining High Utility Itemsets from its Concise and Lossless Representations. IOSR Journals (IOSR Journal of Computer Engineering),1(17), 8-14.
[4] Pyun, G., Yun, U., & Ryu, K. H. (2014). Efficient frequent pattern mining based on linear prefix tree. Knowledge-Based Systems, 55, 125-139.
[5] Navaz, A. S., Ravi, M., & Prabhu, T. (2013). Preventing Disclosure of Sensitive Knowledge by Hiding Inference. arXiv preprint arXiv:1308.6744.
[6] Duraiswamy, K., Manjula, D., & Maheswari, N. (2008). A new approach to sensitive rule hiding. Computer and Information Science, 1(3), 107.
[7] Li, Y., Chen, M., Li, Q., & Zhang, W. (2012). Enabling multilevel trust in privacy preserving data mining. IEEE Transactions on Knowledge and Data Engineering, 24(9), 1598-1612.
[8] Tseng, V. S., Wu, C. W., Shie, B. E., & Yu, P. S. (2010, July). UP-Growth: an efficient algorithm for high utility itemset mining. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 253-262). ACM.
[9] Tzvetkov, P., Yan, X., & Han, J. (2005). TSP: Mining top-k closed sequential patterns. Knowledge and Information Systems, 7(4), 438-457.
[10] Vidyabanu, R., & Nagaveni, N. (2010). A model based framework for privacy preserving clustering using SOM. International Journal of Computer Applications, 1(13).
[11] Agrawal, Rakesh, and RamakrishnanSrikant. "Fast algorithms for mining association rules." Proc. 20th int. conf. very large data bases, VLDB. Vol. 1215. 1994.
[12] Evfimievski, Alexandre, et al. "Privacy preserving mining of association rules." Information Systems 29.4 (2004): 343-364.
[13] JayasudhaV.Umarani:" EUP-Growth+ - Efficient Algorithm for Mining HighUtility Itemset