GENERATING HIGH UTILITY PATTERNS FROM LARGE DATA SETS USING REDUCED TRANSACTION PATTERN LIST

[1]S.Srilekha, [2]Dr. R. Jegadeesan [3]M.Amulya, [4]Ch.Anil, [5]G.Mahesh, [6]G.Srilatha

[1,3,4,5,6] U.G.B.Tech Final Year Student

[2,6]Associate Professor-Department Of Computer Science and Engineering

[1,2,3,4,5,6]Jyothishmathi Institute of Technology and Science,Karimnagar.India

Abstract : The frequent item set mining will not address both the quantity and profit of the item sets. The high utility itemset mining is an emerging area of mining which considers and addresses the quantity and profit of the item sets in the database. High utility itemset mining is done efficiently by using the SPHUI-Mining algorithm, which reduces the time and space complexity by using the reduced transaction pattern list (RTPL) and projects the reduced transaction pattern list. Selective projections of the database used in the SPHUI-mining algorithm plays a major role in reducing the scanning time of the database, which makes the approach more effective. Most challenging part of the HUI mining is the exponential complexity in both time and space. HUIM is used in wide range of applications, such as the analysis of clickstream on websites, mobile computing, finding top-k itemsets and biomedical applications.

Index Terms: Data mining, frequent itemset mining, high utility mining, database projection.

I.Introduction

Data mining is a prominent and important research area for extracting the information contained in large databases. Discovering patterns hidden, unexpected trends in the data is the primary goal of the data mining.

An essential function in several data mining tasks, such as frequent pattern mining, weighted frequent pattern mining and high utility pattern mining, is to discover the useful patterns hidden in a database.

The sets of items that appear frequently in the transaction are called sets of frequent items. Identifying all sets of frequent items in a transaction data set is the goal of frequent mining. Support value of the itemsets is the criterion of being frequent. The number of transactions that contain the item is termed as the support value. The frequent itemsets are generated by considering the minimum support value. The frequent itemsets are generated from itemsets the which satisfies the minimum support value.

Generally, during the mining process, our aim should not be to identify frequent itemsets but our aim should be to identify itemsets which are more utilizable to us. Thus, a new approach in data mining which is based on the concept of itemset utility called as utility mining is proposed. A utility based mining approach, gives flexibility to the user to express their perspective for the usefulness of as utility values and then find which have higher utility values than the threshold.

This approach has been proposed as the limitation of frequent mining motivated the researchers. The frequent item-set mining follows the downward-closure property where the support value of an itemset is anti-monotonic. Anti-monotonic is defined as the subsets of a frequent are frequent and supersets of an infrequent are infrequent, which is an efficient property to trim the search space. As high utility may have a super set or subset with lower, equal or higher utility, the HUIM does not follow monotonic or anti-monotonic properties. So, as to prune the search space in HUIM, a compact data format named high utility-reduces transaction pattern list(RTPL) is generated with the help of the transaction bitmap matrix(TBM).

The process of high-utility itemset mining remains costly in terms of time and space complexity as it creates projections on all promising itemsets. In order to address this challenge of high cost by providing efficient data representation and selective database projection which improves pruning technique of the search space.

Section II is the literature survey. Section III introduces the definitions. Section IV introduces the implementation. Section V is the conclusion.

II.Literature Survey

Transaction-Weighted Downward Closure Property is followed by the methods used for mining high utility itemset. Only the combinations of high transaction weighted utilization items are added into the candidate set at each level during the level-wise search. Sometimes, phase I may overestimate some low utility items, but it never underestimates any items. So, one additional scan is conducted for the database to filter the overstated material in the phase II[1].

Two-phase algorithm generates a large number of candidates because of the level-wise method. A new Compressed Transaction Utility (CTU) Mine algorithm that extracts high utility items using a pattern growth approach is proposed [2] .

Temporal High Utility Itemset (THUI) for mining temporal high utility itemsets, efficiently and effectively, is proposed. THUI-Mine can effectively identify all itemsets of high temporal utility by generating fewer sets of elements of weighted use of transactions in such a way that the execution time can be substantially reduced by extracting all sets of highly useful elements in data flows. This meets the critical time and space efficiency requirements for data mining [1].

To address issue of generating a large number of candidates, (Utility Pattern) UP-Growth has recently been proposed and it uses Potential High Utility (PHU) model. For reducing the number of candidate, the UP-Growth applies four strategies, Discarding Global Unpromising items (DGU), Decreasing Global Node utilities (DGN), Discarding Local Unpromising items (DLU), and Decreasing Local Node utilities (DLN). A UP-Tree, a tree structure, is built with two database scans and performs sets of high utility mining [3].

High Utility Itemset Mining(HUIM) is a high utility with a list data structure, called utility list. It first creates an initial utility list for itemsets of the length 1 for promising items. Then, High Utility Itemset (HUI)-Miner constructs recursively a utility list for each of the length k using a pair of utility lists for itemsets of the length k-1. For mining high utility, each utility list for an itemset keeps the information of TIDs for all transactions containing the itemset, utility values of the item set in the transactions, and the sum of utilities of the remaining items that can be included to super itemsets of the itemset in the transactions [4].

Faster high utility (FHM) extends the HUI-Miner Algorithm. It is a Depth-first search Algorithm. It relies on utility-lists to calculate the exact utility of the itemsets. This algorithm integrates a novel strategy called Estimated Utility Co-Occurrence Pruning (EUCP) to reduce the number of join operations by extracting sets of high-utility elements using the data structure of the utility list. The TWU of all 2-itemsets is in Estimated Utility Co-Occurrence Structure (EUCS) [5].

Efficient High Utility Mining(EFIM), which introduces several new ideas to more efficiently discovers high-utility item sets both in terms of execution time and memory. EFIM relies on two upper-bounds named sub-tree utility and local utility to more effectively prune the search space. This algorithm also introduces Fast Utility Counting, a novel array-based utility counting technique, to calculate these upper-bounds in linear time and space. Transaction merging is obviously desirable. However, a key problem is to implement it efficiently [6].

An efficient utility mining approach is proposed that adopts an indexing mechanism to accelerate the execution and reduce the memory requirement in the mining process. The indexing mechanism can imitate the traditional projection algorithms to achieve the objective of projecting sub-databases for mining. In addition, a pruning strategy is also applied to reduce the number of sets of unpromising elements in mining [7].

The growth of the utility pattern (UP-Growth) and UP-Growth + are proposed, for the extraction of sets of high-profit elements with a set of effective strategies for the elimination of sets of candidate elements. High utility element set information is maintained in a tree-based data structure called a utility pattern tree (UP-Tree), so that sets of candidate elements can be generated efficiently with only two scans of the database [8].

The two-phase approach has problems due to the large number of candidates. This article proposes a novel algorithm that finds highly useful patterns in a single phase without generating candidates. The novelties are in a highly useful pattern growth approach, an early search strategy and a linear data structure. Specifically, the pattern growth approach is to look for an inverse set enumeration tree and eliminate the search space by the upper limit of the utility. Discovery of association rules has become a major problem in knowledge discovery and data extraction. The task of association mining is to identify the sets of common elements and then form conditional implication rules between them [9].

An efficient algorithm for the discovery of sets of frequent elements that form the intensive processing phase of the task is presented. Algorithms use the structural properties of frequent element sets to facilitate rapid discovery. The elements are organized in a subset of lattice search, which is divided into small fragments or independent sub structures, which can be resolved in memory. Efficient network routing techniques are presented that quickly identify all common element sets and their subsets, if necessary[10] .

III.PROPOSED METHOD

To develop the SPHUI-Miner algorithm the following ideas have been taken into considerations:

1. **HUI-RTPL:** High utility transaction pattern list is a compact data format to represent a database which stores only unique transactions. The transaction list is reduced using the HUI-RTPL, making the memory requirement independent from the size of original database.
2. **Projection Creation:** The ordered projections are recursively created using the SPHUI-Miner algorithm and merges transactions that are similar the worst-case search cost, a database projection is created only if the cost is less. This algorithm also performs selective database projections on dimensions for mining HUI.
3. **Upper bound:** To effectively prune the search space the loose upper bound and tighter upper bound are designed.

To optimize a database projection, there are two ways. They are:

● The first way is to reduce the cost of database scan by representing database using efficient HUI-RTPL by creating projections selectively.

● The second way is by enumerating less number of probable candidates using SPU-list and tail count list(t_l).

Selective projection based approach and novel techniques to reduce the time and memory requirement are present.Selective projection based approach creates new projections of smaller sizes in terms of vertical columns(number of items) horizontal rows(number of unique transactions). These results in faster computing of utility of itemsets. There are two upper bounds with pruning strategies to reduce the overestimation of the utilities of the itemsets, which reduces the search space for discovering HUIs and many unpromising candidates can be pruned early.

A.PHASE 1

To determine high transaction-weighted utility itemsets, SPHUI-Miner algorithm uses upper bound TWU model. TWU model uses the transaction weighted downward closure (TWDC) property for pruning all the unpromising items. In this phase,

1. Calculate utility of items in each transaction(transaction utility).
2. Calculate the transaction weighted utility(twu) of an item.
3. If twu ≥ minimum utility value then it is considered as high transaction weighted utility itemsets.

The utility values are computed from the definitions used in the paper[11].

B.PHASE 2

In this phase, only one copy of these similar transactions is stored along with their total items utility. This unique and complete database representation using this structure is called high utility reduced transaction pattern list(HUI-RTPL)(D1).

a.Construction of the database D_1:

1.A transaction bitmap matrix(TBM) is created which contains 0 and 1. 0 represents presence of item in transaction. 1 represents absence of item in transaction.

2.The size of the database is compressed, as the size of the database is large, by keeping only relevant information required for mining. Using HUI-RTPL(D1), the number of transaction in the database is reduced by merging similar transactions as well as utilities of items. This technique:

- The database can be compressed vertically.
- To store database it requires less memory.
- It also reduces the time taken to scan the database.

HUI-RTPL: It is an instance of projected database, which compresses the database and quickens computing of utility of itemsets.

C.PHASE 3:

Reducing the cost of database scans is a better way to reduce the database size. Due to this, the database projection based approach into consideration. The database scanning time is reduced by the database projection.

D.PHASE 4:

For optimizing the search space, less number of probable candidates should be generated. There are two novel data structures, they are

1. SPU-LIST
2. TAIL-COUNT LIST

Two novel upper bounds are:

1. Transaction utility in projection(tup).
2. Projection utility(pu) for each item in the projection.

These two reduce the number of unpromising itemsets.

1.Computing tail-count list in the projection P_X : Tail-count list structure reduces the exploration of items in extension list having same cardinality as that of X.

2.Computing upper bounds for projection P_X : Designing effective and powerful search space pruning techniques is a key challenge in HUI mining. The search space pruning techniques are used to discover efficiently. Selective projection based utility structure (SPU-List) is designed which,

1. Reduces the cost of multiple database scans.
2. Removes impertinent items efficiently.

a.Pruning techniques: If the unpromising supersets can be identified early then pruning strategies can greatly reduce the search space.

- Let X be an itemset, if twu(X) < minutil then X and its supersets are low utility itemsets.
- Let X be an and an item i E E. For Y=X U i, if pu(Y,P_X)<minutil, it indicates itemset Y and its extensions are of low utility. If tup(Y,P_X)<minutil, its extensions are of low utility in the projection.

By using these two pruning steps, we can identify the unpromising.

E.PHASE 5:

Creation of new projections selectively from the parent projection. After computing tup and pu, compute e_l, c_l and t_l list. All items in the projection, which are not present in c_l and e_l are ignored as they cannot be part of any high utility. Then smaller projection is created. The itemsets which are considered as the HUI from the projection should satisfy the condition u(x) ≥ minutil.The recursion algorithm is recursively executed until no candidates are generated.The termination of algorithm gives HUIs obtained.

F.PHASE 6:

A non-selective approach is used to create projection for each item explored. But the (memory) space complexity can be more, so the projections should be created judiciously. A judicious method is one which is employed to create the projection has one condition, that is, to create a new projection, the size of the new projections should be less than half of the current projection. In addition to that an upper bound is found on the number of the new projection may have.

G.ALGORITHMS

Algorithm 1: Construction of Database (H1):

Step 1: Give transactional database, D and external utility table, p as input.

Step 2: Initialize new database H_1 to null.

Step 3: For every transaction,'T_q' in D repeat , for every item,'i' in the transaction, add the utility value of the item 'i' in the transaction T_q to the transaction utility of the transaction T_q ($tu(T_q)$).

Step 4: For every transaction in D, add the transaction utility of the transaction T_q to the total utility value of the database (TU).

Step 5: Compute the minutil value by multiplying TU(total transaction) and minimum high utility threshold (δ).

Step 6: For each item 'i' in itemset 'I', Add the transaction utility value of the transaction 'T_q' where i is present in T_q to the transaction weighted utility (twu(i)). If twu(i) is greater than or equal to the 'minutil' value, then add 'i' to the H_1.

Step 7: For every 'T_q' in 'D', remove 'i' if twu(i) is less than 'minutil' value.

Step 8: Sort all items of 'T_q' in 'D', according to twu in ascending value.

Step 9: Return H_1.

Algorithm 2: Construction of Reduced Transaction Pattern List (D,H_1)

Step 1: Give transactional database, D and set of high transaction weighted utility items list, H_1 as input.

Step 2: Construct TBM for each item 'i' in H_1 and sort.

Step 3: For every transaction,'T_q' in D, for every transaction,'T_r' in D, if T_r is equal to T_q, then add the $tu(T_r)$ value to $tu(T_q)$. For every item 'i' in T_r, add the $u(i_r,T_r)$ to the $u(i_q,T_q)$.

Step 4: Return HUI-RTPL.

Algorithm 3: Pruning the projection P_X:

Step 1: Give P_x, projection on itemset X, X, an itemset, E, extensions of X as input.

Step 2: for every 'i' in c_l, add 'i' to 'X' and store it in an itemset, 'Y'. Add $u(Y,T_q)$ for every T in P_X and u(i,T) + remaining utility value of (Y,T) for every i in T, in c_l to tup(Y,P_X). Add $u(Y,T_q)$ and re(Y,T) to pu(Y,P_X). Compute t_l(i).

Step 3: Set e_l to null and c_l to null.

Step 4: for every i in c_l, add 'i' to 'X' and store it in an itemset, 'Y'. If pu(Y,P_X) is greater than or equal to minutil, then add i to e_l. If tup(Y,P_X) is greater than or equal to minutil, then add i to c_l.

Step 5: if t_l(i) is equal to |P_X| then Add i to X Remove i from e_l Remove i from c_l.

Step 6: For each i in PEP-list, check utility and update HUI.

Step 7: Return e_l,c_l.

Algorithm 4: Construction Of New Projection:

Step 1: Give P_x, projection on itemset X, i, item, t_l, tail count as input.

Step 2: If |P_x| is greater than or equal to t_l(i), then Add the transaction T_q to the new projection ($P_{x'}$). For every transaction,'$T`_q$' in $P_{x`}$, for every transaction,'$T`_r$' in '$P_{x`}$', if T_r is equal to T_q, then add the $tu(T`_r)$ value to $tu(T`_q)$. For every item 'i' in $T`_r$, add the $u(i_r,T`_r)$ to the $u(i_q,T`_q)$.

Step 3: Else, store all the items of P_x to $P_{x'}$(new projection).

Step 4: Return $P_{x`}$.

Algorithm 5: Compress Projection $P_{X`}$:

Step 1: Give P_x, projection on itemset X, i, item, e_l , extension list of itemset X as input.

Step 2: for every transaction T_q in P_x , if item 'i' is present in T_q, then Remove the item 'i' from the T_q and store in the $T`_q$. Remove the utility value of 'i' ($u(i,T_q)$) from $tu(T_q)$ and store in the $tu(T`_q)$. Else, store all the values of T_q in $T`_q$ and store all the values of $tu(T_q)$ in the $tu(T`_q)$.

Step 3: Ignore the bit representing 'i' in P_X.

Step 4: for every transaction $T`_q$ in $P_X$, for every transaction,'$T`_r$' in 'P_x', if $T`_r$ is equal to $T`_q$, then add the $tu(T`_r)$ value to $tu(T`_q)$. For every item 'i' in $T`_q$, add the $u(i_r,T`_r)$ to the $u(i_q,T`_q)$.

Step 5: Return $P_{X`}$

Algorithm 6: Recursive Miner:

Step 1: Give X, an itemset, e_l, extension list of X; c_l, candidates list of X; P_X, projection on itemset X as input.

Step 2: for every 'i' in e_l,Compute e_l,c_l,t_l using the algorithm 3. Then construct projection P_X.

Step 3: for every 'j' in e_l, excluding 'i', Add 'j' item to the itemset 'X'.

Step 4: Check utility u(X) and update HUI.

Step 5: if u(X) is greater than or equal to 'minutil', then Add X to the HUI.

Step 6: if new database projection is created then Compress the projection using algorithm 5.

Step 7: Else, store the projection 'P_X' to '$P_{X`}$'.

Step 8: Repeat steps 2 to 7 for every e_l created.

Step 9: Return HUIs.

Algorithm 7: High Utility Itemset Miner

Step 1: Give D, transactional database, a external utility table, 'minutil', a minimum high utlity threshold as input.

Step 2: Set HUI to null, e_l to null, c_l to null, t_l to null.

Step 3: Construct H_1 using algorithm 1.

Step 4: Reduce the transaction pattern list H_1 and store in D_1.

Step 5: Set e_l from H_1.

Step 6: Set c_l from H_1.

Step 7: Compute e_l, c_l, t_l using Algorithm 3.

Step 8: for every i in e_l, add i to the X. Check utility u(X) If u(X) is greater than or equal to minutil, then add X to the HUI.

Step 9: Using algorithm 4 construct $P_{X`}$.

Step 10: Algorithm 6

Step 11: if new database projection is created then Store the P_X values in the $P_{X`}$ using algorithm 5.

Step 12: Else, store D_1 in $P_{X`}$.

Step 13: Return HUIs.

IV.EXPERIMENTAL ANALYSIS

An exceptional number of experiments are performed to verify the accuracy of the proposed algorithm. For generating high utility patterns from large data sets, the algorithms are implemented in Java. An Intel(R) Core(TM) i5 8250 CPU@ 1.6GHz processor with 8GB of RAM system, running on the 64 bit Microsoft Windows 10 OS, is used for execution.

A. RUNTIME PERFORMANCE

Comparison of the runtime performance of the algorithms is done in this section. Selectively creating the database projection is the key reason for faster performance. This generates valid combinations of sets of existing elements in the projected database. Besides, it also indicates that the two proposed upper-bounds can be used to efficiently and effectively prune unpromising candidates early, while mining the HUIs. SPHUI-Miner prunes a larger number of unpromising candidates as compared to the fastest algorithm EFIM when minutil threshold is set to low. In summary, the SPHUI-Miner algorithm with its pruning strategies outperforms the other algorithms since the search space is considerably reduced by adopting the looser and tighter upper-bounds with Tail-Count list.

B. MEMORY CONSUMPTION

In this section, we compare the memory usage of the algorithms. In this algorithm, the memory usage is reduced as the large number of unpromising candidates will be pruned using the proposed upper bound models. In the SPHUI Miner algorithm, the proposed Tail-Count list with bound condition is applied to reduce the memory usage in the form of selective projection. Hence, the memory usage of the proposed SPHUI-Miner algorithm is considerably less than that of the other state-of-the-art algorithms.

V.CONCLUSION

In this paper, we have presented a novel selective database projection based approach, called SPHUI-Miner, which integrates upper bounds with pruning strategies to efficiently prune unpromising candidates in the search space while mining the HUIs. The proposed algorithm introduces HUI-RTPL to store database in a compact structure. During selective database projection, it compresses the database relevant to the itemset being investigated. This structure significantly reduces the memory consumption. The search over the projections is optimized for faster retrieval of HUIs using Tail- Count and SPU-List. The combined effect of faster computing of utility of itemsets and integrated pruning techniques results in significant performance increase. We have also shown that at any time during the creation of projection, the algorithm limits the number of transactions held by newly created projections, which is half the number of transactions in the parent database. For all datasets, the proposed algorithm prunes the tree efficiently and thus requires very less amount of time and comparable memory. This is the first algorithm which performs selective database projection not only on data instance but also on dimensions for mining HUI. The scalability of SPHUI-Miner is also studied and it is proved experimentally that the algorithm is scalable in terms of memory as well as time. The design of the algorithm is independent of the order in which projections are processed, making it suitable for distributed or parallel implementation. These projections can be stored on disk and processed one by one. Our linear database representation allows us to estimate a tight bound for efficient pruning and directly find high utility itemsets in an effective and scalable way. Proposed algorithm exhibits high candidates pruning around 2% to 15% as compared to the EFIM algorithm. Extensive experiments on sparse and dense datasets suggest that the proposed algorithm significantly outperforms the state-of-the-art algorithms.

However, in real-life, a database may dynamically change as a result of insertion, deletion, and modification operations and mining can be difficult in an incremental database. To overcome this issue, we need to explore selective database projections for mining high average-utility itemsets for the incremental databases specifically using proposed HUIRTPL structure with proposed upper bounds and pruning techniques.

References

[1] Overview of Itemset Utility Mining and its Applications, Jyothi Pillai, O.P.Vyas, Professor, Indian Institute of Information Technology, Allahabad, U.P., India.

[2] High Utility Itemset Mining Sudip Bhattacharya, Research Scholar, Dept. of Computer Science & Engg., Chhatrapati Shivaji Institute of Technology, Durg, India; Deepty Dubey, Asst. Professor, Dept .of Computer Science & Engg., Chhatrapati Shivaji Institute of Technology, Durg, India.

[3]Hua-Fu Li, Hasin-Yug Huang, Yi-Cheng Chen, Yu-Jiun , Suh-Yin Lee, Fast and Memory Efficient Mining of High Utility in Data Streams, IEEE International Conference on Data Mining,2008.

[4]A Survey on Discovering High Utility Itemset Mining from Transactional Database, Shekhar Patel, B Madhushree L.J Institute of Engineering and Technology, GTU,Ahmedabad, India.

[5]FHM: Faster High-Utility Itemset Mining Using Estimated Utility Co-occurrence Pruning; Philippe Fournier-Viger, Cheng-Wei Wu, Souleymane Zida, and Vincent S. Tseng

[6]Survey on Mining High Utility Patterns in One Phase; Harshita Taran M.Tech Student, Computer Science and Engineering Kavikulguru Institute of Technology And Science Rmatek(M.H.),India, Shilpa Ghode Assistant Professor, Computer Science and Engineering Kavikulguru Institute of Technology And Science Rmatek(M.H.), India.

[7]A Projection-Based Approach for Discovering High Average-Utility Itemsets; guo-cheng Lan, Tzung-Pei Hong, Vincent S. Tseng

[8]Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases;Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, IEEE

[9]Mining High Utility Patterns in One Phase without Generating Candidates; Junqiang Liu, Member, IEEE, Ke Wang, Senior Member, IEEE, and Benjamin C.M. Fung, Senior Member, IEEE

[10]Scalable Algorithms for Association Mining; Mohammed J.Zaki.

[11]Selective Database Projections Based Approach for Mining High-Utility Itemsets; ANITA BAI, PARAG S. DESHPANDE, AND MEERA DHABU, (Member, IEEE).

[12].Jegadeesan,R.,Sankar Ram M.Naveen Kumar JAN 2013 “Less Cost Any Routing With Energy Cost Optimization” International Journal of Advanced Research in Computer Networking,Wireless and Mobile Communications.Volume-No.1: Page no: Issue-No.1 Impact Factor = 1.5

[13]. Jegadeesan,R.,Sankar Ram, R.Janakiraman September-October 2013 “A Recent Approach to Organise Structured Data in Mobile Environment” R.Jegadeesan et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (6) ,Page No. 848-852 ISSN: 0975-9646 Impact Factor:2.93

[14]. Jegadeesan,R., Sankar Ram October -2013 “ENROUTING TECHNICS USING DYNAMIC WIRELESS NETWORKS” International Journal of Asia Pacific Journal of Research Ph.D Research Scholar 1, Supervisor2, VOL -3 Page No: Print-ISSN-2320-5504 impact factor 0.433

[15]. Jegadeesan,R., Sankar Ram, M.S.Tharani (September-October, 2013) ”Enhancing File Security by Integrating Steganography Technique in Linux Kernel” Global journal of Engineering,Design & Technology G.J. E.D.T., Vol. 2(5): Page No:9-14 ISSN: 2319 – 7293

[16]. Ramesh,R., Vinoth Kumar,R., and Jegadeesan,R., January 2014 “NTH THIRD PARTY AUDITING FOR DATA INTEGRITY IN CLOUD” Asia Pacific Journal of Research Vol: I Issue XIII, ISSN: 2320-5504, E-ISSN-2347-4793 Vol: I Issue XIII, Page No: Impact Factor:0.433

[17]. Vijayalakshmi, Balika J Chelliah and Jegadeesan,R., February-2014 “SUODY-Preserving Privacy in Sharing Data with Multi-Vendor for Dynamic Groups“ Global journal of Engineering,Design & Technology. G.J. E.D.T.,Vol.3(1):43-47 (January-February, 2014) ISSN: 2319 –7293

[18]. Jegadeesan,R.,SankarRam,T.Karpagam March-2014 "Defending wireless network using Randomized Routing process" International Journal of Emerging Research in management and Technology

[19].Jegadeesan,R.,T.Karpagam, Dr.N.Sankar Ram , "Defending Wireless Network using Randomized Routing Process" International journal of Emerging Research in management and Technology ISSN: 2278-9359 (Volume-3, Issue-3) . March 2014

[20]. Jegadeesan,R., Sankar Ram "Defending Wireless Sensor Network using Randomized Routing "International Journal of Advanced Research in Computer Science and Software Engineering Volume 5, Issue 9, September 2015 ISSN: 2277 128X Page | 934-938

[21]. Jegadeesan,R., Sankar Ram,N. "Energy-Efficient Wireless Network Communication with Priority Packet Based QoS Scheduling", Asian Journal of Information Technology(AJIT) 15(8): 1396-1404,2016 ISSN: 1682-3915,Medwell Journal,2016 (Annexure-I updated Journal 2016)

[22]. Jegadeesan,R.,Sankar Ram,N. "Energy Consumption Power Aware Data Delivery in Wireless Network", Circuits and Systems, Scientific Research Publisher,2016 (Annexure-I updated Journal 2016)

[23]. Jegadeesan,R., Sankar Ram , and J.Abirmi "Implementing Online Driving License Renewal by Integration of Web Orchestration and Web Choreogrphy" International journal of Advanced Research trends in Engineering and Technology (IJARTET) ISSN:2394-3785 (Volume-5, Issue-1, January 2018

[24]. Pooja,S., Jegadeesan,R., Pavithra,S., and Mounikasri,A., "Identification of Fake Channel Characteristics using Auxiliary Receiver in Wireless Trnsmission" International journal for Scientific Research and Development (IJSRD) ISSN (Online):2321-0613 (Volume-6, Issue-1, Page No. 607-613, April 2018

[25]. Sangeetha,R., Jegadeesan,R., Ramya,P., and Vennila.,G "Health Monitoring System Using Internet of Things" International journal of Engineering Research and Advanced Technology (IJERAT) ISSN :2454-6135 (Volume-4, Issue-3, Page No. 607-613, March 2018.