A Study on Mining Top Utility Itemsets In A Single Phase

B. Srinivas[1], Gadde Ramesh[2], Shoban Babu Sriramoju[3]
[1]Associate Professor, Dept of Computer Science and Engineering
[2]Assistant Professor, Dept of Computer Science and Engineering
[3]Professor, Dept of Computer Science and Engineering
[1, 2] Madda Walabu University, Ethiopia
[2]Ethiopian Institute of Technology, Ethiopia
[3]SR Engineering College, India

Abstract*- This paper presents a study on finding Top K itemsets with high utility. High Utility Item sets (HUI) mining has emerged as an interesting and challenging research topic in data mining. It finds applications in web-click analysis, cross- marketing in retail stores and bio medical analysis etc. The number of high-utility itemsets that can be extracted from a transactional database depends upon the value of minimum utility threshold. It is often difficult for a user to find a suitable threshold value which fits their purpose. The database can generate many high-utility itemsets at low threshold value and very few itemsets at higher threshold values. In order to relieve the user from this tedious task, we use an efficient algorithm named TKO for mining top-k high utility itemsets from a large transactional database. The parameter k can be set by the user according to his/her needs. We conduct extensive experiments on real and synthetic datasets and the experimental results demonstrate the effectiveness of our Threshold raising strategies in terms of total execution time and the memory usage.*

Keywords*-* Utility mining, high utility mining, top k high utility itemset mining

I. INTRODUCTION

Frequent pattern mining finds patterns from a database, which have frequency no less than a given minimum support threshold. Frequent pattern mining finds applications in market-basket analysis, mining association rules [1], plagiarism detection and biomedical data analysis. The algorithms developed for mining frequent patterns have mostly employed the monotone/anti-monotone property to prune the exponential search space effectively. The monotone property states that the subsets of a frequent pattern are also frequent and the anti-monotone property states that the supersets of an infrequent pattern are also infrequent. However, the frequent patterns extracted can be of low profit value. The concept of high-utility pattern mining was introduced to capture the notion of utility, which has been observed in real life. High-utility pattern mining finds patterns from a database which have their utility value no less than a given minimum utility-threshold. The utility function measures the importance of a pattern and varies according to the application. For example, in a retail store domain, a manager may be interested in finding combinations of products with high profits, which relates to the unit profits and purchased quantities of products that are not considered in frequent pattern mining. High-utility pattern mining has wide range of applications in cross-marketing in retail stores, web-click stream analysis, medicine etc. Be that as it may, effectively mining HUIs in databases is not a simple assignment on the grounds that the downward closure property utilized as a part of FIM does not hold for the utility of itemsets. At the end of the day, pruning scan space for HUI mining is troublesome on the grounds that a superset of a low utility itemsets set can be high utility. To handle this issue, the idea of exchange weighted usage (TWU) model was acquainted with encourage the execution of the mining assignment. Although many algorithms have been introduced in HUI mining, it is very difficult for users to set an appropriate minimum support because it highly depends on data types. If it is set too high, no result itemsets are found while too small value makes an enormous number of result patterns which cause inefficiencies in terms of computation time and memory usage. to address this issue, and to control the itemsets with the highest utilities without setting the thresholds, a better solution is to change the task of mining HUIs as mining top-k high utility itemsets. Here the users specify k. Here k is the number of desired itemsets, instead of specifying the minimum utility threshold. Top k HUI mining is used to find, what are the top-k sets of products that contribute the highest profit to the company and how to efficiently found these itemsets without setting the min-utility threshold. A naive approach for extracting top-k high-utility itemsets can be to set the minimum utility threshold to zero and apply any high-utility itemset mining algorithm to and the complete set of high-utility itemsets. Top-k itemsets can be then chosen from the result set. However, this approach is computationally very inefficient as the search space is exponential in the number of different items. In order to improve the efficiency, we propose effective strategies to raise the minimum utility threshold from zero as quickly as possible. top-k HUI mining is essential to many applications, developing efficient algorithms for mining such patterns is not an easy task. It poses four major challenges those are,

A. The utility of itemsets is neither monotone nor anti monotone.
B. How to incorporate the concept of top-k pattern mining with the TWU model.
C. The min_util threshold is not given in advance in top-k HUI mining.
D. How to effectively raise the min_util Border threshold without missing any top-k HUIs.

In this paper we are presents the literature survey study over the concept of Top K high utility itemset mining using the concepts of data mining. we study an efficient algorithm named TKO (mining Top K utility itemsets in One phase) is proposed for mining the complete set of top k HUIs in databases without the need of specify the min_util threshold. Here we conduct the experiments on real and synthetic datasets and these experiment results demonstrate the effectiveness of our approach. The paper is organized as follows. Section 2 reviews the related work and background knowledge and definitions related to utility mining is explained in Section3. TKO algorithm and strategies for improving the performance in Section 4. The experimented results are presented in Section 5 & 6 which concludes the paper.

II. REVIEW OF LITERATURE

High Utility Itemset Mining is a popular concept and many algorithms have been proposed for HUI mining such as two phase[6], IHUP[10], IIDS, UP-Growth[11], D2HUP and HUIMiner[8]. These algorithms can be generally classified in two types: Two-phase and one-phase algorithms. The characteristics of two-phase algorithm is that it consists of two phases. In the first phase, they create a set of candidates that are potential high utility itemsets. In the second phase, the calculation of precise utility of each candidate is found in the first phase to identify high utility itemsets. Two-phase, IHUP, IIDS, and UP-Growth are two-phase based algorithms. D2HUP and HUIminer are one phase based algorithms.et al in [6] proposes two-phase algorithm for finding high utility itemsets is proposed. This paper explains transaction weighted utilization in Phase I, only the combinations of high transaction weighted utilization itemsets are added into the candidate set at each level during the level-wise search. In phase 2, other database scan is performed to remove the overestimated itemsets. Second phase requires fewer database scans, less computational cost and less memory space. It performs efficiently in terms of memory and speed cost both on synthetic and real databases, even on large databases.
Shuning Xing et alIn [4] proposed a process of UP-Tree by introducing a Fast Utility Tree (FU-Tree) is proposed. In this method, they introduce the Link Queue to reduce the number of scanning the original database and adopt prefix utility to minimize the overestimated utility. The theoretical analyses and experimental results show that FU-Tree outperforms UP-Tree in the time consumption of construction trees, and enhances the efficiency of mining high utility itemsets.

M. Liu et In [8] proposed a novel data structure called utility-list, and developed an efficient algorithm HUI Miner, for high utility itemset mining. Utility-lists provide not only utility information about itemsets but also important pruning information for HUI- Miner. The framework of the algorithm HUI-Miner does not generate candidate high utility itemsets. After constructing the initial utility-lists from a original database, HUI-Miner can mine high utility itemsets from these utility-lists without scan the original database. The initial utility lists are considered by scans the database twice. HUI-Miner gains significant performance Improvement over state art algorithms in terms of both running time and memory consumption.

Vincent S. Tseng et al In [2] proposes a novel framework for top-k high utility itemset mining, where k is the desired number of HUIs to be mined is proposed. For mining such itemsets without the need to set min_util we propose the algorithms named TKO (mining Top-K utility itemsets in one phase) and TKU (mining Top-K Utility itemsets).

III. TERMS AND DEFINITIONS

In this section, define the related mathematical definitions for discovering interesting and useful information from a transaction database.

TABLE 1:Transactional Database

Tid	Transaction	Count
T1	{a,c,d}	{1,1,1}
T2	{a,c,e,g}	{2,6,2,5}
T3	{a,b,c,d,e,f}	{1,2,1,6,1,5}
T4	{b,c,d,e}	{4,3,3,1}
T5	{b,c,e,g}	{2,2,1,2}

TABLE 2: Utility Table

Item	a	b	c	d	e	f	g
Profit	5	2	1	2	3	1	1

TABLE 3: Transaction Utility Table

TID	TU
T1	8
T2	27
T3	30
T4	20
T5	11

TABLE 4: Transaction Weighted Utility Table

ITEM	TWU
A	65
B	61
C	96
D	58
E	88
F	30
G	38

A. Transaction Database

Let I be a set of items. A transaction database is a set of transactions D = {T1,T2,....., Tn} such that for each transaction T ∈I and Tchas a unique identifier c called its Tid. For each item i∈I is associated to a positive number eu(i), called its external utility. For each transactionTc, Tsuch that i∈Tc, a positive number iu(i,Tc) is called the internal utility of i (e.g. purchase quantity).

B. Absolute utility of an item

Utility of an item ib in a transaction tv is denoted as U(ib, tv) and defined as
U(ib, tv)= Q(ij, tv) * P(ij, I)

C. Absolute utility of an itemset in a transaction

Utility of itemset c in transaction tv is denoted as U(c, tv) and defined as followsU(X, T) =∑i∈X∧X⊆T U(i, T).
For example, in TABLE 1, U({a, e}, T3) = U(a, T3) + U(e, T3)= 4 × 1 + 1 × 4 = 8, and U({a, e}) = U({a, e}, T3) + U({a, e},T2) = 8 + 13 = 21.

D. Transaction utility and total utility:

The transaction utility (TU) of a transaction Tr is defined as TU (Tr) = U(Tr, Tr).
The total utility of a database D is denoted as TU(D) and defined as ∑Tr∈D TU(Tr).

E. Transaction-weighted utilization:

The transaction weighted utility of itemset X in DB, denoted as TWU(X), where
TWU(X) = ∑T∈DB∧X⊆T TU(T).

F. Remaining utility of an itemset in a transaction:

The remaining utility of itemset X in transaction T, denoted as RU(X, T), is the sum of the utilities Σ of all the items in T/X in T, where RU(X, T) = ∑i∈(T=X) u(i, T).

G. Z-element

An element is called Z-element if and only if its remaining utility value (RU) is equal to zero. Otherwise, the element is called NZ- element. The set of all Z-elements in the utility list of X is denoted as ZE(X).

H. High Utility Itemset

An item set X is called high utility item set (HUI) if U(X) is no less than a user-specified minimum utility threshold min-util otherwise X is a low utility item set.

I. Top-k high utility item set

An item set X is called top-k high utility itemset in D iff there are less than k itemsets whose utilities are larger than EU(X) in fHUI(D, 0).

J. Optimal minimum utility threshold

An absolute minimum utility threshold d is called optimal minimum utility threshold iff there does not exist a threshold.

IV. SYSTEM OVERVIEW AND METHODOLOGY

The problems with previous HUI miner algorithms are: It generates a huge set of HUIs. These huge number of HUIs forms a challenging problem to the mining performance and higher processing time it consumes. It also generates huge number of candidate itemsets, then higher processing time it consumes. and setting an appropriate threshold value to mine HUIs is difficult task to user. The burden of threshold setting arises a new area called top-k mining. In top-k no threshold is set externally. It is set internally by the algorithm itself. Initially zero is set to minimum utility threshold. The threshold is set automatically as the work progresses. There are different methods for raising the threshold. Using k instead of minimum utility threshold helps in finding only top-k sets of products that contribute highest profits to the company. In this system using TKO algorithm to mine High Utility Itemsets without using minimum threshold value in one phase from large transactional databases. In the previous algorithm analysis we study the high utility algorithms are divided into one phase and two phase algorithms i.e. tree based and list based data structure algorithms.

A. Data Structure

The TKO algorithm uses utility-list data structure to store the utility information about a database. The utility lists of each itemset is called as initial utility-lists, which are constructed by scanning the database twice.

The utility-list of an item(set) X consists of one or more tuples. Each tuple represents the information of X in a transaction Tr and has three fields: <Tid, iutil, rutil>. Fields Tid and iutil respectively contains the identifier of Tr and the utility of X in Tr. Field rutil indicates the remaining utility of X in Tr.

B. TKO Algorithm

In this subsection, we study an efficient algorithm for discovering top-k high-utility itemsets from a large transactional database(See Algorithm 1). The algorithm takes as input a transactional database and a parameter k. The algorithm maintains a min heap structure top-k-cl-list sorted of size K dynamically, which contains the top-k high-utility itemsets. The minimum utility threshold (min utility) stores the utility of the current kth itemset. The minimum utility threshold is initialized to zero before the start of mining process as the top-k list is empty. First, the database scanned twice to build initial utility lists ULs. In the first database scan Transaction-Weighted Utilities of all itemsets are accumulated. If the Transaction-Weighted Utilities of all item is less than given minimum utility threshold value, then the item is no longer considered according in the subsequent mining process. For the items whose Transaction-Weighted Utilities exceeds the minimum utility, they are stored in Transaction-Weighted Utility ascending order. In second database scan constructs the initial utility lists. Fig. 1 shows the initial utility lists of above transactional database.According to ascending order of TWU values. After that, If the initial utility(U) of an itemset is greater than the minimum threshold, the algorithm explores the search space of high utility episodes with the current episode as the prefix. The algorithm follows a depth first search strategy for finding new HUIs. To explore the search space, an itemset is {xy} can be constructed by the intersection of the utility list of {x} and that of {y}. The algorithm identifies common transactions by comparing the Tids in the two utility-lists. Suppose the lengths of the utility-lists are p and q respectively, and then (p + q) comparisons at most are enough for identifying common transactions. The identification process is actually a 2-way comparison.

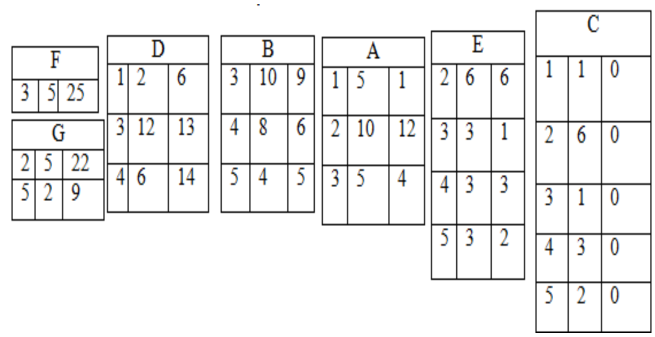


Fig1. Initial utility lists

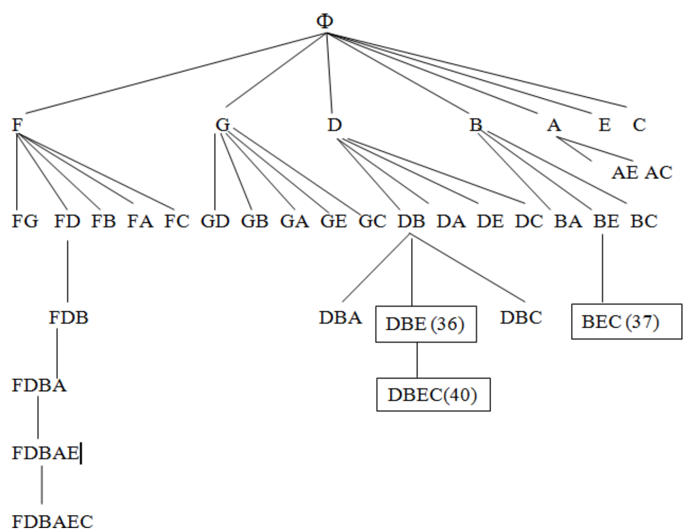


Fig.2. Search space to find top k HUIs

Each new item set, formed by concatenation, if U(itemset) is greater than minimum utility threshold value then it calls RUC procedure which updates the list of Top-k itemsets(See Algorithm 2).The input itemset is added to the list if the size of the list is less than user defined k or its utility is no less than the minimum utility threshold. Once k candidates are discovered the minimum utility is raised to kth highest utility in the list i.e. to the least utility in the list. Further, only those itemsets satisfying minimum utility is inserted in the list and the (k+1)th episode is removed from the buffer. After the algorithm terminates, top-K list contains the desired output of top-k high utility itemset mining in the large transactional database.

C. Algorithm 1: TKO (D, K) Algorithm

1) *Input:* D: a transaction database**,** K: number of HUIS
2) *Output:* The set of Top k high-utility itemsets

// step 1: construct initial utility list for candidate 1 itemsets.

a) Scan database D to calculate the TWU of single items.
b) Initially δ: Min_util set to 0.
c) I* ← each item i such that TWU(i) ≥δ.
d) Let be the total order of TWU ascending values on I*.
e) Scan database D to build the utility-list of each item i ∈ I*.

//step 2: explores search space by calling search procedure.

f) Search (∅, I*, null, min_util, k, TopK-CL-List);

i) For each X = {x1,x2,……, xL}∈Class[P] do
ii) If (SUM(X.iutils) >= δ) then
iii) Raise min_util Border by the strategy RUC
iv) δ ← RUC(X, k, TopK-CI-List)
v) End if
vi) If (NZEU(X) + RU(X) >= δ) then
vii) Initialize Class[X] and ULS[X] to NULL.
viii) For each Y = {y1, y2,…., yL }∈Class[P] yL>xLdo
ix) Concatenate X and Y then store in Z Z ←X∪Y

//step 3: construct candidate L+1itemsets by construct procedure

x) Construct utility list of Z by calling construct method ul(Z)←Construct(ul(P), X, Y, ULS[P])
xi) Class[X] ← Class[X] ∪ Z
xii) ULS[X]← ULS[X] ∪ul(Z) revised utility list after construct of z itemset for further search process.
xiii) End for
xiv) Recursive call Top K-HUI-Search(X, ULS[X], Class[X],δ, Top K-CI-List)
xv) End if
xvi) End for

D. Algorithm 2: RUC Algorithm

1) *Input:*X an itemset, TopK-CL-List: list that contains Top K HUIs, K: number of HUIS
2) *Outpu*t: δ, optimal threshold value.

i) if size(TopK-CL-List)<k then
ii) Add X item (which utility > min_util Border) to Topk-CL-List.
iii) Else
iv) if utility(X) > min utility then
v) Remove kth high utility itemset i.e. itemset(s) having least utility.
vi) Add itemset to the List.
vii) Sort the list in decreasing order of utility values of itemsets.
viii) min utility = least utility in TopK-CL-List.
ix) End if
x) End if

The Top K itemsets for the above transactional database in Table 1 where k=3 are, {<dbec>: 40, <bec>: 37, <dbe>:36}. The search space was shown in Fig. 2.

E. Threshold raising Strategies

The TKO algorithm generates many candidates since the minimum threshold start from zero. Algorithm raises the minimum threshold before mining high-utility itemsets from a large transactional database by bellow specified strategies.

1) RUC (Raising threshold by Utility of Candidates)
2) RUZ (Reducing estimated utility values by using Z elements)
3) EPB (Exploring the most Promising Branches first)

V. EXPERIMENTAL RESULTS

A. Experiments

To evaluate the performance of TKO Algorithm, we have done extensive experiments on various databases, in which TKO algorithm is compared with the state-of-the-art mining algorithms UP-growth and HUI-Miner with optimal thresholds.

B. Experimental setup

For experiments we use both real and synthetic datasets. Foodmart is a real-life sparse dataset from a retail store, with real utility values. which is acquired from Microsoft Food-mart 2000 database and mushroom dataset is type of dense dataset which is obtained from the FIMI repository. The databases do not provide item utility (external utility) and item count for each transaction (internal utility). Here internal utilities for items are generated randomly ranging from 1 to 10 by transaction utility values Generation code. Table 5 shows the statistical information about these databases, the number of transactions, the number of distinct items, the average number of items in a transaction, and the maximal number of items in the longest transaction(s).

Table 5. characteristics of datasets

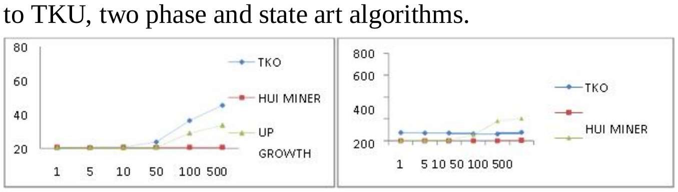
Dataset	Transactions	Avg. Length Transactions	No. Of Items	Type
Foodmart	4141	4.4	1559	Sparse
Mushroom	8124	23	119	Dense
DB_Utility	2880	4	10	Synthetic

C. Time Comparison

The running time of algorithm on foodmart and mushroom datasets with varied K respectively. We compare these execution times with UP Growth, HUI Miner algorithms with optimal thresholds in Fig.3 From the above experiments we observe that TKO algorithm was executes well and take less than second when the K value is low. If the k value increases the execution time also increases for sparse dataset such as foodmart. For dense dataset the algorithm takes almost equal time to execute for various K values and also higher than the hui miner because dense datasets contains large size transactions. Running time was recorded by the "Timestamp" command, and it contains starting time, and ending time of the algorithm. For almost all databases, proposed algorithm performs the best and faster than the existing algorithm.

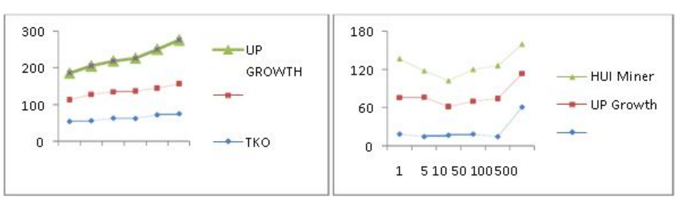
D. Memory Comparison

Fig. 4 shows memory usage of algorithms. TKO uses list data structure to maintain the utility information and high utility itemsets and also it is one phase algorithm. For that reason TKO takes less memory compare to up-growth and little bit higher or equivalent to the HUI Miner because we maintain and update extra Topk-cl-list for top k high utility itemsets. TKO algorithm generally uses less memory compare to TKU, two phase and state art algorithms.



(A)foodmart(b) mushroom

Fig. 3. Runtime of TKO, HUI Miner, UP Growth



(a) foodmart (b) Mushroom

Fig4 Runtime of TKO, HUI Miner, UP Growth

VI. CONCLUSION

In this we are presenting a literature survey on various algorithms used for mining high utility itemsets. we studied the difficulty of setting threshold value in high utility mining. To address this problem the authors has proposed a novel representation of high-utility itemsets named Top k High-Utility Itemsets (Top k HUI). In our framework the top k high utility itemsets are find out by using TKO algorithm which was faster than all top k HUI algorithms. With our framework we find out several k value high utility itemsets on both real and synthetic datasets efficiently. The algorithm was implemented for different types of datasets. Performance factors like time, memory space of TKO algorithm are compared with HUI Miner and UP growth algorithms.

REFERENCES

[1] Vincent S. Tseng, Cheng-wei Wu, Philippe Fournier-Viger and Philip S. Yu, "Efficient algorithms for mining Top-K high utility itemsets", in IEEE Transactions on Knowledge and data engineering, vol.28 no.1 January 2016.

[2] Serin Lee, Jong Soo Park, "Top-K high utility itemset mining based on Utility List Structures", In Proceedings of IEEE International Conference on Data Mining(ICDM), Maebashi, pp. 101 - 108, 2016.

[3] Shuning Xing, Fangai Liu, Jiwei Wang, Lin Pang, ZhenguoXu, "Utility Pattern Mining Algorithm bases on Improved Utility Pattern Tree", in 8thbinternational symposium on computational intelligence and design, pp.258 – 261,2015.

[4] Junqiang Liu, Benjamin C.M. Fung "Mining High Utility Patterns in One Phase without Generating Candidates" in ," IEEE Transaction Knowledge in Data Engineering, vol. 10, no. 12, pp. 1-14, Dec.2015.

[5] R. Chan, Q. Yang, and Y. Shen, "Mining high-utility itemsets," in Proc. IEEE Int. Conf. Data Mining, 2003, pp. 19–26.

[6] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in Proc. ACM Int. Conf. Inf. Knowl. Manag., 2012, pp. 55–64.

[7] H. Ryang and U. Yun, "Top-k high utility pattern mining with effective threshold raising Strategies," Knowl.-Based Syst., vol. 76, pp. 109–126, 2015.

[8] C. Ahmed, S. Tanbeer, B. Jeong, and Y. Lee, "Efficient tree structures for high-utility pattern mining in incremental databases," IEEE Trans. Knowl. Data Eng., vol. 21, no. 12, pp. 1708–1721, Dec. 2009.

[9] V. S. Tseng, C. Wu, B. Shie, and P. S. Yu, "UP-Growth: An efficient algorithm for high utility itemset mining," in

Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010, pp. 253–262.

[10] Frequent Itemset Mining Implementations Repository [Online]. Available: http://fimi.cs.helsinki.fi/

[11] FoodMart2000, Microsoft Developer Network (MSDN) [Online]. Available: https://technet.microsoft.com/en-us/library/aa217032 (v=sql.80).aspx

[12] Shoban Babu Sriramoju, "Analysis and Comparison of Anonymous Techniques for Privacy Preserving in Big Data" in "International Journal of Advanced Research in Computer and Communication Engineering", Vol 6, Issue 12, December 2017, DOI 10.17148/IJARCCE.2017.61212 [ISSN(online) : 2278-1021, ISSN(print) : 2319-5940]

[13] Shoban Babu Sriramoju, " Review on Big Data and Mining Algorithm" in "International Journal for Research in Applied Science and Engineering Technology", Volume-5, Issue-XI, November 2017, 1238-1243 [ISSN : 2321-9653], www.ijraset.com

[14] Shoban Babu Sriramoju, "Heat Diffusion Based Search for Experts on World Wide Web" in "International Journal of Science and Research", https://www.ijsr.net/archive/v6i11/v6i11.php, Volume 6, Issue 11, November 2017, 632 - 635, #ijsrnet

[15] Dr. Shoban Babu Sriramoju, Prof. Mangesh Ingle, Prof. Ashish Mahalle "Trust and Iterative Filtering Approaches for Secure Data Collection in Wireless Sensor Networks" in "International Journal of Research in Science and Engineering" Vol 3, Issue 4, July-August 2017 [ISSN : 2394-8299].

[16] Dr. Shoban Babu, Prof. Mangesh Ingle, Prof. Ashish Mahalle, "HLA Based solution for Packet Loss Detection in Mobile Ad Hoc Networks" in "International Journal of Research in Science and Engineering" Vol 3, Issue 4,July-August 2017 [ISSN : 2394-8299].

[17] Shoban Babu Sriramoju, "A Framework for Keyword Based Query and Response System for Web Based Expert Search" in "International Journal of Science and Research" Index Copernicus Value(2015):78.96 [ISSN : 2319-7064].

[18] Sriramoju Ajay Babu, Dr. S. Shoban Babu, "Improving Quality of Content Based Image Retrieval with Graph Based Ranking" in "International Journal of Research and Applications" Vol 1, Issue 1,Jan-Mar 2014 [ISSN : 2349-0020].

[19] Dr. Shoban Babu Sriramoju, Ramesh Gadde, "A Ranking Model Framework for Multiple Vertical Search Domains" in "International Journal of Research and Applications" Vol 1, Issue 1,Jan-Mar 2014 [ISSN : 2349-0020].

[20] Mounika Reddy, Avula Deepak, Ekkati Kalyani Dharavath, Kranthi Gande, Shoban Sriramoju, "Risk-Aware Response Answer for Mitigating Painter Routing Attacks" in "International Journal of Information Technology and Management" Vol VI, Issue I, Feb 2014 [ISSN : 2249-4510]

[21] Mounica Doosetty, Keerthi Kodakandla, Ashok R, Shoban Babu Sriramoju, "Extensive Secure Cloud Storage System Supporting Privacy-Preserving Public Auditing" in "International Journal of Information Technology and Management" Vol VI, Issue I, Feb 2012 [ISSN : 2249-4510]

[22] Shoban Babu Sriramoju, "An Application for Annotating Web Search Results" in "International Journal of Innovative Research in Computer and Communication Engineering" Vol 2,Issue 3,March 2014 [ISSN(online) : 2320-9801, ISSN(print) : 2320-9798]

[23] Shoban Babu Sriramoju, "Multi View Point Measure for Achieving Highest Intra-Cluster Similarity" in "International Journal of Innovative Research in Computer and Communication Engineering" Vol 2,Issue 3,March 2014 [ISSN(online) : 2320-9801, ISSN(print) : 2320-9798]

[24] Shoban Babu Sriramoju, Madan Kumar Chandran, "UP-Growth Algorithms for Knowledge Discovery from Transactional Databases" in "International Journal of Advanced Research in Computer Science and Software Engineering", Vol 4, Issue 2, February 2014 [ISSN : 2277 128X]

[25] Shoban Babu Sriramoju, Azmera Chandu Naik, N.Samba Siva Rao, "Predicting The Misusability Of Data From Malicious Insiders" in "International Journal of Computer Engineering and Applications" Vol V,Issue II,Febrauary 2014 [ISSN : 2321-3469]

[26] Ajay Babu Sriramoju, Dr. S. Shoban Babu, "Analysis on Image Compression Using Bit-Plane Separation Method" in "International Journal of Information Technology and Management", Vol VII, Issue X, November 2014 [ISSN : 2249-4510]

[27] Shoban Babu Sriramoju, "Mining Big Sources Using Efficient Data Mining Algorithms" in "International Journal of Innovative Research in Computer and Communication Engineering" Vol 2, Issue 1,January 2014 [ISSN(online) : 2320-9801, ISSN(print) : 2320-9798]

[28] Ajay Babu Sriramoju, Dr. S. Shoban Babu, "Study of Multiplexing Space and Focal Surfaces and Automultiscopic Displays for Image Processing" in "International Journal of Information Technology and Management"Vol V, Issue I, August 2013 [ISSN : 2249-4510]

[29] Dr. Shoban Babu Sriramoju, "A Review on Processing Big Data" in "International Journal of Innovative Research in Computer and Communication Engineering"

Vol-2, Issue-1, January 2014 [ISSN(online) : 2320-9801, ISSN(print) : 2320-9798]

[30] Shoban Babu Sriramoju, Dr. Atul Kumar, “An Analysis around the study of Distributed Data Mining Method in the Grid Environment : Technique, Algorithms and Services” in “Journal of Advances in Science and Technology” Vol-IV, Issue No-VII, November 2012 [ISSN : 2230-9659]

[31] Shoban Babu Sriramoju, Dr. Atul Kumar, “An Analysis on Effective, Precise and Privacy Preserving Data Mining Association Rules with Partitioning on Distributed Databases” in “International Journal of Information Technology and management” Vol-III, Issue-I, August 2012 [ISSN : 2249-4510]

[32] Shoban Babu Sriramoju, Dr. Atul Kumar, “A Competent Strategy Regarding Relationship of Rule Mining on Distributed Database Algorithm” in “Journal of Advances in Science and Technology” Vol-II, Issue No-II, November 2011 [ISSN : 2230-9659]

[33] Shoban Babu Sriramoju, Dr. Atul Kumar, “Allocated Greater Order Organization of Rule Mining utilizing Information Produced Through Textual facts” in “International Journal of Information Technology and management” Vol-I, Issue-I, August 2011 [ISSN : 2249-4510]

[34] Ramesh Gadde, Namavaram Vijay, “A SURVEY ON EVOLUTION OF BIG DATA WITH HADOOP” in “International Journal of Research in Science and Engineering”, Vol-3, Issue-6, Nov-Dec 2017, 92-99 [ISSN : 2394-8299].

[35] Namavaram Vijay, S Ajay Babu, "Heat Exposure of Big Data Analytics in a Workflow Framework" in “International Journal of Science and Research”, Volume 6, Issue 11, November 2017, 1578 - 1585, #ijsrnet

[36] Ajay Babu Sriramoju, Namavaram Vijay, Ramesh Gadde, “SKETCHING-BASED HIGH-PERFORMANCE BIG DATA PROCESSING ACCELERATOR” in “International Journal of Research in Science and Engineering”, Vol-3, Issue-6, Nov-Dec 2017, 92-99 [ISSN : 2394-8299].

[37] Namavaram Vijay, Ajay Babu Sriramoju, Ramesh Gadde, “Two Layered Privacy Architecture for Big Data Framework” in “International Journal of Innovative Research in Computer and Communication Engineering” Vol 5, Issue 10, October 2017 [ISSN(online) : 2320-9801, ISSN(print) : 2320-9798]

[38] Amitha Supriya. "Implementation of Image Processing System using Big Data in the Cloud Environment." International Journal for Scientific Research and Development 5.10 (2017): 211-217.

[39] SA Supriya. "A Survey Model of Big Data by Focusing on the Atmospheric Data Analysis." International Journal for Scientific Research and Development 5.10 (2017): 463-466.

[40] Guguloth Vijaya, A. Devaki, Dr. Shoban Babu Sriramoju, “A Framework for Solving Identity Disclosure Problem in Collaborative Data Publishing” in “International Journal of Research and Applications” (Apr-Jun © 2015 Transactions), Vol 2, Issue 6, 292-295

[41] Siripuri Kiran, 'Decision Tree Analysis Tool with the Design Approach of Probability Density Function towards Uncertain Data Classification', International Journal of Scientific Research in Science and Technology(IJSRST), Print ISSN : 2395-6011, Online ISSN : 2395-602X,Volume 4 Issue 2, pp.829-831, January-February 2018. URL : http://ijsrst.com/IJSRST1841198

[42] Ajmera Rajesh, Siripuri Kiran, " Anomaly Detection Using Data Mining Techniques in Social Networking" in “International Journal for Research in Applied Science and Engineering Technology”, Volume-6, Issue-II, February 2018, 1268-1272 [ISSN : 2321-9653], www.ijraset.com

[43] Shoban Babu Sriramoju, “OPPORTUNITIES AND SECURITY IMPLICATIONS OF BIG DATA MINING” in International Journal of Research in Science and Engineering”, Vol 3, Issue 6, Nov-Dec 2017 [ISSN : 2394-8299].