Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 8 (2017) pp. 2381-2394

http://www.ripublication.com

Fault Tolerant Frequent patterns mining in large datasets having certain and uncertain records

S M Arif Ashraf
Institute for System Studies & Analyses, Delhi-110054

Md. Tabrez Nafis
Assistant Professor,Department of Computer Science & Engineering,
Jamia Hamdard, New Delhi-110062

Abstract

Finding frequent patterns has been a focused area in the knowledge discovery process. Large numbers of algorithms have been proposed for frequent patterns mining. However, most of the frequent patterns mining approaches have been centered on the certain or deterministic datasets, where records are either present or absent. Because of the large areas, where uncertain datasets are in use, various algorithms for finding frequent patterns in uncertain datasets have also been developed. In uncertain or non-deterministic or probabilistic datasets, an existential probability is attached with every record. Existential probabilities indicate the likelihood of records appearance in the datasets. Furthermore, in areas such as geo spatial, remote image, object identification , weather forecasting, IoT enabled applications, used or generated datasets contain both certain and uncertain records. Mining frequent patterns over such datasets which have both certain and uncertain records require a different approach altogether.

In this work, we propose a Fault Tolerant Frequent patterns mining for large datasets that contain both certain and uncertain records. Fault-tolerant mining is relatively new and fruitful direction of research among data mining community. In fault tolerant mining, instead of exact matching based on support and confidence value, an approximate matching is done to find frequent patterns. This approximate matching on the basis of fault gives a more insight of interesting information even if the datasets have some errors or missing information or changed information which could not be possible otherwise with exact matching.

Experimental results with synthesized datasets show that our algorithm is fairly efficient for performing fault tolerant frequent patterns mining.

Keywords: Frequent itemsets mining, fault tolerance, massive datasets, certain records, and uncertain records.

1. INTRODUCTION

Mining of frequent pattern and association rule(s) have become a relatively important area of data mining after it was first studied [1][2]. Earlier algorithms used certain datasets for frequent pattern mining that is, items are either present or not. However, with exploration of new applications such as sensors applications, geospatial applications, moving objects search, network analysis [20] among others, uncertain datasets relatively received great attention.

Unlike the certain datasets, in uncertain datasets, an existential probability is attached with every item. This existential probability represents the likelihood of the presence of item in datasets. Since frequent pattern mining is one of the fundamental areas in data mining, mining frequent pattern over uncertain database soon become a hot pursue. A major challenge with uncertain datasets mining as compare to certain or precise data mining is its greater solution space because of the probabilistic nature of datasets. For uncertain data mining, answer is never guarantee rather a suspicion only. For example a medical practitioner is 80% certain about diseases although an element of 20% uncertainty is also here. [3] Provides comprehensive survey of techniques for uncertain data mining.

Since real life applications data are dirty and varying in nature, a fault tolerant extension of pattern mining techniques was proposed [17] to provide a great and much useful insight of datasets. In fault tolerance mining, we relax the constraint of exact matching up to some tolerable limit that will not have critical impact on final result for data analysis purpose. For example in case of certain datasets, let’s say that there is a rule that if a student is good in all four subject (AI, Statistics, Algorithm and Programming), then he will also be good in data mining as well, for knowledge discovery. A fault tolerant version of this rule will be that if a student is good in 3 out of 4 subject (a fault of 25%), then he will also be good at data mining too. Unlike the first rule, the second rule is more general and will provide more fruitful insight of datasets. Similarly for uncertain data, if a patient has high probability of 3 out of 4 symptoms (Polypepsia, Polyuria, Calf Muscles Pain,Polyphagia), then he will have a high probability of diabetics as well. Therefore unlike the stringent discovery rule, the fault tolerant discovery rule is more general, more helpful and will have more result orientation in pattern matching.

Performing fault tolerant frequent patterns mining over datasets that have a mixture of both certain and uncertain items will require a different approach altogether since both probabilistic and binary natured items will be considered in parallel for frequent pattern mining.

In this paper we propose an approach for performing fault tolerant frequent pattern mining over the datasets that are mixture of certain and uncertain records.

The rest of the paper is organized as follows. In section 2, a set of definitions related to the work is provided. In section 3, we reviewed related work. Section 4 provides a discussion of our approach and the formal presentation of algorithm. Section 5 provides Experimental results and a discussion over it. Finally we conclude in section 6 with future work direction.

2. GENERAL DEFINITION

We provide, in this section, various general definitions related to fault tolerant frequent pattern mining over a mixture of certain and uncertain records.

For our work clarity and experimental purpose, we used transactional datasets for frequent pattern mining.

Let $I = \{i_1, i_2, i_3, i_4 \ldots\ldots i_m\}$ be a set of distinct m items. Let X is a set of items derived from the set *I.* If X has k items then it will be considered as k-Itemsets. A dataset *MDB* is obtained over the items defined in set *I.* For certain items in datasets records or transactions no probability is attached whereas for uncertain items an existential p ($0 \leq p \leq 1$) is attached. Higher the probability, higher the chances of its appearance in the transactional records. Therefore each record/transaction in *MDB* is represented as <tid,T> where tid is the transaction ID and T={t1(*p1*),t2(*p2*),t3(*p3*)……..tn(*pn*)}.

Given the *MDB,*

Definition 1: Given an N transactions dataset with a minimum support (min_sup) and probabilistic frequent threshold (pft), an itemset X is probabilistic frequent itemset, if X contains only those items for which following two condition match.

i). appearance of items in complete datasets is equal and/or greater than min_sup.

ii) probability of items is equal and/or greater than pft.

Both min_sup and pft are users defined.

Definition 2: (Fault): A fault is no of the items from the total items that can be ignored during frequent mining. For example, for a set of 5 items (a,b,c,d,e,f), if we say that any transaction that have 3 out of 5 items with any possible combination, then we have a fault of 2. Fault is nothing but an approximation, where we approximate the no of items in a transaction with the help of fault.

Definition 3: (Fault – tolerant frequent pattern): Given a user defined fault tolerance ∂ ($\partial > 0$).Let P be an itemset such that $|P| > \partial$. A transaction T= (tid, X) is said to be **FT-contain** itemset P if and only if there exists P' subset of X and $|P'| \geq (|P| - \partial)$. The number of transactions in a database that contain P is called the **FT-support** of P, denoted as sup_FT (P).

If the sup_FT(P) is equal and/or greater than user supplied support than it will be fault tolerant frequent pattern.

Definition 4: (Fault – tolerant frequent pattern for certain and uncertain records):

An itemset is fault tolerant frequent itemset if the no of itemsets is equal or greater than the minimum support and threshold probability with any combination of items of at least total unique items count minus specified fault tolerance that is ∂

3. EARLIER WORK

We reviewed earlier work under three categories which are:

i). Works related to frequent pattern mining over certain datasets.

ii). Works related to frequent pattern mining over uncertain datasets.

iii). Works related to fault tolerance data mining.

3.1. Work related to frequent pattern mining over certain dataset

In this section, a review of works related to frequent pattern mining over certain or deterministic or precise dataset is provided.

Earlier work [1][2] known as Apriori Algorithm ,were the most simplest way to generate frequent pattern. An itemset is frequent if its appearance in the datasets is equal to/more than the user specified support value. Apriori is a recursive, brute force methodology for pattern generation. The key concept is that an itemset cannot be frequent if it's all subsets are not frequent as well. The approach is primarily generate and test approach where in each iteration candidate for frequent items are generated (based upon support value) and then tested against the subsets constraints to prune the candidates. The process continues until there are no candidates to generate. The prime disadvantage of this approach is that it generates large number of candidate and makes multiple scan to the database. The problem becomes more severe when the datasets size is too large to fit into main memory.

[4][5] provided FP Tree based approaches that eliminate the need of candidate generation and multiple scan to the database. This is a partitioning based, divide and Conquer method unlike the bottom up method used in Apriori. The algorithm does not subscribe to generate and test paradigm of Apriori. Instead, it encodes the data set using a compact data structure called an FP-tree and extract frequent itemsets directly from the structure. FP achieved efficiency by three ways[4] :(a) by significantly reducing the cost involved in repeated scan of database by compressing the database into a highly condensed data structure, a tree (b) by avoiding the generation of large candidates (c) be reducing the search space. H-mine [6] is another pattern growth, divide and conquer based frequent pattern mining algorithm. It uses a hyper linked data structure called H-struct. The algorithm is highly efficient in finding the short as well as long pattern.

Incremental/decrement mining was also discussed and various solutions were proposed .In incremental/decrement mining, instead of replying the frequent pattern finding algorithm every time whenever database increases or decreases, the algorithm is only applied in increased part of database (incremental mining) or decreased part (decrement mining)[26].

3.2. Work related to frequent pattern mining over uncertain dataset

With the advent of many new applications [7], [8], [9] ,by the increase use of sensors devices in network analysis, analysis of non-constants objects, increased use of fuzzy databases, the uncertain frequent pattern mining has become relatively important are in data mining. In some applications such as IoT, wireless sensing devices are deployed heavily. However with the inherent uncertainty in the devices used for information capturing, data often are inaccurate.

Therefore, in this section, a review of works related to frequent pattern mining over uncertain or nondeterministic or imprecise dataset is provided

The problem of frequent itemsets mining in uncertain dataset was investigated [10] and authors proposed U-Apriori algorithm. The proposed algorithm was essentially a modified Apriori algorithm for uncertain data. Authors proposed *Expected Support* notion to obtain frequent itemsets.To reduce the number of possible candidates, the authors also proposed a trimming strategy. The notion of probabilistic support for frequent itemsets was introduced in [11].The probabilistic support is an exact measure of an itemset support in the possible world model.

Approximation based approaches were discussed in [12][13]. The idea behind [13] is to use Poisson law to approximate the distribution of support whereas central limit theorem was used for approximation in [12].

In [14] an approach for performing decremental mining over uncertain data was provided. A study on closed frequent itemsets mining over probabilistic data was made in [15].

A tree bases approach for uncertain mining was made in [16].In this, authors proposed a tree structure *UF-tree* for representing the uncertain dataset and *UF-growth* algorithm to obtain frequent patterns from the *UF-tree.*

In [27], authors proposed an algorithm (SCTA), a clustering technique for large uncertain as well as unstructured datasets. The authors specifically emphasized on high dimensionality of uncertain datasets.

3.3 Work related to fault tolerance pattern mining

In real world application, the underlying dataset can be dirty and diverse. The dataset may missed some actually present value during the stage of capturing ,storing, processing or even in transmission from one geographical location to another. Moreover there are usually few patterns with high support and high confidence. High support and high confidence though have the high prediction accuracy, often miss some interesting pattern. Thus instead of finding pattern on the basis of exact matching over the datasets, an approximate matching method or fault tolerant approach can effectively be used .In this section ,we will review works related to fault tolerant mining.

In [17], authors gave a general definition of frequent itemsets where errors in databases were included in definition. The authors also proposed an algorithm that can find out error-tolerant clusters of items over variety of datasets including transactional datasets, datasets related to pages browsing over webs, unstructured datasets like texts over web among others. In [21], a depth-first approach was to generate candidates for frequent pattern. In [21], a vector based approach was discussed. Here two vectors were defined. First vector called appearing vector was used to store, in which transaction, an item was appeared. A second vector known as Fault appearing vector was defined to store the items with regards to some given fault tolerance value or limit. Frequent patterns were then obtained by performing vectors operation on those defind vectors. To find only long interesting fault tolerant frequent pattern, [22] proposed a depth first search based Max-FTP (Maximal Fault-Tolerant Frequent Pattern Mining) algorithm. Based on FP tree concept for frequent itemsets mining,[23] used FP tree in fault tolerant. The first study on large database for fault tolerant mining was undertook in [24].

However all these study were made on certain databases. No study was found on uncertain databases. To fill this gape current work is undertaken.

4. PROPOSED SOLUTION

In this section we will present our approach and a formal presentation of the algorithm will be provided.

It has been found that most of the frequent mining algorithm over dataset that have the existential probability, used three approaches (i) Expected support based frequent algorithms[10][14][18][19][20] (ii) Exact probabilistic frequent algorithms[20] and (iii) Approximate probabilistic frequent algorithms[20].

All three approaches scan the entire dataset to get their relative occurrence probability in the dataset. This scanning some time may involve great performance cost if the dataset is massive. This calculated relative probability then used as filtering out infrequent pattern. Instead, in our approach, we will directly compare the given support probability with the existential probability associated with each item. This approach will have one more advantage. In case of distributed environment, where part of massive dataset are processed on different nodes, it will effectively handle the problem of false positive (an itemset is frequent in part of the dataset but not in complete dataset) and false negative (an itemset is infrequent in part of the dataset but frequent in complete dataset).

Given a dataset *MDB,* let min_sup is minimum support, min_sup_prob is the minimum probability of item (that is all those items that have support of 60% of total transaction and have the probability greater 0.6 in each transaction), FT_prob_min_sup (probabilty of each item in the Fault tolerant Item set X), FT_min_sup is minimum fault tolerant support(for example 80% of total transaction), ∂ is fault tolerance, min_length is minimum fault tolerant itemset length and X is fault tolerant frequent pattern.

The min_sup is used to filter out the infrequent itemsets whereas FT_min_sup is used to find out the itemset that have at least support of FT_min_sup. For brevity we assumed that FT_min_sup is more than min_sup to avoid big number of short pattern, however this is not a mandatory condition. Also FT_prob_min_sup is greater than min_sup_prob.

We also assume that $|X|>\partial$ and $|X| \geq$ min_length that is length of items in a frequent pattens must be greater than the fault tolerant limit and greater than or equal to minimum length for frequent pattern. These two constraints are basically used as check on the number of patterns that are generated.

The steps to find out the Fault Tolerant itemset X are as follows:

a. Scan the dataset and find those items that have at least min_sup support and min_sup_prob probability. The set of items will constitute the global frequent items.

b. Since the minimum itemset length will be greater than ∂, now find the itemset of length 2(assuming that ∂ is 1).This time consider only those items that have the at least probability specified by min_sup_FT_prob and the support as specified by min_sup_FT. If the $|X| \geq$ min_length then out put the X as frequent itemsets.

c. Again find the itemset of length 3 with the minium support of FT_min_sup and item probability of FT_prob_min_sup. If $|X| \geq$ min_length then out put the X as frequent itemsets otherwise proceed.

d. Repeat the step until no item set of length $\geq$ min_length can be generated or no itemset can be formed from the previous itemset that is the no candidate is left.

Now we will present our algorithm in more formal way

Algorith:**LFTFM (Large Fault Tolerant Frequent Mining)**

Input:

MDB	: Transaction dataset
min_sup	: Global Frequent Support
FT_min_sup	: Support for Fault tolerant frequent itemset
min_sup_prob	: Probability for Global frequent itemset
FT_prob_min_sup	: Probability for Fault tolerant frequent itemset
∂	: Fault tolerance (greater than 0 and always a real number)
min_length	: Minimum length of Fault tolerant itemset

Output:

A complete set of fault tolerant frequent pattern X with minimum length of min_length .

Steps:

1. Scan the database and find those items that have at least support of min_sup and probability min_sup_prob. The itemsets constitute the global frequent itemsets.

2. Let C_i be the set of all length ∂+1 subset from global frequent itemset set. Also i=∂+1.

3. REPEAT THE STEPS

a) Scan the dataset and check itemsets in C_i against the FT_min_sup and

FT_prob_min_sup.

b) Let set F_i be the itemsets generated from C_i..if i$\geq$ min_length then consider

it as frequent pattern.

c) From the set F_i , generate set C_{i+1}

d) i=i+1

Terminate when either F_{i-1} or C_i is empty.

4. Exit

5. EXPERIMENTAL RESULT AND DISCUSSION

In this section we will present our experimental result and also a formal discussion over the finding. All experiments are done on 64 bit machine with 4 GB RAM, i3-2.20 GHz processor, ubuntu 14.4 OS with Hadoop 2.7.2. All programs were written on java.

The result demonstrates following things.

i) Frequent pattern mining and its relationship with fault parameter value ∂.

ii) Frequent pattern mining and its relationship with ∂ and minimum fault tolerant pattern length.

iii) Also the memory requirement and time analysis.

Since most of the available synthesized data generator tools generate certain datasets, therefore we created our own synthesized datasets for the experiment purpose. For all experiments, we primarily used transactional datasets.

Table 1 shows a typical synthesized transactional dataset.

From table 1, a dataset of 6 transactions over a set of 4 items are shown. For clarity in TID 1, the item 1 does not has any probability, therefore it is certain, whereas second item, written as (4,0.5), implies that a probability of .5 is attached with item 4.

Table 1: Typical synthesized transactional dataset.

TID	Transaction
1	(3) , (4,0.5)
2	(1), (2,0.6), (1,0.4) ,(3,0.9), (4)
3	(1,0.9), (3,0.4), (1,0.6)
4	(4) ,(3,0.6), (2,0.0)
5	(1,0.8), (2,0.6), (1,0.3) ,(3,0.9) ,(4,0.6)
6	(1,0.5), (2), (1,0.2), (3,0.9), (4)

We did analysis on three datasets with the following properties.

i) Each dataset has a mixture of certain item (item with no probability) and uncertain (item with probability).

ii) One dataset is of 6 transactions distributed over 4 items; second 50 transactions distributed over 10 items, third with 100 transactions distributed over 10 items and fourth with 200 transactions distributed over 10 items.

Table 2 shows the time and memory requirement for the all four datasets with a min_sup=2.

Table 2: Time and Memory requirements.

S.No	Dataset	Memory Required	Time Required
1	6 Transactions With 4 items	0.62 mb	~ 0 ms
2	50 Transactions With 10 items	0.93 mb	~ 16 ms
3	100 Transactions With 10 items	1.24 mb	~ 78 ms
4	200 Transactions With 10 items	1.55 mb	~ 94 ms

Table 3 shows the no of candidates generated and no of frequent itemsets for a **minimum probabilistic threshold of 0.5 and a min_sup =2**

S.No	Dataset	No of candidate	No. of frequent itemset
1	6 Transactions With 4 items	15 Stopped at size 5 candidate.	11
2	50 Transactions With 10 items	228 Stopped at size 8 candidate.	165
3	100 Transactions With 10 items	332 Stopped at size 8 candidate.	204
4	200 Transactions With 10 items	511 Stopped at size 8 candidate.	285

Table 3: Candidate and Frequent itemsets count.

Table 4 shows the no of Fault tolerant frequent itemsets for a fault of one item that is ∂=1 with minimum **probabilistic threshold of 0.5, min_sup=2,FT_min_sup=2, FT_prob_min_sup=0.6**

Table 4: Fault tolerant Frequent itemsets for a fault tolerance of one item (∂=1)

S.No	Dataset	∂	Fault Tolerant Frequent itemset
1	6 Transactions With 4 items	1	3
2	50 Transactions With 10 items	1	9
3	100 Transactions With 10 items	1	2
4	200 Transactions With 10 items	1	2

For False positive and false negative analysis, we break the input dataset file into two parts and run both the file separately and then find frequent itemset individually. After that, compared the frequent itemsets count for both files with single file frequent itemsets count. We found that results were pretty encouraging.

The overall finding can be briefly summarized as:

1. The proposed approach can effectively handle the both certain and uncertain datasets.
2. The approach performed fault tolerant mining over the synthesized datasets and the results are encouraging enough.
3. The approach can potentially be used in distributed environment as well with the consideration of false positive and false negative problem.

6. CONCLUSION

In this work, presented an algorithm for finding fault tolerant frequent pattern mining in massive dataset that contain both certain and uncertain record. Fault tolerant frequent pattern mining in certain dataset has been studied in limited way, but to best of our knowledge this is the first work where fault tolerant mining is extended to both certain and uncertain composite dataset. The Experimental result shows that our approach is fairly efficient in finding pattern.

We also observed that this approach also handle the problem of false positive and false negative when the frequent pattern mining is to be done over distributed computing environment.

This study can further be extended to other sub area of frequent mining like finding fault tolerant frequent sequence mining, sub sequence mining and sub structure mining among others. Also in the area of IoT, this study can have a potential application and further exploration as well.

REFERENCES

[1] R. Agarwal, T. Imielinski, and A. Swami "Mining association rules between sets of items in large database", Proceeding ACM-SIGMOD International Conference of Management of Data, ACM Press, Washington DC, pp. 207-216,May 1993.

[2] R. Agarwal and R. Srikant "Fast Algorithms for mining association rules " Proceeding of 20th International Conference on Very Large Databases ,Morgan Kaufmann, pp. 487-499,1994.

[3] C. C. Aggarwal. Managing and Mining Uncertain Data, Springer, 2009.

[4] J. Han, J. Pei and Y. Yin "Mining Frequent Patterns without Candidate Generation" The ACM SIGMOD International Conference on Management of Data, pp.1-12 ,2000.

[5] J. Han, J. Pei, Y. Yin and R. Mao "Mining Frequent Patterns without Candidate Generation: A Frequent – Pattern Tree" Data Mining and Knowledge Discovery, pp.53-87, IEEE 2004.

[6] J. Pie, J. Han "H –Mine: Hyper –Structure Mining of Frequent Pattern in Large Databases". ICDM .

[7] M. Li and Y. liu ”Underground coal mine monitoring with wireless sensor networks". TOSN, 5(2):10, 2009.

[8] Y. Liu,K. Liu and M. li ”Passive diagnosis for wireless sensor networks". IEEE/ACM Transaction Network 18(4):1132-1142, 2010.

[9] L. Mo, Y. He , Y. Liu J. Zhao, S. Tnag X. Y. Li and G. Dai”Canopy Closure estimates with green orbs sustainable sensing in the forest". In SenSys,Pages 99-112,2009.

[10] C.K.Chui, B. Kao and E. Hung ”Mining frequent itemsets from uncertain data". In PAKDD, pp. 47-58 2007.

[11] T. Bernecker,H.P. Kriegel, M. Renz,F. Verhein and A. Zuefle ”Probabilistic Frequent itemset mining in uncertain databases". In Proceeding of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 119-128, 2009

[12] T. Calders, C. Garboni and B.Goethals ”Approximation of frequentness probability of itemsets in uncertain data". In Proceeding of the 2010 IEEE International Conference on Data Mining, ICDM pp. 749-754, 2010.

[13] L. Wang, R. Cheng, S. D. Lee and D Cheung ”Accelerating probabilistic frequent itemset mining : a model based approach ". In Proceeding of the 19th ACM International Conference on Information and Knowledge Management CIKM, pp. 429-438, 2010.

[14] C.K. Chui and B. Kao ”A decremental approach for mining frequent itemsets from uncertain data". In PAKDD,pp. 47-58, 2007.

[15] Y.Tong,L. Chen and B.Ding ”Discovering threshold based frequent closed itemsets over probabilistic data". In ICDE,pp. 270-281, 2012.

[16] T. Washio et al.”A tree based approach for frequent pattern mining from uncertain data". PAKDD 2008 ,LNAI 5012,pp. 653-661,2008

[17] C.Yang, U.Fayyad and P.S Bradley "Efficient Discovery of Error-tolerant frequent item sets in High Dimensions". Proceeding of 7th ACM SIGMOD

International Conference on Knowledge Discovery and Data Mining (pp-283-296) 2001.

[18] C.C Aggarwal,Y.Li and J.Wang "Frequent pattern mining with uncertain data". In KDD, (pp-29-38) 2009.

[19] C.K.S. Leung,M.A.F Mateo and D.A. Brajczuk "A tree based approach for frequent pattern mining from uncertain data". In P KDD, (pp-653-661) 2008.

[20] Y. Tong, L.Chen,Y.Cheng and P.S. Yu "Mining frequent itemsets over uncertain databases".

[21] J. L. Koh, P. Yo, "An Efficient Approach for Mining Fault-Tolerant Frequent Patterns based on Bit Vector Representations", in the proceedings of 10th International Conference, DASFAA 2005, Beijing, China, April 17-20, 2005.

[22] S. Bashir, A. Rauf Baig, "Max-FTP: Mining Maximal Fault-Tolerant Frequent Patterns from Databases", In the proceedings of 24 British National Conference on Databases, Springer (LNCS), Volume 4587, July 3-5, Glasgow, UK, 2007.

[23] S. Bashir, A. Rauf Baig, "Mining fault tolerant frequent patterns using pattern growth approach". IEEE/ACS International Conference on Computer Systems and Applications 2008.

[24] S.-S. Wang and S.-Y. Lee"Mining Fault-Tolerant Frequent Patterns in Large Databases". Workshop on Databases and Software Engineering 2005.

[25] T. Chen "An Efficient Algorithm for Proportionally Fault-Tolerant Data Mining". APWeb/WAIM 2007 International Workshops: DBMAN 2007, WebETrends 2007, PAIS 2007 and ASWAN 2007, Huang Shan, China, 2007, Proceedings pp 674-683. Published by Springer Berlin Heidelberg.

[26] D.W. Cheung, J. Han ,V. Ng and C.Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In Proc. 1996 Int. Conf. Data Engineering (ICDE'96), pp-106-114, New Orleans, LA, Feb. 1996.

[27] Md. Tabrez Nafis, Ranjit Biswas "A Secure clustering Technique for Unstructured and Uncertain Big data". In proceeding of 1st International Conference on Advanced Computing and Intelligent Engineering (ICACIE) Springer, 451, 2016.