Research Paper

Volume : 2 | Issue : 9 | September 2013 • ISSN No 2277 - 8179

A Proposal to Effective Long Itemstes Association Rule Mining LIARM (Applied on proposed stego-images dataset)

Computer Science

KEYWORDS : data mining, association rules, stego-images, large itemsets, maximal itemset

Soukaena H. Hashem	Computer Sciences Department, University of Technology, Baghdad, Iraq, 2013
Alia K. Abdulhassan	Computer Sciences Department, University of Technology, Baghdad, Iraq, 2013
Hala B. Abdulwhab	Computer Sciences Department, University of Technology, Baghdad, Iraq, 2013

ABSTRACT

An attempt to improve the efficiency of the association rules technique in data mining is done by proposing an algorithm that deals with databases with extremely long itemsets. This algorithm which focuses on the long size of itemsets is called LIARM (Long Itemsets Association Rules Mining). The set of the most frequent itemsets has been found based on finding the longest frequent itemset by using a proposed traversing approach called two-phase traversing approach that aide to reduce the large search space in which it is important to speed up the process and save the storage area. The remaining frequent itemsets are searched and appended to the previous set. The complete set of frequent itemsets has been classified to normal, closed and maximal to support and strengthen the prediction in the analysis stage. And then the set of association rules for the itemset's has been extracted using the traditional association rules procedure. The proposal applied on proposed system is to improve the usage of steganography by the following considerations:

- *First build a proposed database contains a huge number of stego-images with all the feature of the stego hiding and detecting operations.*
- *Mining the proposed database using proposed LIARM technique will be effective because the itemsets which represent the stego features are very long and this will take considerable amount of time by using the traditional association rule mining.*
- *After generating association rules the analysis stage will begin, this stage will explain the discovered knowledge, the new patterns will represent the relationships among the features of stego. These patterns will predicate a new knowledge that will be useful for administrators to improving the steganography to be immune against the steganalysis.*

1. Introduction

Data mining is the search for relationships and global patterns that exist in large databases but are ' hidden' among the vast amount of data, such as a relationship between patient data and their medical diagnosis. These relationships represent valuable knowledge about the database and the objects in the database and, if the database is a faithful mirror, of the real world registered by the database [1, 2]. Efficient discovery of frequent patterns from databases is an active research area in data mining with broad applications in industry and deep implementations in many areas of data mining. So many frequent-pattern mining algorithms have been proposed: Doug Burdick, Manuel Calimlim, and Johannes Gehrke proposed in 2001, "MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases", for mining maximal frequent itemset from a transactional database. This algorithm is efficient when the itemset in the database is very long. The search strategy of this algorithm integrates depth first traversing of the itemset lattice with effective pruning mechanisms [3]. Doug Burdick, Manuel Calimlim, Jason Flannick, Tomi Yiu and Johannes Gehrke presented in 2001, a performance study of MAFIA. In this study MAFIA has been tested with different types of databases [4]. Takeaki Uno, Masashi Kiyomi and Hiroki Armura, proposed the second versions of LCM, LCM fre and LCM max, for enumerating closed, all, and maximal frequent itemsets. LCM is an abbreviation of Linear time Closed itemset Miner. In this version, the problems of dense datasets can be solved in short time by adding database reduction to LCM, moreover LCM max includes a pruning method, thus the computation time is reduced when the number of maximal frequent itemset is very small [5]. Mohammad El-Hajj and Osmar R. Zaiane proposed in 2005, a new traversing approach for the itemset lattice called leap-traversal. Based on this approach they have proposed a framework of three algorithms called COFI-CLOSED, COFI-MAX, and COFI-ALL for mining closed, maximal, and all patterns respectively, forming the YAFIMA package (Yet Another Frequent Itemset Mining Algorithms) to mine effectively small and extremely large databases [6]. H. D. K. Moonesinghe, Samah Fodah, and Pang-Ning Tan proposed PGMiner, a graph-based algorithm for mining frequent closed itemsets. This algorithm consists of constructing a prefix graph structure and decomposing the database to variable length bit vectors, which are assigned to nodes of the graph to facilitate fast frequency counting of the itemsets via intersection operation, it also devised several inter-node and intra-node pruning strategies to substantially reduce the combinatorial search space [7]. Jianyong Wang and George Karypis, proposed several pruning methods and optimization techniques support constraint and developed an algorithm called BAMBOO that finds all closed itemsets satisfying the length-decreasing support constraint. BAMBOO incorporates two pruning methods called invalid item pruning and unpromising prefix pruning that are sensitive to the structure of the projected databases and eliminate unpromising portions of the search space [8].

2. The General Proposed Algorithm of LIARM

The process for finding association rules has two separate phases. In the first phase, the set of frequent itemsets in the database must be found. In the second phase, this set has been used to generate interesting patterns. The proposed algorithm that attempts to improve the efficiency of association rules technique in databases with extremely long itemsets. This algorithm provides a proposed way to generate the set of frequent itemsets in a manner that speeds up the process, then classify this set to Maximal Frequent Itemsets MFI, Closed Frequent Itemsets CFI, and Normal Frequent Itemsets NFI which it is important to support and strengthen the prediction with the association rules. The proposed algorithm finds the set of the most frequent itemsets from the longest frequent itemset using the two-phase approach. Find the remaining frequent itemsets using depth first approach and append them to the set of the most frequent itemsets to constitute the final set of frequent itemsets. Count the support of each frequent itemset by scanning the database and then classify the frequent itemsets to MFI, CFI and NFI. Find the set of association rules according to the sets of MFI, CFI, and NFI. Analyze the resulting association rules from the database after implementing the proposed algorithm on proposed stego-image dataset.

The Steps in Long Itemssets Association Rule Mining are:

1. Finding the longest frequent itemset by using the two-phase traversing approach.
2. Generating the set of the most frequent itemsets from the longest frequent itemsets.
3. Finding the difference frequent itemsets by using depth first approach and appending them to the set of frequent itemsets.
4. Classifying the set of frequent itemsets to three classes: maximal, closed, and normal.
5. Generating the set of association rules by using the tradi-

tional association rules procedure, finally analyze them.

6. Build a proposed database consists of long itemsets to prove our proposal.

LIARM Algorithm

The Input: The proposed database DB (with long itemsets), The Minimum support Min_supl, Minimum confidence Min_confl.

The Output: The set of association rules support the database with concern of proposed classification maximal, closed, and normal itemsets.

- First step: Find the longest frequent itemset by using the two-phase approach.
- Second step: Find the set of the most frequent itemsets from the longest frequent itemset.
- Third step: Find the difference frequent itemsets using depth manner and append them to the set of frequent itemsets.
- Fourth step: Count the support of each frequent itemset.
- Fifth step: Find the sets of maximal, closed, and normal frequent itemsets.
- Sixth step: Find the set of association rules.
- Seventh step: Analyze the association rules to introduce knowledge used in improvement.

3. Proposal of Finding Long Itemset's

To find the set of association rules in long itemsets database, the set of frequent itemsets must be found. A proposed algorithm for finding the set of frequent itemsets in transactional databases has been used in this part. This algorithm deals with databases that have long itemsets by using a proposed traversing approach called two-phase approach that consists of depth and breadth search to find the longest frequent itemset. Once the longest frequent itemset has been found all of its children have been generated to constitute the set of the most frequent itemsets. So instead of checking each item if it is frequent or not only it needs to search for the longest frequent itemset and then generate all of its children which they are certainly frequent. The efficiency of this process appears when the initial items represent the longest frequent itemset.

It is important to note that when the initial items do not represent the longest frequent itemset some of frequent itemsets don't appear in the longest frequent itemset children which are called difference frequent (DF). An exception is made for those itemsets by checking the support of each item that doesn't appear in longest frequent itemset, taking the frequent ones and appending them to the beginning of the longest frequent itemset. The resulting set of itemsets has been represented in a tree which starts with null root; the first level consists of 1-extention while the second level consists of 2-extention, and so on. This tree has been traversed in depth, by taking each itemset, counting its support in the database, and adding the frequent ones to the set of frequent itemsets. In fact, the tree is pruned and only the itemsets that contains the difference frequent itemsets have been checked. The longest frequent itemset children are excluded form the search by checking only the branches that contain the frequent items that do not appear in the longest frequent itemsets. The supports of the discovered frequent itemsets have been computed by counting the occurrence of each element in the database. Then this set has been classified to three classes: the first one is the set of Closed Frequent Itemsets (CFI) an itemset is closed if it appears in the other frequent itemsets but not with the same support, the second class is the set of Maximal Frequent Itemsets (MFI) an itemset is maximal if it does not show in any other frequent itemset, the third class is the set of Normal Frequent Itemsets (NFI) an itemset is normal if it is neither closed nor maximal.

In other words, let I be the set of items, I= {1,..., n}, and let $X \subseteq I$ an itemset, X is called k-itemset if the cardinality of X is k. Let T database be a multiset of subsets of I, and let support (X) be the percentage of itemsets Y in T such that $X \subseteq Y$. If X is frequent and no superset of X is frequent, then X is called maximally frequent. An itemset X is called closed if an itemset X' dose not exist such that $X' \supset X$ and t(X) =t(X'), with t(Y) defined as the set of transactions that contain itemset Y. It is straightforward to see that the following relationships hold: MFI $\subseteq$ CFI $\subseteq$ FI. This means that the set of MFI is orders of magnitude smaller than the set of CFI, which itself is orders of magnitude smaller than the set of FI as shown in Figure (1).

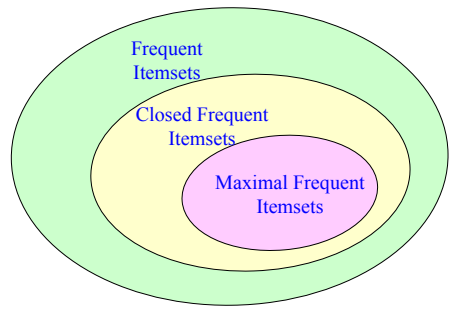

Figure (1) The relationship between the sets of maximal, closed, and, frequent itemsets.

After classification process the association rules are generated using the traditional association rules procedure. Each rule must pass the user specified minimum confidence threshold. Finally, the resulting association rules from the database are analyzed.

3.2. Two-phase Traversing Approach

Many algorithms find patterns using one of the two methods, namely: breadth search or depth search. Breadth search can be viewed as bottom up approach where the algorithm visits patterns of size k+1 after finishing the k sized patterns. The depth search dose the opposite where the algorithm starts by visiting patterns of size k before those of size k-l. Because the goal is to find the longest frequent itemset a combination of these two approaches has been proposed to use in an approach known as two-phase traversing. This approach has been used to minimize the amount of nodes to be visited, in which it is important to reduce the search space and speed up the search process. Two-phase traversing approach consists of two phases the first phase must be applied while the second phase will be applied when it is necessary. The first phase of that traversing consists of depth search from this search only the deepest node on the left most side has been taken. The support of this node in the database has been computed, if its support passes the minimum support threshold, the search will be terminated; otherwise the second phase will be applied. The second phase of the two-phase approach consists of breadth search by taking the node that has been generated in the first phase and considering it the root of the tree, then traversing that tree in breadth manner looking for the longest frequent itemset. The search will be terminated when the longest frequent itemset has been found. A detailed explanation of the two phases will be presented in the following two points:

2.1.2.1. The First Phase: Depth Search Phase

Depth first search always expands the deepest node in the current fringe of the search tree. So the search proceeds immediately to the deepest level of the search tree. In this phase, only the deepest node on the left most side has been taken.

For example, suppose the set of initial items are (A,B,C,D,E,F,G,H), those items are represented in a tree which starts with null root, each node in the first level consists of 1-extension, while each node in the second level consists of 2-extension and so on. The father of each node is called the node's head and the following items called the node's tail. Each node consists of the node's head U 1-item in the node's tail. Figure (2) shows the depth phase of the proposed search. The cycled node is the only node that has been taken and checked in that phase.

Volume : 2 | Issue : 9 | September 2013 • ISSN No 2277 - 8179

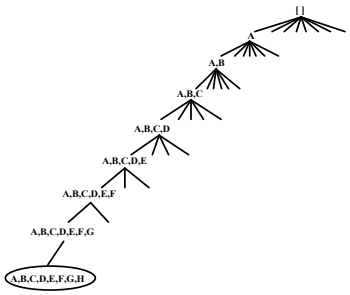

Figure (2) The depth phase of the two-phase approach.

2.1.2.2. The Second Phase: Breadth Search Phase

Breadth first search is a simple strategy in which the level k is traversed before traversing level k+1. In this phase, the node that has been generated in the first phase is considered to be the root of the lattice that will be traversed. The children of each node are derived from their father after removing one item from each one. The lattice of the itemset that has been generated in Figure (2) is shown in Figure (3). Suppose the longest frequent itemset in the database is (B,C,D,E,F,G).

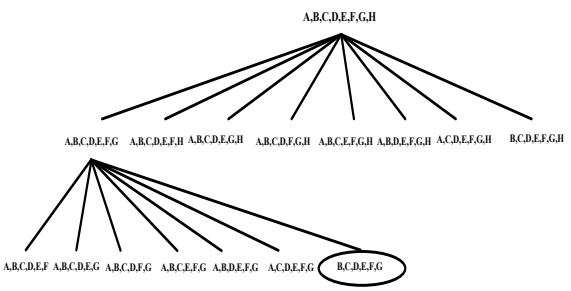

Figure (3) The breadth phase of the two-phase approach.

The main reason for that representation is to check the itemset of length l before checking the itemsets of length l-1, which guarantees that the longest itemset will be chosen. This phase will be terminated when the longest frequent itemset will be found.

3. Applying LIARM in Proposed Stego-Image Database

The proposed system aims to improve the usage of steganography that by extracting the knowledge and predicate implicit pattern explain implicit relationships among all features of stego images. To explain our proposed system we will describe it in detail in the following steps:

First step: Prepare the stego images database and this database is transactional one. So it has two attributes the first one as traditional called TID which contain the numbers identifier for stego images. The second attribute will contain items these items represent all features of stego image related with the two operations hiding and hiding detection.

Second step: in this step we will determine all the important features of stego and coding them as in the following:

- Size of stego image 1024*1024: A else the size small than then no A.
- Hide text : B . else hide image then no B.
- If hide image size of image 1024*1024: C. else the size small than then no C.
- If hide text is it encrypted: D. else not encrypted no D.
- Method of hiding is Hide : E. else then no E.
- Method of hiding is Seek : F. else then no F.
- Method of detect success is Average Absolute Difference (AD) Test: G. else then no G.
- Method of detect success is Mean Squared Error (MSE) Test: H. else then no H.
- Method of detect success is Laplacian Mean Squared Error (LMSE) Test: I. else then no I.
- Method of detect success is Signal-to-Noise Ratio (SNR) Test: J. else then no J.
- Method of detect success is Peak Signal- to- Noise Ratio (PSNR) Test: K. else then no K.
- Method of detect success is Normalized Cross-Correlation (NCC): L. else then no L.
- Method of detect success is Correlation Quality (CQ) Test: M. else then no M.
- Method of detect success is Histogram Similarity (HS) Test: N. else then no N.
- Method of detect success is Chi Square Test: O. else then no O.
- Method of detect success is Visual Pixel Detection: P. else then no P.
- Method of detect success is Stegdetect: Q. else then no Q.

Third step: in this step we will display a sample of the proposed stego image database, see table (1).

Table (1) Sample of Proposed Stego-Image Database

TID	Specifications	TID	Specifications
1	A,F,G,I,J,K,L,M	26	C,D,E,F,G,H,I,J,K,L,M
2	C,,D,E,G,H,I,J,K	27	A,F,G,H,I,J,K,L,M
3	A,C,D,E,F,J,K,L,M,O	28	C,D,E,F,G,H,I,J,K
4	A,C,D,E,F,G,I,K,O	29	A,D,E,F,G,K,L,M,O
5	C,D,E,F,G,H,I,J	30	C,D,E,F,G,H,I,J,K,L,M,N,O
6	C,D,E,F,G,H,I,J,K,L,M,N,O	31	C,D,E,F,G,H,I,J,K
7	F,G,H,I,J,K,L,M,O	32	C,D,E,F,G,H,I,J,K,Q
8	C,D,E,F,I,J,K,M,P,Q	33	A,F,G,H,I,J,K,L,M
9	D,E,F,G,H,I,J,K,L,M	34	A,C,D,E,F,G,I,K,O
10	C,D,E,F,G,H,I,J,K,L,M,N,O	35	A,B,C,D,E,F,G,H,I
11	A,D,E,F,G,H,K,L,M	36	C,D,E,F,G,H,I,J,K,L,M
12	C,D,E,F,K,L,O,P	37	C,D,E,F,G,H,I
13	C,D,E,F,G,H,I,J,K,L,M,N,O	38	C,D,E,F,G,H,I,J,K,L,M,N,O
14	A,D,E,F,I,K,L,M,O	39	C,D,E,F,G,H,I,J,K
15	C,D,E,F,G,H,I,J,K,L,M,N,O,Q	40	A,C,D,E,F,G,I,O,K
16	C,D,E,F,G,H,I	41	C,D,E,F,G,H,I
17	C,D,E,F,G,H,I,J,K,L,M,N,O	42	C,D,E,F,G,H,I,J,K,L,M,N,O
18	C,D,E,F,G,H,I,J,K,L,M,N,O	43	C,D,E,F,G,H,I,J,K,L,M,N,O
19	A,D,E,F,G,H,K,L,M	44	A,C,D,E,F,G,I,K,O
20	C,D,E,F,G,H,I,J,K,L,M,N,O	45	A,D,E,F,G,K,L,M,O
21	B,C,D,E,F,G,H,I	46	C,D,E,F,G,H,I,J,K
22	B,C,D,E,F,G,H,I,J,K	47	C,D,E,F,G,H,I,J,K,L,M,N,O
23	A,D,E,F,G,H,K,L,M	48	A,F,G,H,I,J,K,L,M
24	C,D,E,F,G,H,I,J,K,L,M,N,O	49	C,D,E,F,G,H,I,J,K,L,M,N,O
25	C,D,E,F,G,H,I,J,K	50	C,D,E,F,G,H,I,J,K

Fourth step: to mining such these database we first use the apriori algorithm which not be efficient because the time take it to find the frequent item set was too much. For that reason we will use LIARM mining method for reduce the search space as possible as then reduce the time of finding the frequent itemset and generating the association rules.

Fifth step: After extracting the rules now we analyzed them to find the different relations among all the features of the stego for both operations hiding and detecting hiding.

5. Discussion and Experimental Works

The present study has dealt with databases have long number of features, and in order to give an example of such databases, it

has been proposed that a stego image database. This will make the steganograhy applications such as steaganograhy usage and stego detection more powerful. It is impossible to apply Apriori algorithm directly on these databases because they are contains a set of attributes, the long itemsets included made the search space very large and need tremendous amount of database scans when looking for the frequent itemsets. it has been proposed to find the set of the most frequent itemsets by finding the longest frequent itemset and generating all of its children which they are also frequent to constitute this set. Searching for the longest frequent itemset is done by using a proposed two-phase approach that aids to reduce the search space.

An exception is made for those frequent itemsets that do not appear in the longest frequent itemset children which is called difference frequent by searching for those itemsets in depth, pruning the infrequent ones depending on the Apriori property "if an itemset X is infrequent then any superset of X is also infrequent", and excluding the longest frequent itemset children in a way that reduces the search space and speeds up the process of finding those itemsets. The resulting itemsets from this process have been appended to the set of frequent itemsets to constitute the final set of frequent itemsets. It is important to note that when there is a big difference between the length of initial and the length of the longest frequent itemset many database scans are required to search for the difference frequent itemsets. In this case Apriori algorithm is used instead.

The support of the discovered frequent itemsets have been computed by scanning the database (note that this is done to the children of the longest frequent itemset while the support of the other frequent itemsets have been counted during the process of checking them if they are frequent or not). The resulting set of frequent itemsets with their support has been classified to maximal, closed, and normal. During the classification process, the longest frequent itemset is excluded from the checking and classification because it represents a special case of frequent itemsets. Finally, the association rules of that part have been extracted using the traditional association rules procedure and the results have been analyzed.

For example:
After determining the minimum support and the minimum confidence thresholds for that part (here the minimum support threshold equals 20% from the total number of records and the minimum confidence threshold equals 90%), the longest frequent itemsets will be generated by using the two-phase approach. In the depth phase, the itemset that has been generated is (A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P). The support of this itemset is 0 (less than the minimum support threshold), so the breadth phase will be applied. After applying the breadth phase, the longest frequent itemset is generated and it is equal (C,D,E,F,G,H,I,J,K,L,M,N,O). Once the longest frequent itemset is found, all of its children will be generated to constitute the set of the most frequent itemsets. The support of those children are computed by scanning the database. The difference frequent itemsets will be generated by scanning the database searching for the frequent items that do not appear in the longest frequent itemset. In current example only the frequent item (A) doesn't appear in the longest frequent itemsets. This item is appended to the beginning of the longest, so the resulting set of items are (A,C,D,E,F,G,H,I,J,K,L,M,N,O). Those items will be represented in a tree and only the branch that rooted with A is needed to traverse in depth while the other branches are excluded from the search. In fact, the infrequent itemset that headed with A are pruned from the search to speed up the process. For example, if AC itemset is infrequent then all the branches that headed with AC are pruned. The resulting difference frequent itemsets will be appended to the set of the most frequent itemsets to constitute the complete set of frequent itemsets.

The resulting frequent itemsets are classified to maximal, closed, and normal. Note that, the longest frequent itemset are excluded from the checking and classification because it represents a special case of frequent itemsets. Some of those itemsets are:

C,D,E,F,G,H,I,J,K,L,M,N : is maximal because it dose not appear in any frequent itemsets.

F,K,O : is closed because it dose not appear in any frequent itemsets with the same support.

C,E : is normal because it dose not appear the sets of closed and maximal frequent itemsets.

Finally, the association rules will be generated by using the traditional association rules procedure. Some of the resulting rules are:

DFGJK $\Longrightarrow$ CH

DGIK $\Longrightarrow$ CF

The analysis stage represents the most important stage done by the miner who has encoded the database. Since the results are the encoded association rules, they must be analyzed to make them clear and useful predicted patterns. In this section some of association rules that have been extracted from the previous application will be analyzed. The following rules have been generated from the previous application:

1) FKO $\Longrightarrow$ CE

2) CDEFGHIJKLMO $\Longrightarrow$ N

These rules give the knowledge that both sides of rules are itemsets in the itemset's part.

· Rule 1 means If method of hide is Seek, detect success is Peak Signal- to- Noise Ratio (PSNR) and also detect success is Chi Square Test Then hide image size of image 1024*1024 and Method of hiding may be Hide. Note that, the stego (CE) is NFI and the stego (FKO) is CFI, from that it has been predicted that a stego with (FKO) specifications will never be found in any other stego with the same support.

· Rule 2 means IF hide image size of image 1024*1024, hide text is it encrypted, Method of hiding is Hide, Method of hiding is Seek, Method of detect success is Average Absolute Difference (AD) Test, Method of detect success is Mean Squared Error (MSE) Test, Method of detect success is Laplacian Mean Squared Error (LMSE) Test, Method of detect success is Signal-to-Noise Ratio (SNR) Test, Method of detect success is Peak Signal- to-Noise Ratio (PSNR) Test, Method of detect success is Normalized Cross-Correlation (NCC), Method of detect success is Correlation Quality (CQ) Test, and Method of detect success is Chi Square Test Then Method of detect Histogram Similarity (HS) Test will success. Note that a stego (CDEFJHIJKLMO) is MFI and a stego (N) is NFI from that it has been predicted that a stego with (CDEFGHIJKLMO) specifications will never be found in any other stego because it is MFI.

5 Conclusions
After implementing the proposed algorithm, the following points have been concluded:

1. The proposed two-phase traversing approach can also be used in other applications that require finding the shortest path to the goal.
2. Classifying the frequent itemsets to three classes strengthens the efficiency of the analysis stage e.g., in steganograhy applications, this step is important to deal with steaganograhy usage and stego detection techniques.
3. Taking in the account the negative rules in addition to the positive ones has made the analysis stage more efficient.
4. Since the proposed database is take all the features of stego image that will make the obtained knowledge more efficient and strong that will support the improving of steganography.
5. LIARM technique is very efficient with our proposed database since it deal with long itemsets and our database has long itemsets caused by considering all features for stego

image for both operation hiding and detecting.

6. Analysis process is basic stone for improving the steganography, that by explaining the prediction with clear and scientific view to support the administrators for improving.

REFERENCE

1. Shaema Akram, " A Proposed Technique To Improve Association Rules In Data Mining ", M. Sc. Thesis in Computer Sciences, Computer Sciences Department, University Of Technology, 2008. | 2. Hilal Hadi Saleh, Soukaena Hassan, Shaima Akram, "Suggestions To Improve The Efficiency Of Association Rules Techniques In Data Mining", Al-Taqani, ISSN 1818-653x, Vol. 23, No. 4, 2010. | 3. Burdick D., Calimlim M., and Gehrke J., "MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases", Department of Computer Science, Cornell University, 2001. | http://WWW.cs.cornell.edu/boom/2001sp/yiu/mafia-camera.pdf | 4. Burdick D., Calimlim M., Flannick J., Gehrke J., and Yiu T., "MAFIA: A Performance Study of Mining Maximal Frequent Itemsets", Department of Computer Science, Cornell University, 2001. | http://WWW.sunesite.informatik.rwth-aachen.de/publications/CEUR-WS/vol-ao/burdick.pdf | 5. Uno T., Kiyomi M., and Arimura H., "LCM ver.2: Efficient Mining Algorithms for Frequent, Closed, Maximal Itemsets" Japan, 2004. | http://research.nii.ac.jp/~uno/papers/0411/cm2.pdf | 6. El-Hajj M. and Zaiane O. R., "YAFIMA: Yet Another Frequent Itemset Mining Algorithm", Department of Computer Science, University of Alberta, Edmonton, AB, Canada, 2005. | http://WWW.cs.ualberta.ca/~zaiane/postscript/idm05.pdf | 7. Moonesinghe H. D. K., Foda S., and Tan P. N., "Frequent Closed Itemset Mining Using Prefix Graphs with an Efficient Flow-Based Pruning Strategy", Department of Computer Science and Engineering, Michigan State University,2005. | http://WWW.cse.msu.edu/~ptan/papers/ICDM1.pdf | 8. Wang J. and Karypis G., "BAMBOO: Accelerating Closed Itemset Mining by Deeply Pushing the Length-Decreasing Support Constraint", 2005. | http://WWW.siam.org/meetings/sdm04/proceedings/sdm04-041.pdf |