IEEE 3rd International Conference on Engineering, Technologies & Social Sciences (ICETSS 2017)

The conference will be held in Asian Institute of Technology Conference Center (AITCC) Bangkok Thailand from 07-08 August, 2017.

AIT Asian Institute of Technology

The IEEE 3rd ICETSS-2017 is being organized by IEEE IIUM SB, technically supported by the IEEE IMS Malaysia Chapter with Collaboration of ETSS Management Malaysia, Knowledge-Now England, Multimedia University Malaysia, Balochistan University of Engineering and Technology and Al-Kitab University College Iraq. The theme of this conference is "Engineering, Technologies & Applications of Social Sciences are Driving our Future". The IEEE 3rd International Conference on Engineering, Technologies and Social Sciences 2017 (ICETSS-2017) will provide a meeting place for the sharing of novel ideas and research findings in the field of engineering, technologies, social sciences & business management. Its main goal is to foster multidisciplinary exchange by researchers and developers as well as research students and professional experts. We invite original and unpublished work by Academics, Researchers, Business Leaders, Experts and Executives from Universities and industrial research institutes to submit for the conference.

PUBLICATION

IEEE

ICETSS is IEEE indexed conference, Code #41681, ISBN No. 978-1-5386-1611-6CFP17L91-ART. All presented papers will be submitted for inclusion in to IEEE Xplore Digital Library as per their format. Extended version of all accepted papers will be submitted to Scopus and ISI Journals.

Scopus

THOMSON REUTERS

CONFERENCE TRACKS

The conference will consist of three tracks, that is, Engineering, Technologies and Social Sciences. Topics of interest in each track include, but not limited to the following:

- **Telecommunication, Networking & Broadcast Technologies**
- **Components, Circuits, Devices and Systems**
- **Computer, Information Technology, Software Engineering**
- **Engineered Materials, Dielectrics and Plasmas**
- **Engineering Profession, Ethics, Safety and Management**
- **Fields, Waves and Electromagnetics**
- **General Topics for Engineers**
- **Geoscience, Remote sensing, Tracking and Measurement**
- **Power, Energy and Industry Applications**
- **Robotics and Control Systems**
- **Signal Processing and Analysis**

IMPORTANT DATES

- **Submission of Full Paper** : ~~30 April 2017~~
  Deadline Extended to : May 10,2017
- **Notification of Acceptance**: 10 June 2017
- **Camera Ready Copy** : 30 June 2017
- **Registration Deadline** : 05 July 2017

Conference Dates 07-08 August 2017

Contact Us

Web: http://icetss.etssm.org/

E-mail: g.chair@icetss.etssm.org

ORGANIZERS

ETSS Management · MMU MULTIMEDIA UNIVERSITY · KNOWLEDGE Get The Leading Edge · MAX STRATEGY · IEEE

HYBRID: An efficient unifying process to mine frequent itemsets

2 Author(s) Nurul Fariza Zulkurnain ; Ahmad Shah View All Authors

Abstract:
Current advancement in technology inexorably leads to data flood. More data is generated from banking, telecom, scientific experiments, etc. Data mining is the process of extracting useful information from this flooded data, which helps in making profitable future decisions in these fields. Frequent itemset mining is one of the focus research areas and an important step to fin association rules. Time and space requirements for generating frequent itemsets are of utter importance. Algorithms to mine frequent itemsets effectively help in finding association rules and also help in many other data mining tasks. In this paper, an efficient hybrid algorithm was designed using a unifying process of the algorithms Improved Apriori and FP-Growth. Results indicate that the proposed hybrid algorithm, albeit more complex, consumes fewer memory resources and faster execution time.

Published in: 2017 IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS)

Abstract | Authors | Figures | References | Citations | Keywords | Metrics | Media

Abstract:
Current advancement in technology inexorably leads to data flood. More data is generated from banking, telecom, scientific experiments, etc. Data mining is the process of extracting useful information from this flooded data, which helps in making profitable future decisions in these fields. Frequent itemset mining is one of the focus research areas and an important step to fin association rules. Time and space requirements for generating frequent itemsets are of utter importance. Algorithms to mine frequent itemsets effectively help in finding association rules and also help in many other data mining tasks. In this paper, an efficient hybrid algorithm was designed using a unifying process of the algorithms Improved Apriori and FP-Growth. Results indicate that the proposed hybrid algorithm, albeit more complex, consumes fewer memory resources and faster execution time.

Published in: 2017 IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS)

Date of Conference: 7-8 Aug. 2017

Date Added to IEEE *Xplore*: 26 March 2018

▼ **ISBN Information:**
Electronic ISBN: 978-1-5386-1611-6
Print on Demand(PoD) ISBN: 978-1-5386-1612-3

INSPEC Accession Number: 17652323

DOI: 10.1109/ICETSS.2017.8324140

Publisher: IEEE

Conference Location: Bangkok, Thailand

Contents

SECTION I. Introduction

Today, information has become the ultimate power that could lead to success. The current advancement in technology has led to the overwhelming process of storing and retrieving a huge amount of data. From this large amount data, it is important to look for ways to analyze and retrieve useful information. Knowledge discovery in databases (KDD) help to extract useful information from this flooded data, which can lead towards making profitable future decisions in many fields [6]. It assists in uncovering important patterns which are useful in information extraction.

Data mining is one of the well-known steps in the KDD process and is a focused

HYBRID: An Efficient Unifying Process to Mine Frequent Itemsets

Nurul Fariza Zulkurnain
Electrical and Computer Engineering
International Islamic University Malaysia
Selangor, Malaysia
nurulfariza@iium.edu.my

Ahmad Shah
Electrical and Computer Engineering
International Islamic University Malaysia
Selangor, Malaysia
Ahmad.s728@gmail.com

Abstract— *Current advancement in technology inexorably leads to data flood. More data is generated from banking, telecom, scientific experiments, etc. Data mining is the process of extracting useful information from this flooded data, which helps in making profitable future decisions in these fields. Frequent itemset mining is one of the focus research areas and an important step to fin association rules. Time and space requirements for generating frequent itemsets are of utter importance. Algorithms to mine frequent itemsets effectively help in finding association rules and also help in many other data mining tasks. In this paper, an efficient hybrid algorithm was designed using a unifying process of the algorithms Improved Apriori and FP-Growth. Results indicate that the proposed hybrid algorithm, albeit more complex, consumes fewer memory resources and faster execution time.*

Keywords—data mining; frequent itemset; association rule; big data.

I. INTRODUCTION

Today, information has become the ultimate power that could lead to success. The current advancement in technology has led to the overwhelming process of storing and retrieving a huge amount of data. From this large amount data, it is important to look for ways to analyze and retrieve useful information. Knowledge discovery in databases (KDD) help to extract useful information from this flooded data, which can lead towards making profitable future decisions in many fields [6]. It assists in uncovering important patterns which are useful in information extraction.

Data mining is one of the well-known steps in the KDD process and is a focused research area because of its importance. The sum of tasks such as classification, clustering, regression, and association rule in data mining is used to extract knowledge from large databases.

Association rule mining is one of the most highlighted areas of research in data mining. A huge amount of data is generated from day to day activities and mining association rule helps to analyze the enormous amount of data and make correct future decisions in many industries [12]. It helps to identify the relationship between items, e.g. in the transactional database to determine the buying pattern. This is important in order to enhance businesses. The concept of association rule mining is explained in [2].

Frequent itemset mining is an important step for many data mining tasks. The discovery of frequent itemsets assists in the investigation of interesting patterns from the database using methods such as correlations, sequences, classifiers and clusters. The most frequent itemsets can easily be extracted. At lower support constraint, many of the itemsets that exist at this lower end of the support spectrum are overlooked. This is due to the limitation of memory storage, as the size of the dataset increase, the production of frequent itemsets also increase.

The main objective is to optimize the process of finding itemsets efficiently, making improvements in term of memory and execution time, based on desired thresholds. Hence, discovering frequent itemsets during the search process should reduce the computation. The outcome will then assists in detecting important patterns for various decision making purposes [2].

Methodologies used for frequent itemsets mining are categorized as: horizontal layout designed algorithms ([2]; [5]; [10]; [16]; [20]), vertical layout designed algorithms ([3]; [12]), and projected layout designed algorithms ([11]). Several papers have proposed hybrid algorithms in order to improve the performance of its former methods ([8]; [9]).

All these algorithms have its own strengths and weaknesses with different datasets as some algorithms perform well on sparse data and vice versa. For example, most of the algorithms reported in the literature suffer from either large memory resources consumption, or long execution time or both.

This paper proposes the discovery of frequent itemsets from large transactional datasets by designing a new hybrid algorithm, which unifies the strengths of two existing algorithms ([8], [9]). The hybrid algorithm shows that it is better in performance through execution time and memory resource consumption.

The paper is organized as follows. Section-II addresses the related work; the outline of the proposed hybrid algorithm is presented in Section-III; in Section-IV deals with result analysis and Section-V concludes.

II. Related Work

Frequent itemset mining is an important step in finding association rules. There are many algorithms for mining frequent itemsets, some are the state of the art algorithms which started a new era in data mining and make the concept of frequent itemset and association rule possible [12]. Others are the variations of the same algorithms used on a different set of data and make improvements in terms of memory and execution time. In this paper, some of the important and widely used algorithms are briefly explained and their strengths and weaknesses are outlined. Below a brief introduction of the algorithms used for mining frequent itemsets and association rules.

Every algorithm has its strength and weakness when it comes to data, minimum support value, time constraint and memory consumptions. Apriori algorithm performs well when it comes to market basket analysis where there are huge numbers of transactions which generate quite a few frequent itemsets [12]. DHP is a variation of the Apriori algorithm which tries to reduce candidate itemsets for better results [10]. It performs well in early stages but leads to I/O overhead in later stages. DIC [5], Partitioning [12], Sampling [16] etc. are also an improvement to reduce database scan and execution time but when it comes to candidates generation its performance worsen. Vertical layout designed algorithm is more time saving then horizontal layout designed algorithm but memory consumption is rather high because of the candidates, database and transaction identifier in main memory. Projected layout designed algorithm like FP-Growth and H-mine perform better then both horizontal and vertical layout designed algorithm because there is no candidate generation but memory consumption is a little high because of the pointer stored in memory. FP-Tree variation such as COFI-Tree [11] and CT-PRO [14] produce the best results in all the stated algorithms. COFI-Tree performs well on dense datasets and its performance degrades for sparse data if the support is low. CT-PRO algorithm performs well on both sparse and dense datasets, yet its pack structure causes challenges.

Improved Apriori algorithm removes the step to generate candidate itemsets which tend to improve the execution time of generating frequent itemsets [1]. Improved Apriori depends on both forward and reverses scan of the given database [15]. If certain conditions are met, the improved Apriori algorithm can reduce the iterations and scan time for the discovery of candidate itemsets [17]. This algorithm mines maximum frequent itemsets and their subset directly and makes a comparison with the items in the database. It prunes all the candidate itemsets according to the support count making sure that all the maximal frequent itemsets are mined.

Compared to the original Apriori, the improved Apriori does not need the operation of joining and pruning and it does not have to process the candidates generated which significantly reduce the database scanning time and require less space to generate frequent itemsets [17]. However, it is not useful if the maximum frequent itemsets cannot be found fast.

One of the latest and widely used algorithms is FP-growth algorithm [7]. Frequent pattern tree or FP-tree is introduced to deal with the problems of Apriori algorithm where all transactions are stored in a tree data structure. Execution time is higher as compared to other algorithms because of the tree data structure.

FP-Growth algorithm uses divide-and-conquer to mine frequent itemsets without candidate generation [7]. It uses a frequent-pattern tree (FP-tree) data structure which helps in retaining the itemset association information. FP-growth compresses the input database and creates FP-tree instances then divide the compressed database according to frequent patterns and then mine each conditional database separately.

It has come to an understanding that Improved Apriori and FP-growth performs better in comparison. Therefore, the proposed algorithm will be a hybrid approach which uses the strength of Improved Apriori and FP-growth to produce better results than both of them.

TABLE I. COMPARISON BETWEEN DIFFERENT FREQUENT ITEMSET MINING ALGORITHMS

Algorithm	**Methodology**	**Strength**	**Limitations**
Apriori	Join and prune	State of the art algorithm	More memory and time consumption
DHP	Hashing technique	Small execution time	Consume more space
Partitioning	Partitioning technique	Utilize less memory due to partitioning	Require more time to find local than global frequent itemset
Sampling	Picking random sample for checking the frequency of the whole database at lower threshold support	Less memory utilization and execution time	Sample selection is difficult
DIC	Dynamic insertion of candidate items	Small execution time	Require different amount of memory at different point
Improved Apriori	Forward and reverse scan	Less memory utilization and small execution time	Useful if the maximum frequent itemsets cannot be found fast
Eclat	Intersection of ids list is used for generating candidate itemsets	Less memory allocation if itemsets are small in number with small execution time	Performance is not feasible
FP-growth	Conditional frequent pattern tree	Consume less memory	Execution time is high

III. DESIGN AND IMPLEMENTATION: HYBRID ALGORITHM

In this section, the hybrid algorithm using a unifying process to combine Improved Apriori and FP-growth is presented.

The design of the HYBRID algorithm included the property of the Apriori that non-empty subsets of the frequent itemsets are also frequent [2]. The detailed step of the HYBRID algorithm is shown in Algorithm 1.

In the first part of the algorithm, the Improved Apriori property [2] was used to discover all the maximal frequent itemsets which are repeating in the transactional database with a support value equal to or greater than the minimum support specified [20]. There are still many itemsets which are frequent-1 but not included in the maximal frequent itemsets. So the database which contains frequent-1 elements are pruned but there are no maximal frequent itemsets which make the database smaller and easy to traverse.

The pruned database becomes the input in the second part of the algorithm which discovers all the frequent-1 itemsets and removes all the infrequent-1 itemsets from the transaction. Then, the FP-Tree algorithm was implemented by constructing an FP-Tree from the pruned transactions [7]. This part of the algorithm assists in discovering all the frequent itemsets remained from the first procedure.

Algorithm 1. HYBRID algorithm

Input: Transaction database *D*, minimum support, *min_sup*
Output: All frequent itemsets
Method:

1. Scan database D.
2. Create a 2-dimensional array, put the transactions at the count of repetition.
3. Arrange in ascending order according to transaction length.
4. Traverse the array to find maximal transactions (k-itemsets) with support count greater than or equal to the min_sup. Combine k-itemsets and (k-1) itemsets to gether in the next (k-1) maximal itemset and so on to mine all the frequent items. If there are no frequent itemset left go to step 6.
5. Take all the non-empty subset of a frequent itemset.
6. If there are frequent itemsets not included in the maximal itemset then find all the frequent-1 itemsets and prune the database by removing the maximal frequent itemset.
7. Find frequent-2 by the right neighbour method.
8. Construct an FP-tree with binary strings.
9. k = 3
10. WHILE height of tree increase
11. Generate k-layer tree
12. k = k + 1
13. END WHILE
14. Remaining frequent itemsets

A. Illustration with example

Table II shows the items in the 2-dimensional array with the count of repetition after scanning the dataset. Suppose the min_sup is 20% or 2 for algorithm illustration.

Now, the frequent itemset {I1, I2, I4, I5} is obtained with a support count of 2 which is equal to the min_sup specified. Apriori property states that all non-empty subset of frequent itemset must be considered as frequent hence {I1, I2, I4, I5} is a maximal frequent itemset having subsets {I1, I2, I4}, {I1, I2, I5}, {I1, I4, I5}, {I1, I2}, {I1, I4}, {I1, I5}, {I2, I4}, {I2, I5}, {I4, I5}, {{I1}, {I2}, {I4} and {I5}.

It has been discovered that I3 is frequent but not included in the maximal frequent itemset list hence all the itemsets that contain I3 are pruned from the dataset. Those are: {I1, I3}, {I2, I3} and {I2, I3, I4}.

The infrequent 1-item from this group of itemsets will be pruned and items that are left will be used to build the FP-tree. In this example, {I3:3} and {I2:2} are used to build the FP-tree as shown in Fig.1.

The FP-tree procedure assists in mining frequent itemsets that are not included in the maximal itemset. In this case {I2, I3} was left unmined in the Improved Apriori procedure. Hence, the output of frequent itemsets discovered are {I1, I2, I4}, {I1, I2, I5}, {I1, I4, I5}, {I1, I2}, {I1, I4}, {I1, I5}, {I2, I4}, {I2, I5}, {I4, I5}, {{I1}, {I2}, {I4}, {I5} and {I2, I3}.

TABLE II. DATASET

TID	Items
1	I1, I2, I4, I5
2	I2, I3, I4
3	I2, I3
4	I1, I2, I4, 15
5	I1, I3

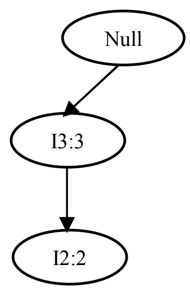

Fig. 1. FP-tree construction

IV. RESULT ANALYSIS

The resulting analysis shows the comparison of HYBRID with Improved Apriori and FP-growth on selected datasets.

A. Execution Time

The execution time of an algorithm is the time required to find all the frequent itemsets in a given dataset. Many experiments were performed on HYBRID using both sparse and dense datasets to evaluate its performance against Improved Apriori and FP-growth.

In the Retail dataset, there are 9000 number of transactions with a high number of repetition as some items are purchased more than once. There are also many maximal frequent itemsets with the support count higher than the minimum support count.

Fig. 2 shows that when it comes to execution time on the Retail dataset, HYBRID algorithm is approximately 55% faster than Improved Apriori and 20% faster than FP-growth. It can also be observed that at lower support count the performance of HYBRID is much better than the Improved Apriori and FP-growth. When the support count increases the performance will then come to a match with FP-growth.

T10I4D100K dataset is a sparse dataset that contains approximately 70000 transactions with too many repetitions. Fig. 3 shows the result of executing the three algorithms. Here a lot of frequent itemsets are left unmined after the Improved Apriori procedure which is taken care by FP-growth procedure to mine all the frequent itemsets in HYBRID. As a comparison, HYBRID is 40% faster than FP-growth and 80% faster than Improved Apriori on this dataset. The gap of the performance becomes less between the algorithms when the support threshold increases.

B. Memory Comparison

Memory consumption is an important aspect of the algorithms use to mine frequent itemsets and plays an important role in the performance. Fig. 4 shows the memory consumption of HYBRID algorithm against Improved Apriori and FP-growth on the Retail dataset. The dataset generates candidate sets which make high memory consumption for Improved Apriori. At low support count, the memory consumption is less for the HYBRID algorithm as compared to FP-growth but as the support increases the memory consumption becomes the same.

Fig. 5 shows the memory performance of all the algorithms on the sparse dataset. In the sparse dataset, there are a large number of unmarked items. Improved Apriori processes the unmarked items and pruned most of the infrequent items in the first pass. Therefore, the memory consumption at a high degree of support matches the HYBRID algorithm. However, for lower support count, HYBRID algorithm performs better as compared to both Improved Apriori and FP-growth.

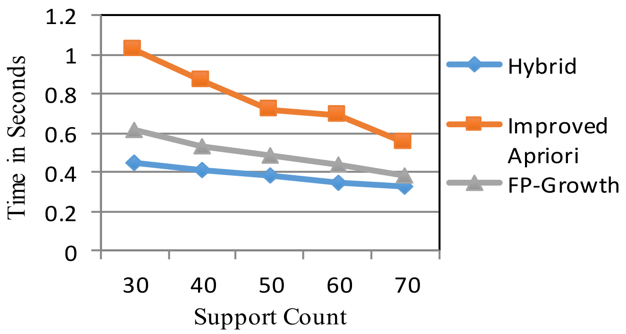

Fig. 2. Execution time for RETAIL dataset

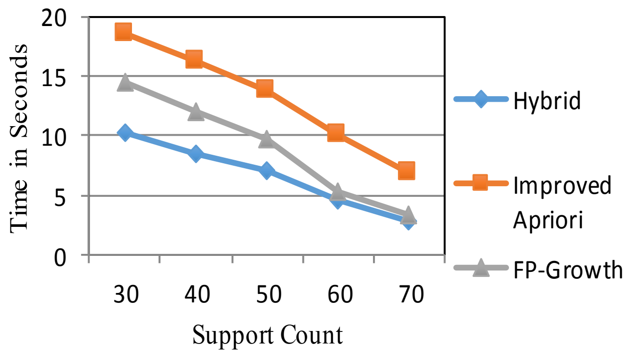

Fig. 3. Execution time for T10I4D100K dataset

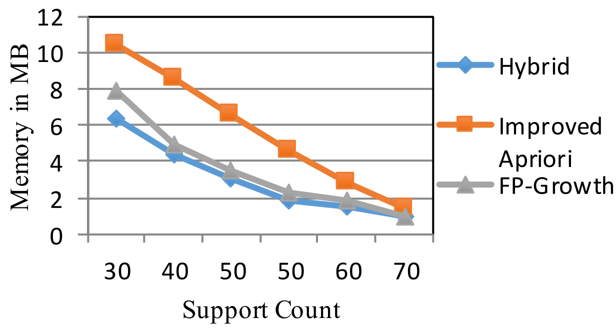

Fig. 4. Memory consumption for RETAIL dataset

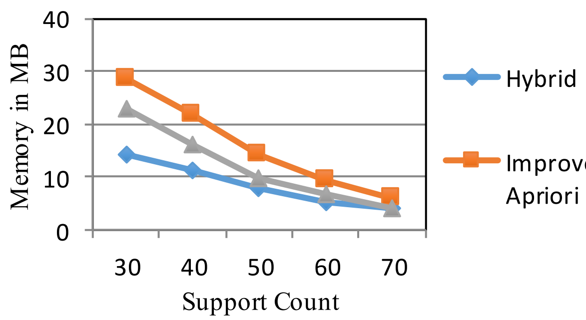

Fig. 5. Memory consumption for T10I4D100K dataset

V. CONCLUSIONS AND FUTURE WORK

The experimental results clearly indicate that when it comes to execution time the HYBRID algorithm performs much better than both Improved Apriori and FP-Growth on both sparse and dense datasets.

Memory consumption is also less in HYBRID algorithm then both Improved Apriori and FP-Growth on both sparse and dense datasets. It is also determined from the experiments that at a lower support value HYBRID algorithm performs much better than the higher support count where the performance almost matches the FP-Growth.

Standard datasets - RETAIL and T10I4D100K were used for this evaluation. The focus was on the execution time and memory consumption and HYBRID algorithm use a unifying process to combine Improved Apriori and FP-Growth algorithm and produce better results than both improved Apriori and FP-Growth.

From the observation of this research, it is crystal clear that performance of all the algorithms depends on support count and type of dataset. This was employed in HYBRID algorithm to generate better results than all the stated algorithms. It was also investigated in this research that if transactions are repeated too much then Improved Apriori is best suited for this type of database to mine frequent itemsets efficiently. The remaining itemsets were treated using FP-tree to find the remaining frequent itemsets. Hence, the algorithm produces all frequent itemsets in the dataset. This way no candidates are generated and FP-tree are produced for pruned database and so it fits into the main memory with ease. HYBRID algorithm save a considerable amount of time and memory and perform better as compared to the other algorithms as showed in the results.

For the RETAIL and T10I4D100K datasets, the execution time of the HYBRID algorithm is less than maximal Apriori. The execution time of the algorithm is also less than the FP-growth at lower support count. The probability of finding frequent itemsets is large at high support count and execution time is almost the same as FP-growth.

The memory consumption of the HYBRID algorithm is also way less then both improved Apriori and FP-Growth.

There are a number of areas to investigate for future research direction on the basis of the work done in this research.

- High dimensional data can be used to compare results and keep an eye on another aspect like memory, result accuracy etc.
- The performance of these algorithms can be evaluated on different datasets e.g. use both sparse and dense for result analysis.

ACKNOWLEDGMENT

This work was supported in part by the Fundamental Research Grant Scheme (FRGS), under grant no. FRGS14-139-0380.

REFERENCES

[1] Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB* (Vol. 1215, pp. 487-499).

[2] Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. In *ACM SIGMOD Record* (Vol. 22, No. 2, pp. 207-216). ACM.

[3] Borgelt, C. (2003, November). Efficient implementations of apriori and eclat. InFIMI'03: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations.

[4] Borgelt, C., 2012. Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(6), pp.437-456.

[5] Brin, S., Motwani, R., Ullman, J.D. and Tsur, S., 1997, June. Dynamic itemset counting and implication rules for market basket data. In *ACM SIGMOD Record* (Vol. 26, No. 2, pp. 255-264). ACM.

[6] Han, J., Cheng, H., Xin, D. and Yan, X., 2007. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*,*15*(1), pp.55-86.

[7] Lan, Q., Zhang, D., & Wu, B. (2009, May). A new algorithm for frequent itemsets mining based on Apriori and FP-tree. In *Intelligent Systems, 2009. GCIS'09. WRI Global Congress on* (Vol. 2, pp. 360-364). IEEE.

[8] Latha, S. P., & Ramaraj, N. (2007, December). Algorithm for Efficient Data Mining. In *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on* (Vol. 2, pp. 66-70). IEEE.

[9] Park, J.S., Chen, M.S., & Yu, P.S. (1995). *An Effective Hash-based Algorithm for Mining Association Rules*, in Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), pp. 175-186.

[10] Said, A. M., Dominic, P. D. D., & Abdullah, A. B. (2009). A comparative study of FP-growth variations. *International Journal of Computer Science and Network Security*, *9*(5), 266-272.

[11] Said, A. M., Dominic, P. D. D., & Abdullah, A. B. (2009). A comparative study of FP-growth variations. *International Journal of Computer Science and Network Security*, *9*(5), 266-272.

[12] Song. M and Rajasekaran , 2006 , A Transaction Mapping Algorithm for Frequent Itemsets Mining, IEEE Transactions on knowledge and Data Engineering, vol. 18, No. 4.

[13] Sucahyo, Y. G., & Gopalan, R. P. (2004, November). CT-PRO: A Bottom-Up Non Recursive Frequent Itemset Mining Algorithm Using Compressed FP-Tree Data Structure. In *FIMI* (Vol. 4, pp. 212-223).

[14] Sun, D., Teng, S., Zhang, W., & Zhu, H. (2007, August). An algorithm to improve the effectiveness of Apriori. In *Cognitive Informatics, 6th IEEE International Conference on* (pp. 385-390). IEEE.

[15] Toivonen, H. (1996, September). Sampling large databases for association rules. In *VLDB* (Vol. 96, pp. 134-145).

[16] Toivonen, H. (1996, September). Sampling large databases for association rules. In *VLDB* (Vol. 96, pp. 134-145).

[17] Wei, Y.Q., Yang, R.H. and Liu, P.Y., 2009, August. An improved Apriori algorithm for association rules of mining. In *IT in Medicine & Education, 2009. ITIME'09. IEEE International Symposium on* (Vol. 1, pp. 942-946). IEEE.

[18] Yen. S. J et. al, 2009 ,The studies of mining frequent patterns based on frequent pattern tree, in proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Lecture Notes of Artificial Intelligence vol. 5476 , pp.232-241.

[19] Yen.S.J. et. al, 2012, A Search Space Reduced Algorithm for Mining Frequent Patterns, Journal of Information Science and Engineering , pp.177-191.

[20] Yong-Qing, W., Ren-Hua, Y., & Pei-Yu, L. (2009, August). An improved Apriori algorithm for association rules of mining. In *IT in Medicine & Education, 2009. ITIME'09. IEEE International Symposium on* (Vol. 1, pp. 942-946). IEEE.