International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6591
***Article Received**: 24 February 2023 **Revised**: 25 March 2023 **Accepted**: 12 April 2023*

MBA: Market Basket Analysis Using Frequent Pattern Mining Techniques

Sallam Osman Fageeri[1*], Mohammad Abu Kausar[2], Arockiasamy Soosaimanickam[3]
[1,2,3]Department of Information Systems, College of EMIS, University of Nizwa
Nizwa, Sultanate of Oman
Sallam, Kausar, arockiasamy@unizwa.edu.om

Abstract—This Market Basket Analysis (MBA) is a data mining technique that uses frequent pattern mining algorithms to discover patterns of co-occurrence among items that are frequently purchased together. It is commonly used in retail and e-commerce businesses to generate association rules that describe the relationships between different items, and to make recommendations to customers based on their previous purchases. MBA is a powerful tool for identifying patterns of co-occurrence and generating insights that can improve sales and marketing strategies. Although a numerous works has been carried out to handle the computational cost for discovering the frequent itemsets, but it still needs more exploration and developments. In this paper, we introduce an efficient Bitwise-Based data structure technique for mining frequent pattern in large-scale databases. The algorithm scans the original database once, using the Bitwise-Based data representations as well as vertical database layout, compared to the well-known Apriori and FP-Growth algorithm. Bitwise-Based technique enhance the problems of multiple passes over the original database, hence, minimizes the execution time. Extensive experiments have been carried out to validate our technique, which outperform Apriori, Éclat, FP-growth, and H-mine in terms of execution time for Market Basket Analysis.

Keywords- Market Basket; Bitwise-Based; Frequent Patterns; Support; Confidence

I. INTRODUCTION

Data mining also can be identified as knowledge discovery in database (KDD) aims to extract valuable, useful, and understandable knowledge and patterns from obtainable databases to discover probably the most relevant and interesting patterns and trends. Data mining is a collection of exploration technique based on advanced analytical methods and tools for handling a large amount of data. The technique can find a novel pattern that may assist an enterprise in understanding the business better and in forecasting [1-3].

Nowadays, it has been used in many different fields, such as health care [4-6], decision support system [5], telecommunication networks [7], crime investigation [8], Intrusion detection [9], and log file analysis [10]. Mining association rule is commonly used in data mining, i.e. to investigate the correlations among the product items that purchased together by customers during a visit to a store. Normally the association rule can be denoted as X Y, where X and Y are two sample products, and the rule condition is that, if item X is purchased, item Y will also be purchased. Two parameters are used to measure the relation of the association rules, for frequent itemset generation support criteria are used, and for extracting association rule confidence is used, with the help of the frequent itemset values generated by using the support. Association rule mining was suggested by Agrawal et al. (1993) [11], and the two main steps include: (1) generate frequent itemsets based on minimum support, (2) generate candidate itemsets based on minimum confidence.

Mining frequent itemsets as well as association rules require to satisfy the two interesting measure, which is minimum support and minimum confidence thresholds, that is specified by the user, the discovered frequent items affected by the specified support value, usually low support value lead to discover great number of frequent items, also based on the Apriori property, that's all superset of frequent itemsets, its subset also will be frequent. Hence, specifying suitable support and confidence value are important criteria.

Support- In [11], the support has been defined as one of the measure parameters used to find the occurrence of an item or set of items among the total number of transactions. In the other words, support can calculate how many times an item or set of items appears in a set of transactions. An item or set of items can be known as frequent or large item if it has greater support. Using probability concept, we can formulate support as:

$$\text{Support} = P(A \cap B) = \frac{\text{number of transactions containing both A an B}}{\text{Total number of transactions}} \quad (1)$$

A and B represents the itemsets in a database D.

Confidence- is used for the purpose of measuring the strength of relation and association between the itemsets [12]. The confidence evaluation determines the probability of an item B occurs in the same transaction that also contains A. In the other

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6591
***Article Received**: 24 February 2023 **Revised**: 25 March 2023 **Accepted**: 12 April 2023*

words, the confidence is used to explore the conditional probability of the used items. The formula of confidence is

$$\text{Confidence} = P(A \mid B) = \frac{P(A \cap B)}{P(A)} = \frac{\text{number of transactions containing both A an B}}{\text{number of transactions containing A}} \quad (2)$$

Association rule can be written in expression as an implication of X Y, where X and Y are items of itemset I. where X I, Y I, and X ∩ Y =. The expression means that if a transactions T contains the items in X, it also tends to contain the items in Y. An illustration of such a rule might be that 60% out of the total transactions that contain milk also contains sugar; 40% of all transactions contain the two items together". Here 60% will be known as confidence of the rule, and 40% will be known as support of the rule. Mining association rules from a set of items idea originates from the data analysis of market-basket, where will be the interest in mining association rules for describing customer's interest in buying product items.

II. LITERATURE REVIEW

Apriori algorithm, introduced by [11], is a first frequent pattern and association rule mining algorithm, in order to control the exponential of candidate itemset growth, the algorithm use the support-based pruning concept, to investigate the concept behind the Apriori principle. Apriori algorithm [11], used to discover the frequent pattern in database transactions. The Apriori algorithm use multiple passes method over the database. Lead to employs an iterative (level-wise search) during the search space, Apriori algorithm start generating candidate 1-itemset C1, the algorithm scans the transactions and count all its items in individually. Based on the given support the frequent 1-itemsets will be determined as L1. Then in the next step the algorithm discover L2 set of frequent 2-itemsets. The algorithm uses L1*L1 to generate the candidate 2-itemset C2. Each itemset in C2 that is greater than the specified minimum support will be add to frequent 2-itemset L2. To determine the frequent 3-itemset L3, the algorithm use L2*L2 to discover candidates of size 3-items C3. Based on the minimum support threshold the frequent L3 will be discovered. The circulation continues like this, until no possibility of generate more combinations.

To overcome the Apriori Bottleneck extensive improvements had been introduced, such as, sampling approach [13], incremental mining [14], dynamic itemset counting [15], hashing technique [16], parallel and distributed mining [17-20], partition technique [21]. Tight upper bound number of candidate patterns derived the association rules, which can be discovered using the level-wise method [22]. The result can be effective through reducing the number of database passes. DHP, proposed by Pork et al.[23] improves the efficiency of finding frequent 2-itemsets by adopting a Hash technology, and this method also improves the process of creating candidate itemsets. DIC, proposed by Brin et al. [24] can append candidate itemsets dynamically in different courses of scanning database. The above algorithms are all have an advantage over Apriori, but they still spend a great deal of time scanning database, finding and testing candidate itemsets.

Éclat (Equivalence CLASS Transformation) algorithm [25] uses a vertical data representation and divide-and-conquer approach, the benefits of using the vertical representation optimize the parallel processing of the search space using depth-first generation of frequent itemsets. Éclat is the first considered algorithm used to generate frequent itemsets in vertical format using only once passes over the database; although, a vertical data representation usually reside in memory. It does not fully exploit the downward closure property of support as it does not utilize step (2) of Apriori, the pruning step; in addition, it generates a candidate (k+l)-itemset if two of its k-subsets are frequent, resulting in a larger number of candidates compared to Apriori [26].

As an alternative solution to the bottleneck of Apriori and Apriori-like algorithms problem of candidate set generation and testing, Han et al [27], proposed the pattern-growth approach for mining frequent patterns. One of the fastest mining methods for frequent itemsets is the well-known FP-growth algorithm [27]. FP-growth utilizes an effective and compact data structure known as Frequent Pattern tree (FP-tree). Unlike Apriori algorithm which uses candidate generation and testing; FP-growth performs pattern growth approach in order to obtain frequent itemsets. FP-growth divides the compact database into set of conditional databases, then all frequent itemsets are generated from the conditional databases. FP-tree is constructed by firstly building a header table. The header table contains item name as well as a corresponding link to each item. Link entries in the header table were initialized to null. Then, every item added first time to the tree, its corresponding entry is updated. Also the root node, marked as "null", is constructed. Child nodes are then attached through database scan. Paths that share same prefixes are searched firstly. If a path has same prefix of a transaction items exist, then the shared prefix part count is increased by one in the FP tree; the rest of the items that don't belong to the shared prefix are then attached to the last node in the 1st order and their counters are set to one. On the other hand if the transaction items don't share prefix part with any path; then they are attached to the root node in the 1st order. Lastly, every path in the FP-tree corresponds to a transaction in the database.

Based on FP-growth, many people have proposed their improved algorithms [28-32], Jian Pei and Jiawei Han [33] proposed the H-mine algorithm which has high performance and very small space overhead by taking the advantage of H-

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6591
***Article Received**: 24 February 2023 **Revised**: 25 March 2023 **Accepted**: 12 April 2023*

struct data structure and re-adjusting the links at mining different "projected" databases. Yahan Hu et al. [34] proposed the MIS-tree structure which is a FP-tree-like structure, and they also proposed a high performance algorithm called CFP-growth to mining association rules with multiple minimum supports. Pei et al. [35] developed a H-mine algorithm, that is used to discover all the frequent itemsets from the given transactional database. The proposed algorithm uses a simple and novel data structure hyper-link, H-struct, and a new mining algorithm, H-mine dynamically able to adjust links in the mining task, taking into account the advantages of the previously mentioned data structure. H-mine algorithm can have a scaled up to very large database using database partitioning, moreover, one of the distinct feature of H-mine is a very limited memory cost.
Apriori and FP-growth both have limitations, especially when the number of attributes is very high and the minimum support degree is very low. Based on the analysis of the advantages and disadvantages of existing algorithms, we propose an efficient algorithm known as Bitwise-Based data structure and algorithm for frequent pattern and association rule mining, compared to Apriori-like algorithm, FP-tree algorithm, m+ and twice respectively, Bitwise-Based technique, scans database only once, doesn't need to generate candidate itemsets, and all FCIs can be found by creating a comparative mask from the frequent 1-itemset. Bitwise-Based algorithm performance better on mining association rules with low minimum support degree from databases with a large number of attributes and large number of transactions [36-37]

The rest of this paper is organized as follows. The problem statement given in Section 3. We describe the proposed algorithm in section 4. In section 5, an empirical evaluation of our approach using synthetic and real dataset are presented. Finally, conclusion is presented in Section 6.

III. PROBLEM STATEMENT

In [11] the problem of mining association rules between a set of database transactions introduced as follows: Let I = {i1, i2, in} be a set of n literal or binary attributes called an items. Let D = {t1, t2. . . tn} be a set of transactions called the database of transactions. Each transaction in D has a unique identifier transaction ID and contains a subset of the items in I. A rule is defined as an implication of the form $X \rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \phi$. The problem is to: -

- Find all the frequent itemsets in database *D*, items which pass the condition of support that is greater than or equal to the predefined minimum threshold.
- Use the frequent items to generate the possible combination and association between a set of database items, after passing the condition of confident greater than or equal to predefined minimum confidence.

IV. MATERIAL AND METHOD

The algorithm used contains four procedures. The first procedure is *getItemSet*, which gets the list of items in the database as well as generate the binary representation of the database. The second procedure is *getFrequentItems* which scan the database and counts the frequent of all items in the item set and keeps those items whose support is greater than or equal minimum support in the frequent item list. The third procedure is *generateFrequentItemSets* which generates the lists of all frequent itemsets that passed the value of the minimum support. Finally the fourth procedure is *generateAssociations*, which generates all the association rules based on the frequent itemsets that has been generated in the previous procedure.

The algorithm use an empty bitSet of size itemSet list. We will use this empty bitSet as a mask. Firstly, we start by setting the bit that corresponds to the first item in the frequent items list, frequentItems. Then we loop through all the remaining items in frequentItems setting the corresponding bit of each one at a time together with the first one that is already set before. This means that at each time we would have two bits set in the mask. Then we **AND** the mask with all bitTransactions in BitSetDb. If the result of the **AND** equals the mask then items appeared in the transaction and their support is increased by one. We do this for all remaining combinations. If the support of paired items is greater than or equal the minsup they are added to the frequent hash table, FreqHT. Once finished generating associations of two items, we record the size of the hash table. Now we repeat the same steps but for three items. We set the bits that correspond to the first two frequent items and alternate setting the remaining items each one at a time. This means that we would have three bits set correspond to three items in the most frequent items list. Each time the mask is **ANDED** as before with all bitTransactions In BitSetDB and we get the support of all combinations of size three. Before start getting combination of length four, we check the size of the FreqHT if the size is same as the size recorded before this means that no frequent combination of size three are found. Thus we directly stop here and display the frequent items of length two as well as their associations. If the size grows then we proceed to find associations of four items, and the process is repeated again.

Consider an example contains a database transaction in Table 3.8. There are 6 items with 5 transactions in the database with TIDs 100, 200… 500. In this example we want to discover the frequent itemsets that satisfy the minimum support count 3.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6591
***Article Received**: 24 February 2023 **Revised**: 25 March 2023 **Accepted**: 12 April 2023*

TABLE 1: SAMPLE OF DATABASE TRANSACTIONS

TID	ITEMST	USED		
100	Bread	Milk	Tissue	
200	Bread	Tissue	Juice	Eggs
300	Milk	Tissue	Juice	Yoghurt
400	Bread	Milk	Tissue	Juice
500	Bread	Milk	Tissue	Yoghurt

Based on the first procedure, the algorithm gets all the items in the database and store them in the list of itemSet. The algorithm generates the corresponding binary database which contains the binary representation of the transactions that is recorded and stored as either 0 or 1. If the database transactions in the binary representation contains 1, means that the item is present in the rule. If the item appeared in the transaction contains 0, which means the items is not present in the rule, table 2.

TABLE 2. BITWISE-BASED DATA REPRESENTATION

ITEMS	Bread	Milk	Tissue	Juice	Eggs	Yoghurt
100	1	1	1	0	0	0
200	1	0	1	1	1	0
300	0	1	1	1	0	1
400	1	1	1	1	0	0
500	1	1	1	0	0	1

During the database transformation we simultaneously calculates the frequency of all items in the itemSet and store the frequency of each item in its corresponding index in array Freq. The procedure starts by scanning the binary database BitSetDB. The current binary transaction is retrieved and stored in a temporary BitSet (b). Then we check the bits of b. If a bit is set to 1, i.e. meaning that its corresponding item appears in this transaction, then its equivalent index in array Freq is increment by one. Once this step finishes then, the algorithm iterates through all elements of array Freq and if the count of an element passes the specified minimum support threshold, then the index of the element, which corresponds to an item in the itemSet, is added to the frequentItems list. This step can be illustrated in table. 3.

TABLE 3: THE FIRST STEP PROCESS

Items	Juice	Bread	Yoghurt	Tissue	Milk	Eggs
Freq	3	4	2	5	4	1
Support	0.6	0.8	0.4	1.0	0.8	0.2

From the first step output, we proceed to discover the frequent itemset of size 2. The important point is that, we have to check if the output of the first part contains frequent itemsets or not. If frequent itemsets that is can be used in the second step were found, then the procedure continues to the second part and calculates frequent itemsets of size ≥2. If no frequent itemsets that is can be used to discover the frequent itemsets of size 2 were found in the first part, then the procedure terminates here and return immediately, the algorithm stops here reporting no frequent itemsets were found.

To calculate the frequent itemsets of size 2, we use a mask to do a combination of itemsets of size two. This is done using an empty bitSet of size itemSet, which will be represented as a mask. Table 4. From the previous step we can design our mask based on the output of table 3.

TABLE 4: MASK BITSET REPRESENTATION

ITEMS	Bread	Milk	Tissue	Juice	Eggs	Yoghurt	Support
Mask	1			1			2

When we AND the mask with the binary BitSetDB the mask will match two occurrences in the database transactions Table.5.

TABLE 5: ITEMSETS OCCURRENCE COUNT

ITEMS	Bread	Milk	Tissue	Juice	Eggs	Yoghurt	Occurrences
100	1	1	1	0	0	0	x
200	1	0	1	1	1	0	√
300	0	1	1	1	0	1	x
400	1	1	1	1	0	0	√
500	1	1	1	0	0	1	x

As it shown in the first AND between the mask BitSet and BitSetDB the two items Bread and Juice occur together two times in the whole transactions, Table 5.

Then we will continue to the remaining combination of the other 3 probabilities as shown in table.6.

TABLE 6: REMAINING COMBINATION OF ITEMSET >=2

ITEMS Mask	Bread	Milk	Tissue	Juice	Eggs	Yoghurt	Support
			1	1			3

ITEMS Mask	Bread	Milk	Tissue	Juice	Eggs	Yoghurt	Support
		1		1			2

ITEMS Mask	Bread	Milk	Tissue	Juice	Eggs	Yoghurt	Support
	1		1				4

ITEMS Mask	Bread	Milk	Tissue	Juice	Eggs	Yoghurt	Support
	1	1					3

ITEMS Mask	Bread	Milk	Tissue	Juice	Eggs	Yoghurt	Support
		1	1				4

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6591
***Article Received**: 24 February 2023 **Revised**: 25 March 2023 **Accepted**: 12 April 2023*

Based on the previous step we will get the following output shown in table 7.

TABLE 7. FREQUENT ITEMSETS OF SIZE <=2

Itemsets	Juice Tissue	Bread Tissue	Bread Milk	Tissue Milk
Freq	3	4	3	4

Again we start masking the remaining part which can be done for the itemsets > two combinations as shown in Table.8.

TABLE.8. COMBINATION OF ITEMSETS > TWO ITEMSETS

ITEMS Mask	Bread	Milk	Tissue	Juice	Eggs	Yoghurt	Support
	1		1	1			2

ITEMS Mask	Bread	Milk	Tissue	Juice	Eggs	Yoghurt	Support
	1	1	1				3

Based on the previous step the final maximal frequent itemsets will be the itemsets Bread, Milk, and Tissue as shown in table.9.

TABLE.9. THE MAXIMAL FREQUENT ITEMSETS

Itemsets	Bread Milk Tissue
Freq	3

Now the frequent itemsets for the full transactions are presented in Table.10.

TABLE 10: THE FREQUENT ITEMSETS AND THEIR SUPPORTS

Itemsets	Support
Bread	4
Milk	4
Tissue	5
Juice	3
Bread Milk	3
Bread Tissue	4
Milk Tissue	4
Tissue Juice	3
Bread Milk Tissue	3

V. EXPERIMENTAL RESULT

To proof the efficiency of Bitwise-Based algorithm, the experiments were conducted on Intel® corei5™ CPU, 2.4 GHz, and 02 GB of RAM computer. Graph 1 and graph 2 in figure (1) and figure (2) respectively shows the comparison of the execution time for mining frequent patterns using synthetic dataset T20I6D100K provided by the QUEST generator from IBM's Almaden research lab, and the real dataset, Mushroom, that's publicly available in the FIMI dataset repository. Mentioned datasets are having different transaction size, item size, and other behaviors. The graph x axis show the support percentage value specified by the user, and y axis are carrying an execution time in milliseconds.

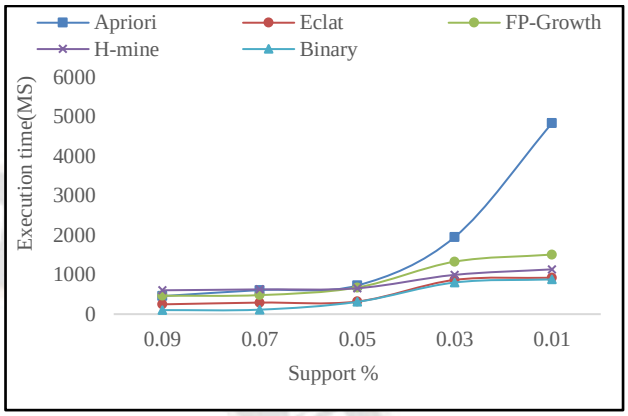

Figure (1) Execution time using T20I6D100K dataset

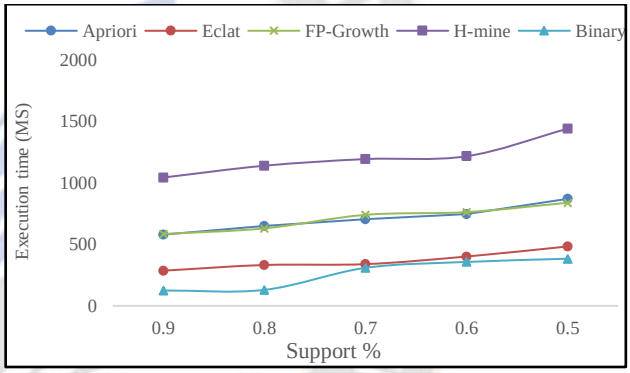

Figure (2) Execution time using Mushroom dataset

In figure (1) The execution time of the all mentioned algorithms make a nearby sense, except Apriori algorithm, the reason of that Apriori use the candidate generating and testing approach that face the problem in execution time when the support value decreased. In figure (2) Bitwise-Based algorithm outperforms all other algorithms, the reason is that Bitwise-Based algorithm use the vertical data layout that is efficiently calculate the item support of the frequent pattern, that lead to affect the execution time. The benefits of using the vertical layout also appear in Éclat which is come in the second order of execution time out of all other algorithms.

Significant of Execution Time Benchmarking A student t-test has been conducted using Matlab 2012b to verify the significant of the obtained results of execution times. The t-test has reported a significant reduction in execution times when using Bitwise-Based approach (mean=657.2 and standard deviation=382.83) against the best results recorded by Eclat algorithm (mean=1188 and standard deviation=259.36) in T20I6D100K dataset which contains 1000 of items and 100000

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6591
***Article Received**: 24 February 2023 **Revised**: 25 March 2023 **Accepted**: 12 April 2023*

database transactions, using α=0.05. Table 11 summarizes the obtained t-test results for T20I6D100K dataset. Here the t column refers to the generated t-test value, while the p column refers to the probability of the t value of the t-test. The significance of the t-test depends on both α and p. If the value of p is less than the value of α then the t-test reports a significant result and hence the h column which refers to the test hypothesis will be 1 which means rejecting the Null hypothesis and accepting the alternative hypothesis; otherwise h will be 0 which means non-significant result.

TABLE 11: T-TEST RESULTS FOR T20I6D100K DATASET (A=0.05)

Test Sample	Bitwise-Based		Éclat		*t*	*p*	*h*
T20I6D100K dataset	$\bar{x}$	*sd*	$\bar{x}$	*sd*			
	657.2	382.83	1188	259.36	-2.5667	0.0370	1

The student t-test has also been conducted to verify the significant of the obtained results of execution times in Retail dataset. The t-test has reported a significant reduction in execution times when using Bitwise-Based approach (mean=790 and standard deviation=232.59) against the best results recorded by Éclat algorithm (mean=1360 and standard deviation=289.65) using α=0.05. Table 4.8 summarizes the obtained t-test results for Retail dataset.

TABLE T-TEST RESULTS FOR RETAIL DATASET (A=0.01)

Test Sample	Bitwise-Based		Eclat		t	p	h
Retail dataset	$\bar{x}$	Sd	$\bar{x}$	sd			
	790	232.59	1360	289.65	-3.4310	0.0096	1

VI. CONCLUSIONS

This paper overall, uses a powerful technique for identifying patterns of co-occurrence among items in transactional data (MBA) for generating insights that can help businesses improve their sales and marketing strategies. We introduce an efficient Bitwise-Based data structure technique for mining frequent pattern in large-scale databases, the algorithm scans the original database once, using the Bitwise based data representations as well as vertical database layout, compared to the well-known Éclat and FP-Growth algorithm, Bitwise based technique enhances the problems of multiple passes over the original database, Hence, minimize the execution time.

REFERENCES

[1] P. B. Jensen, L. J. Jensen, and S. Brunak, "Mining electronic health records: towards better research applications and clinical care," *Nature Reviews Genetics*, vol. 13, pp. 395-405, 2012.

[2] R. Gupta, "Analysis and design of data mining techniques for prevention and detection of financial frauds," 2013.

[3] A. Bansal and M. R. Rastogi, "LEARNING BEHAVIOR OF ANALYSIS OF HIGHER STUDIES USING DATA MINING," International Journal of Advanced Research in Computer Engineering & Technology (IJAR*CET)*, vol. 1, pp. pp: 80-84, 2012.

[4] S. U. Kumar, H. H. Inbarani, and S. S. Kumar, "Bijective soft set based classification of medical data," in *Pattern Recognition, Informatics and Medical Engineering (PRIME), 2013 International Conference on*, 2013, pp. 517-521.

[5] R. Al Iqbal, "Hybrid clinical decision support system: An automated diagnostic system for rural Bangladesh," in *Informatics, Electronics & Vision (ICIEV), 2012 International Conference on*, 2012, pp. 76-81.

[6] B. Milovic, "Prediction and decision making in Health Care using Data Mining," *International Journal of Public Health Science (IJPHS)*, vol. 1, pp. 69-78, 2012.

[7] M. V. Joseph, "Data Mining and Business Intelligence Applications in Telecommunication Industry," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 2, pp. 525-528, 2013.

[8] R. Sujatha and D. Ezhilmaran, "A Proposal for Analysis of Crime Based on Socio–Economic Impact using Data Mining Techniques," *International Journal of Societal Applications of Computer Science*, vol. 2, pp. 229-231, 2013.

[9] A. Chauhan, G. Mishra, and G. Kumar, *Survey on Data mining Techniques in Intrusion Detection*: Lap Lambert Academic Publ, 2012.

[10] S. O. Fageeri, R. Ahmad, and B. Baharum, "A Log File Analysis Technique Using Binary-Based Approach," in *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, 2014, pp. 3-11.

[11] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, 1993, pp. 207-216.

[12] M. M. Mazid, A. Shawkat Ali, and K. S. Tickle, "Finding a unique association rule mining algorithm based on data characteristics," in *Electrical and Computer Engineering, 2008. ICECE 2008. International Conference on*, 2008, pp. 902-908.

[13] H. Toivonen, "Sampling large databases for association rules," in *VLDB*, 1996, pp. 134-145.

[14] H. Cheng, X. Yan, and J. Han, "IncSpan: incremental mining of sequential patterns in large database," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 527-532.

[15] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," in *ACM SIGMOD Record*, 1997, pp. 255-264.

[16] J. S. Park, M.-S. Chen, and P. S. Yu, "Using a hash-based method with transaction trimming for mining association rules," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 9, pp. 813-825, 1997.

[17] J. S. Park, M.-S. Chen, and P. S. Yu, "Efficient parallel data mining for association rules," in *Proceedings of the fourth*

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6591
***Article Received**: 24 February 2023 **Revised**: 25 March 2023 **Accepted**: 12 April 2023*

international conference on Information and knowledge management, 1995, pp. 31-36.

[18] R. Agrawal and J. C. Shafer, "Parallel mining of association rules," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 8, pp. 962-969, 1996.

[19] D. W. Cheung, J. Han, V. T. Ng, A. W. Fu, and Y. Fu, "A fast distributed algorithm for mining association rules," in *Parallel and Distributed Information Systems, 1996., Fourth International Conference on*, 1996, pp. 31-42.

[20] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, "Parallel algorithms for discovery of association rules," *Data mining and knowledge discovery*, vol. 1, pp. 343-373, 1997.

[21] A. Savasere, E. R. Omiecinski, and S. B. Navathe, "An efficient algorithm for mining association rules in large databases," 1995.

[22] F. Geerts, B. Goethals, and J. Van den Bussche, "A tight upper bound on the number of candidate patterns," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, 2001, pp. 155-162.

[23] J. S. Park, M.-S. Chen, and P. S. Yu, *An effective hash-based algorithm for mining association rules* vol. 24: ACM, 1995.

[24] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," in *ACM SIGMOD Record*, 1997, pp. 255-264.

[25] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, "New Algorithms for Fast Discovery of Association Rules," in *KDD*, 1997, pp. 283-286.

[26] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, 1994, pp. 487-499.

[27] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *ACM SIGMOD Record*, 2000, pp. 1-12.

[28] F.-y. DENG and Z.-y. LIU, "(Dept. of Management Science, Xiamen University, Xiamen 361005, China); An Ameliorating FP-growth Algorithm Based on Patterns-matrix [J]," *Journal of Xiamen University (Natural Science)*, vol. 5, 2005.

[29] Z. Y. LüHongbing, "An Incremental Updating Algorithm to Mine Association Rules Based on Frequent Pattern Growth [J]," *Computer Engineering and Applications*, vol. 26, p. 055, 2004.

[30] K. Wang, L. Tang, J. Han, and J. Liu, *Top down fp-growth for association rule mining*: Springer, 2002.

[31] Y. Qiu, Y.-J. Lan, and Q.-S. Xie, "An improved algorithm of mining from FP-tree," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, 2004, pp. 1665-1670.

[32] A. Pietracaprina, "Mining frequent itemsets using patricia tries," 2003.

[33] J.-W. Han, J. Pei, and X.-F. Yan, "From sequential pattern mining to structured pattern mining: a pattern-growth approach," *Journal of Computer Science and Technology*, vol. 19, pp. 257-279, 2004.

[34] Y.-H. Hu and Y.-L. Chen, "Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism," *Decision Support Systems*, vol. 42, pp. 1-24, 2006.

[35] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang, "H-mine: Hyper-structure mining of frequent patterns in large databases," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, 2001, pp. 441-448.

[36] Fageeri, S.O., Hossain, S.M.E., Arockiasamy, S., Al-Salmi, T.Y. (2022). High-Utility Pattern Mining Using ULB-Miner. In: Aurelia, S., Hiremath, S.S., Subramanian, K., Biswas, S.K. (eds) Sustainable Advanced Computing. Lecture Notes in Electrical Engineering, vol 840.

[37] Fageeri, S., Ahmad, R., Alhussian, H. (2020). An Efficient Algorithm for Mining Frequent Itemsets and Association Rules. In: Subair, S., Thron, C. (eds) Implementations and Applications of Machine Learning. Studies in Computational Intelligence, vol 782. Springer, Cham