Journal of Theoretical and Applied Information Technology
15th May 2023. Vol.101. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN ENHANCED DYNAMIC ALGORITHM FOR ASSOCIATION RULES MINING

[1] ALA IBRAHIM [2] MOHAMMAD SHKOUKANI

[1] Department of Computer Science, Applied Science Private University, Amman, Jordan

[2] Prof., Department of Computer Science, Applied Science Private University, Amman, Jordan

E-mail: [1]ala.emad.ibrahim@gmail.com, [2] m.shkokani@asu.edu.jo

ABSTRACT

The data mining techniques have been applied to various fields. Frequent pattern mining is one of the active research themes in data mining. It has an important role in all data mining tasks such as clustering, classification, prediction, and association analysis. A frequent pattern is the most time-consuming process due to the massive number of patterns generated. Frequent patterns are generated by using association rule mining algorithms that use candidate generation and association rules such as the Apriori algorithm. Researchers showed an extraordinary enthusiasm for information mining. Likewise, industrial companies call for different information mining methods to better comprehend client behavior, enhance the service provided and expand business revenues. Due to the huge size of data that exists in databases and warehouses, also because these data are big, dynamic, and change frequently. It is hard and costly to perform mining for frequent patterns and association rules from scratch every time. This paper explores three algorithms. The first one is the classic Apriori algorithm which finds frequent patterns and association rules from scratch every time it works; any update on data will enforce the algorithm to repeat the full process. The second algorithm is the dynamic algorithm which finds frequent item sets by using the accumulated knowledge stored in a database table. This method takes less processing time and computations to find frequent patterns from the data than the classic Apriori. The third algorithm is an enhanced algorithm that we have developed. It includes an indexing technique that helps to discover frequent patterns quickly. The three algorithms were tested on a dataset containing 127936 transactions that contain the items of the market that have been purchased with 207 different types of these market items. The results initially showed that the classical Apriori algorithm was better than the dynamic algorithm and an enhanced algorithm. However, our enhanced algorithm was 78.54% better than the classic Apriori and 27.49% than dynamic algorithm in frequent runs. The results were 169.93, 49.53, and 37.18 seconds for the classic Apriori, the dynamic, and the enhanced algorithms, respectively.

Keywords: *Big data, Data mining, Association rules, Indexing*

1. INTRODUCTION

Data mining has recently become one of the most progressive and promising fields for the extraction and manipulation of data to produce useful information. Undoubtedly, the research and utility of data mining are very challenging [1]. Data mining must go via idea presentation, idea acceptance, substantial research and exploration, gradual application, and mass application ranges like growing other new technologies, so the researcher should make it useful by analyzing and processing it to extract and discover the unknown useful information, which is known with Data mining [2]. Data Mining is a process to discover the added value of information that has not been recognized manually from a database through extracting patterns of data to manipulate data into information obtained by using extracting and recognizing the necessary or interesting patterns of records that are in the database. Data is a representation of a real match that represents an object like a human [3]. Big Data is a modern buzzword that is frequently used to describe one-of-a-kind sides of Data Analytics and Data Mining. Big data is also data that contains greater variety, arriving in increasing volumes and with more velocity. This is also known as the three Vs [2]. Massive data nowadays may also not meet the threshold in the future due to the fact storage capacities will increase, allowing even bigger data sets to be captured. Other elements such as the area or industry the data is from, as well as the kind of data (images, audio, text), play a role in defining high-volume data [4].So, when speaking about mining big data there are a few sequential steps to follow, such as performing data organizing, then facts integration, after that data analysis, and subsequently comes the decision making [5]. The aim of using data mining is to develop precise knowledge of the hidden relations between items to support decisions and expand revenue, where these relations are known as association rules. the rules can be defined as implication statements, which uncover associations between evidently irregular data in a database, that contains two parts; first, Antecedent (if) that is an item found in the data, secondly the consequent (then), which is an item that is found with the Antecedent together. Association rules are built by researching data for frequent patterns and using such criteria to distinguish the most essential and valuable connections [6]. In data mining, association rules are useful for analyzing and predicting customer behavior. They play an important part in customer analytics, market basket analysis, product clustering, catalog design, and store layout. Due to the huge size of data that exists in databases and warehouses, also because these data are big, dynamic, and change frequently. It is hard and costly to perform mining for frequent patterns and association rules from scratch every time [7].

The main objective of this research is to answer the following two main questions:

1. How to implement a dynamic algorithm without the need to perform the Apriori algorithms from scratch?
2. Can the Indexing technique improve the dynamic algorithm? and how effective is it compared with the dynamic algorithm?

After that, the authors implemented the proposed model which is enhanced algorithm on a real data set by using an indexing technique to improve the efficiency of dynamic algorithm.

2. PROBLEM STATEMENT

In data mining, association rules are useful for analyzing and predicting customer behavior. They play an important part in customer analytics, market basket analysis, product clustering, catalog design, and store layout. Due to the huge size of data that exists in databases and warehouses, also because these data are big, dynamic, and change frequently. It is hard and costly to perform mining for frequent patterns and association rules from scratch every time.

The main problem is that mining for frequent patterns in association rules works from scratch each time, it takes effort to implement algorithms such as the classic Apriori algorithm, any update of data like adding a new transaction will implement the algorithm to operate the full process.

In this research paper the authors proposed an algorithm to generate frequent item-sets dynamically without the need to do the Apriori algorithm from scratch. They also added a clustering index to increase performance and to help discovering frequent patterns quickly.

3. RELATED WORKS

In this section, we will give some related work: Gunaseelan and Uma discussed the most important principles of data mining after that delivered a new enhancement over the Apriori algorithm by decreasing the wide variety of scanning through the transposed database DT and reducing redundancy by way of the time of producing sub-transaction set tests and checking them in the database. To locate everyday patterns in large datasets with extra columns than rows, she introduced a entire shape for the transposition. Finally, they proved that their proposed algorithm always beats the Apriori algorithm. Hence, their proposed algorithm is practical to use in large datasets to reduce the size of the database [8]. Jie Gao mentioned proposes an improved association rule mining algorithm based on FP-tree and implements the algorithm with the aid of dividing consumer get admission to data into individual sessions in accordance to the customized advice step of association rules [9]. Aldaher and Shkoukani discussed the result in contrast to their proposed dynamic algorithm and Apriori algorithm and proved that the proposed dynamic algorithm is better mainly in the universal runs and updated data due to the fact it makes use of amassed expertise rather than working from scratch each time like the Apriori algorithm that fails due to the fact each time it will generate item sets and tables which costs a lot of effort, time, and storage [10]. Kanhere et al presented in this paper an algorithm-based approach with the aid of lowering the dataset dimension as a solution. The dataset used to decrease the use of 55% of top-selling products to achieve an identical set of rules via identifying frequency and price as parameters. This paper's method goes beyond current published lookup and efficiently reduces the dataset with top revenue-generating merchandise which are simply 7% of the performance compared

Journal of Theoretical and Applied Information Technology
15th May 2023. Vol.101. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

with the Apriori algorithm [11]. Zhao et al discussed that the proposed approach through mining successfully controls the noise data, and the precision price is also saved very high, which shows the best possible accuracy of the technique. This article makes a systematic and certain evaluation of facts mining technological know-how via the usage of the Apriori algorithm and makes use of it to compress the size and decrease the scanned transaction database dimension by discount of needless transactions, to enhance the efficiency of mining, they can delete these transactions because they are no longer needed when scanning the database [12]. Jha et al proposed a new improved Apriori algorithm with the most important motive of decreasing the time and wide variety of scans required to become aware of the standard itemset and affiliation rules among schooling facts the usage of a bottom-up approach. Moreover, the proposed algorithm also replaces the support minimum threshold with a popular deviation-based practical mannequin as bringing up the minimum guide fee in increase may lead to an influence on the performance [13]. Dharmaraajan et categorized user conduct in figuring out the patterns of browsing using The Apriori algorithm and FP Growth algorithms are in contrast by means of applying the fast miner tool to find out ordinary user patterns along with user behavior in the weblog. Both algorithms assist to analyze the patterns of website usage and the facets of user behavior understanding bought from net usage. The result indicates that the FP increase algorithm gives a higher performance in phrases of time execution [14]. Cong established the association rule mining with the aid of combining the genetic algorithm with the Apriori algorithm. Compared with different methods, the experimental effects exhibit that the proposed technique has better performance, higher mining efficiency, and higher data mining [15]. Essam et improved conditional pattern base their method ambitions to eliminate infrequent items from the conditional pattern base earlier than constructing nearby FP-Trees [16]. Baohua Zhang discussed the optimization of the FP-Growth algorithm below cloud computing and computer large data, the experimental outcomes in this paper exhibit that the performance of the expanded FP-growth algorithm is better than the original algorithm, and the traversal time is reduced [17].

Finally, each researcher has used various methods and techniques like dynamic algorithms, cloud computing for big data, reducing frequent item sets for reducing storage space, and combining genetic algorithm with Apriori algorithm, however regrettably no longer discovered anybody who examined the indexing technique with clustered index in the dynamic algorithm as we used in our enhanced algorithm.

Table 1: Summary Of Related Works For Association Rule Techniques

Authors	Percentage of Improvement
Gunaseelan and Uma [8]	33%
Jie Gao [9]	10%
Aldaher and Shkoukani [10]	63%
Zhenyi Zhao, Zhou Jian [11]	14%
S. Kanhere, A. Sahni [12]	7%
Jha, Jayshree and Leena [13]	No execution
K.Dharmarajan [14]	18%
Y. Cong [15]	48%
A.Essam, M.Abdel-Fattah [16]	25%
Baohua Zhang [17]	13%

Table 1 shows the percentage of improvement in execution times according to the proposal models of some related works.

4. THE PROPOSED MODEL

To achieve the objectives of the study, we divided the proposed model into four sub algorithms which are the clustered indexing technique, Initial state-building algorithm (ISB), Frequent Used Algorithm (FU), and Rule generation algorithm (RG). The proposed algorithm performs its task by executing its sub algorithms as shown in figure 1.

Journal of Theoretical and Applied Information Technology
15th May 2023. Vol.101. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

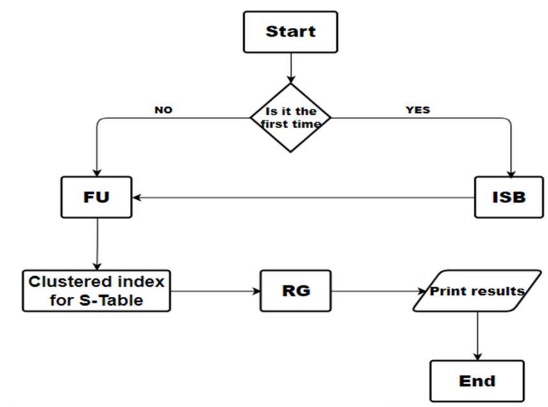


Figure 1: Flow Chart Of The Proposed Model

A. Initial state building algorithm (ISB):

This algorithm is responsible for building the initial state when using the system for the first time and it works, as shown in tables 2 and 3.
Stable: is the summation table, it consists of three columns: item, summation, and size, where: • Item is all the candidate item sets that can be generated of market basket items • Summation is the count of this item-set, in other words, is the support of this item-set • Size: is the number of items in the item-set Algorithm: • Read all items in the database and display them to the user • Generate all possible combinations of item-sets • For each item-set in generated item-sets •Insert item-set in Stable as (item-set, summation =0, size of item-set).

Table 2: Ttable

TID	A	B	C	flag
1	0	1	0	F
2	3	5	1	F
3	1	0	1	F
4	0	0	2	F
5	0	1	0	F
6	1	4	0	F
7	0	0	3	F
8	2	0	1	F
9	0	1	1	F
10	1	1	0	F

Table 3: Stable after ISB algorithm

ITEM	SUMMATION	SIZE
ABC	0	3
AB	0	2
AC	0	2
BC	0	2
B	0	1
A	0	1
C	0	1

B. Frequent Used Algorithm (FU):

This algorithm is responsible for building the updated state when using the system frequently and it works, as shown in tables 4 and 5.
Ttable: is a horizontal representation of real basket market transactions, that contains all the items of the market and flag as columns and the transactions as rows; 0 represents that this item isn't found in this transaction, and any number greater than or equals 1 represents that this item is found in this transaction. If flag is f then the transaction was not counted else it is already counted and there is no need to read again.
Algorithm
• For each item-set in Stable
• If item-set size == 1
• Count all rows where item value is >=1 and flag = f in Ttable
• update summation of item in Stable (summation += count)/total number of transactions
• else
• Count all the rows where the value of all items of the item-set > =1 and flag = f in Ttable
•Update summation of item-set in Stable (summation += count)/total transactions
• Set flag for all the rows of Ttable = t

Table 4: Ttable After FU Algorithm

TID	A	B	C	flag
1	0	1	0	T
2	3	5	1	T
3	1	0	1	T
4	0	0	2	T
5	0	1	0	T
6	1	4	0	T
7	0	0	3	T
8	2	0	1	T
9	0	1	1	T
10	1	1	0	T

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5: Stable After FU Algorithm

ITEM	SUMMATION	SIZE
ABC	0.1	3
AB	0.3	2
AC	0.3	2
BC	0.2	2
B	0.6	1
A	0.5	1
C	0.6	1

C. Indexing Technique:

In this research, we proposed an improved frequent pattern mining algorithm with indexing. This algorithm makes effective use of indexing for generating large frequent patterns from the database transactions [7]. The indexing technique helps to reduce passes of transaction database scans the number of candidates [18]. In our proposed model, we used the clustered index technique for Stable (summation table) specifically summation column, which contain all possibilities support of item sets. D. Rule Generation Algorithm (RG): After the frequent item-sets have been discovered, it becomes simple and clear to generate association rules from them as shown in table 6. Rtable: is the association rules table, it consists of four columns left, right, support and confidence, where:

• Left is an item -set that is on the left-hand side of the rule

• Right is an item -set that is on the right-hand side of the rule

• Support is the count of this item-set of left U right
• Confidence: is the division result of support over support the left-hand side Algorithm:

• Clear Rtable

• For each item-set (I) in Stable have summation > = minimum support I is frequent

• For each frequent item-set (I) with size > 2 // because sets of size 1 can't generate association rules

• Generate all the subsets (S) of the item-set (I)

• For each S

• if support (I) / support (S) >= minimum confidence and minimum support

• insert in Rtable (left, right, support, confidence) values (S, I-S, support (I), support (I) / support (S)).

Table 6: Rtable after RG algorithm

LEFT	RIGHT	SUPPORT	CONFIDENCE
A	B	0.3	0.6
A	C	0.3	0.6

5. IMPLEMENTATION

The enhanced algorithm was implemented using.Net technology (windows forms) C# programing language and SQL language, the income of the implementation for how many transactions you need to use in our test, as shown in figure 2.

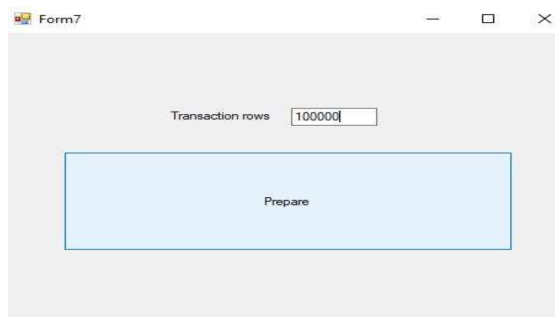
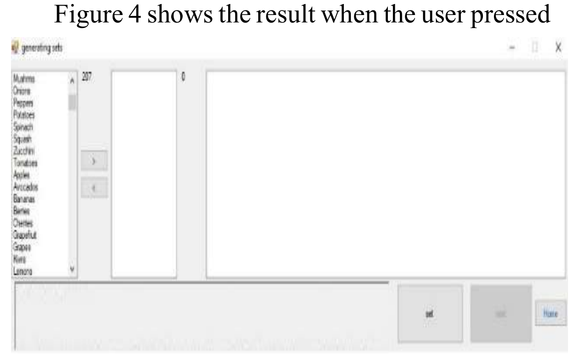


Figure 2: Number Of Transaction Screen

The software of our proposed model consists of 3 main screens (generating sets, counting sets, and generating rules) These screens represent the 3 algorithms that we described before (ISB, FU, and RG). Figure 3 shows the ISB algorithm screen; here user should select the groups of items using (<,> and set) buttons, the group contains at least two elements, then generates all subsets of the selected group and insert it in the database.

Figure 3: Generating sets screen

Figure 4 shows the result when the user pressed



set button, and the output is all the subsets with their sizes, after setting one group the user can proceed to

Journal of Theoretical and Applied Information Technology
15th May 2023. Vol.101. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the next step and finish the current one by pressing the next button.

Figure 4: Generating Sets Screen After Set Button.

Figure 5 shows how the Stable would be after the ISB algorithm runs, the table is ready to be processed each time the software works, and it will save the accumulated results.

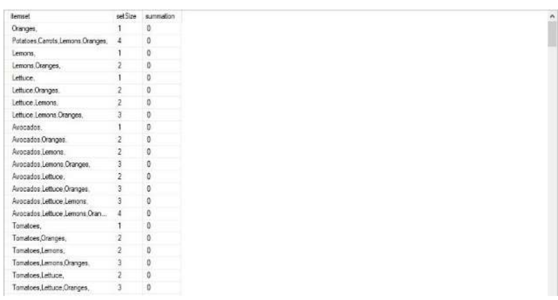


Figure 5: Stable After Set Button

Figure 6 displays all the items set and their counted transactions.

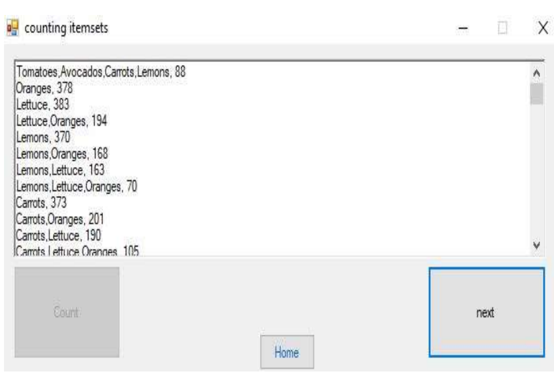


Figure 6: Results Of Counting Sets

Figure 7 shows the results in Stable after being updated when executing FU algorithm.

Figure 7: Stable After Counting.

Figure 8 shows how the user selected the rule generation parameters, so the user should input min_sup and min_conf and then press generate and the results will be then generated. after that the RG algorithm runs and check each set in S table with support >= min_sup and with size >2 for all found sets it starts generating subsets and checking if any of the rules greater than or equal min_sup & min_conf and can be generated then displays it in the screen and insert it in the R table.

Figure 8: Generated Results.

Figure 9 shows the results after executing RG algorithm.

ruleLeft	ruleRight	support	confidence
Lettuce	Tomatoes	0.33238	0.6202045
Lemons	Avocados	0.30362	0.5934831
Tomatoes	Carrots	0.30043	0.5600231
Lemons	Carrots	0.30087	0.5881076
Tomatoes	Avocados	0.33842	0.6308392
Potatoes	Oranges	0.30344	0.5703652
Avocados	Carrots	0.3073	0.5805562
Carrots	Lemons	0.30087	0.5751123
Avocados	Tomatoes	0.33842	0.6393486

Figure 9: Rtable After Generated Rules.

Now if the data was updated by adding new transaction, and the user started the software, the first thing that will be seen is figure 10 that asks the user if you want to perform the process from scratch or use the accumulated knowledge stored from the last run.

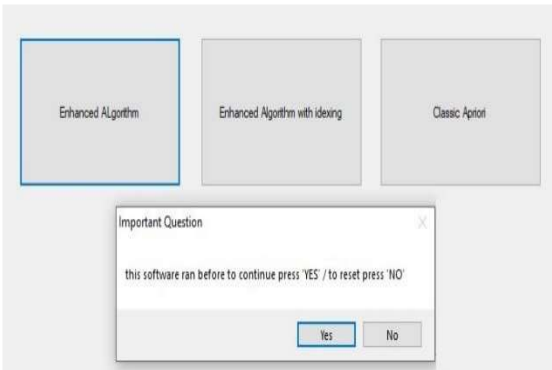


Figure10: Important Question For Frequent Runs

If user chooses yes it will read each item set, and check its size if the size is 1 it will count all the transactions in T table, where flag = f and the column name is the item name, and update the summation of the set , if the size is larger than 1 it will decompose the set into its basic items and count all the transactions in T table where flag = f and where all items found on that transaction, and update the summation of the set.

6. RESULTS AND DISCUSION

The experiments in this research were done using .Net technology (windows forms) C# programing language and SQL language. The experiments were carried out using a Windows 10.1 64-bit laptop with a Core-i7 processor, 2.30 GHz, and 16 GB of RAM. Figure 11 shows that the Classic Apriori algorithm needs 169.93 seconds to run 100000 transactions.

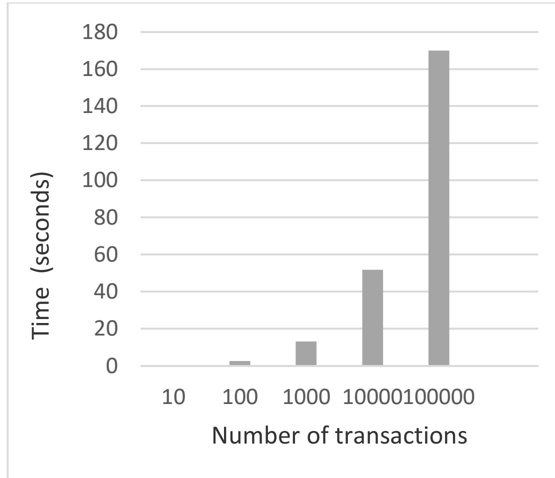


Figure 11: Results Of The Classic Apriori Algorithm

Figure 12 shows that the dynamic algorithm needs 49.53 seconds to run 100000 transactions

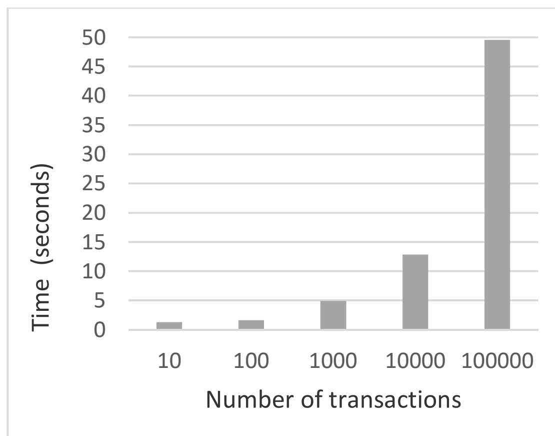


Figure 12: Results Of Dynamic Algorithm

Figure 13 shows that the proposed algorithm needs 37.18 seconds to run 100000 transactions.

Journal of Theoretical and Applied Information Technology
15th May 2023. Vol.101. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

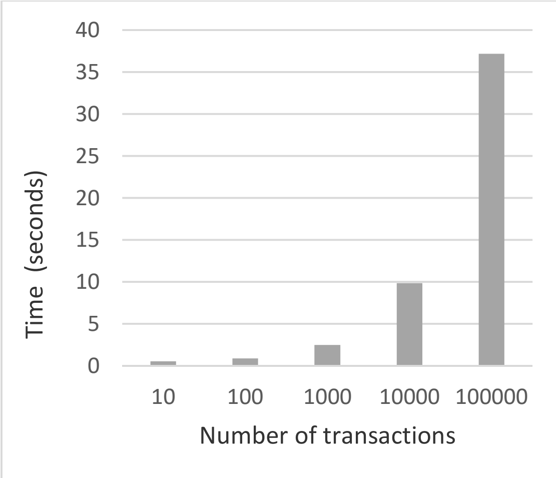


Figure 13: Results Of Proposed Algorithm

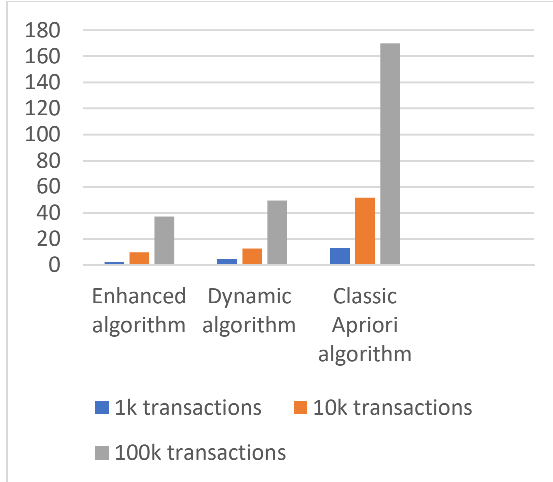


Figure 14: Comparison Between The Three Algorithms Results

Table 7: Classic Apriori Vs Dynamic Vs Proposed Algorithms

Number of transactions	Enhanced algorithm	Dynamic algorithm	Classic Apriori algorithm
10	0.54	1.31	0.06
100	0.88	1.6	2.54
1000	2.46	4.9	13.07
10000	9.84	12.86	51.65
100000	37.18	49.53	169.93
Average running time of each algorithm	10.18	14.04	47.45
Average of the Performance = (14.04-10.18) / 14.04 * 100 =		27.49%	
Average of the Performance = (47.45-10.18) / 47.45 * 100 =			78.54%

Table 7 shows the results of three algorithms which are Classic Apriori, dynamic, and proposed algorithms. Based on the results our proposed algorithm provided the best result.

Figure 14 shows the comparison between the three algorithms. It appears that our proposed algorithm is more efficient than others.

According to the results shown in table 7, which compares the three algorithms which are the proposed algorithm, dynamic algorithm, and the Classic Apriori algorithm. We found that in the initial implementation of the Apriori algorithm, it seems positive, however for dynamic uses, it fails because each time it will generate item sets and tables the cost of effort time. Figure 14 shows that our proposed algorithm is more efficient than other algorithms by tested 1k, 10k, and 100k transactions, so it took 2.46, 9.84, and 37.18 seconds respectively to complete them. On another hand, the dynamic algorithm took 4.9, 12.86, and 49.53 seconds respectively, and classic Apriori needed 13.07, 51.65, and 169.93 seconds respectively to execute 1k, 10k, and 100k transactions respectively. Also, we found that the proposed algorithm was 78.54% better in frequent runs with less processing time than the classic Apriori. Additionally, the proposed algorithm, which includes a clustered index, gave a 27.49% boost in performance than the dynamic algorithm.

7. CONCLUSION AND FUTURE WORK

The purpose of using the association rule mining algorithm is to find the meaning and relationship of large data sets. Therefore, the authors introduce the Apriori algorithm, and how an effective indexing technique can improve the dynamic algorithm. In this thesis, we suggested an enhanced algorithm for the extraction of association rules in big data, which aims to find useful relationships between unknown or hard to find items from a big database in an efficient way without the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

need to repeat all the steps from scratch; like what the classic algorithms do. Subsequently, we added an indexing technique (cluster index) that gave us a higher preference. As a result, the enhanced algorithm was 78.54% better in frequent runs with less processing time than the classic Apriori. Additionally, the enhanced algorithm, which includes a clustered index, gave a 27.49% boost in performance than the dynamic algorithm in frequent run. We conclude that this research is a seed in data mining and that it can be improved to be more efficient.

For future work in this area, it is hoped that some future work can determine the feasibility of the proposals below:

•Implementing sorting algorithms to enhance searching and counting methods.

•Provide support for different database management systems.

•A follow-up survey could be conducted between algorithms with a larger sample size.

8. ACKNOWLEDGMENTS

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the full financial support granted to this research.

REFERENCES:

[1] Provost, Foster and Tom Fawcett. (2013). Data Science for Business book: [what You Need to Know About Data Mining and Data-analytic Thinking]. Sebastopol, Calif., O'Reilly, pp. 05-15.

[2] Shynu, P. G, Shayan, H. M, & Chowdhary, C. L. (2020). A fuzzy based data perturbation technique for privacy preserved data mining. International Conference on Emerging Trends in Information Technology and Engineering (icETITE) (pp. 1-4).

[3] M. M. Rahman, Y. Watanobe, T. Matsumoto, R. U. Kiran and K. Nakamura. (2022). "Educational Data Mining to Support Programming Learning Using Problem-Solving Data," in IEEE Access, vol. 10, pp. 26186-26202.

[4] Yusuf Aytas. (2021). "An Introduction: What's a Modern Big Data Platform," Designing Big Data Platforms: How to Use, Deploy, and Maintain Big Data Systems, Wiley, pp.1-9.

[5] S.Agarwal. (2013). "Data Mining: Data Mining Concepts and Techniques," International Conference on Machine Intelligence and Research Advancement, pp. 203-207.

[6] Sotiris Kotsiantis, Dimitris Kanellopoulos. (2006). Association Rules Mining: A Recent Overview, GESTS International Transactions on Computer Science and Engineering, Vol.32 (1).

[7] H. Wu et al. (2017). "A Performance-Improved and Storage-Efficient Secondary Index for Big Data Processing," 2017 IEEE International Conference on Smart Cloud (SmartCloud), pp. 161-167.

[8] Gunaseelan, Uma. (2012). An Improved Frequent Pattern Algorithm for Mining Association Rules, International Journal of Information and Communication Technology Research, Volume 2 No. 5, ISSN 2223-4985.

[9] Jie Gao. (2021). Research on Application of Improved Association Rules Mining Algorithm in Personalized Recommendation, Journal of Physics, Vol 1744.

[10] Anas Aldaher, Mohammmad Shkoukani. (2017). A proposed dynamic algorithm for association rules mining in big data journal of Theoretical and Applied Information Technology,Vol. 95. No. 13.

[11] S. Kanhere, A. Sahni, P. Stynes and P. Pathak. (2021). "Clustering Based Approach to Enhance Association Rule Mining," Conference of Open Innovations Association (FRUCT), pp. 142-150.

[12] Zhenyi Zhao, Zhou Jian. (2021). An improved association rule mining algorithm for large data, International Journal on Intelligent Systems, Vol. 30: 750–762.

[13] Jha, Jayshree and Leena Ragha. (2013). "Educational Data Mining using Improved Apriori Algorithm.", International Journal of Information and Computation Technology, ISSN 0974-2239 Volume 3, Number 5, pp. 411-418.

[14] K. Dharmaraajan and M. A. Dorairangaswamy. (2016). "Analysis of FP-growth and Apriori algorithms on pattern discovery from weblog data," 2016 IEEE International Conference on Advances in Computer Applications (ICACA), pp. 170-174.

[15]Y. Cong. (2020). "Research on Data Association Rules Mining Method Based on Improved Apriori Algorithm," 2020 International Conference on Big Data & Artificial Intelligence (AI) & Software Engineering (ICBASE), pp. 373-376.

[16]A. Essam, M. A. Abdel-Fattah and L. Abdelhamid. (2022). "Towards Enhancing the Performance of Parallel FP-Growth on Spark," in IEEE Access, vol. 10, pp. 286-296.

[17] Baohua Zhang. (2021). "Optimization of FP-Growth algorithm based on cloud computing and computer big data," International Journal of System Assurance Engineering and Management, vol. 12(4), pages 853-863.
[18] A. K. Suman and M. Gyanchandani. (2018). "Improved Performance of Hive Using Index-Based Operation on Big Data", Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1974-1978.