Comparison of Information Transfer Delay in Standard Apriori Algorithm and Improved Apriori Algorithm

Hooman Bavarsad Salehpour [1], Seyed Hamid Seyed Javadi [2], Parvaneh Asghari [3], Mohammad Ebrahim Shiri Ahmad Abadi[4]

[1] Department of Computer Engineering, Borujerd Branch, Islamic Azad University, Borujerd, Iran
Email: Bavarsad.hooman@yahoo.com

[2] Department of Mathematics and Computer Science, Shahed University, Tehran, Iran
Email: h.s.javadi@shahed.ac.ir

[3] Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran
Email: P_asghari@iauctb.ac.ir (Corresponding Author)

[4] Department of Mathematics and Computer Science, Amirkabir University, Tehran, Iran
Email: shiri@aut.ac.ir

Receive Date: 10 June 2022, Revise Date: 15 July 2022, Accept Date: 10 August 2022

Abstract

One of the most famous algorithms in the field of focused exploration of data mining correlation rules is the Apriori algorithm and its many developed versions. But what can be raised as a major challenge in this field is the proper application of this algorithm in the distributed environments of today's world. In this research, a parallelization-based approach is proposed to improve the performance of the Apriori algorithm in the process of exploring recurring patterns on network topologies. The proposed approach includes two major features: (1) combining the node centrality criterion and the Apriori algorithm to identify frequent patterns, (2) using the mapping/reduction method in order to create parallel processing and achieve optimal values in the shortest time. Also, this approach pursues three main goals: reducing the temporal and spatial complexity of the Apriori algorithm, improving the process of extracting dependency rules and identifying recurring patterns, comparing the performance of the proposed approach on different network topologies in order to determine the advantages and disadvantages of each topology. To prove the superiority of the proposed method, a comparison has been made between our approach and the basic Apriori algorithm. The evaluation results of the methods prove that the proposed approach provides an acceptable performance in terms of execution time criteria compared to other methods.

Keywords: data mining, Apriori algorithm, mapping and reduction, parallelization, network topology

1. Introduction

Data mining is a sub-process of "discovery of knowledge in the database" in which available data sources are analyzed using various data mining algorithms. [1]. In data mining from a part of statistical science called exploratory data analysis It is used in which the discovery of hidden and unknown information is emphasized from the massive volume of data. In addition, data mining is closely related to artificial intelligence and machine learning, so it can be said that in data mining, the theories of databases, artificial intelligence, machine learning and the science of statistics are combined to provide an applied context.

The term data mining refers to the semi-automated process of analyzing large databases in order to find useful patterns [2-4]. In fact, data mining means searching a

database to find patterns. between the data [4-6].

Data mining is one of the steps in the process of discovering knowledge from databases, which plays an important role in this process. Existence of correct and consistent information is one of the requirements that is needed in data mining. Mistakes and lack of correct information lead to wrong conclusions and incorrect decisions. Most organizations suffer from an information gap. In such organizations, information systems are usually built over time with different architectures and managements, so that integrated and specific information is not observed in the organization. The main reason that made data mining the center of attention in the information industry was the availability of a large amount of data and the strong need to extract useful information and knowledge from this huge amount of data. Data mining can be seen as the result of the natural evolution of information technology, which is the result of an evolution in the database industry. The evolution of database technology and its extensive use in various applications has led to the collection of large volumes of data. This abundant data has created the need for powerful tools for data analysis, because we are currently data-rich but information-deficient. In data mining operations, the goal is to find hidden and possibly useful patterns. A widely used and well-known type of these patterns are dependency rules. Many researchers have studied the development of methods and algorithms for the discovery of dependency rules, and one of these fields of development is the use of fuzzy logic. In fact, the development of algorithms for discovering dependency rules using fuzzy logic is in line with matching the assumptions of the problem with real conditions and overcoming its complexities; Therefore, some researchers have focused their studies in this field. One of the most famous algorithms in the field of exploring repeated items is the Apriori algorithm, whose similar algorithms are called Apriori series algorithms [6-9]. The a priori algorithm was discovered by Agrawal et al. at the IBM Almaden Research Center and can be used to generate all sets of frequent itemsets. Apriori is the basic algorithm of community rule mining (ARM) and its emergence led to the stimulation of research in the field of data mining. Apriori is one of the top 10data mining algorithms identified by the IEEE International Conference on Data Mining (ICDM) in 2006based on the most effective data mining algorithms in the research community. As a well-known algorithm, this algorithm is the basis of many methods presented not only in the field of focused exploration of dependency rules, It is also about parallel and distributed exploration of these rules. Therefore, it is necessary to give a brief explanation about this algorithm.

Running Apriori directly on the MapReduce framework adds scheduling and waiting time overheads. Due to the iterative nature of Apriori, the new task is triggered to recurse and the new task cannot be started until the previous task is completed. Lin et al. proposed three versions of Apriori based on MapReduce, which are single pass counting (SPC), fixed pass compound counting (FPC) and dynamic pass compound counting (DPC). The FPC and DPC algorithms greatly improve performance and are the most efficient implementations of Apriori on the MapReduce framework with the least number of iterations. SPC is a direct

implementation of Apriori on MapReduce, while FPC and DPC combine the candidate generation and counting of successive SPC phases/tasks into a single phase/job. FPC combines a fixed number of consecutive SPC phases (generally 3phases) into one MapReduce phase, hence reducing the number of scheduling requests[9-11]. DPC merges candidates from multiple consecutive phases to consolidate the workload between phases. Now suppose we combine the candidate generation from three consecutive passes k, k+1, and k+ 2into a multipass phase, and then k candidate itemsets are generated from (k-1) repeated itemsets, k+ 1candidate itemsets from candidate itemsets K are generated and candidate item sets (k+2) are generated from candidate item sets k+1. This leads to false positive candidates when the next level itemsets are generated from the candidate itemsets themselves. FPC may have poor performance due to too many false positive candidates and this is because FPC combines a fixed number of passes in all phases. DPC overcomes this problem by dynamically combining successive phases. The drawback of DPC is the strategy it uses to determine the dynamic number of passes. DPC is directly dependent on the execution time of the previousphase to determine the composition of passes. The execution time cannot be an absolute parameter because it varies in clusters with different size and capacity as well as on new datasets [11-14].

The most important problem of the a priori algorithm is that due to repeated visits to the database and the generation of multiple and frequent candidate item sets (for d items in a data set, there will be 2^d possible candidate item sets), this The algorithm is in the category of exponential algorithms both in terms of time complexity and space complexity. On the other hand, database topics and especially database association rules are transactional and therefore contain huge data. Therefore, it is clear that applying the a priori algorithm to find interesting connections between the abatoms in the database will not be possible in practice and requires the use of cloud systems that are almost impossible to access[14-18].

Therefore, the idea that is considered in this research to solve this problem or improve the performance of the a priori algorithm is to use parallel exploration techniques, which by distributing data on different processors and applying the algorithm in parallel and simultaneously, the speed of calculations is expected to be the same. Acceptable and the association rules can be extracted in a reasonable period of time. And as a result, an effective step should be taken in the direction of reducing the time and space complexity of this algorithm.

Another issue that is followed in this study was to load the parallelized a priori algorithm on the main topologies of the network (linear, star and ring) in order to compare the results (in terms of operation speed, required space and the number of generated association rules) introduced the best solution for using the apriori algorithm.

2. Materials and Method (proposed method based on parallelization)

2-1- The function of Apriori algorithm

Apriori is an iterative algorithm that switches between two important tasks, the first task is to generate candidates from the set of frequent items in the previous iteration, and the second task is to check the database to support candidate counts against

each transaction. In the k-th iteration (k>2), a set of Ck items is generated to support counting. The set of candidate items Ck is obtained by conditional linking of Lk-1with Khd and then cutting this set of items that does not satisfy the Apriori property[19]. According to this property, all itemsets of Ck can be removed from Ck if any of their (k-1) subsets are not contained in Lk-1.

Various algorithms have been presented to identify frequent patterns. In this research, the combination of node centrality criterion and Apriori algorithm is used to identify frequent patterns.

Apriori algorithm is one of the first algorithms used to find the set of frequent items. Its name is derived from the methods it uses, that is, using the knowledge of the previous step, which we will describe below. The a priori algorithm was discovered by Agrawal et al. at the IBM Almaden Research Center and can be used to generate all sets of frequent itemsets.

The Apriori algorithm is a surface search algorithm, which moves to the next step, i.e. k+1, when the exploration is finished at the kth stage. This process is repeated until the final condition or conditions are fulfilled. In the kth step, a set of k items will be produced. After calculating the support value for each and comparing it with the value of minsup, k frequent patterns are identified.

In the next step, the algorithm generates a set of (k+1) candidate items that can potentially be repeated with the help of k repeated patterns. In the same way, according to the value of minsup, some will be removed and the set of (k+1) repeated items will be formed. This process continues until the last repeated font set is found.

During execution, this algorithm uses the so-called Apriori rule, which is expressed as follows: "If we have a repeated pattern, all its subsets are also repeated." In other words, if the set of items I is not frequent, then any set that contains I cannot be frequent either.

With the help of the Apriori rule, the search space is reduced. The Apriori rule belongs to a special group of rules that have the unique property. This property is briefly expressed as if the set fails in a test, all its supersets will also fail in the same test[20].

Suppose J can be any instance of the set of items that results from the set I. A scale f has the uniformity property if:

$$\forall X, Y \in J: (X \subseteq Y) \Longrightarrow f(X) \leq f(Y) \qquad (1)$$

which shows that if X is a subset of Y, then f(X) must not be greater than f(Y). On the other hand, f has anti-monotonic property if:

$$\forall X, Y \in J: (X \subseteq Y) \Longrightarrow f(X) \leq f(X) \qquad (2)$$

Any scale and rules such as Apriori, which has anti-uniformity properties, can be effective for data mining algorithms such as generating sets of frequent items.

Below is the pseudocode of the Apriori algorithm:

```
Apriori(T, ε)
    L1 ← { large 1-itemsets }
    k ← 2
    while Lk-1 ≠ ∅
        Ck ← {c|c = a ∪ {b} ∧ a ∈ Lk-1 ∧ b ∈ ⋃Lk-1 ∧ b ∉ a}
        for transactions t ∈ T
            Ct ← {c|c ∈ Ck ∧ c ⊆ t}
            for candidates c ∈ Ct
                count[c] ← count[c] + 1
        Lk ← {c|c ∈ Ck ∧ count[c] ≥ ε}
        k ← k + 1
    return ⋃k Lk
```

The a priori algorithm produces candidate item sets of length k from item sets of length k-1. Then, candidates that prune rare subpatterns (subpatterns that occur infrequently). According to the downward closure lemma, the candidate set includes all frequent item sets of length k. After that, it scans the transactional database to determine frequent item sets from among the candidates.

Another challenge in this field is to increase the processing speed in this field. For this purpose, in this research, the map/reduce model was used to increase the speed of parameter calculations by using parallel processing. Values were obtained in the shortest processing time.

In general, the proposed method includes the following sections:

- Problem modeling and expression of service quality parameters and calculation of these parameters
- Using the optimization algorithm to improve these indicators
- Use of mapping and reduction model to increase processing speed in this system

2 -2- Different Parts of Distributed Computing

In this study, the parallelization of the Apriori algorithm was discussed using the mapping and reduction model, and the performance of the proposed system was evaluated.

1-2-2-Parallel Architecture: Parallelism is an architectural pattern that is based on a network of clients that receive services from provider devices. These devices have computing and storage capacity that allows them Assign data and instances of cloud services to each customer. Therefore, data and service management policies are needed to decide where and when to place services and data. The problem of placing services is a big problem in the field of services.

Parallel algorithms in computer science, unlike traditional sequential algorithms, are algorithms in which, each time, a part of the program is executed on a different processor, and finally, the results are put together to obtain the desired result.

Parallel algorithms should also be optimized in terms of communication between different processors. Parallel algorithms communicate with processors in two ways, shared memory, and message exchange. Shared memory processing requires additional locking for data, hence incurs the cost of bus cycles and additional processors, and also causes non-parallelism. become parts of the algorithm. Processing through message passing uses channels and message boxes, but this type of communication increases the cost of transmission on the bus, additional memory for queues and message boxes, and delays in messages. In multi-processor designs, special buses are used in order to reduce the costs of interactions, but it is the processor that determines the volume of traffic. The problem of other parallel algorithms is to ensure their appropriate balance.

2-2-2-Mapping and reduction model: It is a parallel programming model for processing data on clusters, which consists of two main phases, including the mapping phase and the reduce phase.

1- **MAP:** The MAP master node receives the input and converts it into smaller parts and divides them among the worker nodes. The worker node itself may repeat this step, resulting in a tree structure

2- **Reduce:** The Reduce Master node receives the answer of the reduced parts and combines them with each other to form the desired output

To implement these two steps, two Map and Reduce functions are needed. This method is a simple programming model that is used to solve computational problems on a large scale and in a distributed manner. and provides a secure and scalable platform for the development of distributed applications. Map Reduce implementations are Master/Slave models.

In this research, the information and matrix of each solution is entered in the map section, and in the reduce section, the value of the fitness function of each solution is calculated in parallel to reduce the processing time, because the largest computational volume in this algorithm is in the fitness function calculation section.

The advantage of map-reduce is that it allows the processing of processing and reduction operations to be distributed. Providing that each of these mappings is independent from others, which implies the parallel execution of these mappings. Figure 3-2-2 shows how mapping and reducing functions work.

2-3- Research hypotheses

- In solving the problem, it is always assumed that
- The transactions used by the problem are stored in a transactional database.
- In providing typical solutions to problems, it is assumed that the transactional database in question will not be updated.
- If the transactional database is updated, this update will not change the most frequently explored patterns.
- The same data is provided to the processors to allow parallelization.
- Network topologies will use exactly the same data.

2-4- Problem modeling

The parameters examined in this research include the following, which can be defined based on the following relationships:

Average processing time: This time is related to processing using two methods map and reduce

$$Average\ Execution\ Time = \frac{\sum_{i=1}^{nm} et_m(i)}{nm} + \frac{\sum_{j=1}^{nr} et_r(i)}{nr} \quad (3)$$

Based on this relationship, etm is the time spent for processing in the mapping method and etr is the processing time in the reduction method. And nm is the number of jobs in map mode and nr is the number of jobs in reduce mode, which in this research, these two values are the same. Because tasks are assigned to both mapping and reduction methods.

Critical time (make span): It is the time required to process all tasks, which must be assigned to the network nodes in such a way that this amount is as low as possible to complete all tasks.

Computing cost of the machine: This value shows the cost of using the CPU based on dollars, which can be calculated in the form of the following relationship.

$$VM\ Computing\ Cost = \left(\sum_{i=1}^{nm} et_m(i) + \sum_{j=1}^{nr} et_r(i)\right) \times VM\ Cost\ per\ Unit\ Time \quad (4)$$

Based on this relationship vm cost represents the cost of using cpu per second time unit. etm is the time spent for processing in the mapping method and etr is the processing time in the reduction method.

2-5- Optimizing the fitness function

The three parameters of the previous index should find a minimum value, based on these three parameters, the final fitness relationship is as follows:

Fitness=w1*AvrageExecutionTime+w2* makeSpan+w3*Vmcost

Based on this relationship, w1, w2, w 3represent proportionality coefficients that are used to sum these three parameters with different unit values. Based on this relationship, the sum of w's is equal to one.

2-6- The data used

The field of research in this article is the pharmaceutical database of the Social Security Organization as a case study. In this database, the number and amount of drugs consumed based on the type of insurance and the patient's referral, the gender of the patient and the specialty of the doctor prescribing the drug exist and is used to determine the amount of consumption of a specific drug in a specific season.

In order to evaluate the discussed algorithms, experiments have been performed on tasks with different sizes according to Table 1. Table 1shows the task length of each machine

Table 1: length task values of each task

Task id	Length task
1	2,160,657
2	1,835,957
3	1,819,923
4	1,747,767
5	1,599,447
6	1,583,413
7	1,607,465
8	1,427,076
9	1,463,154
10	1,447,119
11	1,527,292
12	1,503,240
13	1,495,223
14	1,362,938
15	1,370,955
16	1,378,972
17	1,471,171
18	1,431,084
19	1,439,102
20	1,419,059
21	1,403,024
22	1,407,033
23	1,407,033
24	1,423,067
25	1,419,059

In addition to the task length values shown in Table 1. Other values must be specified for the machines, which are shown in Table 2. These values, in addition to the length task value, must be entered into the simulator so that the simulator can be executed.

Table 2. Specifications of other data center parameters

Parameters	Value
Lentgh (MIPS)	1,362,938 – 2,160,657
Pes	1
Input Size (Bytes)	291,738
Output Size (Bytes)	5,662,310

To evaluate this project, ten machines have been used to test the system. For this purpose, the results obtained using the proposed method were discussed first, and in the next step, the new proposed algorithm was discussed to compare this system. For this, the MATLAB simulator environment is used.

2–7- Evaluation criterion:

In this research, the criterion for evaluating the proposed algorithm is the delay in information transmission, which indicates the amount of time required to transmit information from the source to the destination.

3- Results

In this section, the comparison of TFI-apriori proposed algorithm - apriori and apriori with other implementations, results and evaluation of the proposed method to improve the apriori algorithm using parallelization technique on network topologies was discussed. In this study, the discussed system is shown as a data center containing N heterogeneous physical nodes. Each node i has the following properties.

- The amount of processor that is determined by MIPS (processing power)
- Amount of memory
- Network bandwidth

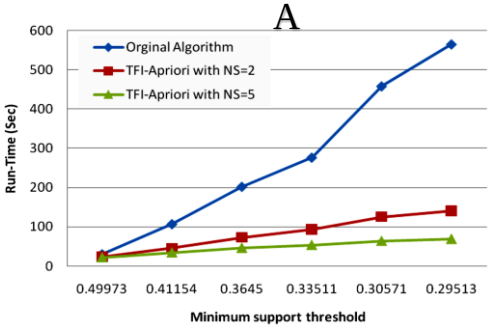
The data center is given to m machines, which have three characteristics: processing power (mips), amount of memory and network bandwidth.

The proposed - apriori and apriori algorithm are both implemented based on Bodon's proposed data. We have used the Bodon source code and the original apriori algorithm implementation. 2.4.9, issue date: 11. (March 1, 2005) 1and its modification to implement the TFI-apriori algorithm. Both algorithms have been implemented on real datasets known and combined with minimum support threshold.

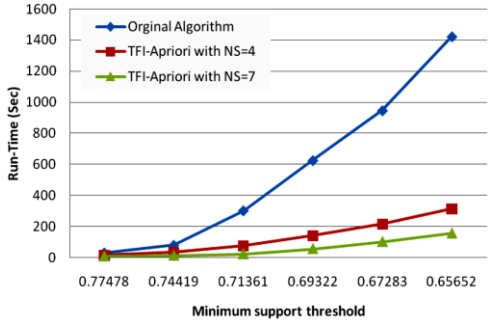
3-1- Evaluating the effect of the NS parameter on the efficiency of the apriori algorithm

Proposed Algorithm - Apriori and Apriori Both algorithms have been implemented on known and combined real data sets with minimum support threshold. To evaluate the effect of the NS parameter on the efficiency of the a priori algorithm, for each data set, TFI-apriori was run with different values for NS P.

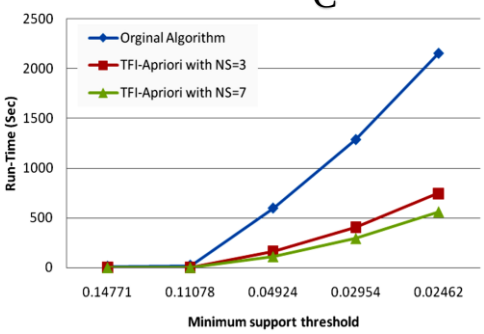
A



B



C



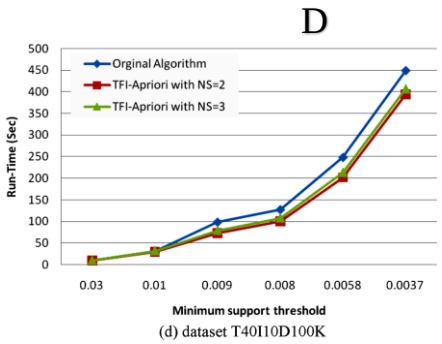


Fig 1. (A) Comparison of execution time with different minimum backups (dataset accident), (B) Comparison of execution time with different minimum backups (dataset pumsb), (C) Comparison of execution time with different minimum backups (dataset mushroomm), (D) Comparison of execution time with different minimal backups (dataset T40I10D100K)

As can be seen in Figure 1, on some real data sets, TFI-apriori had an outstanding performance compared to the original apriori algorithm, and in most cases, the best results belong to an NS parameter between 5and 7. For lower minimum support values, the efficiency of the algorithm was more visible. As can be seen from these results, the yield reached ٪90 in some cases. However, in the combined data set, there was not much increase in the efficiency of the proposed algorithm. This lack of improvement was due to the fact that there are no high-frequency items in the well-known hybrid dataset. For example, and based on the obtained statistics, the frequency of the highest number of synthetic datasets is T40I10D100K, which is about ٪8. This amount is very small and cannot be neglected. The frequency of the largest number of data in the random data set is about ٪100, which means that there is one item that is included in all transactions. The second, third and fourth (repeated cases after the first TFI) in the random data set have a frequency higher than ٪99. The frequency of the Pumsb and Fungi datasets is more than ٪99. The frequency of TFI in pumsb-star data set is about ٪75

Figure 2 shows the memory consumption of Apriori compared to the original Apriori, showing that not only the implementation of the proposed algorithm does not require additional memory allocation, but it is also reduced in most cases. In TFIApriori execution, there is almost a monotonic relationship between memory consumption and runtime efficiency.

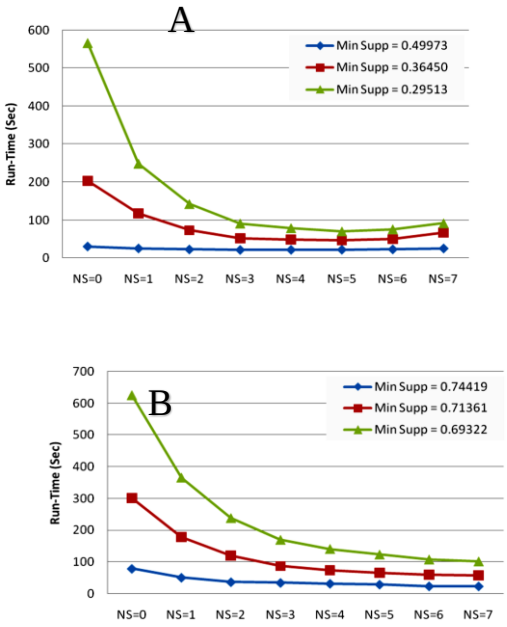


Fig 2. (A) Effect of NS parameter during execution (dataset accident), (B) Effect of NS parameter during execution (dataset pumsb)

3-2-Evaluation of the Proposed Method

To evaluate this project, ten, twenty or thirty machines have been used to test the system. For this purpose, each of the algorithms has been tested three times consecutively with this number of machines, and the results obtained using the proposed method as well as the Apriori algorithm have been expressed in most of the graphs and tables.

Table 3. The amount of delay using both algorithms in different modes

Number of cars	The proposed algorithm	Standard Apriori algorithm
10	2178	2298
20	2479	2509
30	2531	2621

Also, in Figure 3, this delay evaluation is shown graphically, and the results indicate the superiority of the proposed method.

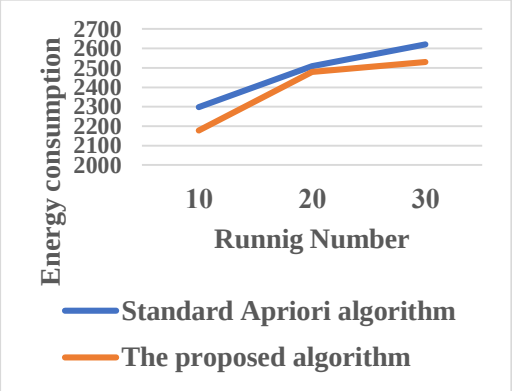


Fig 3. The amount of delay in four different modes

The evaluation of various tests showed the superiority of the proposed method both in terms of energy consumption and in terms of data center transmission rate and delay. Also, the proposed algorithm has a better performance than the basic algorithm due to the use of a larger search space and the use of more operators to find the optimal solution.

Also, Table 4 shows the execution time of the proposed method and the standard Apriori article. As seen in these experiments, the proposed method has a shorter execution time, which shows that the performance quality of this method is better and it has less computational complexity.

Table 4. Execution time using both algorithms in different modes

Number of cars	The proposed algorithm	Standard Apriori algorithm
10	142	187
20	209	253
30	289	308

Also, in Figure 4, this delay evaluation is shown graphically, and the results indicate the superiority of the proposed method.

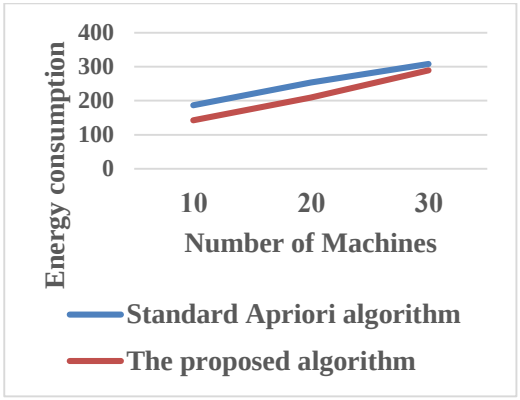


Fig 4. execution time in four different modes

In addition, Table 5 examines the lifetime of the network in the proposed method and the standard Apriori article. As can be seen in this table and figure, in all cases, the proposed method has improved the lifetime of the network.

Table 5. Network lifetime using both algorithms in different modes

Number of cars	The proposed algorithm	Standard Apriori algorithm
10	456	401
20	519	471
30	629	528

Also, in Figure 5, this delay evaluation is shown graphically, and the results indicate the superiority of the proposed method.

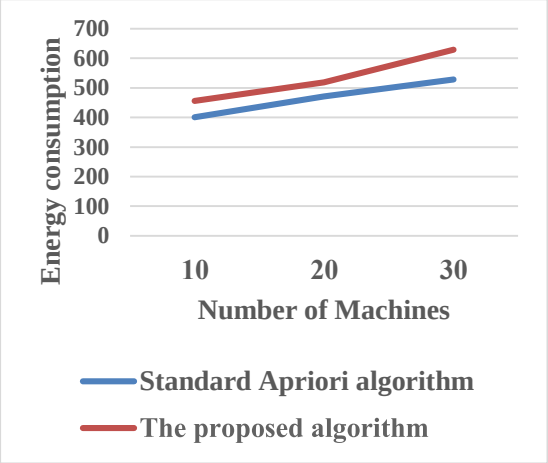


Fig 5. Running time in four different modes

Conclusion

In recent years, data management has become increasingly important to organizational performance, and organizations have realized that they can learn a lot from analyzing the data that is generated as a result of their daily operations.

Data mining and the studies conducted in this field, as well as the development of common data mining methods, including the exploration of dependency rules, have all been in line with the realization of this goal. The use of fuzzy logic along with data mining is one of the current solutions to solve this problem and increase efficiency, accuracy and speed due to the fact that it is more compatible with the facts and improves the results.

In other words, data mining tries to discover information and hidden rules from various data that remain unknown to humans. The exploration of association rules is considered a fundamental problem in data mining and database exploration, and recently it has been able to attract significant popularity from the industrial and academic communities.

There are many algorithms for exploring association rules, among which Apriori is considered a basic and famous algorithm.

With the rapid development of the Internet, data sets have grown and databases have become larger and larger. As a result, many problems arise in data mining. Traditional data mining algorithms face many obstacles when dealing with massive data. On the other hand, voluminous data has brought challenges to data mining.

With large databases and valuable sparse information, the time spent dealing with datasets is growing rapidly, and big data applications cannot afford this time cost associated with the initial Apriori process. When the data size increases, the Apriori algorithm faces problems and the calculation time increases dramatically.

Real and practical data sets usually include a series of records with a number of characteristics, the type and scale of different characteristics are often different from each other, however, in the algorithms in the field of discovering fuzzy dependency rules, the characteristics of the database are always considered the same for the sake of simplicity. And deterministic classification is often used, while in some real applications, deterministic classification of characteristics may cause the loss of part of the information or lead to incorrect results.

Another challenge in this field is to increase the processing speed in this field. For this purpose, the mapping and reduction model was used in this research to increase

the speed of parameter calculations by using parallel processing so that the most optimal values can be obtained in the shortest processing time. reached In this research, the information and matrix of each solution is entered in the map section, and in the reduce section, the value of the fitness function of each solution is calculated in parallel to reduce the processing time, because the largest computational volume in this algorithm is in the fitness function calculation section.

To show the efficiency of the presented algorithm, comparing the performance of the proposed method with the basic method showed the superiority of the proposed method and the results showed that the proposed method has superiority in terms of efficiency compared to the method of the basic article.

One of the fields of study is the development of the exploration of dependency rules along with fuzzy logic, which is possible in different ways. Experimental results show that major improvements in execution time are achieved for repeated itemsets in real datasets. However, in datasets where the difference between case frequencies is less significant (i.e., where there are no cases with very high frequencies compared to other cases), the proposed -apriori and apriori algorithm is usually not very effective in the real world. In addition, the experimental results show that not only the implementation of the proposed algorithm does not require additional memory allocation compared to the apriori-based algorithms, but also the memory required for its implementation has been reduced. Experimental results show a decrease in execution time without increasing memory allocation compared to Apiori. Algorithm. It should be noted that the memory required for this algorithm is reduced. In the presented algorithm, the criterion for choosing TFI has the highest frequency. This selection method leads to some remarkable results for real databases. However, we do not consider other criteria for Apiori selection, there may be other TFI selection criteria that can be considered in the future. Such a study may also provide further insight into finding optimal NS values for any given input database and minimum support threshold. In addition, we may be able to generalize the presented idea to other data mining tasks and optimize them in a more efficient way using some special cases.

Most of the existing algorithms based on the mapping/reduction framework, such as the proposed solution, only implement the generation phase of the set of frequent items in parallel, and no effort has been made to generate the rules in parallel. Therefore, the direct generation of all association rules from the dataset and based on the mapping/reduction framework can be one of the very interesting ideas for the future.

By improving the proposed algorithm, a faster version can be achieved, which will be needed to develop it into a parallel and distributed environment. This algorithm implements only the step of finding the set of frequent items in parallel. The phase of extracting association rules should be done directly and secretly. Also, the proposed solution depends on the min_sup threshold value. Because compared to the k-phase algorithm, and for high min_sup values, it has the opposite result and the execution time of the proposed algorithm is higher.

One of the main problems in exploring association rules is the need for multiple disk accesses. The proposed algorithm also scans the data set of each transaction in each phase. Repeated traversals increase the execution time of the algorithm. Among other limitations of the proposed algorithm, we can mention the large number of implemented phases. When the number of phases is large, the overhead of waiting and scheduling each phase imposes a high execution cost on the algorithm. Therefore, you should think of a plan to reduce the number of execution phases and improve the execution time as a result.

The combination of data mining with fuzzy logic is an important and extensive field of study as well as a suitable platform for research due to the importance, application and scope of each. The gap that usually exists in the proposed algorithms in this field is to deal with theoretical ideas and pay less attention to providing practical methods in fuzzy development.

In this research, the focus was on extracting fuzzy multi-level dependency rules using fuzzy classification, which at the same time by adding features such as considering the value of min_sup and max-approximate differently for different items, as well as the possibility of defining multiple fuzzy membership functions for each level. From the classification, it was tried to make the presented algorithm practical and comprehensive.

The algorithm implemented in this research can be made faster with the help of advanced techniques and data structures. It is also necessary to provide a method that can determine the maximum length of the set of candidate items. In this way, it is prevented from producing a set of candidate items that have no effect on the final output and cause memory loss. The dominant cost in the implemented algorithms based on the mapping/reduction framework is related to the cost of communication. In order to increase the efficiency of the algorithm, reducing the communication between mappers and reducers and optimizing the heavy operations related to the intermixing and merging phases are other future goals. One of the technical strategies in exploring efficient associative rules is to optimize the framework that executes the algorithm. In other words, in addition to designing and implementing an optimal algorithm for exploring association rules, the configuration of the framework can be improved. If we run the parallel algorithm on the optimized framework, we get better results. However, there are also some methods that improve the performance of the parallel Apriori algorithm, such as algorithms that change the counting step to make the exploration process faster.

Most of the existing algorithms based on the mapping/reduction framework, such as the proposed solution, only implement the generation phase of the set of frequent items in parallel, and no effort has been made to generate the rules in parallel. Therefore, the direct generation of all association rules from the dataset and based on the mapping/reduction framework can be one of the very interesting ideas for the future.

References

[1] Simon Fraser University, Morgan Kaufmann publishers, Two Crows Corporation. (1983). Introduction to Data Mining and Knowledge Discovery. ISBN,pp.40-47.

[2] David Hand, Heikki Mannila and Padhraic Smyth. (2001). Principles of Data Mining. The MIT Press, pp. 546.

[3] Jiawei Han, Micheline Kamber. (2006). Data Mining: Concepts and Techniques. Second Edition, by Elsevier Inc, pp.772.

[4] Agrawal,R. and Srikant.R. (1994). Fast algorithms for mining association rules. Proceeding of the VLDB, Santiago de chile, pp.487–499.

[5] Wan, J.W. and Dobbie,G. (2004). Mining Assoaciation Rules From Xml Data Using XQuery. In proceeding of the second workshop on Australasian information security, Data mining and web intelligence, and software internalization.,vol.32, pp.169-174.

[6] Zhao, Q. and Bhowmick, S.S. (2003). Association Rule Miining: A survey. CAIS, No. 2003116,pp. 20-31.

[7] Verma, N., Malhotra, D. and Singh, J., 2020. Big data analytics for retail industry using MapReduce-Apriori framework. Journal of Management Analytics, pp.1-19.

[8] Sornalakshmi, M., Balamurali, S., Venkatesulu, M., Krishnan, M.N., Ramasamy, L.K., Kadry, S. and Lim, S., 2020. An efficient apriori algorithm for frequent pattern mining using mapreduce in healthcare data. Bulletin of Electrical Engineering and Informatics, 10(1), pp.390-403.

[9] Luna, J.M., Padillo, F., Pechenizkiy, M. and Ventura, S., 2017. Apriori versions based on mapreduce for mining frequent patterns on big data**.** IEEE transactions on cybernetics, 48(10), pp.2851-2865.

[10] Verma, N. and Singh, J., 2017. An intelligent approach to Big Data analytics for sustainable retail environment using Apriori-MapReduce framework. Industrial Management & Data Systems, pp 381-394.

[11] Han, J., Kamber, M. and Pei, J., 2011. Data mining concepts and techniques third edition. The Morgan Kaufmann Series in Data Management Systems, 5(4), pp.83-124.

[12] Agarwal, S., 2013, December. Data mining: Data mining concepts and techniques. In 2013 International Conference on Machine Intelligence and Research Advancement (pp. 203-207). IEEE.

[13] Gupta, M.K. and Chandra, P., 2020. A comprehensive survey of data mining. International Journal of Information Technology, pp.1-15.

[14] Maksood, F.Z. and Achuthan, G., 2016. Analysis of data mining techniques and its applications. International Journal of Computer Applications, 140(3), pp.6-14.

[15] Altaf, W., Shahbaz, M. and Guergachi, A., 2017. Applications of association rule mining in health informatics: a survey. Artificial Intelligence Review, 47(3), pp.313-340.

[16] Huang, M.J., Sung, H.S., Hsieh, T.J., Wu, M.C. and Chung, S.H., 2019. Applying data-mining techniques for discovering association rules. Soft Computing, pp.1-7.

[17] Ghorbani, M., Abessi, M.: A New Methodology for Mining Frequent Itemsets on Temporal Data. IEEE Transactions on Engineering Management, Vol. 64, No. 4, pp. 566-573 (2017).

[18] Keyvanpour, M.R., Mehrmolaei, S. and Etaati, A., 2020. PLI-X: Temporal Association Rules Mining in Customer Relationship Management Systems. Computer and Knowledge Engineering, 2(2), pp.29-48.

[19] Fournier-Viger, P., Lin, J.C.W., Vo, B., Chi, T.T., Zhang, J. and Le, H.B., 2017. A survey of itemset mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7(4), p.e1207.

[20] Luna, J.M., Fournier-Viger, P. and Ventura, S., 2019. Frequent itemset mining: A 25 years review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(6), p.e1329.