A PROPOSED DYNAMIC ALGORITHM FOR ASSOCIATION RULES MINING IN BIG DATA

[1]ANAS HANI AL-DAHER, [2] MOHAMMAD SHKOUKANI

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

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

E-mail: [1]anas-aldaher@live.com, [2] m.shkokani@asu.edu.jo

ABSTRACT

Because of the explosive growth of data that we are suffocating in, while we are starving for knowledge, mining data and information from substantial databases has been perceived as a key research topic. So, Due to the huge size of data that exists in the databases and warehouses and because these data are big, dynamic and change frequently it is difficult and expensive to do mining for frequent patterns and association rule from scratch. In light of such an interest, this paper proposes a Dynamic Algorithm for Association Rules Mining in Big Data that is capable of finding frequent item-sets dynamically and generating association rules from the item-sets by using accumulated knowledge stored in a database table, this table will be modified frequently when the system runs every time and the new value of the table will be the result of processing new inserted data added to the results of previously processed data. The proposed solution is implemented using C#.net and SQL server. The results compared with the Apriori algorithm. It was conclude that Apriori algorithm showed better results than the proposed algorithm in the initial runs, on the other hand the proposed dynamic algorithm provided results near to Apriori algorithm on frequent runs that use small number of transactions but the proposed dynamic algorithm took less processing time than Apriori algorithm by 63.95% on the frequent runs that use big number of transactions.

Keywords*:* *Big data, Apriori algorithm, Association rules, Data mining*

1. INTRODUCTION

Data size has expanded clearly through the previous years, it has raised nine times in the previous five years and it will continue to double every two years in the future as the International Data Corporation (IDC) shown in their introduced report in 2012. Because of the data volume, variability, velocity, and ambiguity we can't utilize it straightforwardly. So we should make it useful by analyzing and processing it to extract and discover the unknown useful information, which is known with Big Data mining [8], [10].

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

- What is the way to do mining over dynamic big data?
- How to implement the solution on real world with real data?

The most important limitations encountered can be summarized as follows:

- Process Big data needs to powerful server, which required certain data storage and data transferring speed.
- Generating all combinations in big data is a complicated high cost process that needs a lot of time.

This paper consists of eight sections. The first section is an introduction. The second Section discusses the Big Data. The third section explains the Data mining. The fourth section presents the problem statement of the paper. The fifth section discusses the proposed algorithm. The sixth explains the implementation process. While the seventh section exhibits the results and the eighth is the conclusion and the future work.

2. BIG DATA

Big data approach was first introduced by META group in 2001 and it was defined as a large set of data with a size that is not easy to handle and be used by traditional management systems, these data usually can be collected and stored from sensors, mobile phones, sales transactions and software logs … Etc.[3]

Big data have three main characteristics to focus on and discuss; those characteristics are known with the 3 V's and they are Volume, velocity and Variety).When we talk about data volume we are actually talking about the enormous size of data that is generated every day and collected from different sources and how to handle this data and do operations over it to get useful results, and when we talk about data velocity it's like talking about drinking water from a waterfall and how to do that efficiently, that is what actually happens when dealing with stream or real time data we need to find a way to process data fast and concurrent to avoid losing it or its profit , the last V is data variety which indicates that data are not all having the same form or shape , data could be structured or unstructured and because of that we need to find a way to handle the different types of data [8], [15].

Finally we should know that when talking about mining big data there are sequential steps to follow such as doing data organizing then data integration after that data analysis and finally decision making, and all of these steps are shown in figure 1 below [5].

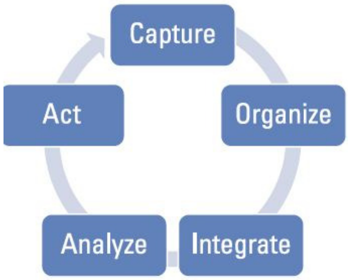


Figure 1: Big data processing cycle

3. DATA MINING

Data mining is the process of extracting fascinating implicit and possibly valuable patterns or knowledge from huge amount of data, also known as knowledge discovery from database (KDD). Data mining is so important for business because it plays a big role in decision support by providing answers for many questions about client behavior, how enhance the provided service and how to expand business revenue [16].

Data mining process consists of many steps we should follow to get the right results, these steps start by data cleaning which is used to repair errors and corruptions in data, solve missing data problem, and uniform all data formats, the second step is data selection that aims to select needed data from the previous step and storing it in data warehouse to be used in the following steps , after that we start finding the interesting relationships by doing mathematical and statistical analysis, after this becomes the turn of pattern evaluation to check that the results of the past work is useful and the correct needed one , if the results of this step accepted then the final step will be presenting them as patterns and graphs that is easy to use in decision support , these steps are shown in figure 2 below [6], [7], [9].

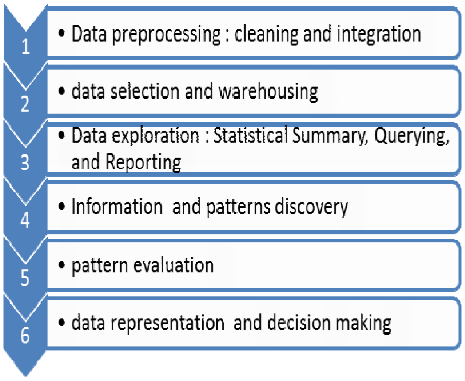

1
• Data preprocessing : cleaning and integration
2
• data selection and warehousing
3
• Data exploration : Statistical Summary, Querying, and Reporting
4
• Information and patterns discovery
5
• pattern evaluation
6
• data representation and decision making

Figure 2: Knowledge discovery steps

There are a lot of requirements and challenges in data mining that should be taken in consideration before using data mining algorithms, the first thing to study that the algorithm can handle different types of data , the second is that the algorithm can extract useful information from a huge amount of data efficiently, the third is to check how much the information discovered by the algorithm is valuable and useful and the next is to see if the algorithm can do mining over different sources of data , the last thing is the protection of user data and if the algorithm provides privacy [2], [4].

Recently a new approach raised in data mining called dynamic data mining which aims to find new knowledge using previously found knowledge with new data updates to maintain the current situation without doing all the calculations from the scratch which saves cost, this approach can be implemented by using summary table of past results combined with the new results and used to regenerate the knowledge that reflects the current situation [1].

4. PROBLEM STATEMENT

Data mining is so important in business to both predict and discover trends, and companies can make better and more effective business decisions such as marketing and advertising decisions that will help these companies to grow and expand their revenue.

Due to the huge size of data that exists in databases and warehouses and because these data are big, dynamic and change frequently it is hard and costly to do mining for frequent patterns and association rules from scratch every time such as the Apriori algorithm works; any update on data will enforce Apriori to do the full process and counting scans from scratch more and more to fetch the current state.

Handling and mining big data is a key problem but what is more important to deal with dynamic big data, two important questions rises here ; the first one is what is the way to do mining over dynamic big data and the second is how to implement the solution on real world with real data.

5. PROPOSED ALGORITHM

As discussed before the urgent need for dynamic algorithm to do mining over big data that is changed frequently and how data mining affects the decision making process and to contribute even with a partial solution, so the objective of this research paper is to propose:

- An algorithm to generate frequent item-sets dynamically without the need to do the Apriori algorithm from scratch.
- An algorithm to generate frequent association rules in dynamic way.

In this section the authors will discuss in details the proposed algorithm that generates frequent item sets and association rules from dynamic big data

The proposed algorithm consists of three sub algorithms which are Initial state building algorithm (ISB), Frequent Used Algorithm (FU), and Rule generation algorithm (RG) as shown in figure 3 and discussed below:

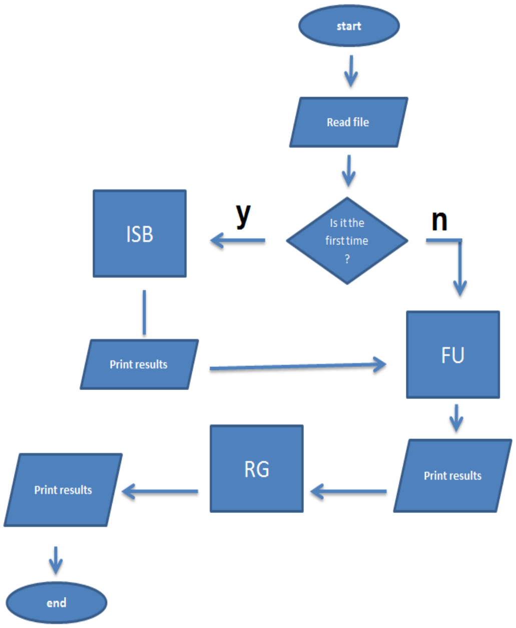


Figure 3: Flow Chart of the Proposed Algorithm

5.1 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 clarified below:

- read all items in data base and display them to the user
- read the user defined groups
- For each group in defined groups:
  - 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).

Stable: is the summation table, it consists of three columns which are: item, summation, and size, where:

- Item is all the candidate item sets that can be generated of market 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.

5.2 Frequent Used Algorithm (FU):

This algorithm is responsible of building updated state when using the system frequently and it works as clarified below:

```
• For each item-set in Stable
    o 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
    o 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
• Set flag for all the rows of Ttable = t
```

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.

5.3 Rule Generation Algorithm (RG):

After the frequent item-sets have been discovered, it is simple and clear to generate association rules from them as clarified below:

```
• 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
    o Generate all the sub sets (S) of the item-set (I)
    o 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))
```

Rtable: is the association rules table, it consists of four columns which are: left, right, support and confidence, where:

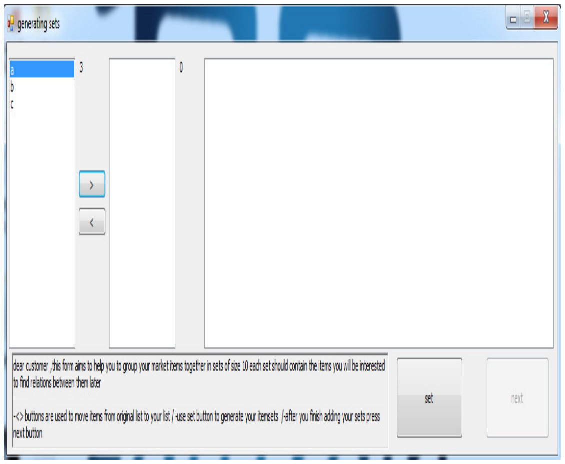
- Left: is item-set that is on the left hand side of the rule.
- Right: is item-set that is on the right hand side of the rule.
- Support: is the count of this item-set of left U right.
- Confidence: is the division result of support over support the left hand side.

6. PROPOSED ALGORITHM IMPLEMENTAION

SQL server database was used in implementation that consists of three tables; the first is rules that represents Rtable, the second is summation that represents Stable, and the last one is transactions that represents Ttable. In the implementation a big transaction table was used that contains 126 different kind of items in 126 column, id column as a key and the flag column, after that 1064 transaction were inserted to be used in the testing.

The proposed algorithm was implemented using .Net technology (windows forms) C# programing language [11], [12] and SQL language [13], [14], the outcome of the implementation was software that consists of 3 main screens (generating sets, counting sets, generating rules) These screens represent the 3 algorithms that were described before (ISB, FU, and RG).

Figure 4 shows the ISB algorithm screen, here user should select the groups of items using (<, > and set) buttons , each time he press (set) button the ISB algorithm checks that the group contains at least two elements then generates all sub sets of the selected group and insert it in the database.

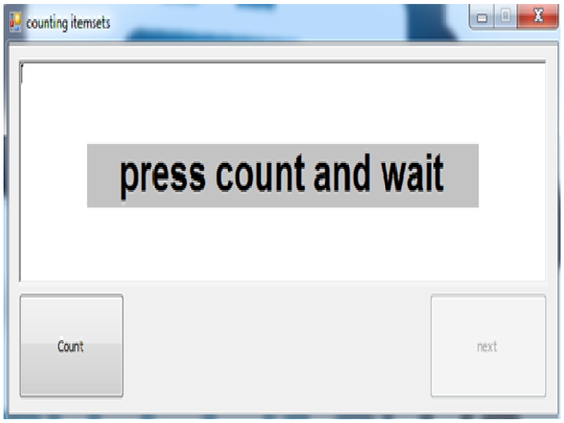


dear customer ,this form aims to help you to group your market items together in sets of size 10 each set should contain the items you will be interested to find relations between them later

-<> buttons are used to move items from original list to your list / -use set button to generate your itemsets /-after you finish adding your sets press next button

Figure 4: ISB Algorithm Screen

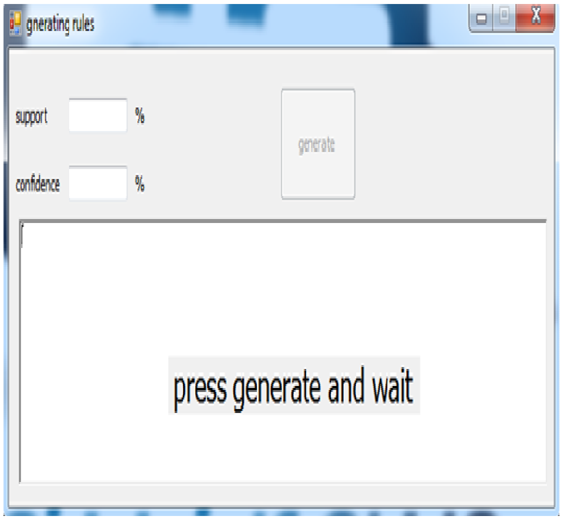
Figure 5 is the result of pressing the next button and it is responsible of counting the item-sets in Stable as, it will read each item set and check its size if the size is 1 it will count all the transactions in Ttable where flag = f and the column name is the item name and update the summation of the set , if the size is larger than 1it will decompose the set into its basic items and count all the transactions in Ttable where flag = f and where all items found on that transaction and update the summation of the set, when the algorithm finish counting all the item sets it will activate the next button and update the flag of the counted transaction to t so they will not be counted again.



press count and wait

Figure 5: FU Algorithm Screen

Figure 6 is the result of pressing the next button and it represents RG algorithm, this screen is responsible of generating frequent rules from Stable and insert them into Rtable into the database.



press generate and wait

Figure 6: RG Algorithm Screen

7. RESULTS AND DISCUSION

The system was tested and the results are explained with graphs as following:

Figure 7 represents relation between number of generated sets and consumed time in seconds, it appears that the ISB algorithm only needed 21.12 seconds to generate 12339 item-sets, and if we want to generate double number of the previous item-sets it will need a time near to the double of previous time.

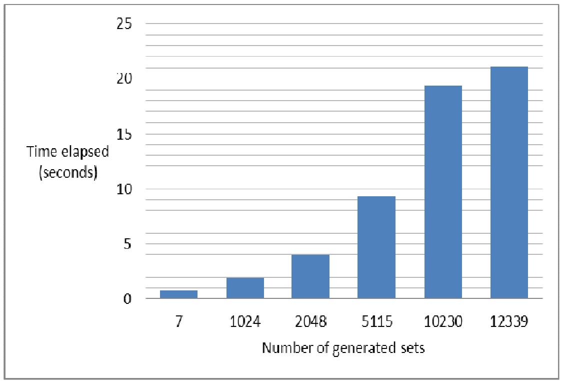


Figure 7: Relationship between number of sets and time

In the following figure 8 that represents relation between number of transactions and consumed time in seconds, it appears that the FU algorithm is not highly affected by the number of transactions.

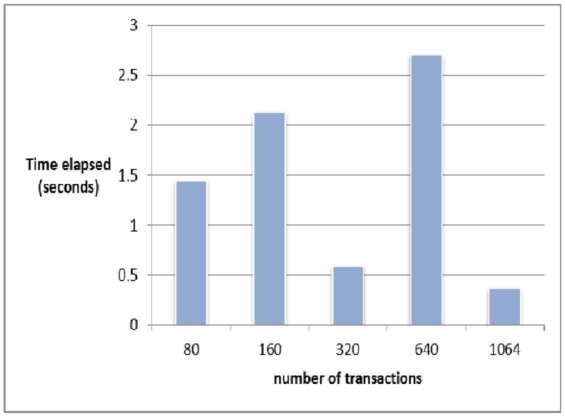


Figure 8: Relationship between number of transactions and time in FU

Figure 9 shows that increasing in number of sets generated by ISB will affect strongly the time consumed in FU.

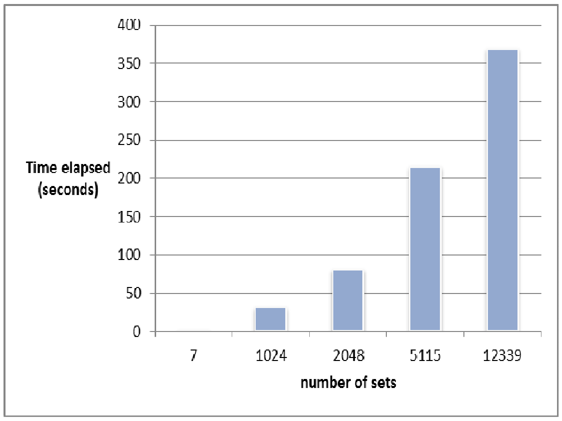


Figure 9: Relationship between number of sets and time in FU

After studying the impact of minimum support and minimum confidence on the results generated by RG algorithm it was found every time we increase the threshold the results will become less and the consumed time will be less too .

Figure 10 shows the impact of number of item-sets on the RG algorithm, and it leads to say that there is a positive relation between both of time consumed and number of item sets, this means increasing number of sets increases time consumed.

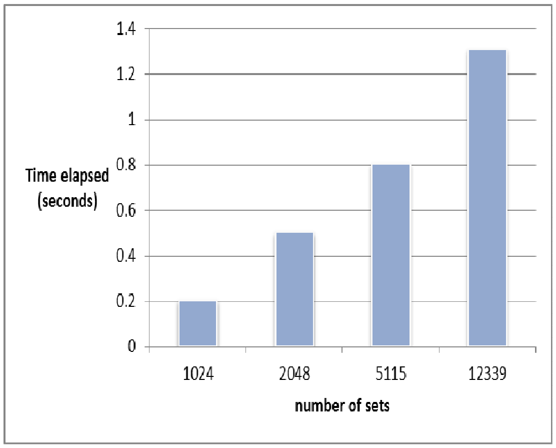


Figure 10: Relationship between number of sets and time in RG

Table 1 consists of 5 columns the state of the algorithm when time was recorded and the number of new transactions and number of the total transactions in the database and the time recorded in the proposed algorithm and Apriori algorithm in seconds.

When a comparison is made between the proposed algorithm and the Apriori algorithm according to the time consumed in dynamic runs it was found that the proposed algorithm save cost in frequent runs because it do less operations than Apriori Algorithm while the Apriori Algorithm keeps using more time for each new added transactions. On the other hand Apriori Algorithm performed better in the initial run and frequent runs with small number of inputs.

Table1: Results of Proposed algorithm Vs Apriori Algorithm

status	# new transactions	# transactions	proposed algorithm	Apriori
first run	10	10	0.5141	0.1730
frequent run	10	20	0.5341	0.4200
frequent run	50	70	0.9132	1.8826
frequent run	200	270	1.1587	3.6100
frequent run	400	670	1.3000	5.0012
frequent run	600	1270	2.6717	6.9812
frequent run	1200	2470	5.2821	14.9987
Average of frequent runs times			1.976633	5.482283
This is a decrease of				63.95%

According to the results shown in figure 7 that represents the relationship between number of generated sets and time in the ISB algorithm it can be concluded that there is a strong Positive relationship between the two factors and each time the user generate more sets the algorithm will take longer time.

Looking to the results shown in figure 8 that represents the relationship between the numbers of inserted transactions and time in the FU algorithm it can be said that there is no strong relationship between the two factors.

Looking to the results shown in figure 9 that represents the relationship between number of item sets and time in the FU algorithm there is a strong Positive relationship between the two factors. Also According to the results that shown in figure 10 between number of item sets and time in the RG algorithm there is a strong Positive relationship between the two factors.

If a comparison between the proposed algorithm and Apriori algorithm is made, it will be found that the proposed algorithm is better especially in the frequent runs and big data because it uses accumulated knowledge rather than working from scratch each time. As the results in table 1 shows that the average performance of frequent run is better that Apriori by 63.95%

In the initial implementation of Apriori algorithm, it seems good because of generating candidates from only frequent patterns but for dynamic uses it fails because each time it will generate item sets and tables which cost a lot of effort, time, and storage. On the other hand the proposed algorithm generates the entire possible candidates once in the initial stage.

Finally the proposed algorithm consists of built in rule generation method that is compatible with the tables that are created from the initial stage, while the Apriori algorithm is only for generating frequent sets which needs external method for association rule generation, which may be incompatible with the tables that are created from the initial stages.

8. CONCLUSION AND FUTURE WORK

This paper proposed a Dynamic Algorithm for Association Rules Mining in Big Data which aims to find useful relationships between unknown or hard to find items from a big database in efficient way without the need to repeat all the steps from scratch like what the classic algorithms do.

after comparing between the proposed algorithm and the Apriori algorithm it was found and proved that the proposed algorithm is better especially in the frequent runs and updated data because it uses accumulated knowledge rather than working from scratch each time like Apriori algorithm that fails because each time it will generate item sets and tables which costs a lot of effort, time, and storage.

The proposed dynamic algorithm was better than Apriori algorithm in frequent runs on large number of transactions by 63.95%while Apriori performed better in initial run. It can be conclude that this paper is a seed in the field of data mining and can be enhanced to perform better and more efficiently by using one or more of the following ideas:

- Using clustering for parallel item-set generation processes to decrease execution time.
- Implementing sorting algorithms and indexing techniques to enhance searching and counting methods.

9. ACKNOWLEDGMENTS

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

REFERENCES:

[1] Vanderveldt, Ingrid V., and Christopher Lee Black. "System and method for dynamic data-mining and on-line communication of customized information." *U.S. Patent No. 6,266,668*, Jul 2001.

[2] Michael Steinbach, Vipin Kumar, Pang-Ning Tan Introduction To Data Mining , second edition, *Pearson*, 2013.

[3] Bharti Thakur, Manish Mann , Data Mining for Big Data: A Review , *International Journal of Advanced Research in Computer Science and Software Engineering*, ijarcsse ,Volume 4, Issue 5, May 2014

[4] Ming-Syan Chen ,Jiawei Han,Yu, P.S. , Data mining: an overview from a database perspective, *Knowledge and Data Engineering, IEEE Transactions* Volume:8 , No. 6, 1996.

[5] Judith Hurwitz, Alan Nugent, Dr. Fern Halper,and Marcia Kaufman , Big Data For Dummies , *John Wiley & Sons*, 2013

[6] Kotsiantis, Sotiris, and Dimitris Kanellopoulos. "Association rules mining: A recent overview." *GESTS International Transactions on Computer Science and Engineering*, Vol. 32, No.1, pp.71-82, 2006.

[7] T. Karthikeyan and N. Ravikumar , A Survey on Association Rule Mining , *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 3, Issue 1, January 2014

[8] Wei Fan , Albert Bifet , Mining Big Data: Current Status, and Forecast to the Future , *SIGKDD Explorations* Vol. 14, Issue 2 , 2012.

[9] Agarwal, Ramesh C., Charu C. Aggarwal, and V. V. V. Prasad. "A tree projection algorithm for generation of frequent item sets." *Journal of parallel and Distributed Computing*, Vol. 61, No. 3, pp.350-371, 2001.

[10] Chen, M., Mao, S., Zhang, Y., & Leung, V. C. Big data: related technologies, challenges and future prospects, *Springer*, 2014

[11] Bill Sempf, Chuck Sphar,and Stephen Randy Davis , C# for Dummies, *John Wiley & Sons*, 2013.

[12] Hejlsberg, Anders, et al. C# Programming Language, *Addison-Wesley Professional*, 2010.

[13] Allen g.Taylor , Sql for dummies , 8th edition , *John Wiley & Sons*, 2013

[14] Brust, Andrew, Leonard G. Lobel, and Stephen Forte, Programming Microsoft SQL Server 2008, *Microsoft Press*, 2008.

[15] Huda Jalil, Mohammad Shkoukani, Suhail Owis, "A New Technique to Manage Big Bioinformatics Data Using Genetic Algorithms", *(IJARAI) International Journal of Advanced Research in Artificial Intelligence*, Vol. 5, No.6, pp. 1-6, 2016.

[16] Xiangyang, She, and Zhang Ling. "Apriori Parallel Improved Algorithm Based on MapReduce Distributed Architecture." Instrumentation & Measurement, Computer, *Communication and Control (IMCCC), 2016 Sixth International Conference on. IEEE*, 2016.