OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

MINING CONDENSED REPRESENTATIONS OF FREQUENT PATTERNS ON BIG DATA USING MAX APRIORI MAP REDUCING TECHNIQUE

Ms. Ansari Bushra Altaf[1], Prof. Harish K. Barapatre[2], Prof. Ankit Sangvi[3]

Dept. of Computer Engineering Alamuri Ratnamala Institute of Engineering and Technology University of Mumbai[1]
a_bushra2005@yahoo.co.in[1]

Dept. of Computer Engineering Alamuri Ratnamala Institute of Engineering and Technology University of Mumbai[2]
harishkbarapatre@gmail.com[2]

Dept. of Computer Engineering Alamuri Ratnamala Institute of Engineering and Technology University of Mumbai[3]
ankit.s.sanghvi1@gmail.com[3]

Abstract: ***Data Mining is one of the most important areas of pattern mining research that generates logical and informative information that applies to raw information. This function aims to extract a set of objects representing any type of sentence of definition of homosexuality and data familiarity. Our propulsion operations are based on a very powerful algorithm called Apriori algorithm and hadoop framework. The proposed system includes the Apriori MapReduce algorithms and the Apriori iterative Max Apriori based on a reduced map framework designed to extract common pattern mines from large databases. These algorithms extract a set of frequency data from any existing data set regardless of its frequency. Max Apriori Map Reduced Algorithm is our proposed algorithm that provides the most common object calculation compared to the big data set compared to the other three algorithms and with our algorithm we suggest removing the time required compared to the other Apriori Reduced Map and -iterative Apriori. The main motivation for this activity is to detect waves set using large amounts of data and to remove unwanted data from a set of large data sets. To evaluate the performance of the proposed algorithms, a separate data set is considered.***

Keywords— *Big Data, Hadoop, Data Mining, MaxApriori, Mapreduce, Matching.*

I INTRODUCTION

Pattern mining [2] is considered as an essential part of data analysis and data mining. Its aim is to extract subsequences, substructures or item-sets that represent any type of homogeneity and regularity in data, denoting intrinsic and important properties [3], [4]. This problem was originally proposed in the context of market basket analysis in order to find frequent groups [5] of products that are bought together [6]. Since its formal definition, at early nineties, a high number of algorithms have been described in the literature [7], [8], [9]. Most of these algorithms are based on Apriori-like methods [10], producing a list of candidate item-sets or patterns formed by any combination of single items. However, when the number of these single items to be combined increases, the pattern mining problem turns into an arduous task and more efficient strategies are required [9], [10]. To understand the complexity, let us consider a dataset comprising n single items or singletons. From them, the number of item-sets that can be generated is equal to $2n - 1$, so it becomes extremely complex with the increasing number of singletons. All of this led to the logical outcome that the whole space of solutions cannot always be analyzed.

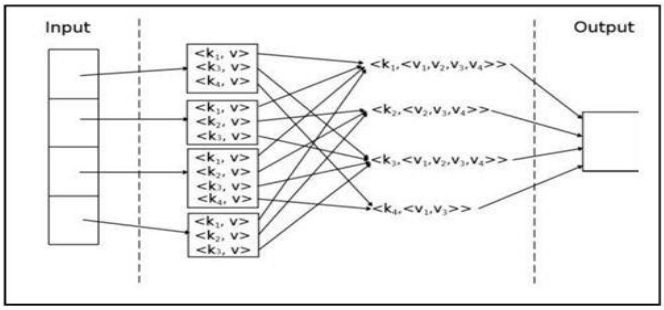

Figure No 1: A generic MapReduce framework

MapReduce [9] is an emerging paradigm that has become very popular for intensive computing. The programming model offers a simple and robust method for writing parallel algorithms. Some authors [4] have recently described the significance of the MapReduce framework for processing large datasets, leading other parallelization schemes such as Message Passing Interface (MPI). This emerging technology has been applied to many problems where computation is often highly demanding, and pattern mining is one of them. These methods were properly implemented by means of Hadoop [10], which is considered as one of the most popular open-source software frameworks for distributed computing on very large data sets which shown in figure no 1.

There are many MapReduce implementations [2], but Hadoop [1] is one of the most widespread due to its open source implementation, installation facilities and the fundamental assumption that hardware failures are common and should be automatically handled by the platform. Furthermore, Hadoop proposes the use of a distributed file system, known as Hadoop Distributed File System (HDFS). HDFS replicates file data into multiple storage nodes that can concurrently access to the data.

II. LITERATURE SURVEY

Temporal data contain time-stamping information that affects the results of data mining. Traditional techniques for finding frequent it emeses assume that datasets are static and the induced rules are relevant across the entire dataset. However, this is not the case when data is temporal. In this paper, we are trying to improve the efficiency of mining frequent it meets on temporal data. Since patterns can hold in either all or some of the intervals, we propose a new algorithm to restrict time intervals, which is called, frequent it emeses mining with time cubes [1]. Our focus is developing an efficient algorithm for this mining problem by extending the well-known a priori algorithm. The notion of time cubes is proposed to handle time hierarchies. This is the way by which the patterns that happen periodically, during a time interval or both, are recognized. A new density threshold is also proposed to solve the overestimating problem of time periods and also make sure that discovered patterns are valid [2]. We evaluate our algorithms via experiments. There are two naive strategies to solve this problem, namely preprocessing and post- processing. The post-processing methods have to generate and consider a huge number of candidate CARs while the performance of the pre-processing methods depends on the number of records altered out. Therefore, such approaches are time consuming [3] [4]. Our linear data structure enables us to compute a tight bound for powerful pruning and to directly identify high utility patterns in an efficient and scalable way, which targets the root cause with Aprior algorithms. Extensive experiments on sparse and dense, synthetic and real world data suggest that our algorithm is up to 1 to 3 orders of magnitude more efficient and is more scalable than the state-of-the-art algorithms. Parallel computers offer a potential solution to the computation requirement of this task, provided efficient and scalable parallel algorithms can be designed. Also presented the two new parallel algorithms for mining association rules [5]. The Intelligent Data Distribution algorithm efficiently uses aggregate memory of the parallel computer by employing intelligent candidate partitioning scheme and uses efficient communication mechanism to move data among the processors. The Iterative apriori algorithm can be used to extract the frequent pattern from the dataset. In this approach, candidate item sets are extracted from the initial dataset. The candidate item sets are generated from the previous iteration. The support count is calculated for each candidate item set. The support value is the frequency of items. The condense value should be calculated for finding the dependency between item sets. The threshold value is calculated and based on this value pruning is performed [6].

III.PROBLEM DEFINATION

Data analysis has a growing interest in many fields like business intelligence, which includes a set of techniques to transform raw data into meaningful and useful information for business analysis purposes. With the increasing importance of data in every application, the amount of data to deal with has become unmanageable, and the performance of such techniques could be dropped. The term Big Data [1] is more and more used to refer to the challenges and advances derived from the processing of such high dimensional datasets in an efficient way.

In many application fields [10], however, it is not required to produce any existing item-set but only those considered as of interest, e.g. those which are included into a high number of transactions. In order to do so new methods have been proposed, some of them based on the anti-monotone property as a pruning strategy. It determines that any sub- pattern of a frequent pattern is also frequent, and any super- pattern of an infrequent pattern will be never frequent. This pruning strategy enables the search space to be reduced since once a pattern is marked as infrequent, then no new pattern needs to be generated from it. Despite this, the huge volumes of data in many application fields have caused a decrease in the performance of existing methods.

Traditional pattern mining algorithms are not suitable for truly big data, presenting two main challenges to be solved: computational complexity and main memory requirements. A series of algorithms based on the MapReduce framework and the Hadoop open-source implementation have been proposed.

IV.PROPOSED SYSTEM

Considering previous studies and proposals [13], the aim of this work is to provide research community with new and more powerful pattern mining algorithms for Big Data. These new proposals rely on the MapReduce framework and the Hadoop open-source implementation, and they can be classified as:

A) No pruning strategy. Two algorithms (AprioriMR and AprioriMR) are properly designed to extract patterns in large datasets. These algorithms extract any existing item set in data regardless their frequency.

B) Pruning the search space by means of the anti- monotone property. Two additional algorithms (SPAprioriMR and TopAprioriMR) are proposed with the aim of discovering any frequent pattern available in data.

C) Maximal frequent patterns. A last algorithm (MaxAprioriMR) is also proposed for mining condensed representations of frequent patterns, i.e. frequent patterns with no frequent supersets.

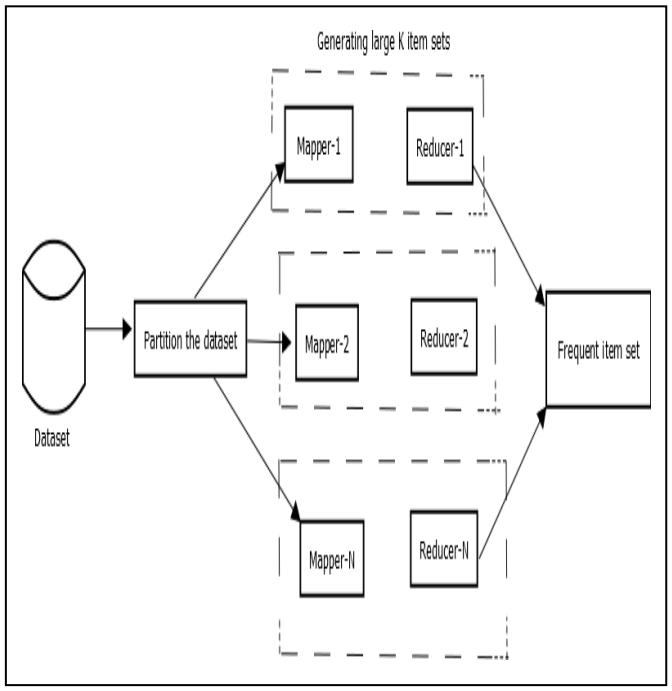

Figure No-2: System Architecture

A. Proposed System Architecture

In this system proposed new efficient pattern mining algorithms to work in big data. All of them rely on the MapReduce framework and the Hadoop open-source implementation. Two of these algorithms (AprioriMR and AprioriMR) enable any existing pattern to be discovered. Two additional algorithms (SPAprioriMR and TopAprioriMR) use a pruning strategy for mining frequent patterns. Finally, an algorithm for mining MaxAprioriMR is also proposed. It is worth mentioning that, when dealing with pattern mining on MapReduce, the number of key- value (k,v) pairs obtained might be extremely high, and the use of a single reducer as traditional MapReduce algorithms do can be a bottleneck. Under these circumstances, the use of MapReduce in pattern mining is meaningless since the problem is still the memory requirements and the computational cost. To overcome this problem, we propose the use of multiple reducers, which represents a major feature of the algorithms proposed in this work. Each reducer works with a small set of keys in such a way that the performance of the proposed algorithms is improved.

For the sake of reducing the time required by this intensive computing problem, we propose a new Apriori version based on MapReduce. This first model based on MapReduce, hereinafter called Apriori MapReduce (AprioriMR), mines the whole set of feasible patterns in the database. AprioriMR works by running a mapper for each specific sub-database and each of these mappers is responsible for mining the complete set of item-sets for each sub-database.

The task of finding all patterns in a database is quite challenging since the search space exponentially increases with the number of single items occurring in the database. The dataset might contain plenty of transactions and the process of calculating the frequency for each pattern might be considered as a tough problem [5]. All of this highlights the importance of pruning the search space, given rise to the paradigm of constraint-based mining [3]. A really Important pruning strategy in the pattern mining field is anti-monotone property, which determines that any sub-pattern of a frequent pattern is also frequent, and any super-pattern of an infrequent pattern will be never frequent.

In many situations, it is interesting to obtain a small set of frequent patterns instead of the whole set of them. In this regard it is possible to use the Apriori algorithm for mining maximal frequent patterns, also known as condensed representations of frequent patterns.

B.ALGORITHUM

In order to reduce the computational time required by the Apriori version, we present a novel proposal (see Algorithm

8) based on MapReduce paradigm. The proposed algorithm, called MaxAprioriMR (Maximal Apriori MapReduce), starts by mining the set of patterns P comprising patterns P of size |P| = s, in such a way that no super-pattern of size s+1 could be mined. For a better understanding,

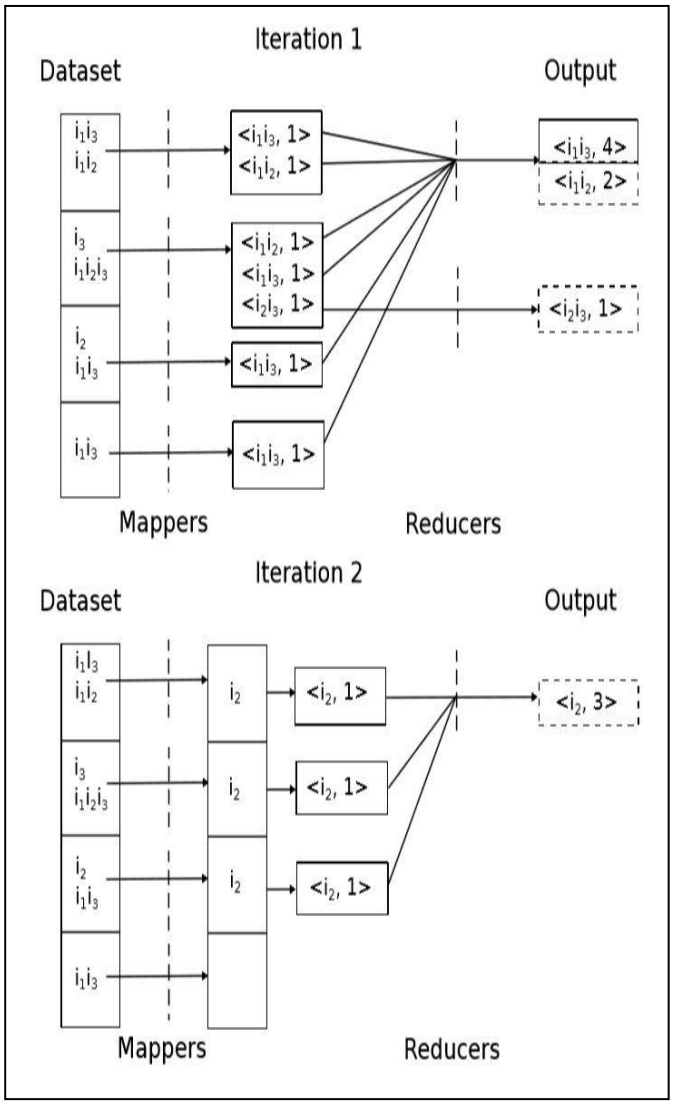

Figure No-3: Iteration on a sample dataset. The minimum support threshold = 4

Figure-3 shows the algorithm iterations for a sample dataset. Here, the minimum support threshold value is fixed to 4, so a pattern P is marked as maximal. Once a maximal frequent pattern of size |P| = s is obtained, then this pattern is kept into an external file that comprises the set of maximal frequent patterns discovered. This set of maximal frequent patterns is used in a second iteration, i.e., in the mining of those maximal frequent patterns of size |P| = s−1. In this algorithm, the aim of each mapper is twofold: first it removes any sub-pattern comprised into a maximal frequent pattern; second, it mines any existing frequent pattern of sizes−1from the resulting dataset this set of discovered frequent pattern is maximal since none of its previous super- patterns are frequent. Thus, the resulting set of maximal frequent patterns comprises the pattern {i1i3}, from which we could obtain the following frequent patterns

{{i1},{i3},{i1i3}}.

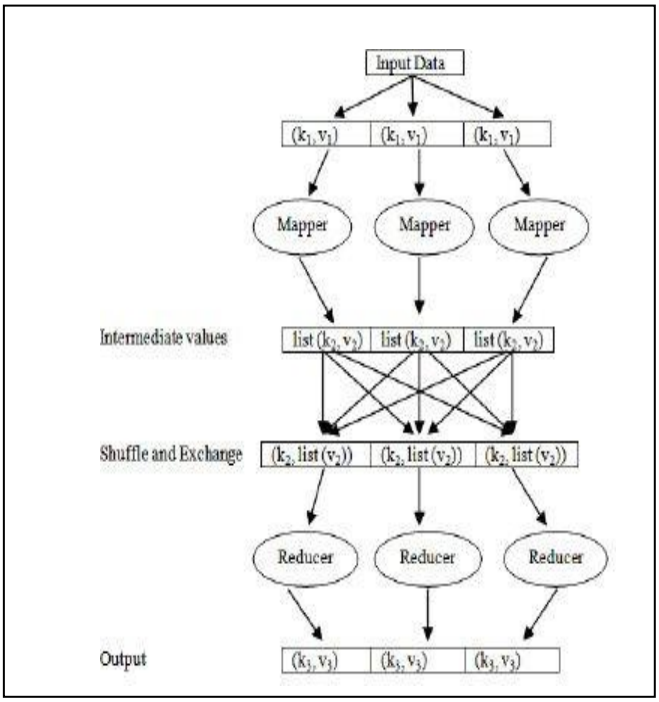

Figure No-4: Shuffle and Exchange with Mapper & Reducer

MapReduce is a parallel programming model designed for parallel processing of large volumes of data by breaking the job into independent tasks across a large number of machines which shown in figure-4. MapReduce is inspired form the list processing languages e.g. LISP. It uses two list processing idioms: map and reduce. Based on it a MapReduce program consists of two functions Mapper and Reducer which runs on all machines in a Hadoop cluster. The input and output of these functions must be in form of (key, value) pairs [3]. The Mapper takes the input (k1, v1) pairs from HDFS and produces a list of intermediate (k2, v2) pairs. An optional Combiner function is applied to reduce communication cost of transferring intermediate outputs of mappers to reducers. Output pairs of mapper are locally sorted and grouped on same key and feed to the combiner to make local sum. The intermediate output pairs of combiners are shuffled and exchanged between machines to group all the pairs with the same key to a single reducer.This is the only one communication step takes place and handle by the Hadoop MapReduce platform. There is no other communication between mappers and reducers take place.

A. Apriori Algorithm on Hadoop MapReduce

To implement an algorithm on MapReduce framework the main tasks are to design two independent map and reduce functions for the algorithm and to convert the datasets in the form of (key, value) pairs. In MapReduce programming, all the mapper and reducer on different machines execute in parallel fashion but the final result is obtained only after the completion of reducer. If algorithm is recursive, then we have

to execute multiple phase of map-reduce to get the final result. Apriori algorithm is an iterative process and its two main components are candidate item sets generation and frequent item sets generation. In each scan of database, mapper generates local candidates and reducer sums up the local count and results frequent item sets. The count distribution parallel version of Apriori is best suited on Hadoop whereas to implement data distribution algorithm we have to control the distribution of data which is automatically controlled by Hadoop. The complete iterative recursive generative 1 item set is shown in figure -5 with example.

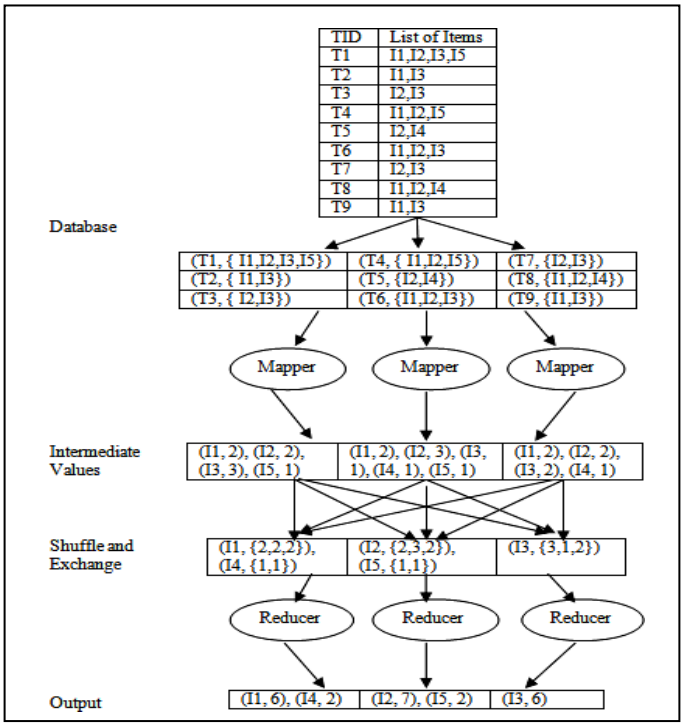

Figure No-5: Generation of Frequent 1-itemsets.

V. INPUT OUTPUT OF THE SYSTEM

Here we have used the different item set and processed by each algorithm using hadoop and used the different condition by using different size of data set which removed the frequent item set of the specified data which shown in Table no-1 that is Input Output for Different Algorithm.

Table No-1: Input Output for Different Algorithm.

CONDITION	INPUT	OUTPUT
Load Dataset	Retail dataset Data set	loaded successfully
AprioriMR	Transaction details	Hadoop Map reduce using Apriori
IApriori	Transaction details, chunk size	Frequent item sets
SPAprioriMR	Output of AprioriMR	Single items removed
TopAprioriMR	Transaction details with items, support, confidence	Frequent items with highest support values
MaxAprioriMR	Support, confidence, retail transaction Maximum	frequent item sets with highest support values
Min-Util	Transaction ids, product ids, product unit price, product utility	Min-util value to calculate Top-K high utility item sets

By using the above data set we proposed MaxAprioriMR base algorithm using the different size of data set. Here we have used amazons prime transaction data set which include the number frequent items.

VI. RESULT ANALYSIS

The below figure shows the comparative analysis by using the existing and proposed algorithm. Our proposed algorithm count the creates the minimum group of item set with less time as compared to other algorithm which mentioned in this graph analysis.

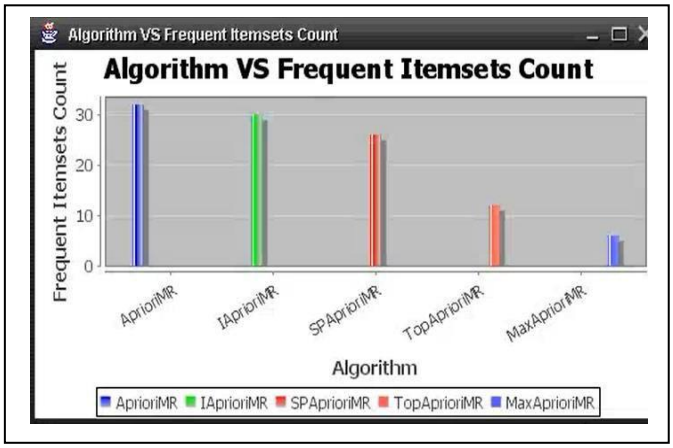

Figure No-6 : Comparative Analysis

MaxApriori is the proposed Algorithm is depends on number of transaction, how many transaction included in one data set

which might be reduced the transaction speed and easily identified the frequent item set as per figure no-5. Every transaction need the intermediate value that, if the intermediate value and transaction found than shuffle was mapping to each transaction which creates the chunk data for each transaction and exchange program is reduced the transaction by merging all chunk files and reduced the number of transaction.

VII. CONCLUSION

In this paper, we have proposed new efficient pattern mining algorithms to work in big data. All the proposed models are based on the well-known Apriori algorithm and the MapReduce framework. The proposed algorithms are divided into three main groups. The experimental stage includes comparisons against both well-known sequential pattern mining algorithms and MapReduce proposals. The ultimate goal of this analysis is to serve as a framework for future researches in the field. Results have demonstrated the interest of using the MapReduce framework when big data is considered. They have also proved that this framework is unsuitable for small data, so sequential algorithms are preferable. No pruning strategy. Two algorithms (AprioriMR and IAprioriMR) for mining any existing pattern in data have been proposed. Pruning the search space by means of anti-monotone property. Two additional algorithms (SPAprioriMR and TopAprioriMR) have been proposed with the aim of discovering any frequent pattern available in data. Maximal frequent patterns. A last algorithm (MaxAprioriMR) has been also proposed for mining condensed representations of frequent patterns.

REFERENCES

[1]Mazaher Ghorbani and Masoud Abessi [2018] “A New Methodology for Mining Frequent Itemsets on Temporal Data” in IEEE Transactions on Engineering Management (Volume: 64 , Issue: 4 , Nov. 2018).

[2] Dang Nguyen, Loan T.T. [2016] “Efficient Mining of Class Association Rules with the Itemset Constrain” in Knowledge-Based Systems –Science Direct, Volume 103, 1 July 2016, Pages 73-88

[3] Junqiang Liu [2016] “Mining High Utility Patterns in One Phase without Generating Candidates” in IEEE Transactions on Knowledge and Data Engineering.

[4] Lakshmi Narayanan [2014] “Frequent pattern mining on big data using Apriori algorithm” International Journal of Engineering Research & Technology (IJERT), February - 2014 IJERT ISSN: 2278-0181.

[5] George Karypis [2016] “Scalable Parallel Data Mining for Association Rules” in IEEE Transactions on Knowledge and Data Engineering (Volume: 12 , Issue: 3 , May/Jun 2016).

[6] J. M. Luna, Pattern mining: Current status and emerging topics, Progr. Artif. Intell., vol. 5, no. 3, pp. 165170, 2016.

[7] C. C. Aggarwal and J. Han, Frequent Pattern Mining, 1st ed. Cham, Switzerland: Springer, 2014.

[8] J. Han, H. Cheng, D. Xin and X. Yan, Frequent pattern mining: Current status and future directions, Data Min. Knowl. Disc., vol. 15, no. 1, pp. 5586, 2007.

[9] J. M. Luna, J. R. Romero, C. Romero, and S. Ventura, On the use of genetic programming for mining comprehensible rules in subgroup discovery, IEEE Trans. Cybern., vol. 44, no. 12, pp. 23292341, Dec. 2014. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2014.2306819.

[10] L. T. Nguyen, B. Vo, T. -P. Hong, and H. C. Thanh, Car-miner: An efficient algorithm for mining class-association rules, Expert Syst. Appl., vol. 40, no. 6, pp. 23052311, 2013.

[11] D. Nguyen, B. Vo, and B. Le, CCAR: An efficient method for mining class association rules with itemset constraints, Eng. Appl. Artif. Intell., vol. 37, pp. 115124, 2015.

[12] Y. -L. Chen, K. Tang, R. –J. Shen, and Y. -H. Hu, Market basket analysis in a multiple store environment, Dec. Support Syst., vol. 40, no. 2, pp. 339 354, 2005.

[13] Y.-L. Chen, T. C.-K. Huang, and S.-K. Chang, A novel approach for discovering retail knowledge with price information from transaction databases, Expert Syst. Appl., vol. 34, no. 4, pp. 23502359, 2008.

[14] M. Shaheen, M. Shahbaz and A. Guergachi, Context based positive and negative spatio-temporal association rule mining, Knowl.-Based Syst., vol. 37, pp. 261273, 2013.

[15] X. Wu, C. Zhang, and S. Zhang, Efficient mining of both positive and negative association rules, ACM Trans. Inf. Syst., vol. 22, no. 3, pp. 381405, 2004.