International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.1, Issue1, pp-084-089 ISSN: 2249-6645

Implementation of Object Oriented Approach To Sequential Pattern Mining From Multidimensional Sequence Data

K.HEMALATA
2/2 M.TECH CSE, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, TEKKALI
ANDHRA PRADESH, INDIA

G.VASANTHAKUMARI
ASSOC. PROFESSOR
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, TEKKALI
ANDHRA PRADESH, INDIA

Abstract— **In recent years, emerging applications introduced new constraints for data mining methods. These constraints are typical of a new kind of data. Sequential pattern mining is applicable in this area, since many types of data sets are in a time related format. Besides mining sequential patterns in a single dimension, mining multidimensional sequential patterns can give us more informative and useful patterns. Due to the huge increase in data volume and also quite large search space, efficient solutions for finding patterns in multidimensional sequence data are nowadays very important. For this reason, in this paper, we present a multidimensional sequence model, Simulation experiments show good load balancing and scalable and acceptable speedup over different data sets and problem sizes.**

***Index Terms*— Data Mining, Sequential Patterns, Sequence Data.**

I. INTRODUCTION

Data mining has been defined as "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data." Mining frequent patterns or itemsets is a fundamental and essential problem in many data mining applications. These applications include the discovery of association rules, strong rules, correlations, sequential rules, episodes, multidimensional patterns, and many other important discovery tasks [1]. The architecture of a typical data mining system may have the following major Components:

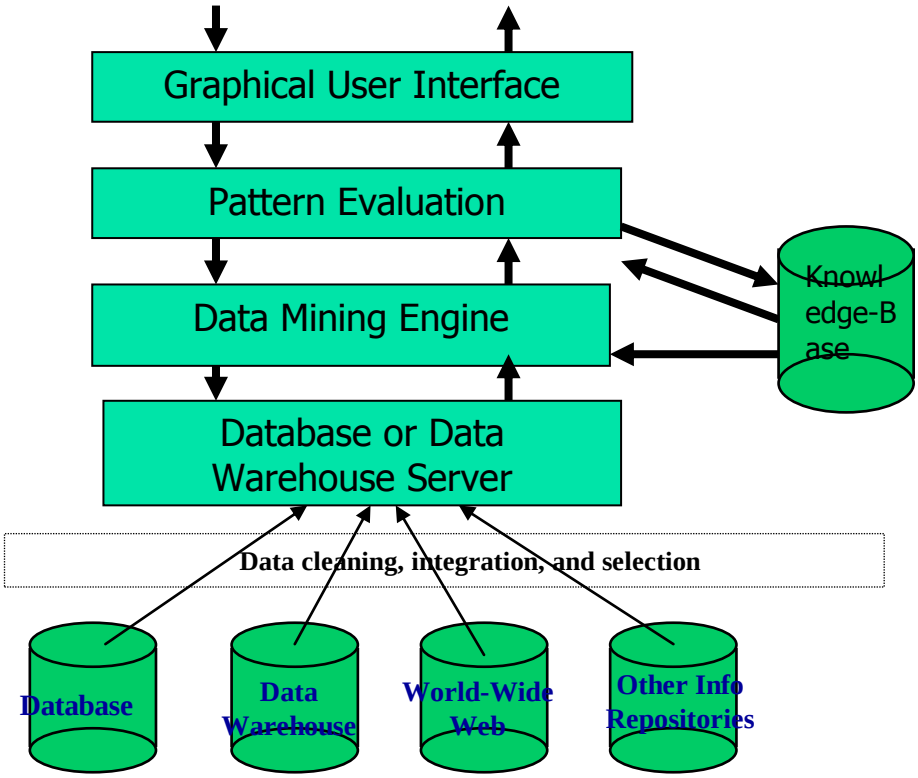

Fig. 1. Data mining Architecture

- **Database, Data warehouse, or other Information repository:** This is one or a set of databases, data warehouses, spreadsheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.
- **Database or Data warehouse server:** The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request.
- **Knowledge base:** This is the domain knowledge that is used to guide the search, or evaluating the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attribute or attribute values into different levels of abstraction.
- **Data mining engine:** This is essential to the data mining system and ideally consists of a set of modules for tasks such as characterization, association, classification, cluster analysis, and evolution and deviation analysis.
- **Pattern evaluation module:** This component typically employs interest measures and interacts with the data mining modules so as to focus the search towards interesting patterns.
- **Graphical user interface:** This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, performing exploratory data mining based on the intermediate data mining results.

One of the most important data mining problems is discovery of frequently occurring patterns in sequence data [2]. There are many domains where sequence mining has been applied, which include analysis of telecommunication systems, discovering customer buying patterns in retail stores, analysis of web access databases, and mining DNA sequences and gene structures.

Usually, sequence patterns are associated with different circumstances, and such circumstances form a multiple dimensional space. For example, bank customer sequences are associated with credit, branch, age, and others. It is interesting and useful to mine sequential patterns associated with multidimensional information [3].

Most of the proposed pattern mining algorithms are a variant of Apriori. Apriori-based algorithms show good performance with sparse datasets such as market-basket data, where the frequent patterns are very short. However, with dense data sets such as telecommunications, where there are many and long frequent patterns, performance of these algorithms degrades incredibly.

On the one side, parallel and distributed computing is expected to relieve current mining methods from the sequential bottleneck, providing the ability to scale to massive datasets, and improving the response time. Achieving good performance on today's multiprocessor systems is a non-trivial task. The main challenges include synchronization and communication minimization, work-load balancing, find good data layout and data decomposition, and disk I/O minimization, which is especially important for data mining [4].

On the other side, the basic single dimension cannot satisfy the requirement of multi-attribute analysis, which is often the case in actual system practice. To address this problem, multidimensional sequence pattern mining is developed.

The most time consuming operation in the discovery process of sequential patterns is the computation of the frequency of the occurrences of interesting subsequences in the sequence database. However, the number of sequential patterns grows exponentially and the task of finding all sequential patterns requires a lot of computational resources, which make it an ideal candidate for parallel processing.

In this paper, we present a model for multidimensional sequence pattern and then propose mining sequential patterns from multidimensional sequence data. The experimental results show that our procedure usually achieves good load balancing and scalability.

II. SEQUENCE MINING AND RELATED WORK

The problem of mining frequent patterns in a set of data sequences together with a few mining algorithms was first introduced in [2]. They also presented three algorithms for solving this problem. The sequence mining task is to discover a sequence of attributes, shared across time among a large number of objects in a given database. Many studies have been contributed to the efficient mining of sequential patterns in the literature, most of which was focused on developing efficient algorithms for finding all sequential patterns such as , GSP [5], SPADE [6], Prefix Span [7] and so on. In addition, enormous sizes of available databases and possibly large number of mined sequential patterns demand efficient and scalable parallel algorithms.

Other work contributes on an extension of the problem of sequential pattern mining like constraint-based in sequential pattern mining, mining cyclically repeated patterns, approximate mining of consensus sequential patterns, mining top-k closed sequential patterns, mining frequent max sequential patterns, mining long sequential patterns in a noisy environment, mining hybrid sequential patterns and sequential rules, etc. Mining multidimensional sequential pattern is also one of the central topics in sequence mining as showed by the research efforts produced in recent years [8,9,10].

III. SYSTEM ARCHTECTURE

In this section we provide the step by step procedure for our proposed method, ask the input(file) based

on input file system will try to retrieve the data, consider the minimum support and confidence after that generate the frequent itemsets. After wards it will generate the support tree, it will a follow a set of procedures finally generate association rules and finally display the items sets and AR rules as shown in the fig 2.

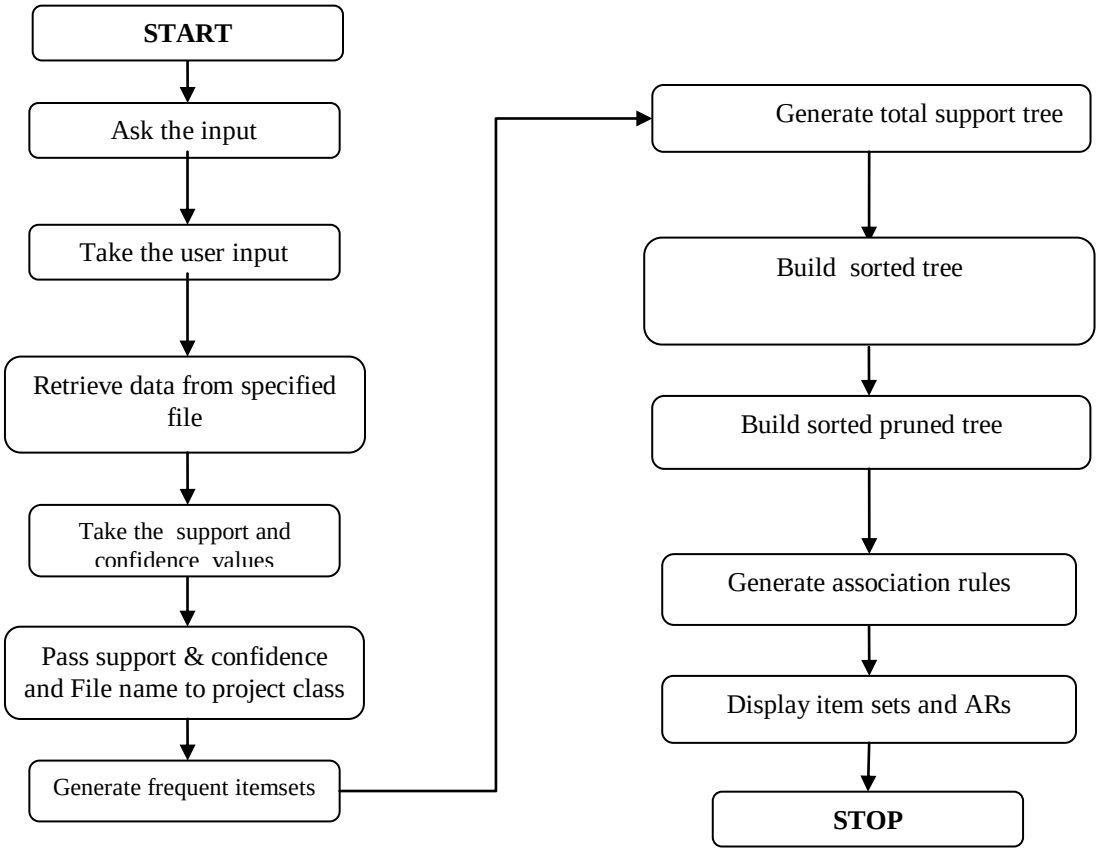

Fig 2 System Architecture

IV. MODULE DESCRIPTION AND IMPLEMENTATION

Design is the power to think, plan, and realize products that serve to accomplishment of any purpose. It is the process or art of defining the components, modules, interfaces, and data for a computer system to satisfy specified requirements. Here the system is divided into three modules.

A. Module1: Discovering the Apriori-T with X-Cheking.

To discover the patterns from the multi-dimensional sequence data. Each sequence pattern check with all possible sequence patterns based on the AprioriMD algorithm. First of all it will take only some records in the database according to the support and confidence levels. It will take the patterns in the tree levels and calculated no of records from each and every individual records and different combinational sequence patterns. It will display only the minimum threshold level.

B. Module2: Discovering Frequent Item Sets

This module deals to find out the most frequently accessed records or web pages. Customer has been visiting same sites in different sessions. From that each session it will calculate the no of frequent records or items. Each sequence pattern stored in the database as tree structure. These sequence patterns kept in the levels of the tree.

C. Module3: Calculating the .Generate ARS

To generate the most frequently accessed records average to be calculated and give the percentage of used records. Using the all records in the database it will generate rules based on the support and confidence level. It has been pick up the records from above the support and confidence percentage.

V. IMPLEMENTATION

A *platform* is the hardware or software environment in which a program runs. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms. In this paper we generate the rules using an object oriented language called JAVA 1.6.

A. FREQUENT ITEMSET

Discovery of frequent item sets in transactional databases is a well-studied data mining topic. Given a database of transactions D and a collection of items L, an item set I belongs to L is frequent if the number of transactions in D that contain I (i.e., the support of I) is at least minimum support.

First, the input data set is scanned once to find all frequent 1-itemsets. These are sorted in descending order of their frequencies. Then, the data set is scanned again to construct an FP-tree. The FP-tree is a compact data structure that stores all patterns which appear in the database. For each transaction, the frequent items (in descending frequency order) are inserted as a path in this tree, where each node corresponds to an item and paths with common prefixes are compressed. An auxiliary structure, called header table, is used to link nodes with the same label. An algorithm, called FP-growth, is used to mine frequent patterns from the FP-tree.

B. AprioriMD:

AprioriMD, for mining the tree structure is a recursive procedure during which many sub trees and header tables are created. The process commences by examining each item in the header table, starting with the least frequent. For each entry the support value for the item is produced by following the links connecting all occurrences of the current item in the tree. If the item is adequately supported, then for each leaf node a set of *ancestor labels* is produced (stored in a *prefix tree*), each of which has a support equivalent to the sum of the leaf node items from which it is generated. If the set of ancestor labels is not `null`, a new tree is generated with the set of ancestor labels as the dataset, and the process repeated. In our implementation all frequent item sets thus discovered were placed in a T-tree, thus providing fast access during the final stage of the association rule mining (ARM) process, while at the same time providing for the deletion of sub trees and tables created "on route" as the AprioriMD algorithm progressed.

In AprioriMD, for mining sequential patterns from a multidimensional sequence database by modifying the Apriori or GSP algorithms. In the presentations afterward, we assume that the sequences are represented by simplified format. The algorithm is similar to the GSP algorithm. However, there are two differences:

1) The method to generate Ck and

2) The method to compute the supports of candidate sequences.

The following discusses how to handle these differences. We first discuss how to generate C2. Then, we discuss how to generate Ck, where k > 2. Traditionally, C2 can be obtained by joining L1 with L1 directly. However, since the first element and the second element in C2, say b and c, may have different dimensional scopes, we need to generate the pairs for all possible scope relations.

C. ASSOCIATION RULE MINING

The T-tree (Total Support Tree) is a compressed set enumeration tree structure. The T-tree differs from more standard set enumeration trees in that the nodes at the same level in any sub-branch are organized into 1-D arrays such that the array indexes represent column numbers.

For this purpose it is more convenient to build a "reverse" version of which permits direct indexing with attribute/column numbers. The T-tree offers two initial advantages

1. Fast traversal of the tree using indexing mechanisms,
2. Reduced storage, in that item set labels are not required to be explicitly stored, and thus no sibling reference variables (pointers) are required.

As each level is processed, candidates are added as a new level of the T-tree, their support is counted, and those that do not reach the required threshold of support are subsequently pruned. When the algorithm terminates, the T-tree contains only the large item sets. At each level, new candidate `K` item sets are generated from identified large `K-1` item sets, using the *downward closure property of item sets*, which in turn may necessitate the inspection of neighboring branches in the T-tree to determine if a particular `K-1` subset is supported. We refer to this process as *X-checking*. Note that X-checking adds a computational overhead; offset against the additional effort required to establish whether a candidate `K` item set, all of whose `K-1` item sets may not necessarily be supported, is or is not a large item sets . In some cases it is more expedient to assume that those subsets of a candidate `K` item sets that are contained in neighboring branches of the T-tree are supported than to carry out X-checking. Results are shown in the Fig 3,4,5.

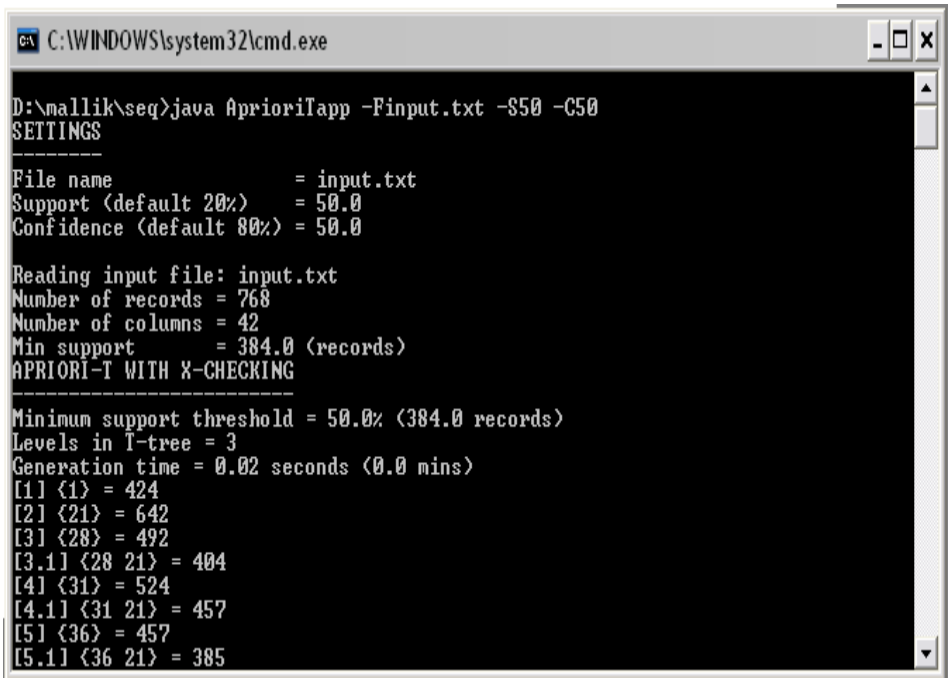

Fig 3 Input Screen

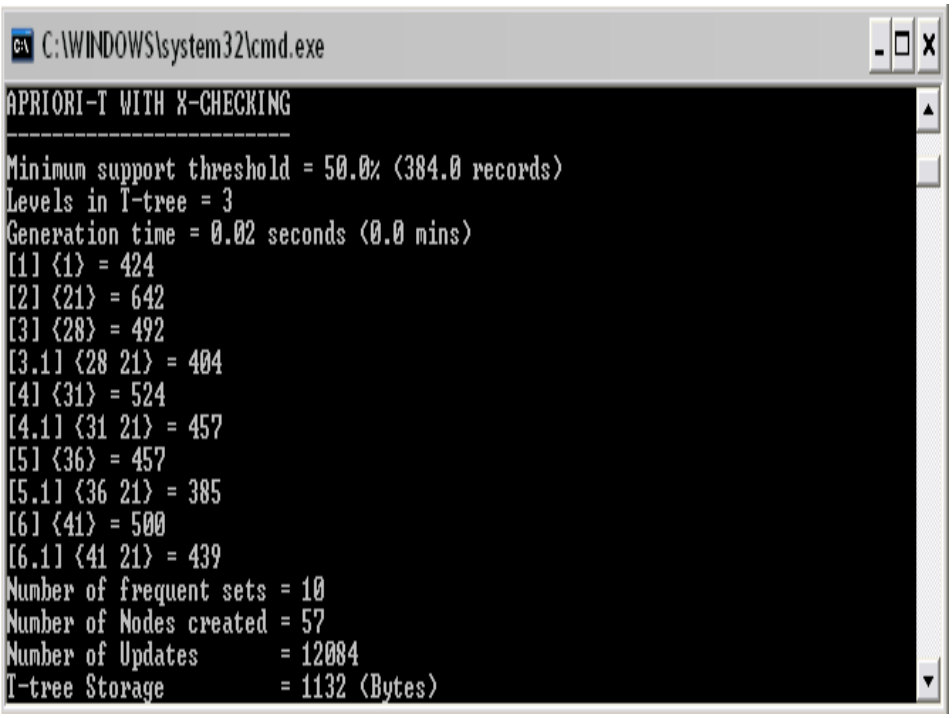

Fig 4 Frequent sets displayed screen

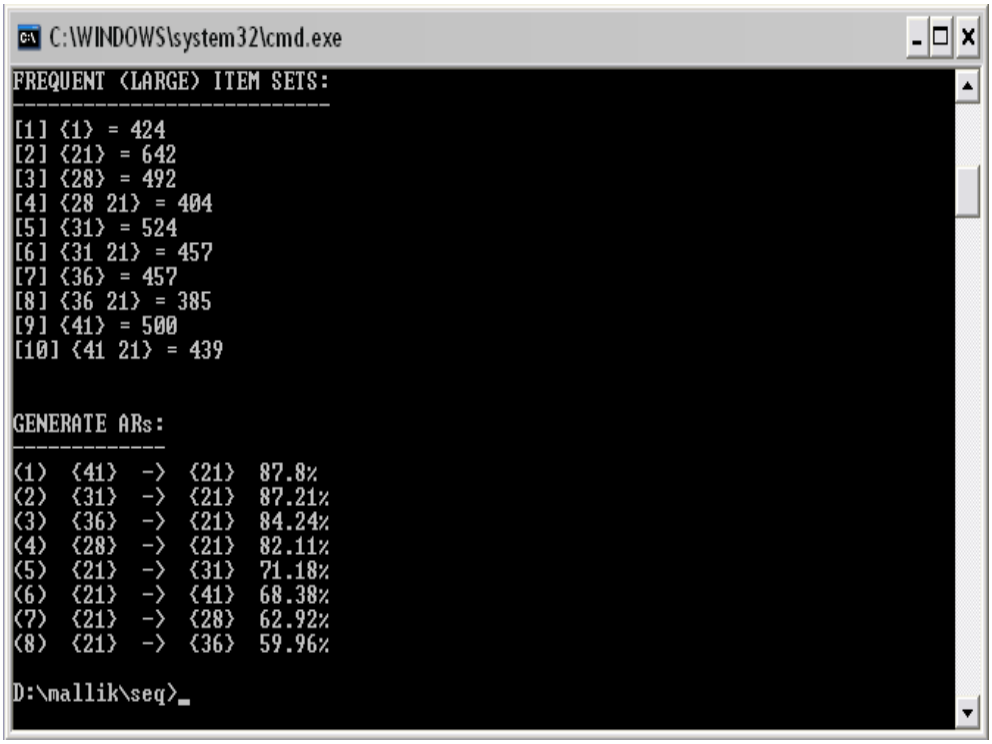

Fig 4 AR rules displayed screen

VI. Conclusions and Future work

Multidimensional mining has been attracting attention in recent research into data mining. Very large search space and data volume have made many problems for mine sequential patterns. In order to effectively mine, efficient parallel algorithm is necessary. In this paper, we theoretically present a multidimensional sequence model and then use this strategy to multidimensional sequential pattern mining.

In this paper we developed a modification of the well-known Apriori algorithm for mining sequential patterns from multidimensional sequence data. The simulation results show that the latter is better than the former.

Future work

Scalability is important for any system; various combinations of algorithms may be used for achieving better result. The accuracy of the mining may be improved by preprocess the data before analysis. So in this way there is scope to the future enhancements.

References

[1] J. Han, and M. Kamber, Data Mining: Concepts and Techniques, Academic Press, New York, 2001.

[2] R. Agrawal, R. Srikant, Mining Sequential Patterns, Proc. of the 11th Int'l Conference on Data Engineering, 1995.

[3] J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal, Multi-dimensional sequential pattern mining, Conference on Information and Knowledge Management, pp: 81–88, 2001 .

[4] M. J. Zaki, C.-T. Ho, Large scale parallel data mining, Lecture notes in computer science, Vol 1759: Lecture notes in artificial intelligence, Springer, 2000 .

[5] R. Srikant, R. Agrawal, Mining sequential patterns: generalizations and performance improvements, In Proc. of the 5th Int'l Conference on Extending Database Technology, pp: 3–17, 1996 .

[6] M. J. Zaki, An efficient algorithm for mining frequent sequences, Machine Learning Journal, vol. 42(1/2), pp: 31–60, 2001 .

[7] J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal, MC. Hsu, PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth, In Proc. of Int'l Conference on Data Engineering, pp: 215–224, 2001.

[8] S. de Amo, D. A. Furtado, A. Giacometti, and D. Laurent, An apriori-based approach for first-order temporal pattern mining, In Simposio Brasileiro de Bancos de Dados,2004.

[9] Plantevit M., Choong Y.W., Laurent A., Laurent D., Teisseire M. , M2SP: Mining Sequential Patterns Among Several Dimensions, Principles of Knowledge Discovery in Databases, PKDD, Volume 3721, pp: 205-216, 2005 .

[10] C.-C. Yu, and Y.-L. Chen, Mining Sequential Patterns from Multidimensional Sequence Data, IEEE Transactions on Knowledge and Data Engineering archive, Volume 17, Issue 1, pp: 136-140, 2005.