Mining Frequent Itemset on Temporal Data using Hybrid Algorithm

A. Monika
CSE Department
TRP Engineering College
Trichy,TamilNadu

A. Niveda Rajlakshmi
CSE Department
TRPEngineering College
Trichy,TamilNadu

M. Pavithra
CSE Department
TRP Engineering College
Trichy,TamilNadu

S K. Karthika
Assistant Professor
CSE Department
TRP Engg College
Trichy, TamilNadu

***Abstract*—Temporal data are data with time-stamping information.The result of data mining also depends on time stamp information.Traditional techniques assume that datasets are static and the rules being used are relevant for the entire dataset. In this paper, we are trying to improve the efficiency of mining frequent itemsets on temporal data. We propose a new algorithm to restrict time intervals, which is called frequent itemset mining with time cubes. Our focus is developing an efficient algorithm for this mining problem by extending the well-known apriori algorithm along with FP Growth algorithm.The notion of time cubes is proposed to handle time hierarchies.**

I.INTRODUCTION

DATA MINING is the process of discovering interesting patterns and knowledge from voluminous amounts of data [1]. One of the most important applications of data mining is the analysis of transactional data. Databases which originate from transactions in a supermarket, bank, department stores and, etc., are all inherently related to time.These are called temporal databases which are databases that contain time-stamping information [3].Capturing the co-occurrence of items in transactions was first proposed by Agrawal et al. [2]. For example, given a transactional database of a supermarket, we may have {milk,bread} bought together with support of 20%. It means that 20% of all transactions contain milk and bread together..One important extension to frequent pattern mining is to include a temporal dimension [4]. For example, milk and bread may be ordered together in 80% of all transactions between 7 and 9A.M while their support in the whole database is 20%.Infact, interesting patterns are often related to the specific period of time; therefore, the time during which they can be observed is important.

ID	ITEMS
1	bread,milk
2	milk,rusk
3	milk,biscuit

From the above,the conclusion is we can obtain different pattern for different time intervals. Discovering such patterns may lead to useful knowledge. The discovery of such association rules has been discussed in the literature. In this context, sequential association rules [5], time intervals for association rules [6], [7] and calendar-based association rules [4], [8] are some interesting studies in recent years.

In this paper, we conduct a study on developing an efficient algorithm for mining frequent patterns and their related time interval from transactional database. Then a new algorithm is proposed based on two thresholds, support, and density as a novel threshold. Frequent itemsets are discovered and neighbouring time intervals are merged.

The rest of the paper is organized as follows. In Section II, we discuss some related work in comparison with our work. In Section III, the proposed algorithm is presented in details. The notion of TCs is described and the essence of a density threshold is illustrated with an example. In Section IV, we present the experimental evaluation of our algorithms using synthetic datasets and give analysis on experimental results. In Section V, we conclude the paper with some discussions.

II. RELATED WORK

Association rule was first proposed by [2]. It has two part, finding frequent itemsets and generating association rules. The major and time consuming part of the algorithm is discovering frequent itemsets and generating association rules is straightforward. Therefore, in our literature review, we consider association rules the same as frequent itemset mining. Classification association rule uses class tables and maximum support count which is efficient and eliminates infrequent item sets. But duplicate data elimination is difficult [9]–[11], context-based association rule uses association rule and accuracy, frequency and performance is high but sequential and parallel construction of table is difficult due to time factor [12]–[14],negative association rule[15],fuzzy association rule [16], generalized association rule [17]–[19] are some of the research areas in this field. Among these extensions, the time attribute of the transactions has attracted many researchers to discover frequent itemsets over time omputational time is low.

The problem of discovering association rule that display regular cyclic variation overtime was first proposed by Ozdenetal. [20]. Two new algorithms were presented, sequential algorithm and interleaved algorithm, to discover hourly, daily, weekly, etc., patterns. Some pruning techniques also used to improve the performance of the algorithms. It should be noted that by their method, each cyclic rule holds in every cycle with no exception. However, in real life, patterns are not perfect. Therefore, Han et al. [21] proposed partial periodic pattern mining in temporal database. Discovered association rules show behavior in some but not all points in time. The work of Ozdenetal in[20]was extended by Ramaswamy et al. in [22] to discover user-defined temporal patterns in association rules. The idea of using calendar algebra was proposed to describe interesting patterns, however, it requires user's prior knowledge to define calendar expressions.In[6],Ale and Rossi proposed the idea of extracting rules over a specific period of time that is shorter than the whole database.The lifetime of each item was used to define time intervals. The concept of temporal support was introduced for the first time and algorithm a priori was modified to incorporate time. Li et al. [4], [23] proposed an approach to discover association rule holds either in all or some time intervals. Instead of using cyclic [20] or user-given calendar algebraic expressions [22], calendar schema is used to restrict the meaningful time intervals. Since items have different exhibition periods, algorithm progressive-partition-miner (PPM) was proposed to discover patterns in such databases [24], [25]. The basic idea of PPM is to first partition the database in light of exhibition periods of items and then progressively accumulate the occurrence count of each candidate 2-itemset based on the intrinsic partitioning characteristics. It's worth noting that exhibition period of items in[24]and[25]is the same as lifetime of items in[6].The study of [24] and [25] was further extended [26], considering the importance of time intervals. Using the same concept proposed in [6] and [24], algorithm segmented-progressive-filter (SPF) was introduced in [27] to first segment the database into sub databases in such a way that items in each sub database will have either starting time or ending time. Then, for each subdatabase, SPF progressively filters candidate 2-itemsets with cumulative filtering thresholds either forward or backward in time. Lee and Lee [28] also used calendar algebra to discover association rules.Since human being tends to be uncertain,fuzzy set theory incorporated to help the construction of desired time intervals. This is similar to the work of Ramaswamy et al. [22], but it is more flexible which is because of the fuzzy sets and fuzzy operators. Junheng[29]added some modification to the algorithm SPF [27] to increase efficiency. Results show that its performance outperform from prior methods. Huang et al. [30] explore the previous studies [27], [29] to remedy drawbacks of their algorithms. Algorithm TWAIN was presented to find association rules that are absent when the whole range of the database is evaluated altogether.Mahantaetal.[8]presented a method which is able to extract different types of periodic patterns that may exist in a temporal dataset.The user do not need to specify the periods in advance. Set operation called set superimposition was used for storing periods associated with item-sets. Adhikari et al. [31] enhanced the study of [8] by proposing a new data structure for storing and managing patterns. They also introduced minor changes to their algorithm to improve its performance. Lee et al. [32] proposed an efficient algorithm for discovering fuzzy periodic association rules. They investigated the problem of discovering regular time intervals, i.e., the periodicity. To overcome the difficulties of finding precise time interval, fuzzy calendar was proposed. Their method scan the database at most twice for discovering association rules and related time periods. Matthews et al. [33] used genetic algorithm to find temporal association rules for the first time. Genetic algorithm was employed to simultaneously search the rule space and temporal space. They used the strength of evolutionary algorithms in searching for association rules and optimizing parameters of induction process (support and confidence values). They further extended their works for finding fuzzy temporal association rules where Duplication can be minimised to very little amount and provides good response time [34], [35]. The problem of it is it doesn't handle very large number of data.

[37] proposed a new form of association rule, i.e., association rule with time windows. The main purpose of their study was to find the time intervals for association rules which may be arbitrary in length and not user specified. They further optimized the process of finding time windows by mathematical modeling [7]. Saleh and Masseglia [38] deal with this problem from another viewpoint. The concept of solid itemset mining was proposed to find the subsets of database that contain frequent itemsets and then the proposed algorithm was developed based on that .The problem is that Usage of tree structure and other nodes uses more memory.Like above studies, the challenge was to find the optimal time interval.

III. MINING FREQUENT PATTERNS WITH TCS

A. *New Structure*

Let I ={i1,i2,...in}be a set of item and D be a database of transactions.If the support of an itemset is greater than or equal to the given threshold value,the itemset is called a frequent itemset.

Each transaction tr is associated with an identifier transaction ID (TID), a time stamp ttr and a set of items. For all transactions, ttr ∈ T where T is the total time span of the database.Let Ts,Te∈Twhere Ts is the start time and Te is the end time be the time interval in each time hierarchy and also it is clear that Ts <Te. For example, (5, 9)Month shows the time interval between 5th and 9th months. In order to deal with temporal parts of the transactions,Timecube is defined.Cubic structure for time hierarchy helps us to easily merge neighboring frequent itemsets to find different temporal patterns. Each TC represents a time interval with different time hierarchies. Fig. 1 is a schematic of a partitioned database for three time hierarchies. For time hierarchies more than three , the term hypercube is used instead. Knowing time hierarchy and its domain, an equal-length partitioning is employed according to the minimum interval defined by the user. These initial cubes are called Basic Time Cubes (BTCs). Since rules can occur during a very narrow time interval, it is important to specify a minimum time

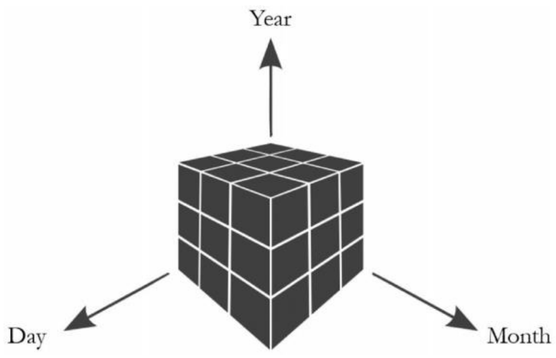

Fig. 1. Example of three time hierarchies shown as cubes

interval to initially partition the database. Considering time hierarchyshowninFig.1,one can consider one hour,one day,one month to define a 1×1×1 BTC. Any two elements in BTC cannot be overlapped such that BTCi ∩BTCj = ∅. Let |NCube| be the number of transactions occurred in the cube,i.e.,time interval,and|N(X)Cube|be the number of transactions containing itemsetX.The support of itemsetX is calculated as follows:

Support(X)=N(X)Cube/ |NCube|. (1)

According to (1), we can now define support of each frequent itemset as follows:

Support(X =⇒Cube Y)=N(X,Y)Cube/ |NCube|(2)

where|N(X,Y)Cube|is the number of transactions which contains both X and Y in that time interval.

Threshold is (minimum support) to evaluate items in dataset. Since records are not equally distributed in time intervals, very few
records may occur. Therefore, discovered
patterns may not be valid, since there is not enough evidence to show that they hold for the time interval. It also causes overestimating problem. In order to overcome these issues, another threshold which is called density is proposed to solve these problems. An illustrative example shows the necessity
of *density* (see Table I). Consider itemset *AB* with *Sup*(*A*,*B*) 9−11 = 8 12 ≥ 60%
Where, 60% is the minimum support. However, we can observe from the database that *AB* is only frequent during the period 9 to 10 rather than the whole range of 9 to 11, which is an overestimated time period. Density
not only ensure the validity of the patterns, but also filter out time intervals with few records which cause overestimating the time periods. The density of a time interval is calculated as follows:

$$A = n/NBTCs \quad (3)$$

$$Density = \alpha \times A \quad (4)$$

where *n* is the total number of records or transactions, *N*BTC*s* is the number of basic time cubes, therefore, *A* is the average transaction per BTC.

\`

TABLE 1
DATABASE TO SHOW DENSITY

TID	Time stamp	Itemset
1	25/07/2016-09:08:23	A B C
2	25/07/2016-09:14:09	A B C D
3	25/07/2016-09:22:40	A B
4	25/07/2016-09:24:12	B C D
5	25/07/2016-09:30:15	A B
6	25/07/2016-09:33:58	A B D
7	25/07/2016-09:42:02	A B
8	25/07/2016-09:43:56	A C D
9	25/07/2016-09:50:01	A B

A user-specified parameter $\alpha \in [0, 1]$ is
introduced to determine the desired density using equation 4. *A* is called density rate. For example, if a dataset contains 1000 records and 10 BTCs, average records per cube is 100. With parameter α = 0.5, cubes less than 50 records are filtered.Therefore, itemset is frequent if and only if for each TC, it satisfies the following conditions.

1) *X* =⇒Cube *Y* has a support greater than or equal to the *minimum support* threshold defined by the user.
2) The time interval (cube) must be dense to ensure the validity of the rules.

We use *a priori and association rule* to discover frequent itemsets in different time intervals. If an itemset X is frequent in a time interval.

1) All subsets of X are frequent in that time interval.
2) All subsets of interval contain frequent itemset.

B. Proposed Algorithm for Mining Frequent Itemsets

The objective of the proposed algorithm is to find itemset X in a contiguous subset of database, where the support of X is above the minimum support and the size of the time interval is optimal. We introduce density threshold where validity of the rules are ensured. Considering time hierarchy, our approach toward discovering frequent itemsets is to first partition the database into many small segments. We use cubes to show these segments. Candidates that have support more than the minimum support in at least one TC are considered to be frequent. Neighboring TCs of the same itemsets are merged if they are frequent. In the following, we present and explain our algorithm to mine frequent itemsets.FPgrowth is an algorithm for discovering frequent itemset in transaction database.It was proposed by Han et al(2000).
It is very fast and memory efficient algorithm and uses special internal structure called an FP-tree.

Algorithm 3.1 uses two procedures to find frequent itemsets. It is common in association rule mining, given a set of itemsets , the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Algorithm 3.1: Algorithm for mining frequent itemsets with TCs.

```
procedure  Apriori(T, minSupport) { //T is the database and minSupport is the minimum support
L1= {frequent items};
for(k= 2; Lk-1!=∅; k++) {
Ck= candidates generated from Lk-1
//that is cartesian product Lk-1x Lk-1
 and eliminating any k-1 size itemset that is not //frequent
for each transaction t in database
 do{
  #increment the count of all candidates in Ck
 that are contained in t
 Lk= candidates in Ck with minSupport
  }//end for each
 }//end for
 return ∪ₖ++;
```

Algorithm 3.2: Algorithm for mining frequent itemsets using FP-tree .

```
Input:
      §D, a transaction database
      §min_sup, the minimum support count threshold
Output: the complete set of frequent patterns.
Method:
(1)the FP-tree is constructed
(2) The FP-tree is mined by calling FP-growth(FP_tree,null):
procedure FP_growth(Tree,α)
if Tree contains a single path P then
  for each combination (denoted as β) of the nodes in the path P
   generate pattern βUαwith support_count = minimum support count of nodes in β;
else for each ai in the header of Tree{
generate pattern β=aiUαwith
support_count=ai.support_count
construct β's conditional pattern base and then
β's conditional FP_tree Treeβ;
if Treeβ!= 0 then
 call FP_growth(Treeβ,β); }
```

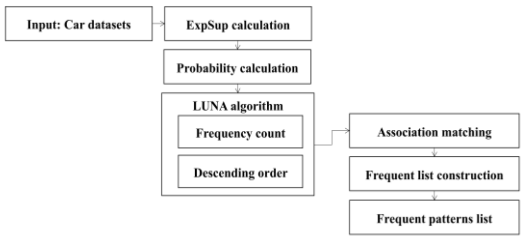

Fig. 2. Story Board of the proposed algorithm.
calculated.

Line 12 merges any neighboring TCs. At the end, Large 1-itemsets with the list of TCs is given. The second procedure, Lower level itemsets are used as seeds for generating higher level candidates. This algorithm uses joint operator to generate higher level itemsets from lower levels. At line 4 of the Algorithm 3.1, we scan the database and calculate the support count of each candidate. Line 7 is used to find temporal patterns by calling explore procedure. Large K-itemsets are generated if support count of the candidates is bigger than the minimum support and P(TC), the power set of TC, be dense line 8).

B. Enhancing Performance of the Algorithm

Here we are discovering frequent itemsets which is time consuming on its own, yet mining patterns with temporal information makes it even harder to find a solution in a reasonable time it is necessary to optimize the performance of the algorithm. We propose a special implementation of a linked list data structure.As discussed earlier, to consider time hierarchies for mining patterns, TCs are proposed. Via linked list data structure, each entity is referenced to its upper and lower level and also to its previous and following entity. Furthermore, there is need to retrieve transactions with their time stamp information easily. To this end, hash-based data structure is proposed. The key is TID and the value is itemset. In other words, hash-based data structure is nested in linked list data structure to exhibit transactions with their time-stamp information. . As we can see, cells which are representation of TCs, contain transactions by hash-based data structure..

IV. COMPUTATIONAL STUDY

In this section, experimental study is carried out by applying the proposed algorithm. The performance of the algorithm is examined in the experiment on synthetic data. In the following, we describe the process of generating synthetic dataset.

A. Generation of Synthetic Data

A synthetic dataset is chosen rather than a real dataset so that a controlled experiment can be conducted to validate the efficacy of our approach. The IBM quest synthetic data generator [5] has been used to generate a dataset for experimentation. As the dataset is non temporal, so it cannot be used directly. We have incorporated the temporal feature in the dataset; so, it can be a temporal dataset and can be handled by our algorithm. We developed a program for this. The program takes starting date, ending date, and synthetic data as inputs and randomly distributes data between the starting and ending dates. Dataset features such as number of items, number of transactions, average transaction length, and time span of the dataset can be set by the user. As there is no guarantee that the generated dataset contains pattern with temporal information, augmentation tothe original dataset is proposed. The augmented pattern act as a target, and the aim of our algorithm is to discover this target pattern. The augmentation method is now described. The a priori algorithm is used to identify frequent patterns on original dataset. One of the patterns is selected and augmented in a new copy of the dataset. After selecting a pattern, augmentation occurs by inserting it as a transaction to the intended time interval with no additional items; so, no unexpected correlations between items are introduced. The result is a dataset that has an increase in the selected pattern in the intended time interval. The objective is to find this pattern by our algorithm.

B. Experiment on Augmented Dataset

The dataset is a synthetic database containing transactions madeupby100itemsfrom01/01/2015-00:00:00to01/01/201600:00:00. Then a pattern is chosenasa target solution. The goal behind using such a large dataset for analysis is to make sure that our proposed algorithm is capable of handling any frequent itemset mining problem. Considering time hierarchies of the time stamp, i.e., month, day, hour, a specific time interval for augmenting a pattern is chosen. From a priori, itemset {61,70,91}is a frequent pattern. We assume that this pattern is frequentbetween7 and9A.M.,1stand2nddaysofthefirstthree months of 2015. This time interval contains 3×2×3 BTCs. For each BTC, 100 transactions are augmented which include only items 61, 70, and 91. Our algorithm should be able to find this itemset and its temporal information. The proposed algorithm is coded by Java programming language and is run on a notebook computer using an Intel Core i7, 2.5 GHz CPU with 16 G of memory and running windows 10. As we expected, target solution, i. e. , pattern{61,70,91}was mined in the specified time interval at a reasonable time. Fig. 4 shows the relationship between the solution time and minimum supports at different densities. As we can see, the lower the minimum support, the higher the solution time is. It is also worth noting that with support level less than 20%, many other useful patterns is found in addition to the target solution. Although there is not a considerable differences between solution time with different density threshold, but now we guarantee that data patterns found are valid and no overestimating of the time periods has been occurred.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Fig. 4. Solution time of the proposed algorithm against different minimum supports (a) α = 30%. (b)α = 50%.

C. Experiment on Real Dataset As the density threshold is proposed in this study and is defined for the proposed algorithm, we could not compare the results of our method with others. But instead, we use a real dataset of a grocery store to compare the results of this section with Section IV-B. The dataset is available from website at http://fimi.ua.ac.be/data/. We use the same procedure which is proposed in Section IV-A to change the non temporal dataset to temporal one. The total number of items is 331 and total number of transactions is 9835. We run the program with minimum support 50% and α equal to 20%. Total of 88 patterns wasfoundbyouralgorithmin1222seconds (20.3minutes).As we can see, runtime of the algorithm heavily depends on minimum support threshold and number of items. Increasing in the number of items and transactions, adds more complexity to the problem and though runtime of the algorithm increases. Some of the patterns with the list of their time interval are shown in Table III. As we can see in Table III, soda, bottled water, and whole milk were sold together during specific time interval. Therefore, many marketing strategies can be applied to increase the profit of the store. For example, according to the cross-selling strategy, items with less interest from the customers perspective can be placed between frequent patterns in specific time intervals to increase the selling rate.

D. Analysis of Scalability

In this section, for analyzing the scalability and fair comparison of the performance of the algorithm, several experiments are carried out and then we compare the results of our algorithm with [8]. Several experiments are designed and in each experiment, the objective is to find a target solution in an augmented dataset. The obtained results from numerous runs of the algorithm with different number of items and transactions at different periods of time is shown in Table IV. In all of the experiments, the minimum support is 25% and density rate is 50%. Also the augmented target is the same for all experiments. We can see how the proposed algorithm perform well in almost all of the experiments. By comparing experiments 1 and 2, we can infer that solution time increases when number of items and average transaction length increase which is reasonable. Comparing experiments 3and4 and also5 and 6, it is seen that solution time increases, when the time period is increased. This seems logical since increasing the time interval, increases the solution space. We also compare the results of our algorithm with the work of [8]. They considered as mall data set consisting of time-stamp and items sold at the corresponding dates which is collected over two months,

i.e., January 2000–February 2000. The total number of items is 5, total number of transactions is 60. The dataset is given in Table V. In order to compare the results, we use the same minimum support, i.e., 50%, and run the program. The runtime of our algorithm is 0.079 seconds and results are shown in Table VI. In terms of the number of detected patterns, the number of patterns detected by [8] is less than ours. In terms of the quality of the detected patterns, the results show that some of the patterns extracted by [8] are not valid. For example, [8] stated that itemset {1,2} is frequent in period [1–2] [4–9]. However, the results of our algorithm indicate that this itemset is frequent in
Time stamps Items Time stamps Items Time stamps Items
the period [4–9] in the first month, and in the period [3–10] in the second month. Even with more accurate survey of dataset, it is clear that the tiny time frame in which itemset {1,2} become frequent is [5–8] and not [4–9]. Our proposed algorithm is able to detect this defect and has extracted more accurate patterns.

V. CONCLUSION

In this paper, we studied the mining of frequent item sets along with their temporal patterns. Some patterns are held during some time intervals while others may happen periodically. The main feature of our proposed algorithm is that a new notion of TCs is presented to consider time hierarchies in data mining process. It enables us to find different kinds of temporal patterns.

For small or medium-sized datasets, it can find the solution in less than one minute. We applied our algorithm to market basket dataset with time stamps. From managerial viewpoint, results of our algorithm can help managers to make better decisions. Duplicate data are eliminated easily. It is easier to implement in large database. Performance is high compared to original apriori .

REFERENCES

[1] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques. Amsterdam, The Netherlands: Elsevier, 2011.

[2] Y. Xiao, Y. Tian, and Q. Zhao, “Optimizing frequent time-window selection for association rules mining in a temporal database using a variable neighbourhood search,” Comput. Oper. Res., vol. 52, pp. 241–250, 2014.

[3] A. K. Mahanta, F. A. Mazarbhuiya, and H. K. Baruah, “Finding calendarbased periodic patterns,” Pattern Recognit. Lett., vol. 29, no. 9, pp. 1274– 1284, 2008.

[4] L.T.Nguyen,B.Vo,T.-P.Hong,andH.C.Thanh,“Car-iner:Anefficient algorithm for mining class-association rules,” Expert Syst. Appl., vol. 40, no. 6, pp. 2305–2311, 2013.

[5] 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. 115–124, 2015.

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

[7] 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. 2350–2359, 2008.

[8] M. Shaheen, M. Shahbaz, and A. Guergachi, “Context based positive and negative spatio-temporal association rule mining,” Knowl.-Based Syst., vol. 37, pp. 261–273, 2013.

[9] Y.-L. Chen and C.-H. Weng, “Mining fuzzy association rules from questionnaire data,” Knowl.-Based Syst., vol. 22, no. 1, pp. 46–56, 2009.

[10] F. Benites and E. Sapozhnikova, “Hierarchical interestingness measures for association rules with generalization on both antecedent and consequent sides,” Pattern Recognit. Lett., vol. 65, pp. 197–203, 2015..

[11] W.-W. Junheng-Huang, “Efficient algorithm for mining temporal association rule,” Int. J. Comput. Sci. Netw. Sec., vol. 7, no. 4, pp. 268–271, 2007. [30] J.-W. Huang, B.-R. Dai, and M.-S. Chen, “Twain: Two-end association miner with precise frequent exhibition periods,” ACM Trans. Knowl. Discovery Data, vol. 1, no. 2, 2007, Art. no. 8.

[12] J. Adhikari and P. Rao, “Identifying calendar-based periodic patterns,” in Emerging Paradigms in Machine Learning. New York, NY, USA: Springer, 2013, pp. 329–357.

[13] W.-J. Lee, J.-Y. Jiang, and S.-J. Lee, “Mining fuzzy periodic association rules,” Data Knowl. Eng., vol. 65, no. 3, pp. 442–462, 2008.

[14] S. G. Matthews, M. A. Gongora, and A. A. Hopgood, “Evolving temporal association rules with genetic algorithms,” in Research and Development in Intelligent Systems XXVII. New York, NY, USA: Springer, 2011, pp. 107–120.

[15] S. G. Matthews, M. A. Gongora, and A. A. Hopgood, “Evolutionary algorithms and fuzzy sets for discovering temporal rules,” Int. J. Appl. Math. Comput. Sci., vol. 23, no. 4, pp. 855–868, 2013.

[16] S. G. Matthews, M. A. Gongora, A. A. Hopgood, and S. Ahmadi, “Web usage mining with evolutionary extraction of temporal fuzzy association rules,” Knowl.-Based Syst., vol. 54, pp. 66–72, 2013.

[17] B. Shen, M. Yao, Z. Wu, and Y. Gao, “Mining dynamic association rules with comments,” Knowl. Inf. Syst., vol. 23, no. 1, pp. 73–98, 2010.

[18] Y. Xiao, R. Zhang, and I. Kaku, “A new framework of mining association rules with time-windows on real-time transaction database,” Int. J. Innov. Comput., Inf. Control, vol. 7, no. 6, pp. 3239–3253, 2011.

[19] B. Saleh and F. Masseglia, “Discovering frequent behaviors: Time is an essential element of the context,” Knowl. Inf. Syst., vol. 28, no. 2, pp. 311 – 331, 2011.