Finding Cyclic Frequent Itemsets

F. A. Mazarbhuiya[1]*, M. Shenify[1], A. Khan[2], A. Farooq[1]

**[1] Dept. of Computer Science, College of Science, Al Baha University
Al Baha, 65431, KSA**

**[2] Department of Computer Engineering, HMS Institute of Technology
Tumkur, India**

Abstract

Mining various types of association rules from supermarket datasets is an important data mining problem. One similar problem involves finding frequent itemsets and then deriving rules from frequent itemsets. The supermarket data is temporal. Considering time attributes in the supermarket dataset some association rules can be extracted which may hold for a small time interval and not throughout the data gathering period. Such rules are called as local association rules and corresponding frequent itemsets as locally frequent itemsets. Mahanta et al proposes an algorithm for extracting all locally frequent itemsets where each locally frequent itemset is associated with sequence time intervals in which it is frequent. The sequence of time intervals associated with a locally frequent itemsets may exhibit some interesting properties e.g. the itemsets may be cyclic in nature. In this paper we propose an alternative method of finding such cyclic frequent itemsets. The efficacy of the method is established through experimental results.

Keywords: *Itemsets, Frequent Itemsets, Local Frequent Itemsets. Cyclic Frequent Itemsets.*

1. Introduction

Association rules mining problem has been defined initially by R. Agrawal et al. [2] for application in large super markets. Finding association rules between items in temporal databases has been described as an important data-mining problem. Transaction data are normally temporal. The supermarket transaction is an example of this type. Mining from temporal dataset is also an important data mining problem. Ale and Rossi [8] propose a method of extracting frequent sets from such datasets. Mahanta et al. [14] propose an algorithm to find locally frequent itemsets from temporal datasets. The algorithm proposed by them finds all frequent itemsets where each frequent itemset is associated with a list of time intervals. The list of time intervals associated with a frequent itemsets can be used to find some interesting results. In this paper we propose to study the problem of cyclic nature of a list of time intervals associated with a frequent itemset and devise a method to extract all frequent itemsets which are cyclic. We call such frequent itemset as cyclic frequent itemsets. In such case, we define the equality between time intervals associated with locally frequent itemsets as follows: two time intervals are said to be almost equal if their lengths are equal up to a small variation. A frequent itemset is said to be cyclic if the lengths of time intervals associated with it is almost equal and the time gaps between any two consecutive time intervals is also almost equal.

The discovery of cyclic and periodic patterns was studied by a couple of researchers (e.g. [18, 13,24]). Ozden et al [18] proposes discovery of cyclic association rules, where the cycle has to be specified by user. Zhang et al [22], propose the method of mining frequently occurring periodic patterns with gap-requirement. In [13], they discussed about a method of extracting temporal association rules with respect fuzzy match i.e. rules holding enough number of intervals given by the corresponding calendar patterns. But here also the user has to specify the calendar schema. So all the above mentioned methods are static in the sense that once the user has specified the period it has to be same throughout the execution of the algorithm. Algorithm propose in this article takes input from the output from algorithm Mahanta et at. [14], which dynamically extract all the locally frequent itemsets along set of list of time intervals, where each locally frequent itemset is associated with a list time intervals. From these lists of time intervals the cyclicity or periodicity is derived in this article. So our method is less user dependent in comparison to other existing methods.

In section II we give a brief discussion on the related work from the literature. The definitions, terms and notations used in this paper are described in section III. In section IV, we give the algorithm proposed in this paper for mining cyclic frequent sets. The experimental results and findings are described in section V. The conclusion and suggestions for future work are given in section VI. References are given section VI.

2. Related Works

The association rules mining problem is first formulated by Agrawal *et al* [2] in 1993. Given a set *I*, of items and a large collection *D* of transactions involving the items, the problem is to find

relationships among the items i.e. the presence of various items in the transactions. A transaction *t* is said to support an item if that item is present in *t*. A transaction *t* is said to support an itemset if *t* supports each of the items present in the itemset. An association rule is an expression of the form $X \Rightarrow Y$ where *X* and *Y* are subsets of the itemset *I*. The rule holds with confidence τ if τ percent of the transaction in *D* that supports *X* also supports *Y*. The rule has support σ if σ percent of the transactions supports $X \cup Y$. Agrawal and Srikant propose a method for extracting association rules, which is known as the *A priori algorithm* (Agrawal & Srikant 1994). This was then followed by subsequent refinements, generalizations, extensions and improvements.

Temporal Data Mining is now an important extension of conventional data mining and has recently been able to attract more people to work in this area. By taking into account the time aspect, more interesting patterns can be extracted that are time dependent. There are mainly two broad directions of temporal data mining [19]. One concern is the discovery of causal relationships among temporally oriented events. The other concern is the discovery of similar patterns within the same time sequence or among different time sequences. The underlying problem is to find frequent sequential patterns in the temporal databases. Manilla *et al.* [16] discuss about the problem of recognizing frequent episodes in an event sequence where an episode is defined as a collection of events that occur during time intervals of a specific size.

The association rule discovery process is also extended to incorporate temporal aspects. In temporal association rules each rule has associated with it a time interval in which the rule holds. The problems are to find valid time periods during which association rules hold, the discovery of possible periodicities that association rules have and the discovery of association rules with temporal features. The problem of temporal data mining is addressed extensively by different researchers and techniques and algorithms have been developed for this [8,9,10,11]. Ale and Rossi [8] describe an algorithm for the discovery of temporal association rules. Ozden *et al.* [18] proposed two algorithms for the discovery of temporal rules that display regular cyclic variations. Similar works are done by others [24, 13] incorporating multiple granularities of time intervals (e.g. first working day of every month) from which both cyclic and user defined calendar patterns can be achieved. Mahanta et al. [14] propose a method of finding locally frequent sets and periodic association rules which is an improvement of other methods in the sense that it dynamically extracts all the rules along with the intervals where the rules hold. The algorithm discussed extracts all frequent itemsets, where each itemset is associated with a list of time intervals where the itemset is frequent. Mazarbhuiya *et al* [17] propose a method of extracting association rules from locally frequent itemsets using rough set and boolean reasoning.

Verma and Vyas [20] propose an algorithm for mining calendar based temporal association rules in sequence data. Their approach is a time-sensitive approach for mining frequent itemsets. Huang and Cheng ---- propose the problem of extracting periodic patterns that regards temporal regularity in sequence data.

Considering time-stamp as a calendar dates, Mahanta et al. [15], propose a method of extracting periodic patterns (viz. yearly periodic, monthly periodic, daily periodic) from temporal data sets. The nicety about their method is that the same algorithm can be used to extract partially and fully periodic patterns. Zeng *et al* [21] suggest a new approach to extract cyclic association rules which addresses the issue of compartmentalization of a cycle into several time segments.

Ahmed *et al* [6], propose a new method called PCAR for extracting cyclic association rules and claim that their method outperforms. Works have been done for the maintenance of these rules [3,4]. Ahmed *et al* [7], propose a method of mining cyclic patterns that occur in some user-defined intervals at regular period of time. Ahmed *et al* [7], introduce new definitions of cyclic association rules and propose a new method to generate such rules from different dimensions.

3. Problem Definition

Let $T = <t_o, t_1, ..>$ be a sequence of time-stamps over which a linear ordering < is defined where $t_i < t_j$ means t_i denotes a time which is earlier than t_j. Let *I* denote a finite set of items and the transaction dataset *D* is a collection of transactions where each transaction has a part which is a subset of the item set *I* and the other part is a time-stamp indicating the time in which the transaction had taken place. We assume that *D* is ordered in the ascending order of the time-stamps. For time intervals we always consider closed intervals of the form $[t_1, t_2]$ where t_1 and t_2 are time-stamps. We say that a transaction is in the time interval $[t_1, t_2]$ if the time-stamp of the transaction say *t* is such that $t_1 \le t \le t_2$.

We define the local support of an item set in a time interval $[t_1, t_2]$ as the ratio of the number of transactions in the time interval $[t_1, t_2]$ containing the item set to the total number of transactions in $[t_1, t_2]$ for the whole dataset *D*. We use the notation $Supp_{[t_1,t_2]}(X)$ to denote

the support of the item set X in the time interval $[t_1, t_2]$. Given a threshold σ we say that an item set X is frequent in the time interval $[t_1, t_2]$ if $Supp_{[t_1,t_2]}(X) \geq (\sigma/100)$* tc where tc denotes the total number of transactions in D that are in the time interval $[t_1, t_2]$. We say that an association rule $X \Rightarrow Y$, where X and Y are item sets holds in the time interval $[t_1, t_2]$ if and only if given threshold τ,

$$Supp_{[t_1,t_2]}(X \cup Y) / Supp_{[t_1,t_2]}(X) \geq \tau / 100.0$$

and $X \cup Y$ is frequent in $[t_1, t_2]$. In this case we say that the confidence of the rule is τ.

3.1 Almost equal intervals

For each locally frequent item set extracted by algorithm (Mahanta *et al* [14]), a list of time intervals is kept in which the set is frequent where each interval is represented as [*start, end*] where *start* gives the starting time-stamp of the time interval and *end* gives the ending time-stamp of the time-interval. *end – start* gives the length of the time interval. Given two intervals [$start_1$, end_1] and [$start_2$, end_2] if the intervals are non-overlapping and $start_2 > end_1$ then $start_2 - end_1$ gives the distance between the time intervals. Similarly two intervals [$start_1$, end_1] and [$start_2$, end_2] are said to be *almost equal* in length if the length of the both intervals are equal up to a small variation say δ% i.e. (end_1- $start_1$)± δ% of (end_1- $start_1$) is equal to (end_2- $start_2$) or (end_2- $start_2$) ± δ% of (end_2- $start_2$) is equal to (end_1- $start_1$) where δ is specified by user.

3.2 Proposed algorithm

To extract cyclic frequent sets we find the time gap between any two consecutive frequent time intervals of the same set. If the time gaps between consecutive intervals are found to be *almost equal* in length and also the lengths of the frequent intervals are found to be *almost equal* (the definition of *almost equal* in length is given in section-3) then we call these frequent sets as cyclic frequent sets. Now to find out such type of cyclicity for each frequent item set we proceed as follows. If the first frequent interval is *almost equal* in length with second frequent interval then we see whether the time gap between the first and the second time interval is *almost equal* in length with the time gap between the second and third periods. If it is, then we take the average of the first two time gaps and see whether it is almost equal to the time gap between the third and the fourth periods. If the average length of the first two intervals of frequency is *almost equal* in length with the third interval of frequency, we proceed further or otherwise stop. In general if the average lengths of the first (*n*-1) frequent intervals is *almost equal* to the length of the *n*-th frequent interval and the average of first (*n*-2) time gaps are almost equal to the (*n*-1)th time gap, then the average of *n* frequent intervals is compared with (*n*+1)th frequent interval and that of the first (*n*-1) time gaps is compared with the *n*-th time gap. This way we can extract cyclic patterns if such patterns exist. However there is a problem, if the sizes of time intervals associated with a frequent itemset are increasing or decreasing progressively then the average length of intervals will also be increased or decreased accordingly. In that case we can add another condition for example the average length of time interval is compared with the first time interval and if they are within a predefined limit, then we can say that the itemset is cyclic. The same thing we can do for the time gaps.

We describe below the algorithm for extracting cyclically frequent item sets.

3.3 Algorithm

```
for each frequent item set iset do
   {L ← list of time intervals for iset
    t1 = L.get();        // t1 is now pointing to the first interval in L
    t1s = t1.stime();
    t1e = t1.etime();
    l1 ← t1e-t1s
    t2 = L.get();
    t2s = t2.stime();
    t2e = t2.etime();
    l2 ← t2e – t2s
    if not almostequal(l1,l2,sgma) then
      {report that iset is not cyclic in nature
       Continue          /* go for the next frequent item set */
      }
    n=1
    avgtg ← t2s-t1e
    avgten ← (l1 + l2)/2
    flag = 0
   while ((tint = L.get()) != null)
{ ts = tint.stime();
 te = tint.etime();
       tg ← ts-t2e
       if almostequal(tg, avgtg, sgma) then
          avgtg ← (n*avgtg + tg)/(n+1)
       else
          {flag = 1; break;}
       len ← te – ts
      if almostequal(len, avglen, sgma) then
         avglen ← ((n+1)*avglen + len) /(n+ 2)
```

```
    else
       {flag = 1; break;}
     t2e ← te
     n←n+1
    }
  if (flag == 1)
      report that iset is not cyclic in nature
  else
 report that iset is cyclic in nature with average time
gap avgtg and average period  length avglen
  }
```

Here the function *stime()* returns the starting time of the corresponding time interval and the function *etime()* returns the ending time. The function *get()* returns a pointer to the next node of the current node in a linked list of intervals.

4. Results and Discussion

Experiments we performed on one synthetic dataset generated through the program provided by the *Quest research group* at IBM Almaden. This generator can be downloaded from http://www.almaden.ibm.com. The detail of the dataset is given below.

Dataset	#Items	#Transactions	*Min* |*T*|	*Max* |*T*|	Avg |*T*|
T10I4D100K	942	100 000	4	77	39

Table 5.1: Dataset Characteristics

Since the dataset is non-temporal with 100,000 transactions. For experiment, we take first 10000 transactions. A program was written to incorporate temporal features in the dataset. The program takes as input a starting date and two values for the minimum and the maximum number of transactions per day. A number between these two limits are selected at random and that many consecutive transactions are labeled with the same date to reflect the fact that many transactions have taken place on that day. This process starts from the first transaction to the end by marking the transactions with consecutive dates (assuming that the market remains open on all week days). The process is repeated for first 20000, 30000, 40000, 50000, 60000 transactions and then for whole dataset to generate the datasets of different sizes. We have given the maximum number of transactions and minimum number of transactions in such a way that the lifetime of each size of dataset is almost one year. In the table given below we describe the dataset used in the experiments.

Data Size (No of Transactions)	Minimum of transactions/day	Maximum of transactions/day	Starting day of transactions	End day of transactions
10000	20	30	1-1-2000	31-12-2000
20000	40	60	1-1-2000	27-12-2000
30000	70	90	1-1-2000	2-01-2001
40000	100	125	1-1-2000	28-12-2000
50000	125	150	1-1-2000	01-01-2001
60000	160	185	1-1-2000	2-1-2001
100000	250	300	1-1-2000	25-12-2000

Table 5.2: Datasets generated

We apply algorithm (Mahanta *et al* [14]) to extract locally frequent itemsets where each locally frequent itemset is associated with a list of time intervals. Then we apply the algorithm discussed in this paper to extract cyclic frequent itemsets along with their periods. We take the value of *sgma*=20 for this purpose.

The table below gives some of the cyclic itemsets extracted by our method along with their time periods, average time gaps, *avgtg* and average length of the cycle's, *avglen*.

Itemsets	Time periods	*avgtg* in days	*avglen* in days
30	[1-1-2000, 10-1-2000], [2-3-2000, 10-3-2000], [1-05-2000, 11-5-2000]	52	9
33	[2-1-2000, 12-1-2000], [1-3-2000, 13-3-2000], [3-5-2000, 13-5-2000]	50	10.25
30, 33	[3-1-2000, 10-1-2000], [4-3-2000, 9-3-2000], [4-5-2000, 10-5-2000]	55	6
50	[5-4-2000,20-4-2000], [6-7-2000,19-7-2000], [6-10-2000,21-10-2000]	78	14.33
70	[3-4-2000,21-4-2000], [2-7-2000,19-7-2000], [4-10-2000,20-10-2000]	72.5	17
50, 70	[6-4-2000,19-4-2000], [8-7-2000,19-7-2000], [6-10-2000,20-10-2000]	79	12.67

Table 5.3: Cyclic itemsets along with cycles

From the above table, we can make the following observations. The itemset {30} is frequent in the 3 intervals [1-1-2000, 10-1-2000], [2-3-2000, 10-3-2000], and [1-05-2000, 11-5-2000]. The lengths of the intervals are respectively 9 days, 8 days and 10 days. If we take *sgma*=20 then the intervals are *almost equal* with average length 9 days. Again the time gap of 1st and 2nd interval is 52 days and that of 2nd and 3rd interval is also 52 days i.e. *equal*. Therefore the average time gap is 52 days. Thus the itemset {30} is cyclic. Again the itemset {33} is frequent in the 3 intervals [2-1-2000, 12-1-2000], [1-3-2000, 13-3-2000], and [3-5-2000, 13-5-2000]. With the value of *sgma* {33} is cyclic in the above 3 intervals with average time gap of 50 days and average length of intervals 10.25. Similarly the intervals {30, 33}, {50}, {70}, {50, 70} are cyclic frequent in their respective time intervals given in the table.

In the table given below, number of cyclic itemsets obtained by our method for the different sizes of datasets are given.

Data Size (No of Transactions)	Number of Cyclic itemsets obtained
10000	2
20000	5
30000	10
40000	16
50000	24
60000	34
100000	91

Table 5.4: Results obtained

Considering transactions along x-axis and cyclic itemsets along y-axis, we can represent the result of table 5.4 graphically as given in fig. 5.1 or using bar diagram as given in fig. 5.2. In the figures we observe that if the number of transactions is increased keeping other parameters constant like lifetime of the datasets, minimum support, minimum size of time intervals etc. , the number of cyclic itemsets will also be increased. Similarly it has been found that if the time period is increased keeping other parameters constant, the number of cyclic itemsets is also increased.

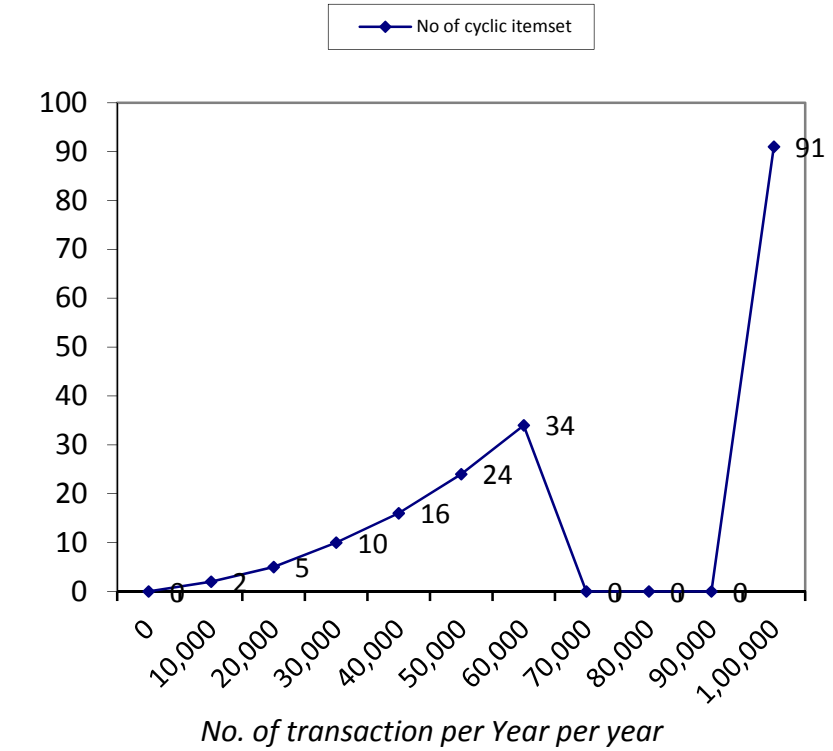

Fig.5.1: No. of transactions vs. No. of cyclic itemsets

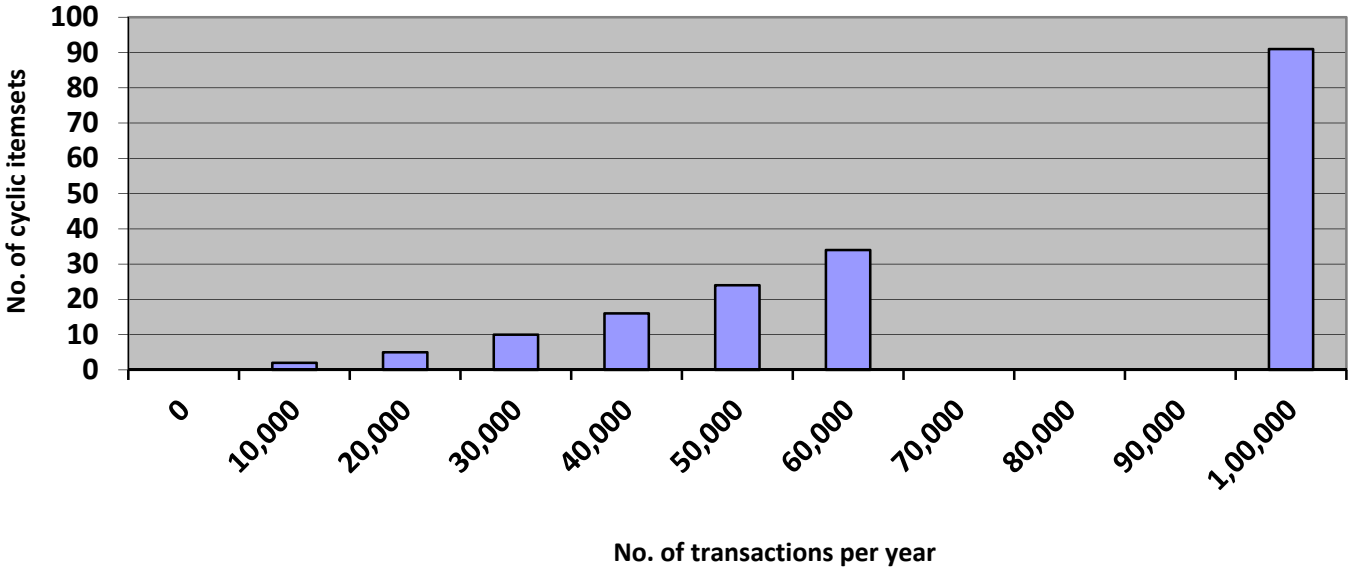

Fig.5.2: No. of transactions vs. No. of cyclic itemsets

5. Conclusions and Suggestions for Future Work

An algorithm for finding cyclic frequent sets from temporal data is given in the paper. The algorithm discuss by Mahanta et al. [14] gives all locally frequent itemsets where each frequent itemsets is associated with a list of time intervals where it is frequent. Our algorithm takes the input from the result of the algorithm proposed by Mahanta et al. [14] and supplies all frequent sets which are cyclic in nature.

We may have some frequent itemsets where the time gaps are *almost equal* in length but the duration of the intervals of frequency are not equal even in the approximate sense. We may also have some frequent itemsets where the time-gaps are not equal in length but durations of the intervals are *almost equal*. The above algorithm can be modified accordingly to find such frequent itemsets. In future, we may also modify our algorithm to get more accurate results. We would also like to find partially periodic patterns and other types of patterns which may exist in the datasets.

References

[1]. Agrawal R. and Srikant R. (1994); Fast algorithms for mining association rules, Proceedings of the 20th International Conference on Very Large Databases (VLDB '94), Santiago, Chile, June 1994.

[2]. Agrawal R., Imielinski T. and Swami A. (1993); Mining association rules between sets of items in large databases; Proceedings of the ACM SIGMOD '93, Washington, USA, May 1993.

[3]. Ahmed E. B., Gouider M. S, (2010); Towards a new mechanism of extracting cyclic association rules based on partition aspect. RCIS 2010, pp. 69-78.

[4]. Ahmed, E. B. (2010); Incremental Update of Cyclic Association Rules. IDEAL 2010, pp. 387-395.

[5]. Ahmed, E. B., Nabli, A., and Gargouri, F. (2011); Cyclic Association Rules: Coupling Multiple Levels and Parallel Dimension Hierarchies, (IKE'2011), The 2011 Int'l Conf on Information and Knowledge Engineering, Las Vegas, USA.

[6]. Ahmed, E. B., Gouider M. S, (2010);Towards an incremental maintenance of cyclic association rules CoRR abs/1009.5149.

[7]. Ahmed, E. B., Nabli, A, and Gargouri, F.,(2011) Mining cyclic association rules from multidimensional knowledge. ICDIM 2011, pp. 12-17

[8]. Ale J. M. and Rossi G.H. (2000); An approach to discovering temporal association rules; Proceedings of the 2000 ACM symposium on Applied Computing, March 2000.

[9]. Chen X. and Petrounias I. (1998); A framework for Temporal Data Mining; Proceedings of the 9th International Conference on Databases and Expert Systems Applications, DEXA '98, Vienna, Austria. Springer-Verlag, Berlin; Lecture Notes in Computer Science 1460, 796-805, 1998.

[10]. Chen X. and Petrounias I. (1998a); Language support for Temporal Data Mining; Proceedings of 2nd European Symposium on Principles of Data Mining and Knowledge Discovery, PKDD '98, Springer Verlag, Berlin, 282-290, 1998.

[11]. Chen X., Petrounias I. and Healthfield H. (1998b); Discovering temporal Association rules in temporal databases; Proceedings of IADT'98 (International Workshop on Issues and Applications of Database Technology, 312-319, 1998.

[12]. Huang K. Y, Chang C. H. (2004); Mining Periodic Patterns in Sequence Data, Data Warehousing and Knowledge Discovery, LNCS, 2004, Volume 3181/2004, pp. 401-410

[13]. Li Y., P. Ning, Wang X. S. and Jajodia S. (2001); Discovering Calendar-based Temporal Association Rules, In Proc. of the 8th Int'l Symposium on Temporal Representation and Reasonong, 2001.

[14]. Mahanta A. K., Mazarbhuiya F. A. and Baruah H. K. (2005); Finding Locally and Periodically Frequent Sets and Periodic Association Rules, Proceeding of 1st Int'l Conf on Pattern Recognition and Machine Intelligence (PreMI'05),LNCS 3776, 576-582.

[15]. Mahanta A. K., Mazarbhuiya F. A. and Baruah H. K. (2008). Finding Calendar-based Periodic Patterns, Pattern Recognition Letters, Vol. 29(9), Elsevier publication, USA, pp. 1274-1284.

[16]. Manilla H., Toivonen H. and Verkamo I. (1995); Discovering frequent episodes in sequences; KDD'95; AAAI, 210-215, August 1995.

[17]. Mazarbhuiya F. A., Mahanta A. K., Abulaish M and Tanvir Ahmad (2009), Mining Local Association Rules from Temporal Data Set, Proceeding of International Conference of Patterns Recognition and Machine Intelligence (PReMI'09), LNCS 5909, pp. 255-260, Springer Berlin / Heidelberg.

[18]. Ozden B., Ramaswamy S. and Silberschatz A. (1998); Cyclic Association Rules, Proc. of the 14th Int'l Conference on Data Engineering, USA, 412-421.

[19]. Roddick J. F., Spillopoulou M. (1999); A Biblography of Temporal, Spatial and Spatio-Temporal Data Mining Research; ACM SIGKDD.

[20]. Verma K., Vyas O. P. (2005); Efficient Calendar based Temporal Association Rule, SIMOD Record, Vol. 34, No. 3, September 2005, pp. 63-70.

[21]. Zeng B., Luo C., Jiang X., L., and Sun J, (2010); The arithmetic of discover Cyclic Association Rules based on

difference sequence arithmetic clustering, 7th Int'l Conf. on Fuzzy Systems and Knowledge Discovery, Vol. 5, pp. 2059-2063.

[22]. Zhang, M., Kao, B., Cheng, W., David, Y., and Kevin, Y., (2007); Mining periodic patterns with gap requirement from sequences, In: ACM Trans. On Knowledge Discovery from Data (TKDD), Vol. 1(2).

[23]. Zimbrado, G., Moreira de Souza, J., Teixeira de Almeida, V., and Wanderson Arauja de Silva; An Algorithm to Discover Calendar-based Temporal Association Rules with Item's Lifespan Restriction, In Proc. of the 8th ACM SIGKDD 2002.

[24]. Zimbrao G., Moreira de Souza J., Teixeira de Almeida V. and Araujo da Silva W. (2002); An Algorithm to Discover Calendar-based Temporal Association Rules with Item's Lifespan Restriction, Proc. of the 8th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (2002) Canada, 2nd Workshop on Temporal Data Mining, v. 8, 701-706.

Fokrul Alom Mazarbhuiya received B.Sc. degree in Mathematics from Assam University (1994), India and M.Sc. degree in Mathematics from Aligarh Muslim University (1997), India. After this he obtained his Ph.D. degree in Computer Science from Gauhati University (2007), India. He had been serving as an Assistant Professor in College of Computer Science, King Khalid University, Abha, kingdom of Saudi Arabia since 2008 to 2011 then he worked as an Assistant Professor in Dept. of Computer Science and Information Systems Al Baha University, Kingdom of Saudi Arabia. His research interest includes Data Mining, Information security, Fuzzy Mathematics and Fuzzy logic.

Mohamed Shenify received B.Sc. degree in Computer Science from Indiana State University, USA, M.Sc. degree in Computer Science from Ball State University, USA, and Ph.D. in Computer Science from Illinois Institute of Technology, USA. Currently working as an Assistant Professor in the Dept. of Computer Science and Information Systems. Al Baha University, Saudi Arabia. His research interest includes Natural Language Processing, Information Extraction, Text mining, semantic and annotation.