IJCTA, 10(07), 2017, pp. 195-203

A Hybrid Pre-processing Approach for Temporal Associative Rule Classification

Mini T. V* Nedunchezhian R and Vijayakumar V*****

Abstract : Mining temporal association rules is an important data mining problem. Association rules represent directed relationships amongst variables wherein the existence of some variables predict the existence of the other variables in a transaction. Association rule classification is a domain which combines association rule mining and classification. Lot of issues are raised during the temporal association rule based classification such as selection of temp_min_sup and temp_min_conf parameter, performance with large databases and accuracy of the classification. This paper deals with the issues in the pre-processing of the temporal association rule mining and classification. It proposed the hybrid pre-processong approach which combines partitioning the large data sets with K-Means clustering algorithm and an automatic threshold selection algorithm of minimum temporal support and minimum temporal confidence thresholds using polynomial function to improve the overall performance of the temporal association rule classification. The experimental results show a positive outcome on association rule generation and classification exhibits high efficiency when proposed algorithms were used in temporal databases.
Keywords : Hybrid Pre-processing, Temporal Data Partitioning, Temporal Association Rule Mining, Temporal Classification.

1. INTRODUCTION

Advances in information technology have accelerated the collection, storage and processing of various sources of data in recent decades. There are different categories of temporal databases [1] are temporal context, temporal sequences, time-stamped and fully temporal databases. These explosive growths in temporal data as well as the emergence of new technologies emphasize the need for automated discovery of temporal knowledge. Data mining is an integral part of Knowledge Discovery in Database (KDD) process, which is the process of converting raw data into useful information [2]. The primary step in the Knowledge Discovery in Database processes is data pre-processing which consists of consists of attribute selection, data filtering, data selection and splitting, errors detection and so on [3].

Association rule mining is used to identify the relationship between items, was first initiated by Agrawal [4]. It intends to extract appealing correlations, frequent patterns, associations or casual structures among a set of items in the transaction databases or other data repositories. All these techniques have been subjected to modification and advancement in order to achieve maximum efficiency in terms of accuracy and speed. The Apriori algorithm [5] considerably reduces the size of candidate sets using the Apriori principle. There have been comprehensive studies on the improvements or extensions of Apriori, *e.g.*, hashing technique [6], sampling approach [7], incremental mining [8]. These algorithms are effective in reducing the number of database scans. Although, Apriori principle can experience two non-trivial costs: (1) generating a huge number of candidate sets and (2) repeatedly checking the candidates by pattern

* Department of Computer Science, Sacred Heart College Chalakudy, Trissur, India, *E-mail : sistermiranto@gmail.com*
** Coimbatore Institute of Technology, Coimbatore, India, *E-mail : rajuchezhian@gmail.com*
*** Department of Computer Science, S.N.R. Sons College (Autonomous), Coimbatore, India, *E-mail : veluvijay20@gmail.com*

matching and scanning the database. Han [9] formulated an FP-growth method that mines the complete set of frequent itemsets and scans candidate generation. The first scan of the database is compressed into a frequent-pattern tree. The FP-tree is mined by starting from each frequent length-1 pattern, creating its conditional pattern base, then building its conditional FP-tree and performing mining recursively on such a tree. By the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree, the pattern growth is achieved.

Temporal Association rule mining is a variant of the association rule mining which finds relationship between items with respect to particular time periods. The attempt to model temporal relationships in the data is one of the primary factors which differentiate temporal rules from traditional association rules. Association rule classification is a domain which combines association rule mining and classification. The general issues temporal association rule mining and classification are: selection of the two user defined parameters such as min_sup and min_conf, number of association rules generated and classification accuracy.

The main goal of the proposed algorithm is to improve the mining knowledge and the classification accuracy based on temporal association rule mining from huge sized temporal databases using hybrid pre-processing techniques. The remainder of the paper is organized as follows. Section 2 discusses the review of literature. The proposed method is presented in Section 3. Section 4 presents the experimental results and section 5 concludes the paper.

2. REVIEW OF LITERATURE

Classification is the process of learning a function or a model from a data set (training data) in order to make the function be useful in predicting the category of a new instance. Liu [10] proposed Classification Based Association. The Temporal associative rule mining algorithms generate an exceptionally large number of association rules. Various strategies have been proposed to reduce the number of association rules. They include generating only “non-redundant” rules, “interesting” rules, or those rules satisfying certain other criteria such as coverage, leverage, lift or strength or pruning out ‘irrelevant’ rules. Usage of reduction techniques to improve the process of associative rule mining algorithm. It can be grouped into two approaches such as replacing the whole database with only part of it based on the current frequent itemsets or reducing the number of passes over the whole database; to make the number of candidate itemsets much smaller, explore different kinds of pruning techniques. Both these techniques are motivated by the serious issue, if the support and confidence thresholds are small, the set of association rules can develop to be unwieldy as the number of transactions increases. The number of rules presented to the user typically increases proportionately, as the number of frequent itemsets increases. The main objective of the proposed algorithm includes reducing the huge set of frequent patterns generated while maintaining the high quality of patterns. As extensions to the traditional [4] method several types of temporal association rule generation algorithms have been proposed. Examples include sequence rules [5], episode rules [11], cyclic association rules [12], inter-transaction rules [13] and calendric rules [14].

Generally, the temporal association rule mining can be performed in three steps, namely, Data Pre-processing, Find temporal frequent itemsets, and Identify temporal association rules. The quality of the input dataset can be imporved by data pre-processing includes partitioning the large size data base into smaller sizes, removal of unwanted or irrelevant data, data exchange and data reduction. The temporal associative rule generation uses time constraints on the two parameters support and confidence to generate frequent itemsets. These temporal frequent itemsets, generates temporal association rules. A T-Apriori algorithm was proposed by Liang [15] which modified the traditional Apriori algorithm for extracting temporal association rules from ecological database. Ning [16] demonstrated a temporal association rule mining algorithm, considering valid time constraint. Since there are no time-constraints, it is unavoidable to scan the database repeatedly. The candidates in frequent itemset generated by the algorithm promptly decreases. Winarko and Roddick [17] introduced the ARMADA based on Memory indexing for sequential pattern mining in the discovery of temporal rules from the frequent patterns. Tseng et al [18] implemented

Temporal N-Gram a prediction model for user navigation to predict temporal navigation patterns. They also presented an Incremental Temporal Association Rule Mining that is able to reduce time requirement for generating new candidates by storing the candidate rules as 2-itemsets. Nazerfard [19] proposed Temporal Relation Discovery of Daily Activities association rule mining algorithm to discover temporal relations of daily activities.

Xi et al [20] proposed temporal classification method that combines one nearest neighbour and dynamic time warping techniques. Revesz and Triplet [21] presented a temporal data classification using linear classifiers, which used the history features along with the current and the class details during classification. Sharma & Saxena [22] proposed a temporal weighted association rule mining for classification considered time-changing characteristics of items and transactions and used cumulative occurrence count of previous partitions to selectively carry over towards the generation of candidate itemsets. Bhuvaneswari and Umajothy [23] proposed a temporal mining algorithm based on association rules using Apriori algorithm for classification of microarray gene data. It was used to mine gene expression data in order to analyse the effects of expression of one gene with another gene for the biological processes like gene functionality and molecular function.

Generally, The TARC consists of three major steps such as: Frequent itemset generation, Association rule generation and Classification. The Temporal frequent itemset is a set of items appearing together in a number of database records meeting user-specified thresholds, minimum support and minimum confidence with temporal constraints. Using these frequent itemsets, TARC discovers a complete set of Class Association Rules from the training dataset, which is used to build the classifier. The identified issues during the temporal association rule based classification are selection of min_sup and min_conf parameter, performance with large databases and Classification accuracy. The above issues are considered and enhanced the conventional Apriori because of the proved performance record and popularity among non-temporal ARC domain algorithm.

This proposed algorithm focuses on to improve the process of associative rule discovery and classification steps of TARC, by providing solutions to the pre-processing issues. The solutions provided in the pre-processing are: Automatic parameter selection estimation algorithm and partitioning algorithm to maintain performance of TARC when applied to large sized datasets.

3. TEMPORAL ASSOCIATION RULE CLASSIFICATION USING ENHANCED TEMPORAL APRIORI ALGORITHM

The proposed Enhanced TARC consists of three main tasks, namely, Hybrid-pre-processing, frequent itemsets generation, association rules mining and temporal classification. The detailed flow diagram presented in Fig 1.

A. Hybrid- Pre-Processing

The hybrid pre-processing performs two tasks, namely, partitioning the dataset and automatic estimation of the two parameters, min_sup and min_conf. This solves the issues of handling large sized datasets and parameter selection respectively.

1. **Temporal Partitioning of the Dataset :** The proposed TARC begins by partitioning the database into 'p' partitions. The partitioning method segments the dataset according to its timestamp. Consider a temporal dataset D having a set of transactions $T = \{T_1, T_2, T_3, \ldots, T_n\}$ having a set of items $I = \{i_1, i_2, \ldots, i_m\}$. The partitioning algorithm segments the database into P partition, $P = \{p_1, p_2, \ldots, p_k\}$, where each partition has a set of transactions that exhibit similar timestamp properties. The K-means clustering algorithm generates a specific number of disjoint, flat (non-hierarchical) clusters. It is well suited to generate globular clusters. The convergence criteria for the K-means algorithm is minimizing the Sum of Squared Error (SSE). The K-Means algorithm always converges to a local minimum. The particular local minimum found depends on the starting

cluster centroids. The K-Means algorithm updates cluster centroids till local minimum is found. The result of the algorithms is a set of *k* clusters having similar items. Each cluster is treated as a separate partition and constitutes P. Each item in the partition P_k contains different utility values. The frequent pattern mining and association rule generation algorithm is applied separately and the results are cumulated at the end. All the algorithms, like automatic estimation of parameters, frequent pattern mining, associative rule generation are performed for each partition. The resultant association rules generated from each partition are then simply combined to form the final set.

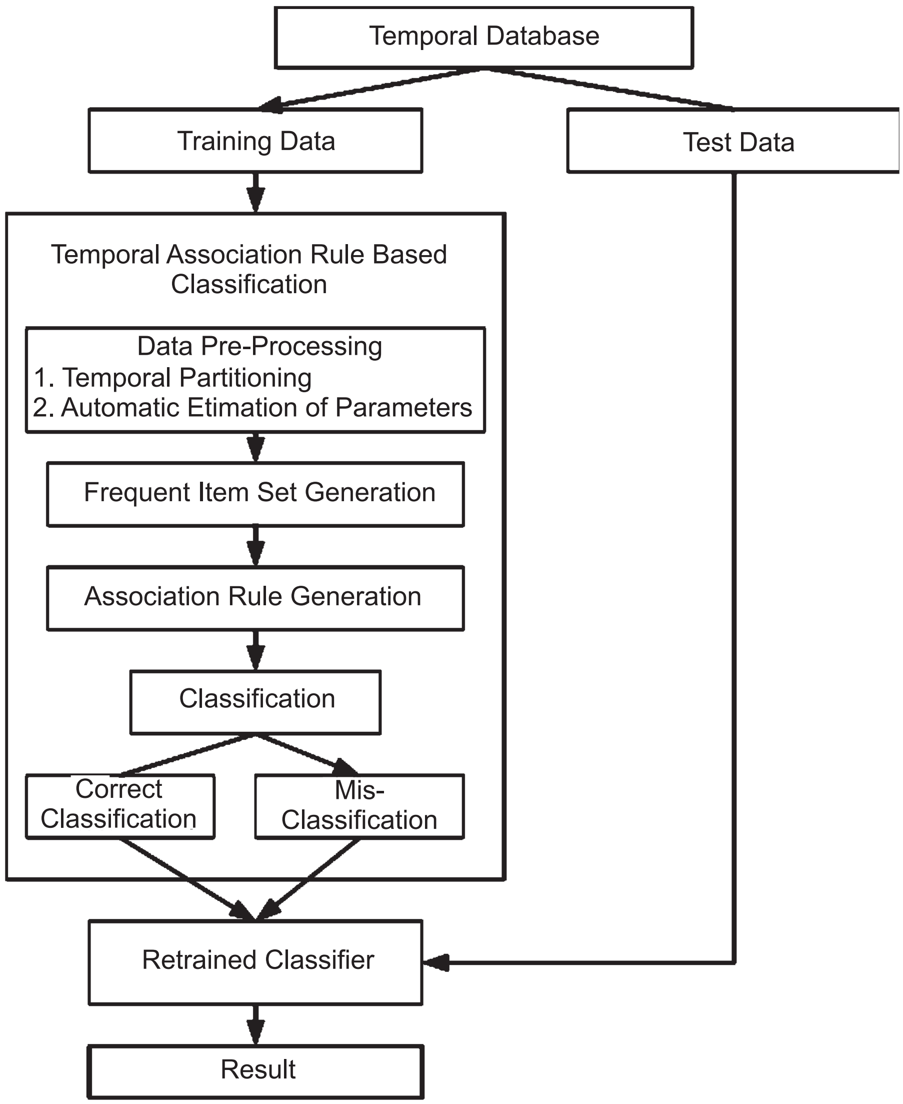

Figure 1: Flow of Proposed TARC

2. **Automatic Estimation of Parameters :** The algorithm begins by using an automatized process to estimate temporal minimum support, using the Hierarchical Temporal Partitioning with Frequent Pattern List (FPL) which finds the temporal frequent itemsets and generates the temporal association rules. Let I = $\{i_1, i_2, \ldots, i_n\}$ be a set of items. Let 'D' be a set of transactions, where each transaction 'T' is a set of items such that T $\subseteq$ I. The main aim of association rule mining is to discover rules that have *t*-support and *t*-confidence greater than a user-specified temp_min_ sup and temp_min_conf. The rule A $\rightarrow$ B holds in the transaction set T with *t*-confidence '*c*', if *c* % of transactions in T that *t*-support 'A' also *t*-support 'B' with temporal constraint. The rule has *t*-support '*s*'in T if s% of the transactions in T contains A $\cup$ B. The performance of frequent itemsets mining and association rule generation depends on two parameters, *t*_min_conf and *t*_min_sup, which are user-specified. Correct selection of these two parameters is critical and is

highly dependent on the nature of database. In general, a high min_sup will result with very few association rules, while a low min_sup will generate very high association rules and both these situations degrade the performance of associative mining. This difficulty brought forward a series of automated methods for the calculation of min_sup (and min_conf). More often, the optimal values for these parameters are selected after repeated runs of the algorithms and choose the set which produces best results. This process is time consuming and hence it is not efficient.

Pyun and Yun [24] have suggested the use of mine-top-*k* frequent patterns for solving this issue. Again, the correct specification of *k* is very important and is dependent on the user. Thus, in order to design an efficient frequent pattern mining algorithm, it is necessary to have optimal min_sup and min_conf values. This proposed method provides an automated method to solve the problem of user-supplied minimum support and minimum confidence thresholds using polynomial function. The automatic estimation of temporal minimum support and minimum confidence thresholds are estimated for each partition in P. Let P_i be a partition of database D with *n* itemsets (i_1, i_2, …, *n*) at a particular timestamp, '*t*'. Let S be the support of each item which is distributed in an interval [*a*, *b*], where *a* and *b* are the minimum and maximum support. Let *m* be the maximum number of items in P. The proposed method, termed as Automated Minimum Support Estimation Method (AMSEM), first scans P to estimate S of each item, from which the average support is calculated. This average value is used as initial min_sup which is then used during the optimal minimum support value estimation.

$$A_s = \frac{\sum_{i=1}^{n} S}{m}$$

The aim of AMSEM is to determine min_sup of *p* within the interval [M_i, M_a], which can be implemented as a mapping $f : [M_i, M_a] \rightarrow [0, 1]$, where M_i is the minimum and M_a is the maximum support in *p*. As, in many cases, this mapping is hidden, it is necessary to find an approximate polynomial approximation function $f^\#$ for *f* as given below. In this Equation, let X in [M_i, M_a] be $x_1, x_2, \ldots, x_n$ and Y in [0, 1] be $y_1, y_2, \ldots, y_n$.

$$\begin{array}{c|c} X & x_1, x_2, \ldots.., x_3 \\ \hline Y & y_1, y_2, \ldots.., y_3 \end{array}$$

A method for finding an approximate polynomial function $f^\#$ for *f* between X and Y can be performed by Zhang et al [25]. From the support value calculated, the minimum confidence measure is estimated using the lift measure. Using the lift and conviction methods, two thresholds are derived, namely, Cumulative support (Cs) and Collective Confidence Measure.

To calculate the cumulative support, the itemsets are first grouped according to their length and then the collective support is calculated from the combination of its individual support from previous level (one-itemset, two-itemset, three-itemset, etc.). The confidence lift is calculated using two measures, namely, the support of the items in the itemset and the cumulative support from the itemsets previous level. The frequent itemsets are then identified using the cumulative support and confidence lift support from which the association rules are generated.

B. Temporal Association Rule Mining with Enhanced T-Apriori Algorithm:

Temporal association rule can be described as "X → Y (*t*-support(*ts*), *t*-confidence(*te*))". The Enhanced T-Apriori algorithm refers temporal time as a constraint. First, nalyse the temporal database with respect to time threshold. Time threshold is the time point or time range. Time range can be expressed as [min_*ts*, min_*te*], while time point [min_*t*]. Then, the time information is detected in the temporal dataset or database in order to decrease computational complexity and apply the Enhanced T-Apriori algorithm to generate frequent itemsets and corresponding temporal association rules.

1. **Generation of Frequent Itemsets :** In Enhanced T-Apriori algorithm, the process of frequent itemsets generation is similar to Apriori algorithm, but we need special treatment of time information. The Inputs are T temporal database and temp_min_s and generates temporal frequent itemsets which have the t-support no less that temp_min_*s*:

   Step 1. For all RecordSets
   //RecordSets ε T, TrecordSets with time information satisfying time threshold
   ItemSets = TrecordSets without TrecordSets.time

   Step 2. C_1={Candidate !-Itemsets}, L_1={c ε C_1|c.count > = mins_s} // delete time information

   Step 3. For(k = 2; L_{k-1}! = ϕ; k ++} //until no more frequent itemsets generate
   C_k= Apriori_Gen(L_{k-1}) //to generate k-item candidate frequent itemsets
   for all transaction tεItemsets do begin
   C_t= subset(C_k, t)
   Forall Candidates c ε C_i do
   c.count++ // the support for each candidate frequest n itemset
   L_k= {c ε C_1|c.count > = mins_s}

   The Apriori_Gen function generates frequent itemset C*k*. It can be separated into two steps: Join and Prune; after the generation of C*k*, subset function scan the database and calculate the support of each subset of C*k*.

2. **Generation of Temporal Association Rule :**
   for L_k, k > = 2
   H_1 = {consequents of rules derived form L_k with one itemin the consequenst};
   Call ap_genrules (L_k, H_1); // generate rules
   Call Add(L_k); // scan the TrecordSets and returns the temporal association rule.

C. Temporal Associative Classification

The Enhanced Temporal Apriori Algorithm used to generate classification association rules. The generated rules are used for building models. Temporal Confidence is calculated for each rule and those rules with temporal confidence greater than or equal to the minimum temporal confidence will form the final set of classification association rules. The resulting models are tested for accuracy. The modal predicts the instance class whose class is unknown. Rules of high temporal confidence are thought to be good for classification. For instance, a rule from an instance that appears only once (high confidence but low support) and may not be a good rule for classification which are useful identifying rare events.

4. EXPERIMENTAL RESULTS

In this experiments, four data set were used for analysis of temporal associative classification such as Ozone data set, El Nino dataset, Forest Fires dataset and Stock Market dataset. Ten-fold cross validation is used. The data set is divided into ten equal size. Nine data sets are used for training and one data set used for testing. All the algorithms are implemented using Java. A Hybrid prepossessing approach was proposed to improve the Temporal association rule classification with T-Apriori Algorithm (TTAA). Two algorithms are proposed in Hybrid-pre-processing such as Partitioning temporal database using K-Means Algorithm (PKM), to partition the algorithm according to its temporal characteristics and Automated Minimum Support and Minimum Confidence Estimation Method (AMSEM), to automate estimation of min_sup and min_conf threshold values. The AMSEM algorithm estimated the minimum support and minimum confidence threshold values as shown in Table from the selected datasets. The effect of the pre-processing tasks in TARC uses precision, recall, F-Measure, accuracy and speed for all the four selected datasets are tabulated in Table 2, Table 3, Table 4, Table 5, and Table 6 respectively.

Table 1
The Min-sup and Min_conf values by AMSEM

Dataset	*Min_sup*	*Min_conf*
Ozone	30	75
El Nino	20	65
Forest Fires	30	75
Stock Market	25	65

Table 2
Effect of Pre-processing on TARC – Precision (%)

Dataset	*TTAA*	*TTAA + PKM*	*TTAA + AMSEM*	*TTAA + PKM + AMSEM*
Ozone	75.66	75.82	75.69	76.60
El Nino	73.92	74.05	73.90	74.70
Forest Fires	81.46	81.70	81.43	81.67
Stock Market	76.33	76.68	76.35	78.07

Table 3
Effect of Preprocessing on TARC – RECALL (%)

Dataset	*TTAA*	*TTAA + PKM*	*TTAA + AMSEM*	*TTAA + PKM + AMSEM*
Ozone	74.21	74.35	74.24	75.77
El Nino	72.54	72.90	72.52	73.14
Forest Fires	80.89	81.84	80.86	81.99
Stock Market	75.17	75.84	75.19	76.15

Table 4
Effect of Pre-processing on TARC – F-MEASURE (%)

Dataset	*TTAA*	*TTAA + PKM*	*TTAA + AMSEM*	*TTAA + PKM +AMSEM*
Ozone	74.93	75.08	74.96	76.18
El Nino	73.22	73.47	73.20	73.91
Forest Fires	81.17	81.77	81.15	81.83
Stock Market	75.75	76.26	75.77	77.10

Table 5
Effect of Pre-processing on TARC – Accuracy (%)

Dataset	*TTAA*	*TTAA + PKM*	*TTAA + AMSEM*	*TTAA + PKM + AMSEM*
Ozone	83.25	83.41	83.28	84.63
El Nino	81.05	81.29	81.03	82.44
Forest Fires	84.89	85.49	84.87	86.10
Stock Market	82.54	82.88	82.57	84.04

Tabele 6
Effect of Pre-processing on TARC – Speed (Seconds)

Dataset	*TTAA*	*TTAA + PKM*	*TTAA + AMSEM*	*TTAA + PKM + AMSEM*
Ozone	11.13	10.99	9.90	10.97
El Nino	13.26	13.15	10.07	13.13
Forest Fires	9.05	8.88	6.81	8.86
Stock Market	12.41	12.27	8.20	12.24

From the results, it is clear that the incorporation of partitioning and AMSEM as a single optimization algorithm in TTAA do not improve the performance much. But when hybrid approach (TTAA + PKM + AMSEM), it increased the performance with the selected performance measures. The inclusion of PKM and AMSEM produced an accuracy efficiency gain of 1.63%, 1.68%, 1.40% and 1.79% with Ozone, El Nino, Forest Fires and Stock Market datasets respectively. The algorithm extracts the temporal association rules for the temporal interval of 4 for the support measures.

This trend changed with speed metric, where the proposed TTAA + PKM + AMSEM took slightly more time than TTAA+ AMSEM. This might be due to the additional computations required by AMSEM algorithm for each partition. But results analysis revealed that it is less than 0.19 seconds at maximum and hence, the TTAA + PKM + AMSEM algorithm is considered as successful in achieving maximum accuracy.

5. CONCLUSION AND FUTURE WORK

Pre-processing is the first phase of the Temporal Association Rule Classification, in which two issues are dealt, in this paper. The first issue was the scalability problem which was solved by using a temporal partitioning K-Means clustering algorithm. Each item in the partition P_k contains different utility values. The second issue is concerned with the automatic estimation of two user-defined parameters namely, minimum support and minimum confidence which is the essential parameters in frequent itemsets mining and association rule generation. It resulted in the finding of polynomial function based solution that automatically helped to estimate temporal minimum support and this modified lift measure is used during temporal minimum confidence. The findings of the experimental results is the pre-processing step has a positive impact on association rule generation process and saves time when compared to user-defined selection of thresholds and also exhibits high efficiency when proposed algorithms were used in temporal databases. The algorithm can be modified in future includes noisy data removal and missing values can be included to strengthen the pre-processing steps.

6. REFERENCES

1. J. F. Roddick and M. Spiliopoulou, "Survey of temporal knowledge discovery paradigms and methods," *IEEE Transactions on Knowledge &Data Engineering*, vol. 14, pp. 750–767, 2002.
2. U. Fayyad, G.P. Shapiro and P. Smyth, "From Data mining to Knowledge Discovery in Databases," *AI Magazine*, vol. 17, pp.37-54. 1996
3. J. S. Malik, P. Goyal and K. A. Sharma, "A comprehensive approach towards data preprocessing techniques and association rules," in *Proc. INDIACom-Computing for Nation Development*, 2010, pp. 1-9.
4. R. Agrawal, T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACMSIGMOD,* ACM , Washington, 1993, pp. 207-216.
5. R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc.ICDE95*, *IEE Computer Society* , Taipei, Taiwan,1995, pp. 3–14.
6. J.S. Park, M.S. Chen and P.S. Yu, "An effective hash-based algorithm for mining association rules," in *Proc. ACM-SIGMOD95*, San Jose, CA, 1995,pp 175-186.

7. H. Toivonen, "Sampling Large Databases for Association Rules," in *Proc. Of the 22th International Conference on VLDB 96*, San Francisco, CA, USA, 1996, pp.134-145.

8. D. Cheung, J. Han, V. Ng, A. Fu and Y. Fu, "A fast distributed algorithm for mining association rules," in *Proc. ICPDIS'96*, Miami Beach, Florida, 1996, pp. 31 – 44.

9. J. Han, J. Pei and Y. Yin, "Mining frequent patterns without candidate generation," in *Proc. ACM-SIGMOD'00*, Dallas, TX, 2000, pp.1-12.

10. B. Liu, W. Hsu and Y. Ma, "Integrating classification and association rule mining," in *Proc. KDD98*, 1998, pp.80-86.

11. H. Mannila and H. Toivonen, "Discovering generalised episodes using minimal occurrences," in *Proc. KDD96*, AAAI, Portland, Oregon, 1996, pp. 146–151.

12. B. Ozden, S. Ramaswamy and A. Silberschatz, "Cyclic association rules," in *Proc. ICDE98*, *IEEE Computer Society Press*, Orlando, Florida, USA, 1998, pp. 412–421.

13. H. Lu, L. Feng and J. Han, "Beyond intratransaction association analysis Mining multidimensional intertransaction association rules," *ACM Transactions on Information Systems*, vol.4, pp. 423–454, 2000.

14. M. Dunham, "*Data Mining: Introductory and Advanced Topics*," Prentice Hall, Sydney, 2003.

15. Z. Liang, T. Xinming, L. Ling and J. Webliang, "Temporal association rule mining based on T-Apriori algorithm and its typical application," in *Proc. International Symposium on Spatio-temporal Modeling, Spatial Reasoning, Analysis, Data Mining and Data Fusion*, 2005.

16. H. Ning, H. Yuan and S. Chen, "Temporal Association Rules in Mining Method," in *Proc. IMSCCS'06*, IEEE Computer Society, vol.2, pp.739-742, 2006.

17. E. Winarko and J.F. Roddick, "Discovering richer temporal association rules from interval-based data," in *Proc. DaWaK'05*, LNCS, Springer, Copenhagen, Denmark, Vol. 3589, pp. 315–325, 2005.

18. V.S. Tseng, K.W. Lin and J.C. Chang, "Prediction of user navigation patterns by mining the temporal web usage evolution," *Soft Computin*g, vol. 12, pp. 157–163, 2008.

19. E. Nazerfard, P. Rashidi and D.J. Cook, "Using association rule mining to discover temporal relations of daily activities," *in Proc. ICOST'11*, Bessam Abdulrazak, Sylvain Giroux, Hélène Pigot, Bruno Bouchard, and Mounir Mokhtari (Eds.). Springer-Verlag, Berlin, Heidelberg, 2011, pp. 49-56.

20. X. Xi, E. Keogh, C. Shelton, L. Wei and C.A. Ratanamahatana, "Fast Time Series Classification Using Numerosity Reduction," in *Proc. 23rd Int'l Conf. on Machine Learning*, New York, 2006.

21. P. Revesz and T. Triplet, "*Temporal data classification using linear classifiers,*" Advances in Databases and Information Systems, pp. 347-361, 2009.

22. P. Sharma and K. Saxena, "Temporal weighted association rule mining for classification," *International Journal of Computer Theory and Engineering*, vol. 4, no. 5, pp. 816-820, 2012.

23. V. Bhuvaneswari and P. Umajothy, "Mining temporal association rules from time series microarray using Apriori algorithm," *Review of Bioinformatics and Biometrics*, vol. 2, no. 2, pp. 29-36, 2013.

24. G. Pyun and U. Yun, "Mining top-k frequent patterns with combination reducing techniques," *Applied Intelligence*, Springer Journals, vol. 41, pp. 76-98, 2014.

25. S. Zhang, X. Wu, C. Zhang and J. Lu, "Computing the minimum-support for mining frequent patterns," *Knowledge Information System*, vol. 15, pp. 233-257, 2008.