ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2011.0026

A NOVEL ALGORITHM FOR ASSOCIATION RULE MINING FROM DATA WITH INCOMPLETE AND MISSING VALUES

K. Rameshkumar
Department of Information Technology, Hindustan University, Tamil Nadu, India
E-mail: rameshkumar_phd@yahoo.co.in

Abstract
Missing values and incomplete data are a natural phenomenon in real datasets. If the association rules mine incomplete disregard of missing values, mistaken rules are derived. In association rule mining, treatments of missing values and incomplete data are important. This paper proposes novel technique to mine association rule from data with missing values from large voluminous databases. The proposed technique is decomposed into two sub problems: database scrutinizes and rules mining phases. The first phase is used to reexamine transactions which are useful to mine frequent itemset. The second phase is to mine frequent itemset amd construct association rules from valid database. This paper uses Apriori based algorithm in which proposed technique. The proposed technique is tested with synthetic and real T40I10D100K, Mushroom, Chess and Heart disease prediction datasets. Experimental results are shown that the proposed technique outperforms than robust association rule mining (RAR) and Association rules from data with Missing values by Database Partitioning and Merging (AMDPM) algorithm.

Keywords:
Data Mining, Association Rule, Frequent Itemset, Missing Values, Incomplete Dataset

1. INTRODUCTION

Discovering association rules is at the heart of data mining. It detects hidden linkages of otherwise seemingly unrelated data. These linkages are called rules. Those that exceed a certain threshold are deemed interesting. Interesting rules allow actions to be taken based upon data pattern. They can also help making and justifying decisions. Today real world databases contain incomplete data and missing values due to human, operational error, hardware faulty and many other factors. The qualities of knowledge (extracting, learning and decision problems) depend directly upon the quality of data set. In KDD and data mining tasks, the handling of missing values is important. Sometimes the missing values in database are often filled by the user or some other techniques. But these methods are unbelievable and at times those are replaced by noise [3] [4] [5] [6]. Association rule mining (ARM) is applied with these kinds of real databases. Missing values have not been considered of interest since [7] [8] because association rules were developed to explore transaction databases where the problem of missing attribute values does not practically exist. This problem will become important if the association analysis tries to find association between values of different attributes in relational tables where missing values are often inevitable.

This paper proposes new technique to detect association from data with incomplete and missing values. The proposed algorithm is called as ARDM (Association Rule mining from Data with Missing Values). The proposed technique is divided into two phases as follows: Database scrutinizing and Construction of association rules. The first phase has aimed to scrutinize attributes and transactions which mean to find valid ones. The second phase has extracted association rule from scrutinized database. The construction of association rule procedure consists of two processes. The first process aims to generate frequent itemset. The new transaction reduction approach to mine frequent itemset has been used in this proposed algorithm. The second process generates association rules with the help of the fastest rule generation algorithm. The proposed ARDM algorithm is compared with RAR and AMDPM algorithms.

The rest of the paper is organized as follows: Section 2 and 3 provides the definition of association rule mining and problem statement. The related works are given in detail in section 4. Section 5 describes the proposed technique and algorithms. Section 6 presents the experimental results and comparative study. Section 7 consists of conclusion and future enhancement.

2. ASSOCIATION RULE MINING

Let I = $\{i_1, i_2, \ldots, i_m\}$ be a set of items. Let D=$\{T_1,T_2,T_3,\ldots,T_n\}$ be a database of transactions, where each transaction T consists of a set of items such that $T \subseteq I$. Given an itemset $X \subseteq I$, a transaction T contains X if and only if $X \subseteq T$. An association rule is an implication of the form $x \Rightarrow y$ x is called antecedent, y is called consequent, where $x \subseteq X$, $y \subseteq X$ and $x \cap y \Rightarrow \phi$. The association rule $x \Rightarrow y$ Y holds in D with confidence c if the probability of a transaction in D which contains X also contains Y is c.

$$Confidence(x \Rightarrow y) = \frac{Support(x \Rightarrow y)}{Support(x)} = \frac{|xy|}{|x|} \quad (1)$$

The association rule $x \Rightarrow y$ has support s in D if the probability of a transaction in D contains both X and Y is s.

$$Support(x \Rightarrow y) = Support(x \cup y) = \frac{|xy|}{|D|} \quad (2)$$

The task of mining association rules is to find all the association rules whose support is larger than a minimum support threshold and whose confidence is larger than a minimum confidence threshold [1]. These rules are called the strong association rules.

3. PROBLEM STATEMENT

The above definitions are employed in this work. The database has missing values in their transaction sets. The above support threshold and the confidence threshold are not effective to mine frequent itemset as well as association rules. This

proposed work modifies support and confidence threshold. A transaction T_m becomes deprived for itemset X if T_m contains missing values for at least one attribute value of an item in X. These transactions are called the deprived transactions. It denotes $Depr(X)$, the set of deprived transactions for X, D_X, the subset of D containing X, and $|D|$ the number of records in D. The valid transactions of an itemset X, which can be utilized to evaluate association rules derived from X, is valid(X) = D−*Depr*(X). The association rules are evaluated by support, confidence and representativity. This paper has modified the existing formulae of support and confidence as follows;

$$Support(x \Rightarrow y) = Support(x \cup y) = \frac{|xy|}{|D| - |Depr(xy)|} \quad (3)$$

$$Confidence(x \Rightarrow y) = \frac{Support(x \Rightarrow y)}{Support(x)} = \frac{|Dxy|}{|Dx| - |Depr(y) \cap Dx|} \quad (4)$$

Representativity threshold [10] is added with the proposed technique. It is needed to restrict the influence of itemsets that are not observed a lot (i.e. all tuples in D have missing values for some of the attributes) on confidence and support. The sample of D that has no missing values for the attribute set of X must be a representative sample.

$$representativity(x \Rightarrow y) = \frac{|valid(xy)|}{|D|} \quad (5)$$

The problem of mining association rules from data with missing value is to find all the rules that satisfy specified minimum support, confidence and representativity thresholds. This proposed problem can be decomposed into two processes like traditional apriori approach as shown below:

Step 1: Generate maximum all k-itemsets that have support above the user specified minimum support and representativity.

Step 2: For each maximum k-itemset found in Step 1, Construct all association rules that have more than the user-specified minimum confidence then the association rule $X \Rightarrow Y$ is derived.

4. RELATED WORKS

4.1 RAR ALGORITHM

The main idea of the RAR algorithm [6] is that each itemset holds a set of transaction identifier lists of disabled transactions. Since it is based on the Apriori algorithm, it finds frequent itemsets for each pass. At the first database scan, it generates a transaction identifier list of disabled transactions for each item. At each pass, it produces a set of candidate itemsets from the frequent ones found at the previous pass. An item which is not frequent at pass k is not considered after pass $(k + 1)$. The transaction identifier list of disabled transactions of itemset X is the union of the same for each item in X. The RAR algorithm then scans all the transactions to obtain the count supports of the candidate itemsets, and determines the frequent ones. It reduces the number of generated candidate itemsets by using the down-closed property of the itemset. if an itemset is frequent, all its sub-itemsets have to be frequent. However, this property is not always consistent for data with missing values. Since it does not find itemsets that contain X, it cannot find $X=>Y$. Thus, it is not always possible for this algorithm to find all association rules.

4.2 AMDPM ALGORITHM

The AMDPM algorithm has used a pattern-growth based technique for mining association rules from data with missing values. It consists of two processes: database partitioning and association rule mining. This process generates database partitions. The database is divided into partitions so that each partition consists of records that have the same missing values in the same attributions. The association rule mining process utilizes these database partitions. The AMDPM algorithm does not perform data imputations, and it utilizes all records not containing missing values for each association rule. This approach divides the database into database partitions, to count the support of itemsets for each database partition, and to mine association rules by combining some of the database partitions. Unnecessary record counting is avoided by estimating global support from the local support counts of local frequent itemsets, and the processing cost is reduced by reutilizing a constructed partial prefix-tree structure and merging redundant database partitions. Experimental results showed that the AMDPM algorithm can always find all association rules satisfying minimum thresholds, and it can attain good performance by reutilizing trees and merging redundant database partitions. The drawback of AMDPM is as follows: the number of database partition reduces performance of AMDPM.

5. PROPOSED ARDM ALGORITHM

This section proposes a novel technique to mine association rules from data with missing values. The proposed algorithm adopts Apriori approach, and uses partitioned databases. It consists of two phases: database scrutinizing phase and construction of association rule phase. The association rule generation phase consists of the following two processes: frequent itemset generation and Association rule construction.

5.1 DATABASE SCRUTINIZING PHASE

This procedure scans transactions and attributes in the database. The scan procedure is to restrict the influence of itemsets that are not considered a lot. The database is divided into partitions so that each partition consists of transactions that have the missing values in the same attributes. When an attribute set is T_i, this work denotes $DBP(T_i)$ as the database partition that consists of transactions containing no missing value of attributes in T_i. The transactions are evaluated using representativity threshold and attributes are evaluated using representativity threshold and support threshold.

```
procedure db_scrutinize (Database DB, minSupp α,
                         minRepr δ)
begin
Step 1:  for all Ti ϵ DB do begin
Step 2:  if not DBP(Ti) then begin
Step 3:  Create database partition DBP(Ti)
Step 4:  Let DBP(Ti) = ϕ
           else
Step 5:  if Repr(Ti) is less then minRepr δ then begin
Step 6:  insert Ti in exclude transaction list
Step 7:  delete Ti from DB
           else
Step 8:  for all Ij ϵ Ti do begin
Step 9:  if Supp(Ij) is less than minSupp α ||
         Repr(Ij) is less than minRepr δ then begin
Step 10: insert Ij in exclude item list
Step 11: delete Ij from Ti
           end if
Step 12: Let modify Ti with remaining item Ij next for
Step 13: find missing Ij ϵ Ti
Step 14: select database partition DBP(Ti)
Step 15: Let Ti assigned into DBP(Ti)
           end if
         next for
Step 16: Merge DBP(Ti) and assigned into SDB
         end procedure
```

Fig.1. Procedure to scrutinize the database

This proposed database scrutinizing phase generates database partition and deletes unnecessary attributes which means attributes are not satisfied for user specified minimum representativity threshold from database. The transaction T_i is deleted from database which fails to satisfy minimum representativity threshold. It is also inserted into excluding transaction list. The attributes in the transaction satisfy user specified minimum representativity, this proposed procedure calculate support and representativity of all item I_j in transaction T_i. If any item I_j does not satisfy minimum support threshold and minimum user specified representativity threshold together, the item I_j will be deleted from the transaction T_i. After being completed the transaction T_i, remaining items I_j are assigned into transaction T_i. Finally this procedure reads the transaction T_i to find available items I_j; the transaction T_i is assigned into DBP(T_i). The database partitions DBP(T_i) are collected into a single database SDB (final step).

5.2 ASSOCIATION RULE MINING PHASE

The association rules are mined by using database partition DBP(Ti) which are created by the above database scrutinizing phase. This phase adopts Apriori technique using transaction reduction based approach. This procedure is divided into two processes as follows: Generation of frequent itemset and Construction of association rules.

5.2.1 The Generation Process of Frequent Itemset:

The transaction reduction technique which is proposed by [9] is used for this generation process of frequent itemset. This procedure is an improved version of AprioriTid and HEA algorithms. The representativity threshold is added with this procedure. The proposed procedure is as follows.

```
procedure find_fim(Database SDB, minSupp α, minRepr δ)
begin
Step 1:  L1 = {large l-itemsets}
Step 2:  C̄1 = Database D; (With all items not in L1 and
                ∀Ti Titems =1 removed)
Step 3: for (k=2; C̄k−1 ≠ Φ ; k++) do begin
Step 4:        Ck = Apriori_gen( Lk−1 )
Step 5:        C̄k = Φ
Step 6:  for all c ϵ Ck do begin
Step 7:        C̄ = Φ
Step 8:        Tc ={t.TID | t ϵ C̄k−1 ,(c – c[k]) ϵ
                 t.set-of-itemsets ∧ (c – c[k]) ϵ
                                 t.set of itemsets)}
Step 9:          if Supp|Tc| ≥ α and Repr Tc |
                                 ≥ δ then begin
Step 10:           Lk = ⋃k {c}
Step 11:       for all p ϵ Tc do begin
Step 12:         if (Titems > k) then begin
Step 13:           C̄ = ⋃k < p,c >;
                 end if
               end for
Step 14:         if |C̄| ≠ 1 then begin
Step 15:           C̄k = C̄
                 end if
             end if
         end for
       end for
Step 16: Ans = ⋃k Lk
end procedure
```

Fig.2. Procedure to mine frequent itemsets

5.2.2 Construction of Association rule process:

Construction of Association rule is a straight forward method. This proposed technique uses the following procedure which is proposed in [1].

```
                procedure gen_AR(k-itemsets)
begin
Step 1: for all large k-itemsets l_k , k ≥ 2 do begin
Step 2:    H_1 = {consequents of rules derived from l_k
                  with one item in the consequent };
Step 3:    Call ap-genrules(l_k, H_1);
end procedure

procedure ap-genrules(l_k: large k-itemset, H_m: set of
                                  m-  item consequents)
begin
Step 1: if(k > m+1) then begin
Step 2:   H_m+1= apriori-gen(H_m);
Step 3:    for all h_m+1 ϵ H_m+1 do begin
Step 4:          conf =support(l_k)/support(l_k – h_m+1);
Step 5:          if(conf ≥ minconf) then
Step 6:             output the rule (l_k -h_m+1)→ h_m+1
                       with confidence= conf and
Step 7:             support=support(l_k);
          else
Step 8:             delete h_m+1 from H_m+1;
        end if
Step 9:          call ap-genrules(l_k, H_m+1);
       end for
end if
end procedure
```

Fig.3. Procedure to construct association rules

6. PERFORMANCE AND COMPARATIVE STUDY

The proposed ARDM algorithm is implemented on a Windows 7 machine (1.66 GHz Celeron Dual Core processor, 1 GB RAM and 160 GB Hard disk drive) and performance evaluations are also studied. This proposed ARDM algorithm is implemented in Java. This paper used RAR and AMDPM algorithms to prove the efficiency of proposed ARDM algorithm.

Table.2. Datasets

Dataset	# of Rows	# of Columns
T40I10D100K	100000	942
Mushroom	8124	119
Chess	3196	37
Heart Disease Prediction	655	25

Table.2 contains four datasets which are used in this performance study. This work appends missing values in these dataset by selecting attribute at random. The missing values append per attributes as 10%, 20%, 30%, 40%, 50% and 60%. This proposed work does not concentrate to generate missing values in datasets. If the missing values increase in the database, the creation of the database partition also increases. The performance of proposed ARDM algorithm decreases as the missing values or database partition increases. The figures 4, 5, 6 and 7 show the comparative study of execution time between proposed ARDM and RAR, AMDPM algorithm varying minimum support threshold using T40I10D100K, Mushroom, Chess and Heart disease prediction dataset respectively. The ratio of missing values is set as 25%, MinConf threshold is set as 70% and MinRepr is set as 50%. The processing cost of proposed ARDM algorithm can be reduced by using transaction reduction method to mine frequent itemset. Most of the transactions are replaced by the earlier similar transaction identifier (ids) which are effectively used for support calculations and easy to construct frequent patterns.

The following figures 4, 5, 6, and 7, the performance of proposed ARDM algorithm is better than small dataset into high density dataset. In heart disease prediction dataset, the execution time of proposed ARDM is better than RAR algorithm in the percentage of 5 to 12 and AMDPM in the percentage of 3 to 4. The execution time of proposed ARDM is reduced with chess and mushroom dataset in the ratio of 8% to 12% and 13% respectively. In T40I10D100K dataset, the proposed ARDM performs better than RAR, AMDPM and earlier datasets. It is reduced for about 28% to 30% of execution time compared with RAR algorithm. It is reduced for about 25% to 28% of execution time compared with AMDPM algorithm. So this proposed ARDM algorithm performs better than big dataset while compared with small dataset. These improvements are achieved by using the new transaction reduction method which effectively handles the transaction and buffer management. Figures 8, 9, 10 and 11 show the comparative study of execution time between proposed ARDM and RAR algorithm varying minimum representativity threshold using T40I10D100K, Mushroom, Chess and Heart disease prediction dataset respectively. The ratio of missing values is set as 25%, MinConf threshold is set as 70% and MinSupp is set as 50%.

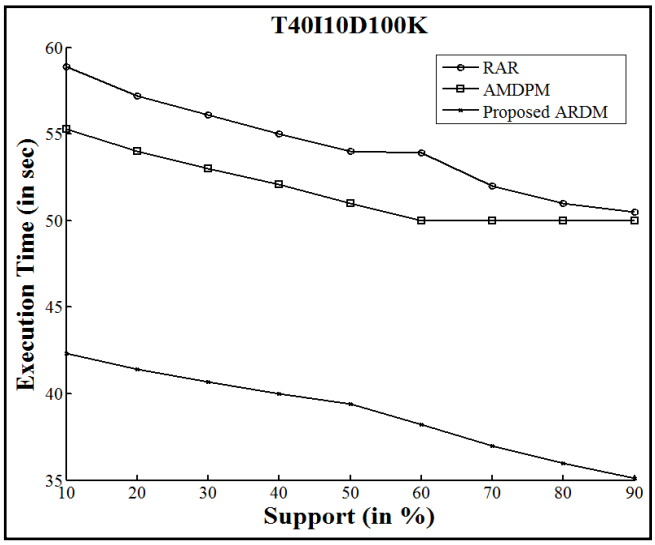

Fig.4. Execution time comparison between proposed ARDM and RAR algorithm with varying support using T40I10D100K dataset

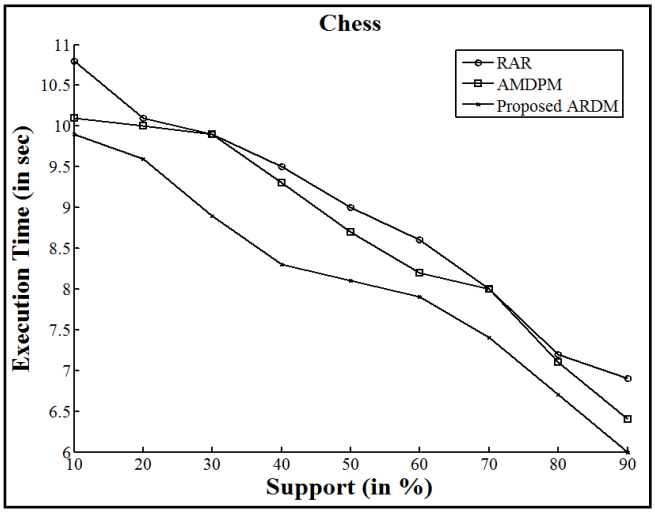

Fig.5. Execution time comparison between proposed ARDM and RAR algorithm with varying support using mushroom dataset

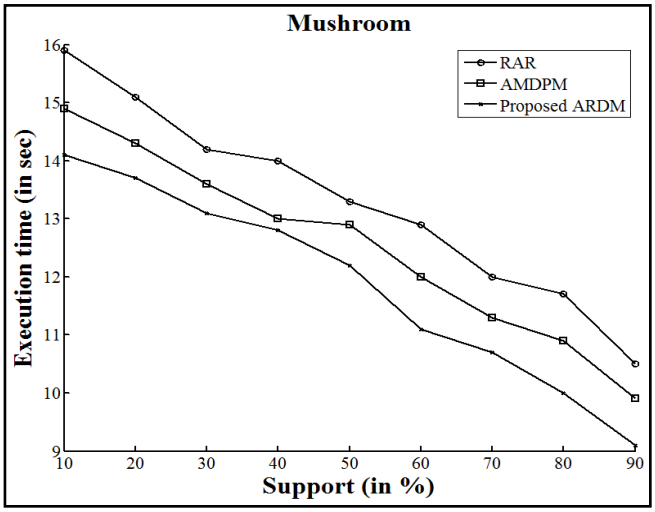

Fig.6. Execution time comparison between proposed ARDM and RAR algorithm with varying support using chess dataset

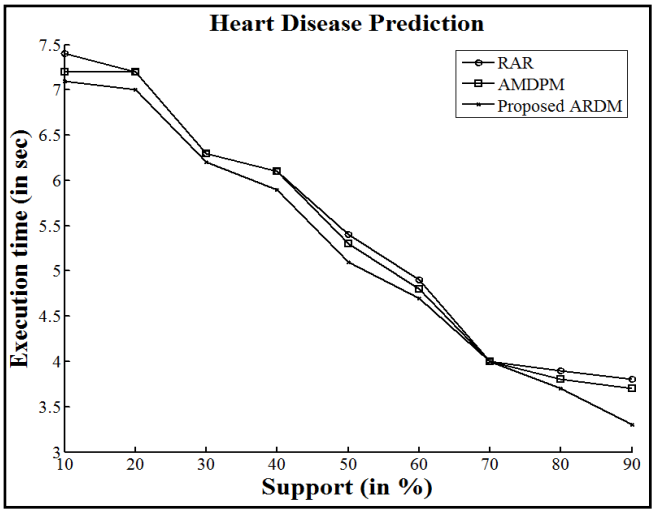

Fig.7. Execution time comparison between proposed ARDM and RAR algorithm with varying support using heart disease prediction dataset

The above figures show the comparative study of execution time between proposed ARDM and RAR algorithm with varying representativity threshold (10% to 50%). Here the performance of proposed ARDM algorithm increases as the representativity threshold increases. In the heart disease prediction, chess and mushroom dataset, the performance decreases as 7%, 10% and 11% respectively while compared with RAR performance. The performance of proposed ARDM algorithm is the ratio of 39% to 29% with T40I10D100K dataset compared with RAR performance. The figures 12 to 15 show the comparative study of execution time between proposed ARDM and RAR algorithm varying ratio of missing values threshold using T40I10D100K, Mushroom, Chess and Heart disease prediction dataset respectively. The MinConf threshold is set as 70% ,MinSupp is set as 50% and MinRepr is set as 50%. The figures 12, 13, 14 and 15 show comparison of execution time between RAR with proposed ARDM algorithm with varying percentages of missing values in the above mentioned datasets. In T40I10D100K dataset (high density), the execution time is reduced about 28% to 30% compared with RAR algorithm.

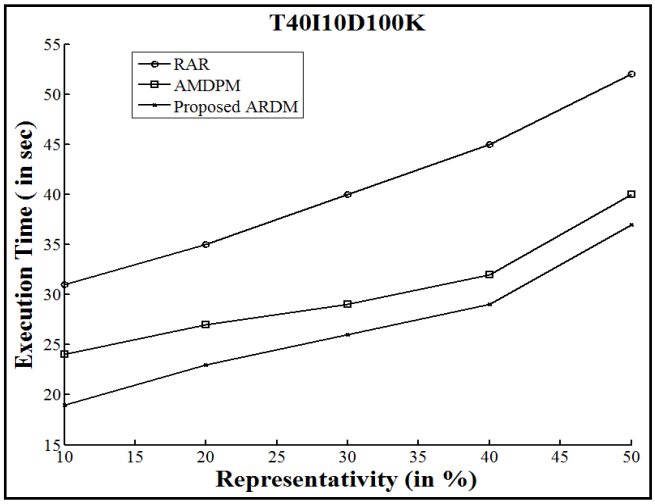

Fig.8. Execution time comparison between proposed ARDM and RAR algorithm with varying representativity using T40I10D100K dataset

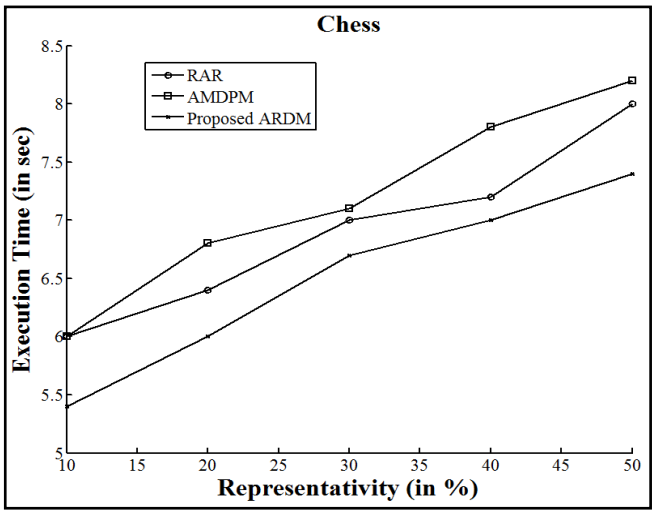

Fig.9. Execution time comparison between proposed ARDM and RAR algorithm with varying representativity using mushroom dataset

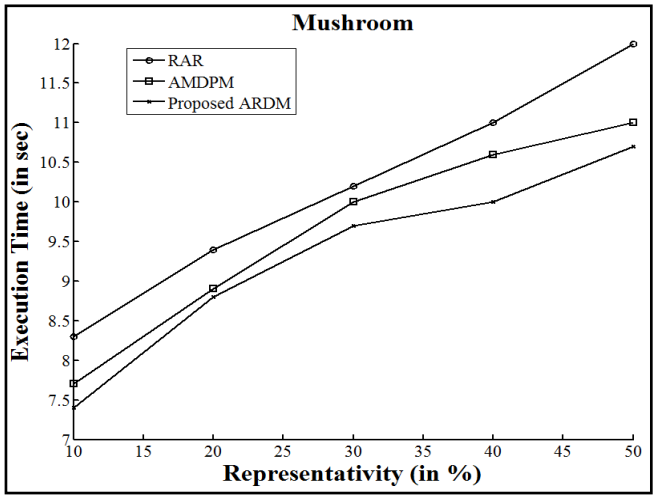

Fig.10. Execution time comparison between proposed ARDM and RAR algorithm with varying representativity using chess dataset

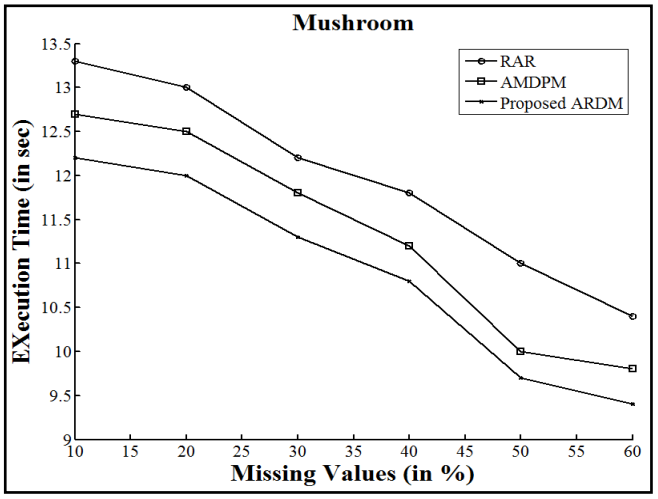

Fig.13. Execution time comparison between proposed ARDM and RAR algorithm with varying ratio of missing values using mushroom dataset

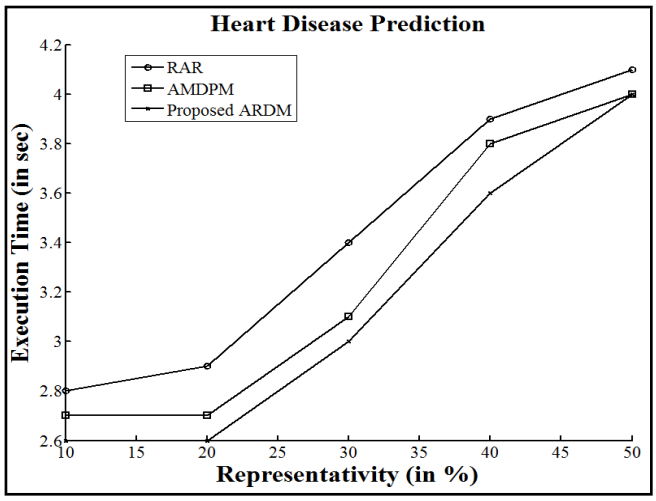

Fig.11. Execution time comparison between proposed ARDM and RAR algorithm with varying representativity using heart disease prediction dataset

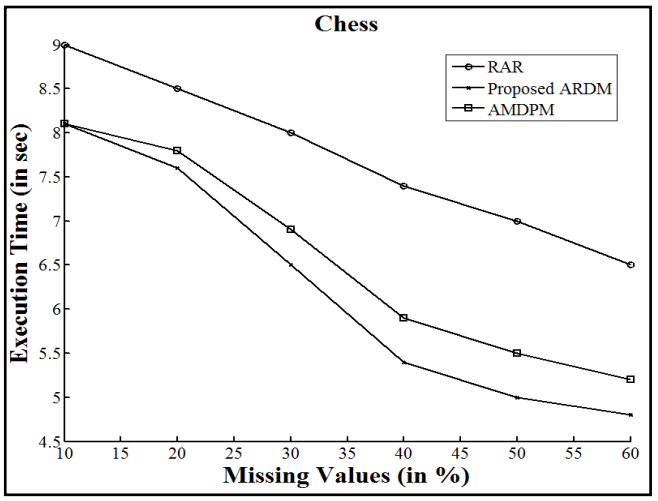

Fig.14. Execution time comparison between proposed ARDM and RAR algorithm with varying ratio of missing values using chess dataset

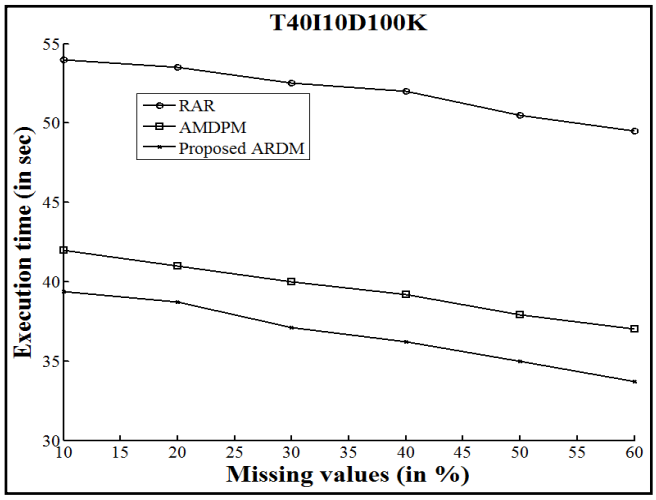

Fig.12. Execution time comparison between proposed ARDM and RAR algorithm with varying ratio of missing values using T40I10D100K dataset

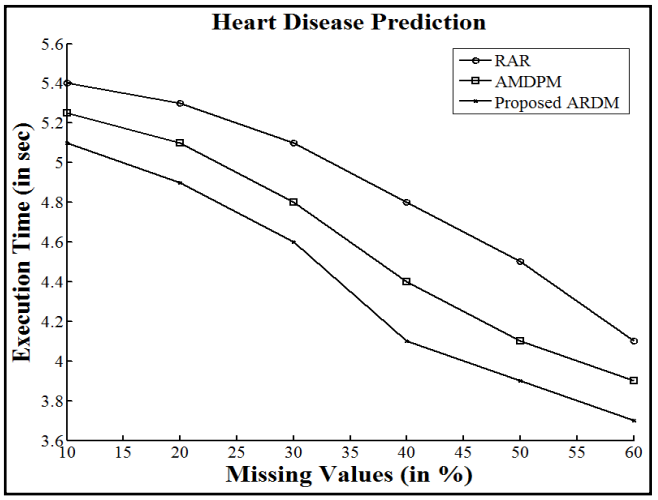

Fig.15. Execution time comparison between proposed ARDM and RAR algorithm with varying ratio of missing values using heart disease prediction dataset

This section summarizes the execution time of other datasets. The execution time is better than that of RAR algorithm with ratio of 8% to 10% AMDPM algorithm with 7% to 9%.

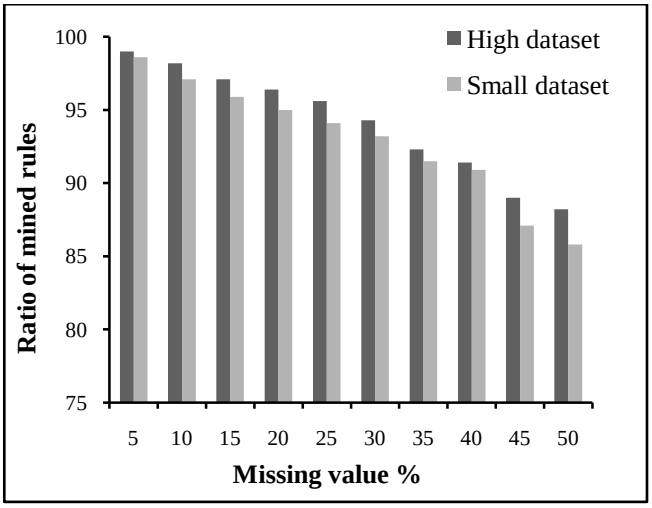

Fig.16. Comparison of percentage of missing values with ratio of mined rules

The above Fig.16 shows the ratio of mined association rule with percentage of missing values by RAR algorithm. If the proposed ARDM algorithm mined 100 rules with 5-50 percentage of missing values in the dataset, Only the RAR algorithm is mined 99.0 – 85.8 percentage of rules. The missing values increase as the ratio mined rules decrease. The advantages of proposed technique are: The percentages of missing values are not affected by ratio of mined rules. In AMDPM algorithm, the number of database partition reduces performance of the algorithm. But this work proposes the creation of new database, these kinds of database creations avoid so many unwanted database partition process. This is one of the major advantages of the proposed work. The proposed procedure arranges the transactions in order through database scrutinizing phase which means that the number of items avail in the transaction. It is of much help to update transaction reduction approach with the proposed frequent itemset mining process. The proposed technique outperforms when the ratio of missing value is low and high, and also when support is minimum and maximum level, and when representativity threshold is low and high. This work is not experimented with high ratio of missing values above 60% in the dataset.

7. CONCLUSION AND FUTURE ENHANCEMENT

This paper proposes a novel technique to mine association rule from data with incomplete and missing values. This proposed technique has successfully scrutinized transactions which are satisfied with minimum support and representativity threshold. This work modifies support, confidence and representativity thresholds that are efficiently calculated in data with incomplete and missing values. The modified transaction reduction approach avoids so many unwanted transaction processes. Experimental results showing the proposed algorithm can always find all association rules satisfying minimum thresholds, and also it can produce good performance in high and small datasets. In future, this work will be implemented with various real time domains like web and medical datasets.

REFERENCES

[1] Agrawal R and Srikant R, "Fast algorithms for mining association rules", *Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Chile*, pp.487–499, 1994.

[2] Frank A and Asuncion A, UCI Machine Learning Repository - http://archive.ics.uci.edu/ml/datasets.html, 2010.

[3] L. Brieman, J.H. Friedman, R.A. Olshen and C.J. Stone, "*Classification and Regression trees*", Boca Raton, CRC Press, 1984.

[4] Celeux.G, Le, "traitement des donnees manquantes dans le logiciel SICLA", *Rapports Techniques No.2, INRIA*, 1988.

[5] J. Ross Quainlan, "C.4.5: Programs for machine learning", *Morgan Kaufmann Publisher*, 1993.

[6] Ragel and B. Cremilleux, "Treatment of missing values for association rules", *Proceedings of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 258–270, 1998.

[7] R. Agrawal, T. Imielinski and A.N. Swami, "Mining associations rules between sets of items in large databases", *proceedings of the ACM SIGMOD Conference on Management of data*, Vol. 22, No. 2, pp. 207-216, 1993.

[8] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A.I.Verkamo, "Fast discovery of association rules, in advances in knowledge discovery and data mining", *AAAI press*, pp. 307-328, , 1996.

[9] R.E. Thevar and R. Krishnamoorthy, "A new approach of modified transaction reduction algorithm for mining frequent itemset", P*roceedings 11th International Conference on Computer and Information Technology*, pp.1-6, 2008.

[10] Takahiko Shintani, "Mining Association Rules from Data with Missing Values by Database Partitioning and Merging", *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAR'06)*, pp. 193-200, 2006.