Design and Analysis of an Efficient Multi-Relational Decision Tree Learning Algorithm

Chhutten Singh Yadav¹, Abhishek Kumar², Ankit Kumar³ and Pankaj Dadheech³

¹School of Management Studies, Graphic Era Hill University, Dehradun, Uttarakhand, India
²Assistant Professor, Department of Computer Science & IT, Jain (Deemed to be University) Bangalore, India
³Assistant Professor, Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur, Rajasthan, India

E-mail: dr.c.s.yadav82@gmail.com

Abstract. In today’s day of modern era when the data handling objectives are getting bigger and bigger with respect to volume, learning and inferring knowledge from complex data becomes the utmost problem. The research in Knowledge Discovery in Databases has been primarily directed to attribute-value learning in which one is described through a fixed set tuple given with their values. Database or dataset is seen in the form of table relation in which every row corresponds to an instance and column represents an attribute respectively. In this paper a New framework is introduced a much more sophisticated and deserving approach i.e., Hybrid Multi-Relational Decision Tree Learning Algorithm which overcomes with Exiting technology drawbacks and other anomalies. Result show that Hybrid Multi-Relational Decision Tree Learning Algorithm provides certain methods which reduces its execution time. Experimental results on different datasets provide a clear indication that Hybrid Multi-Relational Decision Tree Learning Algorithm is comprehensively a better approach.

1. Introduction

The word data mining is extensively used interchangeably in applying two different processes for discovery of knowledge in databases (KDD) and the other one is data prediction. The earlier version delivers information which provides a readable format that which can be inferred or interpreted by any user. On the other hand forecasting modelling or predictive data mining presents predictions for the events which can be developed in future. Moreover, some mining processes viz. neural networks are mainly geared toward prediction and recognition of pattern besides knowledge discovery which predicts future trends and behaviour in order to make proactive knowledge-driven decisions [1].

1.1 Techniques of Data Mining

Various major data mining techniques are as follows:

Classification: Classification is a process which constructs a basic prototype that draws data classes or concepts to predict unknown class label of a class object. It involves searching those concepts which help in partitioning the database into different groups.
Classification method is performed by:
- If-then rules
- Decision Trees
- Neural Network Model

Clustering: It can be defined as segregation of data into groups of similar objects. This technique is quite important topic in the field data mining. It is generally regarded as the very initial step towards discovery of knowledge. It is the act of segregating data objects into a set of different classes referred as clusters, in such a way that objects have huge similarity with among themselves [2]. It figures data by its clusters. In accordance with the machine learning paradigm

Clusters refers to invisible arrangement. The searching or looking in these groups basically is an unsupervised learning, and its final system depicts a new concept. Within a practical aspect clustering performs a crucial and indispensable act in data mining. Applications viz. data exploration, data regeneration [3], analysis of web traffic, medical works, and genomal biology.

Association: It is an act for discovering meaningful relation through variables from large databases. It is meant to determine strong rules which are found in databases by employing various measures of valid paradigms. It has two goals:
- To find frequent item sets
- To find different association rules between them

Given a database of shopping mall if one would like to find those items which are purchased together on a regular basis within the same transaction, using a mined rule from transactional database saying:

buys (P, coke)→buys(P, chips) [support = 2%, confidence = 60%]

Where P is the customer's vector. 60% trust or assurance means that if a buyer buys a coke, 60% of the customer is more going to buy chips. 2% of the sale under review suggests that 2% implies that the coke and the chips have been acquired together. This association rule involves a single attribute or predicate (buys) that repeats. Support means how often X and Y occurs altogether on total transactions. Confidence estimates how an item is dependent on another item. Thus, the association depicting a pattern that is very common is a pattern that comes regularly in the database and which is very distinctly inferred from the end user.

2. Literature Review

When relational learning was introduced, the suggested educational algorithms that could be used to convert multiple relational databases into uni-relationships were the development of novel features of a central relationship that eliminated information or aggregated information from differential tables in the related database. Different versions of this system have been used in this system.

More recently, proposed a novel transformation-based ILP apprenticeship with previously mentioned systems that can handle non-destructive relationships, un-labelled literally, and is geared toward the domain of business where relational databases are often fundamentally simple. However huge in size. This concept is called Relay on Aggregation or sometimes (RAELGS), and it has been successfully applied to learning problems from the business domain to the domain of biology [4]. Continuing the same trend, [5] is an easy-to-use but powerful propositional classifier for relational domains, spawning a use. The method maintains the related structures and objects "flat" when required by the algorithm. Interpretation of a topic can help in making assumptions or understanding about the issues related to the interpretation; therefore, assumptions with a high degree of uncertainty regarding an object are returned to the parent dataset and then used to determine the posterior burdens or information about a potentially relevant object.
Garcia et al. (2012) investigates about the performance of different classifier using different class imbalance ratio and different resampling techniques. Authors also categorizes classification of data as binary class and multiclass dataset. Many applications have multiclass dataset so converting them into binary class dataset. Authors experiments over 17 dataset in which some dataset are binary class and some are multiclass dataset. In these eight different classifiers and four different resampling techniques are used. Classifier performance is evaluated on four different evaluation metrics. From the paper oversampling outperforms under sampling technique in most of the datasets used.

Fernandez et al. (2013) discusses about two binarization techniques for converting multiclass dataset into binary class dataset. These two techniques are One Vs. One (OVO) and One Vs. All (OVA). In One vs. One classifier will be built for each possible pair of classes. In One vs. All classifiers will be built for each class with considering that class as positive class and other all classes as negative class.

Another problem with Condensed Nearest neighbour is that as it follows 1-NN rule so only one instance from majority class is selected for 1-NN rule. There may be possibility that any dataset may consist of two subset of any class so that under sampling will occur in only one subset and another subset will remain untouched so it will lead to the problem of Over fitting while building any classification model (Barella et al., 2014).

Das et al. (2012) proposed Cluster Based Under sampling which resolves class imbalance problem by deleting instances from majority class which are in overlapping condition in any dataset. In this all the training dataset will be divided into K clusters discarding instances from majority class which satisfy $0 < R < 1$ and $R > \$$. Where: $R = \frac{C_i}{|C|}$

$C_i$ is the number of minority instances in cluster i and $|C|$ is the total number of instances. $\$ is any empirical value.

Thus it will lead to the removal majority class instances by removing class overlapping and also there is removal of instances from majority class thus solving the problem of class imbalance.

Sowah et al. (2016) presents another Cluster Based Under sampling Technique (CUST) which further improves the performance of the classifier model for the imbalanced dataset with the removal of all inconsistent instances from majority class by using k-means clustering.

The Probabilistic Relational Model Framework, [5] proposed an extension in the very language to enable PRMs to continue learning. Bayesian networks provide a natural framework for this type of education. This is different from the static relational method which takes a snapshot of the relational database at random, generates the classifier, and uses it for unsafe instances. [6] The proposed dynamic strategy changes the model of education over time to represent reality in a more perfect way. Structural data mining techniques have also been proposed in the form of graphs [7][8] In this scenario, the objects in the data represent the top of the graph, and the relationships between objects are represented by both directional or unchanged edges in the module graph. Finds the patterns present in the corresponding system graphs. If a pattern is found, then it is added to the graph so that it simplifies the instance of the layer with the simplification by replacing it.

In the next section, the multi-related framework is discussed in more detail based on the multi-related decision tree learning framework. Accordingly, it also identifies the corresponding drawbacks and provides a better approach to overcome these barriers.
3. Proposed Methodology

3.1. Multi-Relational Data Mining (MRDM)

The current work provides a wide description of the framework proposed by Knobbe et al. [9] in order to search for meaningful insights through relational databases containing the queries of SQL. It also provides a description for the implementation of MRDTL algorithm that performs the development of decision trees on relational databases with the help of some earlier approach.

3.1.1. Relational Vocabulary

A database that is relational in nature contains set of tables, \( Z = \{1, z_2, \ldots, z_n\} \) with some other association in between tables. These associations are inferred as constraints so that in which way a record from one relation gets related to the data from other table. All the tables and its related associations can be referred to as relations. We will use them distinctly. The columns under a particular table represent the attributes of the table and the rows represents each records. Each table must bear at least a key value that uniquely specifies the records. We will mark and represent this attribute as \( Z.K \). The other remaining attributes may be descriptive or foreign attributes. The attribute represented as foreign key may be defined as a major attribute for other table. These keys define linking’s between tables. The description of these types of relationship describes the multiplicity (one-to-many, many-to-one). Under the convention in [9], an attribute which is descriptive \( M \) of table \( Z \) will be shown as \( Z.M \), and the representing domain is \( D(Z.M) \). However, regarding foreign key it has to define its type of domain \( \text{Dom}[F] \) and its range may be represented as \( \text{Range}[F] \). \( F \) is a foreign key into the table \( Z \) and can be a primary key within table \( X \), then in that case \( \text{Dom}[F] \) is \( Z \) and its range may be described as is \( Y \). The foreign key relations are bidirectional. In cases an attribute tends from that table in which that attribute works as primary key and in which that attribute works as foreign key and the other one vice-versa. Let they be represented as \( F \) and \( F^{-1} \). In a database that is relational in nature, an object may have contain several records scattered across various tables which are interlinked through associations.

![Figure 1. Schema of the Mutagenesis Database](image)

For e.g., a chunk of data model taken from the mutagenesis database is utilized and described in Fig. 1 through an entity-relation diagram. The figure depicts that the database has three tables and have corresponding connecting four interrelations among themselves. A molecule contains one atom, and one bond at least (hence making a final of dual atoms per molecule). The table bond depicts the relation type in between the two atoms, hence all atoms should present in the tables.
3.1.2. Multi-Relational Data Mining (MRDM) Framework

Under this framework [10], even the data model consists of many tables, every one describing features of particular objects, in which there is only one object is kept under central analysis. The user has the choice to select the type of objects which is to be analysed by choosing one tables which is target table (the table will be referred to as \( T_0 = T_i \) for some values \( i \in \{1, ..., n \} \)). The definition and working of selection graphs is demonstrated [10][11].

Definition:

“Selection graph G” depicts a directed graph with N, E in which N depicts a group of triple values referred to selection nodes, in which T represents a table within data model, An empty conditional set C is represented on attributes within T viz. \( T.A \oplus c \); where \( \oplus \) represents obvious selection operators like “\( =, \leq, > \)” and \( s \) depicts a flag having all possible values either open or closed.

In addition E depicts the ordered set of tuples \( (q, p, e, a) \) referred to as selection edges, referring which \( p \) and \( q \) represents nodes of selection, ‘a’ depicts an linking in between i.e. \( p.T & q.T \) within the data class. ‘e’ depicts a flag with possible values that are both subsequently present and non-present values. The selection graphs have one node \( n_0 \) which corresponds to the target table \( T_0 \). The directed selection graphs may be depicted as directed graphs with proper labelling. For e.g. (Figure. 2) which depicts the data model of the database i.e Mutagenesis database [mutagenesis]. The graph below selects molecules which have charge less or equal to 0.392.

In the given above instance schema of relational database, aimed table represents Molecule table, and in that schema of molecule the main resultant attribute is finding to predict as the Label. An edge which is present in between tables depicts a join among the above described two tables with the help of using the join attribute which acts as a primary key with tablex works as foreign key with y in addition with two exceptions which are explained in the later course (see the act of adding a condition which is negated and consequently then adding a absent edge with a closed node). With the help of this edge which takes together with many of other conditions choose all records which satisfies all the conditions and which also relates to the join schema.

Other than this, an absent edge belonging to table x & y satisfying all the conditional list y. C takes these records in X which don’t corresponds to conditions list. Whichever sub graph which denotes an absent edge hence tends to conditions of a set of negative.

Algorithms which are used to convert graphs into SQL are showed [12]. They enumerated and described that the final SQL statements must not be incorporated in real and actual implementations. It was also proposed a corresponding architecture [13], in which these SQL statements employed as simple and primitive ways in a way so as to processed in much more efficient pattern. As main aim of this process is to check the “goodness” of the subsequent approach pointing out its merits and demerits, we didn’t showed the server part. Collection statistics is managed by Oracle DBMS server with search algorithm running on a different machine. Hence, we applied SQL statements in order to aggregate statistics which searches for possible refinements. Utilizing the algorithms proposed in [14], showed simplifies to the following:

\[
\text{select distinct } X0.molc\_id \text{ from Molecule } X0, \text{ Atm } X1 \\
\text{where } X0.molc\_id = X1.molc\_id \text{ and } T1.\text{charge} \leq -0.392
\]
and $X_0.molc\_id\ not\ in\ (select\ X2.molc\_id\ from\ Atom\ X2\ where\ X2.ch\ <=\ -0.39\ and\ X2.elmnt\ =\ 'b')$

Consequently, after the algorithm have traversed a selection graph, it can be seen as returning, a set of table_list which tends to all the open nodes within the given graph. The final list also contains the depiction of sub graphs having conditions which are negative. Hence, the format for general query concluding graph for selection is as follows:

\[\text{select distinct } X_0.pri\_key\ \text{from } \text{table-list} \]
\[\text{where } \text{join-list and condition-list}\]

Note that the table_list may not ever be void but join_list/ condition_list can have empty space. Hence with these conditions, these query must be managed in accordance.

Figure 2. An Instance of Selection Graph taken out of Mutagenesis Database

3.2 Multi-Relational Decision Tree Learning Algorithm (MRDTL)

It designs decision trees with multifaceted node patterns. The parent algorithm suggested in [15], which is called MRDTL, depicts a strictly logical induction expansion that we know as TILDE proposed in [16]. It uses the first-order logic space to view decisions (nodes) in a tree. In terms of gathering many unsettled documents inside a partnership database the data should be interpreted with the first-order logic. The MRDTL extends the methodology used by TILDE [17] to manage all the documents in relation databases. They use separate complexity operators, and the trees are induced according to the same logic. Contraire to the proposal version of the algorithm for the decision tree [18], the tree adds decision nodes into a successive refinement method to the degree that such termination conditions are identified. Whenever a termination requirement is met, instead it adds a leaf node that corresponds to its class. A suitable impurity measure influences the choice of decision node to be applied to each move (e.g., information gain). Therefore the MRDTL starts with a single root node containing all objects in the relationship database. The target table $T_0$ with the defined target attribute corresponds to this node.
The optimum refinement feature discusses all potential changes to the existing pattern \( S \) in respect of database \( D \), and then selects the optimum refinement in a greedy method (i.e., one that optimizes information gain). The current collection graph, the database configuration and the multiplicity of the relationships concerned govern the proper set representing the refining’s to be rendered at allocations throughout the process. S TER denotes and is a collection graph supplement (i.e., extracts items not picked by \( S \)) from the database.

General Working Structure of a Decision Tree Learning Algorithm:

**Tree Induction (T: tree, D: database, S: selection graph) Input: dbase dB, sel graph S**

**Output:** final root or node for the tree, \( T \)

**Step 1:** \( R: = \text{optimal-refinement}(S) \)

**Step 2:** if satisfying criteria(S)

**Step 3:** return node (leaf)

**Step 4:** else

**Step 5:** \( T: = \text{tree induction (D; R(S)))} \)

**Step 6:** \( T: = \text{tree induction (D; \overline{R}(S))} \)

**Step 7:** return node (\( T_1; T_2; R \))

The algorithm depicted above creates binary splits in the corresponding decision graphs. A possible change may be done to consider multiple splits whenever an attribute with condition on some query is selected for the optimal refinement in the graph. At the instance, we can get a single branch for each of the permissible values for the corresponding attribute.

3.2.1 Refinements

As seen above, a whole different number of permitted refinements can be added to the very first node representing the first \( T_0 \) at a moment where the target table with its corresponding target attribute has been selected to define a hypothesis compatible with the data in the training database. Six separate refining options were suggested for the graphs range. All three are fundamental refinements. This are all illustrated below by using suitable examples in the Mutagenesis database.

Mutual exclusion: the sub-sets associated with patterns should be mutually exclusive, created by the same parent, thus with complementary practices, two types of finishes were added (adding a condition and adding an edge with a node in addition).

Adding condition (positive): This refining makes it easy, but does not change the layout of the ideal selection node within Graph \( S \). The "atom.element = ‘b’ " is believed to have been completed. Earlier this atomic state package composed of of "atom.charge<= -0.392" as a single object and other void node molecular conditions. Therefore the resulting graph forms the specified structure after adding a discussed entity:

Add condition (negative): This condition is the complementary version of operation for the previous one. In [19] this refinement is elucidated as below. That node which is being refined if does not depict target table, then the refinement adds to it new absent edge linked from parent node to new closed node which represents a duplicate of that working selection node on which the refinement being done. The corresponding condition list with its join list should be duplicated or copied to the new node which is closed.
Figure 3. Demonstrating the “add (positive)” Condition

Figure 4. Demonstrating the atm

Figure 5. Demonstrating the Adding of Negative Condition
In this, only if the negated valued condition is made to be added into the whole graph then it will tend to that selection graph & corresponding its SQL query which is demonstrated in sec b) of the same figure. The final result is described in sec c).

Hence as earlier explained, the below SQL query as follows is for handling addition of negative condition:

```
Select distinct (T0.molc_id) From Mol T0, Atm T1
Where T0.molc_id=T1.molc_id And T1.ch <= -.392
And T0.molc_id not in (Select distinct (T1.molc_id) From Atm T1
Where T1.ch <= -.392 and T1.elmnt = 'b')
```

Segment a) described in figure is description regarding the subset from Mutagenesis database which tends to set of records chosen by graph mentioned above, and assume that desired condition which has made as negated and to be linked to previous graph is `atm.elmnt='n'`. With employing the technique for adding the condition which is negated proposed in [20], the selection graph, after refinement, the resultant described under part b) of Fig. 6. Its linking SQL translation is in its right.

![Diagram](image)

**Figure 6.** Adding the Condition (Negative) in Refinement when Object Connected Indirectly to m0

Despite the fact that the graph shown selects both molecules, the resultant graphs with the condition added of the previous with the negation must chose at least one molecule each. That graph which contains all the supportive condition select e2 and the condition that comprise the graph with its negation of that condition selects the second molecule i.e., e19 (i.e., molecules that doesn’t have of their atoms element ‘n’). But in the
case that graph depicted in part b) of Fig. 6 demonstrating the first way of including a negative(complementary) condition, chooses that all both molecules viz. e2, plus 19. Hence, the accurate graph that is complementary in nature regarding above condition is the one that is depicted in section 3. Hence, regarding such cases in which a negated condition is made to be linked or added on a particular node that is not directly linked with the target table's node, it is essentially important that whole graph is complemented the by involving an absent edge from node with sub graph representing the real graph with not negated condition linked to the corresponding node’s. The sole dissimilarity in between these newly resultant subgroup hand the previous original one is that it un anonymously bears a copy of the root node that is closed (Fig. 6 (c) for example).

The above mentioned method could has been made more effectual (in reference with of space and its time complexity) if instead taking the whole graph with only having the sub graph is copied to which the node being refined belongs. It means, we have to add just an absent edge from node to a copy of the first node linked with the sub graph on that which the linked node exist in which the main result has to be added belongs (for e.g. the Bond node provided). This replication for node must be done closed. In addition to this the nodes and edges which make the subgraph must be made and replicated into the sub graph.

Observe in the case if we use the initial approach (sec(c) of Fig. 6), instead of that described in the previous section, an edge which does not tends to instantiate any of the linked association within data model will be made connected to the trained target node linked to a replication of itself but that will be in closed state condition.

- **Adding present edge and open node**: Under this class of refinement, it starts up a relationship as a current edge with it comparing table as open hub. At that point these are added to the diagram.

\[
\text{Figure 7. Adding present edge and open node}
\]

Under this refinement, an affiliation is launched in the information model as a current edge alongside its table addressed as an open hub and afterward these are added to determination diagram.

- **Add absent edge and closed node**: Add missing edge and shut hub: This is integral to the past one. It starts up a relationship in the information model with a missing edge alongside its comparing table addressed as a shut hub and afterward added to the chart S.

\[
\text{Figure 8. Adding absent edge and closed node}
\]

A point which is important to check is that a closed node refers to that node which in not or cannot be in future refined either by involving or including more conditions with its attributes or linking edges through them.

- **Adding edge (present or absent) between two existing nodes**: It doesn’t comes under the category for refinements on its own rather it plays an vital or important constituent when including on a negative condition linked onto a selection node. Graphically, the “refinement” can be depicted as bellow:
Look-ahead refinement: In some cases during the course of refinement for a selection graph, at some point of time further addition of any refinement may not result in any information gain or improvement. Then in that case, the search made in that that single path is discarded and a resultant leaf portion or terminal node is incorporated instead.

But, in this case also by database definition it will not achieve any improvement because the number of records to be analysed is the same as before, hence the linked entropy of the new selection graph will seldom change with context on to its dependable parent.

This condition can be dealt only by introducing with certain sort of capability which is referred to as look ahead task to the learner which was earlier introduced in the approach TILDE [20]. Our implementation for MRDTL encompasses such look ahead technique when encountered with these above said circumstances in which no information gain is achieved positively although the final ending or satisfying criterion is not achieved in the same context. Assuming that a conditional branch induced in Mutagenesis Database has reached some condition.

Figure 9. Adding edge (absent and present) and closed node

Figure 10. Information Retrieval

After this no improvement can be done either by adding any condition or any edge. Then, in this case the sole idea is to check the corresponding effect of adding a fresh edge with its corresponding linking open node in addition to this any condition linked to that attribute for fresh node on first arm and its corresponding negation for the second one. In the above description the refinement for maximum information retrieval is depicted in Fig. 10. The continuous refinements done on the far-left arm is to include an edge taking of molecule to the table schema of atom with a condition (charge<= -0.392) corresponding to atom table and on its right arm with its corresponding negation.
3.2.2 Shortcomings of Multi Relational Decision Tree Learning Algorithm

Multi Relational Data Mining was primarily introduced by [21] which incorporates structured query language to sum up and retrieve information from multi-relational datasets. Based on this very framework, Multi Relational Decision tree learning algorithm was introduced [22] which was intended for the construction of selection classifiers for learning datasets which are multi-relational in nature. It was earlier shown that this proposed algorithm was considerably success on this stage for developing classifiers with constructed decision trees having accuracies which are comparable to the results which were obtained using other several algorithms. Though this technique is never been implemented practically on real world data sets, yet it suffers from two significant drawbacks from the view of multi-relational data mining which is explained below:

- **Slow performance with respect to its running time:**
  One of the major bottlenecks which came in the performance paradigm of the algorithm was its slow performance in terms of the running time. As all the prior proposed approaches based on multi relational data mining including this uses *selection graphs* to query relevant information from databases in order obtain the information and constructing statistics which is required for developing classifiers, it is shown using datasets used in KDD cup ’01 [23] that this problem is of considerable concern for the slow performance of the algorithm [24]. From the straightforward implementation of the algorithm based on the discussion in earlier part it develops an efficiency problem and makes the application of the algorithm very complex in real world and makes it totally infeasible. In practice, as we proceed in the construction of the decision trees, its corresponding selection graph grows rapidly [25]. And each time while refining an existing selection graph an unnecessary repeated work is performed each time by re-retrieving those instances which are previously covered by that selection graph. This repeated work causes a major hindrance in the performance of the algorithm and makes it consequently slow in its working [26].

- **Handling of Missing Values:**
  This problem of handling missing values in multi-relational datasets are encountered inevitably, as major part of the real world data sets have several missing attribute values [27].

### 4. Experiments & Results

With the consolidation of the revisions in MRDTL calculation the resultant methodology is similarly improved in each part. We gathered the information and the outcomes from the writing in setting of exactness factor [28] and contrasted and the best announced outcomes which we got utilizing Hybrid Multi-Relational Decision Tree Learning Algorithm [29].

| Table 1. Running Time Comparisons with Mutagenesis Database |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| **System**      | **B0**          | **B1**          | **B3**          | **B4**          |
| Progol          | 8695            | 4627            | 4974            | 6530            |
| Foil            | 4960            | 9238            | 500             | 502             |
| Tilde           | 41              | 170             | 140             | 352             |
| MRDTL           | 18              | 332             | 221             | 221             |
It is clear from the Table 2 That MRDTL approach takes almost one-thousandth of the time taken by other approaches. Hence MRDTL [30] again proves to be winner among other approaches. The graphical comparison on execution time factor is shown in figure 11.

![Figure 11. Graphical Comparison of MRDTL (Execution Time) with previous approaches](image)

We also compared the algorithm execution time for the same datasets with the ones provided to previous methods [31] in the literature and concluded that MRDTL-2 exceeds all previous approaches.

| Dataset     | Running Time (MRDTL) | Running Time (Hybrid MRDTL) |
|-------------|----------------------|-----------------------------|
| Mutagenesis | 52.15                | 28.45                       |
| Localization| 1256.38              | 202.9                       |
| Function    | 307.82               | 151.19                      |
| Thrombosis  | 198.22               | 127.75                      |

From the graph obtained, it is clear that the novel approach of Hybrid MRDTL takes almost one–sixth of the time taken by previous approach. Hence it is clear that the execution rate of Hybrid Multi-Relational Decision Tree Learning Algorithm is faster in comparison with its predecessor approach [32].
Figure 12. General Performance (Execution Time) Comparison

With all the comparisons in the implementation of the algorithm on different datasets based on the data provided in the literature it is clearly indicative that Hybrid Multi-Relational Decision Tree Learning Algorithm outperforms all its previous versions and proves to be clear winner in both aspects, accuracy and execution time respectively.

5. Conclusion
A comparative graphical analysis based on accuracy and time factor epitomizes clearly that Hybrid MRDTL overthrows all its predecessor approaches and convincingly proves to be a clear winner. Developments are under progress which are intended on Development of more intelligent approach for better performance of Hybrid MRDTL, more experimental evaluation on real world datasets. Incorporation of more efficient pruning approach without increasing its complexity factor. Providing more promising methods for the development of MRDTL Algorithm which is more extensive for solve the Real World Data Set Problem. Handling missing value: improved Multi-Relational Decision Tree Learning Algorithm incorporates an efficient approach for handling missing values by using Naïve Bayes classifiers. Nowadays datasets in real world are very common with missing values; hence improved MRDTL comprehensively enhances the capability and performance of the classification tasks done by multi- relational decision tree learning approach.

6. References
[1] J. Guo, J. Li and W. Bian, "An Efficient Relational Decision Tree Classification Algorithm," Third International Conference on Natural Computation (ICNC 2007), Haikou, 2007, pp. 530-534. doi: 10.1109/ICNC.2007.195.
[2] R. Rivera-Lopez and J. Canul-Reich, "Construction of Near-Optimal Axis-Parallel Decision Trees Using a Differential-Evolution-Based Approach," In IEEE Access, vol. 6, pp. 5548-5563, 2018. doi: 10.1109/ACCESS.2017.2788700.
[3] W. Kuang, Y. Chan, S. Tsang and W. Siu, "Machine Learning-Based Fast Intra Mode Decision for HEVC Screen Content Coding via Decision Trees," In IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 5, pp. 1481-1496, May 2020. doi: 10.1109/TCSVT.2019.2903547.
[4] Q. Yin, J. Cheng, F. Zhang, Y. Zhou, L. Shao and W. Hong, "Interpretable POLSAR Image Classification Based on Adaptive-Dimension Feature Space Decision Tree," In IEEE Access, vol. 8, pp. 173826-173837, 2020. doi: 10.1109/ACCESS.2020.3023134.

[5] C. Cho, W. Chung and S. Kuo, "Using Tree-Based Approaches to Analyze Dependability and Security on I&C Systems in Safety-Critical Systems," In IEEE Systems Journal, vol. 12, no. 2, pp. 1118-1128, June 2018. doi: 10.1109/JSYST.2016.2635681.

[6] Mr. Ankit Kumar, Dr. Dinesh Goyal, Mr. Pankaj Dadheech, (2018), “A Novel Framework for Performance Optimization of Routing Protocol in VANET Network”, Journal of Advanced Research in Dynamical & Control Systems, Vol. 10, 02-Special Issue, 2018, pp-2109-2109, ISSN: 1943-023X.

[7] Mr. Pankaj Dadheech, Dr. Dinesh Goyal, Dr. Sumit Srivastava, Mr. Ankit Kumar, (2018), “A Scalable Data Processing Using Hadoop & MapReduce for Big Data”, Journal of Advanced Research in Dynamical & Control Systems, Vol. 10, 02-Special Issue, 2018, pp-2099-2109, ISSN: 1943-023X.

[8] Vengatesan, K. et al. "Analysis of Mirai Botnet Malware Issues and Its Prediction Methods in Internet of Things". Lecture Notes on Data Engineering and Communications Technologies 31. (2020): 120-126.

[9] Vimal, V. et al. "Artificial intelligence-based novel scheme for location area planning in cellular networks". Computational Intelligence. (2020).

[10] Kumar, A. et al. "Black hole attack detection in vehicular ad-hoc network using secure AODV routing algorithm". Microprocessors and Microsystems. (2020).

[11] Kumar, A. et al. "Comparative Analysis of Data Mining Techniques to Predict Heart Disease for Diabetic Patients". Communications in Computer and Information Science 1244 CCIS. (2020): 507-518.

[12] Vengatesan, K. et al. "Credit card fraud detection using data analytic techniques". Advances in Mathematics: Scientific Journal 9. 3(2020): 1185-1196.

[13] Preetha, J. et al. "Data mining technique based critical disease prediction in medical field". Advances in Parallel Computing 37. (2020): 104-108.

[14] Kumar, V.D.A. et al. "Exploration of an innovative geometric parameter based on performance enhancement for foot print recognition". Journal of Intelligent and Fuzzy Systems 38. 2(2020): 2181-2196.

[15] Pankaj Dadheech, Dinesh Goyal, Sumit Srivastava & C. M. Choudhary, (2018), “An Efficient Approach for Big Data Processing Using Spatial Boolean Queries”, Journal of Statistics and Management Systems (JSMS), 21:4, 583-591.

[16] A. Kumar and M. Sinha (2014), "Overview on vehicular ad hoc network and its security issues," International Conference on Computing for Sustainable Global Development (INDIACom), pp. 792-797. doi: 10.1109/IndiaCom.2014.6828071.

[17] Pankaj Dadheech, Ankit Kumar, Chotheomal Choudhary, Mahender Kumar Beniwal, Sanwta Ram Dogiwal & Basant Agarwal (2019), “An Enhanced 4-Way Technique Using Cookies for Robust Authentication Process in Wireless Network”, Journal of Statistics and Management Systems, 22:4, 773-782, DOI: 10.1080/09720510.2019.1609557.

[18] Vengatesan, K. et al. "Intrusion detection framework using efficient spectral clustering technique". Advances in Parallel Computing 37. (2020): 98-103.
[19] Vikrant, S. et al. "M-learning for promoting advancements in agriculture: An innovative educational model for ethiopian farmers". Advances in Mathematics: Scientific Journal 9. 7(2020): 5057-5064.

[20] Vengatesan, K. et al. "Secure Data Transmission Through Steganography with Blowfish Algorithm". Lecture Notes on Data Engineering and Communications Technologies 35. (2020): 568-575.

[21] Lone, T.A. et al. "Securing communication by attribute-based authentication in HetNet used for medical applications". Eurasip Journal on Wireless Communications and Networking 2020. 1(2020).

[22] Vengatesan, K. et al. "Simple Task Implementation of Swarm Robotics in Underwater". Lecture Notes on Data Engineering and Communications Technologies 35. (2020): 1138-1145.

[23] Mallikalava, V. et al. "Theft vehicle detection using image processing integrated digital signature-based ECU". Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020. (2020): 913-918.

[24] Kesavan, S. et al. "An investigation on adaptive HTTP media streaming Quality-of-Experience (QoE) and agility using cloud media services". International Journal of Computers and Applications. (2019).

[25] Ankit Kumar, Pankaj Dadheech, Vijander Singh, Linesh Raja & Ramesh C. Poonia (2019), “An Enhanced Quantum Key Distribution Protocol for Security Authentication”, Journal of Discrete Mathematical Sciences and Cryptography, 22:4, 499-507, DOI: 10.1080/09720529.2019.1637154.

[26] Ankit Kumar, Pankaj Dadheech, Vijander Singh, Ramesh C. Poonia & Linesh Raja (2019), “An Improved Quantum Key Distribution Protocol for Verification”, Journal of Discrete Mathematical Sciences and Cryptography, 22:4, 491-498, DOI: 10.1080/09720529.2019.1637153.

[27] Ankit Kumar and Madhavi Sinha (2019), “Design and analysis of an improved AODV protocol for black hole and flooding attack in vehicular ad-hoc network (VANET)”, Journal of Discrete Mathematical Sciences and Cryptography, 22:4, 453-463, DOI: 10.1080/09720529.2019.1637151.

[28] Ankit Kumar, Linesh Raja, Pankaj Dadheech, Manish Bhardwaj (2020), “A Hybrid Cluster Technique for Improving the Efficiency of Colour Image Segmentation”, World Review of Entrepreneurship, Management and Sustainable Development, Nov. 2020, Vol. 16, Issue 6, pp. 665-679, Print ISSN: 1746-0573 Online ISSN: 1746-0581, https://doi.org/10.1504/WREMSD.2020.111405.

[29] Dadheech P., Goyal D., Srivastava S., Kumar A., Bhhardwaj M. (2021) Performance Improvement of Heterogeneous Cluster of Big Data Using Query Optimization and MapReduce. In: Goyal D., Bălaş V.E., Mukherjee A., Hugo C. de Albuquerque V., Gupta A.K. (eds) Information Management and Machine Intelligence. ICIMMI 2019. Algorithms for Intelligent Systems. Springer, Singapore, pp 85-100, First Online: 17 September 2020, Print ISBN: 978-981-15-4935-9, Online ISBN: 978-981-15-4936-6, https://doi.org/10.1007/978-981-15-4936-6_9.

[30] A. Kumar, P. Dadheech and U. Chaudhary, (2020), "Energy Conservation in WSN: A Review of Current Techniques," 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), Jaipur, India, 2020, pp. 1-8, ISSN: 978-1-7281-1683-9, doi: 10.1109/ICETCE48199.2020.9091736.

[31] Kumar A., Dadheech P., Beniwal M.K., Agarwal B., Patidar P.K. (2020), A Fuzzy Logic-Based Control System for Detection and Mitigation of Blackhole Attack in Vehicular Ad Hoc Network. In: Chaudhary A., Choudhary C., Gupta M., Lal C., Badal T. (eds) Microservices in Big Data Analytics.
Springer, Singapore, First Online: 27 November, 2019, pp 163-178, Print ISBN: 978-981-15-0127-2, Online ISBN: 978-981-15-0128-9, DOI https://doi.org/10.1007/978-981-15-0128-9_15.

[32] Kumar, A. et al. "Teaching literacy through animation & multimedia". *International Journal of Innovative Technology and Exploring Engineering* 8. 5(2019): 73-76.