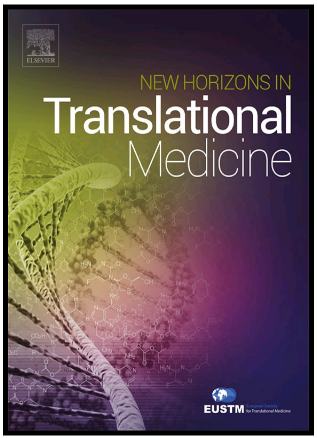

www.elsevier.com/locate/nhtm

Improving disease diagnosis by a new hybrid model

Bikash Kanti Sarkar

PII: S2307-5023(17)30023-1
DOI: http://dx.doi.org/10.1016/j.nhtm.2017.07.001
Reference: NHTM49

To appear in: *New Horizons in Translational Medicine*

Received date: 20 June 2017
Revised date: 5 July 2017
Accepted date: 6 July 2017

Cite this article as: Bikash Kanti Sarkar, Improving disease diagnosis by a nev hybrid model, *New Horizons in Translational Medicine* http://dx.doi.org/10.1016/j.nhtm.2017.07.001

This is a PDF file of an unedited manuscript that has been accepted fo publication. As a service to our customers we are providing this early version o the manuscript. The manuscript will undergo copyediting, typesetting, an review of the resulting galley proof before it is published in its final citable forn Please note that during the production process errors may be discovered whic could affect the content, and all legal disclaimers that apply to the journal pertain

Improving disease diagnosis by a new hybrid model

Dr. Bikash Kanti Sarkar[*]

Department of Computer Science and Engineering Birla Institute of Technology, Mesra, Ranchi, India

bk_sarkarbit@hotmail.com

[*]Corresponding author. Dr. Bikash Kanti Sarkar

ABSTRACT

Knowledge extraction is an important part of e-Health system. However, datasets in health domain are highly *imbalanced*, *voluminous*, *conflicting* and *complex* in nature, and these can lead to erroneous diagnosis of diseases. So, designing accurate and robust clinical diagnosis models for such datasets is a challenging task in data mining. In literature, numerous standard intelligent models have been proposed for this purpose but they usually suffer from several drawbacks like lack of *understandability*, incapability of operating *rare cases*, inefficiency in making *quick* and *correct* decision, etc. In fact, specific health application using standard intelligent methods may not satisfy multiple criteria. However, recent research indicates that hybrid intelligent methods (integrating several standard ones, can achieve better performance for health applications. Addressing the limitations of the existing approaches, the present research introduces a new hybrid predictive model (integrating C4.5 and PRISM learners) for diagnosing effectively the diseases (instead of any specific disease) in comprehensible way by the practitioners with better prediction results in comparison to the traditional approaches. The empirical results (in terms of *accuracy*, *sensitivity* and *false positive rate*) obtained over fourteen benchmark datasets demonstrate that the model outperforms the base learners in almost all cases. The performance of the model also claims that it can be good alternative to the specialized learners (each designed for specific disease) published in the literature. After all, the presented intelligent system is effective in undertaking medical data classification task.

Keywords: classifier; C4.5; PRISM; hybrid-model; disease-prediction; accurate

Focal points

Disease modelling

The present research focuses on designing hybrid model to better understand clinical conditions and to detect diseases more accurately for undertaking treatment properly.

Important terminologies

- *Classification dataset*: A classification dataset(D) is described by a number of non-target attributes (say a_1, a_2, .. a_n) and a target (or class) attribute (say C). Each instance (*i.e*., example or case) in D takes specific values of the attributes. The values may be string (*i.e*., nominal), e.g., values of temperature are -low, medium and high. Value may be continuous that can occupy any value over a continuous range, e.g., 2.0045. It may have long-range value, e.g., 10^5, 2^{13} or more. However, any data-discretizer may be applied on such a dataset to get only integer attribute values corresponding to their non-discretized values by performing suitable mapping scheme. *MIL* (minimum information loss)-discretizer is such a data-discretized which is adopted in the present research. For more details regarding dataset and discretized attribute values, one may refer Appendix-A.
- *Imbalanced dataset*: A dataset in which the number(s) of instances of some class(es) is/are very less in comparison to other classes, is termed as imbalanced dataset. The instances of a class with very less in number are known as *rare cases*.
- *Voluminous dataset*: Dataset that consists of large number of instances or large number of attributes (*i.e*., high dimensionality) or both, is usually called as voluminous dataset.
- *Conflicting data set*: Dataset that possesses instances with different class values for identical non-target attribute values, is termed as *conflicting dataset*. In particular, such instances are, indeed, *inconsistent* instances that cause uncertainty in making decision by the system.
- *Incomplete dataset*: A dataset in which many information (*i.e*., values of attributes) are either *missing* or *incomplete*, is known as incomplete dataset.
- *Complex* (or *uncertain*) *dataset*: If drawing any conclusion is very difficult for a dataset, then the set is called as complex or uncertain dataset. An uncertain dataset is often called as vagueness dataset.
- *Non-normal data set*: Data set with one or more above mentioned issues is called as non-normal dataset.
- *Parametric and non-parametric learning model*: A learning model that summarizes data by a set of fixed size parameters (*i.e*., co-efficients) and a function (*e.g*., $y=x_0+x_1a_1+x_2a_2+..$, where x_i represents the *i*-th parameter and a_i denotes the *i*-th attribute of the dataset) is called as parametric learning model. In particular, the values of the parameters are learned from training data. However, no such function is used in non-parametric learning model.

- *Entropy-based classifier*: An entropy-based approach uses an entropy function (e,g., information gain function) based on its instantaneous output probabilities for each example and combines the output probabilities of the different classifiers before making the final decision. It is detailed in Section-2.1.

I. Introduction

Designing automated intelligent model is a growing need from data, as the amount of data stored in databases increases in rapid manner and the number of human data analysts grows at a much smaller rate than the amount of stored data. Machine learning[1], a field of data mining [2] is an excellent process for designing such models. The process has capability to discover insightful, interesting and novel patterns which are descriptive, understandable and predicative from large amount of data. In particular, it is an important part of knowledge discovery from databases[3-4]. A number of machine learning and knowledge discovery techniques have been developed for inducing decision rules and are being used in various disciplines. Some of the widely used techniques are decision trees(DT)[5], neural networks [6], rough sets [7] and decision tables[8], PRISM[9], Repeated Incremental Pruning to Produce Error Reduction(RIPPER)[10], naïve Bayes[11], etc. Truly speaking, each of these has some merits and demerits. Precisely, no model is well-suited for all data sets. However, the primary advantages of these techniques are that they are usually *data driven* (based on past data), *non-parametric* and *less restrictive* to a priori hypothesis. Importantly, decision tree learner (among the commonly used learners) is considered suitable for both *non-normal* and *non-homogeneous* datasets and it shows an average prediction performance for datasets of almost all domains. Also, the predictive power of PRISM can be seen as acceptable when contrasted to other classic data mining approaches such as search methods, decision trees, neural networks, associative classification and many others, as it explores more generalised rules [9]. In particular, the algorithm optimizes the *purity* of a rule, that is, it maximizes the percentage of positive examples among all covered example[12].

At the present date, the use of data mining techniques is gradually increasing in medical diagnosis because of their potential capabilities. In practice, treatments are made by the physicians where a physician typically accumulates his/her knowledge based on the patient's symptoms and applies the knowledge/memory (prognostic relevance of symptoms) towards diagnosing diseases. It is well-accepted that diagnostic accuracy of patients is here highly dependent on physician's experience, that is, it varies from expert to expert. Also, manual diagnostic is a time consuming job. So, designing computerized system from past diagnosis data may be the essential solution in this purpose. Now-a-days, the use of machine learning techniques[1] is gradually increasing in medical diagnosis because of their potential capabilities. In particular, any *accurate*, *precise* and *reliable* predictive model may significantly assist the medical practitioners to improve *diagnosis* and *treatment* processes of individual's diseases in faster way. At the same time, it reduces the cost associated with patient treatment. However, medical data are usually unstructured and they are by nature *imbalanced, conflict*, *incomplete* and *vagueness*. So, designing accuracy-based reliable automated diagnostic model is a challenging task to the researchers. A wide range of computerized *clinical decision support systems* (CDSS) have been modelled over the years to assist physicians in making decisions. For a review, one may see[13-22, 64-72, 75]. Undoubtedly, the systems are used for diagnosis, prediction, classification and risk forecasting of various diseases on the basis of *electronic medical records*(EMR) of patients. A schematic of CDSS is depicted in Fifure-1.1.

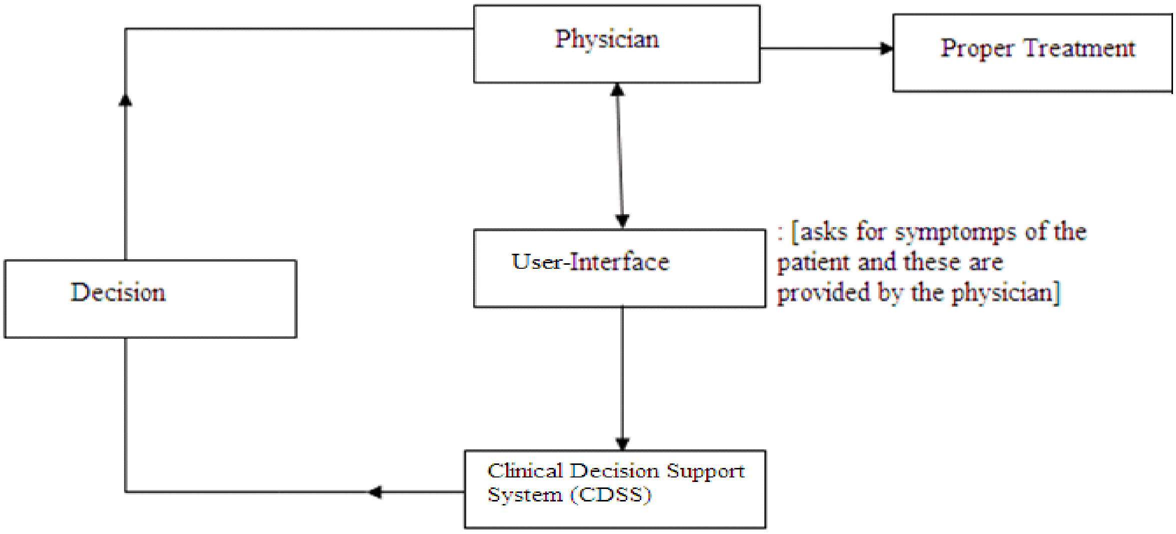
Physician
Proper Treatment
User-Interface
: [asks for symptomps of the patient and these are provided by the physician]
Decision
Clinical Decision Support System (CDSS)

Fig-1.1: A schematic of CDSS

Given below are some desirable characteristics of any good CDSS.

i) Learned knowledge in any CDSS is preferred in 'IF-THEN' decision rule format so that practitioner can easily *interpret* the rules for predicting diseases.

ii) Rules in 'IF-THEN' structure must be *accurate* (*i.e*., high prediction rate).

iii) Size of the rule set should be *concise*, and less number of informative pre-conditions in each rule is highly desired.

iv) Rules of all classes (even for rare cases) must be present in the rule set for predicting unseen data accurately.

v) There must not have any *conflict* rules in the set.

In context of CDSS, the PRISM algorithm (although, it is an old classifier) may be successfully applied in *medical diagnosis* and *prediction*, since it has ability to generate *pure* decision rules for all class-labelled instances. Accordingly, it does not ignore the *rare case* instances which are frequently observed in medical datasets.

Although, several clinical models have been developed but each of these is unfortunately suffering from one or more *deficiencies* as listed below.

- *Disease specificity of model*: No generalized model is designed for showing better or on an average disease prediction *accuracy* over all medical datasets. In other words, each of the existing systems is well-suited for a specific dataset. Examples include the studies presented in [13, 15, 17, 21-27, 65-72, 76].

- *Black-box model*: Most of the present diagnostic methods are black-box models, that is, they have no explanation power in terms of understandability of decision rules [15, 17, 28, 29, 66-72, 75]. As a result, the models are unable to provide the reasons underlying diagnosis to physicians; therefore, further insights are needed for those algorithms.
- *Incapability of operating high-dimensional and inconsistent data*: In general, each of the existing systems has deficiency to handle high dimensional, inconsistent and vagueness (uncertain) clinical data.
- *Low power of accurate rule generation*: Most of the existing approaches suffer from generating *accurate rules* which are highly desired in CDSS. As a result, such systems result uncertainty and imprecision in decision making[77].
- The models are usually dependent on the hypothesis of statistical techniques.

Obviously, resolving all the issues together and constructing a generalized accurate disease predictive model (*i.e*., model with highly accurate rules) is a key challenge in medical applications. Recall that any individual competent learner in general performs well on specific diseases. That is why, combination of the learners may be effective to design a generalized disease predictive model and the research has gained much importance in this respect.

It may be noted here that Sarkar *et al*.[30, 31] proposed some ensemble approaches (combining decision tree learner with genetic algorithm) to improve classification performance over datasets irrespective to *domain*, *size* and *class imbalance* issues. More specifically, they have used classification problems of both medical and non-medical domains in their experiments. However, learning time of these approaches increases unexpectedly due to the application of genetic algorithm(GA). Further, the objective functions (*i.e*., fitness functions in respect to GA) are proposed with the point in mind that the datasets may belong to any domain. Hence, these approaches may not be treated as specialized for medical data sets.

In recent years, rough set rule induction algorithms are being actively utilized for the extraction of decision rules from various medical datasets[32-37] because rough set theory has capability to handle the issues like uncertainty, missing values, conflict instances present in database. However, its main drawbacks are:

(i) The theory relatively generates lager number of rules and (*ii*) the learning time is *exponential*.

As a result, this approach may not be well-suited for larger input data files.

Also, in machine learning, neural networks have significant advantages for medical decision support applications[75]. However, one key limitation of this approach is the lack of ability to explain the prediction[78].

Contribution of the present study

To resolve the identified issues of the existing systems, the author attempt to design a generalized hybrid CDSS (combining some well-suited individual learners) and finally proposes a model by integrating decision tree based learner (C4.5) and PRISM learner. However, it is well-accepted that choosing the learners that perform best for a particular dataset is a challenging task in data mining. Anyway, some reasons behind favouring C4.5 and PRISM are listed below.

- For *asymmetrical distribution* of the medical datasets, decision tree and PRISM methods are the suitable tools, as they are not confined to non-parametric datasets only.
- Both the learners are easy to implement and the training speed of each of the two approaches is not high.
- Each of C4.5 and PRISM learners is an example of *white-box* model, and has capability to express *knowledge* in terms of *meaningful* decision rules (in 'IF-THEN' form). As a result, they are closed compatible to be combined each other. In fact, rules in 'IF-THEN' format are simple and easy to understand, and these can empower decision makers (particularly in domain applications) that necessitate interpretations. In particular, such rules are highly desired in medical diagnosis, as they can easily be applied for predicting unseen objects.
- Importantly, the PRISM algorithm focuses only on the rule’s accuracy to find more *accurate* rule that reduces chance of *false negative rate* and it is highly expected in medical expert system. Actually, a high negative rate of cases increases worry and stress in patients, and increases the risk of patients [79]. Likewise, the C4.5 (an entropy-based classifier) also shows good performance on unseen data. In other words, it also has strength to generate accurate rules.
- Again, C4.5 has high ability to handle uncertainty (vagueness) in data. That is why, this classifier shows better or *on an average* performance over datasets of almost all domains.
- Lastly, the PRISM algorithm can tackle rare case issue to a great extent, as it separately considers instances of each class, including the rare case instances too.

The above mentioned promising strengths of both the learners (C4.5 and PRISM) theoretically assure that they are well-suited for modelling an ensemble learner to operate medical datasets. As evidence, in 2014, Stahl and Bramer proposed a PRISM-based ensemble model and showed that the model was able to generate results comparable with classic PRISM algorithm[61]. Also, PART[62] is an example of integration of DT- based learner and PRISM learner for handling medical datasets. In particular, the integrated system uses DT to filter out the rules generated by PRISM. However, in spite of having strengths of PRISM and C4.5 methods, they have drawbacks such as:

i) PRISM can’t handle *noisy* datasets that contain incomplete attributes and missing values.

ii) There is no clear mechanism on how to resolve *conflicting rules* in PRISM.

iii) No clear rule pruning methodology is present in the original PRISM, and this may lead to generation of large numbers of rules[38].

iv) On the other hand, the DT algorithm C4.5 usually ignores the rules of *rare case* instances in the generated rule set[73], and it is true that most of the medical datasets are imbalanced in nature.

Now, in order to resolve the limitations of PRISM and C4.5 learners, the present study aims to adopt the following tools and strategies:

- The MIL-discretizer [52] to tackle noisy data
- An appropriate data sampling scheme (*i.e*., a data level method) to manage rare case issue
- An innovative idea introduced in the hybrid approach to filter out high quality decision rules discarding conflicting rules- the idea mainly focuses on extracting small number of more accurate but conflictless rules, since a smaller size of informative rule set must work fine for applications such as medical diagnoses and the general practitioners can enjoy a concise set of decision rules for daily diagnoses of their patients.

The paper is organized as follows. Section-2 presents the survey on the necessary methods which include C4.5, PRISM learners, the Interface s/w (for representing rules) and the classifier's performance measuring metrics. Also, the description of the selected medical datasets (drawn from UCI Machine Learning Repository) and their preprocessing are covered in this section too. The proposed hybrid model (C4.5+PRISM) is detailed in Section-3. Section-4 presents the experiments conducted on the chosen datasets. Also, this section deals with discussion of the results obtained by the suggested model, its base learners and some state-of-the art other models for the chosen datasets. Concluding remarks as well as future scopes of the present research are summarized in Section-5, whereas a short executive summary on the work is given in Section-6.

2. Necessary Methods and Materials

In this section, a brief description for each of the base learners (used in the present study) is first presented. The Interface[60] s/w for tabular like rule-representation, the performance evaluation metrics of the classifiers adopted in this research, the selected datasets (*i.e*., materials) and its pre-processing are also explained in this section.

2.1 C4.5: A decision tree-based classifier

C4.5[5] is a well-known rule induction algorithm to solve classification task. The algorithm was proposed by Professor Ross Quinlan, University of Sydney, in 1992. It is, indeed, the extended version of ID3 algorithm[39]. In comparison to ID3, C4.5 includes extra features like handling *missing* values, managing *continuous* attributes, *pruning* trees and others.

Anyway, the primary goal of this learner is to minimize the *doubt* (*i.e*., impurity) in a dataset (representing information). In this purpose, the learner starts by choosing the best *informative attribute* and splits the dataset. The process is applied recursively on each partitioned of the dataset, and continues till no data is left to split or no new attribute is left to process (which one appears earlier). The output given by the process is a *decision tree*.

Selection of best attribute at each stage

In order to select the *best* relevant attribute for each node of the tree, an entropy function (as defined below) is considered.

$$\mathrm{Entropy}(\mathrm{S}) = H(S) = -\sum_{i=1}^{c} \mathrm{p_i} \log_2 \mathrm{p_i}$$

, where S represents the number of currently considered learning examples and p_i is the non-zero probability of the examples (say $\mathrm{S_i}$ in S) belonging to class i , out of c classes.

It may be noted that C4.5 consists of mainly three phases: *tree construction* (C4.5u), *tree pruning* (C4.5p) and *rule induction* (C4.5r). Obviously, each phase operates at distinct level.

Finally, the C4.5r algorithm results the pruned tree obtained from C4.5p, and each path from the *root* to a *leaf* of the tree yields a prospective decision rule representing in simple: If (*conditions*)–Then (*decision class*) like form. For better *understanding* the decision tree, one may refer *Appendix- A.*

Decision tree and classification task

Decision tree classifiers have been widely used to represent predictive models, due to its comprehensible nature that resembles the human reasoning. In this respect, one may refer some standard studies[30,31,40-44]. Notably, DT-based classifiers have gained much importance in prediction of diseases. In 2003, Stasis *et al*. proposed a decision support system for *heart* sound diagnosis using C4.5 learner[45]. D.E. Brown introduced the data mining as medical informatics by applying a classification tree on Pima Indians dataset [46]. In 2003, Azar and EI-Metwally presented a decision support tool for investigating the *breast cancer* using three types of decision trees[28]. For more, one may refer the recent study[64].

The short review presented above says that DT-based classifier is applied not only for artificial domain datasets but also for medical datasets at the present date.

2.2 PRISM: A sequential covering rule inducing algorithm

Sequential covering is also an 'IF-THEN' rule mining approach. Here, the 'IF-THEN' rules are extracted directly from the training data without constructing a decision tree. In particular, the rules are learned sequentially, i.e., one rule at a time. Each time a rule is learned, the tuples covered by the rules are removed, and the process repeats on the remaining tuples. More specifically, rules are learned for one class at a time. Actually, while learning a rule for a class, say '*c'*, it would like to cover the training tuples of class '*c'* but none of the tuples from other classes. In this way, the learned rules must have *high accuracy* but the rules need not necessarily be of high coverage. The reason is that, there may have multiple rules for a class, and different rules may cover different tuples for the same class. The process continues until the *terminating condition* such as when there are no more training tuples to learn, reaches. A high-level description of the approach is given below (in the box).

```
PRSIM rule induction algorithm
for each class 'c' do
  begin
Initialize examples (E) to the training set (T).
temp-dataset = T
  while T contains examples of class $c_i$ do
    begin
    • Create a rule r with an empty left-hand side (LHS) that predicts class c, i.e., r:= { } → c
    If r is not perfect, then do the followings:
          begin
    for each attribute: A not mentioned in r , and each value v do
    • Consider the following to add the condition A = v to the LHS of r.
    • Select A and v to maximize the accuracy: $a = p/t$ in the current temp-dataset, where
      p=number of instances in temp-dataset belonging to the class ($c_i$) by A=v and t= number of
      instances in temp-dataset covered by A=v irrespective to any class.
                    (break ties by choosing the condition with the largest p)
    • Add A=v to r
                endfor
    • Remove the instances (say t instances) covered by r from T, i.e., T=T- t and update the
      temp-dataset consisting of the present content of T (removing the earlier contents of temp-
      dataset).
              endwhile
           endfor
    Note: For better understanding the approach, one may refer Appendix-B.
```

One may note that there exist various versions of PRISM algorithm, *e.g.*, RIPPER [10] that reduces the size of rules using pruning. However, RIPPER may lead to *loss of knowledge*, as it employs excessive pruning to reduce the size of the classifier. Another version namely N-PRISM algorithm[47] is proposed to resolve the problem of noisy data, whereas J-Pruning[48] employs pre-pruning strategy. In 2008, Stahl and Barmer introduced Parallel PRISM(P-PRISM)[49] method to overcome PRISM's excessive computational process of testing the entire population of data attribute inside the training dataset.

It may be noted that very little academic research has been found in medical domain using PRISM and its successors. For a review in this respect, one may refer the study[63].

2.3 The IF-THEN rules and the Interface

This section contains a short description of the rule-representation scheme to deal the suggested hybrid model more conveniently. The representation is, indeed, internally used by the model to perform the specific tasks such as *computing accuracy*, *resolving conflict rules*, *finding accurate rules*, etc., *i.e*., it may be hidden to the users (practitioners). In particular, the scheme is applied over the rules generated by the pure C4.5 and PRISM classifiers.

In general, the knowledge induced by most of the supervised learning algorithms is represented by decision rules of the form: *IF* (*conditions*) *THEN* (*decision class*), where conditions (also termed as *pre-conditions*) in each rule are *conjunctions* of elementary tests on values of attributes, and *decision* part indicates the assignment of an object to a given *decision class*. In fact, each 'IF-THEN' rule can be viewed as: *antecedent→consequent*, where antecedent part consists of *conjuncts* (*i.e*., pre-conditions or in short conditions) and consequent is the *decision* (*i.e.*, action). Certainly, the left hand side (LHS) of a rule (*i.e*., antecedent) does not necessarily contain all the non-target attributes. It may be noted that such 'IF-THEN' rules are one of the most popular types of knowledge representations used in practice. The main reason behind the wide application of such rules is the expressive and easy human-readable representation[2].

Recall that the rules produced by both C4.5 and PRISM methods are close to 'IF –THEN' form. Their formats are shown respectively in Appendix-*A* and Appendix-*B* taking a tiny dataset of the *golf-playing* problem. Unfortunately, 'IF- THEN' rules are not easy to interpret by the system while finding the necessary tasks such as *computing accuracy*, *resolving conflict rules*, *finding accurate rules*, etc. So, to overcome the interpretability issue with respect to system, *tabular* like representation of the rules is preferred here. It is interesting to notice that the *sequence* of attribute-names placed at the columns of *tabular*-like representation must follow the sequence of attributes of original the dataset. In particular, the *last column* always represents the target attribute. However, if a data set contains target attribute at its first column, then necessary transformation is made before passing it to the learner.

The *Interface*[60] adopted in the present study provides tabular representation of rules (as shown in Appendix-C), removing 'IF’ and ‘THEN' parts (clauses) from those. More specifically, the values of the attributes are listed below the names of the respective attributes (representing the columns of tabular structure). For each rule, the *interface* places ‘*’ (*don’t care*) symbols for non-target attributes whose *pre-conditions* are absent in that rule. Thus, the attribute corresponding to the *position* of symbol: ‘*’ in a rule simply implies that the attribute has no importance in that rule itself. Truly speaking, all the non-target attributes irrespective to their presence or *absence* in rule(s) are herein strictly considered in rule(s) with a view to *simpler access*.

2.4 Classifier’s performance measuring metrics

Performance of any classification algorithm needs to be tested with some metrics in order to assess the result and hence the quality of the algorithm. In the present research, to evaluate the effectiveness of the suggested model over the medical datasets, performance metrics such as *accuracy*, *sensitivity* (true positive rate) and *false positive rate* are computed. These are defined below.

i) ***Accuracy***: For measuring *accuracy performance* of a classifier, the well-accepted formula (as given in *equ*.(2.1)) is adopted here.

$$\text{Accuracy (acc.)} = \frac{\text{m}}{\text{n}} \text{ x } 100 \qquad \text{- - - - - - - - - - -} \ (2.1),$$

where *m* denotes the number of *correctly* classified test examples *(i.e*., unseen data) and *n* is the *total* number of test examples. This is, indeed, the average accuracy (%) measure of the learner, and it also can be computed as:

$$\frac{TP+TN}{P+N}\times 100$$

Here, P and N denote respectively the numbers of *positive* and the *negative* examples present in the test set, whereas TP and TN refer respectively the numbers of predicted *true positive* and *true negative* examples. Literally, the terms -TP, TN, FP, FN have the following meanings:

- True positive(TP): case is positive and predicted as positive.
- True negative (TN): case is negative and predicted as negative.
- False positive (FP): case is negative but predicted as positive.
- False negative (FN): case is positive but predicted as negative.

Now, based upon equ.(2.1), the *error-rate* (*e*%) of any classifier can be computed as:

$$\text{e} = (100 - acc.).$$

Generally speaking, accuracy measure reports the *overall exactness* statistic of a classifier. Hence, error-rate (*e*) gives an overall estimation of errors. Further, *precision* (another metric) defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \qquad \text{- - - - - - - - - - -} \ (2.2)$$

emphasises on the exactness measure of instances of a particular class. This class is known as *positive class*, and it is, indeed, a class of user interest. Conceptually, these two metrics (*accuracy* and *precision*) are closely related to each other, since both of them emphasize on correct classification of cases. That is why, interest is not separately shown here on *precision*, rather attention is paid on another two useful metrics, viz., *true positive rate* (TPR *i.e*., *sensitivity*) and *false positive rate* (FPR) assuming the *presence of disease* as the true positive case (*e*.g., Sick people correctly identified as sick) and the cases belonging to the rest categories together are treated as negative cases. Now, the questions are:

- Why the rest examples are put into negative category?
- Why the present study focuses on measuring TPR and FPR?

The answer to the first question is that most of the chosen problems are multi-category problem (*i.e*., not a binary class problem). For this reason, one class is treated as the positive class and the rest are grouped into negative class. Next, the second query is defended as follows:

The datasets belong to medical domain, and identifying the disease-affected person correctly must be of much importance in this aspect. Simultaneously, any instance (*i.e.*, case) under negative category should not be treated as positive to increase *unnecessary* mental worry among the persons. With these points in mind, attention is paid on computing the useful metrics- *true positive rate* and *false positive rate* of the model. These two are defined below.

ii) *True positive rate* (Sensitivity) **:** This measure finds the proportion of positive cases that are correctly identified, and so it is expressed as:

$$TPR = \frac{TP}{P} ....(2.3)$$

iii) *False positive rate*: It is the proportion of negative cases that are incorrectly identified positive. It is formulated as:

$$FPR = \frac{FP}{N} ....(2.4)$$

2.5. Discussion on the selected datasets and their pre-processing

Datasets

Recall that the datasets in the present study are collected from UCI (University of California at Irvine), a *machine learning repository*[50]. They all belong to real world *medical* domain. Their features are *summarized* in Table 2.1. The problem names are arranged in alphabetical order in the table. The first four columns in the table say respectively *problem name* (*i.e.*, name of the dataset), *non-target attributes*, *number of classes* and *number of instances*. On the other hand, the last three columns show the *class imbalance* behaviour of the datasets. More specifically, the last two columns report the percentage of *minority* and *majority* class instances of each dataset. Note that the imbalance ratio of each dataset (placed at the 3rd last column in the table) is computed by the formula introduced by Tanwani and Farooq [51]. The formula is once again given here in equ.(2.5).

$\text{Imbalance ratio}(I_R) = \frac{N_c - 1}{N_c}\sum_{i=1}^{N_c}\frac{I_i}{I_n - I_i}...........(2.5)$, where I_i denotes the number of instances of i^{th} class, whereas I_n represents the total number of instances. On the other hand, N_c stands for the number of classes present in the dataset. The value of I_R (imbalance ratio) lies in the range: $1 \le I_R < \infty$, where $I_R = 1$ implies that the dataset is completely *balanced* having equal instances of all classes.

Table-2.1: Summary of the selected UCI Datasets (original)

Problem name	Number of non-target attributes	Missing Value presence	Number of classes	Number of examples	Imbalance ratio(I_R)	Minority class % with minimum instances	Majority class % with majority instances
Breast Cancer Wisconsin	10	Yes	2	699	1.2133	34.47	65.52
Dermatology	34	Yes	6	366	1.0526	5.4	30.60
Pima Indian Diabetes	8	Yes	2	768	1.2008	34.89	65.11
Ecoli	8	No	8	336	1.2495	0.5	42.55
Heart (Hungarian)	13	Yes	5	294	1.7389	5.1	63.94
Heart (Swiss)	13	Yes	5	123	1.1409	4.06	39.02
Heart (Cleveland)	13	Yes	5	303	1.3693	4.29	54.12
Hepatitis	19	Yes	2	155	2.051	20.64	79.35
Liver Disorder	6	No	2	345	1.0522	42.02	57.98
Lung Cancer	56	Yes	3	32	1.02	28.12	40.62
Lymphography	18	No	4	148	1.46	1.35	54.72
New-thyroid	5	Yes	3	215	1.7673	13.95	69.76
Primary Tumor	17	Yes (more)	22	339	1.3334	0.5	24.77
Sick	29	Yes	2	3772	7.6971	6.12	93.87

Simply looking into Table-2.1, it is clear that all the selected datasets except *Ecoli* and *Liver-disorder* have missing attribute values. One noticing point is that the *Primary Tumor* database has more number of missing values. Also, the datasets viz. *Heart*(Hung./Swiss/Cleveland), *Hepatitis*, *New-thyroid*, *Sick* and *Primary Tumor* are imbalanced and accepted as uncertain in nature, whereas *Sick* database is highly imbalanced among these. Further, *Lung Cancer* and *Sick* are high-dimensional datasets. In particular, *Sick* dataset is comparatively voluminous.

Data pre-processing

Recall that each dataset in this study is a medical classification problem (P). In reality, attributes of a dataset may contain mix-up of *string*, *continuous*, *long- range* or *missing* values. So, it is essential to pre-process each dataset before passing it to any learner. In this purpose, MIL data discretizer[52] is employed here. The discretizer emphasizes on preventing loss of information that may occur due to discrtization of data. In addition, it has capability to resolve *inconsistency* issue of instances present in the dataset as well as its occurrence during discretization process. In fact, the discretizer performs separately one extra step after discretiziation to tackle the issue. More specifically, it verifies each instance (I) of a dataset with rest of the instances and keeps one with *majority* class instance among the conflict class instances (if found) for I in the dataset. Simultaneously, the rest conflicting instances for the instance *I* are discarded from the dataset.

At this point, it is necessary to mention that many classifiers such as [9, 55, 56] cannot handle *continuous* attributes, whereas each of them can operate on discretized attributes. Furthermore, even if an algorithm can handle continuous attributes, its performance can be significantly improved by replacing continuous attributes with its discretized values[57-58]. The other *advantages* in operating discretized attributes are the need of *less memory space* and *less processing time* in comparison to their non-discretized form. Lastly, small number of rules are produced, while processing discretized attributes[7-9].

3. The proposed hybrid model

The basic requirements to construct any expert system are: (*i*) the training set (*i.e.*, past experience), (*ii*) learner that results knowledge from training set and (*iii*) finally test set to assess the performance of the system. Technically, both the training and the test sets are constructed from original dataset. It is important to note here that training set plays a vital role in designing expert system. In the purpose of constructing better training set, appropriate data-partitioning is the essential solution. In Section-3.1., a new partitioning scheme followed in the present research is discussed.

3.1 Proposed data splitting scheme: construction of optimal proportion for training and test sets

As noted, for building any intelligent system, each dataset (D) is split into two distinct parts, say T_1 (training set) and T_2 (test set) using any *splitting approach*. The *training set* is used to train the learner(s), whereas the *test set* is used to evaluate the performance of the learned model. There are several ways for partitioning data. The two well-accepted methods namely *hold-out* and *k-fold cross validation* are briefly explained below.

In hold-out approach, the *proportion* of data reserved for training and testing is typically at the discretion of the analysts (e.g., 50-50 or two-thirds for training and one-third for testing). On the other hand, *k-fold cross validation* technique takes a set of *m* examples and partitions them into *k* sub-sets (*folds*), each of size *m*/*k*. For each fold f_i, (*i*=1,.. , *k*) , a classifier is trained on a set combining other folds (f_j, *j*=1, .., *k* and $i \neq j$) and then tested on the fold, f_i. The trained accuracies are averaged over all *k* results. Such a strategy may be run a specific number of iterations, and a *standard deviation* is recorded to estimate the reliability of the classifier. In fact, the particular combined (k-1) folds which results maximum accuracy performance in comparison to the other combinations, may be treated as the best training set for the dataset.

However, the following limitations are identified for the above mentioned approaches.

i) More training examples normally cause *biasness* of the model over training set only.

ii) The estimated accuracy computed from the smaller test set is *less reliable* for prediction.

iii) In practice, there is no choice of *class distribution* (*i.e*., the percentage of examples of classes) among the examples in the training set.

So, deciding the best proportion to construct any intelligent model is a challenging task in data mining. More explicitly, what proportion of training and test sets is to be chosen in training and test sets for constructing effective model?

In this respect, Sarkar [59] performs an investigation addressing the question of what *proportion* of the samples should be devoted to the training set for developing a better classification model. The study suggests that any *equi-class distribution* data partition with less amount of training data (usually (30%, 70%) to (40%, 60%)) may be treated reasonably good for building a classification model irrespective to *domain*, *size* and *class imbalanced,* since such a partition gives usually better accuracy over test set resulting less number of informative rules in comparison to other partitions. One may here note that x% and y% in (x%, y%) denote respectively the percentages of training set and the test set, where x + y=100. However, one may apparently claim that less amount of training data is not significant enough for building classification model, assuming that less amount of training examples carries *less information*. But this claim may not be correct in case of imbalanced dataset. To justify it, we may take the concept of information theory where the amount of information (entropy) for an ensemble with multiple outcomes (*e.g*., X= {x_1, x_2, ..x_n}) is measured as:

$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$. Here, p($x_i$) denotes the probability of occurring event x_i . Mathematically, H(X) (*i.e*., entropy) becomes maximum if each of x_i has uniform probability. However, chances of occurring uniform probability in dataset (especially in imbalanced dataset) is practically very less. But for multi-class problem, it may be mathematically observed that if less percentage of equi-class distribution is taken in training set, then amount of entropy value reaches reasonably better.

Keeping the above view in mind, the present study primarily supports *equi-class* distribution data-partitioning scheme with less percentage of training data. Although, there exists scope of research to decide an optimal proportion of training and test sets for each dataset, and the present study makes use of parallel computing to identify the *optimal proportion*, starting a proportion closer to (25%, 75%). It is detailed below. First, the idea on *equi-class distribution* data partitioning is explained below before discussing the parallel approach.

Equi-class distribution of instances

- Suppose that 30% examples of each class from a dataset (D) are to be randomly included into training set (T_{Train}). Assume that there are 3 class values (say c_1, c_2 and c_3) and 150 examples in total in D, and the numbers of examples of class- types: c_1, c_2 and c_3 are respectively, say 33, 42 and 75. Then 10, 15 and 23 examples of class- types: c_1, c_2 and c_3 (based on the concept of *ceil* function, *e.g.*, $\left\lceil \frac{33\times 30}{100} \right\rceil = \left\lceil \frac{990}{100} \right\rceil = \lceil 9.9 \rceil = 10$) are included into T_{train} by random selection over D. Conceptually, this strategy is known as *sampling without replacement* because the examples which are selected for T_{Train} are immediately removed from D.

3.1.1 Use of parallel processing to find optimal proportion

From the viewpoint of knowledge discovery, deciding an optimal proportion (*i.e.*, how much training data is sufficient) is a key issue in data mining, as it varies from problem to problem. Precisely, tackling the issue needs lot amount of time, as it comes under combinatorial problem. Further, in medical domain, any pre-decided amount of training data for any dataset may not necessarily be the best one, as new significant changing occurrences may be included in the database. That is why, the essence of parallel processing is employed here to resolve this issue. More specifically, interest is shown to identify an optimal or near optimal proportion for training and test sets for each dataset by creating number of *threads* (or *processes*), where each thread/process operates on the same dataset but for a distinct partition (*e.g.*, (25% , 75%) by thread-0, (26%, 74%) by thread-1, and so on up to (40%, 60%)). In fact, the partitions are to be obtained parallelly as the outputs of a procedure named DATA-SPLITTER ( ) running in different threads/processes. Next, each pair: (*training set* and *test set*) is to be passed to PRISM classifier that also will be run by individual thread/process. Finally, an optimal pair (*i.e.*, proportion, say (m%, n%)) is to be identified based on the accuracy results computed over the test sets. Algorithmically, the code corresponding to the above discussed parallel logic is outlined below. Actually, three procedures namely DATA-SPLITTER( ), PRISM( ) and ACCURACY( ) are used in sequence to fulfil this job. More specifically, the procedure DATA-SPLITTER( ) is parallelized to operate different instances. It is parallelized, since the concept of random number generation is used in the procedure: DATA-SPLITTER( ) to chose instances in training set and the parallel m/c's have high probability to variate the random numbers in the same run as well as different runs.

Parallelized code for finding an optimal proportion:(training set, test set)

```
for  all  $p_i$  (i=0, … (n-1))  do  parallel
    begin
    call DATA-SPLITTER($s+p_i * d_i$)    /* The call results  a distinct training set say $E_{train}$  (local copy to $p_i$). The splitter uses random number generation function used in the language (in which it is implemented) to pick up examples at random from data file. Special care must be taken to maintain the variation in the examples from training set to training set generated by the threads or processes.
    variable: $s$ takes a fixed value (i.e., the starting percentage)  for resulting the training set, whereas  the variable $d$  takes a fixed difference value. Thus, the  expression: $s+p_i*d$,  gives the exact percentage of training examples to be included into the training set ($E_{train}$)  to be resulted by thread-id: $p_i$. Surely, the splitter is responsible for including: ($s+p_i*d$)  percentage of  instances of each class  in $E_{train}$  resulted by each thread /process, whereas (100- ($s+p_i *d$))  for the corresponding test set. */
    call PRISM($E_{train}$)     //  This call results  a distinct rule set R (local copy to $p_i$) for $E_{train}$ .
    call ACCURACY(R,  $E_{test}$)
    /* The ACCURACY( ) procedure finds prediction measure (local) over $E_{test}$ (test set corresponding to the training set $E_{train}$) by applying the rule set R. */
    end
```

- *Last step of the parallel strategy for finding optimal proportion*

After computing accuracy result (a local result: l_c) over the respective test set by individual thread/process, an optimal proportion (say, (m%, n%)) is returned by comparing l_c with a *global-accuracy* (say, g_c) via *mutual exclusion scheme* (in case of shared-memory model environment) or *message passing scheme* (in case of distributed memory model environment). To be more specific, as soon as a better accuracy result (achieved by an individual thread/process) is found, the corresponding *training* set and the *test* set are captured. This process continues until all the threads/processes finish such an adopted scheme. Finally, the *best* proportion is identified.

Why PRISM learner is used for selecting the optimal proportion?

Recall that the medical datasets are imbalanced (*i.e.*, rare case) and the PRISM algorithm can tackle rare case issue to a great extent, as it separately concentrates instances of each class. So, consideration of PRISM learner may be the best treatment for identifying an optimal proportion.

3.2 The exact hybrid model

To build the exact model, each dataset (D) is here first split into three sub-sets namely T_1, T_2 and T_3, as follows.

- T_1 : This set is, indeed, the training set based on optimal proportion (containing (m%, n%) examples of each class) and it is found by adopting the strategy as discussed in Section-3.1.1.
- T_2 : 15% examples of each class are selected at random from (D-T_1) and included into T_2. Simultaneously, these examples are removed from (D-T_1).

- T_3 : It is now D –(T_1+T_2).

The proposed hybrid architecture is constructed by integrating two individual classifiers viz. C4.5 and PRISM. The model consists of three phases, and it is depicted in Figure-3.1.

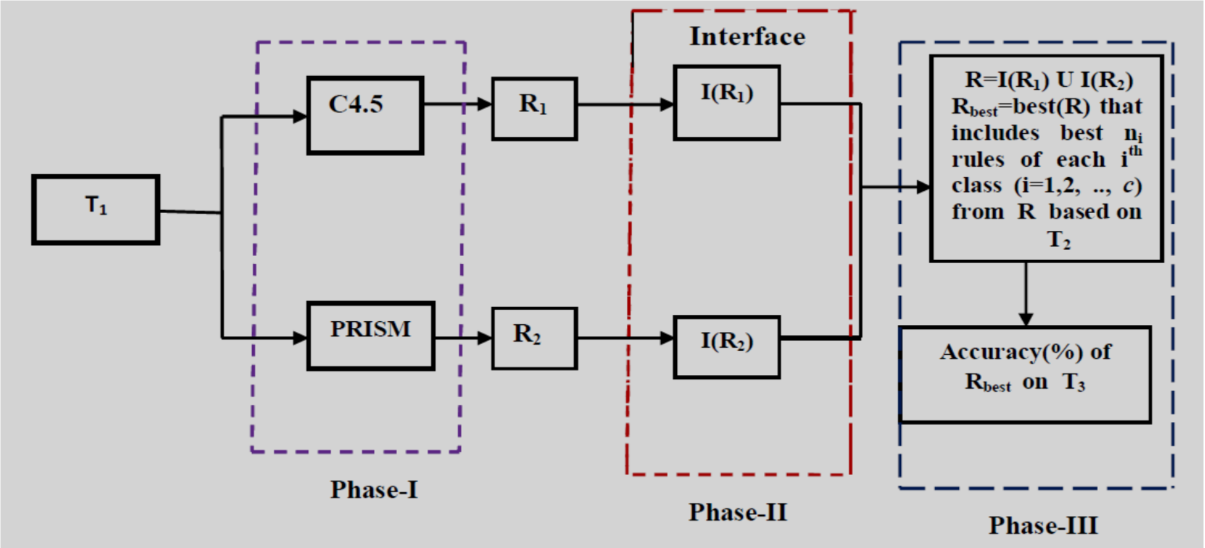
Interface
C4.5
R₁
I(R₁)
T₁
PRISM
R₂
I(R₂)
R=I(R₁) U I(R₂) R_best=best(R) that includes best n_i rules of each i^th class (i=1,2, .., c) from R based on T₂
Accuracy(%) of R_best on T₃
Phase-I
Phase-II
Phase-III

Fig-3.1: Hybrid model consisting of three phases

Let D be a dataset with '*c*' classes, and T_1, T_2 and T_3 are its partitions obtained by applying the suggested partitioning scheme (as discussed above). The three rule sets: R_1, R_2 and R_{best} (as shown in Figure-3.1) are explained below.

- R_1 : rule set generated by C4.5 classifier and I(R_1) represents its tabular format (applying interface[60]).
- R_2 : rule set generated by PRISM classifier and I(R_2) represents its tabular format.
- R_{best} : $\sum_{i=1}^{c} best(n_i)$, where $n_i = \left\lceil \frac{M_{C_i} + N_{C_i}}{2} \right\rceil$. Here, M_{C_i} and N_{C_i} represent the numbers of *i*-th class rules present in rule sets: R_1 and R_2 respectively.

Now, each phase of the hybrid approach is detailed below.

Algorithmic version of the proposed model

Phase-I : The selected two learners- C4.5 and PRISM are separately trained over T_1. Suppose they generate rule sets, say R_1 and R_2.

Phase-II **:** Apply the Interface s/w [60] to obtain the formatted tabular structure of R_1 and R_2 (denoted as I(R_1) and I(R_2)).

Next, find the *number* of rules of each class (out of '*c*' classes) in I(R_1) and I(R_2). These are, denoted as M_{C_i} and N_{C_i} (*i*=1, ..c).

Phase-III: The formatted rule sets: I(R_1) and I(R_2) are merged to result R (*i.e.*, R=I(R_1)U I(R_2)) and an attempt is made to derive high quality rule set (say R_{best}) from R by applying the following steps.

Initialize R_{best} =Φ // Φ denotes *empty set.*

```
Step-1: Remove the *default* rule (as explained in Appendix-C) from R requiring the demand of accurate rule in context of CDSS.
/* Steps-2 and 3 are the pre-processing steps for constructing a refined rule set. In fact, due to merging of two rule sets (each derived from one distinct learner), some rules in R may be redundant and/or conflict, and these two steps take care of this part. */
________________________________________________________________
                    /* Removing redundant rules */
Step-2: Remove the *redundant* rules from R, *i.e.,* R=R- $R_{red}$, where $R_{red}$ is a set of *redundant* rules found by applying the strategy suggested in Section-3.2.1 (*i*) and (*ii*).
    $R_{temp} \leftarrow \Phi$ // $R_{temp}$: a file to store temporarily some rules of R.
________________________________________________________________
                    /* Removing conflict rules */
Step-3: *for* each *unprocessed* rule: $r_i \in$ R *do*
            *begin*
  Step-3.1 Identify the *conflict* rules for $r_i$ in R (if any) by applying the strategy explained in
        Section-3.2.1 (*iii*) and dump them (including $r_i$) in the set: $R_{temp.}$ and finally perform:
                    R=R- $R_{temp}$.
    Step-3.2 *for* each rule $r_j$ in $R_{temp}$ *do*
      *begin*
      Step-3.2.1 Compute the correct classification *rate* of $r_j$ on $T_2$ by applying the formula:
          $f(r_j) = \frac{m}{|T_2|}$ .....(3.1), where *m* denotes the number of *correctly* classified examples
          in $T_2$ by $r_j$ and $|T_2|$ gives total number of examples present in $T_2$.
      *endfor // for Step-3.2*
  Step-3.3. Include the *best* rule (say $r_{best}$) of $R_{temp}$ into R, *i.e.*, R= RU{$r_{best}$}
  Step-3.4   $R_{temp} \leftarrow \Phi$
      *endfor // for Step-3*
________________________________________________________________
                    /* Finding $R_{best}$ */
Step-4: Measure the performance of each rule(*r*) in the current set: R on $T_2$ using:
$f(r) = \frac{m}{n+k}$ .....(3.2) , where *m* and *n* represent respectively the numbers of training examples *correctly* and *incorrectly* classified by *r* over $T_2$. Also, *k* denotes the number of *pre-conditions* present in rule *r*.
/* The equ.(3.2) plays an important role to resolve collision occurred among the rules of same class in R and it assists to chose the high quality rules with less number of pre-conditions .*/
Step-5: Arrange the rules of R class-wise in *descending* order of their performance over $T_2$.
```

Step-6: Choose the first n_i rules of each class '*i*' (*i*=1, 2, .., *c*) (as the best rules; assuming that sufficient rules are present in R) from R and include those into R_{best}.

/* *The value of* n_i *is initially decided and shown earlier*. */

Step-7: Apply R_{best} on T_3 to get its accuracy percentage (by applying equ.(2.1) given in Section-2.4).

3.2.1 Strategy to identify distinct and identical (i.e., sub/super) rules

To understand the idea on *distinct rule*, *identical rule* and *conflict rule*, let us take a *discretized* dataset of a classification problem (P) with 4 non-target attributes, say A_1, A_2, A_3 and A_4 and class (*i.e.*, target) attribute, say *C*.

i. *Identifying distinct rules:*

Two rules: r_1 and r_2 are distinct if *min* ($|pre(r_1)|$, $|pre(r_2)|$) ≠ *match*($pre(r_1)$, $pre(r_2)$), where $|pre(r)|$ results the number of *pre-conditions* (each with a numerical value) present in rule: *r* and the function: *min*(m_1, m_2) returns the *minimum* between two numbers: m_1, m_2. Further, the function: *match*( ) returns the number of pre-conditions matched between two selected rules.

Illustration: Let r_1 and r_2 be two rules for P as follows:

- r_1: *If* (A_1=4) *and* (A_2= 2) *and* (A_3=1), *then* C=1
- r_2: *If* (A_1=4) *and* (A_2= 2) *and* (A_4=2), *then* C=1

Clearly, each of r_1 and r_2 has 3 pre-conditions because each of these pre-conditions has numeric value (*e.g.*, A_1=4, A_2=2 and so on). However, r_1 and r_2 are *not* identical to each other because the *third* (from left) pre-conditions of r_1 and r_2 are not same (*i.e.*, these are respectively (A_3=1) and (A_4=2)), although their first *two* pre-conditions match exactly. Obviously, the number of *matched* pre-conditions between r_1 and r_2 is here 2, and it is not equal to *min* ($|pre(r_1)|$, $|pre(r_2)|$)= *min*(3,3)=3, *i.e.*, *min*(3,3)≠*match*($pre(r_1)$, $pre(r_2)$)=2. Hence, both r_1 and r_2 are here *distinct*. This implies that an instance belongs to class C=1 if its attributes' values are as follows:

(*a*) (A_1=4) and (A_2= 2) and (*b*) either (A_3=1) or (A_4=2) .

Certainly, the above two conditions may be parsed by two distinct rules.

ii. *Identical rules and redundant rules*

Two rules r_1 and r_2 are identical if *min* ($|pre(r_1)|$, $|pre(r_2)|$) = *match*($pre(r_1)$, $pre(r_2)$).

Let r_1 and r_2 be two rules for P as follows:

- r_1: *If* (A_1=4) *and* (A_2= 2) *and* (A_4=1), *then* C=1
- r_2: *If* (A_1=4) *and* (A_4=1), *then* C =1

The number of pre-conditions present in r_1 is 3, whereas it is 2 in r_2. In fact, the rules *match* at *two* places except for (A_2=2) of r_1. Clearly, the number of *matched* pre-conditions (m) is here 2 (*i.e.*, *m* =2). Again, *min* ($|pre(r_1)|$, $|pre(r_2)|$)= *min*(3,2) returns 2, and it equals to *m*.

Hence, both the rules are *identical* but one supersedes the other. In other words, out of these two rules, one is the *super* rule of the other. Obviously, $|pre(r_2)| = 2$, and it is *less* than $|pre(r_1)|=3$. So, rule r_2 is here treated as the *super* rule of r_1, and r_2 (instead of r_1) is well-expected to be present in rule set with the aim to classify more *test* examples. Definitely, r_1 is *redundant*, and it is to be removed from rule set R.

iii. Identifying conflict rules

Two rules are termed as *conflict* rules if their antecedent parts are *identical* but consequent parts (*i.e.*, class values) are *different*. For better realization, let r_1 and r_2 be two rules of P as:

- r_1: *If* (A_1=4) *and* (A_2= 2) *and* (A_4=1), *then* C=1
- r_2: *If* (A_1=4) *and* (A_2= 2) *and* (A_4=1), *then* C=2

Certainly, these two rules are the example of conflict rules, since their *antecedent* parts are same but class values are different (*i.e*., these are C=1 and C=2 respectively).

4. Experimental results and discussion

This section first discusses about the experiment conducted in the present study. Next, the obtained experimental results are arranged in tables. Finally, the results are analysed. For carrying out the experiment, the necessary materials are either downloaded or implemented in C language. *For example*, the classifier C4.5[2] is a downloaded s/w, whereas the PRISM algorithm (presented in Section-2.2), the interface[60], the proposed data-splitting procedure: DATA-SPLITTER( ) and the suggested hybrid method (discussed in Section-3.2) including the performance measuring programs are all implemented in C. Further, the DATA-SPILTTER() (for deciding an optimal *proportion*) is parallelized using OpenMP (Open Multi-Processing: an application programming interface that supports multi-platform shared memory multi-processing programming in C) on Cluster HPC machine(FUJITSU) with a total 256 cores (under one Master node). The Master node has 64GB main memory with 2 HDD, each of size 1TB, Speed-2.4GHz. The supporting operating system in the said HPC machine is the CentOs-6.2 with GNU/Linux Kernel. All the programs run in same machine.

4.1 Experiment and results

The performance of the proposed hybrid model (DTPR) and its base learners is experimented on 14 real-world medical datasets drawn from UCI repository[50].

Before conducting experiment over each dataset by the introduced hybrid model, one *pre-experiment* for identifying the optimal proportion of training and test sets for each data set is carried out by employing the suggested parallelized approach on the said HPC machine. From pre-experiment point of view, the parameters for DATA-SPLITTER($s+p_i * d_i$) are set as follows:

- *s* (*i.e*., the starting percentage) = 25, *d* (the fixed difference value) = 1.

- Total number of employed threads=35, *i.e.*, $0\leq p_i \leq 35$, where $p_{i\ (i=0,\ 1,\ \ldots 34)}$ denotes *thread-id*, and *ids*' are usually numbered as 0, 1, ..

For illustrating the parameter: $s+p_{i*}d_i$, we may first take, $p_0=0$ (i=0), then $s+p_{0*}d=25$; likewise, for $p_1=1$ (i=1), $s+p_{1*}d=25+1=26$, and so on.

Why thirty five threads are considered?

Based on the illustration of $s+p_{i*}d_i$, it is clear that unit interval in proportion is allocated between two consecutive threads. As per this calculation, total 35 threads are sufficient to reach the proportion (60%, 40%) starting from (25%, 75%). The partition: (60%, 40%) is considered here as the maximum limit for building classification system, since beyond this measure any developed system may result better performance on training data but not on unseen data, *i.e.*, chance of biasness of the system increases on training data.

For validating the optimal proportion (for each case), the suggested parallelized approach is repeated 10 times. Finally, the mean of 10 results along with *standard deviation* (s.d.) value is reported in the box below.

Breast-cancer: (45%, 55%) ± 2.13%, Dermatology: (55%, 45%) ± 2.61%, Pima-Indian: (56%, 44%) ± 1.6%, Ecoli: (45%, 55%) ± 0.9%, Heart(Hung.): (48%, 52%) ± 1.7%, Heart(Swiss): (35%, 65%) ± 1.13%, Heart(Clev.): (45%, 55%) ± 2.96%, Hepatitis: (48%, 52%) ± 3.11%, Liver-disorder: (54%, 56%) ± 1.11%, Lung-cancer:(60%, 40%) ± 1.34%, Lymphography:(50%,50%) ± 1.02%, New-Thyroid: (42%, 58%) ± 1.63%, Primary-Tumour: (48%, 52%) ± 1.45%, Sick: (38%, 62%) ± 1.36%

The *standard deviation* values displayed with *mean* proportion values infer that proportion value does not vary much from their *mean* values at different runs. That is why, *mean proportion* for each dataset is considered as the standard proportion in the present study.

Experiment (e) over each dataset

Experiment(*e*) for each problem consists of two sub-experiments denoted as: e_1 and e_2. Both e_1 and e_2 are detailed below.

Sub-experiment(e_1): At each run of this part, three distinct sub-sets of each dataset (D), viz.T_1, T_2 and T_3 are first constructed by applying the suggested data-splitting approach (as discussed in Section-3.1). It may be noted that the *percentage* of training examples in T_1 for each dataset is specified in the above box. As the training percentage in T_1 is known, so we may easily find the percentage of examples in T_2 and finally in T_3.

Now, the implemented hybrid approach is run in sequence on T_1 and T_2 respectively to train and to refine the model. More explicitly, T_1 is separately passed to the base learners: C4.5 and PRISM to generate two rule sets- R_1 and R_2. These two are first merged and then refined by the suggested model. Finally, the refined model is tested on T_3 to get the accuracy performance (as per the formula given in equ.(2.1)) for each dataset.

Sub-experiment(e_2): Here, each of C4.5 and PRISM is first trained on ($T_1 + T_2$). Then, the trained models are separately run on the test set: T_3 to obtain the accuracy performances for the dataset.

Logic for reliable estimation

For better estimation of the accuracy performance of the learners, the processes followed in e_1 and e_2 are repeated 20 times for each dataset. Finally, *mean* classification result over 20 results and standard deviation for each set are computed and reported in the performance Table-4.1 in favour of each learner. It is interesting to note that the size of training set in sub-experiment: e_2 is larger than the size of training set used in e_1.

In addition to accuracy result, TPR and FPR measures (as per *equns*(2.3) and (2.4)) for each of the learners are also computed at each run and finally the mean of 20 values at each case is noted in Table-4.2. They are shown pair-wise as: (TPR, FPR). However, *s.d.'s* measures for TPR and FPR are reported for hybrid model only.

Table 4.1: Accuracy results(%) of the classifiers based on the proposed data sampling approach over the selected datasets

Problem name	**C4.5** $(acc. \pm s.d)$	**PRISM** $(acc. \pm s.d)$	**DTPR** $(acc. \pm s.d)$
Breast Cancer Wisconsin	93.36 ± 2.45 (3)	93.95 ± 3.54 (2)	**96.25 ± 1.86 (1)**
Dermatology	91.96 ± 4.91 (2)	87.41 ± 5.13 (3)	**96.90 ± 2.76 (1)**
Pima Indian Diabetes	77.63 ± 1.51 (2)	75.34 ± 1.64 (3)	**86.28 ± 1.42 (1)**
Ecoli	83.23 ± 1.37 (2)	74.91 ± 3.77 (3)	**86.27 ± 1.28 (1)**
Heart (Hungarian)	76.33 ± 2.56 (3)	79.08 ± 2.43 (2)	**83.23 ± 1.21 (1)**
Heart (Swiss)	44.23 ± 6.90 (3)	46.29 ± 5.32 (2)	**52.81 ± 3.45 (1)**
Heart (Cleveland)	77.26 ± 3.40 (3)	78.20 ± 3.40 (2)	**82.01 ± 3.06 (1)**
Hepatitis	82.00 ± 3.40 (2)	80.77 ± 4.37 (3)	**86.59 ± 3.19 (1)**
Liver Disorder	80.17 ± 7.80 (2)	78.34 ± 6.24 (3)	**88.01 ± 4.06 (1)**
Lung Cancer	73.17 ± 9.29 (2)	64.72 ± 10.19 (3)	**80.81 ± 7.10 (1)**
Lymphography	76.98 ± 7.18 (2)	72.74 ± 7.74 (3)	**84.68 ± 7.01 (1)**
New-thyroid	91.33 ± 4.18 (3)	91.96 ± 3.85 (2)	**97.86 ± 1.74 (1)**
Primary Tumor	34.56 ± 3.98 (3)	36.21 ± 3.06 (2)	**41.82 ± 2.60 (1)**
Sick	97.72 ± 0.45 (2)	97.02 ± 0.56 (3)	**98.28 ± 0.51 (1)**
Average-rank	34/14=2.428	36/14 = 2.57	14/14=1

Note The value appearing just before '±' at each column indicates the *mean* accuracy (*acc*.), whereas the value appearing after '±' represents standard deviation value (*s.d.*).

Based on the accuracy results displayed in Table-4.1, the *rank* value of individual learner is placed within *parenthesis* along with the accuracy result. The rank values are later used to carry out statistical test for significant inference of the learners.

Table- 4.2: True positive and false positive rates of the classifiers based on the proposed data sampling approach over the selected datasets

Problem name	**C4.5** **(TPR, FPR)**	**PRISM** **(TPR, FPR)**	**DTPR** **(TPR, FPR)**
Breast Cancer Wisconsin	(0.821, 0.085)	(0.824, 0.081)	(0.921 ± 0.003, 0.042 ± 0.002)
Dermatology	(0.901, 0.045)	(0.873, 0.050)	(0.991 ± 0.001, 0.031 ± 0.002)
Pima Indian Diabetes	(0.874, 0.061)	(0.805, 0.062)	(0.901 ± 0.004, 0.060 ± 0.007)
Ecoli	(0.835,0.063)	(0.721, 0.080)	(0.885 ± 0.007, 0.090 ± 0.008)
Heart (Hungarian)	(0.811, 0.051)	(0.820, 0.041)	(0.989 ± 0.0021, 0.060 ± 0.006)
Heart (Swiss)	(0.644, 0.071)	(0.691, 0.063)	(0.790 ± 0.041, 0.091 ± 0.010)
Heart (Cleveland)	(0.802, 0.054)	(0.859, 0.052)	(0.916 ± 0.004, 0.079 ± 0.006)
Hepatitis	(0.795, 0.073)	(0.721, 0.074)	(0.849 ± 0.021, 0.051 ± 0.008)
Liver Disorder	(0.801, 0.064)	(0.743, 0.063)	(0.905 ± 0.003, 0.051 ± 0.004)
Lung Cancer	(0.813, 0.067)	(0.703, 0.101)	(0.885 ± 0.011, 0.058 ± 0.007)
Lymphography	(0.806, 0.062)	(0.729, 0.067)	(0.905 ± 0.006, 0.053 ± 0.004)
New-thyroid	(0.843, 0.007)	(0.861, 0.008)	(0.998 ± 0.001, 0.001 ± 0.001)
Primary Tumor	(0.358, 0.19)	(0.386, 0.16)	(0.635 ± 0.087,0.091 ± 0.010)
Sick	(0.889, 0.047)	(0.860, 0.049)	(0.989 ± 0.002, 0.042 ± 0.003)

4.2 Discussion on results

Discussion among the selected learners

On the basis of the empirical results over the UCI datasets, some important *findings* about the chosen learners are listed below.

- The head to head performance analysis of the learners (based on Table-4.1) infers that the pure C4.5 classifier performs better prediction over eight datasets, viz., *Dermatology, Diabetes, Ecoli, Hepatitis, Liver-disorder, Lung-cancer, Lymphography* and *Sick,* as compared to the individual learner PRISM. On the other hand, the table reveals that the performance of PRISM learner is comparatively good in comparison to C4.5 for some datasets with *rare* cases. Examples include *Breast-cancer*, *Heart(Hungerian)*, *Heart(Swiss)*, *Heart*(*Cleveland*) and *Primary Tumor.*

However, it is worth noting that the proposed ensemble model: DTPR outperforms its base algorithms over all the datasets. In particular, it gains genuinely higher accuracy with *low* standard deviation over 12 datasets namely *Breast-cancer*, *Diabetes*, *Ecoli*, *Heart*(*Cleveland*), *Heart*(Swiss), *Liver-disorder*, *Lung-cancer*, *Lymphography*, *New-thyroid*, *Primary-tumor* and *Sick*, in comparison to the pure C4.5 and PRISM learners. Further, the low standard deviation values attained by the model for the datasets affirm that the obtained accuracies are less scattered around the mean values. Accordingly, the introduced model is reliable for predicting unseen data.

- Another highlighting point is that the defined new system is more likely successful to operate *voluminous* and *high-dimensional* datasets using the suggested data-splitting scheme (selecting usually less amount of training data). Evidence includes *Lung-cancer* and *Sick* data sets in the present study.
- High mean TPR results with low *s.d*. values reveal that the proposed system has ability to detect accurately the disease-affected persons. On the other hand, the system also shows less FPR results on the diseases which in turns avoid unnecessary mental worry among the persons reporting the presence of disease for diseaseless people.
- The performance Table-4.1 indicates that the new model minimizes the *error-rate,* since classification accuracy increase in each case.

Learning algorithms and Nemeny test

Technically, comparing two or more algorithms based on their *mean accuracies* and *standard deviations* does not give always significant inference. That is why, *two-tailed Nemenyi statistical test*[53] is employed here in this purpose over the average ranks attained by the learners. For convenience, the critical values for the two-tailed Nemenyi test are furnished in Table-4.3, referring the paper published by Demsar [54]. Importantly, it is well-accepted that the performance of two classifiers is significantly different if the corresponding *average ranks* (achieved by two classifiers) differ by at least the *critical difference*(CD) which is defined as:

$\mathrm{CD} = q_{\alpha}\sqrt{\frac{k(k+1)}{6M}}$ …..(4.1), where *k* denotes the number of learners; M, the number of data sets; and q_α the critical value based on the Studentized range statistics divided by √2.

Note that the number of learners (k) in the present experiment is 3, and M is 14. Now, consulting Table-4.3, the value of CD for $q_{0.05}$ is computed as 0.8863, whereas it is 0.7755 for $q_{0.10}$. These values are used subsequently in this section for significant assessment of the learners.

Table 4.3: Critical values for the two-tailed Nemenyi test

#algorithms	2	3	4	5	6	7	8	9	10
$q_{0.05}$	1.960	2.343	2.569	2.728	2.850	2.949	3.031	3.102	3.164
$q_{0.10}$	1.645	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920

Significant assessment of the learners:

The average rank of each learner is already computed and shown at the bottom of Table-4.2. Now, based on the values of average rank and CD, the following statistical inferences can be drawn for the learners used in this study.

- The *difference* between the average-ranks of C4.5 and DTPR system is (2.428-1) =1.428. Clearly, this value is greater than both the values: 0.8863 and 0.7755. Therefore, the hybrid system is significantly better than the pure C4.5 at both $q_{0.05}$ and $q_{0.10}$.

- The difference between the average-ranks of PRISM and DTPR is (2.57-1)=1.57, and this value is also greater than both 0.8863 and 0.7755. Hence, the suggested model is significantly better than the PRISM for both $q_{0.05}$ and $q_{0.10}$.

Discussion with other state-of-art clinical systems

Of greater interest, the performance of the present system is compared with some unique models (especially designed for particular diseases) and these are collected from the standard survey papers[64,75] listed in Science Citation Index (SCI).

- *Breast cancer data set*: The DTPR system achieves 97.25% accuracy, whereas a hybrid model(GA+ANN)[66] developed by Bhardwaj and Tiwar results 97.24% accuracy by (50-50) training-testing partition. However, the main drawback of (GA+ANN) system is that no explicit model (in understandable: 'IF-THEN' format) is delivered by the said system for medical practitioner. Also, the number of examples included in training set is around 5% more than that of DTPR model.
- *Hepatitis:* On this data set, the model: DTPR gives 86.59% accuracy, whereas SVM-SA method[67] produces astonished outcome: **96.25%** which is very promising with regard to the other classification methods in the literature for this problem. But here too, the main drawback is that the knowledge gained by SVM-SA is implicit.
- *Heart disease*(*Hung. and Cleveland*): The DTPR model achieves **83.23%** and **82.01%** accuracies separately over these two sets, whereas a GA-SVM hybrid model[68] proposed by Xiaoyong Liu and Hui Fu shows 80% accuracy over the combined sets. Therefore, the accuracy result attained by DTPR is better than that of GA-SVM.

- *Liver disorder:* The DTPR framework exhibits **88.01%** accuracy, whereas the framework proposed by Fan *et al*.[69] gives around 85% accuracy. However, from the survey paper[74], it has been observed that any ANN-based hybrid intelligent model especially designed for Liver disorder data set usually achieves 90% or above performance accuracy. But the main drawback of any ANN-based system is that explicit rules are not generated for assisting the practitioners. So, the present system would be more suitable if both the criteria (performance and interpretability) are taken together.
- *Lymphography*: The present architecture yields 84.68% accuracy, whereas the hybrid technique: iNN(K)-L and CCBR introduced by McSherry[70] results **86.5%** accuracy. This says that the presented generalized model is better than the specific one.
- *Pima Indian Diabetes*: The DTPR model earns 86.28% accuracy, whereas the hybrid model (Case based reasoning and AI techniques) suggested by Marling *et al*.[71] attains 89.10% accuracy. This report infers that DTPR is not bad in comparison to the specialized one for diagnosing this disease.
- *Thyroid*: The 97.86% accuracy result achieved by DTPR model is very closer to the result (99%) obtained by the hybrid model proposed by Prasad *et al*. [72].

The short comparison of DTPR with the specialized models infers that the presented generalized model competes parallel with the specialized models.

5. Conclusion and Future work

Many predictive models for medical data mining have been introduced in the past decades but they have drawbacks like disease *specificity* of model and *vagueness* of patient's data. In this work, a novel generalised hybrid approach for diagnosing medical diseases is developed by combining C4.5 and PRISM learners. On the basis of performance comparison among the chosen classifiers in the present study and some specialised learners in the literature, the following remarks can be made in favour of the hybrid DTPR model.

- The decision rules refined and derived by DTPR are in easy understandable: IF-THEN form.
- The presented approach works well for all the chosen medical datasets (*i.e*., it is not disease specific) and it can be a good alternative to the well-known machine learning methods.
- The model achieves *low* standard deviation results computed over the datasets, that is, the achieved accuracy performance of the model does not vary uncertainly from run to run. Hence, it claims the *reliability* of the proposed hybrid approach for predicting unseen instances of medical datasets.
- The *high* TPR values bagged by the introduced ensemble approach give assurance of resulting more *accurate rules* for predicting unseen instances, and it is highly desired in CDSS.

- It has high capability to manage *rare case* issue (due to the application of equi-distribution of instances of all classes in training set).
- The defined new system is more likely successful to operate *voluminous* and *high-dimensional* datasets using data-splitting scheme with small amount of training data.

The present research has the following *potential implications*.

- The model provides less number of accurate and explanatory 'IF -THEN' rules for each dataset. As a result, the domain user (*i.e.,* practitioners) can easily detect diseases in quick and more correct way in comparison to other models in the literature, and can easily recommend proper medicine whenever required. Consequently, it claims to save lives and cost to a great extent.

Future scopes

There are few aspects of this research that may be improved further or extended in nearest future.

- Many data sets such as *Heart* (Hung./Swiss/Cleveland), *Hepatitis*, *Lung-cancer* in the present study are small or very small in size. So, more data with variation can be collected from different hospitals for testing the generalizability of the proposed model.
- The computation burden of the presented hybrid learning approach can be reduced by applying any feature screening approach on the original data sets. Accordingly, extracting excellent features of each database may assist the model to achieve better performance.
- The hybrid model can be converted to a specialized model for specific clinical data set for improving accuracy using genetic algorithm (GA). In this regard, each rule set refined by DTPR can be optimized by identifying appropriate fitness function for the specific data set.
- It would be interesting if the proposed framework is applied over big medical data such as MIL-Leukemia with number of non-target attributes=12583, number of instances=72 and number of classes=3, collecting from site: http://mldata.org/repository/data/viewslug/leukemia-mll/

6. Executive summary

- Medical datasets are imbalanced and uncertain in nature.
- CDSSs abate the cost of canonical treatments of diseases and save lives. Many CDSS's have been developed in the literature for effective treatments of diseases.
- Designing generalized accurate CDSS is a challenging task. The present research focuses to design a hybrid but generalized disease predictive model to handle any kind of disease datasets with better accuracy results.
- The present research has the following *potential implications.*

The model provides less number of accurate and explanatory 'IF -THEN' rules for each dataset. As a result, the domain user (*i.e.,* practitioners) can easily detect heart disease in quick and more correct way in comparison to other models in the literature, and can easily recommend proper medicine whenever required. Consequently, it claims to save lives and cost to a great extent.

- In summary, the model attains capability to drop global burden of disease treatments.

Author's contributions

The author's contribution in this research is original. The author has carefully read and approved the manuscript.

Compliance with Ethical Standards

The study is not funded by any agency. It does not involve other human participants and/or animal. The author declares that there is *no conflict of interests* regarding the publication of this paper.

Conflict-of-interest
The study is not funded by any agency. It does not involve other human participants and/or animal.
The author declares that there is no conflict of interests regarding the publication of this paper

References

[1] T.M. Mitchell, Machine learning, McGraw-Hill, New York, 1997

[2] W. Klosgen and J.M. Z' ytkow, Handbook of Data Mining and Knowledge Discovery, *Oxford University Press*, Oxford, 2002

[3] U.M. Fayyad, G. Piatetsky-Shapiro and P. Smyth, From Data Mining to Knowledge Discovery: An Overview, In *Advances in Knowledge Discovery and Data Mining,* MIT Press, Cambridge, Mas., (1996)1-36.

[4] J. Han and M. Kamber, Data Mining Concepts and Techniques, *San Francisco*, CA; *Elsevier Inc.* 2nd Edition, 2006.

[5] J.R. Quinlan, C4.5: Programs for machine learning, *San Mateo, CA*: Morgan Kaufman, 1993

[6] L. M. Fu, Knowledge discovery based on neural networks, *Communications of the ACM*, 42(11) (1999) 47-50.

[7] Z. Pawlak and R. Slowinski, Rough set approach to multi-attribute decision analysis, European Journal of Operational Research, 472 (1994) 43-459.

[8] M. Montalbano, Decision tables, SRA: Chicago, 1974

[9] Jadzia Cendrowska, PRISM: An algorithm for inducing modular rules, Int. Journal of Man-Machine Studies, 27 (1987) 349-370

[10] William W.Cohen, Fast Effective Rule Induction, In proceeding of Twelfth International Conference on Machine Learning (1995) 115-123.

[11] R.O. Duda and P.E. Hurt, Pattern Classification and Scene Analysis, *John Wiley and Sons*, 1973

[12] J. FURNKRANZ, Separate-and-Conquer Rule Learning, Artificial Intelligence Review, 13 (1999)3–54, Kluwer Academic Publishers (Printed in the Netherlands).

[13] H. Chen and C. Tan, Prediction of type 2 diabetes based on several element levels in blood and chemo metrics, Bio. Trace Elem. Res., 147(1-3) (2012) 67-74.

[14] A.X. Garg, N. K Adhikari, H. McDonald, M.P. R. Arellano, P.J. Devereaux and J. Beyene, Effects of computerized clinical decision support systems on practitioner performance and Patient outcomes: A systematic review, *JAMA* 2005 (March 9), 293(10) (2005) 1223-1238.

[15] K. Kensaku, A. Caitlin, E. Houlihan, B. Andrew and F. L. David, Improving clinical practice using clinical decision support systems: A systematic review of trials to identify features critical to success, BMJ (Clinical Research Ed.), 330(7494) (2005) 765.

[16] L. Moja, K.H. Kwag, T. Lytras, L. Bertizzolo, L. Brandt, V. Pecoraro, G. Rigon, A. Vaona, Ruggiero, F., Mangia, M., Iorio, A., Kunnamo, I. and S. Bonovas, Effectiveness of computerized decision support systems linked to electronic health records: A systematic review and meta analysis, American Journal of Public Health, 104(12) (2014) 12-22.

[17] M. Narasingarao, R. Manda, G. Sridhar, K. Madhu and A. Rao, A clinical decision support system using multilayer perceptron neural network to assess well being in diabetes, Journal Assoc. Phys. India, 57 (2009) 127-133.

[18] Tanveer Syeda-Mahmood, Plenary Talk: The role of machine learning in clinical decision support, SPIE Newsroom, 2015

[19] M. Thirugnanam, P. Kumar, S.V. Srivatsan and C.R. Nerlesh, Improving the prediction rate of diabetes diagnosis using fuzzy, neural net work, case based approach (FNC), Procedia Eng., 38 (2012) 1709-1718.

[20] K. Wagholikar, V. Sundararajan, and A. Deshpande, Modeling Paradigms for Medical Diagnostic Decision Support: A Survey and Future Directions, *Journal of Medical Systems* (Journal of Medical Systems), 36(2012)3029-3049.

[21] C.Z. Ye, J. Yang, D.Y. Geng, Y. Zhou and N.Y. Chen, Fuzzy rules to predict degree of malignancy in brain glioma, Medical and Biological Engineering and Computing, 40(2) (2002) 145-152.

[22] P.K. Srimani and S. Koti Manjula, Rough set approach for optimal rule generation in medical data, International Journal of Conceptions on Computing and Information Technology, 2(2) (2014) 9-13.

[23] J. Komorowski and A. Ohrn, Modelling prognostic power of cardiac tests using rough sets, Artificial Intelligence in Medicine, 15(1999) 167-191.

[24] M.S. Shanker, Using neural networks to predict the onset of diabetes mellitus, Journal of Chemical Information and Computer Science, 36(1996) 35-41.

[25] S. Lekkas and L. Mikhailov, Evolving fuzzy medical diagnosis of Pima Indians diabetes and of dermatological disease, Artificial Intelligence in Medicine, 50(2010) 17-126.

[26] M.W. Aslam, Z. Zhu and A.K. Nandi, Feature generation using genetic programming with comparative partner selection for diabetes classification, Expert Systems with Applications, 40(2013) 5402-5412.

[27] H. Temurtas, N. Yumusak and F. Temurtas, A comparative study on diabetes disease diagnosis using neural networks, Expert Systems with Applications, 36 (2009) 8610-8615.

[28] A.T. Azar and S.M. EI-Metwally, Decision Tree classifiers for automated medical diagnoses, Neural Computing and Applications, 23(7) (2013) 2387-2403.

[29] Mark Hall and Eibe Frank, Combining I Bayes and Decision Tables, Proceedings of the Twenty-First International FLAIRS Conference, Coconut Grove, Florida, published by the AAAI Press, Menlo Park, California, (2008) 318-319.

[30] B.K. Sarkar, S.S. Sana and K.S. Chaudhuri, Selecting Informative rules with Parallel Genetic Algorithm in Classification Problem, Applied Mathematics and Computation, 218(7) (2011) 3247-3264.

[31] B.K. Sarkar, S.S. Sana and K.S. Chaudhuri, A Genetic Algorithm-based Rule Extraction System, Applied Soft Computing , 2(1) (2012) 238-254.

[32] Z. Pawlak, K. Slowinski and R. Slowinski, Rough classification of patients after highly selected vagotomy for duodenal ulcer, International J. Man-Machine Studies, 24 (1986) 413-433.

[33] K. Slowinski, R. Slowinski and J. Stefanowski, Rough sets approach to analysis of data from potential lavage in acute pancreatitis, Medical Informatics, 13(1988)143-159.

[34] XM Huang and YH Zhang, A new application of rough set to ECG recognition, International Conference on Machine Learning and Cybernetics (IEEE Explore), 3(2003) 1729–1734.

[35] S. Tsumoto, Mining diagnostic rules from clinical databases using rough sets and medical diagnostic model, International Journal of Information Science, 162(2) (2004) 65-80.

[36] M. Ohsaki, H. Abe, S. Tsumoto, H. Yokoi and T. Yamaguchi, Evaluation of rule interestingness measures in medical knowledge discovery in databases, Artificial Intelligence in Medicine, 41(3) (2007) 177–96.

[37] P.K. Srimani MS. Koti, Cost sensitivity analysis and the prediction of optimal rules for medical data by using rough set theory, International Journal of Industrial and Manufacturing Engineering, (2012) 74–80.

[38] N. Abdelhamid and F. Thabtah, Associative Classification Approaches: Review and Comparison, Journal of Information and Knowledge Management (JIKM) Worldscinet., 13(2014)

[39] J.R. Quinlan, Induction of decision trees, Machine Learning, 1(1986) 81-106.

[40] K. Park, K.M. Lee and S. Lee, Perceptual grouping of 3D features in aerial image using decision tree classifier, Proceedings of 1999 International Conference on Image Processing, 1(1999) 31-35.

[41] S.D. Macarthur, E.B. Carla, C.K. Avinash and L.S. Broderick, Interactive content-based image retrieval using relevance feedback, Computer Vision and Image Understanding, (2002) 55-75.

[42] J.P. Gonzalez and U. Ozguner, Lane detection using histogram-based segmentation and decision trees, Proceedings of IEEE Intelligent Transportation Systems, (2000) 346-351.

[43] M. Chen, A. Zheng, J. Lloyd, M. Jordan and E. Brewer, Failure diagnosis using decision trees, Proceedings of the International Conference on Autonomic Computing, 2004

[44] F. Bonchi, G. Manco, C. Renso, M. Nanni, D. Pedreschi and S. Ruggieri, Data mining for intelligent web caching, Proceedings of International Conference on Information Technology: Coding and computing, (2001) 599-603.

[45] A.C. Stasis, E.N. Loukis, S.A. Pavlopoulos and D. Koutsouris, Using Decision Tree Algorithms as a basis for a Heart Sound Diagnosis Decision Support System, Proceedings of the 4th Annual IEEE Conf on Information Technology Applications in Biomedicine, UK, (2003) 354-357.

[46] D.E. Brown, Introduction to Data Mining for Medical Informatics, Clinics in Laboratory Medicine, 28(2008) 9-35.

[47] M.A. Bramer, Automatic induction of classification rules from examples using N-PRISM, Research and Development in Intelligent Systems(Springer-Verlag),16 (2000) 99–121.

[48] A. Bramer, An information-theoretic approach to the pre-pruning of classification rules, Intelligent Information Processing, Kluwer (in: B. N. M Musen, R. Studer (Eds.), (2002) 201-212.

[49] F. Stahl and M. Bramer, P-PRISM: A Computationally Efficient Approach to Scaling up Classification Rule Induction, Artificial Intelligence in Theory and Practice II, IFIP – The International Federation for Information Processing, 276(2008) 77-86

[50] C. Blake, E. Koegh. and C.J. Mertz, (1999): Repository of Machine Learning, University of California at Irvine. URL: http: //www.*mlearn.ics.uci.edu/MLRepository.html..*

[51] A. Tanwani and M. Farooq, The role of biomedical dataset in classification, Proceedings of AMIE: 12th International Conference on Artificial Intelligence, Springer, Verlag, Berlin, Heidelberg, (2009) 370-374.

[52] B.K. Sarkar, S.S. Sana and K.S. Chaudhuri, MIL: a data discretization approach, Int. Journal of Data Mining, Modeling and Management (IJDMMM), 3(3) (2011) 303–318.

[53] P.B. Nemenyi, Distribution-free multiple comparisons, PhD thesis, Princeton University, 1963

[54] J. Demsar, Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Machine Learning Research, 7(2006) 1–30.

[55] C. Apte and S. Hong, Predictive equity returns from security data, Advance in Knowledge Discovery and Data Mining, AAAI press and MIT press, 22 (1996) 541-569.

[56] P. Clark and T. Niblett, The CN2 algorithm, Machine Learning, 3(1989) 261-283.

[57] J. Catlett, On Changing Continuous Attributes into Ordered Discrete Attributes, In Proceedings of European Working Session on Learning, (1991) 164-178.

[58] B. Pfahringer, Supervised and unsupervised discretization of continuous features, In Proceedings of 12th international conference on machine learning, (1995) 456-463.

[59] B.K. Sarkar, A case study on partitioning data for classification, International Journal of Information and Decision Sciences, 8(1) (2016) 73-91.

[60] B.K. Sarkar, K. Sachdev, Bharati Swaraj and A. Bhaskar, An Interface for converting rules generated by C4.5 to the most suitable format for Genetic Algorithm, Proceedings of the Eighth International Conference on IT (CIT-2005), (2005)113-115, Bhubaneswar, India, (2005), Dec. 20-23).

[61] F. Stahl and M.A. Bramer, Random PRISM: An alternative to Random Forests, Research and Development in Intelligent Systems(Springer), 28(2011) (2014) 5-18.

[62] E. Frank and I. Witten, Generating accurate rule sets without global optimization, Proceedings of the Fifteenth International Conference on Machine Learning, Madison Wisconsin, (1998)144–151.

[63] Ramesh Kumar, M. Sambath, and S. Ravi, Relevant association rule mining from medical dataset using new irrelevant rule elimination technique, Proceedings of ICICES, (2013) 300-304.

[64] S. Gambhir, S.K. Malik and Y. Kumar, Role of soft-computing approaches in healthcare domain: A mini review, Journal of Medical Systems (Springer), (2016) doi:10.1007/S10916-016-0651.

[65] C.V. Subbulakshmi and S.N. Deepa, Medical Dataset Classification: A Machine Learning Paradigm Integrating Particle Swarm Optimization with Extreme Learning Machine Classifier, Scientific World Journal, (2016) doi:2016:7137054.

[66] A. Bhardwaj and A. Tiwari, Breast cancer diagnosis using genetically optimized neural network model, Expert Systems with Applications, (2015) 1– 15.

[67] J.S. Sartakhti, M.H. Zangooei and K. Mozafari, Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (SVM-SA), Computer Methods and Programs in Biomedicine, 108(2) (2012) 570– 579.

[68] Liu Xiaoyong and Fu Hui, PSO-Based Support Vector Machine with Cuckoo Search Technique for Clinical Disease Diagnoses, The Scientific World Journal, Article ID 548483, (2014) 7 pages http://dx.doi.org/10.1155/2014/548483

[69] C.Y. Fana, P.C. Chang, J.J. Lin and J.C. Hsieh, A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification, Applied Soft Computing, 11(2011) 632–644.

[70] D. McSherry, Conversational case-based reasoning in medical decision making, Artificial Intelligence in Medicine, 52(2) (2011) 59–66.

[71] C. Marling, S. Montani, I. Bichindaritz and P. Funk, Synergistic case-based reasoning in medical domains, Expert Systems with Applications, 41(2014) 249–259.

[72] V. Prasad, T.S. Rao and P. Babu, Thyroid disease diagnosis via hybrid architecture composing rough data sets theory and machine learning algorithms, Soft Computing, 20(3) (2016) 1179–1189.

[73] N. Japkowicz and S. Stephen The class imbalance problem: A systematic study, Intelligent Data Analysis, 6(5) (2002) 429–449

[74] A. Singh and B. Pandey, Intelligent techniques and applications in liver disorders: a survey, Int. J. Biomedical Engineering and Technology, 16(1) (2014) 27-70

[75] J. Downs, R.F. Harrison, R.L. Kennedy and S.S. Cross, Application of the fuzzy ARTMAP neural network model to medical pattern classification tasks, Artificial Intelligence in Medicine, 8(4) (1996) 403–428.

[76] H. Kahramanli and N. Allahverdi, Extracting rules for classification problems: AIS based approach, Expert Systems with Applications, 36(7) (2009) 10494 – 10502.

[77] M.G. Tsipouras, C. Voglis and D.I. Fotiadis, A Framework for Fuzzy Expert System Creation—Application to Cardiovascular Diseases, IEEE Transactions on Biomedical Engineering, 54, 11(2007) 2089–2105.

[78] U. Markowska-Kaczmar and R. Matkowski, Experimental Study of Evolutionary Based Method of Rule Extraction from Neural Networks in Medical Data, Applications in Medicine, Web Mining, Marketing, Image and Signal Mining, Lecture Notes in Computer Science, 4065(2006) 76–90.

[79] P. Luukka, Feature selection using fuzzy entropy measures with similarity classifier, Expert Systems with Applications, 38(4) (2011) 4600–4607.

APPENDIX- A

Classification problem: A classification problem (P) is described by a set of attributes categorized as: non-target (i.e., *feature*) attribute and class (also known as target) attribute. Each problem contains only one target attribute but many feature attributes.

For better understanding the classification problem, let us consider the '*golf -playing'* problem. The problem takes here four *feature* attributes viz., *Outlook*, *Temperature*, *Humidity* and *Windy*. The target is named as *Playing-decision*. The feature attributes are denoted as respectively A_1, A_2, A_3 and A_4, whereas C is used for the class attribute. The possible non-discretized values of the attributes are noted below.

Name of attribute	Values
Outlook (A_1)	Sunny, Overcast, Rain

Humidity (A_2)	High, Normal
Temperature (A_3)	Hot, Mild, Cool
Windy (A_4)	Strong, Weak
Playing-decision (C)	No, Yes

A non-discretized data set of 14 days observations for this problem is shown in Table-A.1. Here, D_i (i=1,…14) represents day.

Table A.1: A sample of non-discretized '*golf-playing*' data set

Sl. No.	**Non-Target Attributes (A_i, *i*=1,…4)**				**Playing-**
	Outlook (A_1)	Temperature(A_2)	Humidity (A_3)	Windy (A_4)	**decision**
D_1	Sunny	Hot	High	Strong	No
D_2	Sunny	Hot	High	Strong	No
D_3	Overcast	Hot	High	Weak	Yes
D_4	Rain	Mild	High	Weak	Yes
D_5	Rain	Cool	Normal	Weak	Yes
D_6	Rain	Cool	Normal	Strong	No
D_7	Overcast	Cool	Normal	Strong	Yes
D_8	Sunny	Mild	High	Weak	No
D_9	Sunny	Cool	Normal	Weak	Yes
D_{10}	Rain	Mild	Normal	Weak	Yes
D_{11}	Sunny	Mild	Normal	Strong	Yes
D_{12}	Overcast	Mild	High	Strong	Yes
D_{13}	Overcast	Hot	Normal	Weak	Yes
D_{14}	Rain	Mild	High	Strong	No

Followings are adopted as the discretized (mapping) values of the respective attributes. The discretized values are shown within parentheses.

Name of attribute	**Discrete values shown within parenthesis**
Outlook (A_1)	Sunny (1), Overcast(2), Rain(3)
Humidity (A_2)	High (1), Normal (2)
Temperature (A_3)	Hot (1), Mild (2), Cool (3)
Windy (A_4)	Strong (1), Weak (2)
Playing-decision (C)	No(0), Yes (1)

Referring the above mentioned discretized values, Table-A.1 looks likeTable-A.2 and it is, indeed, the output of any discretizer like MIL[23].

Table A.2: Discretized '*golf-playing*' data set

Day	**Outlook**	**Humidity**	**Temp**	**Windy**	**Playing-decisio**
1	**1**	**1**	**1**	**2**	**O**
2	**1**	**1**	**1**	**1**	**O**
3	**2**	**1**	**1**	**2**	**1**
4	**3**	**1**	**2**	**2**	**1**
5	**3**	**2**	**3**	**2**	**1**
6	**3**	**2**	**3**	**1**	**O**
7	**2**	**2**	**3**	**1**	**1**
8	**1**	**1**	**2**	**2**	**0**
9	**1**	**2**	**3**	**2**	**1**
10	**3**	**2**	**2**	**2**	**1**
1 1	**1**	**2**	**2**	**1**	**1**

12	2	1	2	1	1
13	2	2	1	2	1
14	3	1	2	1	O

C4.5 and the unpruned decision tree

Based on the concept of *entropy* (as discussed in Section-2.1), the unpruned decision tree built by C4.5 learner on the non-discretized data set (as shown in Table-A.1) is depicted in Figure-A.1

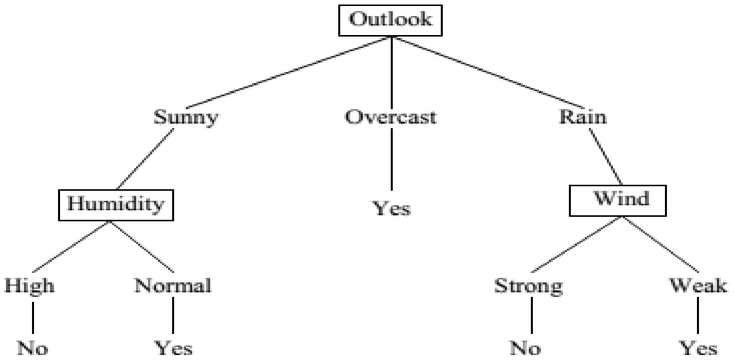

Fig-A.1: Decision tree built by C4.5 over non-discretized '*golf-playing*' data set

Unpruned tree and the decision rule

The decision rules (from root to leaf) derived by C4.5 learner are shown below.

Rule-1: If (Outlook = Overcast), Then Playing-decision= Yes;

Rule-2: If (Outlook=sunny) and (Humidity = High), Then Playing-decision = No

Rule-3: If (Outlook = Sunny) and (Humidity = Normal), Then Playing-decision = Yes;

Rule-4: If (Outlook = Rain) and (Windy = Strong) , Then Playing-decision = No;

Rule-5: If (Outlook = Rain) and (Windy = Weak), Then Playing-decision = Yes;

Rule-6: ( ) → C = Yes (*default rule*: a rule with *majority* class instances)

After *rule-post pruning* (*i.e.,* that prunes each rule independently of others by removing any pre-conditions that result in improving its estimated accuracy), we get,

Rule-1: If (Outlook = Overcast), Then Playing-decision = Yes;

Rule-2: If (Humidity = High)**,** Then Playing-decision = No

Rule-3: If (Outlook = Sunny) and (Humidity = Normal), Then Playing-decision = Yes;

Rule-4: If (Outlook = Rain) and (Windy = Strong) , Then Playing-decision = No;

Rule-5: If (Outlook = Rain) and (Windy = Weak), Then Playing-decision = Yes;

Rule-6: ( ) → C = Yes (*default rule*: a rule with *majority* class instances)

APPENDIX-B

A brief illustration on PRISM algorithm

It is already noted that there are 14 binary-class (i.e., *yes* and *no*) examples in '*golf-playing*' data set.

- Let us first consider '*yes*' as the recommended class, *i.e.,* Playing-decision=Yes.
- Presently, temp-data-set contains entire *golf-playing* data set.
- Now, compute *a* (accuracy)= p/t with respect to the present *temp-dataset* for each value of individual attribute (*i.e*., A=v). Here, *p*=number of instances (in *temp-dataset*) in which A=v and Playing-decision= '*yes*'. However, *t*= number of instances (in temp-dataset) in which A=**v** but Playing-decision = yes or no. Calculation is shown below.

Attribute	**Values of the attribute and a= p/t**
Outlook	For *Outlook = Sunny,* there are total 5 instances (out of 14) in which *Sunny* appears, *i.e*., *t*=5. But out of 5, only 2 give '*yes*', *i.e*., p=2 Thus, $\mathit{Outlook}_{(\text{Sunny})}$=2/5. Likewise, $\text{Outlook}_{(\text{Overcast})}$=4/4(max), $\text{Outlook}_{(\text{Rain})}$==3/5
Temperature	$\text{Temperature}_{(\text{Hot})}$ = 2/4, $\text{Temperature}_{(\text{Mild})}$ = 4/6 $\text{Temperature}_{(\text{Cool})}$ = 3/4
Humidity	$\text{Humidity}_{(\text{High})}$ = 3/7, $\text{Humidity}_{(\text{Normal})}$ = 6/7
Windy	$\text{Windy}_{(\text{Weak})}$ = 6/8 , $\text{Windy}_{(\text{Strong})}$ = 3/6

So, r_1: If (Outlook=Overcast), Then class = PL. [a complete rule]

At this stage, discard the instances covered by r_1 from T and update the *temp-dataset* consisting of present T (removing the earlier contents of *temp-dataset*). Continuing in this way, we finally get the complete rule set as:

r_1: If (Outlook=Overcast), Then Playing-decision = Yes.

r_2: If (Humidity=Normal) and (Windy=Weak), Then Playing-decision =Yes

r_3: If (Humidity =Normal) and (Outlook=Sunny), Then Playing-decision = Yes

r_4: If (Outlook=Rain) and (Windy=Weak), Then Playing-decision =Yes.

r_5: If (Outlook=Sunny) and (Humidity=High), Then Playing-decision = No

APPENDIX-C

Role of the Interface s/w: Let us take the rules generated by C4.5 from '*golf.playing'* problem (as shown in Appendix-A) . These are once again presented below.

Rule-1: If (Outlook = Overcast), Then Playing-decision = Yes;

The rule with discretized attributes' values is as: If (Outlook=2), Then Playing-decision =1

Rule-2: If (Humidity = High), Then Playing-decision = No;

The rule with discretized attributes' values is as: If (Humidity=1), Then Playing-decision =0

Rule-3: If (Outlook = Sunny) and (Humidity = Normal), Then Playing-decision = Yes;

The rule with discretized attributes' values is as: If (Outlook=1) and (Humidity=2), Then Playing-decision =1

Rule-4: If (Outlook = Rain) and (Windy = Strong) , Then Playing-decision = No;

The rule with discretized attributes' values is as: If (Outlook=3) and (Windy=1), Then Playing-decision =0

Rule-5: If (Outlook = Rain) and (Windy = Weak), Then Playing-decision = Yes;

The rule with discretized attributes' values is as: If (Outlook=3) and (Windy=2), Then Playing-decision =1

Rule-6: ( ) → C =Yes

The rule in discretized form is as: ( )→ C=1

The Interface s/w[21] gives the tabular representation of above presented rules with discretized attributes' values as follows by eliminating 'If ' and 'Then' parts.

Output of Interface s/w

	Outlook	*Humidity*	*Temperature*	*Windy*	*Playing-decision*
Rule 1:	2	*	*	*	1
Rule 2:	*	1	*	*	0
Rule 3:	1	2	*	*	1
Rule 4:	3	*	*	1	0
Rule 5:	3	*	*	2	1
Rule-6	*	*	*	*	1 (*default rule*)

The symbol '*' in a rule denotes here the *don't care* symbol, and implies that the attribute corresponding to '*' has no importance in that rule. Now, *for illustration* of tabular representation of rule, let us consider a rule, say Rule-1. In fact, this rule has only *one* pre-condition with numeric value (2), and it is undoubtedly for attribute *Outlook,* since *pre-conditions* of the rest attributes: *Humidity*, *Temperature* and *Windy* are absent in this rule. That is why, numeric value 2 is placed just below the *Outlook* attribute in the row representing Rule-1 and '*' for the respective positions of the other non-target attributes. Surely, the row-representing this rules will be read as:

If (Outlook=Overcast(2)), *Then* (Playing-decision=Yes(1)).

Note that the *length* of each of these rules is measured here 5, since total number of attributes (including the target one) is 5. Apart from the above rules in the rule set, a *default rule* is added to the set. In fact, it is originally generated by C4.5 for each data set. This rule is without any *conditions* and has a *consequent* part only. The assigned class-label in the consequent part is the *majority* class label of the samples in the training set. In general, it is placed at the bottom of the generated rule set.

The original form of the rules generated by the PRISM learner from '*golf-playing.data*' are as follows:

r_1: If (Outlook=Overcast), Then Playing-decision = Yes.

r_2: If (Humidity=Normal) and (Windy=Weak), Then Playing-decision =Yes

r_3: If (Humidity =Normal) and (Outlook=Sunny), Then Playing-decision =Yes

r_4: If (Outlook=Rain) and (Windy=Weak), Then Playing-decision =Yes.

r_5: If (Outlook=Sunny) and (Humidity=High), Then Playing-decision = No

Thus. tabular like representation of the above rule set performed by the *Interface* s/w (eliminating 'If – Then' parts from each rule) is shown below:

	Outlook	*Humidity*	*Temperature*	*Windy*	*Playing-decision*
Rule 1:	2	*	*	*	1 (*Identical* rule with Rule-1 in C4.5)
Rule 2:	*	2	*	2	1 (*Distinct* rule with Rule-3 in C4.5)
Rule 3:	1	2	*	*	1 ((*Identica*l rule with Rule-3 in C4.5)
Rule 4:	3	*	*	2	1 (*Identical* rule with Rule-5 in C4.5)
Rule 5:	1	1	*	*	0 (*sub-rule* of Rule-2 in C4.5)

Highlights

- The proposed hybrid system provides user friendly environment to the practitioners for detecting diseases.
- The suggested system keeps ability to predict very good accuracy rate in comparison to the other state-of-the-art-models in the literature.
- True positive rate yield by the system for each dataset is high, whereas false positive rate is low.
- The empirical outcomes positively demonstrate that the new system is effective in undertaking disease treatment.