Suitability of Various Intelligent Tree Based Classifiers for Diagnosing Noisy Medical Data

Abeer M.Mahmoud[1, 2]
[1]Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt.
[2] Faculty of Computer & Information Sciences, Princess Nora University, Riyadh – KSA.
abeer_f13@ yahoo.com, ammahmoud@pnu.edu.sa

Abstract

Building intelligent classification models can essentially help in medical data analysis and diagnosis specially when the available medical data sets are described as noisy data. This paper investigates various supervised machine learning decision tree classifiers on a number of noisy medical data. These tree based algorithms were grouped into three main categories. These are single tree classifiers (Decision Stump, C4.5, Rep Tree), the ensemble models (bagging and Adaboost and random forest) and the Credal Decision Trees (CDTs), that is based on imprecise probabilities and uncertainty measures. Various experiments are tested with different values of distortions. Experimental results in this paper show that although there is no standard algorithm that fits all types of data, the ensemble classifiers reported higher classification accuracy than single tree approaches. Also, the Credal DTs outperforms the single tree classifiers and recorded almost equal classification accuracy as ensemble models with a shorter model building time. It specially suitable in noisy domains of numerical attributes databases. This is due the robustness of Credal classifiers to missing data.

Keywords: *Decision Tree Classifies, Noisy Medical Data, Credal Decision Tree*

1. Introduction

One of the main benefits of using intelligent decision tree based classifiers is the ability to generate understandable and simple knowledge structures in the form of hierarchical trees or sets of rules. In addition, intelligent decision tree based classifiers have a low computational cost when predicting or classifying new cases, ability to handle symbolic and numeric input variables, high ability to handle noisy training data. In the other hand, the instability of decision trees was successfully solved by ensembles methods where multiple trees built from different subsets to improve the robustness of the final classifier [1]. Building tree based classifiers requires going through two main phases. These are growing phase and pruning phase. In growing phase, the tree-based model is achieved by recursively splitting the training set based on successful select of best feature until all or most of the records belong to one class. In the pruning phase, the problem of over fitting the data is handled. It also generalizes the tree by removing the noise and distortion. Usually, the accuracy of the classification increases in the pruning phase[1-3].

From statistical point of view, the classifiers resulted from individual decision trees are not strong enough to achieve the desired high classification accuracy, but they can be improved by building ensemble modeling instead. In the other hand, Predictions made by imprecise-probability models are often uncertain. Measuring the quality of an indeterminate prediction by a single number is important to fairly compare different models, but a principled approach to this problem is currently missing. Therefore, combining the strengths of both imprecise probability and decision tree based classifier may improve overall performance of the classifier, this is achieved in [4-6].

The rest of this paper is organized as follow: Section2, explains briefly a number of intelligent tree based classifiers that of main interest in this paper. Section 3, presents the medical databases used to evaluate and investigate the suitability of the tree based classifiers. Results and discussion are shown in Section 4. Section 5, ends up with conclusion.

2. Intelligent Tree-Based Classifiers

A classification problem is made of a training set with main target to map a set of features (£,),to a class label (Ç).The values of (£,Ç) are sampled and identically distributed according to the algorithm splitting and processing criteria. In medical domain, the classification goal is to discover from data how a patient's profile history in the form of a set of medical tests is related to a specific disease. The classification problem can be solved using many machine learning approaches, one of them is the supervised decision tree models which have been known in literatures since long time as a simple to understand and an easy to implement. These models usually provide the capability of modeling different attribute types and handling complex relations between high number of variables.

A *decision tree model T* is a directed, a cyclic graph in form of a tree. It starts with the root node that has zero incoming edges and proceeds by more nodes that have exactly one incoming edge and zero or more outgoing edges. The tree nodes may be called a leaf node if it has no outgoing edges, otherwise it is called *internal nod*e. While the leaf node is labeled with one class label; the internal node is labeled with one predictor attribute called the *splitting attribut*e. For an internal node n, let $E = \{e_1,e_2,....,e_k\}$ be the set of outgoing edges and let $Q= \{q_1,q_2,.....,q_k\}$ be the set of predicates such that edge e_i is associated with predicate q_i. Let us define the notion of the *family* of tuples of a node with respect to database D. The family $F(r)$of the root node r of decision tree T is the set of all tuples in D. For a non-root node $n \in T, n \neq r,$ let p be the parent of n in T and let $q_{p\rightarrow n}$ be the predicate on the edge $e_{p\rightarrow n}$ from p to n. The family of the node n is the set of tuples $F (n)$ such that for each tuple $t \in F(n), t\in F(p)$ and $q_{p\rightarrow n} (t)$ evaluates to true. Informally, the family of a node n is the set of tuples of the database that follows the path from the root to n when being classified by the tree. Each path W from the root r to a leaf node n corresponds to a classification rule $R=P \rightarrow c,$ where P is the conjunction of the predicates along the edges in W and c is the class label of node n. Pruning, the method most widely used for obtaining right sized trees, which aim to reduce size of the tree and minimize misclassification rate as well. In the following, a brief description of three main categories of intelligent tree based classifiers that are of main interest of this paper results [1-3]

2.1 Single Tree Classifier

C4.5 algorithm constructs the decision tree with divide and conquer approach, each node is associated with a set of training examples and a test sample. To reach the discriminating attribute which splits the training data into subsets associated to node children, C4.5 use the Entropy (S), Gain (S, A), Split Information (S,A) and Gain Ratio (S, A). These are shown in Eq.1- Eq.4, respectively[2].

$$Entropy\ (S) = \sum_{i=1}^{c} -P_i log_2 P_i \qquad (1)$$

Where $\boldsymbol{P_i}$is the proportion of $\boldsymbol{S}$ to class $\boldsymbol{i}$and $\boldsymbol{c}$ is the possible values of class (target) attribute.

$$\mathrm{Gain(S,A)} = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \qquad (2)$$

Where $\boldsymbol{A}$is an attribute, and $\boldsymbol{S_v}$is the subset of $\boldsymbol{S}$ for which attribute $\boldsymbol{A}$ has value $\boldsymbol{v}$

$$\mathrm{SplitInformation(S,A)} = -\sum \frac{|S_i|}{|S|} log_2 \frac{|S_i|}{|S|} \qquad (3)$$

$$GainRatio\,(S,A) = \frac{Gain(S,A)}{SpiltInformation(S,A)} \qquad (4\ )$$

In Wekamining tool [7] the J48 refers to the C4.5 algorithm which was developed by J. Ross Quinlan[2]. J48 supports two methods of C4.5 pruning. The first method is known as sub tree replacement, and it works by replacing nodes in decision tree with leaf. The second method is sub tree raising and is achieved by moving nodes upwards toward the root of tree and also replacing other nodes on the same way.

Decision Stump: is a single tree based classifier that builds simple binary tree called “stumps”(1 level decision tree) for both nominal and numeric training samples. It deal with mission values by extending a third branch from the stump or treating ‘missing ‘as a separate attribute value. It performs regression or classification using mean-squared error and entropy, respectively [3].

REP Tree: is based on C4.5 algorithm. It produces a classification or regression tree using information gain as the splitting criterion, and prunes it using reduced error pruning (back fitting). Actually, it sorts the values of numeric attributes once, then use the sorted list to calculate the best split. It also, handle the missing attribute values using fractional instances. Rep Tree, handle non-numeric (discrete) variables using regular decision tree with reduced-error pruning but when dealing with numeric attributes it minimizes total variance[3].

2.2 Ensemble Tree Classifier

Sometimes it is needed to combine several imperfect hypotheses to get a better one. In ensemble tree classifiers, the goal is to combine several trees in order to improve generalizability/robustness over a single one. Two families of ensemble methods are distinguished. First, *averaging methods*, where the trees are built independently and then average their predictions. Usually, the combined tree records a better classification accuracy because its variance is reduced. Second, *boosting method*, where the base estimators are built sequentially and one tries to reduce the bias of the combined estimator. The motivation is to combine several weak models to produce a stronger ensemble. There exists a variety of procedures on how to generate an ensemble. The most popular are bagging, random forests (Breiman [8,9]) as well as the Boosting algorithms of Freund and Schapire [10].

Bagging: Combines hypotheses via majority voting[8-11].It uses a bootstrap technique to resample the training data sets D. To forma resampled data set D_i. Each sample in D has a probability of1/n of being drawn in any trial. The most often predicted class label will be the final classification result. bagging methods form a class of algorithms which build several instances of a black-box estimator on random subsets of the original training set and then aggregate their individual predictions to form a final prediction. These methods are used as a way to reduce the variance of a base estimator, by introducing randomization into its construction procedure and then making an ensemble out of it. In many cases, bagging methods constitute a very simple way to improve performance over with a single model. It also provides a way to reduce over-fitting. Where it works best with strong and complex models.

Bootstrapping: Most popular ensemble learning technique. This approach computes a weighted majority of hypotheses which can boost performance of a weak learner. It start by operating on a weighted training set, where each training instance has a weight. When an input is misclassified by a hypothesis, the approach increase its weight so that the next hypothesis is more likely to classify it correctly[7-11].

Ada Boost: The initial classifier is constructed from original data set where every sample has an equal distribution ratio of 1. In the Boosting method training data set D_i, the distribution ratios are made different among samples depending on their prediction accuracy in the previous data set D_{i-1}. If a sample has a lower prediction accuracy rate in D_{i-1}, it will be given a higher weight in D_i and therefore get a higher possibility to be selected in D_i. [9-11]

Random forests: ensemble decision tree methods using random feature selection, produced by Leo Breiman [8,9]. This early random decision trees method combines bagging and random feature selection methods to generate multiple classifiers. Random forests based on CART method. Prediction is made by aggregating (majority vote for classification or averaging for regression) the predictions of the ensemble. Random forest generally exhibits a substantial performance improvement over the single tree classifier such as CART andC4.5. It yields generalization error rate that compares favorably to Adaboost, yet is more robust to noise. In random forest , each tree is built from a sample drawn with replacement from the training set. In addition, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model[3,7,10]

2.3 Credial Tree Classifier

Information theory connects information with the concept of uncertainty where the amount of information obtained by an action must be measured by a reduction in uncertainly. Abell´an and Moral [12] have developed an algorithm for designing decision trees, called *credal decision trees* (CDTs) that is based on the imprecise probability theory (Imprecise Dirichlet Model (IDM))[13].The variable selection process for this algorithm is based on imprecise probabilities and uncertainty measures on credal sets, i.e. closed and convex sets of probability distributions. In this manner, this algorithm considers that the training set is not reliable when the variable selection process is carried out. This method obtains good experimental results, especially when noisy data are classified [6,11]. According to[6], the Credal-C4.5depends on a parameter s and it has a low computational cost with $s\leq 1$. The split criterion employed to build Credal Decision Trees (CDTs) [12] is based on imprecise probabilities and the application of uncertainty measures on credal sets. The mathematical basis of this procedure can described as follows: Let Z be a variable with values in $\{z_1,.....,z_k\}$. Let us suppose a probability distribution $p(z_j)$, $z_1,.....k$ defined for each value z_j from a dataset. Walley's Imprecise Dirichlet Model (IDM) [13] is used to estimate probability intervals from the data set for each value of the variable z, in the following way (Equ.5)

$$p(z_j) \in \left[\frac{n_{zj}}{N+s}, \frac{n_{zj+s}}{N+s}\right], \quad j = 1, \dots\dots, k \qquad (5)$$

With n_{zj} as the frequency of the set of values($Z=z_j$) in the data set, N the sample size and s a given hyper-parameters.

This representation gives rise to specific kind of creedal set on the variable Z, K(Z) defined as Eq.6

$$K(Z) = \left\{p \middle| p(z_j) \in \left[\frac{n_{zj}}{N+s}, \frac{n_{zj+s}}{N+s}\right], \quad j = 1, \dots\dots, k \quad \right\} \qquad (6)$$

On this type of sets (creedal sets) , uncertainty measures can be applied. The product to build CDTs uses the maximum of entropy function on the above defined creedal set. This function denoted as H*, is defined as $H^*(K(Z))=\max\{H(p)|p \in K(Z)\}$, where the function H is the shannon's entropy function. H* is a total uncertainly measure which well known for this type of set. The procedure for H* in the IDM reaches its lowest cost with $s<= 1$.

The method for building Credal-C4.5 trees is similar to the Quinlan‘s C4.5 algorithm [8]. The main difference is that Credal-C4.5 estimates the values of the features and class variable by using imprecise probabilities and uncertainty measures on credal sets. Credal-C4.5 considers that the training set is not very reliable because it can be affected by class or attribute noise. So, Credal-C4.5 can be considered as a proper method for noisy domains. Credal-C4.5 is created by replacing the *Info-Gain Ratio* split criterion from C4.5 with the *Imprecise Info-Gain Ratio* (IIGR) split criterion. This criterion can be defined as follows: in a classification problem, let *C* be the class variable, $\{X_1,\ldots\ldots\ldots..,X_m\}$ the set of features, and *X* a feature; then $\mathrm{IIGR}^D(C,X)$ $=\frac{IIG^D(C,X)}{H(X)}$, where *Imprecise Info-Gain* (IIG) is equal to:

$$IIG^D(C,X) = H^*\left(K^D(C)\right) - \sum_i \mathrm{p}^D(\mathrm{X}=\mathrm{x}_i)\mathrm{H}^*\left(\mathrm{K}^D(\mathrm{C}|\mathrm{X}=\mathrm{x}_i)\right), \qquad (7)$$

with $K^D(C)$ and $K^D(C|X=x_i)$ are the credal sets obtained via the IDM for the *C* and $(C|X=x_i)$ variables respectively, for a partition *D* of the data set (see Abell´an and Moral [1]); $P^D(X= x_i)$ (i= 1, ………,n) is a probability distribution that belongs to the credal set $K^D(X)$.We choose the probability distribution P^D from $K^D(X)$ that maximizes the following expression: $\sum_i p^D(X = x_i)H^*\left(K^D(C|X = x_i)\right)$, it is simple to calculate this probability distribution. From the setB=$\{x_j \in X|H(C|X=x_j)=\max_{x_i}\{H(C|X=x_i)\}\}$, The probability distribution P^D will be equal to

$$P^D(x_i) = \begin{cases} \frac{n_{xi}}{N+s} & if\ x_i \notin B \\ \frac{n_{xi+\frac{s}{m}}}{N+s} & if\ x_i \in B \end{cases} \qquad (8)$$

Where m is the number of elements of B. This expression shares out s among the values x_i with $H(C|X=x_i)$ maximum.

where *P(C=)* is the relative frequency of class value .

3 . Medical Databases

For evaluating the intelligent tree based classifiers, four medical databases were chosen. In the following the databases diagnosis domain, attributes, number of instance and the necessary preprocessing needed before applying the classifiers are briefly described.

3.1 Thrombosis Disease Database

The data was made through the “ Discovery Challenge Competition “, organized as part of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Database in Prague [14]. Some preprocessing were necessary for the data as some information was missing and other was duplicated or has illegal values. After preprocessing, the database contains 406 patient instances with 13 features, where the target attribute thrombosis represents the degree of thrombosis and has the values 0:negative (no thrombosis), 1: positive (the most sever one), 2:positive (sever), 3: positive (mild).

3.2 Hypothyroid Disease Database

The data was downloaded from UCI machine learning repository [15]. It contains 2800 instance of 30 features mostly numeric. The target class label is named diagnosis level that mainly mapped to one of three values (negative, Primary hypothyroid or Compensated hypothyroid). It suffers from several missing attribute values (signified by "?"). A preprocessing is needed as some decision tree classifiers require a conversion of numerical attributes to categorical ones.

3.3 Arrhythmia Disease Database

The data was downloaded from UCI machine learning repository[15]. It contains 452 patient instances and 279 attributes, 206 of which are linear valued and the rest are nominal. The aim is to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 16 groups. Class 01 refers to 'normal' ECG classes 02 to 15 refers to different classes of arrhythmia and class 16 refers to the rest of unclassified ones. Therefore, the database must preprocessed to convert class labels to categorical feature and treatment of missing values.

3.4 Heart Disease Database

The data was downloaded from UCI machine learning repository[15]. The database contains 13 features. The class target is to classify patient case as absence (1) or presence (2) of heart disease. The class attribute is not categorical and therefore it was converted to categorical before building the classifiers.

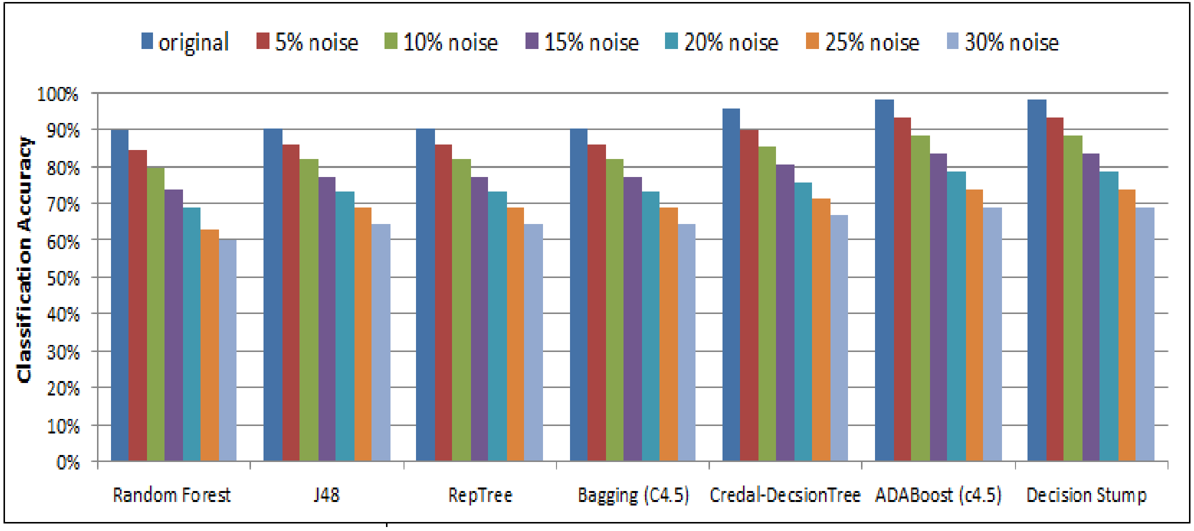

Fig 1. Illustration of classification accuracy versus various tree based classifiers on thrombosis disease

4. Results& Discussion

The results obtained for each database are analyzed, where all experiments and results were performed using libraries from WEKA 3.7.3 machine learning environment[7]. In WEKA software[7], the C4.5 algorithm is named J48. Applying WEKA filters on each database, we create seven versions of each where a different random noise percentages {0%,5%,10%,15%,20%,25%,30%} were added. Therefore, the total number of database are (7*4)=28 databases. For Credal-tree classifier, the Weka-IP: a Weka plugin for credal classification was used in our experiments. This is an extended version of WEKA[16]. Credal Decision Tree (CDT) [12,13]: is an extension to imprecise probability of the classification trees, which uses imprecise probability for both deciding on which feature to branch and for classifying the instances. Finally, each of the following results were validated using a 10-fold cross validation method.

Table 1: Results of various tree based classifiers on Thrombosis data base

Results	Decision Stump	ADABoost + c4.5	Credal-DecsionTree	Bagging (C4.5)	RepTree	J48	Random Forest
Original	98.18%	98.17%	95.56%	90.22%	90.22%	90.22%	89.83%
5%	93.09%	93.09%	89.96%	85.79%	85.79%	85.79%	84.62%
10%	88.27%	88.27%	85.43%	81.75%	81.75%	81.75%	79.40%
15%	83.31%	83.31%	80.69%	77.31%	77.31%	77.31%	73.79%
20%	78.49%	78.49%	75.41%	73.01%	73.01%	73.01%	68.84%
25%	73.53%	73.53%	71.43%	68.58%	68.58%	68.58%	62.97%
30%	68.71%	68.71%	66.68%	64.28%	64.28%	64.28%	59.97%

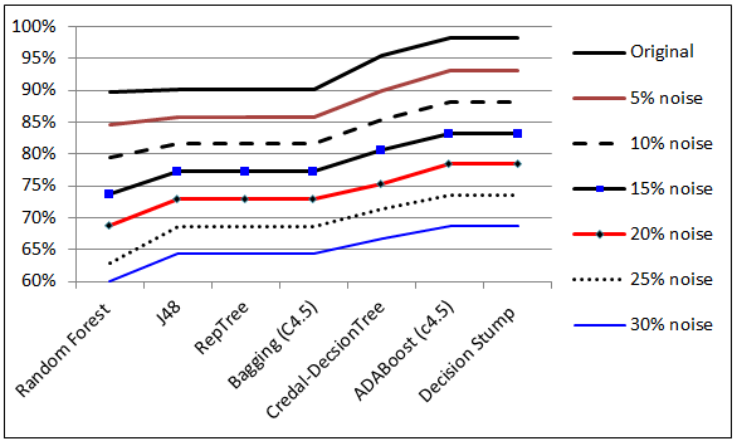

Fig 2. Various tree based classifiers results on thrombosis database with different noise levels

4.1 Thrombosis Disease Database

Building seven intelligent tree based algorithms for the thrombosis disease database reported the results shown in table1,Fig.1a nd Fig.2. From these illustrations, First, it is noted that the decision stump gave the highest classification accuracy for original thrombosis dataset. The fig 1. shows that the order in classification accuracy results from (decision stump, ADA Boost(C4.5), Credal-Decision tree, Bagging (C4.5), Rep Tree, J48, Random Forest). Therefore , the decision stump recorded is the best suitable algorithm for thrombosis database nature. Also, the rep tree and J48 gave almost identical results for all noise percentages as well as the original data, therefore one of them is illustrated. From fig 2. it is obvious that the performance of all previous algorithms were infected by adding the noise by relatively small decrement in classification accuracy.

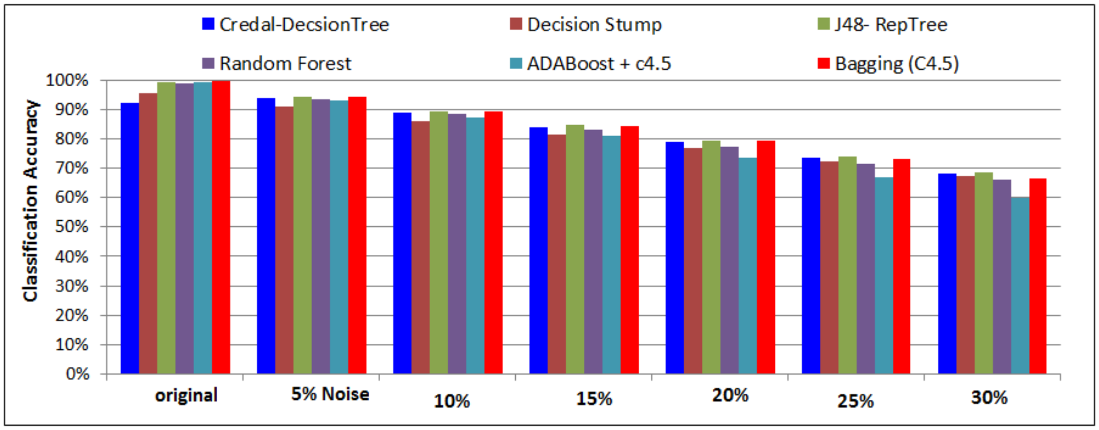

Fig 3. Illustration of classification accuracy versus various tree based classifiers on Hypothyroid disease

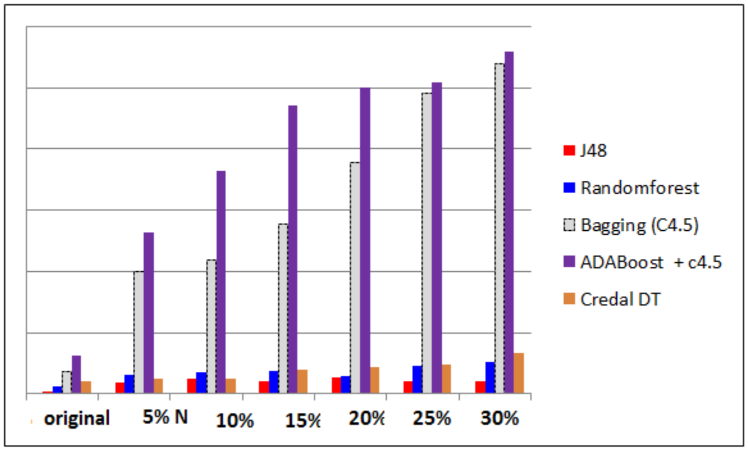

Fig 4. Time consumed during building different tree based classifiers on Hypothyroid database

4.2 Hypothyroid Disease Database

Fig. 3 reports different classifiers behavior than illustrated for thrombosis database due to different attribute and data nature (mostly numerical attributes). Fig. 3 shows that the bagging(C4.5), (ADABoost+C4.5), J48, Rep Tree, Random Forest, Decision Stump and Credal-Decision Tree(CDT)) reported the classification accuracy on the original database(0%) ordered from highest to lowest, respectively. In the other hand, the CDT recorded the highest stability and accuracy among the previous algorithms proportional to the noise ratio. Fig. 4 , adds a privilege for the CDT, where despite recording a very close accuracy to bagging and ADA boost(C4.5), it recorded smaller time to build the model than both bagging and ADA boost(C4.5) .

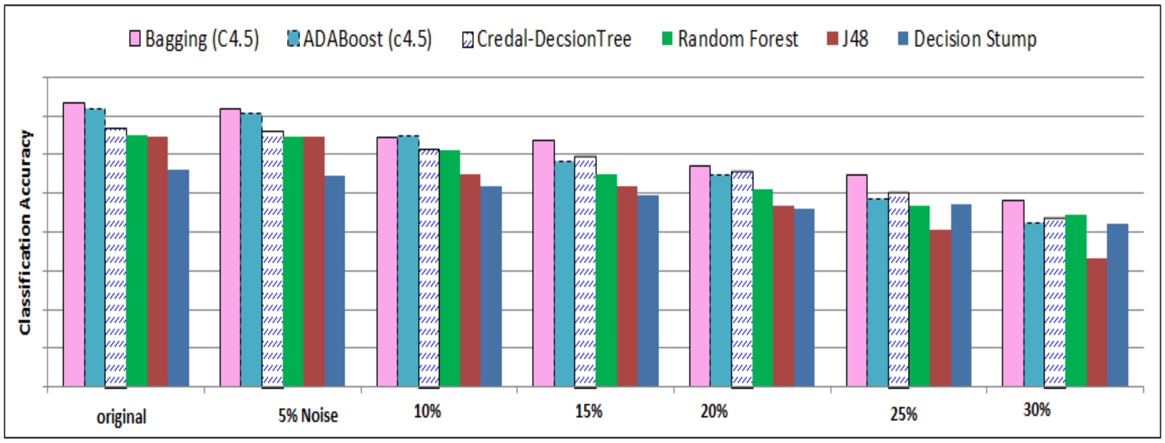

Fig 5. Illustration of classification accuracy versus various tree based classifiers on Arrhythmia disease

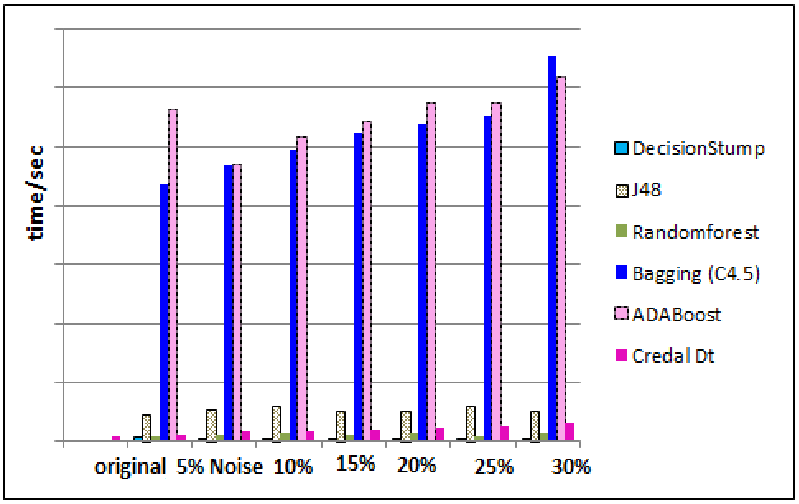

Fig 6. Time consumed during building different tree based classifiers on Arrhythmia database

4.3Arrhythmia Disease Database

The bagging (C4.5), (ADABoost+C4.5), and Credal-Decision Tree recorded the highest classification accuracy on the original and the different noisy databases as well (see fig.5). Fig. 6also reflects the competitive performance of the Credal DT with other classifiers in terms of classification accuracy and time consumed to build the intelligent classifiers.

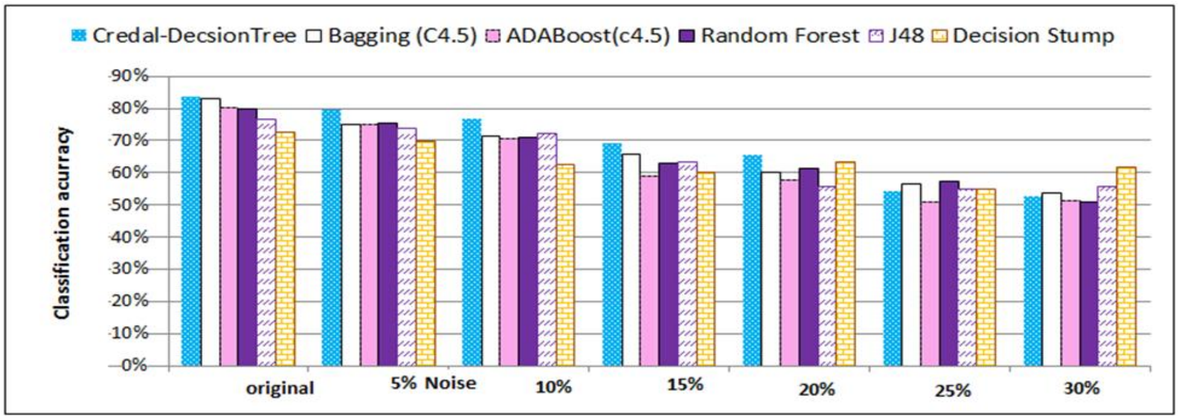

Fig 7. Illustration of classification accuracy versus various tree based classifiers on Heart disease

4.4 Heart Disease Database

Fig. 7 shows that the Credal-Decision Tree reported the highest classification accuracy on the original data. Then Bagging(C4.5), ADA boost(C4.5), Random Forest, J48 and Decision Stump. The Credal DT competed well and gave higher classification accuracy even with different distortion levels.

5. Conclusions

Decision tree based classifiers are of main interest for researchers in knowledge discovery and data analysis as they don’t require complexity in understanding their models and can handle different types of data, especially when the available medical data sets are described as noisy one. The paper investigated various supervised machine learning decision tree classifiers on a number of noisy medical data. The classifiers were grouped into three main categories. (single tree classifiers (Decision Stump, C4.5, Rep Tree), ensemble models (bagging and Adaboost and random forest) and Credal Decision Trees (CDTs)). Various experiments were tested with different values of distortions. Although there is no standard algorithm that fits all types of data, the ensemble classifiers reported higher classification accuracy than single tree approaches. The Credal DTs outperformed the single tree classifiers and recorded a better accuracy specially in noisy domains and mostly numerical attributes databases. It recorded near equal classification accuracy to ensemble classifiers but with shorter time to build the models. This is due the robustness of Credal classifiers to missing attribute values in the training data.

References

[1] Tom M.Mitchell,.” Machine Learning”, publisher MaGRAW-Hill-Internaltional editions, computer science series , 1997.

[2] Quinlan JR. C4.5: programs for machine learning. MorganKaufmann; 1993.

[3] Witten IH, Frank E., “ Data mining: practical machine learningtools and techniques”. 2nd edition. San Francisco: MorganKaufmann; 2005.

[4] J. Abell´an and S. Moral. “Upper entropy of credal sets. applications to Credal classification”. International Journal of Approximate Reasoning, vol.39, pp.235–255, 2005.

[5] G. Corani and M. Zaffalon. “ Credal model averaging: an extension of Bayesian model averaging to imprecise probabilities”, In Daelemans, W., Goethals, B., Morik, K. (Eds), ECML PKDD 2008: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Lecture notes in Computer Science 5211, Springer, pp.257–271, 2008.

[6] Carlos J. Mantas and Joaqu´ın Abell´an , “Credal decision trees in noisy domains” , proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning., pp.23-25, 2014.

[7] Bouckaert R, Frank E, Hall M, Kirkby R, Reutemann P, Seewald A, Scuse D. WEKA manual for version 3-7-3. Hamilton, NewZealand: University of Waikato; 2010.

[8] Breiman L. “Bagging predictors. Machine Learning”; vol.24, no.2, pp.23–140. 1996.

[9] Bauer E, Kohavi R. “An empirical comparison of voting classification algorithms: Bagging, boosting, and variants”. Machine Learning , vol.36, no.2, pp.105–39,1999.

[10] Breiman L. ‘Random forests – random features”, Technical Report 567, University of California, Berkley, 1999.

[11] Abell’an, J. and A. Masegosa, “Bagging Decision Trees on Data Sets with Classification Noise”, in S.Link and H.Prade, eds, `Foundations of Information and Knowledge Systems', vol. 5956 of Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, pp. 248-265. , 2010.

[12] J.Abell’an J. and S. Moral,” Building classification trees using the total uncertainty criterion”, International Journal of Intelligent Systems, vol. 1812, pp.1215-1225, 2003.

[13] P. Walley, Inferences from multinomial data, learning about a bag of marbles, Journal of the Royal Statistical Society, Series B, vol.58, pp.3-57, 1996.

[14] Zytkow, Jan.M., and Rauch, Jan. Principles of data mining and knowledge discovery, third European conference, PKDD'99 Prague, Czech Republic, September 1999, proceedings, Springer, Lecture Notes in Artificial Intelligence 1704.

[15] Frank, A. and A. Asuncion (2010), `UCI Machine Learning Repository'.

[16] URL: http://archive.ics.uci.edu/ml

[17] URL: http://decsai.ugr.es/˜andrew/weka-ip.html.