Journal of Advanced Computing
(2013) 1: 9-27
doi:10.7726/jac.2013.1002
Research Article

Medical Diagnosis Using Ensemble Classifiers - A Novel Machine-Learning Approach

P. K. Srimani[1*] and Manjula Sanjay Koti[2]

Received 1 November 2012; Published online 17 November 2012

Abstract

In designing high-performance computer-aided diagnosis systems, improving the accuracies of the machine-learning algorithms is vital, and ensemble data-mining methods (EDMM), learning algorithms having a combination of multiple base models, are the most suggested methods. In the present study, the experiments are conducted on five medical datasets and the results justify that there is a drastic enhancement in the performance of the base classifiers, and this certainly would facilitate effective medical diagnosis, which in turn would contribute to the health index of the patients. Further, it is concluded that only selected classifiers are to be used for each data set, and for some specified cases, ensemble classifiers need not be proposed. A proper selection of the classifier is recommended in order to achieve optimal accuracy with regard to a specific medical data set.

Keywords: Accuracy; Ensemble classifiers; Machine-learning algorithms; Medical diagnosis; Meta-classifiers

1. Introduction

Supervised learning methods are methods that attempt to discover relationships between the input attributes (independent variables) and the target attributes (dependent variables), and the relationship discovered is represented in a structure referred to as a model (Rokach, 2009). Usually, models can be used for predicting the value of the target attribute by knowing the values of the input attributes.

Recent developments in computational learning theory have led to methods that enhance the performance or extend the capabilities of the basic learning schemes. These learning schemes have been called “meta-learning schemes” or “meta-classifiers” or “ensemblers”. Ensemble Data-Mining Methods, also known as Committee Methods or Model Combiners, are machine-learning methods that leverage the power of multiple models to achieve better prediction accuracy than any of the individual models could do on their own (Dietterich, 2000). The basic goal while designing an

*Corresponding e-mail: profsrimanipk@gmail.com
1* Professor, Doctor, Former Chairman, Dept. of CS & Maths, Bangalore University, Director, R&D, B.U., Bangalore, India
2 Assistant Professor, Dept. of MCA, Dayananda Sagar College of Engineering, Bangalore.
Research Scholar, Bharathiar University, Coimbatore, India

ensemble is the same as that for establishing a committee of people: each member of the committee should be as competent as possible, but the members should be complementary to one another.

Research in ensemble methods has largely revolved around designing ensembles consisting of competent yet complementary models. Actually, the meta-classifier operates in two phases. The first is the training phase during which the system is trained on known data for the problem. Additional parameter adaptation is embedded in the training phase, which enables the system to select its parameters on its own and thus works autonomously without any intervention. Moreover, this feature allows the system to work properly for different medical diagnosis problems in a dynamic way. After training, the main working phase follows, during which the system operates for the classification of new unlabeled data.

The main purpose of an ensemble methodology is to combine a set of models, each of which solves the same original task, in order to obtain a better composite global model with more accurate and reliable estimates or decisions than can be obtained from using a single model. One of the most active areas of research in supervised learning has been to study methods for constructing good ensembles of classifiers. The main discovery is that the ensemble classifier constructed by ensemble machine-learning algorithms, such as bagging and boosting approaches, often performs much better than the single classifiers that make them up. The idea of ensemble methodology is to build a predictive model by integrating multiple models. It is well known that ensemble methods can be used for improving prediction performance. There are several factors that differentiate the various ensembles methods. The main factors are:

1. Inter-classifiers relationship—How does each classifier affect the other classifiers? The ensemble methods can be divided into two main types: sequential and concurrent.
2. Combining method—The strategy of combining the classifiers generated by an induction algorithm. The simplest combiner determines the output solely from the outputs of the individual inducers.
3. Diversity generator—In order to make the ensemble efficient, there should be some sort of diversity between the classifiers. Diversity may be obtained through different presentations of the input data, as in bagging, variations in learner design, or by adding a penalty to the outputs to encourage diversity.
4. Ensemble size—The number of classifiers in the ensemble.

1.1 Ensemble Machine-Learning Methods

Ensemble methods are learning algorithms that construct a set of base classifiers and then classify new data points by taking a vote of their predictions. The aim of ensemble machine learning is to combine a number of rough “rules-of-thumb” into a more accurate aggregate class prediction rule.
Fig. 1 depicts the experimental procedure that we have utilized in this paper.
The learning procedure for ensemble algorithms can be divided into the following two parts (Quinlan, 1996):

1. Constructing base classifiers/base models: the main tasks of this division are (a) data processing: prepare the input training data for building base classifiers by perturbing the original training data, and (b) base classifier constructions: build base classifiers on the perturbed data with a learning algorithm as the base learner.

2. Voting: the second stage of ensemble methods is to combine the base models built in the previous step into the final ensemble model.

There are various kinds of voting systems. Two main voting systems are generally utilized, namely weighted voting and un-weighted voting. In the weighted voting system, each base classifier holds different voting power. On the other hand, in the un-weighted system, individual base classifier has equal weight, and the winner is the one with most number of votes.

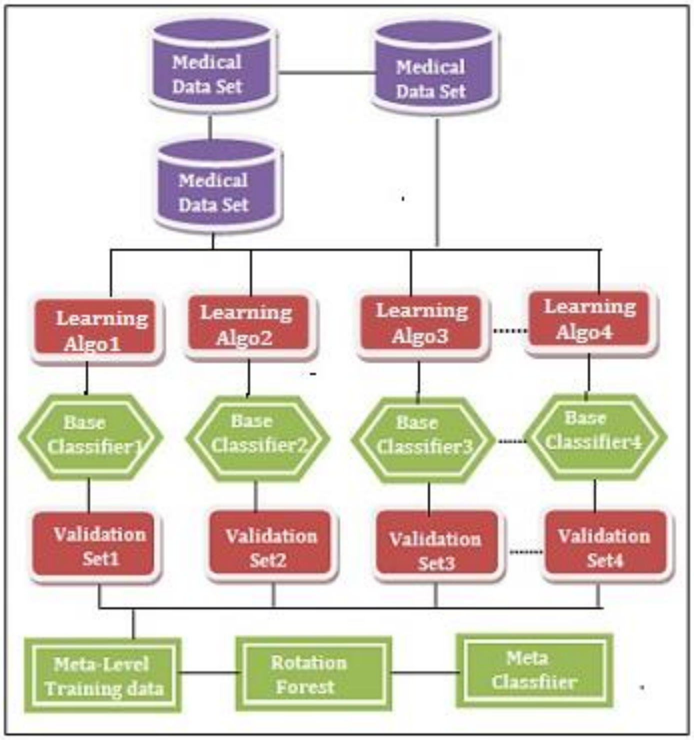

Fig 1. Experimental Procedure for Rotation Forest

1.2 Genesis of Classifier Ensembles

Multiple base models when suitably combined results in ensembles ((Kittler et al., 1998; Kuncheva, 2004; Tumer and Ghosh, 1996) and the final results of the classification strongly depend on the consolidated outputs of the individual models. This clearly suggests that the classification accuracy of the ensembles is excellent only when the accuracies of the individual models are good (Hansen and Salamon, 1992). If the performance of a classifier is better than the random guessing of the test data point class, then it is considered as accurate. In other words, if the errors made by two classifiers on the data points are different, then they are considered to be diverse. The performance of ensembles is better in the case of unstable base models (for eg. Decision trees, neural networks and rule-learning algorithms) whose outputs undergo drastic changes for small changes in the training data.

1.3 The construction methods for Ensembles

Classifier ensembles can be constructed in five popular ways/approaches:

1. In the first approach, the result of changing the distribution of the training data points generates a classifier in the ensemble by utilizing a different sample of the training sets. This approach is generic in nature and works with any classifier. Some of the examples are Bagging (Breiman, 1996) and boosting (Schapire, 2002).
2. In the second approach, the result of changing the attributes in the training set, manipulates the attribute space of the data set. Further, the training of each classifier is done on different attribute sets; which may be newly created attributes or from the training data. Random subspaces (Skurichina and Duin, 2001) and Rotation forests (Rodríguez et al., 2006) are some of the examples belonging to this approach.
3. In the third approach, in order to create diverse data sets, the output of the training is manipulated.
4. This approach (Dietterich and Bakiri, 1995) is extremely useful for multiclass problems and is referred to as "Error-correcting codes approach".
5. This technique is quite popular and is widely used for creating decision tree ensembles where the selection of split attributes and split points is done so that the splitting criterion is optimum.

2. Literature Survey

In the construction of some ensemble methods, mechanisms like Bagging (Breiman, 1996), AdaBoost (Schapire, 2002) and Random subspaces (Skurichina and Duin, 2001) are employed. In some other cases the techniques with different mechanisms are combined in the design of ensemble methods. This can be seen in the following cases: (i) The combination of Bagging with Random subspaces is done through Random forests (ii) The combination of Bagging with AdaBoost is done though MultiBoosting (iii) The combination of randomization in the attribute space division with Bagging is done through Rotation forest (Rodríguez et al., 2006). The special hidden feature of this hybrid ensemble technique is that the combination may outperform either in isolation or in combination with other algorithms whenever the mechanisms for different ensemble methods differ.

Recent researchers reveal that quite a number of methods are available to introduce randomness in the node splitting criterion. Dietterich and Bakiri (1995) propose an approach for randomly selecting a test in the set of K-best splits. Among the K-randomly selected attributes, the split attributes are selected as mentioned above in the case of random forests. Some of the recent works include García-Pedrajas et al. (2012), Marqués et al. (2012), and Srimani and Koti (2011). A thorough survey of the literature pertaining to this topic reveals that no in-depth work is available. Hence, the present work is carried out to throw light on this topic.

3. Data Set Description

We have extracted the datasets (Thyroid, Bupa Liver, Haberman, Hepatitis and Wisconsin) from the UCI repository (Blake and Merz, 1998; Frank and Asuncion, 2010). These data sets are associated with 215, 345, 306, 152 and 699 instances. The Thyroid dataset constitutes 215 instances and 5 attributes and the class attribute (1 = normal, 2 = hyper, 3 = hypo with the class distribution 150, 35 and 30) and the remaining attributes viz., T3-resin uptake test, total serum thyroxin as measured by the isotopic displacement method, total serum triiodothyronine as measured by radioimmuno assay, basal thyroid-stimulating hormone (TSH) as measured by radioimmunoassay and maximal absolute difference of TSH value after injection of 200 micrograms of thyrotropin-releasing hormone as compared to the basal value. In this case, all the attributes are continuous. Five laboratory tests are used to predict whether a patient belongs to the class euthyroidism, hypothyroidism or hyperthyroidism. The diagnosis (the class label) would be based on a complete medical record, including anamnesis, scan etc., with no missing values.

The Bupa Liver dataset has 7 attributes and 345 instances, and the attributes are: mean corpuscular volume, alkaline phosphotase, alamine aminotransferase, aspartate aminotransferase, gamma-glutamyl transpeptidase, number of half-pint equivalents of alcoholic beverages and the selector field is used to split data into two sets. The first 5 variables are all blood tests that are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. Each line in the Bupa data file constitutes the record of a single male individual. Further, one of the selectors is of the form drinks > 5.

The Haberman dataset contains 306 instances and 4 attributes including the class attribute: survival status (1 = the patient survived 5 years or longer and 2 = the patient died within 5 years). This is associated with the age of patient at the time of operation, patient's year of operation and the number of positive axillary nodes detected.

The Hepatitis dataset contains 155 instances and 20 attributes including the class attribute and 20 missing values. Hepatitis is associated with class (die, live) with the class distribution 32 and 123 and the remaining attributes viz., age (10-80), sex, steroid (no, yes), antivirals (no, yes), fatigue (no, yes), malaise (no, yes), anorexia (no, yes), liver big (no, yes), liver firm (no, yes), spleen palpable (no, yes), spiders (no, yes), ascites (no, yes), varices (no, yes), bilirubin (0.39, 0.80, 1.20, 2.00, 3.00, 4.00), alk phosphate. (33, 80, 120, 160, 200, 250), SGOT (13, 100, 200, 300, 400, 500), albumin (2.1, 3.0, 3.8, 4.5, 5.0, 6.0), protime (10, 20, 30, 40, 50, 60, 70, 80, 90) and histology (no, yes).

The Wisconsin dataset has 10 attributes and 699 instances. The attributes are sample code number, clump thickness, uniformity of cell size (1-10), uniformity of cell shape (1-10), marginal adhesion (1-10), single epithelial cell size (1–10), bare nuclei (1–10), bland chromatin (1-10), normal nucleoli (1-10), mitoses (1-10) and class (2 for benign, 4 for malignant, with the class distribution 458 and 241). Further, the dataset is associated with 16 missing attribute values.
The descriptions of the data sets are summarized in Tables 1-5.

Table 1 Features of Thyroid dataset

ID	Attributes	Type
1	T3	Numeric
2	Serum thyroxin	Numeric
3	Serum triodo	Numeric
4	TSH	Numeric
5	Max. abs dif	Numeric
6	Class	Categorical

Table 2 Features of Liver dataset

ID	Attributes	Type
1	MCV	Numeric
2	Alkphos	Numeric
3	Sgpt	Numeric
4	Sgot	Numeric
5	Gammagt	Numeric
6	Drinks	Numeric
7	Selector	Numeric

Table 3 Features of Haberman dataset

ID	Attributes	Type
1	Age	Numeric
2	Year of operation	Numeric
3	Pos. Axillary nodes	Numeric
4	Status	Numeric

Table 4 Features of Wisconsin dataset

ID	Attributes	Type	ID	Attributes	Type
1	Code	Numeric	7	Nuclei	Numeric
2	Clump	Numeric	8	Chromatic	Numeric
3	Size	Numeric	9	Nucleolp	Numeric
4	Shape	Numeric	10	Mitosis	Numeric
5	Marginal	Numeric	11	Class	Numeric
6	Epthiletia	Numeric			

Table 5 Features of hepatitis dataset

ID	Attributes	Type	ID	Attributes	Type
1	Age	Numeric	11	Spiders	Categorical
2	Sex	Categorical	12	Ascites	Categorical
3	Steroid	Categorical	13	Varices	Categorical
4	Antivirals	Categorical	14	Bilirubin	Numeric
5	Fatigue	Categorical	15	Alk Phospate	Numeric
6	Malaise	Categorical	16	Sgot	Numeric
7	Anorexia	Categorical	17	Albumin	Numeric
8	Liver big	Categorical	18	Protime	Numeric
9	Liver firm	Categorical	19	Histology	Numeric
10	Spleen palpable	Categorical	20	Class	Categorical

4. Methodology

Machine learning algorithms play a key role in the design of computer aided diagnosis (CAD) systems. Accordingly, an optimum efficiency of high performance CAD systems could be achieved only by enhancing the accuracies of the associated machine learning algorithms (MLA). From the ensemble literature, it is found the through the application of ensemble classifier strategies it is possible to enhance the performance of a base classifier. The two strategies viz., the application of feature selection methods on the data set under consideration and the construction of ensemble classifiers play a significant role in the prediction of accurate decisions by the classifiers. Filtering approaches and Wrapper methods are the two currently used feature selection strategies. In the latter case, the selection and performance of the classification algorithm in respect of the training data determine the feature subsets; while in the former case, in selecting the best feature subset, there is a strong dependency of filters on the properties of the features. In both the cases, individual ranking, forward and backward search procedures are utilized. It is interesting to note that the correlation-based feature subset selection (CFS) (Hall, 1999) is a widely used (in particular, medical diagnosis application) filter approach of multivariate type that returns the most relevant variable by evaluating the strength of the features (Ozcift, 2011).

Feature selection has been an active and fruitful field of research in pattern recognition, machine learning, statistics and data mining communities. The main objective of feature selection is to choose a subset of input variables by eliminating features that are irrelevant or of no predictive information.

The correlation-based feature subset selection algorithm is a heuristic for evaluating the worth or merit of a subset of features. The usefulness of individual features for predicting the class label along with the level of intercorrelation among them will be used in CFS. We have applied CFS with a best first search algorithm (Table 6) to search the feature subset space in reasonable time. The best first starts with an empty set of features and generates all possible single feature expansions. The subset with the highest evaluation is chosen and expanded in the same manner by adding single features. If expanding a subset results in non-improvement, the search drops back to the next best unexpanded subset and continues from there.

Table 6 Algorithm for BFS

```
Input: A graph $G$ and a root $v$ of G
Procedure BFS($G$, $v$):
    create a queue $Q$
    enqueue $v$ onto $Q$
    mark $v$
    while $Q$ is not empty:
        $t$ ← Q.dequeue()
        if $t$ is what we are looking for:
            return $t$
        for all edges e in G.incidentEdges(t) do
            $o$ ← G.opposite($t$, $e$)
            if $o$ is not marked:
                mark $o$
                enqueue  $o$ onto Q
```

We have used various base classifiers along with ensemble methods to evaluate the performance. The base classifier description is as follows:

4.1 ADTree

An alternating decision tree consists of decision nodes and prediction nodes. Decision nodes specify a predicate condition. Prediction nodes contain a single number. ADTrees always have prediction nodes as both root and leaves. An instance is classified by an ADTree by following all paths for which all decision nodes are true and summing any prediction nodes that are traversed. This is different from binary classification trees such as CART or C4.5 in which an instance follows only one path through the tree.

4.2 BFTree

BFTree is the class for building a best-first decision tree classifier. This class uses a binary split for both nominal and numeric attributes. For missing values, the method of 'fractional' instances is used.

4.3 DecisionStump

DecisionStump is a class for building and using a DecisionStump which is usually used in conjunction with a boosting algorithm. This performs regression (based on mean-squared error) or classification (based on entropy) and missing is treated as a separate value.

4.4 FunctionalTrees (FT)

FunctionalTree is a classifier for building 'functional trees', which are classification trees that could have logistic regression functions at the inner nodes and/or leaves. This algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values.

J48

J48 is a class for generating a pruned or unpruned C4.5 decision tree.

4.5 J48graft

J48graft is a class for generating a grafted (pruned or unpruned) C4.5 decision tree.

4.6 LADTree

LAD Tree is a class for generating a multi-class alternating decision tree using the LogitBoost strategy.

4.7 LMT

LMT is a classifier for building 'logistic model trees', which are classification trees with logistic regression functions at the leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values.

4.8 RandomForest

RandomForest is a classifier for constructing a forest of random trees.

4.9 RandomTree

Random Tree is a classifier for constructing a tree which considers at each node, K randomly chosen attributes, which performs no pruning.

4.10 Naïve Bayes Tree (NB Tree)

A class for a Naive Bayes classifier is estimated using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data.

4.11 REPTree

REPTree is a fast decision tree learner that builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with backfitting). This only sorts values for numeric attributes once and missing values are dealt with by splitting the corresponding instances into pieces (i.e. as in C4.5).

In this context, rotation forest refers to a technique to generate an ensemble of classifiers which performs the training of each base classifier with a different set of extracted attributes. The main heuristic is to apply feature extraction and to subsequently reconstruct a full attribute set for each classifier in the ensemble. To this end, the feature set F is randomly split into L subsets. Principal component analysis (PCA) is run separately on each subset, and a new set of linear extracted attributes is constructed by pooling all principal components. Then, the data are transformed linearly into the new feature space. The classifiers are trained with this data set. Different splits of the feature set will lead to different extracted features, thereby contributing to the diversity introduced by the bootstrap sampling.

In the data, the preservation of the variables of information is done by principal components and the determination of these is done through rotation forest algorithm (Rodriguez et al., 2006) which applies each K set a Principal Component Analysis (PCA) transformation. The formation of new features for base classifiers is accomplished by K-axis rotations. Its main idea is to simultaneously encourage diversity and individual accuracy within an ensemble classifier. Specifically, diversity is promoted by using PCA to do feature extraction for each base classifier and accuracy is sought by keeping all principal components and also using the whole dataset to train each base classifier.

Algorithm: Rotation Forests

```
For $k = 1,\ldots,L$
Take a bootstrap sample $S_k$ from Z of size N.
Build a tree-classifier $D_k$ using $S_k$ as the training set.
Form a new set of extracted features and use this set to build the classifier.
End k
Majority voting: for an unlabeled x, take the votes of the L classifiers and calculate $g_k(x) = \Sigma$ votes
for $\omega_k$ , $k = 1,\ldots, c$.
Pick the class with the largest support
```

The steps involved in rotation forests with feature extraction:
Step 1: Split randomly the feature set into K subsets (Assume K is a factor of the number of features)
Step 2: For each feature subset, apply PCA on the data using only these features and a random sub-sample of the classes
Step 3: Pool all principal components to form a new set of extracted features.

It is noted that no principal components are discarded and applying a PCA is equivalent to rotating the feature axes. K different rotations are carried out to obtain a new set with extracted features.

The three metrics viz., the accuracy of classification (ACC), Kappa error (KE) and the area under the receiver operating characteristic (ROC) curve (AUC) evaluate the performance of the various algorithms when the experiments are carried out with a 10-fold cross-validation. The receiver operating characteristics (ROC) curve is a graphical representation of the tradeoff between the false negative and false positive rates for every possible cutoff. Equivalently, the ROC value is the representation of the tradeoffs between sensitivity and specificity. The diagnosed samples provided by the medical experts will be used for the initial training of these algorithms and these will assist the medical experts in all their future diagnosis events. Already we have seen how the two strategies can improve the ability of prediction of the methods of analysis. Further, it is to be observed that too many features in the process of classification may affect the accuracy of the classification strategies in the negative sense leading to overfitting. In such situations, the initial size of the training samples allows the noise or the irrelevant features to decrease the accuracy drastically.

5. Experiments and Results

The performance evaluation of various machine learning algorithms (MLA) together with their rotation forest algorithm is carried out by implementing the various algorithms using WEKA version 3.6.1.

Table 7 Classification results for Thyroid dataset

Algorithm	Acc (%)	eAcc (%)	Diff (%)	AUC	eAUC	Kappa	eKappa	Time (ss)	eTime (ss)
BFTree	92.1	96.3	4.2	0.907	0.996	0.824	0.918	0.05	0.37
FT	97.2	96.7	0.5	0.998	0.984	0.939	0.930	0.11	0.53
Decision Stump	77.2	84.7	7.5	0.719	0.998	0.435	0.603	0.02	0.08
J48	92.1	97.2	5.1	0.9	0.999	0.829	0.938	0.02	0.11
J48graft	92.6	96.4	3.8	0.863	0.998	0.837	0.917	0.19	0.11
LADTree	94.0	96.7	2.7	0.978	0.998	0.868	0.929	0.08	0.55
LMT	97.7	96.7	1.0	0.998	0.997	0.95	0.930	0.58	6.4
Random Forest	94.9	96.7	1.8	0.985	0.998	0.887	0.928	0.05	0.25
Random Tree	94.0	97.7	3.7	0.92	0.998	0.867	0.949	0	0.08
REPTree	92.1	95.8	3.7	0.905	0.996	0.826	0.906	0.02	0.08

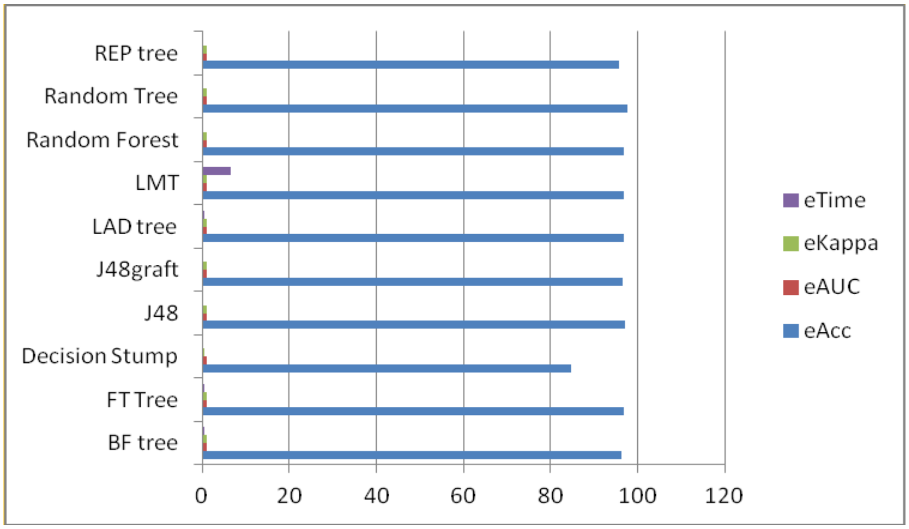

Fig 2. Performance evaluation of the ensemble classifiers for Thyroid data set

Table 7 and Fig. 2 predict that LMT performs well as a base classifier, while all the ensemble classifiers perform in an excellent manner. In particular, RandomTree and J48 have eAcc values of 97.7% and 97.2% with eT=0.08ss and 0.11ss. In Fig. 3, the graph of the difference between the ensemble and base accuracies is presented for the algorithms considered in this analysis. The greater the value of the difference, the greater is the accuracy achieved by the ensemble classifier. Although all the ensemble classifiers perform much better than the base classifiers in general, decision stump is found to have an enhanced value of accuracy (diff: 7.5%). In this case, the percentage of accuracy lies in the range 84.7<eAcc<97.7. Here $(eAUC)_{max}$ = 0.999 (J48) and $(eAUC)_{min}$ = 0.984 (FT).

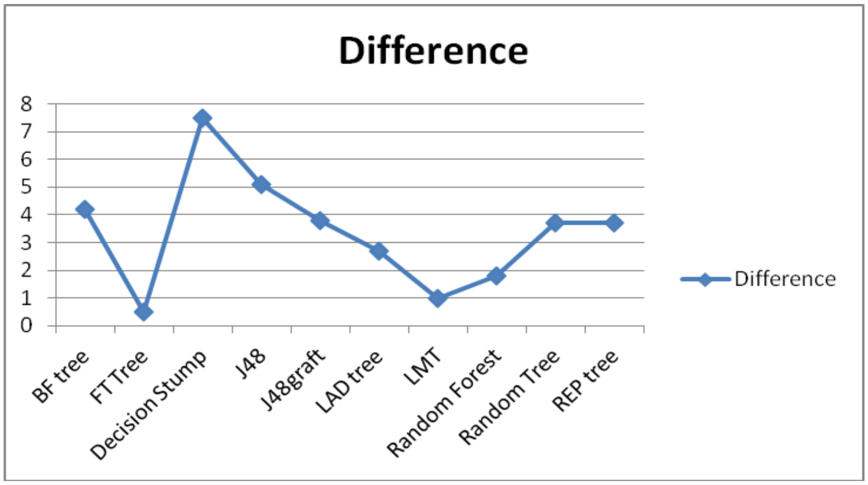

Fig 3. Graph of (eAcc - Acc) for Thyroid data set

Table 8 and Fig. 4 predict that FT (Acc: 75.1%) performs well while REPTree performs well (eAcc: 73%) as an ensemble classifier. In this case, the percentage of accuracy lies in the range 65.8< eAcc <73. In Fig. 5, the graph of the difference between the ensemble and base accuracies is presented for the algorithms considered in this analysis. Although all the ensemble classifiers perform much better than the base classifiers in general, J48graft is found to be a poor performer, while ADTree turns out to be the optimal ensemble classifier. Here $(eAUC)_{max}$ = 0.777 (Random Forest) and $(eAUC)_{min}$ = 0.726 (FT).

Table 8 Classification results for Liver dataset

Algorithm	Acc (%)	eAcc (%)	Diff (%)	AUC	eAUC	Kappa	eKappa	Time (ss)	ETime (ss)
ADTree	59.7	72.5	12.8	0.668	0.776	0.170	0.429	0.06	0.52
BFTree	64.9	72.2	7.3	0.684	0.74	0.268	0.408	0.12	1.2
Decision Stump	57.7	68.7	11.0	0.536	0.734	0.090	0.316	0.02	0.11
FT	75.1	71.3	3.8	0.769	0.726	0.485	0.382	0.16	0.75
J48	68.7	70.7	2.0	0.665	0.739	0.340	0.374	0.03	0.16
J48graft	68.7	70.4	1.7	0.665	0.741	0.340	0.369	0.06	0.17
LADTree	65.5	70.4	4.9	0.701	0.758	0.291	0.371	0.09	0.75
LMT	66.4	70.1	3.7	0.701	0.727	0.284	0.365	1.03	7.24
NBTree	66.1	70.7	4.6	0.651	0.738	0.285	0.367	0.36	0.69
Random Forest	69.0	72.8	3.8	0.738	0.777	0.366	0.422	0.09	0.83
Random Tree	67.8	65.8	2.0	0.676	0.729	0.347	0.298	0.02	0.16
REPTree	64.1	73.0	8.9	0.654	0.748	0.245	0.424	0.03	0.13

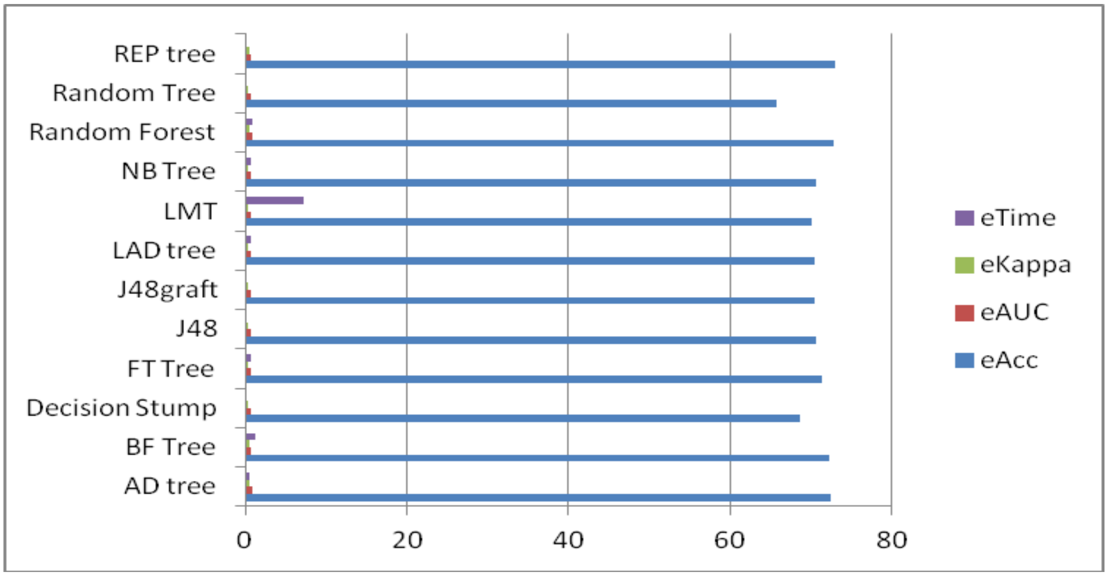

Fig 4. Performance evaluation of the ensemble classifiers for Liver data set

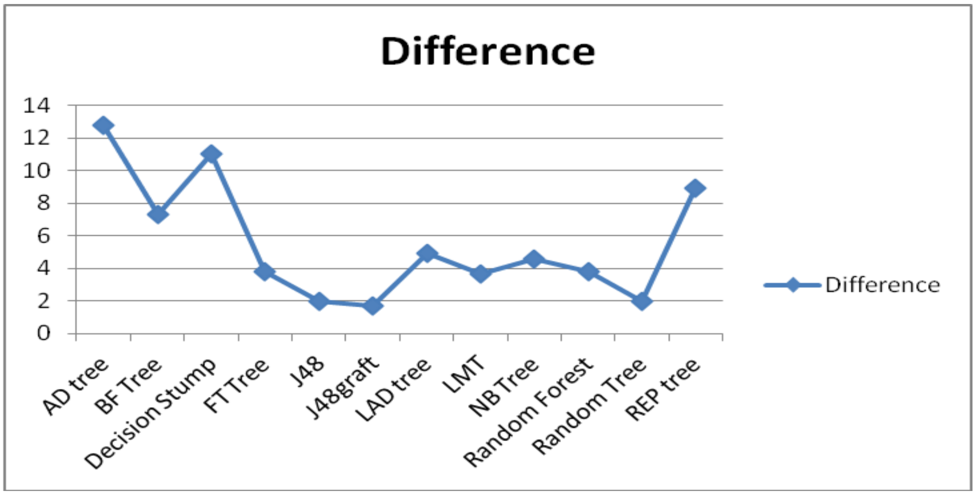

Fig 5. Graph of (eAcc - Acc) for Liver data set

Table 9 and Fig. 6 predict that ADTree is the best base classifier (Acc: 75.2%) while BFTree, DecisionStump, LADTree and NBTree have the same level of performance (Acc: 74.8%). Further, J48graft and NBTree are found to be the ensemble classifiers with eAcc: 75.2%. One of the important observations made is that the performance of base classifier is better than the ensemble classifiers in most of the cases. In Fig. 7, the graph of the difference between the ensemble and base accuracies is presented for the algorithms considered in this analysis. The following observations are made: (i) there is a fluctuation in the performance of the ensemble classifiers on the Haberman dataset as observed earlier, (ii) the difference in the accuracies is found to be maximum in the case of Random Forest and (iii) in the case of BFTree, FT and LADTree, there is no enhancement in the values of the accuracy. In this case, the percentage of accuracy lies in the range 69.6 %< eAcc <75.2%. Here $(eAUC)_{max}$ = 0.699 (LAD) and $(eAUC)_{min}$ = 0.603(FT Tree).

Table 9 Classification results for Haberman dataset

Algorithm	Acc (%)	eAcc (%)	Diff (%)	AUC	eAUC	Kappa	eKappa	Time (ss)	ETime (ss)
ADTree	75.2	74.8	0.4	0.68	0.683	0.241	0.140	0.05	0.22
BFTree	74.8	74.8	0	0.587	0.62	0.159	0.043	0.08	0.47
Decision Stump	74.8	74.2	0.6	0.634	0.652	-0.006	-0.019	0	0.06
FT	74.2	74.2	0	0.609	0.603	0.054	0.042	0.03	0.27
J48	73.9	74.8	0.9	0.491	0.607	0.0003	-0.006	0.02	0.06
J48graft	73.9	75.2	1.3	0.491	0.604	0.0003	0	0	0.08
LADTree	74.8	74.8	0	0.657	0.699	0.211	0.168	0.05	0.28
LMT	74.5	73.9	0.6	0.661	0.635	0.083	0.070	0.25	2.82
NBTree	74.8	75.2	0.4	0.62	0.604	0.043	0	0.05	0.25
Random Forest	72.9	71.2	1.7	0.647	0.664	0.199	0.130	0.05	0.66
Random Tree	69.9	69.6	0.3	0.589	0.645	0.187	0.094	0.02	0.09
REPTree	74.2	74.8	0.6	0.55	0.658	0.054	0.067	0	0.08

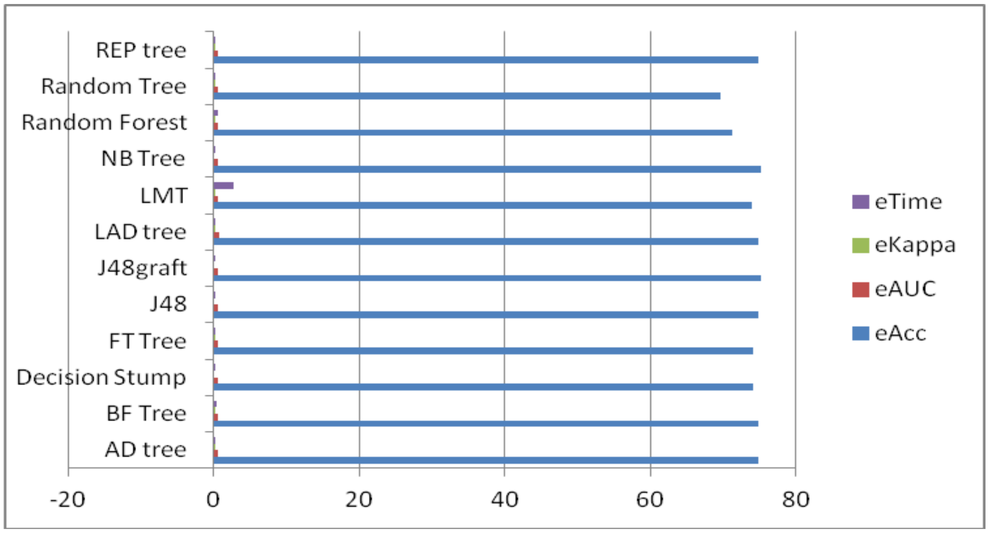

Fig 6. Performance evaluation of the ensemble classifiers for Haberman data set

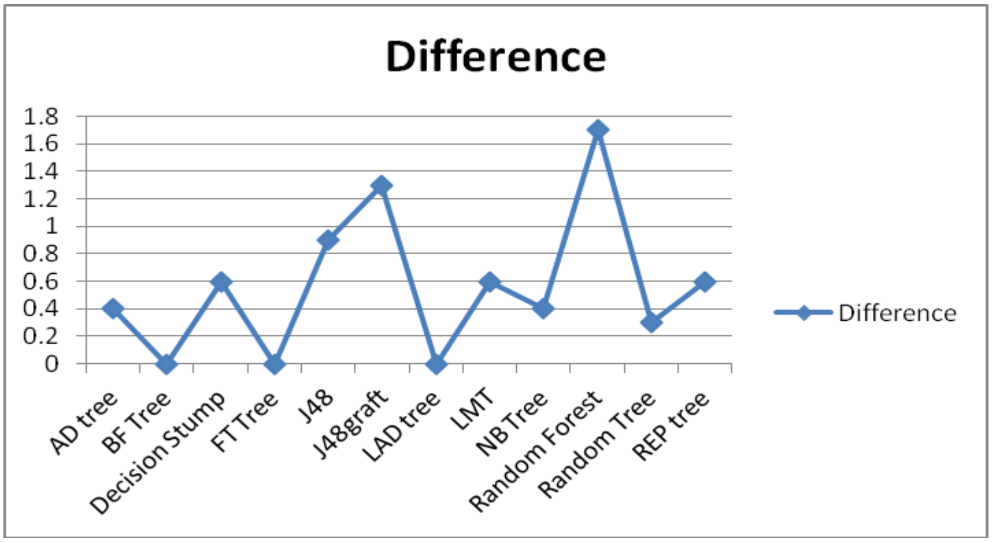

Fig 7. Graph of (eAcc - Acc) for Haberman data set

Table 10 and Fig. 8 predict that all ensemble classifiers perform well when compared to base classifiers. In the case of base classifiers, FT and LAD Tree perform equally well with Acc: 86.8%. However, in the case of ensemble classifiers, J48graft happens to be the best performer with eAcc: 90%. It is interesting to note that (i) there is a drastic enhancement in the accuracy in the case of J48 and J48graft (Diff: 4.5, 4.6) (ii) LMT has the maximum eTime: 18.03ss and (iii) the percentage of eAcc lies in the range 82.1 < eAcc < 90. Fig. 8 clearly predicts the performance of the different ensemble classifiers for the Wisconsin data set. Fig. 9 gives the measure of accuracy improvement enhancement with regard to the different classifiers considered here. Here $(eAUC)_{max}$ = 0.972 (LADTree) and $(eAUC)_{min}$ = 0.859 (DecisionStump).

Table 10 Classification results for Wisconsin dataset

Algorithm	Acc (%)	eAcc (%)	Diff (%)	AUC	eAUC	Kappa	eKappa	Time (ss)	eTime (ss)
BF Tree	86.4	88.1	1.7	0.921	0.963	0.774	0.802	0.16	1.76
Decision Stump	78.4	82.1	3.7	0.804	0.859	0.612	0.678	0.03	0.17
FT	86.8	88.6	1.8	0.942	0.959	0.785	0.807	0.36	3.74
J48	84.8	89.3	4.5	0.915	0.969	0.748	0.818	0.06	0.44
J48graft	85.4	90.0	4.6	0.918	0.97	0.757	0.830	0.08	0.52
LADTree	86.8	89.0	2.2	0.965	0.972	0.784	0.814	0.27	2.47
LMT	86.6	89.0	2.4	0.956	0.971	0.778	0.813	2.09	18.03
NBTree	86.7	87.7	1.0	0.966	0.965	0.784	0.802	0.55	9.19
Random Forest	86.6	89.1	2.5	0.955	0.967	0.779	0.818	0.14	1.22
Random Tree	85.0	87.8	2.8	0.875	0.95	0.749	0.798	0.03	0.27
REPTree	85.7	88.0	2.3	0.93	0.965	0.765	0.798	0.05	0.27

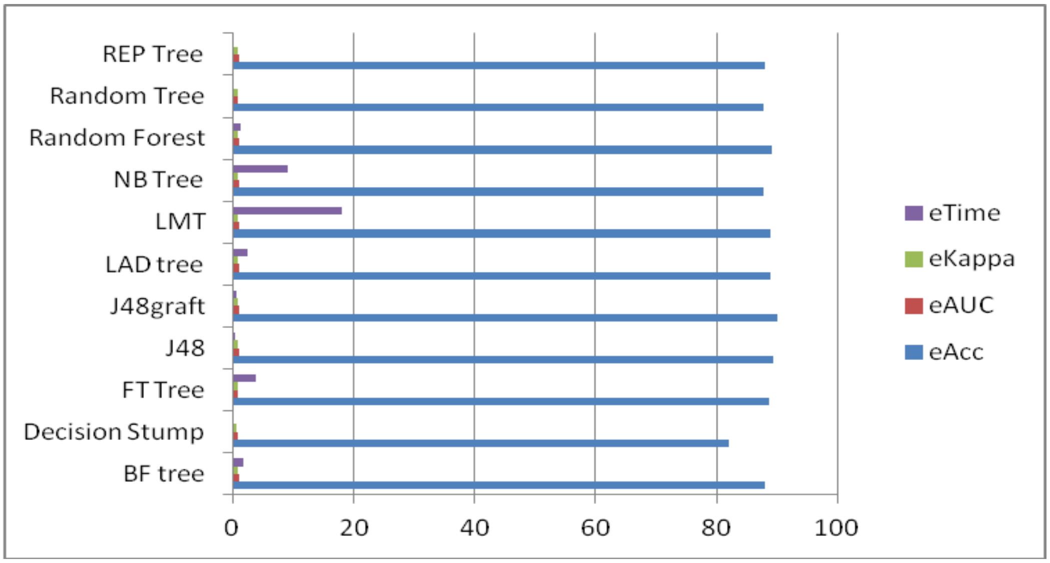

Fig 8. Performance evaluation of the ensemble classifiers for Wisconsin data set

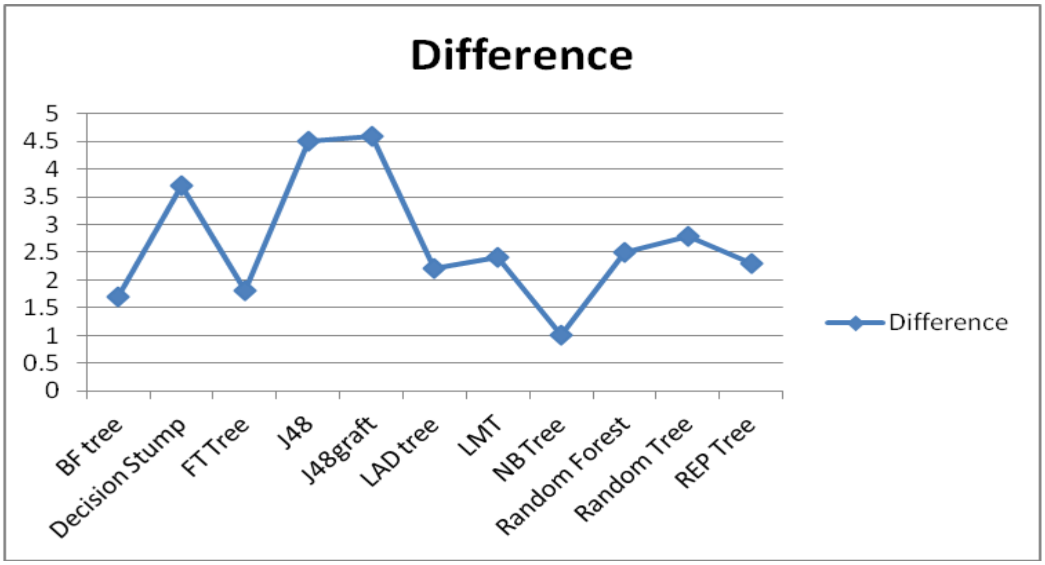

Fig 9. Graph of (eAcc - Acc) for Wisconsin data set

Table 11 and Fig. 10, predict that the accuracy and the performance efficiency is enhanced in the case of all ensemble classifiers except for J48 and J48graft with (Acc, eAcc) = (82.9%, 82.2%) and (83.6%, 82.2%). In this case, the percentage of accuracy lies in the range 80.3% < eAcc < 85.5%. In Fig. 11, Random Tree has the maximum difference in the accuracies (Diff: 7.9%) and LMT has the maximum eTime: 2.76ss. Here $(eAUC)_{max}$ = 0.87(LMT) and $(eAUC)_{min}$ = 0.796 (BFTree).

Table 11 Classification results for Hepatitis dataset

Algorithm	Acc (%)	eAcc (%)	Diff (%)	AUC	eAUC	Kappa	eKappa	Time (ss)	ETime (ss)
ADTree	78.3	82.2	3.9	0.821	0.854	0.323	0.432	0.03	0.38
BFTree	78.9	83.6	4.7	0.674	0.796	0.263	0.416	0.11	0.28
Decision Stump	79.6	80.3	0.7	0.726	0.823	0	0.116	0.02	0.06
J48	82.9	82.2	0.7	0.658	0.804	0.417	0.386	0.06	0.08
J48graft	83.6	82.2	1.4	0.663	0.814	0.432	0.386	0.02	0.13
LADTree	78.3	80.3	2.0	0.774	0.823	0.306	0.361	0.06	0.36
LMT	83.6	85.5	1.9	0.85	0.87	0.461	0.519	0.67	2.76
NBTree	82.9	84.9	2.0	0.849	0.858	0.473	0.477	0.41	1.59
Random Forest	80.9	84.9	4.0	0.802	0.823	0.358	0.463	0.05	0.23
Random Tree	74.3	82.2	7.9	0.635	0.782	0.254	0.418	0	0.08
REPTree	78.3	83.6	5.3	0.66	0.83	0.044	0.382	0.02	0.08

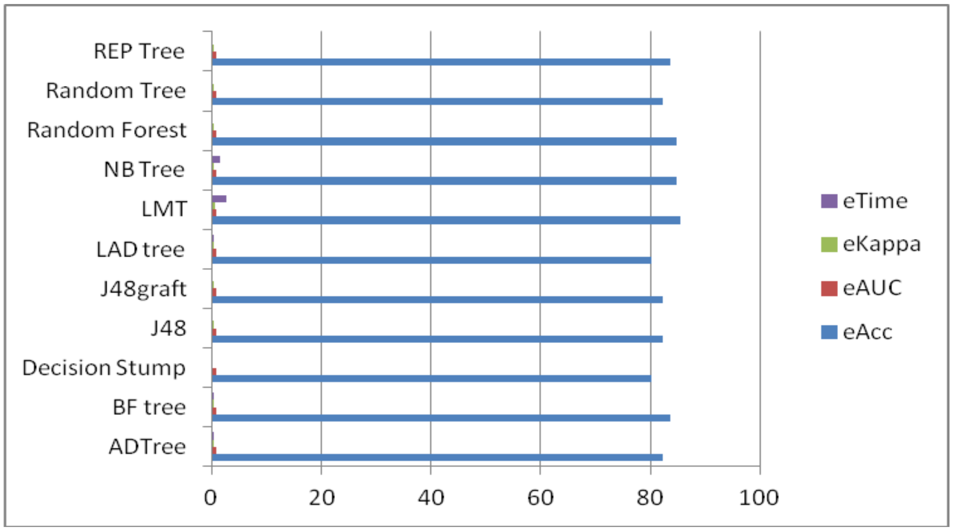

Fig 10. Performance evaluation of the ensemble classifiers for Hepatitis data set

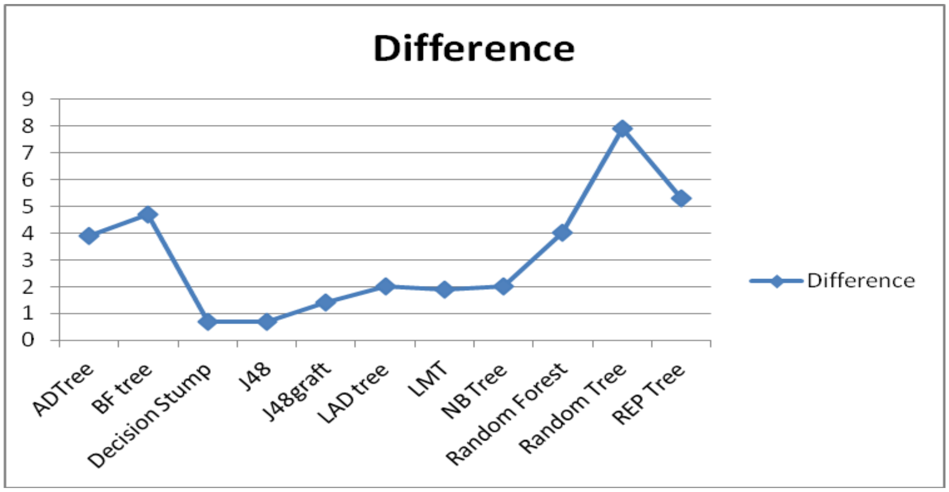

Fig 11. Graph of (eAcc - Acc) for Hepatitis data set

Table 12 Table of max. eAcc and eAUC

Data Set	eAcc	AUC
Thyroid	RandomTree (97.7%)	0.998
Liver	REPTree (73%)	0.748
Haberman	J48graft, NBTree (75.3%)	0.491 0.62
Hepatitis	LMT (85.5%)	0.87
Wisconsin	J48graft (90%)	0.918

It should be noted that the model is perfect if its area under the curve is equal to 1, while the model performs random guessing if its area under the curve is equal to 0.5.

6. Conclusion

Machine learning algorithms play a key role in the design of computer aided diagnosis (CAD) systems. Accordingly, an optimum efficiency of high performance CAD systems could be achieved only by enhancing the accuracies of the associated machine learning algorithms (MLA). From the ensemble literature, it is found the through the application of ensemble classifier strategies it is possible to enhance the performance of a base classifier. The present investigation was undertaken in order to substantiate/justify the performance of ensemble methods in medical data sets, which is of vital interest in making effective diagnosis, which in turn would increase the health index. Our study was conducted on Thyroid, Liver, Haberman, Wisconsin and Hepatitis datasets. The detailed experimental results are presented in Tables 7 to 11. It is concluded that (i) base classifier performance is certainly very much enhanced by using the different ensemble strategies (ii) selected classifiers for are to be used for each dataset and (iii) in some specified cases, ensemble classifiers need not be proposed. In other words, one cannot generalize that ensemble classifiers do improve/enhance the classification accuracy in all the cases. For example (i) in the case of the Thyroid dataset, Acc: 97.7% and eAcc: 96.7% for the LMT classifier (ii) in the case of Hepatitis dataset, Acc: 83.6% and eAcc: 85.5% for LMT classifier and (iii) in the case of Haberman data set Acc = eAcc = 74.8 for BFTree and LAD classifier. Therefore, one has to make a proper selection of the classifier in order to achieve the optimal classification accuracy with regard to the medical data set.

Acknowledgement

Mrs. Manjula Sanjay Koti is grateful to Bharathiar University, Tamil Nadu and Dayananda Sagar College of Engineering, Bangalore for providing the facilities to carry out the research work.

References

Blake C. L. and Merz C. J., 1998. UCI Repository of Machine Databases, Learning http://www.ics.uci.edu/mlearnMLRepository.html.

Breiman. L, 1996. Bagging predictors. Machine Learning 24, 123-140. http://dx.doi.org/10.1007/BF00058655

Dietterich, T. G., 2000. Ensemble methods in machine learning. First International workshop on Multiple Classifier systems, 1-15.

Dietterich, T.G. and Bakiri. G, 1995. Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence 2.

Frank, A. and Asuncion, A. 2010. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] Irvine, CA: University of California, School of Information and Computer Science.

García-Pedrajas, N., Maudes-Raedo, J., García-Osorio, C., Rodríguez-Díez, J. J., 2012. Supervised subspace projections for constructing ensembles of classifiers. Science Direct, Information Sciences 193 1–21. http://dx.doi.org/10.1016/j.ins.2011.06.023

Hansen, L. K. and Salamon, P., 1992. Neural Network Ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(10) 993–1001. http://dx.doi.org/10.1109/34.58871

Hall M., 1999. Correlation-based Feature Selection for Machine Learning. Ph.D. thesis, Dept. of CS, University of Waikato, New Zealand, pp. 51-74.

Kittler J., Hate, M., Duin, R. P. W. and Matas, J., 1998. On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3) 226–239.
http://dx.doi.org/10.1109/34.667881

Kuncheva, L. I., 2004. Combining Pattern Classifiers: Methods and Algorithms. Wiley-InterScience.
http://dx.doi.org/10.1002/0471660264

Marqués, A. I., García, V., Sánchez, J. S., 2012. Two-level classifier ensembles for credit risk assessment. Expert Systems with Applications 39, 10916–10922
http://dx.doi.org/10.1016/j.eswa.2012.03.033

Ozcift, A., 2011. Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms, Computer Methods and Programs in Biomedicine, 443-451.

Quinlan, J. R., 1996. Bagging, boosting, and C4.5. Proceedings of the Thirteenth National Conference on Artificial Intelligence: 725-730.

Rodríguez, J. J., Kuncheva, L. I., Alonso, C. J., 2006. Rotation forest: a new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) 1619–1630.
http://dx.doi.org/10.1109/TPAMI.2006.211
PMid:16986543

Rokach, L., 2009. Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. Science Direct-Computational Statistics and Data Analysis 53, 4046–4072.
http://dx.doi.org/10.1016/j.csda.2009.07.017

Schapire, R., 2002. The boosting approach to machine learning: An overview. MSRI workshop on Nonlinear Estimation and Classification.

Skurichina, M., Duin, R. P. W., 2001. Bagging and the random subspace method for redundant feature spaces, in: J. Kittler, R. Poli (Eds.), Proceedings of the Second International Workshop on Multiple Classifier Systems MCS, Cambridge, UK, pp. 1–10.

Srimani, P. K. and Koti, M. S., 2011. A comparison of different learning models used in data mining for medical data. The Smithsonian/NASA Astrophysics Data System, AIP Conf. Proc. 1414, 51-55; doi: 10.1063/1.3669930.
http://dx.doi.org/10.1063/1.3669930

Tumer K. and Ghosh J., 1996. Error Correlation and Error Reduction in Ensemble Classifiers. Connect. Sci. 8(3) 385–404.
http://dx.doi.org/10.1080/095400996116839
www.cs.man.ac.uk