Intelligent Decision Technologies 16 (2022) 315–324
DOI 10.3233/IDT-210074
IOS Press

Rotation forest of random subspace models

Stamatios-Aggelos N. Alexandropoulos[b,*], Christos K. Aridas[a,*], Sotiris B. Kotsiantis[a], George A. Gravvanis[b] and Michael N. Vrahatis[a]
[a]*Computational Intelligence Laboratory, Department of Mathematics, University of Patras, Patras, Greece*
[b]*Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece*

Abstract. During the last decade, a variety of ensembles methods has been developed. All known and widely used methods of this category produce and combine different learners utilizing the same algorithm as the basic classifiers. In the present study, we use two well-known approaches, namely, Rotation Forest and Random Subspace, in order to increase the effectiveness of a single learning algorithm. We have conducted experiments with other well-known ensemble methods, with 25 sub-classifiers, in order to test the proposed model. The experimental study that we have conducted is based on 35 various datasets. According to the Friedman test, the Rotation Forest of Random Subspace C4.5 (RFRS C4.5) and the PART (RFRS PART) algorithms exhibit the best scores in our resulting ranking. Our results have shown that the proposed method exhibits competitive performance and better accuracy in most of the cases.

Keywords: Ensembles of classifiers, rotation forest, random subspace, machine learning, data mining, classification

1. Introduction

In general, effective models and algorithms are required in order to tackle knowledge extraction for big data analytics. For producing a reliable and efficient machine learning method, empirical studies have shown that we should focus on the overall problem field instead of a specific problem. The theoretical work carried out by Wolpert and Macready [1] indicates that there is no single algorithm that outperforms all the other algorithms regarding all problems. A recent study [2] based on the work of Wolpert and Macready indicates that a machine learning algorithm is not performing equally well on all problems. For a further study in these issues we refer the interested readers to [3].

We have to take into consideration that there could exist a given learning algorithm that outperforms all the other ones under certain conditions in specific real-world problems. Thus in a particular subset of input data, we are able to conclude that it may be possible a learning algorithm to outperform other ones in a specific set of problems. On the other hand, it is unusual to achieve the best performance for all problems [4].

Based on the above, various researchers focus their attempts on building multiple learner systems, such as an ensemble of classifiers, in order to produce a model which takes advantage of the behaviour of different base-classifiers. This issue results in creating an inductive system with high accuracy and reliability. Another essential advantage of such a system is that even if a learning algorithm fails, it does not mean that the overall system fails.

The main concept concerning an ensemble of classifiers is the combination of individual decisions using in some way different classifiers [5]. In the last decade, their use has been widespread, and for this reason, many different methods of ensembles have been introduced [4]. The main issues for creating an ensemble of classifiers are the following:

(i) A different subset of training data with a unique learning algorithm can be used.
(ii) The usage of different learning models is suggested.
(iii) Several training variables concerning an individual learning algorithm can be used.

*Corresponding authors: Stamatios-Aggelos N. Alexandropoulos, Department of Electrical and Computer Engineering, Democritus University of Thrace, GR-67100 Xanthi, Greece, E-mail: stalexan@ee.duth.gr. Christos K. Aridas, Computational Intelligence Laboratory (CILab), Department of Mathematics, University of Patras, GR-26110 Patras, Greece. E-mail: char@upatras.gr.

ISSN 1872-4981/$35.00

An informative review in comparing ensemble methods can be found in [4]. In addition in [4], the design and application of multiple classifier systems are presented. However, the following issue needs to be considered. In a plethora of methods, the appropriate selection could be a difficult task. Nevertheless, the effectiveness of an ensemble method is based on the following issues:

(i) There are essential representational factors.
(ii) Computational issues give an advantage to an ensemble.
(iii) Statistical aspects.

Thus, an ensemble of classifiers is preferable than a single one [4]. A critical fact that affects the success of a multiple learner system is the diversity of the participants. If the classifiers participating in the combination have different characteristics, then possibly the predictions will be stable and more accurate. More specifically, although all learners may misclassify instances, it is important to notice that learners misclassify different instances.

Ensemble methods have been of great interest, as indicated by the growing number of the applications that have been presented in recent studies [6,7]. Let us address a critical problem belonging to the field of medicine. This problem is related to breast cancer, a disease that plagues women all over the globe. As it is easily understood, such issue is significant to be forecasted in time for the patient to receive appropriate treatment. In conclusion, an accurate diagnosis of this disease is the first and perhaps an essential step in providing the most appropriate treatment. During the last years, there are various developed techniques regarding this issue, that are adopted by experts. Aličković and Subasi in [6] have provided a comparison between data mining techniques for the prognosis of breast cancer.

In addition, there is no doubt that the technologies which use Android platforms are increasing. Indicatively, we mention the number of applications found on the internet that serve market trends. A common issue regarding such platforms and applications is the attacks that can be received by malware. To address such issues, Zhu et al. [7] proposed a highly effective and, at the same time, a low-cost method that protects the users from malware attacks. Specifically, using the well-known and widely used Rotation Forest method, Zhu et al. [7] have considered sensitive addresses or suspicious systems in order to detect malware.

In the paper at hand, we combine two well-known methods, namely the Rotation Forest [7,8] and the Random-Subspace method [9] in order to produce an effective single model. For testing our model, we performed comparisons with other well-known ensemble methods, including Bagging [10], Boosting [11], Decorate [12], Random-Subspace method as well as widely used ensemble methods that have been presented in [13]. In most cases, our experiments were based on standard benchmark problems and the experimental results have shown that the provided method has been stable and accurate. Furthermore, widely used methods, such as Decision Trees [14] and Rule Learners [15] have been included in the experiments.

In Section 2, we discuss the most common algorithms used in the literature to create an ensemble of classifiers that are based on a single learning method. In Section 3, the proposed ensemble method is presented. The experiments and the comparison of our approach to well-known methods of this category are presented in Section 4. Finally, in Section 5 we give some concluding remarks and a discussion for further research.

2. Ensemble methods: Background material and well-known techniques

In this section, we briefly mention the fundamental characteristics of ensemble methods, as well as useful information about the main approach and the essential aspects concerning the ensemble of classifiers.

Let us denote by P a given *problem*, by A a *learner* or *learning algorithm*, and by H a *searching* or *hypothesis* space. The main goal of a classifier is the appropriate connection between these elements. Consequently, the algorithm A aims to the best solution in the space H. A difficult issue that we face in many real-world problems is when the available data are restricted. In addition, various real-world data sets may have several problems, such as incomplete instances. In such cases, the learner A should reach diverse solutions in H that could achieve efficient results. Hence, we have the opportunity to select the most suitable set of solutions in respect of accuracy and low capacity.

During the last years, various researchers focus their attention on ensemble methods. The advantage of this approach is the combination of multiple classification algorithms in order to build a single more robust, effective and efficient model instead of the usage of a single classification algorithm. A difficult issue that has to be tackled is the selection of a single strong classifier and an ensemble method. In order to achieve a good selection, we have to examine several issues. For this purpose, we present the basic characteristics of ensemble methodology and how an ensemble of classifiers can be created. It is worth noting that most of ensemble methods are based on varying the given data in some way, and hence techniques of varying the dataset are quite useful [16,17]. Thus, allow us to briefly present *sampling* techniques, *distortion* techniques and *adaptive re-sampling* techniques.

One of the well-known algorithms of this category is the *Bagging* (*Bootstrap aggregating*) algorithm [10]. The aim of this algorithm is to improve a classification task through different classifications using random training samples. Specifically, if T is an initial training set of size t, then the Bagging method produces T_i new training sets of size t'. These sets T_i are produced from the Bagging training set T by a uniform distribution with replacement. Thus, some elements may be repeated in T_i. For each set T_i, a learner L_i is provided. The final, strong learner L_s is introduced through the aggregation of i learners. As a result, an element x is classified through a voting procedure created uniformly by the $L_1, L_2, \ldots, L_i$. Specifically, x is classified into that class which is most voted by L_i.

Boosting [11] is also a well-known method quite similar to the bagging one, but on the other hand it exhibits a significant difference. Specifically, for the *bootstrap* method the assignment of weights takes place such as to create a new strategy sequentially in contrast to the Bagging method. Thus, some elements may be participants of more data sets according to their weight. In each iteration, misclassified instances gain a higher weight and examples that classify correctly lost weight. It is worth noting that *Boosting* method may lead to over-fitting. On the other hand, ***Dagging*** reduces this issue in most of the cases [18].

In order to tackle the usage of the full attribute set to train the classifier, the method that has been proposed in [9] uses random sub-samples of features, i.e. the well-known *Feature Bagging* or *Random Subspace* method. Assume that, the initial training set consists of n instances and that f is the number of the features. Assume further that, m is the number of participants in the ensemble scheme. Then, for each participant, a number $f' < f$ is selected as input data. Also it is recommended to choose an amount of 50% of the original features and to use the same number f' for all the participants. In addition, the ensemble method combines the predictions of all m individuals through a majority voting process.

In [19] an improvement of *AdaBoost* method (which is the most well-known Boosting method) has been provided and a way for improving parameters setting has been presented. In addition, a refined criterion for training weak hypotheses has been performed. Hence, the authors of [19] have concluded the following: (a) when the training data are "poor", then the base learner does not respond well, and so, the AdaBoost will fail, and (b) when the training errors are highly grown, then the AdaBoost will probably also fail.

Webb has presented in [20] a variation of *AdaBoost*. The algorithm, named *MultiBoosting*, is a combination of the AdaBoost method with *Wagging*, which is a variation of the Bagging method. The main difference between Bagging and Wagging is the re-weighting of each training point used in Wagging in order to achieve differently the outcome of Bagging.

Kuncheva et al. [21] have used the *nearest mean classifier* and the *pseudo-Fisher linear discriminant classifier* to test the applicability and the efficiency of well-known ensemble methods, such as Bagging and Boosting. The main goal was to test the usefulness of diverse ensembles. Their outcomes have shown that Boosting succeeds in inducing diversity even for stable linear classifiers whereas Bagging does not.

Another well-known method of ensembles is the *Random Forests* method. This method is presented in combination with the *Random Subspace* method that has been proposed in [22]. In particular, the random forests method uses two other known techniques: the *Decision Trees* approach and the *Feature Bagging* method. It should be mentioned that, random subsamples of the training data are selected as well as random features are used for learning the base classifiers. Consequently, the algorithm selects at random a set of feature data in each step of the decision tree creation and hence, the best tree is established.

An alternative approach that uses a robust classifier as a meta-classifier in order to build a model with diversity has been presented in [12]. Specifically, the authors Melville and Mooney in [12] have proposed a new meta-learner that uses a learner with high accuracy in the training data in order to create a final model with high classification ability. Their method is called *DECORATE* (Diverse Ensemble Creation by Oppositional Re-labeling of Artificial Training Examples). During *DECORATE* execution, the following process is performed: during the participation of a new learner in the set of the classifiers, different constructed examples are randomly added to the training set. The aim was to create a model where its participants would be more diverse. To this end, these newly constructed examples are labelled as conflicting with the current classification. As a result, when a new entrant is added to the classification scheme, the diversity is instantly increased as it is trained in the new training set. Then, the sub-learner is added to the new algorithm scheme.

A popular method belonging to the class of ensemble methods is the so-called *Rotation Forest* [8]. This method is based on feature extraction and uses *Principal Component Analysis (PCA)* [23] on a subset of classes. Specifically, the set of features is divided at random into n subsets, where n is a factor of the number of features. Each subset of features consists of a number of selected features. Let us denote by S_{ij} the j-th feature subset given for training to the L_i learner. The concept of the rotation method is to motivate participant accuracy and diversity simultaneously into the ensemble. The first issue is sought by maintaining all principal components and using the whole training set to train each base learner. The second one is performed using feature extraction.

Another approach that uses the well-known method of Rotation Forest was proposed in [24]. This method aims to predict landslides in an area using the Geographic Information System (GIS) [25]. The proposed method has been named *Rotation Forest fuzzy rule-based Classifier Ensemble (RFCE)*. At this point, let us briefly present some interesting issues regarding this method. Mainly, it is a hybrid model that combines tested techniques for predicting landslides in a specific region of India. Thus, the authors encounter the hybrid method of rotation forest and fuzzy unordered rules induction algorithm classifier. In this research, 930 related landslide sites as well as 15 factors that are proven to occur in a landslide have been considered. In order to test the performance of the proposed hybrid method, the authors conducted tests and comparisons with a set of the most well-known and widely used ensemble methods. The obtained results have shown that the proposed model is competitive and can provide a reliable prediction of landslides based on statistical estimates with indicators.

Randomness can increase the model diversity, but might reduce the individual accuracy of the base classifiers. Blaser and Fryzlewicz in [26] have proposed a method that controls the diversity of ensemble classifiers so that the accuracy of the basic classifiers does not fall significantly. In particular, the feature space of the data set is randomly rotated before passed to the base classifiers in order to make predictions. The authors have pointed out that the improvement in the forecasts of the well-known ensemble method is crucial.

3. Proposed method

It is essential to estimate the expected error of a learning algorithm for different target functions and different training sets. Generalization ability of the learning algorithm is a desirable goal. Over-fitting on the training data set is not desirable and thus, the following three key features should be considered:

1. How well responds the classifier produced by the learning algorithm regarding the given target function.
2. The minimum classification error associated with the well-known Bayes algorithm for the target function.
3. The discrepancy between the decisions of the classifiers.

Rotation Forest [8] is an established ensemble method where each base classifier is trained in a set of data which is formed by the PCA technique. Particularly, with the usage of PCA, the initial axes of the data set features are rotated. In order to structure the training set, that the base classifiers of the ensemble will use, the feature set is randomly divided into N subsets. The PCA technique is applied to each subset. All principal components are preserved to maintain the variability information in the set of data. Therefore, there are as many axis rotations as the subsets of data in which the feature set is separated. The new data sets are provided in order to train the base classifiers of the ensemble scheme. The rotation forest method has the advantage of offering diversity in the ensemble of the classifiers and, at the same time, maintains the individual accuracy at a high level. During the rotation forest method, the so-called feature extraction process takes place, which enhances the diversity. Simultaneously, as all the principal components are kept intact, the accuracy does not fall. Thus, the training set is maintained for each classifier, and consequently no useful information is lost.

In order to improve the performance of the proposed method, the Rotation Forest method is combined with the Random Subspace method [9]. This combination creates a more robust final ensemble approach, which is presented in Algorithm 1. Specifically, we used five sub-classifiers for each sub-ensemble method.

At this point, we discuss the reason for which the proposed ensemble method is effective. First of all, a good reason is the good representation that it offers. The hypothesis space may not contain the actual function. However, it will have good approximations. In this way, classifiers outside the hypothesis space may be represented using the above good approximations. Then, a voting scheme takes place. The majority of the votes provide the opportunity to eliminate the issue of randomness. In addition, when different subspaces of features are used, we have an increase in the diversity of classification decisions that is desirable for our model.

Algorithm 1 : Rotation Forest of Random Subspace (RFRS) algorithm

Require: Learning set A, where $A = 5$ the number of bootstrap samples
Ensure: Learning Algorithm (LA) outputs C^* classifier
for $i = 1$ to A **do**
Group the initial variables at random.
For each group of input variables consider the following:
Regard a data set formed by this initial variables and the whole set of examples.
Delete from the dataset all the instances from an appropriate subset of the classes.
Delete from the dataset a subset of the examples.
Apply PCA technique using the remaining data set.
Regard the PCA's components as a new set of variables: None of the components is rejected.
Let T_i be the training data set using as new variables the components of PCA technique for each group
for $j = 1$ to A **do**
Let T_j be a random projection from the $d-$dimensional input space of T_i to a $k-$dimensional subspace;
Let $C_{A(i-1)+j}$ be a LA (T_j) for the generation of a base classifier
end for
end for
Output: The classifier

$$C^* = \arg\max \sum_{i=1}^{A^2} C_i(x) = y$$

4. Experimental results and comparisons

In this section, we present in detail the datasets, the experiments and the comparisons with other methods that we have considered. We have considered well-known and widely used datasets from UCI repository [27] corresponding to real-world data that possess diversity in terms of characteristics. Also, we have used datasets from different and varied domains, such as the pattern and image recognition, the medical diagnosis and commodity trading. In addition, we have also considered datasets belonging to the field of music composition and computer games among others.

In addition, we provide specific details for the datasets that we have used concerning the number of classes and instances and the type of attributes in Table 1. Moreover, regarding the accuracy of the classifiers, we divide the training set into ten mutually exclusive and equal-sized subsets. After that, for every subset, the classifier is trained on the union of all remaining datasets. Furthermore, the technique of cross-validation is applied for each of the algorithms. Specifically, we have carried out a ten-time cross-validation for every algorithm, taking into consideration their average value.

For some of the considered methods, the reduction in error seems to have arisen after 10 to 15 classifiers. Specifically, this issue occurs to Bagging, Boosting, Decorate and Random-Subspace methods. Nevertheless, the AdaBoost method persists in considerably im-

Table 1
Description of the datasets, where "Inst." denotes the number of the instances, "Cat.f." declares the number of categorical features, "Num.f." denotes the number of numerical features on the dataset and "Clas." indicates the number of the classes

Dataset	Inst.	Cat.f.	Num.f.	Clas.
Anneal	898	32	6	6
Audiology	226	69	0	24
Autos	205	10	15	6
Breast-cancer	286	9	0	2
Breast-w	699	0	9	2
Colic	368	15	7	2
Credit-g	1000	13	7	2
Diabetes	768	0	8	2
Dimin	3949	12	0	5
Haberman	306	0	3	2
Heart-c	303	7	6	5
Heart-h	294	7	6	5
Heart-statlog	270	0	13	2
Hepatitis	155	13	6	2
Hypothyroid	3772	22	7	4
Ionosphere	351	34	0	2
Iris	150	0	4	3
Kr-vs-kp	3196	35	0	2
Letter	20000	0	16	26
Lymphotherapy	148	15	3	4
Monk1	124	6	0	2
Monk2	169	6	0	2
Mushroom	8124	22	0	2
Primary-tumor	339	17	0	21
Segment	2310	0	19	7
Sick	3772	22	7	2
Sonar	208	0	60	2
Soybean	683	35	0	19
Spambase	4601	0	58	2
Student	344	11	0	2
Titanic	2201	3	0	2
Vote	435	16	0	2
Vowel	990	3	10	11
Waveform	5000	0	40	3
Zoo	101	16	1	7

Table 2
Friedman ranking (using C4.5)

Rank	Algorithm
2.72353	RFRS C4.5
2.96765	Rotation Forest C4.5
3.95588	MultiBoost C4.5
4.29412	Boosting C4.5
4.76471	Random Subspace C4.5
5.16176	Bagging C4.5
5.50000	Decorate C4.5
6.63235	Dagging C4.5

Table 3
Finner Post-hoc (using RFRS C4.5 as control method)

Comparison	Statistic	Adjusted p-value	Result
RFRS C4.5 vs Dagging C4.5	6.41121	0.00000	H0 is rejected
RFRS C4.5 vs Decorate C4.5	4.50517	0.00002	H0 is rejected
RFRS C4.5 vs Bagging C4.5	3.93584	0.00019	H0 is rejected
RFRS C4.5 vs Random Subspace C4.5	3.26749	0.00190	H0 is rejected
RFRS C4.5 vs Boosting C4.5	2.47537	0.01858	H0 is rejected
RFRS C4.5 vs MultiBoost C4.5	1.90603	0.06577	H0 is accepted
RFRS C4.5 vs Rotation Forest C4.5	0.07426	0.94080	H0 is accepted

Table 4
Comparisons of the proposed ensemble method with established ensemble methods using the C4.5 classifier as base classifier, where "MultiB." denotes the MultiBoost algorithm, "Rot.Forest" declares the Rotation Forest method and "Rand.-Sub." indicates the Random-Subspace algorithm

Dataset	RFRS C4.5	Bagging C4.5	Dagging C4.5	Boosting C4.5	MultiB. C4.5	Decorate C4.5	Rot.Forest C4.5	Rand.-Sub. C4.5
Anneal	99.01	98.79	**83.73**	99.61	99.62	98.72	99.22	98.73
Audiology	82.31	80.76	**46.69**	84.62	<u>85.27</u>	81.97	**80.57**	**79.08**
Autos	82.78	83.95	**50.19**	86.05	86.24	83.38	84.31	85.37
Breast-cancer	72.74	73.15	71.71	**69.87**	**68.01**	70.16	71.82	73.25
Breast-w	97.07	96.12	96.08	96.51	96.55	96.17	97.14	96.50
Colic	85.02	85.29	**81.76**	**81.76**	83.56	84.31	84.51	84.85
Credit-g	75.87	74.27	**70.71**	**72.79**	74.59	**72.53**	75.80	74.44
Diabetes	76.57	76.38	75.48	**72.81**	74.67	74.91	76.44	74.61
Dimin	96.62	97.16	**89.83**	96.03	96.33	96.95	97.39	95.00
Haberman	73.30	72.62	73.16	71.12	71.08	72.66	72.86	73.07
Heart-c	83.01	**79.47**	82.34	**79.60**	**80.16**	**78.58**	83.46	81.39
Heart-h	82.35	80.11	81.92	**78.25**	79.97	79.09	81.69	81.48
Heart-statlog	82.93	81.19	83.44	80.15	80.93	80.56	83.33	83.59
Hepatitis	84.89	81.50	**79.38**	82.74	82.93	82.19	83.13	82.81
Hypothyroid	98.20	99.59	98.65	99.67	99.68	98.65	99.66	**95.62**
Ionosphere	94.42	92.45	81.05	93.62	93.50	92.60	93.17	93.71
Iris	95.73	94.67	81.00	94.47	94.40	95.33	94.67	94.33
Kr-vs-kp	98.33	99.43	94.48	99.62	99.62	99.53	99.41	96.97
Letter	95.21	93.50	**82.03**	96.62	94.98	**92.85**	96.75	93.01
Lymphography	83.69	78.75	77.41	83.09	82.42	80.38	86.48	79.76
Monk1	92.29	**82.99**	**58.71**	96.54	93.49	90.13	96.67	**85.97**
Monk2	67.99	**60.33**	**60.87**	61.86	60.44	**59.19**	70.48	61.90
Mushroom	100	100	**98.52**	100	100	100	100	100
Primary-tumor	46.29	45.16	**31.74**	**41.65**	**41.71**	44.22	45.12	45.40
Segment	97.98	97.55	**92.00**	98.42	98.34	98.18	98.23	97.54
Sick	98.38	98.85	97.55	99.06	99.01	97.55	98.91	**95.83**
Sonar	85.07	**79.78**	**70.45**	83.03	83.33	**78.33**	85.14	**82.08**
Soybean	94.24	93.15	**62.90**	93.21	93.28	94.29	94.58	94.36
Spambase	95.26	94.44	**91.48**	95.11	95.61	94.13	94.54	94.87
Students	87.04	86.29	86.81	**81.46**	**82.78**	**80.52**	86.08	86.19
Titanic	78.88	78.00	77.60	78.89	78.65	79.05	78.83	78.19
Vote	96.41	96.50	95.61	95.33	95.52	94.94	96.79	95.42
Vowel	98.59	**91.68**	**56.57**	**95.42**	**95.11**	95.96	98.79	**95.26**
Waveform	85.76	**83.02**	83.24	83.32	83.64	**79.48**	85.52	83.78
Zoo	93.21	93.00	**46.37**	95.38	95.37	93.18	**91.18**	94.27

proving their test-set error up to 25 classifiers [4]. In order to utilize the behaviour mentioned above, we have used 25 sub-learners for all the tested ensembles of classifiers. Regarding the time complexity, our method is competitive with Rotation Forest, Decorate, Bagging, Boosting and Random-Subspace methods with 25 sub-learners. This is easily understood if we consider the use of 5 sub-classifiers for every sub-ensemble.

In our study we have used two well-known and widely used algorithms as base classifiers, namely, the Decision Trees and Rule Learner algorithm. Consequently, in the following subsections, we present the results regarding the algorithms mentioned above. Hereafter, let us briefly describe two significant issues.

Table 5
Friedman ranking (using PART)

Rank	Algorithm
2.45714	RFRS PART
2.74286	Rotation Forest PART
4.32857	MultiBoost PART
4.42857	Boosting PART
4.47143	Bagging PART
4.70000	Random Subspace PART
5.95714	Decorate PART
6.91429	Dagging PART

Firstly, in our experiments, we have attempted to reduce the outcome of any expert bias by avoiding tuning any of the algorithms to a particular dataset. Secondly, the standard learning parameter assignments have been used in all our experiments. Through this simple way, we have achieved lower estimates of the true error rate. However, that bias influences all learning models uniformly. It should be mentioned that we have conducted our experiments using the widely used machine learning software of Waikato Environment for Knowledge Analysis (Weka) [28].

4.1. *Using C4.5 algorithm as base classifier*

In our first attempt, we have used an established classifier as the base classifier in the proposed scheme, the well-known Decision Tree algorithm. According to [29,30], decision trees are one of the most common classifiers in the field of machine learning. Specifically, the classification process is based on the values of the features. Also, the algorithm sorts the examples by their attribute values. Thus, every node of the tree briefly describe an attribute in an instance to be classified. Every branch of the decision tree presents a value that the node is associated. The classification task starts at the root of the tree. The instances are classified according to their feature values. The feature that optimally divides the training data occupies a position as the root of the tree.

The procedure described above is repeated similarly in each division of the tree. Thus, smaller trees are created, the so-called sub-trees, until the data sets are subdivided into smaller batches, and finally, the examples in a node belong to the same class. Nevertheless, a crucial issue that may occur is the case of overfitting [31]. This case is a common problem that may arise in different learning algorithms, such as decision trees or artificial neural networks. Precisely, during the overfitting case, the learning algorithm fits perfectly to the training data. On the other hand, it may be imprecise in predicting the results of the unseen data. Subsequently, we briefly describe three of the most common processes to balance the issue mentioned previously.

A reliable solution can be provided by applying data pre-processing steps [32]. In particular, using this technique, we avoid the immediate simplification of the results. Thus, implementing the appropriate pre-processing steps, we try to simplify or extract useful information from the dataset to feed the learning algorithm with a small amount of data. In general, the purpose of these techniques is to strive to clean the dataset in order to select the most suitable features for building a simpler learner. Secondarily, the usage of pre-pruning and post-pruning technique is adopted. According to the first technique, a termination criterion can be applied for controlling the height of the decision tree. On the other hand, post-pruning eliminates some of the terminal branches in order to increase the classification accuracy.

In the paper at hand, we use one of the most known and widely used algorithms [33] that belongs to the family of decision trees, namely the C4.5 learner. In order to make the partition process optimal, the C4.5 algorithm uses a term called *information gain*. According to that statistical term the feature that best divides the training dataset is determined. The problematic issue of overfitting is handling through a set of rules. One rule for every path from the root to a leaf node is generated, and by appropriate generalisation, the accuracy is maximised. In general, the algorithm of decision trees is a model which is characterised by instability and hence small changes to the training set lead to significant final prediction changes.

The comparison is between the proposed model and the established ensemble methods, such as Bagging, Dagging, Boosting, MultiBoost, as well as Decorate, Rotation Forest and Random-SubSpace using as base classifier the C4.5 classifier with 25 sub-classifiers. In order to test the statistical performance, we have used the well-known Friedman test [34] and the Finner Post-hoc test [35]. Thus, the proposed ensemble method has exhibited statistically better performance than the other considered ensemble methods. In Tables 2 and 3, the results obtained by using the Friedman and Finner Post-hoc tests are presented. Moreover, using underline (in Table 4) we denote the performance of the methods outperforming the proposed method according to the same statistical tests. Particularly, the methods that are outperformed by the proposed ensemble method are displayed in bold.

Table 6
Finner Post-hoc (using RFRS PART as control method)

Comparison	Statistic	Adjusted p-value	Result
RFRS PART vs Dagging PART	7.44124	0.00000	H0 is rejected
RFRS PART vs Decorate PART	5.80661	0.00000	H0 is rejected
RFRS PART vs Random Subspace PART	3.65963	0.00059	H0 is rejected
RFRS PART vs Bagging PART	3.26927	0.00189	H0 is rejected
RFRS PART vs Boosting PART	3.19607	0.00195	H0 is rejected
RFRS PART vs MultiBoost PART	3.02529	0.00290	H0 is rejected
RFRS PART vs Rotation Forest PART	0.14639	0.88362	H0 is accepted

Table 7
Comparisons of the proposed ensemble method with established ensemble methods using the PART classifier as base classifier, where “MultiB.” denotes the MultiBoost algorithm, “Rot.Forest” declares the Rotation Forest method and “Rand.-Sub.” indicates the Random-Subspace algorithm

Dataset	RFRS PART	Bagging PART	Boosting PART	Ran.-SubS. PART	Dagging PART	MultiB. PART	Decorate PART	Rot.Forest PART
Anneal	99.11	98.69	99.60	98.73	**84.63**	99.45	98.89	99.33
Audiology	83.16	82.68	85.28	**79.00**	**46.55**	85.50	**81.32**	**81.56**
Autos	83.79	81.98	84.38	83.75	**50.56**	83.75	82.33	83.83
Breast-cancer	74.16	**71.33**	**68.93**	73.13	72.29	**69.38**	**68.92**	74.16
Breast-w	97.28	96.37	96.50	96.72	96.08	96.61	95.57	96.99
Colic	84.23	85.21	**81.30**	84.69	81.90	83.99	83.96	84.50
Credit-g	76.10	75.18	74.03	75.81	**71.56**	74.62	**72.30**	77.80
Diabetes	77.35	75.95	**73.78**	74.96	74.79	74.88	75.53	76.43
Dimin	97.47	97.75	96.51	95.46	**85.13**	96.89	95.37	97.09
Haberman	73.53	73.30	72.18	72.84	73.46	72.18	74.48	73.52
Heart-c	84.44	81.95	80.47	83.27	82.61	**81.03**	**79.49**	83.80
Heart-h	82.03	82.25	81.47	82.45	82.13	81.53	**78.62**	83.71
Heart-statlog	83.70	81.70	80.78	83.44	83.33	81.22	**77.78**	81.48
Hepatitis	84.54	83.71	83.52	83.83	**79.38**	84.48	82.66	84.46
Hypothyroid	98.94	99.63	99.66	96.26	98.60	99.66	98.60	99.71
Ionosphere	94.89	92.77	93.51	93.51	**80.97**	93.54	92.32	95.17
Iris	95.33	94.6	94.93	94.67	**79.80**	94.73	95.33	95.33
Kr-vs-kp	98.94	99.46	99.67	97.28	**95.68**	99.68	98.59	99.62
Letter	95.18	94.20	94.33	94.12	92.96	94.67	93.13	95.11
Lymphography	85.76	83.20	84.13	83.01	**77.21**	83.19	**81.67**	84.38
Monk1	92.56	98.57	98.81	**84.74**	**61.87**	98.49	91.79	95.83
Monk2	65.04	67.75	66.24	62.49	60.12	65.78	**58.05**	68.09
Mushroom	100	100	100	100	**98.60**	100	100	100
Primary-tumor	45.40	45.34	**42.21**	46.11	**31.77**	**42.6**	44.82	44.52
Segment	98.01	97.80	98.45	97.69	**91.95**	98.28	97.92	98.18
Sick	98.44	98.79	98.92	96.69	97.44	98.85	97.44	98.65
Sonar	87.55	**81.41**	**83.05**	**83.75**	**70.45**	**82.95**	85.60	88.00
Soybean	93.40	93.26	93.22	94.13	**56.79**	93.43	93.26	93.99
Spambase	95.47	95.22	95.44	94.08	**92.96**	95.11	94.04	95.41
Students	86.95	84.08	**81.37**	85.44	85.73	**82.43**	**78.50**	84.92
Titanic	78.83	78.37	78.98	78.27	77.60	78.78	78.92	78.96
Vote	96.10	96.52	94.94	95.56	95.61	95.28	94.49	96.33
Vowel	98.38	**91.33**	94.32	96.15	**57.08**	**93.71**	96.36	97.98
ïijůaveform	85.88	84.42	84.19	84.94	83.82	84.19	83.32	86.34
Zoo	94.27	93.20	95.75	94.28	**46.37**	94.28	93.27	**91.18**

4.2. *Using PART algorithm as base classifier*

In this subsection, we present the second application with a different base classifier. In that attempt, we have used a rule-based learner as the base classifier of the ensemble scheme. In rule-based strategies [36] the main concept includes a series of Boolean clauses related to logical AND operators that jointly denote membership in a specific class. This process aims in constructing the smallest rule-set that is consistent with the training data. The most common conditions are the following: when the number of rules is large, the algorithm tries to memorise as much training data as possible. Contrary to the situation we need, i.e. to detect the hidden pattern in the data. In our study, we have used one of the most known and widely used algorithms [37] belonging to the family of rule-based learners, the PART algorithm.

Therefore, regarding the PART version, Friedman and Finner Post-hoc procedures have been employed and the obtained results are presented in Tables 5 and 6.

The comparisons between the proposed method and the established ensemble methods, such as Bagging, Boosting, Random-SubSpace, as well as Dagging, MultiBoost, Decorate and Rotation Forest using as base classifier the PART classifier are presented in Table 7.

5. Conclusion remarks and discussion

The issue of creating an efficient, reliable and competitive ensemble of classifiers is an up-to-date scientific field of supervised machine learning. Well-known and widely used studies [38,39] indicate that an ensemble of classifiers outperforms the individual ones. The dominant cause is that most of learners use local optimization techniques, which may remain in local solutions. It is well-known that decision trees utilise a greedy local search. Even though a learning algorithm may, in principle, attain the best hypothesis, that is not possible in practice. On the other hand, an ensemble of classifiers may obtain a more reliable approximation, even if no further information is provided [39].

In our study, we have provided the RFRS method that combines the effectiveness of two widely used and established methods in a single algorithm, namely, the Rotation Forest and Random Subspace method. The experiments that we have conducted indicate that the RFRS method achieves better accuracy in most cases compared to other well-known ensemble methods, such as Bagging, Boosting and Random Subspace. The proposed ensemble algorithm achieves lower error in comparison to other ensembles (including Boosting, Random Subspace and other established methods) when base learning algorithm, such as C4.5 and PART, has been utilized.

Recently, significant progress has been made in the construction of new, efficient ensemble algorithms. However, there are significant issues that need further investigation. Thus, a challenging future goal is to test the proposed RFRS method in regression problems. Moreover, a considerable issue is the automatic selection of appropriate and more effective learners in order to integrate the best ensemble scheme. Furthermore, in general, a diachronic issue for creating a good ensemble is the selection of the most suitable number of participants. This issue will be further studied in combination with other well-known machine learning algorithms.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

References

[1] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation. 1997; 1(1): 67-82.

[2] Adam SP, Alexandropoulos SAN, Pardalos PM, Vrahatis MN. No free lunch theorem: A review. Approximation and Optimization. 2019; pp. 57-82.

[3] Pardalos PM, Rasskazova V, Vrahatis MN, editors. Black Box Optimization, Machine Learning, and No-Free Lunch Theorems. vol. 170 Springer Optimization and Its Applications-Mathematics. Springer, 2021.

[4] Dietterich TG. Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Systems. Springer, 2000; pp. 1-15.

[5] Opitz D, Maclin R. Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research. 1999; 11: 169-198.

[6] Aličković E, Subasi A. Breast cancer diagnosis using GA feature selection and Rotation Forest. Neural Computing and Applications. 2017; 28(4): 753-763.

[7] Zhu HJ, You ZH, Zhu ZX, Shi WL, Chen X, Cheng L. DroidDet: effective and robust detection of android malware using static analysis along with rotation forest model. Neurocomputing. 2018; 272: 638-646.

[8] Rodriguez JJ, Kuncheva LI, Alonso CJ. Rotation forest: A new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006; 28(10): 1619-1630.

[9] Ho TK. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1998; 20(8): 832-844.

[10] Breiman L. Bagging predictors. Machine Learning. 1996; 24(2): 123-140.

[11] Freund Y, Schapire RE. Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning. vol. 96. Morgan Kaufmann; 1996. pp. 148-156.

[12] Melville P, Mooney RJ. Constructing diverse classifier ensembles using artificial training examples. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence. vol. 3. Morgan Kaufmann; 2003. pp. 505-510.

[13] Zhou ZH. Ensemble methods: foundations and algorithms. CRC press, 2012.

[14] Rokach L, Maimon O. Decision trees. In: Data Mining and Knowledge Discovery Handbook. Springer; 2005; pp. 165-192.

[15] Cohen WW, Singer Y. A simple, fast, and effective rule learner. AAAI/IAAI. 1999; 99(335-342): 3.

[16] Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer; 2009; pp. 475-482.

[17] Ros F, Guillaume S. Sampling Techniques for Supervised or Unsupervised Tasks. Springer, 2020.

[18] Yariyan P, Janizadeh S, Van Phong T, Nguyen HD, Costache R, Van Le H, et al. Improvement of best first decision trees using bagging and dagging ensembles for flood probability mapping. Water Resources Management. 2020; 34(9): 3037-3053.
[19] Schapire RE, Singer Y. Improved boosting algorithms using confidence-rated predictions. Machine Learning. 1999; 37(3): 297-336.
[20] Webb GI. Multiboosting: A technique for combining boosting and wagging. Machine Learning. 2000; 40(2): 159-196.
[21] Kuncheva LI, Skurichina M, Duin RP. An experimental study on diversity for bagging and boosting with linear classifiers. Information Fusion. 2002; 3(4): 245-258.
[22] Breiman L. Random forests. Machine Learning. 2001; 45(1): 5-32.
[23] Vidal R, Ma Y, Sastry SS. Generalized Principal Component Analysis. Springer; 2016.
[24] Pham BT, Bui DT, Prakash I, Dholakia M. Rotation forest fuzzy rule-based classifier ensemble for spatial prediction of landslides using GIS. Natural Hazards. 2016; 83(1): 97-127.
[25] Burrough PA, McDonnell R, McDonnell RA, Lloyd CD. Principles of Geographical Information Systems. Oxford University Press, 2015.
[26] Blaser R, Fryzlewicz P. Random rotation ensembles. The Journal of Machine Learning Research. 2016; 17(1): 126-151.
[27] Dua D, Graff C. UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2019; 2019.
[28] Witten I, Frank E, Hall M, Pal C. Data mining fourth edition: Practical machine learning tools and techniques. San Francisco: Morgan Kaufmann Publishers Inc; 2016.
[29] Kotsiantis SB. Decision trees: a recent overview. Artificial Intelligence Review. 2013; 39(4): 261-283.
[30] Quinlan JR. Induction of decision trees. Machine Learning. 1986; 1(1): 81-106.
[31] Roelofs R, Shankar V, Recht B, Fridovich-Keil S, Hardt M, Miller J, et al. A meta-analysis of overfitting in machine learning. Advances in Neural Information Processing Systems. 2019; 32: 9179-9189.
[32] Alexandropoulos SAN, Kotsiantis SB, Vrahatis MN. Data preprocessing in predictive data mining. The Knowledge Engineering Review. 2019; 34.
[33] Ruggieri S. Efficient C4.5 [classification algorithm]. IEEE Transactions on Knowledge and Data Engineering. 2002; 14(2): 438-444.
[34] Hodges J, Lehmann EL. Rank methods for combination of independent experiments in analysis of variance. In: Selected Works of EL Lehmann. Springer; 2012; pp. 403-418.
[35] Finner H. On a monotonicity problem in step-down multiple test procedures. Journal of the American Statistical Association. 1993; 88(423): 920-923.
[36] Weiss SM, Indurkhya N. Rule-based machine learning methods for functional prediction. Journal of Artificial Intelligence Research. 1995; 3: 383-403.
[37] Frank E, Witten IH. Generating accurate rule sets without global optimization. 1998.
[38] Džeroski S, Ženko B. Is combining classifiers with stacking better than selecting the best one? Machine Learning. 2004; 54(3): 255-273.
[39] Murty MN, Devi VS. Combination of classifiers. In: Pattern Recognition. Springer; 2011; pp. 188-206.