Noname manuscript No.
(will be inserted by the editor)

A Fine-Grained Random Forests using Class Decomposition

An Application to Medical Diagnosis

Eyad Elyan · Mohamed Medhat Gaber

Received: date / Accepted: date

Abstract Class decomposition describes the process of segmenting each class into a number of homogeneous subclasses. This can be naturally achieved through clustering. Utilising class decomposition can provide a number of benefits to supervised learning, especially ensembles. It can be a computationally efficient way to provide a linearly separable dataset without the need for feature engineering required by techniques like Support Ve]ctor Machines (SVM) and Deep Learning. For ensembles, the decomposition is a natural way to increase diversity; a key factor for the success of ensemble classifiers. In this paper, we propose to adopt class decomposition to the state-of-the-art ensemble learning Random Forests. Medical data for patient diagnosis may greatly benefit from this technique, as the same disease can have a diverse of symptoms. We have experimentally validated our proposed method on a number of datasets in that are mainly related to the medical domain. Results reported in this paper shows clearly that our method has significantly improved the accuracy of Random Forests.

Keywords Machine learning · Random Forests · Clustering · Ensemble learning

Eyad Elyan
School of Computing Science and Digital Media, Robert Gordon University
Garthdee Road, Aberdeen, AB10 7GJ
Tel.: +44-1224-262737
E-mail: e.elyan@rgu.ac.uk

Mohamed Medhat Gaber
School of Computing Science and Digital Media, Robert Gordon University
Garthdee Road, Aberdeen, AB10 7GJ
Tel.: +44-1224-262737
E-mail: m.gaber@rgu.ac.uk

1 Introduction

Ensemble of classifiers, also known as multi-classifier systems (MCS) or committee of experts, has been long studied as a more accurate predictive model than a single classifier [27]. It is usually being illustrated by its resemblance to the democratic process of voting. As with politics, diversity of the voters can enrich the final decision. However, when we look at the supervised learning problem, applying the techniques on the same dataset will result in no diversity, and thus, in the ensemble being equivalent to only one classifier. The solution taken is to vary the dataset, as in bagging [7] and boosting methods [10], or to vary the classification techniques, as in stacking [32]. Some less commonly used solutions have worked on varying the output, like error-correcting codes.

Over two decades of work has proved two ensemble learning techniques to stand out, namely, Random Forest [8] and AdaBoost [15], or its variation Gradient Tree Boosting [16]. More recently, a large scale evaluation of all of these techniques and other state-of-the-art classification methods, has been conducted [14]. The outcome of this evaluation has shown that on average Random Forest is the most accurate classifier, followed by the single classifier, Support Vector Machines. Motivated by this result, in this paper, we propose an important enhancement to Random Forest that has several benefits. We use clustering within classes to decompose the classification problem to a larger number of classes. The benefits gained from such a process can be summarised as follows: (1) diversity will be boosted as class/instance association is increased; (2) unlike feature engineering done in support vector machines, class engineering is a computationally more efficient process that can lead to linearly separable classes; and (3) in many machine learning problems, labelling is done at a higher granularity level, either because the finer level is not known, or it is not significant to document it, e.g., it is good enough to label a patient as being diagnosed with a disease, despite having many subtypes of this disease.

Stimulated by these motives, we propose, develop and validate our class decomposed random forest. Despite the specific application to Random Forest, our method is applicable to any classification method, including single classifier system. We first apply k-means clustering to instances that belong to each class with varying the number of clusters (k). Once each class is decomposed in its clusters (subclasses), we apply Random Forest to the newly class engineered dataset. This process is iterative, as we tune the k parameter. For the work reported in this paper, we used a fixed value for k in each iteration, however, parameters can also be tuned such that the number of subclasses can vary among different classes in the dataset.

In our experiments, we use a number medical datasets for patient diagnosis. As discussed earlier, medical datasets can be a good representative of a higher level granularity being used in labelling the data. Our method is proved to have consistent improvement over traditional Random Forest, with tuning of the k parameter. Despite having fixed k value among all classes for the one

dataset, the method has exhibited higher accuracy. We believe that further tuning of the parameter can even lead to higher accuracy.

The paper is organised as follows. Related work on previous attempts to improve Random Forests and class decomposition is discussed in Section 2. Our proposed method for enhancing Random Forests through class decomposition is discussed in Section 3. A thorough experimental study validating the proposed method for medical diagnosis is presented in Section 4. Finally, the paper is concluded with a short summary and pointers to future work in Section 5.

2 Related Work

Owing to its notable predictive accuracy, many extensions have been proposed to further improve Random Forests. In [29], the use of five attribute goodness measures was proposed, such that diversity in the ensemble is boosted. In addition to Gini index used in CART and random trees that make up Random Forests, Gain ratio, MDL (Minimum Description Length), Myopic ReliefF and ReliefF were used. Also unlike Random Forest in its traditional form, weighted voting was proposed. Both extensions have empirically shown potential in enhancing the predictive accuracy of Random Forests. In [21], McNemar non-parametric test of significance was used to limit the number of trees contributing to the majority voting. In a related work to this one, authors in [30] used more complex dynamic integration methods to replace majority voting. Stimulated by the low performance reported in high dimensional data sets, weighted sampling of features was proposed in [3]. In [6], each tree in a Random Forest is represented as a gene with the trees in that Random Forest represent an individual. Having a number of trained Random Forests, the problem has turned to be a Genetic Algorithm optimisation one. Extensive experimental study has shown the potential of this approach. For more information about these techniques, the reader is referred to the survey paper in [13]. In a more recent work, diversification using weighted random subspacing was proposed in [12].

Class decomposition has been proposed for a single classifier system, enhancing the performance of instance based classification in [1]. Adding to the distance measure adopted in this class of classifiers, density and gravity were proposed to assess point/cluster matching, i.e., instance labelling. In later work such as [2], the technique was tailored to real-time human activity recognition, which has been experimentally validated. Class decomposition for low variance classifiers including Naive Bayes and linear classification methods have been proposed in [31]. Such classifiers characterised by their lack of flexibility in their decision boundary. Utilisation of class decomposition was proved to enhance the performance, at the time of maintaining a low variance. More recently, Polaka [26] has used class decomposition to boost the performance of three classifiers (C4.5, Random Forests and Support Vector Machines) on a number of medical datasets. With the motive that diseases can have more than one

form, the class decomposition was only applied to the positive class. Both hierarchical clustering and k-means were tested. When used with Random Forests, k-means has resulted in a better accuracy when compared to hierarchical clustering, as the accuracy has increased in all the 8 datasets used, as opposed to only 3 datasets for the hierarchical clustering.

Having reviewed the closely related literature, we assert that the potential of diversifying Random Forests using class decomposition has not been explored. Thus, this work augments the existing body of knowledge in this area in the following ways.

- The proposed method is the first to apply class decomposition across all classes in the dataset in multiple classifier systems. The other work that has used class decomposition in Random Forests applied it to only the positive class of medical diagnosis datasets [26]. We argue that the negative class can also benefit from the class decomposition process, as healthy people may still exhibit some symptoms of the disease under consideration for diagnosis.
- Diversification of the ensemble has been motivated in this work, and as such we hypothesise that applying clustering to all classes is beneficial regardless of the quality of the produced clusters.
- Unlike the work in [31] that merged clusters after producing them, we have not used the merging step, as this will decrease the chances for ensemble diversification. This has not been an issue in the previous work that only targeted single classifier systems with low variance.

3 Methodology

Our method is mainly based on decomposing the class labels for a given dataset. In other words, finding the within-class similarities between different instances/observations of a dataset and group them accordingly. With this approach, we can introduce more diversity to the dataset, aiming at improving classification accuracy. Diversity can formally be introduced by increasing the output space of the relationship between the feature vector X and the class label set Y. If Y is decomposed to $Y^{'}$, then $|Y^{'}| > |Y|$, where $|Y^{'}|$ and $|Y|$ are the number of class labels after and before decomposition respectively, the set relationship $X \times Y^{'} > X \times Y$, producing enriched diversity in the dataset.

To motivate the discussion, assume that we have a dataset A with m number of instances $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m$, where each instance $\mathbf{x}_i$ is defined by an n number of features as $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{in})$. In a typical supervised machine learning scenario, these instances are often labelled or categorised (i.e. by human experts in case of medical sets or from historic data i.e. banking details, customer information etc). Formally, such dataset along with its labels may be represented as follows:

$$A = \begin{bmatrix} x_{11} & x_{12} & ..., & x_{1n} \\ ... & x_{22} & ..., & ... \\ ... & ... & ... & ... \\ x_{m1} & ... & ..., & x_{mn} \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ .. \\ .. \\ y_m \end{bmatrix} \quad (1)$$

where $y_i \in R^k$ represents the class or label of the i^{th} training example and k represents the number of unique classes in the data set. The aim of a learning algorithm is to devise a function *h(x)* that maps an instance $\mathbf{x}_i \in A$ to a class $\mathbf{y}_j \in Y$. The learning algorithm is trained using a subset of A, often referred to as the training set, while the remaining instances are used for testing how good *h(x)* generalises. Intiutively speaking, $h(x) = y$ would only be considered a correct classification if the class of $x = y$.

With this simplification, it is clear that the main components that play a critical role in devising an accurate mapping function with high accuracy are: (1) the choice of learning algorithm, which is often influenced by the type of data (i.e. dimension, size, etc); (2) the choice of features or attributes to represent the training examples; and (3) the class labels. Although a lot of work has been done regarding the first two components, to the best of our knowledge very little work has been done in terms of class labels or class decomposition. In this paper, our focus will be on the class labels of the training samples as will be discussed in the following sections. But first, we will briefly discuss the machine learning algorithm that we will be using in this work, and consequently justify this choice.

3.1 Random Forests

The method we are proposing could be applied to any learning algorithm, however, for the purpose of this paper, we chose Random Forests (RF). Over the past few years RF proved to be one of the most accurate techniques and is currently considered a state of the art. In a recent paper [14], RF came top out of a 179 different classifiers of different families (i.e. Bayesian, Neural Nets, Discriminant Analysis, etc) when used in classifying 121 different datasets from the UCI repository.

Random Forests is an ensemble learning technique that has been successfully used for classification and regression. RF was developed by Brieman [8]. The method works as a set of independent classifiers (typically decision trees), where each classifier casts a vote for a particular instance in a dataset, and then majority voting is considered to determine the final class label.

The method combines Breiman's bagging method [7], and the random selection of features introduced by Ho ([17] and [18]) and Amit et al. [4]. The decision trees in the ensemble are constructed using sampling with replacement from the training set. With such technique, approximately 63.2% of the samples are likely to appear in the training process (constructing each decision tree) and these are referred to as in-bag sample, while the remaining $\approx$ 36.8%

is often referred to as the out-of-bag sample and is used for testing the performance of the forest. RF has several advantages over other learning techniques, but one of its key advantages is its robustness to noise and overfitting. The random selection of features is done at each node split for building the tree. Typically this setting is $\sqrt{n}$, where n is the number of features. However, in some implementations of Random Forests, $\log_2(n)$ is used instead, encouraging less features to be drawn at each node split for higher dimensional datasets. Trees are allowed to fully grow without any pruning applied. As different set of features are chosen at each node split, it is likely that all the features will be used to build the one tree. However, each tree will have a different structure, attributed to the bootsrap sampling applied and random selection of features for goodness evaluation at each node split.

3.2 Class Decomposition

The idea of decomposing the class labels for a particular data set is based on the assumption that within each class of instances, further clustering could be identified. For example, in classical hand-written digit recognition, the digit "8" could be written in so many different ways, which may or may not share common characteristics, hence decomposing the set of instances that are labelled as "8" into a set of clusters that share certain characteristics may certainly improve diversity and consequently improve classification accuracy. Similarly, in a medical dataset with hundreds of observations, assume that each of these observations is labelled to indicate whether a disease is present or not (i.e. 0, 1 respectively). Further class decomposition could be applied and may lead to better representation of the data (i.e. a disease is present and mild, present and severe, etc).

In this paper, and in order to achieve class-decomposition we used *kmeans* clustering algorithm [24] aiming at minimising the within-cluster sum of squares for each group of instances that belongs to the same class label (as will be discussed in the following sections). In other words, for a given dataset with a feature set A as in Equation 1, we apply *kmeans* clustering algorithm to obtain another set A_c with a new set of class labels $Y^{'}$ where $Y^{'}$ is defined as in Equation 2

$$Y^{'} = (y_{01}, y_{02}, ..., y_{0c}, y_{11}, y_{12}, ..., y_{1c} ..., y_{k1}, y_{k2}, ..., y_{kc}) \tag{2}$$

where c is the number of clusters within each class in A. It should be noted here that with such restructuring of the class labels which is applied accross all labels (multi-class decomposition) , the number of unique class labels will increase from k to ck. In addition, for any classifier $h(x)$, where x belongs to a class y_i, $h(x) = y_{ij}$ is considered as a correct classification $\forall j \in 0, 1, ..., c$.

For the purpose of illustration, consider the data sets A and A_c shown in Equation 3, where A_c is equal to A but with its class labels decomposed into c subclasses:

$$A = \begin{bmatrix} x_{00} & x_{01} & ... & x_{0N} & a \\ x_{10} & x_{11} & ... & . & a \\ . & . & . & . & a \\ . & . & . & . & a \\ . & . & . & . & . \\ x_{M0} & . & ... & x_{MN} & b \end{bmatrix}, A_c = \begin{bmatrix} x_{00} & x_{01} & ... & x_{0n} & a_1 \\ x_{10} & x_{11} & ... & . & a_2 \\ . & . & . & . & a_2 \\ . & . & . & . & a_c \\ . & . & . & . & . \\ x_{m0} & . & ... & x_{mn} & b_c \end{bmatrix} \tag{3}$$

Let us now assume that we applied a learning algorithm to both datasets A and A_c, which results in the classification functions Φ and Φ_c respectively. Suppose also that we applied Φ and Φ_c to a testing set resulting in h and h_c confusion matrices shown in (4).

$$h = \begin{bmatrix} & a & b \\ a & 50 & 0 \\ b & 0 & 50 \end{bmatrix}, h_c = \begin{bmatrix} & a_1 & a_2 & b_1 & b_2 \\ a_1 & \mathbf{10} & \mathbf{5} & 0 & 0 \\ a_2 & \mathbf{4} & \mathbf{31} & 0 & 0 \\ b_1 & 0 & 0 & \mathbf{18} & \mathbf{6} \\ b_2 & 0 & 0 & \mathbf{9} & \mathbf{17} \end{bmatrix} \tag{4}$$

We can measure the performance of Φ and Φ_c using the confusion matrices denoted by h and h_c 4. Lets assume for simplicity that $c = 2$ (number of subclasses within each class label), the number of instances in the testing set is 100 and that classification accuracy of both functions is 100%. Measuring the performance of the learning algorithm when applied to the set A (Φ), we simply sum up all the diagonal elements of the confusion matrix h and divide it by the total number of observations (i.e. 100), as shown in Equation 5

$$Accuracy(\phi(A)) = \left(\sum_{i=0}^{m} h_{ii}\right)/m \tag{5}$$

where m is the number of instances in the dataset, and in many types of data sets m is often greator than n. Notice however that computing the accuracy for Φ_c using the confusion matrix h_c, is slightly different from Equation 5. Here, we not only sum all the diagonal elements in the confusion matrix, but also all elements within the same clusters even if they are off-diagonal of the matrix. In Equation (4) for h_c, all elements that are considered as correct classifications are highlighted with **bold** font face.

3.3 Putting it together

Algorithm 1, depicts the method presented in this paper which could be summarised by four major steps:

1. Pre-process input data (i.e. the feature set A of the given dataset).
2. Decompose the class labels of A using *kmeans* and store the resulting set in A_c.
3. Apply Random Forests to both datasets and compare the results for parameter tuning.

4. If parameter tuning is needed, repeat (2) and (3). Otherwise, terminate the process with current settings.

In Algorithm 1 RF and RFC represent the application of Random Forest on the original dataset and the decomposed dataset respectively. At the same time, $bestRF$ and $bestRFC$ represents the best performing random forest on the original dataset A and the clustered dataset A_c respectively. In other words, $bestRF$ is considered to be the best performing Random Forest (e.g. among all $RF's$) subject to the k value and the corresponding number of trees in the iteration, while $bestRFC$ is considered to be the best ensemble classifier subject to the same parameters but when performed on the decomposed dataset.

Algorithm 1 Compute bestRF, bestRFC (Multi-Class Decomposition)

Require: $minK, maxK, minNTree, maxNTree, treeIncrement$
  $bestRF \leftarrow 0$
  $bestRFC \leftarrow 0$
  $k \leftarrow minK$
  $i \leftarrow 0$
  while $k < maxK$ **do**
    $A_c \leftarrow kmeans(A, k)$
    for $(n = minNTree, n < maxNTree, n = n + treeIncrement)$ **do**
      $RF \leftarrow randomForest(A, n)/2$
      $RFC \leftarrow randomForest(AClust, n)/2$
      if $(RF > bestRF)$ **then**
        $bestRF \leftarrow RF$
      end if
      if $(RFC > bestRFC)$ **then**
        $bestRFC \leftarrow RFC$
      end if
    end for
    $k \leftarrow (k + 1)$
  end while

Notice that although there are several parameters that can be tuned in order to improve the accuracy of the Random Forests, we held all these parameters fixed, and only changed the number of trees in the Random Forests to vary between $minNTree$ and $maxNTree$ trees. In addition, we varied the number of clusters (k) value subject to the size of the training set. This simplification is needed to assess the value of class decomposition when fixing the number of parameters.

It is worth pointing out here that in Algorithm 1 we apply class decomposition (clustering) to all available classes in the dataset (multi-class decomposition). In this paper and in order to compare the predictive accuracy of our technique to the previously proposed method by [26] we have applied single-class decomposition, where only positive classes in the datasets have been decomposed. Algorithm 2 is similar to Agorithm 1 apart from the change to the clustering algorithm, where we pass which **class** to be decomposed

($kmeans(A, class, k)$) as a paramter. In such scenario, the class to be decomposed is chosen to be the postive class in a binary classification problem (e.g. presense of a disease such as cancer, etc..).

Algorithm 2 Compute bestRF, bestRFC (Single-Class Decomposition)

Require: $minK, maxK, minNTree, maxNTree, treeIncrement, class$
  $bestRF \leftarrow 0$
  $bestRFC \leftarrow 0$
  ....
  while $k < maxK$ **do**
    $A_c \leftarrow kmeans(A, class, k)$
    ..
    ..
  end while
  ..

4 Experimental Study

The aim of this experimental study is to establish the usefulness of class decomposition when applying Random Forests to medical diagnosis datasets. To achieve this aim, we applied multi-class decomposition to 7 real datasets from the medical diagnosis domain, varying the two parameters (number of clusters k, and number of tree in the ensemble), as discussed in the previous section. We have also applied single-class decomposition to 5 datasets, these sets have been chosen out of the 7 used sets because they are binary classification problems. Description of the used datasets, discussion of the results, and details of the implementation environment are discussed in details in this section.

4.1 Datasets

Seven datasets have been selected for testing the method presented in this paper. All these sets are from the medical domain as can be seen in Table 1, and most of it have been downloaded from the UCI repository [5]. These include (1) Breast Cancer Wisconsin (Original) Data Set [25], (2) Heart Disease [28], (3) Lung Cancer [19], (4) Mammographic Mass Data Set [11], (5) Parkinsons Data Set [23], (6) Pima Indians Diabetes [5], (7) Thyroid [1]. Table 1 summarises the main Charasteristics of these datasets.

As can be seen in Table 1, these sets vary in terms of number of instances (from 32 to 7000 instances), number of attributes (from 5 to 56 attributes) and number of class labels (i.e. 2 and 3 classes). Notice that the data sets highlighted in **bold** font in Table 1 have been used to apply single-class decomposition. In other words, for comparision purposes between our method

[1] https://archive.ics.uci.edu/ml/machinelearningdatabases/thyroiddisease/annReadme

Table 1 Sets used for experiments

No	Dataset	Size	Number of Attributes	Number of Classes
1.	**Breast Cancer Wisconson**	569	31	2
2.	**Heart**	269	13	2
3.	Lung Cancer	32	56	3
4.	**Mammographic**	961	5	2
5.	**Parkinsons**	195	23	2
6.	**Diabetes**	768	8	2
7.	Thyroid	7198	21	3

and the single class decomposition, we held the settings of all parameters the same, and the only change is related to the chosen data sets, where only those with binary classes have been considered.

4.2 Experiments Setup

In this subsection, we will use an illustrative example of running our method on the widely used Optical Recognition Character set (OCR) dataset from the UCI repository [5]. This illustration serves two purposes: (1) establishing the generality of the method for applying it to other domains; and (2) exemplifying the methodology used in the evaluation when applied in the medical diagnosis domain. The feature set of every dataset A used in this experiment was subject to pre-processing where appropriate, in particular normalisation where features values are standardised in the range of 0 to 1 as can be seen in Equation 6

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{6}$$

Where x_i represents the i^{th} value of feature/ attribute x in the dataset, and $max(x), min(x)$ represent the maximum and minimum values in feature x. This step is important in particular to supress the sensitivity of *kmeans* algorithms to outliers. Once data is normalised then *kmeans* clustering algorithm was applied to it (Equation 7), resulting in a new set with its class labels decomposed into a set of sub-classes k.

$$A^{'} = normalize(A) + cluster(A) \tag{7}$$

For each dataset, Random Forest was applied to it twice. One time on the original data set (i.e. set A in Equation 7) and we refer to this experiment as RF, and the other time, Random Forest was applied to the clustered version of the dataset (i.e. $A^{'}$), and we refer to this as RFC, denoting *clustered* based Random Forest. All parameters were held fixed apart from number of trees in the Random Forests and the k value. This allowed us to establish the usefulness of class decomposition. Figure 1 depicts these settings.

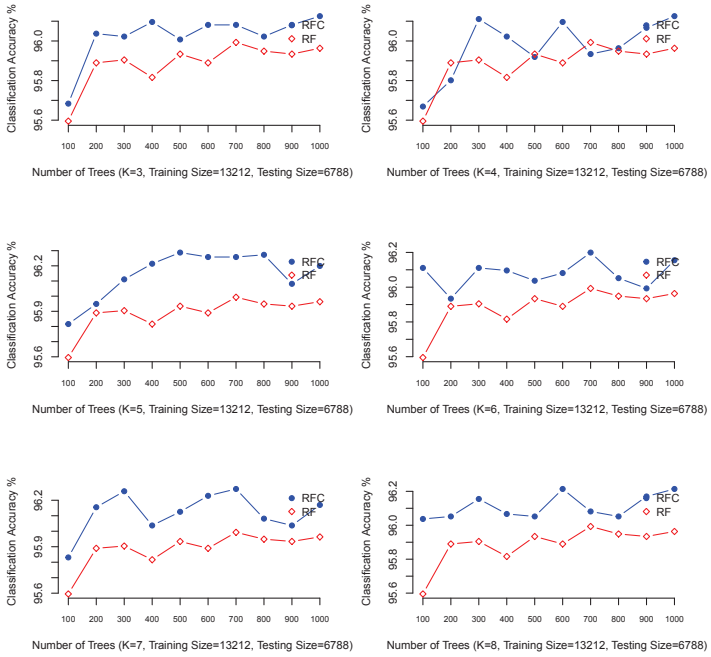

Fig. 1 Experiment setup, single run

We used hold-out method in evaluating the predictive accuracy of the method. As can be seen in Figure 1, the size of the training set is almost 66% of the total size of the set (20,000), and each observation in the set has 16 attribute. It can be observed that our method (RFC) is winning in all settings over (RF) according to Figure 1. The improvement in this experiment is statistically significant; adopting the paired *t-test*, the p *-value* is 6.932×10^{-7} with 95% confidence, and using the Wilcoxon signed rank test [33], the p-value 0.005889.

This experiment is given for illustration purposes, and thus we have only used one run. To ensure consistency of the results when applied on the medical diagnosis datasets, each experiment was repeated 10 times, averaging the results.

4.3 Implementation & Working Environment

All datasets used in these experiments (Table 1) were divided into two sets, training set with 66% of the samples and testing set which represents the remaining 34% of the set, adopting a hold-out methodology in assessing the performance of the techniques. The datasets were randomly splitted within the class labels of the observations in order to balance the class distributions

within the splits (i.e. training, testing). A typical result of one experimental run is shown in Table 2. Notice here, that the best performing Random Forests results are selected (i.e. the ones with the best k values and the number of trees).

Table 2 Results of one experimentś run (multi-class decomposition)

No	Dataset	Best k values	*RFC*	*RF*	Win-Lose
1.	Breast Cancer Wisconson	3	93.78	93.78	Tie
2.	Heart	3	82.42	84.62	Lose
3.	Lung Cancer	1	70	70	Tie
4.	Mammographic	3	85.58	84.97	Win
5.	Parkinsons	3	93.94	90.91	Win
6.	Diabetes	2	75.86	74.71	Win
7.	Thyroid	3	99.55	99.51	Win

A computational framework was implemented using **R** and randomForest package [22] which implements Brieman and Cutler Random Forest for Classification and Regression[2]. The experiment was carried out using Mac machine (OS X) with 16 GB of RAM and 2.6 GHz Intel Core i7. For both multi-class and single class decomposition we set the minimum k value to equal 2, while the maximum is set to equal 8. For the Lung cancer data set, we set k value to vary between 1 and 2, due to the very small size of the set. Notice that when k equals one then, both *RF* and our method *RFC* must result in the same classification accuracy as can be seen in Table 2. The maximum k value was set to equal 8 for these sets, because it turned out – experimentally – that increasing the value beyond this number does not improve the performance of our method. In [26], it was found that only 2 or 3 clusters for the positive class decomposition in a number of biomedical datasets are able to produce highest possible separable clusters. For validation purposes, we increased this number to 8. However, our study is consistent with the results reported in [26] that found only 2 or 3 clusters are adequate for biomedical datasets, and that clustering with k equals 4 through 10 yielded less separable clusters (in the case of multi-class decomposition). It is worth mentioning that diversity created by clusters is an important motive behind our method, and hence we experimented with up to 8 clusters.

4.4 Results & Discussion

A single run over the datasets according to the above framework shows clearly that re-engineering the class labels improves the Random Forest performance as shown in Figure 1 and as can be clearly seen in Figure 2.

[2] http://cran.r-project.org/web/packages/randomForest/randomForest.pdf

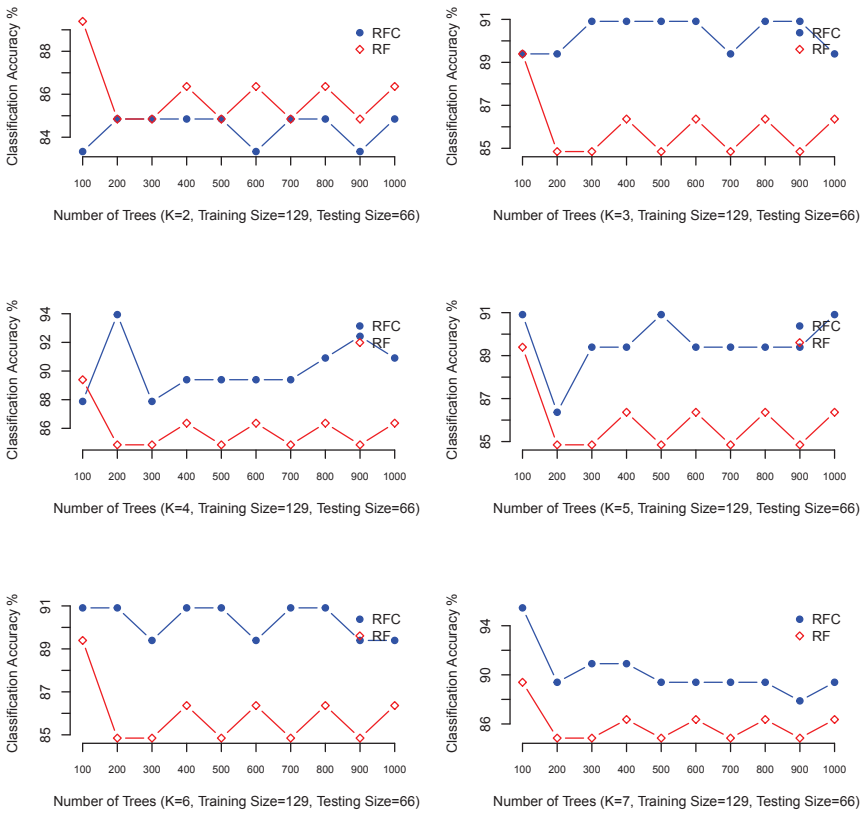

Fig. 2 Parkinsons Dataset, single run

As mentioned earlier, in order to ensure consistency of the results, the experiment was repeated 10 times for each set, and the results were averaged. Here, we also compare our approach to Random Forest and to AdaBoost. Table 3 summarises the results for all medical sets used in this paper.

Table 3 Average results of 10 runs, multi-class decomposition

No	Dataset	Size	k values	RFC_{avg}	RF_{avg}	$AdaBoost_{avg}$
1.	Breast Cancer Wisconson	596	2 *to* 7	**96.735**	95.957	88.238
2.	Heart	269	2 *to* 7	82.637	82.089	84.066
3.	Lung Cancer	32	1 ***to* 2**	**68.956**	66.737	63.000
4.	Mammographic	961	2 *to* 7	**83.835**	83.181	82.147
5.	Parkinsons	195	2 *to* 7	**94.998**	91.363	90.154
6.	Diabetes	768	2 *to* 7	**78.198**	77.931	71.072
7.	Thyroid	7198	2 *to* 7	**99.561**	99.671	99.323

RFC_{avg} represents the results based on our method, while RF_{avg} represents classical Random Forest (i.e. without class decomposition), and $AdaBoost_{avg}$ represents the AdaBoost method. The results shown in Table 3 shows clearly that re-engineering class labels improves the performance of Random Forests. The improvement has an appropriate statistical significance for the application domain; adopting the paired t-test, the p-value is 0.06189 with 95% confidence, and for the Wilcoxon signed rank test, the p-value = 0.03125. It is important to note here, that these results have been achieved with minimum parameter tuning. Thus, there is a clear potential for even further improvement. It is also clear from Table 3 that our proposed method outperformed the evidenced highly accurate ensemble classifier (AdaBoost). These results are also statistically significant, adopting the paired $t-$test, the p-value is 0.03427 with 95% confidence, and the Wilcoxon signed rank test yielded a p-value of 0.04688.

A unique feature for our method is its applicability to multi-class problems, as discussed earlier in the paper. Thus, for the special case of binary classification problems, we run a set of experiments using the 5 binary datasets we use in this experimental work to compare the predictive accuracy of our technique to the previously proposed method by [26] to perform class decomposition on the positive class only. To ensure consistency, we used the same experimental setup which includes repeating the experiment 10 times for each set, and then averaging the results. Table 4 shows the superiority of our method (denoted by RFC_{avg}) over the single-class decomposition method ($RFCS_{avg}$) in 3 out of the 5 datasets. Paired t-test with 95% confidence revealed that the results are of minor statistical significance with p-value = 0.6298, and using the Wilcoxon signed rank test, the p-value = 0.625. Class decomposition over the positive class has helped improve the performance, as previously reported in [26]. However, it is worth noting that our method is more generic in terms to its applicability to all classification problems, and thus the results prove the potential of the new technique.

Table 4 Average results of 10 runs and comparision with the proposed method

No	Dataset	# of Classes	RFC_{avg}	$RFCS_{avg}$	Difference
1	Breast Cancer Wisconson	2	**96.73**	96.27	+0.47
2	Heart	2	82.64	84.07	-1.43
3	Mammography	2	**83.83**	82.85	+0.98
4	Parkinsons	2	95.00	95.23	-0.23
5	Diabetes	2	**78.20**	76.63	+1.57

This enhancement in the medical diagnosis results can be attributed to a number of reasons. Most medical diagnosis datasets have a high granularity of labelling the data. As we can see in the 7 medical datasets used in this experimental study, only 2 or 3 class labels were used. Even with appropriate

level of labelling, inherent subclasses can be always discovered at later stage. This is especially true in the medical domain. In recent years, a number of flu viruses have been affecting the world of travel and business. Modelling flu in a binary classifier that predicts the existence of one of the viruses can be achieved. However, precision can be enhanced, if class decomposition of the flu is applied, resulting in a multi-class problem. Finally, medical diagnosis is a complex process with a high degree of uncertainty and non-linearity, a class-decomposed data can simplify the process by looking at cohesive subclasses, instead of modelling more complex higher granular set of classes.

5 Conclusion & Future Work

This paper proposed adopting class decomposition using clustering instances that belong to the same label separately. The outcome of these clustering processes is a fine grained labelled dataset, that ultimately diversifies among individual trees in the trained Random Forests. Naturally, medical diagnosis datasets can further benefit from clustering by grouping the instances sharing similar feature values in one subclass (cluster). The proposed technique has been validated experimentally over 7 medical datasets, showing its potential in enhancing the predictive accuracy of Random Forests in medical diagnosis.

We can indicate a number of future directions to further enhance the performance of the proposed method. Optimising the number of clusters per class can further enhance the performance. Some classes are in need to fine-grained decomposition where the number of clusters can be high; where some others may not need any class decomposition. The nature of the dataset can determine whether or not clustering is needed and the optimum number of clusters. This can be objectively measured using one of the cluster quality measures (e.g., DBI [20]). Also other clustering techniques can be used instead of k-means.

Another direction of future work is to investigate cluster proximity among different classes when classifying unseen instance. Currently, if an instance is assigned to a cluster that belongs to one class, it is assigned the label of the parent class. However, there is a possibility of having a cluster that has proximity to other classes. This is especially true in medical datasets, when symptoms of the different diseases have high degree of similarity.

Finally, as the proposed technique converts binary classification problems to multi-class ones, Error COrrecting Code (ECOC) ensemble methods [9] can be applied to further diversify the classification from the output side. This is especially interesting investigation, as artificial classes (clusters) can enrich possible combination of binary classification problems for the ECOC.

References

1. Z. S. Abdallah and M. M. Gaber. Kb-cb-n classification: Towards unsupervised approach for supervised learning. In *Computational Intelligence and Data Mining (CIDM), 2011*

IEEE Symposium on, pages 283–290. IEEE, 2011.
2. Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy. Adaptive mobile activity recognition system with evolving data streams. *Neurocomputing*, 150:304–317, 2015.
3. D. Amaratunga, J. Cabrera, and Y.-S. Lee. Enriched random forests. *Bioinformatics*, 24(18):2010–2014, 2008.
4. Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Comput.*, 9(7):1545–1588, Oct. 1997.
5. K. Bache and M. Lichman. UCI machine learning repository, 2013.
6. M. Bader-El-Den and M. Gaber. Garf: towards self-optimised random forests. In *Neural Information Processing*, pages 506–515. Springer, 2012.
7. L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, Aug. 1996.
8. L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001.
9. T. G. Dietterich and G. Bakiri. Error-correcting output codes: A general method for improving multiclass inductive learning programs. In *AAAI*, pages 572–577. Citeseer, 1991.
10. H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, and V. Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6(6):1289–1301, 1994.
11. M. Elter, R. Schulz-Wendtland, and T. Wittenberg. The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process. *Medical Physics*, 34:4164, 2007.
12. K. Fawagreh, M. M. Gaber, and E. Elyan. Diversified random forests using random subspaces. In *Intelligent Data Engineering and Automated Learning–IDEAL 2014*, pages 85–92. Springer, 2014.
13. K. Fawagreh, M. M. Gaber, and E. Elyan. Random forests: from early developments to recent advancements. *Systems Science & Control Engineering: An Open Access Journal*, 2(1):602–609, 2014.
14. M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014.
15. Y. Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.
16. J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.
17. T. K. Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282 vol.1, Aug 1995.
18. T. K. Ho. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):832–844, Aug 1998.
19. Z.-Q. Hong and J.-Y. Yang. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition*, 24(4):317 – 324, 1991.
20. A. K. Jain, R. C. Dubes, et al. *Algorithms for clustering data*, volume 6. Prentice hall Englewood Cliffs, 1988.
21. P. Latinne, O. Debeir, and C. Decaestecker. Limiting the number of trees in random forests. In *Multiple Classifier Systems*, pages 178–187. Springer, 2001.
22. A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.
23. M. Little, P. McSharry, S. Roberts, D. Costello, and I. Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine*, 6(1), 2007.
24. J. MacQueen. Some methods for classification and analysis of multivariate observations, 1967.
25. O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *OPERATIONS RESEARCH*, 43:570–577, 1995.
26. I. Polaka. Clustering algorithm specifics in class decomposition. *No: Applied Information and Communication Technology*, 2013.
27. R. Polikar. Ensemble based systems in decision making. *Circuits and systems magazine, IEEE*, 6(3):21–45, 2006.

28. U. Repository. Heart Disease dataset. https://archive.ics.uci.edu/ml/datasets/Statlog+(Heart), 1996. [Online; accessed Dec-2014].
29. M. Robnik-Šikonja. Improving random forests. In *Machine Learning: ECML 2004*, pages 359–370. Springer, 2004.
30. A. Tsymbal, M. Pechenizkiy, and P. Cunningham. Dynamic integration with random forests. In *Machine Learning: ECML 2006*, pages 801–808. Springer, 2006.
31. R. Vilalta, M.-K. Achari, and C. F. Eick. Class decomposition via clustering: a new framework for low-variance classifiers. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 673–676. IEEE, 2003.
32. D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.
33. R. Woolson. Wilcoxon signed-rank test. *Wiley Encyclopedia of Clinical Trials*, 2008.