Random Forests with a Steepend Gini-Index Split Function and Feature Coherence Injection

Mandlenkosi Victor Gwetu, Jules-Raymond Tapamo, Serestina Viriri

Random Forests with a Steepend Gini-Index Split Function and Feature Coherence Injection

Mandlenkosi Victor Gwetu[1], Jules-Raymond Tapamo, and Serestina Viriri

University of KwaZulu-Natal, Private Bag X54001, Durban, 4000, South Africa,
gwetum@ukzn.ac.za

Abstract. Although Random Forests (RFs) are an effective and scalable ensemble machine learning approach, they are highly dependent on the discriminative ability of the available individual features. Since most data mining problems occur in the context of pre-existing data, there is little room to choose the original input features. Individual RF decision trees follow a greedy algorithm that iteratively selects the feature with the highest potential for achieving subsample purity. Common heuristics for ranking this potential include the gini-index and information gain metrics. This study seeks to improve the effectiveness of RFs through an adapted gini-index splitting function and a feature engineering technique. Using a structured framework for comparative evaluation of RFs, the study demonstrates that the effectiveness of the proposed methods is comparable with conventional gini-index based RFs. Improvements in the minimum accuracy recorded over some UCI data sets, demonstrate the potential for a hybrid set of splitting functions.
Keywords: random forest, gini-index, feature engineering, feature coherence, circularity.

1 Introduction

Legacy supervised machine learning algorithms that are commonly used in pattern recognition tasks include Bayesian Networks, Neural Networks, Support Vector Machines, k-Nearest Neighbours and Decision Trees (DTs) [16]. Although sustained research into each of these individual classification and regression algorithms has led to improved effectiveness, it is widely accepted that ensemble methods generally out perform them [26]. Ensemble methods such as bagging, boosting and Random Forests (RFs) combine the individual outputs of several base learners into a more reliable aggregate committee decision [26]. RFs are ensembles of DTs generated from a bootstrapped training data set [6]; they have gained cross-disciplinary popularity due to their high accuracy rates and simple interpretation [32].

The history of RFs can be traced back to the first breed of DTs: CART [7], ID3 and C4.5 [21], which use a common recursive divide-and-conquer approach to partition the training data set until class homogeneity is achieved. These DT types differ in terms of splitting criteria, types of attributes allowed, type of output provided (regression and/or classification), support for missing values, tree

pruning strategy and ability to detect outliers [25]. Tree pruning is normally required to reduce the problem of overfitting the classification model on the given training instances. Although the concept of bagging was one of the first techniques to combine outputs from several random DTs, the resultant trees were found to be highly correlated [26]. RFs seek to improve the effectiveness of bagging by reducing the resemblance between trees and thus, simultaneously reduce variance and bias errors. Bias error reflects how inaccurate, learned models are at capturing the significant trends in the training data while variance error captures how monotonous the models are at labeling training instances [18, p. 311]. Simultaneously low, bias and variance errors, are desirable in ensemble classifiers as they indicate that individual base classifiers are highly accurate but diverse[1].

An alternative DT-based ensemble technique is boosting, which aims to improve weak learning algorithms through a committee method that gives more focus to incorrectly classified instances [14, 26]. This is done by specifically assigning weights to: 1) individual classifiers in the committee based on training set error and 2) misclassified instances in the training set in order to increase their influence in the next iteration. Boosting has however been found to be less popular than RFs and bagging due to its lack of consistency and low convergence likelihood [26, p. 150]. Furthermore, RFs offer greater opportunity for parallel execution than boosting which has sequential iterations that are dependent on their predecessors.

In DT induction, the criteria used to decide which attribute is the most suitable for partitioning the data set portion at each node, is crucial for achieving high classification accuracy. A study by Raileanu et al. [22] sought to theoretically analyze the two most commonly used splitting criteria/metrics: the gini-index and information gain functions in order to solve the general problem of selecting the most suitable criteria for a given data set. Their findings revealed that the frequency of disagreement between the two criteria is only 2%, thus confirming previously published empirical results which assert that it is impossible to decide on which of the two tests is preferable [22]. This however does not mean, the two criteria can not be optimized individually, neither should it preclude the search for other criteria that may be more effective. Although RFs are an elegant and effective classification technique, there is room for achieving locally optimal attributes [23] and a need for capturing other attribute relations besides conditional interactions [32].

Feature engineering/construction refers to the common practice of creating new features from an existing feature set in a bid to assist classification algorithms to better distinguish between similar previously encountered instances [15]. It is a part of the broader topic of data representation, which seeks to unravel more informative perspectives of a data set by mainly using dimensionality reduction methods such as Linear Discriminant Analysis (LDA) and Principal Component analysis (PCA) [2]. In practise, the new augmented feature set from feature engineering may be used as is or subsequently complemented by a feature selection stage which identifies the most suitable subset of features. Typical

[1] The few misclassifications that individual classifiers make, are in different contexts.

feature engineering strategies include computations such as rule based conjunctions, rational differences and polynomial relations based on existing features [15].

This study proposes an optimized gini-index metric and a shape based feature engineering technique as a means towards improving the effectiveness of RFs. A steepend gini-index function is used to replace the conventional gini-index function in order to induce a preferential bias towards probabilities that suggest purity. A novel feature coherence model, based on the shape of a synthetic radial feature contour, is proposed for injecting new attributes to reflect general feature correlation within an instance. The specific question that the study seeks to answer is whether this steepened gini-index splitting function and the proposed shape feature injection can improve the effectiveness of RFs.

The remainder of this paper is structured as follows. Section 2 outlines the RF algorithm in detail and reports on previous work centred on its optimization. Section 3 explains the proposed new methods while Section 4 describes the framework used for experimentation. The results of the study are presented in Section 5 then Section 6 concludes the study and looks at proposed future work.

2 Random Forests

Although RFs also use sampling of training instances with replacement (bootstrapped sampling) like bagging, they introduce additional stochastic behaviour by choosing a random set of predictors (attributes or features) without replacement at each DT node. The conventional RF algorithm (commonly referred to as Forest-RI) can be formally described by Algorithm 1; alternative descriptions can be found in [6, 14, 26].

The maximum permissible purity of DT nodes, along with the parameters ns and d, can be used as constraints for limiting the size and sensitivity of each DT in the RF. The purity of a node is the highest proportion of any class present in its sample. A choice can be made between the constraints imposed by parameters ns and d, since either constraint can effectively limit tree depth, albeit though different criteria. The most commonly used values in literature for the parameter m are 1, $sqrt(M)$ and $log_2(M) + 1$ [3]. The sample size, n is normally set to N, the size of the training set [6], to yeild a sampling ratio of 1. In each of the $Ntree$ iterations, a DT is induced based on the given inputs; thus creating a forest of DTs, $\{T_t,\ t = 1, \ldots, Ntree\}$.

Each DT, T_t can be considered as a classifier, $\{h(x, \theta_t)\}$ where θ_t is a set of independent but identically distributed stochastic vectors and x is a new instance to be classified [6]. θ_t is determined during induction by the set of random samples and features chosen with and without replacement respectively. An individual DT classifier is considered to have perfectly mastered its training set if $h(x_i, \theta_t) = y_i \ \forall \ (x_i, y_i) \in S_T$. The algorithm terminates when L_{nt} is empty; at this point all leaf nodes in a given DT are labelled with one of the C possible classifications based on the majority class within their node's sample. For a

Algorithm 1 The Forest-RI Algorithm

Input: $S_T = \left\{ (x_i, y_i) \;\middle|\; \begin{array}{l} i = 1, \ldots, N, \\ x_i = \{x_{i_j} \mid j = 1, \ldots, M\}, \\ y_i \in \{1, \ldots, C\} \end{array} \right\}$, where S_T is the training set.
$m \leq M$, where m is the number of features to select.
$n <= N$, where n is the sample size.
$n_s <= n$, where n_s is the minimum node size.
$d \in \mathbb{N}$, where d is the maximum tree depth.
$Ntree \in \mathbb{N}$, where $Ntree$ is the number of trees in the forest.
Output: A forest $\{T_t,\ t = 1, \ldots, Ntree\}$ of DT classifiers
for $t = 1, \ldots, Ntree$ **do**
create a bootstrapped sample, S_t by randomly selecting (with replacement) n elements from S_T such that $|S_t| = n$.
create a root node to a DT, T_t based on S_t.
add the root node to the set, L_{nt} of non-terminal leaf nodes of T_t.
while L_{nt} is not empty **do**
let $l_{nt}.depth$ and $l_{nt}.sample$ represent the depth of l_{nt} in T_t and the sample associated with l_nt respectively, where l_{nt} is a non-terminal leaf node in L_{nt}. Calculate $l_{nt}.depth$, $|l_{nt}.sample|$ and $purity(l_{nt}.sample)$.
if $|l_{nt}.sample| < n_s$ or $purity(l_{nt}.sample) == 1$ or $l_{nt}.depth > d$ **then**
calculate $l_{nt}.class$ as the majority class of $l_{nt}.sample$.
remove l_{nt} from L_{nt}.
else
choose a random feature set $\{f_j, 1 \leq j \leq M\}$ of size m, without replacement.
find the best feature, $f_b \in \{f_j\}$ for splitting $l_{nt}.sample$.
split l_{nt} into l_{nta} and l_{ntb} using f_b.
remove l_{nt} from L_{nt}.
add l_{nta} and l_{ntb} to L_{nt}.
end if
end while
end for

given test instance, a RF solicits predictions from each of its $Ntree$ DTs and the ensemble prediction is usually determined by a majority vote.

Although the gini-index was used in the Forest-RI algorithm to determine the attribute value yielding the best split, it has since been found to be weak at identifying strong conditional associations among features [17]. A study by Robnik and Sikonja [23] sought to improve the effectiveness of RFs by using several attribute evaluation measures instead of just one, then aggregating DT votes using the margin achieved on similar out-of-bag instances as a weight. The evaluation heuristics used were: gini-index, gain ratio, Minimum Description Length (MDL), ReliefF or Myopic ReliefF; with each heuristic being applied to its fifth of the trees in a RF. A slight increase in effectiveness is observed when comparing the use of five heuristics against the Gini index alone. This improvement is especially visible on data sets with strong feature dependencies and is attributed

to the use of the ReliefF algorithm which manages to decrease DT correlation while retaining prediction strength. A more significant improvement in RF classification accuracy is achieved across several data sets when adopting the new voting strategy.

The pioneering study on RFs by Breiman [6] also proposed a variation in the induction of RFs by using random linear combinations of inputs, a procedure known as Forest-RC. At any given node, L existing numerical features are randomly chosen and added together using random weights in the range [-1,1] to form a new feature. F such features are generated and the best splitting condition is chosen from the range of all possible feature values. Experiments on 19 data sets using $L = 3$ and $F = 2$ or 8 show that although Forest-RC is generally more comparable to Adaboost than Forest-RI, it is not necessarily superior.

Due to the multiple steps in the Forest-RI algorithm, there are several options for improving performance and effectiveness. Improvement in performance can be facilitated through an implementation that exploits parallelization opportunities presented by modern multi-core processors and GPUs like the FastRF, LibRF and the CUDARF algorithms [12]. Improvement in RF effectiveness is generally facilitated by creating a committee of diverse but highly accurate tree based classifiers; a comprehensive survey of such RF variants can be found in [17, 11].

3 Proposed Methods

This study aims to improve the effectiveness of RFs by using an alternative splitting function and a feature construction technique that captures inter-feature relationships. The foundations for these two concepts are elaborated below.

3.1 Split functions

In the original Forest-RI algorithm, the gini-index[7] is used to obtain the best splitting criteria from the random subset of features that represent the instances internal to each DT node. Since the gini-index is a measure of impurity, it can be used to estimate how well a given splitting condition separates the instances within a node into their different classes. The gini-index (also known as gini impurity[2]) was proposed by Brieman et al. [8] as a splitting criteria for DTs known as Classification and Regression Trees (CART). Although 4 other splitting criteria (symmetric Gini, twoing, ordered twoing and class probability) were also proposed, the Gini index generally performs best. Other impurity measures that can be used as alternatives to the gini-index include entropy and classification error [30]. Because the CART algorithm is generally used to build binary DTs, it is more applicable to binary, numerical and ordinal attributes as their values can be partitioned into 2 groups. For numerical attributes, all possible values of each feature are evaluated as possible thresholds for splitting a node and the value that yields the highest gini-index is chosen.

[2] Higher scores are achieved on impure data sets, so it can be seen as measuring impurity.

The Gini index of a DT node can be formulated as follows [11]:

$$Gini(S) = 1 - \sum_{i=1}^{C} p(i)^2, \tag{1}$$

which is equivalent to:

$$Gini(S) = \sum_{i=1}^{C} p(i) * (1 - p(i)), \tag{2}$$

where S is the sample of instances in a node and C is the number of class labels in the data set. If all the classes in the data set are enumerated from 1 to C, then $p(i)$ is the probability of the i^{th} class in a particular node. Likewise the entropy and classification error metrics can be formulated as in Equations 3 and 4 respectively [24, 30].

$$E(S) = -\sum_{i=1}^{N} p(i) * log(pi). \tag{3}$$

$$CE(S) = 1 - \max_{1 \leq i \leq C} p(i). \tag{4}$$

When considering the viability of an attribute splitting condition or value, the chosen impurity metric is calculated for each potential child node resulting from such a split and normalized using the probability of the child node. These normalized total impurity values are then summed up to represent the combined impurity for the splitting condition in question. The splitting condition yielding the lowest normalized total impurity is potentially the best condition as it corresponds to the highest purity in the given context. The general approach in literature is to use the impurity gained at a particular child node relative to its parent as opposed to the absolute impurity of the child node [31, 23]. The generic formulation of this approach for all impurity based measures in the context of binary DTs is as follows [23]:

$$\Delta impurity(a_v) = impurity(S_0) - \sum_{j=1}^{2} p(S_j|a_v) * impurity(S_j|a_v), \tag{5}$$

where a is the attribute to split on using a_v as the specific condition or value, S_0 is the parent node and S_j is one of the two child nodes. $\Delta impurity$ is normally referred to as impurity gain; when applied to the gini-index it is referred to as gini gain. Despite the availability of other gain splitting functions such as Information Gain and Gain Ratio, the Gini Gain was chosen by Breiman for use in RFs because of its simplicity and effectiveness [23]. Information Gain and Gain Ratio are both based on entropy, with the latter being a normalized derivation of the former. The complexity of Information Gain and Gain Ratio arises mainly from their logarithmic computations.

The approach adopted towards formulating alternative impurity measures was to visually analyse the behaviour of existing metrics (gini-index, entropy and class error) in the context of different probabilities. The assumption made was that the gini-index and entropy were superior to class error [8, 30], with the gini-index being preferred primarily because of its computational simplicity [22]. The task was then to identify or formulate alternative functions that had a similar shape to the gini-index and entropy functions for input in the range [0,1]. Two proposals were made: the Gaussian and Steepened Gini Index (SGI) functions, with the latter proving more effective than the former.

Gaussian Impurity Function The Gaussian function was proposed due to its graph which is a symmetric bell curve shape that is similar to the graph of the gini-index. It is widely used in statistics to describe normal distributions and can be formulated as follows [13]:

$$G(x) = a * e^{-(\frac{(x-b)^2}{2c^2})}, \tag{6}$$

where a is the amplitude, b is the position of the peak within the bell shape and c is the standard deviation which controls the spread of the bell shape. Since the input of an impurity measure is a probability, this Gaussian distribution is over input values between 0 and 1. In this study, the parameter b was set to 0.5 since probabilities at the tails of the distribution are generally indicative of lower impurity while those in the middle are more likely to correspond with diversity. In a statistical normal distribution, about 99.7% of the data values lie within three standard deviation [9], hence the standard deviation is one third of the distance from the mean to either of the tail ends. We thus set parameter c to 0.1667 (which is $\frac{0.5}{3}$). After considering a few options (0.125, 0.25, 0.5, and 1), the parameter a was set to 0.5 as this value seemed to yield higher accuracy rates on the sonar UCI data set[3] [19]. The resulting Gaussian impurity function, shown in Equation 7, simply replaces class probabilities with corresponding Gaussian distribution outputs.

$$Gaussian(S) = \sum_{i=1}^{C} G(i). \tag{7}$$

Figure 1 shows the behaviour of the Gini index, entropy, classification error, SGI and the proposed Gaussian metric for probabilities encountered in two-class nodes of a dichotomous DT[4]. It can be observed that all metrics are:

1. symmetric, showing that a $p(i)$ vs $1 - p(i)$ node class split has the same impurity as its inverse, $1 - p(i)$ vs $p(i)$;

[3] Since this data set was used for parameter tuning, final evaluation is mainly based on other data sets to ensure an unbiased experimental context.

[4] Each node has at most two child nodes and each node has at most two classes. The permutations of nodes with more than two classes were not explored due to the computational overhead of computing them.

2. maximized at probability 0.5 when classes are equally represented within a node and,
3. minimized for pure nodes which are represented at the tail ends where all metrics give an impurity value of 0 except the Gaussian metric which slightly deviates from the trend with a value of 0.011.

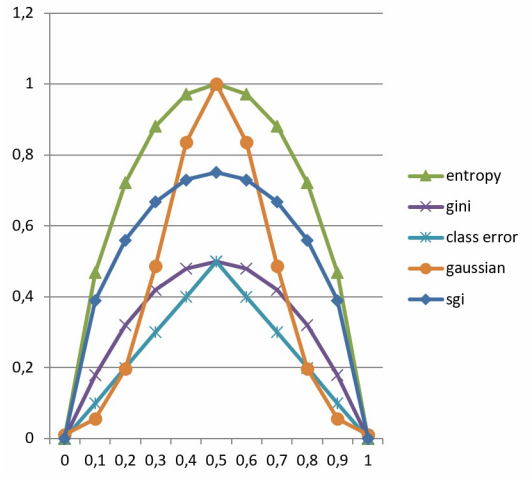

Fig. 1. Impurity functions for two-class node distributions

The Gaussian metric was anticipated to emphasize the difference between the impurity of probabilities in the middle of the distribution and that of those towards the tails; this was expected to favour splits where one of the classes is clearly dominant. Although preliminary experiments on the sonar data set showed the Gaussian metric to be comparable with the gini-index, the former was consistantly lower than the latter. Hence, the Gaussian metric was not explored in further experiments and an alternative was sort.

Steepend Gini Index Our interpretation of the inferiority of the Gaussian metric to the gini-index is based on the fact that split scores are determined using the relative as opposed to absolute impurity of a node. This highlights the importance of comparing the metrics based on gradient in order to model the concept of relative impurity. Since nodes are expected to gain in purity as we move down a DT, it is proposed that a desirable metric would be one that has a steeper negative impurity gradient towards the leaves of the DT. At this stage, a split that leads to greater node purity should be favoured since this node is unlikely to undergo further purification.

This reasoning led us to explore the possibility of modifying the gini-index function such that it yields a steeper gradient towards the two ends of its symmetric shape. After a few attempts at adapting the gini-index function in order to achieve this behaviour, the following function was adopted:

$$SGI(S) = \sum_{i=1}^{C} \frac{p(i) * (1 - p(i)) + \sqrt{p(i) * (1 - p(i))}}{2}. \tag{8}$$

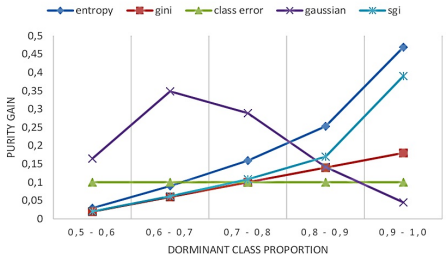

Fig. 2. Gradual change in purity for two-class node splits

Figure 1 shows that the SGI metric is indeed a steepend version of the gini-index. Figure 2 simulates the situation where the dorminant class within a node consistantly achieves an increase of 10% in purity as we move down a DT. The root node is assumed to have an equal proportion of two classes while the leaf nodes are pure. From this simulation it is evident that although the Gaussian metric has a similar shape to other metrics, the same can not be said for its gradient function. The gini-index is observed to maintain a consistant gradient while the SGI tries to initially mimick this consistancy but then steepens towards the end. Although entropy appears to have similar behaviour to SGI, it has an initially less consistant gradiant change than SGI and is less steeper at the final node transformation.

3.2 Shape Based Feature Engineering

It was initially envisaged that the adoption of the SGI metric would be enough to yield a significant improvement in the effectiveness of RFs; previous literature however seems to suggest that alternative impurity measures alone may only provide minor improvement [22]. Robnik [23] observes that although the gini-index offers good performance, it evaluates attributes separately and does not take attribute inter-relationships into account. The ReliefF measure was then proposed as a solution for alleviating misclassifications due to high feature interactions. Our work explores feature engineering as an alternative approach for capturing important information from multiple features simultaneously.

Feature engineering/construction entails transforming a given input feature set in order to give a more adequate representation of instances during the training and testing of machine learning models [27]. In some cases, the generated set of new features is deliberately smaller than the original feature set for improved computational efficiency and in order to remove irrelevant features. Examples of such approaches include clustering, LDA and PCA; which are also used for data compression [27]. In other cases, the size of the set of generated features is not constrained and may even be bigger than that of the original feature set. Examples of such approaches in the context of DTs include the FRINGE [20] and Forest-RC [6] algorithms which used Boolean operators and linear combinations respectively to generate a new set of composite features.

Instead of constructing new features that are each based on a selected subset of the original input feature set, this study proposes the formulation of new features that capture the level of coherence between all the feature values in a given instance. The underlying assumption is that if such properties can be empirically quantified, they could be used as extra features for improved class discrimination. This proposed formulation is inspired by the statistical radar/spider chart, which graphically displays multivariate data using a two-dimensional chart of multiple numerical variables represented on a radial axes with a common starting line [29]. Normalized input attributes are allocated fixed orientations from the centre using the order provided by the data set and the contour produced by a given instance is used to characterize its level of feature coherence. These contours can then be analyzed using existing shape descriptors; at this stage only the circularity property has been used as the viability of this concept is still being explored. A circularity score is normally deduced by measuring the perimeter of a closed contour as well as the area of the region it encloses, then computing [5]

$$\frac{perimeter^2}{4 * \pi * area}. \tag{9}$$

The fact that the axes of radar charts are numerical, restricts the application of this method to data sets with only numerical feature values. A hypothetical radar chart is shown in Figure 3 as an illustration of how normalized feature values can be used to plot radial graphs. The main conclusion that can be drawn from the plots is that spherical contours should be more indicative of greater feature coherence than jagged shapes; it is also expected that instances from the same class should have similarly shaped radial graphs. In cases where instance feature values are in the range [0,1), the generated shapes were observed to be generally the same. This effectively meant that there was no improvement in instance differentiation after adding circularity as an extra feature, especially for data sets with few attributes. This problem was alleviated by scaling the normalized values to the range [0,10). It was envisage that there would be a need for the normalization of feature values so as to reduce the bias of circularity towards any one feature. Indeed, experiments on the sonar data set showed an improvement in classification effectiveness when the injected circularity score was calculated

from normalized feature values. The original feature values were however, left un-normalized to avoid distorting the data sets.

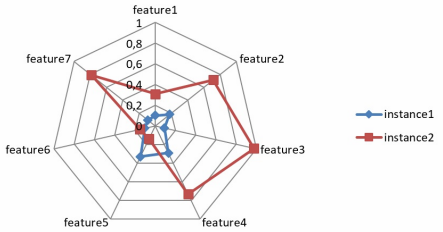

Fig. 3. Hypothetical feature coherence example. Instance1 has greater circularity than Instance2

4 Experimental Protocol

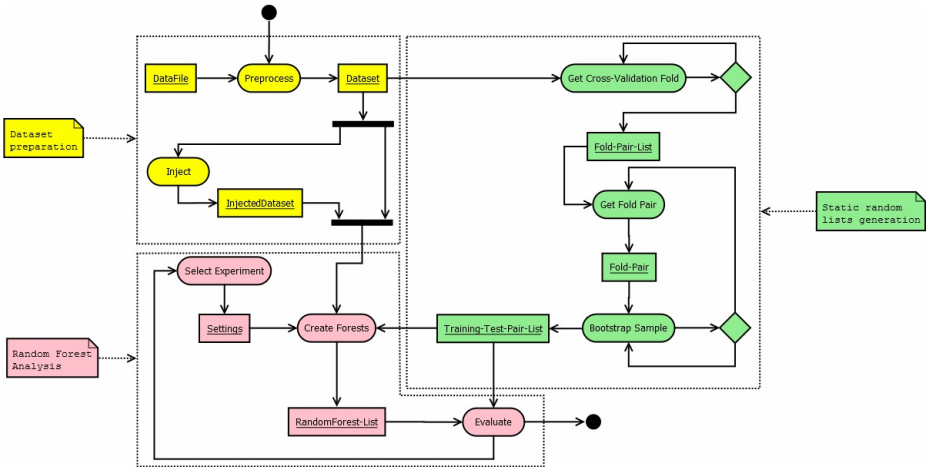

Fig. 4. Experimental Design

The main of objective of this study is to improve the effectiveness of RFs through the use of a SGI feature evaluation heuristic and a shape-based feature set characterization method. In order to ascertain the effectiveness of these two techniques on RFs, four experiments are conducted in line with the experimental

design shown in Figure 4. The remainder of this section ellaborates on the sub-components of this design.

4.1 Dataset Preparation

The effectiveness of the methods proposed in this study is tested using 10^5 data sets from the UCI repository [19]. These data sets are drawn from Robnik [23] and Breiman's [6] studies on RFs; we exclude data sets with nominal attributes and missing values as these properties are beyond the scope of the present study. An additional constraint is enforced to exclude data sets with more than 3000 instances, for computational reasons. The characteristics of the chosen data sets are summarized in Table 1, which reveals the diversity of the problems represented, in terms of data set size (N), number of features (M) and number of classes (C).

Table 1. UCI Datasets

Dataset	N	M	C	**Dataset**	N	M	C
bupa	345	6	2	iris	150	4	3
ecoli	336	7	8	segmentation	2310	19	7
german-numeric	1000	24	2	sonar	208	60	2
glass	214	9	7	vehicle	846	18	4
ionosphere	351	34	2	yeast	1484	8	10

Each of the chosen data sets is preprocessed by converting all feature values to floating points and class labels to integers, then saved in a consistant format. The output of the data preparation phase is two data set files: one with the original feature set only and another with the circularity feature included.

4.2 Static Random List Generation

Given the highly stochastic nature of RFs, it is imperative to ensure a significant level of contextual consistancy in their induction and evaluation in order to minimize the effect of uncertainty on the outcome of comparative analyses. A deliberate effort is made to subject each RF variant taking part in a comparative experiment, to the same training and testing instances. This is accomplished by generating static/fixed lists of cross validation folds and training samples, before any tree induction or experimentation takes place.

The lists that are generated only contain indexes of instances in the data set, for the sake of efficiency. One random number generator seed is used to produce an entire collection of static random lists. This collection is subsequently used to ensure RF training and testing consistancy in four evaluation experiments. The random number generator is reset at the start of each experiment to enable

[5] The sonar data set is used for parameter tuning.

flexible reenactment of all 4 experiments. The same results can be reproduced repeatedly regardless of the order in which the individual experiments are conducted.

Although the adopted experimental design (including the RF algorithm), offers several opportunities for parallelization on multi-core processors, this generally comes at the expense of a deterministic execution cycle since parallel loop iterations are typically non-deterministic and may differ from run to run [4, p. 670]. In our case, we require loop iterations to follow a specific order of execution in order to ensure that experimental results can be reproduced verbatim. As a result, no parallelization is implemented in this study. This unfortunately forces us to cap the size of data sets to be investigated, in order to avoid computational overhead.

4.3 Random Forest Analysis

In line with the methodology adopted by Robnik [23] we execute all experiments under the following settings: 1) The recommended RF parameter values are used: the number of trees, $Ntrees = 100$; the number of attributes randomly chosen at each DT node, $m = sqrt(M)$ [6]; and the cut-off node size, n_s =5. 2) All data sets are evaluated using 10-fold cross-validation. The following additional default parameter values are used: a sampling ratio of 1 and a maximum tree depth, d equal to n.

The four experiments conducted, test the effectiveness of the standard Forest-RI algorithm using either the gini or SCI impurity measure, with and without shape feature injection. We refer to each of these four modifications to the Forest-RI algorithm as RF variants. The use of 10-fold cross validation means that each experiment trains and tests 10 RFs. The accuracy of a RF is calculated as the percentage of test set instances that it correctly classifies. To capture the overall effectiveness of a particular RF approach, we record statistics such as the minimum, maximum, mean and median from the 10 fold accuracies. The code to facilitate all these experiments was implemented in C++, with the opencv library being used for shape feature calculation.

The output of the four experiments is used to compare the effectiveness of the Forest-RI algorithm in the following five contexts: 1) using gini impurity with and without shape feature injection, 2) using SGI impurity with and without shape feature injection, 3) using gini impurity vs SGI, both without shape feature injection, 4) using gini impurity vs SGI, both with shape feature injection and 5) using gini impurity without shape feature injection vs SGI with shape feature injection. We consider this comparison to be objective, since all Forest-RI variants are trained and tested on precisely the same instances.

Since preliminary experiments revealed that it was possible for results to differ on different runs of cross validation, it seemed appropriate to adopt repeated cross validation [33]. For each data set, the cross validations of four experiments

[6] M is the number of attributes used to represent each instance in the data set.

are repeated 30 times [28, 34], with a different seed in each case. To avoid ambiguity, we propose some terminology for refering to the statistics recorded in this study. For each complete run of 10-fold cross validation using one of the four experiments, we record the following fold accuracies (FAs): minimum-FA, maximum-FA, mean-FA and median-FA. After the 30 cross validation repeats, we calculate the cross validation accuracies (CVAs) for each of the four experiments. Specifically, minimum-CVA, maximum-CVA and mean-CVA are derived using the minimum minimum-FA, maximum maximum-FA and mean mean-FA respectively.

We ultimately seek to establish whether any comparative difference in effectiveness can be attributed to the proposed techniques under investigation or the stochastic properties of RFs and training vs. testing set splits. The Wilcoxon signed-ranks test [23, 10] is used to evaluate the statistical significance of such a difference in RF effectiveness. For each of the five effectiveness comparisons conducted, the null hypothesis assumes there is no difference in the performance of the two RF approaches in question.

5 Results

Table 2. Wilcoxon signed-ranks test p-values from comparing 30 repeated experiment pairs. p-values are classified as either significant at 0.05 level✔, significant at 0.1 level✔, or non-significant✘. +gini and -gini represent RFs using the gini-index with and without shape feature injection respectively. Likewise with steepend gini-index (sgi).

RF Pair \ **Dataset**	bupa	ecoli	german-numeric	glass	ionosphere	iris	segmentation	vehicle	yeast
-gini vs +gini	0.866✘	0.178✘	0.137✘	0.118✘	0.091✔	0.208✘	0.05✔	0.232✘	<0.01✔
-sgi vs +sgi	0.15✘	0.125✘	<0.01✔	0.021✔	0.851✘	0.08✔	0.268✘	0.551✘	0.838✘
-gini vs -sgi	0.461✘	0.144✘	0.732✘	0.187✘	0.035✔	0.779✘	0.315✘	0.316✘	<0.01✔
+gini vs +sgi	0.757✘	<0.01✔	0.018✔	0.407✘	0.155✘	0.333✘	0.025✔	0.078✔	<0.01✔
-gini vs +sgi	0.232✘	0.011✔	<0.01✔	0.275✘	0.028✔	0.067✔	0.834✘	0.202✘	<0.01✔

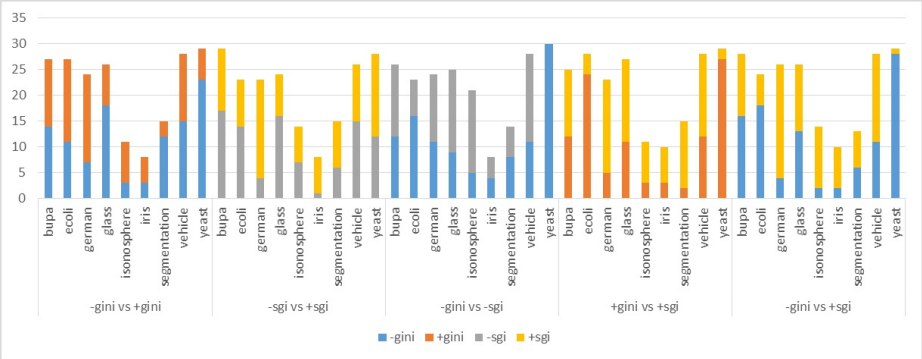

Fig. 5. Frequencies of median-FA superiority after 30 experiment repeats

Table 2 shows the p-values associated with the Wilcoxon signed-ranks test comparing pairs of median-FAs from 30 repeated experiments. Each pair either compares RFs based on splitting function or inclusion of shape feature injection; the training and testing sets used in both cases are however identical. We thus have two groups of 30 corresponding median-FAs and the p-values indicate the overall probability that the difference in corresponding median-FAs is due to chance. A lower p-value is indicative of a more significant difference in median-FAs of the two groups of RFs. We used this test to demonstrate whether the RF variants proposed in this study yield significant differences in effectiveness. Out of the 45 tests done, 5 and 13 tests were significant at an α level of 0.1 and 0.05 respectively. Although this means that in the majority of tests, the RF variants under comparison showed insignificant differences in effectiveness; we note that the +gini vs +sgi and -gini vs +sgi tests each recorded significant differences in 5 of the 9 data sets used. For the bupa data set all tests yielded insignificant differences while the yeast data set attained significant differences in all but the -sgi vs +sgi test.

Since the Wilcoxon signed-ranks test merely highlights the significance of differences in performance, we rely on median-FA comparisons to infer on the superiority of one RF variant over another. Figure 5 shows the number of times the median-FA of a RF variant was higher than that of its competitor over 30 cross validation repeats. Although no clear trend of superiority is demonstrated in the 9 data sets used, we note that the RF variants considered have strengths in different contexts. For example, the german-numeric data set seems to favour +gini over -gini, +sgi over -sgi, +sgi over +gini and +sgi over -gini; which is the exact opposite of the yeast data set context. For some data sets (for example isonosphere, iris and segmentation), most of the 30 experiment repeats yielded exactly the same median-FA while the remaining repeats favoured one RF variant.

[7] Results after parameter tuning.

Table 3. CVA statistics(%) over 30 repeats

CVAs \ Dataset		bupa	ecoli	german-numeric	glass	ionosphere	iris	segmentation	sonar[7]	vehicle	yeast
minimum-CVA	-gini	52.94	66.67	64	42.86	80	73.33	52.38	52.38	60.71	49.32
	+gini	55.88	66.67	65	33.33	82.86	80	66.67	55	61.18	50.68
	-sgi	47.06	66.67	64	47.62	80	73.33	66.67	55	60	45.27
	+sgi	44.12	63.64	65	42.86	80	80	66.67	57.14	63.53	43.24
mean-CVA	-gini	71.63	85.21	76.47	76.38	92.94	94.78	87.67	83.18	74.98	61.52
	+gini	71.36	85.43	76.97	74.9	93.17	95.02	87.38	83.25	74.73	60.96
	-sgi	72.45	84.63	76.69	76.8	93.55	94.84	87.75	84.08	75.01	59.49
	+sgi	70.95	84.57	77.26	75.48	93.53	95.2	87.73	84.36	75.07	59.28
	robnik[23]	71.90	86.60	75.80	78.10	94	96	98.10	84.10	74.60	61.40
	bader[1]	-	70.51	-	77.10	97.24	95.08	-	87.97	75.31	-
maximum-CVA	-gini	91.43	100	90	100	100	100	100	100	87.06	71.81
	+gini	88.57	100	91	100	100	100	100	100	87.06	70.95
	-sgi	91.43	97.06	91	100	100	100	100	100	88.24	72.48
	+sgi	91.43	100	89	95.45	100	100	100	100	89.29	70.47

Table 3 shows the range of FAs over the 300 (30 cross validation repeats, each with 10 folds) times RFs are trained and tested for each data set. In previous literature such as [23, 1], classification effectiveness is reported using the mean-FA of just one cross validation cycle. Although our mean-CVA results are shown to be comparable with accuracies in previous literature, this statistic alone does not give a comprehensive and reliable picture of RF performance. By reporting the minimum-CVA, maximum-CVA and mean-CVA, we give an indication of the worst, best and average performance of a given RF variant. The distribution of our mean-CVAs confirm the finding of a disagreement of only 2% between splitting criteria, made in previous literature[22]. However, the minimum-CVAs show a greater level of variance. For example, a difference of -14% is shown between the mean-CVAs of -gini and other RF variants.

6 Conclusion

This study sought to improve the effectiveness of RFs through the use of a steepend gini-index and shape feature injection. Although such improvements are indeed recorded over some data sets, the general trend is that of an insignificant difference in effectiveness. When considering the mean-CVA and minimum-CVA results of -gini vs +sgi, we note that the latter outperforms the former over more datasets; we therefore conclude that the steepened gini-index splitting function and the proposed shape feature injection can improve the effectiveness of RFs.

In addition to the proposed RF variants, a major contribution of this study is an experimental framework which allows for a high level of contextual consistancy and repeatability in the induction and evaluation of RFs. Previous studies such as [23, 1] have used the outcome of single runs of cross validation on multiple data sets as evidence of apparent algorithm optimization. We have argued that any claimed superiority should be demonstrated under highly controlled conditions that limit unnecessary stochastic variation, and sustained over multiple repetitions.

Over the course of this study, some opportunities for further work have been identified, we conclude by outlining some of these areas. The large differences in minimum-CVA over several data sets, highlight the potential of creating a RF that uses a hybrid of the RF variants considered in this study. In such a case, the hybrid RF would be equiped to deal with the varying level of complexity in different data sets. Additionally, the extreme weaknesses of one RF variant in some contexts could be compensated for by the better performance of another. Future work will focus on exploring this idea of a hybrid set of RF variants in conjuction with weighted voting. Since some of the minimum-CVAs may have been caused by unfavourable random cross validation splits, the use of stratified cross validation in future work may provide a slightly more controlled training and testing environment. A simple shape descriptor has been adopted in this study; extensions to this work may consider other more advanced shape characterization methods such as moments.

References

1. Bader-El-Den, M.: Self-adaptive heterogeneous random forest. In: Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on, pp. 640–646. IEEE (2014)
2. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)
3. Bernard, S., Heutte, L., Adam, S.: Forest-rk: A new random forest induction method. In: International Conference on Intelligent Computing, pp. 430–437. Springer (2008)
4. Bischof, C.: Parallel Computing: Architectures, Algorithms, and Applications, vol. 15. IOS Press (2008)

5. Bottema, M.J.: Circularity of objects in images. In: Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on, vol. 4, pp. 2247–2250. IEEE (2000)
6. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
7. Breiman, L., Friedman, J., Olshen, R.: Stone (1984): Classification and regression trees. Wadsworth, Belmont (1984)
8. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. CRC press (1984)
9. Carroll, T.A., Pinnick, H.A., Carroll, W.E.: Probability and the westgard rules. Annals of Clinical & Laboratory Science **33**(1), 113–114 (2003)
10. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research **7**(Jan), 1–30 (2006)
11. Fawagreh, K., Gaber, M.M., Elyan, E.: Random forests: from early developments to recent advancements. Systems Science & Control Engineering: An Open Access Journal **2**(1), 602–609 (2014)
12. Grahn, H., Lavesson, N., Lapajne, M.H., Slat, D.: Cudarf: a cuda-based implementation of random forests. In: Computer Systems and Applications (AICCSA), 2011 9th IEEE/ACS International Conference on, pp. 95–101. IEEE (2011)
13. Guo, H.: A simple algorithm for fitting a gaussian function. IEEE Signal Process. Mag. **28**(5), 134–137 (2011)
14. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning. 2001. NY Springer (2001)
15. Heaton, J.: An empirical analysis of feature engineering for predictive modeling. In: SoutheastCon, 2016, pp. 1–6. IEEE (2016)
16. Kotsiantis, S.B., Zaharakis, I., Pintelas, P.: Supervised machine learning: A review of classification techniques (2007)
17. Kulkarni, V.Y., Sinha, P.K.: Random forest classifiers: a survey and future research directions. Int J Adv Comput **36**(1), 1144–53 (2013)
18. Manning, C.D., Raghavan, P., Schütze, H., et al.: Introduction to information retrieval, vol. 1. Cambridge university press Cambridge (2008)
19. Newman, C.B.D., Merz, C.: UCI repository of machine learning databases (1998). URL http://www.ics.uci.edu/∼mlearn/MLRepository.html
20. Pagallo, G.: Learning dnf by decision trees. In: IJCAI, vol. 89, pp. 639–644 (1989)
21. Quinlan, J.: Building classification models: Id3 i c4. 5. Dane udostepnione pod adresem: http://yoda. cis. temple. edu **8080** (1993)
22. Raileanu, L.E., Stoffel, K.: Theoretical comparison between the gini index and information gain criteria. Annals of Mathematics and Artificial Intelligence **41**(1), 77–93 (2004)
23. Robnik-Šikonja, M.: Improving random forests. In: European Conference on Machine Learning, pp. 359–370. Springer (2004)
24. Rokach, L., Maimon, O.: Decision trees. In: Data mining and knowledge discovery handbook, pp. 165–192. Springer (2005)
25. Singh, S., Gupta, P.: Comparative study id3, cart and c4. 5 decision tree algorithm: A survey. International Journal of Advanced Information Science and Technology (IJAIST) Vol **27**, 97–103 (2014)
26. Siroky, D.S., et al.: Navigating random forests and related advances in algorithmic modeling. Statistics Surveys **3**, 147–163 (2009)
27. Sondhi, P.: Feature construction methods: a survey. sifaka. cs. uiuc. edu **69**, 70–71 (2009)

28. de Sousa, J.M., Pereira, E.T., Veloso, L.R.: A robust music genre classification approach for global and regional music datasets evaluation. In: 2016 IEEE International Conference on Digital Signal Processing (DSP), pp. 109–113 (2016). DOI 10.1109/ICDSP.2016.7868526
29. Tague, N.R.: The quality toolbox, vol. 600. ASQ Quality Press Milwaukee (2005)
30. Teknomo, K.: Tutorial on decision tree (2009). URL URL http://people.revoledu.com/kardi/tutorial/decisiontree
31. Timofeev, R.: Classification and regression trees (cart) theory and applications. Ph.D. thesis, Humboldt University, Berlin (2004)
32. Touw, W.G., Bayjanov, J.R., Overmars, L., Backus, L., Boekhorst, J., Wels, M., van Hijum, S.A.: Data mining in the life sciences with random forest: a walk in the park or lost in the jungle? Briefings in bioinformatics p. bbs034 (2012)
33. Vanwinckelen, G., Blockeel, H.: On estimating model accuracy with repeated cross-validation. In: BeneLearn 2012: Proceedings of the 21st Belgian-Dutch Conference on Machine Learning, pp. 39–44 (2012)
34. Zhou, S., Chen, Q., Wang, X.: Active deep networks for semi-supervised sentiment classification. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 1515–1523. Association for Computational Linguistics (2010)