Novel Decision Forest Building Techniques by Utilising Correlation Coefficient Methods

Efthyvoulos Drousiotis[1(✉)], Lei Shi[2(✉)], Paul G. Spirakis[3,4], and Simon Maskell[1]

[1] Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 3GJ, UK
{E.Drousiotis,S.Maskell}@liverpool.ac.uk
[2] Department of Computer Science, Durham University, Durham DH1 3DE, UK
lei.shi@durham.ac.uk
[3] Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK
spirakis@liverpool.ac.uk
[4] Department of Computer Engineering and Informatics, University of Patras, 26504 Patras, Greece

Abstract. Decision Forests have attracted the academic community's interest mainly due to their simplicity and transparency. This paper proposes two novel decision forest building techniques, called Maximal Information Coefficient Forest (MICF) and Pearson's Correlation Coefficient Forest (PCCF). The proposed new algorithms use Pearson's Correlation Coefficient (PCC) and Maximal Information Coefficient (MIC) as extra measures of the classification capacity score of each feature. Using those approaches, we improve the picking of the most convenient feature at each splitting node, the feature with the greatest Gain Ratio. We conduct experiments on 12 datasets that are available in the publicly accessible UCI machine learning repository. Our experimental results indicate that the proposed methods have the best average ensemble accuracy rank of 1.3 (for MICF) and 3.0 (for PCCF), compared to their closest competitor, Random Forest (RF), which has an average rank of 4.3. Additionally, the results from Friedman and Bonferroni-Dunn tests indicate statistically significant improvement.

Keywords: Decision forests · Tree-based learning · Ensemble learning · Classification · Machine learning

1 Introduction

Technological development has altered our approach to data management throughout the years. Data mining is currently being used for diverse datasets, aiming to discover hidden patterns and generate suitable predictions and/or descriptions. Data mining is set of techniques that extract hidden information such as patterns, correlations, or rules from massive data. Classification is highly

L. Iliadis et al. (Eds.): EANN 2022, CCIS 1600, pp. 90–102, 2022.
https://doi.org/10.1007/978-3-031-08223-8_8

essential in the field of data mining for both predicting the 'class' of an unknown instance and identifying trends in data. Furthermore, machines are increasingly being held accountable for societal decisions and various domains such as injustice [24], medicine [20], policing [19], and education [8,9], while the algorithmic transparency is an undeniable characteristic, they must have. Algorithms functioning as black boxes produce results and decisions that humans are eager to follow since they are proven to be helpful. Errors do exist and will continue to occur regardless of how much the underlying systems grow as more data becomes accessible to them, and more sophisticated algorithms learn from it. This awareness has given rise to either focusing more on more transparent algorithms such as decision trees and decision forests or trying to transparentise classical black-box algorithms such as Neural Networks.

Decision forests are a popular classification method, as they can learn the patterns in a dataset in an easy way that closely matches human thinking. Importantly, unlike other classifiers (e.g., neural networks, k-nearest neighbours, and support vector machines), decision forests can train on both categorical and numerical data [18], and generate human-interpretable knowledge [22], which enable them to increase their application domains further. Decision forests are considered among the fastest machine learning algorithms in terms of training, testing, and predicting. It comes as no surprise that improving classification accuracy on unknown data within the restrictions given by the training data is a desirable goal.

In this paper, we propose two novel decision forest building methods, i.e., the Maximal Information Coefficient Forest (MICF) and the Pearson's Correlation Coefficient Forest (PCCF). We aim to achieve a higher classification accuracy, than other famous variants of the decision forest algorithms, including Bagging, Random Subspace, Random Forest and Random Features Weights.

The rest of the present paper is organised as follows: Sect. 2 describes the related work. Our novel approach and the new algorithm are denoted in Sect. 3. Experimental results are drawn in Sect. 4 with a conclusion in Sect. 5.

2 Related Work

Many forest building methods have been proposed to produce more accurate and diversified trees by distinguishing the training dataset in various ways. As follows, we will examine several well-known algorithms.

Bagging: In Bagging [3], the dataset is randomly divided into a test set T and a learning set L. A new learning set L' is created randomly from the original learning set L, containing the same number of samples. Consequently, some samples in L may be selected several times and others may not be selected at all. This method of generating a new learning set is called bootstrap sampling. In bagging, bootstrap sampling is used to generate number ($|T|$) of bootstrap samples $L_1, L_2, L_3, ..., L_r$. Afterwards, a decision tree algorithm uses each bootstrap sample $L_i (i = 1, 2, 3..., |T|)$ to build ($|T|$) number of trees for the forest.

Random Subspace: The Random Subspace method [14] is also called attribute bagging and feature bagging. It attempts to reduce the correlation between individual weak learners in an ensemble by training them on random selection of a subset D' of features from the entire attribute space D. Features in D' can be drawn at both node level and tree level. When drawn at the tree level, features in D' continue to be the same for the tree, whereas when drawn at the node level, features D' vary from one node to another in a tree. Through any known decision tree algorithms such as CART [5], the best attribute in D' is calculated and determined to be the best splitting feature for the corresponding node.

Random Feature Weights: Random Feature Weights [17] is a tree ensemble construction method, where diversity is introduced into each individual tree using a random weight from a uniform distribution associated with each attribute. A weight stays the same for every node of a tree, while each tree acquires a different weight. In order to determine the best splitting feature at each node, merit values are calculated for each feature by multiplying their classification capacities such as Gini Index [5] by their respective random weights. Finally, the attribute with the highest excellence value is chosen as the splitting feature.

Random Forest [4] *(RF):* RF is considered to be among the state-of-the-art decision forest building algorithms, as it simply combines Random Subspace and Bagging algorithms where, in its simplest form, features D' are randomly selected at the node level. Despite all the variants of decision forests algorithms, RF is the most popular among the research community mainly because of its publicly availability through the sci-kit learn Python library[1]. Moreover, [6] compared 179 classification algorithms emerging from 17 learning families over 121 datasets where it concluded that forests, and specifically random forests, tend to outperform the rest of the classification algorithms. Those results indicate that any enhancement beyond Random Forest will have a substantial impact on its broad application scope.

Parallel Random Forest (PRF): PRF [16] is a modification of RF to be more suitable for 'big data'. A PRF algorithm is optimised using the MIC optimisation technique as a single splitting criterion. Firstly, each feature correlation capacity score is calculated through MIC, and then, according to the level of score, the features are divided into three groups: 'low', 'medium', and 'high'. Features fell in the 'low' group are discarded, and thus a new feature subset (D') with all the features from the 'medium' and 'high' groups is created. For each node, the splitting feature is chosen randomly from D'. A similar approach but for regression problems utilises MIC with information gain [13] as well, and it discards the low correlation features similarly to the PRF algorithm. Interestingly, they employ the roulette method so as to keep only the features with a high correlation capacity score.

Therefore, intending to provide an enhanced generic decision forest building technique, this study considers satisfying two splitting criteria (MIC and Gain Ratio, PCC and Gain Ratio) focusing on classification tasks as well as taking

[1] https://scikit-learn.org.

into consideration low correlation features as hidden patterns that may still exist. Moreover, in contrast to the latest trend of improving decision forest algorithms in a problem specific manner, we present two generic methods, which improves the overall predictive accuracy.

In general, our experimental result shows that MICF and PCCF are more balanced and accurate decision forest algorithms. In brief, we itemise the novel contributions of both algorithms as follows:

- Proposing a weight assignment strategy that works in favour of the features with the highest classification capacity, but it does not discard features with lower classification capacity.
- Proposing a double metric strategy (Gain Ratio and MIC, Gain Ratio and PCC), which determines the best feature and threshold on each node on classification problems.
- Proposing a weight assignment strategy that helps maintain the diversity among the individual decision trees.

3 Proposed Methods

We propose two methods that create subsets from the feature space of the whole original dataset using correlation capacity scores(MIC and PCC), resulting in a higher predictive accuracy. In this paper, to the best of our knowledge, it is the first time the MIC and Gain Ratio (for MICF) and the PCC and Gain Ratio (for PCCF) are combined as splitting criteria (impurity measure) to improve the overall accuracy of a decision forest classifier algorithm. Next, we present the splitting criteria and learning algorithms, including the two main functions for MICF and PCCF methods. The other steps of the algorithms are identical to existing decision tree building algorithms such as CART.

3.1 Splitting Criteria

Gain Ratio. The normalisation of the Information gain of an attribute against how much entropy that attribute has. Entropy (see Eq. 1, p_i is the probability of a data point in the subset of D_i of a dataset D) can be described as the degree of uncertainty or a measure of purity, and it is bounded between 0 and 1. The higher entropy the higher diversion in data, while our aim is to determine a split to create a purer distribution (close to 0) of class values in the succeeding partitions than the original dataset D.

Entropy plays an important role in estimating the Information Gain, which is used in ID3(the preliminary Decision Tree algorithm) [21] to determine the best features that provide as much information about a class as possible. The aim is to decrease the level of entropy, as it begins with the root node and progresses to the leaf nodes by computing the difference in entropy before and after the split (see Eq. 2, where $Entropy_{t-1}$ is the entropy before splitting and $Entropy_t$ is the entropy after splitting).

Gain Ratio normalises Information Gain of a feature based on the amount of entropy it has (see Eq. 3). As shown in Eq. 3, when entropy is low, the Gain Ratio will be high, and vice versa.

$$Entropy(P) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{1}$$

$$InformationGain = Entropy_{t-1} - Entropy_t \tag{2}$$

$$GainRatio = Information\ Gain\ / Entropy \tag{3}$$

Pearson's Correlation Coefficient. In statistics, Pearson's correlation coefficient is used to measure the statistical relationship or correlation among variables. It is based on the covariance matrix of the data to determine the strength of the connection between two vectors. Pearson's correlation coefficient between two vectors a_i and a_j is:

$$P(a_i, a_j) = \frac{cov(a_i, a_j)}{\sqrt{var(a_i) \times var(a_j)}} \tag{4}$$

where cov(a_i,a_j) is the covariance, var(a_i) is the variance of a_i and var(a_j) is the variance of a_j.

Maximal Information Coefficient. Maximal Information Coefficient (MIC) is a powerful approach to measuring the correlation between two features. MIC can deal with the correlation analysis of linear, nonlinear, and potential non-functional relationships in large datasets. The fundamental idea of MIC is that if a specific relationship exists between two features, a grid can be drawn on the scatter-plot to partition them. Then, it will be able to encapsulate the mutual information of the two features according to the approximate probability density distribution in the grid. MIC is calculated based on mutual information and the grid partition method. Given two independent features with n samples, $x = \{x_i | i = 1..., n\}$ and the target variable $y = \{y_i | 1, .., n\}$, a finite set $D = (x_i, y_i | i = 1, .., n)$ of ordered pairs can be obtained. Given a grid G, we can partition the x_i values of D into x bins and the y_i values of D into y bins. MIC is obtained according to the following equations:

$$MI(D, x, y) = \max MI(D|G) \tag{5}$$

where $MI(D, x, y)$ denotes the maximum mutual information of D over grids G. $D|G$ represents the distribution induced by the data points in D on the cells of grid G. The characteristic matrix of D is defined by the following equation:

$$M(D)_{xy} = \frac{MI(D, x, y)}{\log \min\{x, y\}} \tag{6}$$

The MIC of D with grid size less than $B(n)$ is defined as:

$$MIC(D)_{xy} = \max_{xy<B(n)} \{M(D)_{xy}\} \tag{7}$$

where $B(n)$ is the upper limit of the mesh division xy. In general, $B(n) = n^{0.6}$. MIC is normalised into a range $[0, 1]$. A higher MIC value indicates a stronger correlation between the variables/features.

First, we use the training dataset to calculate the MIC (in the case of MICF) and PCC (in the case of PCCF) (see Table 1) between x and y. We then normalise those values between 0 and 1, to create an interval for the features space $[0, 1]$. Features with stronger correlation (Linear correlation for PCCF; both linear and nonlinear for MICF) given the target variable having more possibilities to be selected uniformly.

When the tree grows, at each split node, we draw random uniform numbers between $[0, 1]$ equivalent to the number of features. Then, we remove the duplicated values, and we end up with a sub-sample of the training dataset. Thereafter, we calculate the Gain Ratio of the new feature space to determine the best split for the current node. In this step, we apply any existing decision tree algorithm such as CART. In our study, we examine the effectiveness of this particular method, specifically on classification problems. To better explain the algorithms and the functions, we explain the key functions below.

3.2 Function *FeaturesImportance*

Let D be the training dataset with d original features. Thus, the original feature space $A_o = A_1, A_2, ..A_d$. In this function, we use the training dataset D to calculate whether a feature d has a positive, neutral, or negative correlation, given the target feature either with PCC or MIC depending on the algorithm we test (see Table 4). At this point, in the case of PCC, we also calculate its absolute values, which range in $[-1, 1]$. Then, we normalise them (see Eq. 5) to be 0 to 1, so as to create an interval for the features space in the range of $[0, 1]$ (see Table 1). Features d with stronger correlation (linear correlation for the PCC; linear and nonlinear for MIC) given the target variable have more possibilities to be selected uniformly. Features Importance function returns an array with the features intervals (see Table 1).

3.3 Function *growTree*

The following steps happen every time we grow the tree until the stopping criteria are met. Providing the algorithm with the stopping criteria is crucial on individual decision trees and consequently on decision forests. On the one hand, the data are generally over-fitted if we continue to expand the tree until each leaf node equates to the highest Gain Ratio; and on the other hand, if splitting is halted too soon, the error on the training data is insufficiently large, and thus the performance suffers as a result of bias. As such, avoiding over-fitting

and under-fitting is crucial. In our case, we deployed the *maximum depth stopping criterion* and the *numbers of labels criterion*, where we check if the current node is homogeneous. We draw random uniform numbers equal to the number of features between 0 and 1, and we store the indexes of the d' features intervals they fall within (see Table 4), so we end up with an array containing less of the original feature space with $D' \subseteq D$. It is noteworthy that, at this point, we do not allow duplicate feature indexes as it does not make any difference in the final result. Then, we calculate the *Gain Ratio* of the new dataset feature space D' to determine the best split for the current node. This step, may apply any existing decision tree algorithm such as CART on the reduced space dataset D'. This particular decision tree can be used for classification based on the training dataset and prediction based on the testing dataset, which contains unlabelled samples. Using two splitting criteria (PCC and Gain Ratio or MIC and Gain Ratio), we promote the features with the best predictive ability without heavily biasing the algorithm (see Algorithm 1).

Table 1. Pearson's correlation coefficient example

Feature ID	Scores	Absolute values	Normalised scores	Thresholds
0	–0.255	0.255	0.322	0.322
1	0.075	0.075	0.094	0.417
2	0.143	0.143	0.180	0.597
3	–0.282	0.282	0.356	0.954
4	–0.035	0.035	0.045	1.000

4 Experimental Setup and Results

This section presents the proposed methods' experimental results in predictive accuracy and running time. The following results were obtained on an Intel(R) Core(TM) i7-10875H CPU @ 2.30 GHz (16 CPUs) processor, with 64 GB RAM and 16 MB CACHE memory.

In order to demonstrate the accuracy improvement of MICF and PCCF, we experiment on 12 widely known datasets that are publicly available through UCI Machine Learning Repository listed in Table 2. In particular, the median average number of records used is 354, with the lowest having 27 records and the largest 4,177. The median average number of features used is 9, with the lowest having 4 features and the largest 279.

For testing purposes, we generate 100 trees for each contending decision forest algorithm since the number is considered to be large enough to ensure convergence of the ensemble effect [2]. We apply majority voting to aggregate results for the forests. Moreover, we test the model with 10-Fold Cross-Validation to ensure that every observation from the original dataset has the possibility of appearing in the training and test sets. We perform hyperparameter optimization for

each dataset using Grid Search and Random Search techniques to ensure the best possible accuracies for each case. All the prediction accuracies reported in this paper are in percentage, and the best results are presented in bold-face. As MICF and PCCF are designed for parallel forest algorithms, for a fair evaluation, we compare them with other parallel forest algorithms, including Bagging (BG), Random Subspace (RS), Random Forest (RF), and two variants of Random Features Weights (RFW) with $p = 1$ and $p = 2$. Moreover, for consistency with RF, CART is utilised as the tree induction algorithm, and Gini Index is employed as the measure of classification capacity for every forest algorithm mentioned above. Finally, both versions of RFW ($P = 2$ and $p = 1$) are applied on bootstrap samples, as a better performance can be observed using this particular technique [17].

Generally, an essential aim for forest algorithms is to improve the Ensemble Accuracy (EA) [1]. As such, every single contending forest algorithm described in this paper aims to increase EA as their principal performance metric. Metrics such as Precision and Recall [23] are mainly and primarily used on imbalanced datasets, therefore evaluating PCCF and MICF as general purpose forest algorithms, we have not involved any imbalance dataset in our experimental evaluation. Table 3 presents the results on EA for all the contending algorithms while all datasets are taken into consideration. Results are presented in the shape of EA Rank, where EA is the Ensemble Accuracy in percentage for the algorithm in

Algorithm 1: Features Importance and Grow tree functions in algorithmic notation

Input: Training Dataset D with original attribute space Ao = {A1,A2,...,Ad}, number of features of the new Dataset D' where D'features < D features(NumFeat)

Output: A Decision Tree(T)

Function `Features_Importance(`D`):`

```
FeaturesImportance = Calculate the features importance(MIC or
 PCC) of Training Dataset D
FeaturesImportanceInterval = Create a features importance
 interval from 0 to 1 such as
 (F1 = 0.1, F2 = 0.3, F3 = 0.55, ..., Fn = 1)
return FeaturesImportanceInterval
```

Function `Grow_Tree(`$FeaturesImportanceInterval, NumFeat$`):`

```
for i until stopping criteria met do
UniformDraw = draw uniform numbers between 0 and 1
FeaturesIndex = get the features index using the uniform numbers
 using UniformDraw
D' = generate the dataset using the column index using
 FeaturesIndex
T = growTree using the D'
End for
return T
```

comparison, and EA Rank is the Ensemble Accuracy Rank for the corresponding algorithm to other contending algorithms according to the rank-ordering used in Friedman Test [12]. Amongst the 7 contending algorithms, the one with the highest EA is assigned an EA Rank of 1, the second highest as EA Rank of 2, and so on. Hence, the lower the EA Rank, the better the EA. In the case of a tie, we average the two or three or the number of algorithms having equal EA. Thus, for example, if two algorithms become the worst in EA, their EA Rank is calculated by $\frac{6+7}{2} = 6.5$. The last row of Table 3 shows the average EA and the average EA Rank in parentheses.

In Table 3, we present the EA percentage of the contenting algorithms for all 12 datasets considered. From Table 3 we observe that $PCCF$ provides the best EA on 1 dataset with an EA Rank of 3.0, and $MICF$ obtains the best EA on 11 datasets out of 12 with an EA Rank of 1.3. BG does not get any first place in an EA, resulting in an EA Rank of 5.4. RS obtains the best EA on 1 dataset with an EA Rank of 4.2. RF does not manage to get any first place in an EA resulting in an EA Rank of 4.3. $RFWp = 1$ obtains higher EA on 1 dataset with EA Rank of 4.3. Finally, $RFWp = 2$ does not manage to get any first place in an EA, resulting in an EA Rank of 5.5. The last row of Table 3 shows that $MICF$ achieves the best overall average performance based in an EA and EA Rank compared to all other contending algorithms.

Table 2. Datasets specifications

Dataset name	Number of records	Number of features
Abalone (AB)	4177	8
Arrythma (AR)	452	279
Balance scale (BS)	625	4
Dermatology (DER)	358	34
Glass identification (GI)	214	9
Ionosphere (ION)	351	34
Liver disorders (LD)	345	6
Lung cancer (LC)	27	56
Pima indians diabetes (PID)	768	8
SCADI (SCD)	206	70
Teaching assistant evaluation (TAE)	1515	5
Yeast (YST)	1484	9

Table 3. Ensemble Accuracy (EA) in percentage with Ensemble Accuracy Rank (EA Rank).

Dataset	PCCF	MICF	BG	RS	RF	RFW p = 1	RFW p = 2
AB	21.3 (7.0)	27.5 (1.0)	25.1 (2.5)	25.0 (4.5)	25.0 (4.5)	25.1 (2.5)	23.7 (6.0)
AR	81.2 (7.0)	82.8 (4.0)	81.6 (6.0)	83.7 (1.0)	83.0 (2.5)	82.6 (5.0)	83.0 (2.5)
BS	84.2 (1.5)	84.2 (1.5)	77.5 (6.0)	72.2 (7.0)	80.5 (5.0)	81.1 (4.0)	82.5 (3.0)
DER	96.0 (2.0)	96.7 (1.0)	88.5 (4.0)	89.0 (3.0)	87.0 (7.0)	87.5 (5.0)	87.3 (6.0)
GI	76.7 (2.0)	77.2 (1.0)	74.1 (3.5)	73.2 (5.5)	74.1(3.5)	73.2 (5.5)	72.2 (7.0)
ION	93.7 (3.0)	94.5 (1.0)	92.6 (7.0)	93.4 (5.0)	93.7 (3.0)	93.7 (3.0)	92.9 (6.0)
LD	72.1 (2.0)	73.6 (1.0)	68.7 (6.0)	69.8 (5.0)	71.5 (3.0)	71.0 (4.0)	67.3 (7.0)
LC	76.6 (2.0)	84.4 (1.0)	63.9 (7.0)	68.9 (4.5)	68.9 (4.5)	68.9 (4.5)	68.9 (4.5)
PID	76.2 (2.5)	77.9 (1.0)	75.6 (6.0)	76.2 (2.5)	75.9 (4.0)	75.6 (6.0)	75.6 (6.0)
SCD	83.7 (3.0)	84.3 (1.5)	80.0 (6.5)	82.9 (4.5)	80.0 (6.5)	84.3 (1.5)	82.9 (4.5)
TAE	60.6 (2.0)	62.4 (1.0)	53.6 (7.0)	59.5 (3.0)	56.3 (4.0)	54.3 (5.0)	54.2 (6.0)
YST	60.3 (2.0)	60.9 (1.0)	60.5 (3.0)	58.6 (5.0)	59.5 (4.0)	57.9 (6.0)	48.9 (7.0)
Average	73.9 (3.0)	**75.7 (1.3)**	70.1 (5.4)	71.0 (4.2)	71.4 (4.3)	71.3 (4.3)	69.9 (5.5)

In the following, we further examine the enhancement we achieved by performing statistical significance tests as recommended in [7]. First, we perform the Friedman test [11], which is a popular non-parametric test for examining various classifiers on multiple datasets. Friedman statistic is distributed according to Eq. 8, where k is the number of algorithms, and N is the number of datasets. As a generic rule, $k > 5$ and $N > 10$ must be hold. In Eq. 9, let r_i^j be the rank of the j_{th} of k algorithms on the i_{th} of N datasets. [15] suggests that Eq. 8 is undesirably conservative, and it derives a better statistical measure, as shown in Eq. 10:

$$x_F^2 = \frac{N}{nk(k+1)} \sum_{j=1}^{k} R_j^2 - 3n(k+1) \tag{8}$$

$$R_j = \frac{1}{N} \sum_{i} r_i^j \tag{9}$$

$$F_f = \frac{(N-1)x_F^2}{N(k-1) - x_F^2} \tag{10}$$

With 7 algorithms and 14 datasets, F_f is distributed according to the F distribution with 66 degrees of freedom, which is calculated using Eq. 11. The critical value of $F(6, 66)$ for $\alpha = 0.05$ is 2.24 and our value of F_f is calculated to be 8.7. As the critical value is lower than our F_f value, the *null hypothesis* is rejected, and so we can proceed with a post-hoc test, i.e., the Bonferroni-Dunn test [10] for detecting pairwise differences of EA Ranks between the controlled classifiers (MICF and PCCF) and the rest of the contending classifiers.

$$FD = (k-1)(N-1) \tag{11}$$

For the Bonferonni-Dunn test, we calculate the critical difference using Eq. 12 to compare our first novel algorithm (MICF) with the rest of the contending algorithms. In Eq. 12, q_a represents the bold number in the last row of Table 3 in brackets (EA Rank). The Critical Difference(CD) is calculated to be 1.14. At this point we observe that CD remains lower than the pairwise difference of EA Ranks between the classifier MICF and the other contending classifiers ($MICF\ vs\ BG$: 4.1, $MICF\ vs\ RS$: 2.9, $MICF\ vs\ RF$: 3.0, $MICF\ vs\ RFWp = 1$: 3.0, $MICF\ vs\ RFWp = 3$: 4.2). This indicates that MICF outperforms the rest classifiers in terms or EA, in a statistically significant manner.

Now, we repeat the same test for our second novel algorithm (PCCF) with the rest of the contending classifiers. The critical difference is calculated to be 1 and as before we observe that CD remains lower than the pairwise difference of EA Ranks between classifier PCCF and the other contending algorithms ($PCCF\ vs\ BG$: 2.4, $PCCF\ vs\ RS$: 1.2, $PCCF\ vs\ RF$: 1.3, $PCCF\ vs\ RFWp = 1$: 1.3, $PCCF\ vs\ RFWp = 3$: 2.5). This indicates that the performance improvement in PCCF in terms of EA is statistically significant.

$$CD = q_a \sqrt{\frac{k(k+1)}{6N}} \tag{12}$$

Table 4 contains correlation scores, which indicate the main differences between our two novel algorithms - MICF and PCCF. The results suggest that MIC can identify more accurately the correlation between the features and the target, as well as take into consideration the uncertainty, which results in a more accurate algorithm. Both MICF and PCCF algorithms outperform their competitors by achieving 4.4 and 2.6% more accurate results, respectively, compared with the third highest accurate algorithm, RF.

Table 4. Comparison between MIC and PCC scores.(left-PCC, right-MIC)

Feature space	AB		TAE		YST		GI		LD		BS	
1	0.009	0.071	0.323	0.079	0.071	0.208	0.056	0.132	0.166	0.081	0.242	0.242
2	0.155	0.204	0.417	0.275	0.191	0.389	0.227	0.252	0.344	0.330	0.485	0.485
3	0.306	0.341	0.598	0.559	0.221	0.515	0.481	0.428	0.408	0.505	0.742	0.742
4	0.452	0.471	0.955	0.647	0.357	0.764	0.685	0.552	0.694	0.642	1.000	1.000
5	0.593	0.604	1.000	1.000	0.499	0.857	0.736	0.626	0.960	0.887		
6	0.704	0.723			0.570	0.871	0.740	0.741	1.000	1.000		
7	0.836	0.855			0.174	0.903	0.740	0.825				
8	1.000	1.000			0.724	0.928	0.936	0.961				
9					1.000	1.000	1.000	1.000				

5 Conclusion

In this paper, we have proposed two novel decision forest building algorithms - Maximal Information Coefficient Forest ($MICF$) and Pearson's Correlation Coefficient Forest ($PCCF$). They combine Gain Ratio with PCC and MIC, which are used to determine the best feature on each splitting node. The larger the correlation score (either MIC or PCC), the greater the possibility of ending up on the newly created dataset D' at each splitting node. To the best of our knowledge, our work of combining Gain Ratio with MIC or PCC in classification problems is the first of its kind and can help improve the accuracy in classification problems significantly. The experimental results have shown that $MICF$ performs significantly better in Ensemble Accuracy than some highly esteemed existing algorithms, including Bagging, Random Subspace, Random Forest, Random Feature Weight, and the newly proposed $PCCF$ algorithm. Moreover, the generation of individual decision trees in both $MICF$ and $PCCF$ is in no way dependent on any previous tree(s) and, therefore, can be generated in parallel. Moreover, considering the consistent performance of both $MICF$ and $PCCF$, makes them a great fit within the big data context and thus enabling it to be used by non-technical individuals.

For future work, we aim to test our algorithms against imbalanced datasets and compare them with probabilistic trees, which are known to capture the uncertainty in the data effectively.

References

1. Adnan, N.: Decision tree and decision forest algorithms: on improving accuracy, efficiency and knowledge discovery (2017)
2. Bernard, S., Heutte, L., Adam, S.: Forest-rk: a new random forest induction method. In: ICIC (2008)
3. Breiman, L.: Bagging predictors. Mach. Learn. **24**, 123–140 (2004)
4. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2004)
5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees (1983)
6. Delgado, M.F., Cernadas, E., Barro, S., Amorim, D.G.: Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. **15**, 3133–3181 (2014)
7. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
8. Drousiotis, E., Pentaliotis, P., Shi, L., Cristea, A.I.: Capturing fairness and uncertainty in student dropout prediction – a comparison study. In: Roll, I., McNamara, D., Sosnovsky, S., Luckin, R., Dimitrova, V. (eds.) AIED 2021. LNCS (LNAI), vol. 12749, pp. 139–144. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78270-2_25
9. Drousiotis, E., Shi, L., Maskell, S.: Early predictor for student success based on behavioural and demographical indicators. In: Cristea, A.I., Troussas, C. (eds.) ITS 2021. LNCS, vol. 12677, pp. 161–172. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-80421-3_19

10. Dunn, O.J.: Multiple comparisons among means. J. Am. Stat. Assoc. **56**(293), 52–64 (1961)
11. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Am. Stat. Assoc. **32**(200), 675–701 (1937)
12. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. Ann. Math. Stat. **11**, 86–92 (1940)
13. Guo, Z., Yu, B., Hao, M., Wang, W., Jiang, Y., Zong, F.: A novel hybrid method for flight departure delay prediction using random forest regression and maximal information coefficient (2021)
14. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. **20**, 832–844 (1998)
15. Iman, R.L., Davenport, J.M.: Approximations of the critical region of the Fbietkan statistic. Commun. Stat.-Theory Methods **9**, 571–595 (1980)
16. Liu, S., Hu, T.: Parallel random forest algorithm optimization based on maximal information coefficient. In: 2018 IEEE 9th International Conference on Software Engineering and Service Science, pp. 1083–1087 (2018)
17. Maudes, J., Rodríguez, J.J., García-Osorio, C., García-Pedrajas, N.: Random feature weights for decision tree ensemble construction. Inf. Fusion **13**(1), 20–30 (2012)
18. Murthy, S.K.: Automatic construction of decision trees from data: a multi-disciplinary survey. Data Mining Knowl. Disc. **2**, 345–389 (2004)
19. Nasridinov, A., Ihm, S., Park, Y.H.: A decision tree-based classification model for crime prediction. In: ITCS (2013)
20. Podgorelec, V., Kokol, P., Stiglic, B., Rozman, I.: Decision trees: an overview and their use in medicine. J. Med. Syst. **26**, 445–463 (2004)
21. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**, 81–106 (1986)
22. Salzberg, S., Murthy, K.: On growing better decision trees from data (1996)
23. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Education India (2016)
24. Zeleznikow, J.: Using web-based legal decision support systems to improve access to justice. Inf. Commun. Technol. Law **11**, 15–33 (2002)