Systems Science & Control Engineering: An Open Access Journal, 2014
Vol. 2, 602–609, http://dx.doi.org/10.1080/21642583.2014.956265

Random forests: from early developments to recent advancements

Khaled Fawagreh, Mohamed Medhat Gaber* and Eyad Elyan

School of Computing Science and Digital Media, Robert Gordon University, Riverside East, Garthdee Road, Aberdeen AB10 7GJ, UK

(*Received 1 November 2013; final version received 16 August 2014*)

Ensemble classification is a data mining approach that utilizes a number of classifiers that work together in order to identify the class label for unlabeled instances. Random forest (RF) is an ensemble classification approach that has proved its high accuracy and superiority. With one common goal in mind, RF has recently received considerable attention from the research community to further boost its performance. In this paper, we look at developments of RF from birth to present. The main aim is to describe the research done to date and also identify potential and future developments to RF. Our approach in this review paper is to take a historical view on the development of this notably successful classification technique. We start with developments that were found before Breiman's introduction of the technique in 2001, by which RF has borrowed some of its components. We then delve into dealing with the main technique proposed by Breiman. A number of developments to enhance the original technique are then presented and summarized. Successful applications that utilized RF are discussed, before a discussion of possible directions of research is finally given.

Keywords: random forests; ensemble learning; bagging; supervised learning

1. Introduction

Data mining is an important and interesting field in Computer Science and has received a lot of attention from the research community particularly over the past decade. It is important because it specializes in analyzing the data from different perspectives and summarizing it into useful information – information that can be used to increase revenues, cut costs, or both. It is interesting because it applies methods at the intersection of multiple disciplines including artificial intelligence, machine learning, statistics, and database systems (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

As stated in Fayyad et al. (1996), data mining involves six common classes of tasks: anomaly detection, association rule learning, clustering, classification, regression, summarization, and sequential pattern mining. The relevant task to us in this paper is classification which organizes data into classes by using predetermined class labels. Classification algorithms normally use a training set where all objects are already associated with known class labels. The classification algorithm learns from the training set and builds a model, also called a classifier, as shown in Figure 1. The model is then applied to predict the class labels for the unclassified objects in the testing data as shown in Figure 2.

More formally, assume the training set is $X \in \mathbb{R}^{n\times m}$ and $y \in \{1, 2, \ldots, z\}$, where X represents the set of predictive attributes in the training set, m is the number of predictive attributes in the training set, n is the number of instances in the training set, and y represents the set of z possible class labels. $x^{(i)}$ is the ith instance in the training set, $y^{(i)}$ is the value of the class in the ith instance, and $x_j^{(i)}$ is the value of the jth attribute in the ith instance.

With the assumption that $x^{(i)} \mapsto y^{(i)}$, then the classification problem can be defined as finding $\hat{f}(x^{(i)}) \approx y^{(i)}$, $\forall x^{(i)} \in X$ and $y^{(i)} \in y$.

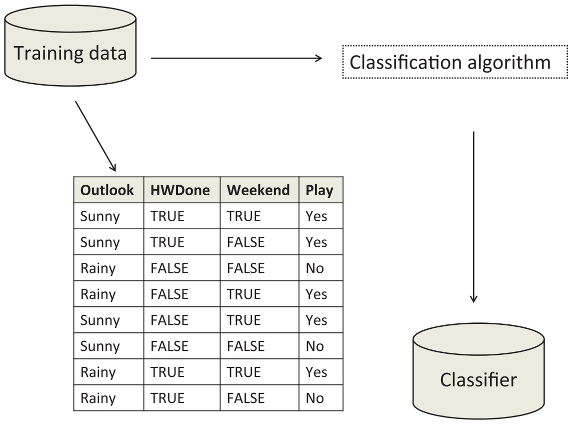
Referring again to Figure 1, we can find that $m = 4$, $n = 8$, and that, for example, $x_2^{(1)} = \text{TRUE}$. In this example, the X was categorized or by nature categorical. However, the classification problem is a general one that can deal with continuous values of the predictive attributes.

Typical applications of classification include (but not limited to) credit approval, marketing, and medical diagnosis. Success stories in these areas are too many to enumerate.

It has been well recognized that single classifier systems have limited performance (Yan & Goebel, 2004). To boost and improve the performance, ensemble classification, which is based on ensemble learning, has been used. Ensemble learning uses multiple models to obtain better predictive performance than could be obtained from any of the constituent models (Kuncheva & Whitaker, 2003; Polikar, 2006; Rokach, 2010). Likewise, ensemble classification employs multiple classifiers and then collectively uses them to identify unlabeled instances.

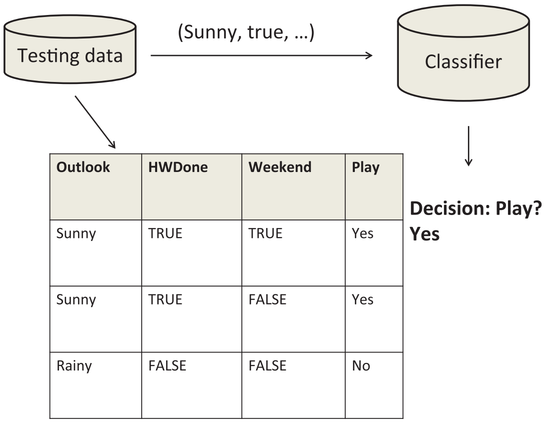
In this paper, we will look at developments of an ensemble classification technique called random forest (RF) from birth to present. Section 2 gives a brief background on ensemble classification. Early developments to RF are discussed in Section 3. Section 4 presents RF in

*Corresponding author. Email: mohamed.m.gaber@gmail.com



Outlook	HWDone	Weekend	Play
Sunny	TRUE	TRUE	Yes
Sunny	TRUE	FALSE	Yes
Rainy	FALSE	FALSE	No
Rainy	FALSE	TRUE	Yes
Sunny	FALSE	TRUE	Yes
Sunny	FALSE	FALSE	No
Rainy	TRUE	TRUE	Yes
Rainy	TRUE	FALSE	No

Figure 1. Classification process – classifier construction.



Outlook	HWDone	Weekend	Play
Sunny	TRUE	TRUE	Yes
Sunny	TRUE	FALSE	Yes
Rainy	FALSE	FALSE	No

Figure 2. Classification process – prediction.

detail. Extensions to RF are given in Section 5. Section 6 provides some applications of RF. Discussion then follows in Section 7. Finally, the paper concludes with a short summary and pointer to future developments in Section 8.

2. Background on ensemble classification

Ensemble classification is an application of ensemble learning to boost the accuracy of classification. Ensemble learning is a machine learning paradigm where multiple models are used to solve the same problem (Kuncheva & Whitaker, 2003; Polikar, 2006; Rokach, 2010). In ensemble classification, multiple classifiers are used and are more accurate than the individual classifiers in the ensemble. A voting scheme is then used to determine the class label for unlabeled instances. A simple and yet effective voting scheme is majority voting (Lam & Suen, 1997). In majority voting, each classifier in the ensemble is asked to predict the class label of the instance being considered. Once all the classifiers have been queried, the class that receives the greatest number of votes is returned as the final decision of the ensemble. Veto voting is an alternative voting scheme where one single classifier vetoes the decision of other classifiers (Shahzad & Lavesson, 2012; Sun & Dance, 2012). A recent voting scheme by Shahzad and Lavesson (2013) is called trust-based veto voting and is an extension of veto voting. This voting scheme considers the trust of each classifier to determine whether a classifier or set of classifiers can veto the decision.

To achieve optimal results, the classifiers in the ensemble should both be accurate and diverse. An accurate classifier is one that has an error rate better than random guessing. Two classifiers are diverse if they make different errors on new data points. The more diverse the classifiers are, the better the results are. In fact, it has been proven empirically that ensembles tend to yield better results when there is a significant diversity among the models (Kuncheva & Whitaker, 2003). This explains why many ensemble methods seek to promote diversity among the models they combine (Adeva, Beresi, & Calvo, 2005; Brown, Wyatt, Harris, & Yao, 2005).

Three widely used ensemble approaches could be identified, namely, boosting, bagging, and stacking. Boosting is an incremental process of building a sequence of classifiers, where each classifier works on the incorrectly classified instances of the previous one in the sequence. AdaBoost (Freund & Schapire, 1997) is the representative of this class of techniques. However, AdaBoost is prone to overfitting. The other class of ensemble approaches is the Bootstrap Aggregating (Bagging) (Breiman, 1996a). Bagging involves building each classifier in the ensemble using a randomly drawn sample of the data, having each classifier giving an equal vote when labeling unlabeled instances. Bagging is known to be more robust than boosting against model overfitting. RF is the main representative of bagging (Breiman, 2001). Stacking (sometimes called stacked generalization) extends the cross-validation technique that partitions the data set into a held-in data set and a held-out data set; training the models on the held-in data; and then choosing whichever of those trained models performs best on the held-out data. Instead of choosing among the models, stacking combines them, thereby typically getting performance better than any single one of the trained models (Wolpert, 1992). Stacking has been successfully used on both supervised learning tasks (regression) (Breiman, 1996b) and unsupervised learning (density estimation) (Smyth & Wolpert, 1999).

Dietterich (2000) conducted an experimental study to compare the performance of three methods for constructing ensembles of classifiers using C4.5 (Quinlan, 1993), namely, bagging (Breiman, 1996a), boosting (Freund & Schapire, 1997), and randomization. With little or no noise in the data, experiments showed that boosting has proved superior to bagging and randomization. Bagging and randomization demonstrated similar performance, however, with low noise in the data, randomization performed slightly better. Boosting performance seemed to deteriorate with noise. Bagging performance, on the other hand, seemed to improve, for it was able to utilize the noise to produce more diverse classifiers. This finding is consistent with Kuncheva and Whitaker (2003) and Adeva et al. (2005) that advocate for diversity to achieve better results. Other experiments showed that, unlike bagging, by increasing the noise rate, randomization failed to produce diverse classifiers. In the next section, we will look at developments that preceded RF.

3. Earlier developments to RFs

Ho (1995) proposed a method to overcome a fundamental limitation on the complexity of decision tree classifiers derived with traditional methods. Such classifiers cannot grow to arbitrary complexity without sacrificing the generalization accuracy on unseen data. The proposed method uses oblique decision trees which are convenient for optimizing training set accuracy. The essence of the method is to build multiple trees in randomly selected subspaces of the feature space. The trees generalize their classification in complementary ways, and their combined classification can be monotonically improved.

Amit and Geman (1997) proposed a shape recognition approach based on the joint induction of shape features and tree classifiers. Because of virtually infinite number of features, they reached the conclusion that no classifier based on the full feature set could be evaluated as it was impossible to determine a priori which features were informative. Due to the number and nature of features, standard decision tree construction based on a fixed length feature vector was not feasible. An alternative approach would be to entertain a small random of sample features at each node, constrain their complexity to increase with tree depth, and grow multiple trees. Terminal nodes contain estimates of the corresponding posterior distribution over shape classes. By sending the image down and aggregating the resulting distribution, the image can be classified.

In another paper by Ho (1998), he proposed a method to solve the dilemma between overfitting and achieving maximum accuracy. This was done by constructing a decision-tree-based classifier that maintained the highest accuracy on training data and, at the same time, improved on generalization accuracy as it grows in complexity. The classifier consisted of multiple trees constructed systematically by pseudo-randomly selecting subsets of components of the feature vector, that is, trees constructed in randomly chosen subspaces. When empirically tested against publicly available data sets, the subspace method proved its superiority when compared to single-tree classifiers and other forest construction methods. The next section introduces RF which is an ensemble method that combines existing techniques in order to construct a collection of decision trees with controlled variation.

4. Random forest

RF is an ensemble learning method used for classification and regression. Developed by Breiman (2001), the method combines Breiman's bagging sampling approach (1996a), and the random selection of features, introduced independently by Ho (1995; 1998) and Amit and Geman (1997), in order to construct a collection of decision trees with controlled variation. Using bagging, each decision tree in the ensemble is constructed using a sample with replacement from the training data. Statistically, the sample is likely to have about 64% of instances appearing at least once in the sample. Instances in the sample are referred to as in-bag instances, and the remaining instances (about 36%) are referred to as out-of-bag instances. Each tree in the ensemble acts as a base classifier to determine the class label of an unlabeled instance. This is done via majority voting where each classifier casts one vote for its predicted class label, then the class label with the most votes is used to classify the instance.

To illustrate RF and majority voting, consider the training data depicted in Table 1 which consist of eight samples and four features. An RF will be created to predict the value of the *Play* feature which will determine whether a child can play or not, given the predefined values for the other features, namely, *Outlook*, *HWDone*, and *Weekend*. For example, it is obvious from the training data that a child can play if he finished the homework on a sunny day regardless of whether it is a weekend or weekday. The child cannot play on a rainy weekday even if he finished the homework. To aid in classifying new samples, an RF of three trees was created as shown in Figure 3. Table 2 shows the result of casting votes to classify the sample (rainy, false, true,?), where a ? mark is used to indicate the value of the *Play* feature to be determined. As shown in Table 2, trees A and C voted for "yes," whereas tree B voted for "no." By majority voting, the winning vote is therefore "yes" (child can play!).

During the construction of the individual trees in the RF, randomization is also applied when selecting the best node to split on. Typically, this is equal to $\sqrt{F}$, where F is the number of features in the data set. Algorithm 1 depicts the RF algorithm where N is the number of training samples and M is the number of features in data set.

Breiman (2001) introduced additional randomness during the construction of decision trees using the classification and regression trees (CART) technique. Using this

Table 1. Training data.

Outlook	HWDone	Weekend	Play
Sunny	True	True	Yes
Sunny	True	False	Yes
Sunny	False	True	Yes
Sunny	False	False	No
Rainy	True	True	Yes
Rainy	True	False	No
Rainy	False	True	Yes
Rainy	False	False	No

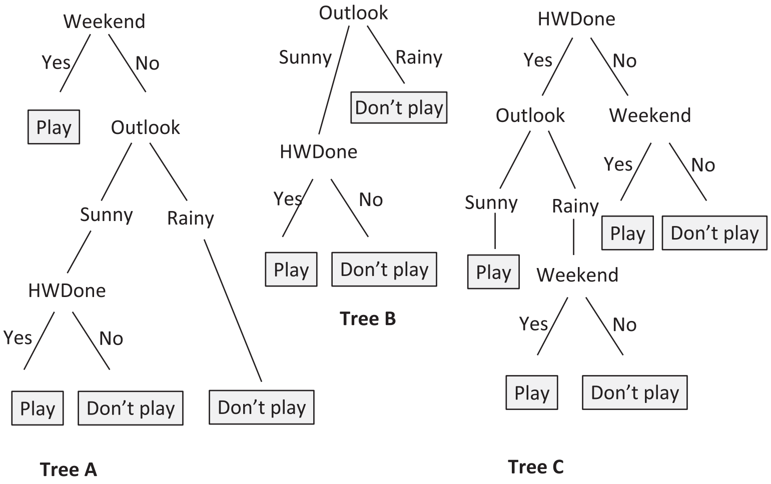


Figure 3. RF of three trees.

Table 2. Vote casting for (rainy, false, true,?).

Tree	Vote
A	Yes
B	No
C	Yes

technique, the subset of features selected in each interior node is evaluated with the Gini index heuristics. The feature with the highest Gini index is chosen as the split feature in that node. Gini index has been introduced by Breiman, Friedman, Olshen, and Stone (1984). However, it has been first introduced by the Italian statistician Corrado Gini in 1912. The index is a function that is used to measure the impurity of data, that is, how uncertain we are if an event will occur. In classification, this event would be the determination of the class label (Bader-El-Den & Gaber, 2012). In its general form, it can be calculated as

$$\mathrm{Gini}(t) = 1 - \sum_{i=1}^{N} P(C_i|t)^2, \tag{1}$$

where t is a condition, N the number of classes in the data set, and C_i is the ith class label in the data set.

In the original paper on RF (Breiman, 2001), it was shown that the RF error rate depends on *correlation* and *strength*. Increasing the correlation between any two trees in the RF increases the forest error rate. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the RF error rate. Such findings seem to be consistent with a study made by Bernard, Heutte, and Adam (2010) which showed that the error rate statistically decreases by jointly maximizing the strength and minimizing the correlation.

Algorithm 1 RF algorithm

```
{User Settings}
input $N$, $S$
{Process}
Create an empty vector $\overrightarrow{RF}$
**for** $i = 1 \rightarrow N$ **do**
    Create an empty tree $T_i$
    **repeat**
        Sample $S$ out of all features $F$ using Bootstrap sampling
        Create a vector of the $S$ features $\overrightarrow{F_S}$
        Find Best Split Feature $B(\overrightarrow{F_S})$
        Create A New Node using $B(\overrightarrow{F_S})$ in $T_i$
    **until** No More Instances To Split On
    Add $T_i$ to the $\overrightarrow{RF}$
**end for**
{Output}
A vector of trees $\overrightarrow{RF}$
```

Key advantages of RF over its AdaBoost counterpart are robustness to noise and overfitting (Boinee, De Angelis, & Foresti, 2005; Breiman, 2001; Liaw & Wiener, 2002; Robnik-Šikonja, 2004). Overfitting generally occurs when a model is constructed in such a way that it fits the data more than is warranted. A model which has been overfit will generally have poor predictive performance, as it does not generalize well. By generalization we mean how well will the model make predictions for cases that are not in the training set. Hawkins (2004) pointed out that overfitting adds complexity to a model without any gain in performance or, even worse, leads to poorer performance. A classifier that suffers from overfitting is likely to have a low error rate for the training instances (in-bag instances), and a higher error rate for the out-of-bag instances.

Other advantages as listed in the original paper about RF (Breiman, 2001):

1. Accuracy is as good as Adaboost and sometimes better.
2. It is faster than bagging or boosting.
3. It gives useful internal estimates of error, strength, correlation and variable importance.
4. It is simple and easily parallelized.

The next section explores some extensions to RF. With one common goal in mind, they were mainly developed in order to further boost its performance.

5. RF extensions

Over the past decade, some research was invested in boosting the performance of RF. One of the earliest to be reported is by Latinne, Debeir, and Decaestecker (2001). A method based on the McNemar non-parametric test of significance was proposed. The method a priori determines the minimum number of trees in the RF to use in order to obtain prediction accuracy comparable to the one obtained with larger ensembles. In addition to maintaining accuracy with fewer trees, the method significantly improves classification speed and reduces memory costs.

Robnik-Šikonja (2004) investigated new ways to improve the performance of RF. By using several attribute evaluation measures instead of just one, the correlation between trees is decreased without any loss in their strength. Another way to improve the performance of RF is to change the voting method. Instead of using majority voting, weighted voting is used. With this voting technique, internal estimates are used to identify instances most similar to the instance being labeled. The votes of the corresponding trees are then weighted with the strength they demonstrate on these near instances. Improvements were demonstrated on several classification data sets.

Tsymbal, Pechenizkiy, and Cunningham (2006) found a way to improve the performance of RF on some data sets by replacing majority voting with more sophisticated dynamic integration techniques. Three techniques were used: Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS). Using DV and DVS integration strategies, experimental studies showed that dynamic integration was able to improve the accuracy of RFs on 12 out of 27 data sets.

Amaratunga, Cabrera, and Lee (2008) investigated the significance decline in RF when the number of features is large and the number of truly informative features is small (as in the DNA microarray data set). The proposed novel and simple approach was to pick the eligible subsets of features to split each node by weighted random sampling instead of simple random sampling, with the weights tilted in favor of the informative features. The approach demonstrated superior performance when applied to several actual microarray data sets.

Saffari, Leistner, Santner, Godec, and Bischof (2009) proposed a novel online RF algorithm to remedy the limitations of the off-line algorithm which has limited usability for many practical problems. Ideas from online bagging, extremely randomized forests, and online decision tree growing procedures were combined to produce the online version of the algorithm. To boost performance, temporal weighting scheme for adaptively discarding some trees based on their out-of-bag error was added. Experiments have shown that the performance of the online algorithm proved comparable to the off-line version.

Bader-El-Den and Gaber developed an approach to enhance the accuracy of RF by using genetic algorithms (Goldberg, 1989). The approach was called genetic algorithm-based random forest (GARF) (Bader-El-Den & Gaber, 2012). Experiments have shown that GARF outperformed other state-of-the-art classification techniques including AdaBoost.

Xu, Huang, Williams, and Ye (n.d.) proposed a hybrid weighted RF algorithm for classifying very high-dimensional data. It was called hybrid because different decision tree algorithms including C4.5, CART, and CHAID were used to build the trees in the RF. The hybrid RF was tested on eight high-dimensional data sets. When compared with the traditional RF, results showed that the hybrid approach consistently performed better.

Table 3 shows a summary in chronological order of the aforementioned extensions to boost the performance of RF. The next section explores some interesting applications of RF.

6. RF applications

Over the past decade, many applications of RF were developed in virtually all disciplines, and new applications are yet to be uncovered. The ones chosen in this section are by no means exhaustive as there are many; in fact, a paper can be written about RF applications developed to date. In this section, we have selected some interesting ones.

In *Ecology*, Cutler et al. (2007) compared the accuracies of RF and four other commonly used statistical classifiers using various species data collected from multiple locations in the USA. The results demonstrated RF's superiority over the other techniques.

In *Medicine*, Klassen, Cummings, and Seldana (2008) conducted some experiments to explore several attribute selection methods with RF that precisely classified cancer using a published benchmark data set. Experimental results showed that RF performed well for microarray data in terms of speed and accuracy with several different gene sets. Hu (2009) applied RF to study the prediction of pathologic complete response in breast cancer. Results showed that the feature selection scheme of RF was able to identify important genes of biological significance.

Table 3. Summary of RF extensions.

References	Technique(s)	Summary
Latinne et al. (2001)	McNemar non-parametric test of significance	Used McNemar non-parametric test of significance to a priori limit the number of trees that will participate in majority voting and without loss in accuracy
Robnik-Šikonja (2004)	Several attributes measures, weighted voting	Decreased correlation between trees by using several attribute evaluation measures. Used weighted voting instead of majority voting
Tsymbal et al. (2006)	Dynamic integration techniques	Replaced majority voting with more sophisticated dynamic integration techniques: DS, DV, and DVS
Amaratunga et al. (2008)	Weighted random sampling	Improved the declining performance when the number of features is large and the number of truly informative features is small by using weighted random sampling instead of simple random sampling when picking features to split each node
Saffari et al. (2009)	Online RF algorithm	Introduced a novel online RF algorithm to remedy the limitations of the off-line algorithm
Bader-El-Den and Gaber (2012)	Genetic algorithms	Used genetic algorithms to boost the performance of RF
Xu et al. (n.d.)	Hybrid RF approach	Proposed a hybrid RF approach for classifying very high-dimensional data that outperformed the traditional RF

Table 4. Summary of RF applications.

References	Domain	Aim	Summary
Cutler et al. (2007)	Ecology	Species classification	Compared RF accuracy with other techniques to classify various species. RF proved to be superior to others
Klassen et al. (2008)	Medicine	Cancer classification	Did experiments to prove that RF was able to precisely classify cancer. Also found that RF performed well for microarray data in terms of speed and accuracy with several different gene sets
Hu (2009)	Medicine	Prediction of pathologic complete response in breast cancer	Applied RF to study the prediction of pathologic complete response in breast cancer. Results showed that the feature selection scheme of RF was able to identify important genes of biological significance
Gao et al. (2009)	Astronomy	Astronomical object classification	Conducted some experiments on multi-wavelength data classification. Results showed that RF proved effective for astronomical object classification
Flaxman et al. (2011)	Autopsy	Cause of death prediction	Introduced a new CCVA method using RF to predict cause of death
Zaklouta et al. (2011)	Traffic and transport planning	Traffic signs classification	Used K-d trees and RFs to classify 43 types of traffic signs using different size HOG descriptors and distance transforms
Löw et al. (2012)	Agriculture	Accurate classification of crops	Used a combination of RF and SVM classifiers to improve crop classification accuracy and to provide spatial information on map uncertainty
Boulesteix et al. (2012)	Bioinformatics	Recent developments survey in bioinformatics	Amalgamated 10 years of RF development including representative examples of RF applications in this domain

In *Astronomy*, Gao, Zhang, and Zhao (2009) conducted some experiments on multi-wavelength data classification. Results showed that RF proved effective for astronomical object classification. RF has proved to be superior due to its own virtues in classification, feature selection, feature weighting, and detection of outliers.

In *Autopsy*, Flaxman, Vahdatpour, Green, James, & Murray (2011) introduced a new computer-coded verbal autopsy (CCVA) method using RF to predict cause of death. This was done by training RF to distinguish between each pair of causes, and then combining the results through a novel ranking technique. The new method outperformed physician-certified verbal autopsy and was recommended for analyzing past and current verbal autopsies.

In *Traffic and Transport Planning*, Zaklouta, Stanciulescu, and Hamdoun (2011) used K-d trees and RFs to classify 43 types of traffic signs using different size histogram of oriented gradients (HOG) descriptors and distance transforms. Results showed that RFs outperformed K-d trees by achieving a classification rate of

97.2% and 81.8% on HOG and distance transforms, respectively.

In *Agriculture*, Löw, Schorcht, Michel, Dech, and Conrad (2012) used a combination of RF and support vector machine (SVM) classifiers to improve crop classification accuracy and to provide spatial information on map uncertainty. Results showed that the feature selection merit of RF improved the performance of SVM. Using this hybrid classifier improved classification accuracy compared with single classifiers and user's and producer's accuracy.

In *Bioinformatics* and *Computational Biology*, Boulesteix, Janitza, Kruppa, and König (2012) amalgamated 10 years of RF development. Practical aspects of RF including selection of parameters, available RF implementations, important pitfalls, and biases of RF and its variable importance measures were covered. The paper also surveyed recent developments relevant to Bioinformatics as well as some representative examples of RF applications in this domain. Table 4 depicts a summary of the RF applications discussed in this section.

7. Discussion

Looking back at the work done to boost the performance of RF in Section 5, we can see that several researchers focused on improving two of RF key merits, namely, feature selection and voting. For feature selection, Robnik-Šikonja (2004) and Amaratunga et al. (2008) provided alternative methods for selecting the best features to split at each node. As for voting, Robnik-Šikonja (2004) and Tsymbal et al. (2006) proposed new voting techniques that proved to be more efficient than the traditional majority voting technique.

As mentioned before, increasing the strength of the individual trees in the forest and decreasing the correlation between trees are key factors in improving the performance by reducing the RF error rate. This approach was used by Bernard et al. (2010).

Another interesting way to improve the performance of RF is to reduce the number of trees that will cast a vote in majority voting. Latinne et al. (2001) used the McNemar non-parametric test of significance but another way to do it would be to select as many trees with low correlation (which tend to decrease the RF error rate), and as few trees with high correlation (which tend to increase the RF error rate). Utilizing the internal estimates of RF is yet another technique to improve the performance of RF. As stated in Breiman (2001), internal estimates monitor error, strength, correlation, and variable importance. We believe that such estimates can be exploited for the purpose of improving the performance.

In Section 2, we have stressed the importance of diversity to achieve better performance. Perhaps, this is a worthwhile area to investigate further by developing new techniques to increase the diversity of trees in the RF. As discussed in Section 5, the hybrid technique used in Xu et al. (n.d.) was a good example of increasing the diversity among the trees. This added diversity can have a positive impact on the accuracy of each tree in the forest, yielding better classification performance of the ensemble.

Creating a hybrid ensemble approach involving RF and some of its rivals would be another research topic to investigate. In such an approach, a combination of RF, Adaboost (Freund & Schapire, 1997), and Random Split Selection (Dietterich, 2000) can be employed yielding a more diverse ensemble. Performance of such a hybrid ensemble can then be compared with RF to see if a performance gain was achieved.

Further research can be conducted to show how RF can be used in applications that have not been addressed by the research community. These include existing or new and emerging applications.

8. Summary

The success demonstrated by RF ever since its inception by Breiman in the Fall of 1999 made it attractive to the research community. Indeed, researchers have competed to enhance RF or to show its significance in the application area. In this paper, we have surveyed the relevant research which focused on two main areas: performance enhancement and applications. As a recap, some extensions to boost the performance were listed in Table 3. Table 4, on the other hand, listed some applications of RF in a number of disciplines. Specific recommendations for additional potential research directions in these areas were suggested in the previous section.

References

Adeva, J. J. G., Beresi, U., & Calvo, R. (2005). Accuracy and diversity in ensembles of text categorisers. *CLEI Electronic Journal*, *9*(1), 1–12.

Amaratunga, D., Cabrera, J., & Lee, Y.-S. (2008). Enriched random forests. *Bioinformatics*, *24*(18), 2010–2014.

Amit Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, *9*(7), 1545–1588.

Bader-El-Den, M., & Gaber, M. (2012, November 12–15). Garf: Towards self-optimised random forests. In T. Huang, Z. Zeng, C. Li, & C.-S. Leung (Eds.), *Neural Information Processing – 19th International Conference, ICONIP 2012, Doha, Qatar, Proceedings, Part II*, Lecture Notes in Computer Science (pp. 506–515). Berlin: Springer.

Bernard, S., Heutte, L., & Adam, S. (2010, August 18–21). A study of strength and correlation in random forests. In D.-S. Huang, T. Martin McGinnity, L. Heutte, & X.-P. Zhang (Eds.), *Advanced Intelligent Computing Theories and Applications – 6th International Conference on Intelligent Computing, ICIC 2010, Changsha, China, Proceedings*, Communications in Computer and Information Science (pp. 186–191). Berlin: Springer.

Boinee, P., De Angelis, A., & Foresti, G. L. (2005). Meta random forests. *International Journal of Computational Intelligence*, *2*(3), 138–147.

Boulesteix, A.-L., Janitza, S., Kruppa, J., & König, I. R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(6), 493–507.

Breiman, L. (1996a). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (1996b). Stacked regressions. *Machine Learning*, *24*(1), 49–64.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees* (1st ed.). New York/Boca Raton, FL: Chapman and Hall/CRC.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: A survey and categorisation. *Information Fusion*, *6*(1), 5–20.

Cutler, D. R., Edwards Jr., T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, *88*(11), 2783–2792.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, *40*(2), 139–157.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, *17*(3), 37.

Flaxman, A. D., Vahdatpour, A., Green, S., James, S. L., & Murray, C. J. L. (2011). Random forests for verbal autopsy analysis: Multisite validation study using clinical diagnostic gold standards. *Population Health Metrics*, *9*(29), 1–11.

Freund Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Gao, D., Zhang, Y.-X., & Zhao, Y.-H. (2009). Random forest algorithm for classification of multiwavelength data. *Research in Astronomy and Astrophysics*, *9*(2), 14–39.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning* (1st ed.). Boston, MA: Addison-Wesley Longman Publishing.

Hawkins, D. M. (2004). The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, *44*(1), 1–12.

Ho, T. K. (1995). Random decision forests. In *Document analysis and recognition, 1995, Proceedings of the third international conference, Montreal, Quebec, Canada* (Vol. 1, pp. 278–282). New York City, NY: IEEE.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *Intelligence, IEEE Transactions on Pattern Analysis and Machine*, *20*(8), 832–844.

Hu, W. (2009). Identifying predictive markers of chemosensitivity of breast cancer with random forests. *Cancer*, *13*, 59–64.

Klassen, M., Cummings, M., & Saldana, G. (2008, April 9–11). Investigation of random forest performance with cancer microarray data. In T. Philip (Ed.), *Proceedings of the ISCA 23rd International Conference on Computers and Their Applications, CATA 2008, Cancun, Mexico* (pp. 64–69). Cary, NC: International Society for Computers and Their Applications.

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, *51*(2), 181–207.

Lam, L., & Suen, C. Y. (1997). Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *27*(5), 553–568.

Latinne, P., Debeir, O., & Decaestecker, C. (2001, July 2–4). Limiting the number of trees in random forests. In J. Kittler & F. Roli (Eds.), *Multiple Classifier Systems, Second International Workshop, MCS 2001 Cambridge, UK*, Lecture Notes in Computer Science (pp. 178–187). Berlin: Springer.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2*(3), 18–22.

Löw, F., Schorcht, G., Michel, U., Dech, S., & Conrad, C. (2012, September 24). Per-field crop classification in irrigated agricultural regions in Middle Asia using random forest and support vector machine ensemble. In *SPIE remote sensing, Edinburgh, United Kingdom* (pp. 85380R–85380R). Bellingham, WA: International Society for Optics and Photonics.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*(3), 21–45.

Quinlan, J. R. (1993). *C4. 5: Programs for machine learning* (Vol. 1). Boston, MA: Morgan Kaufmann.

Robnik-Šikonja, M. (2004, September 20–24). Improving random forests. In J.-F. Boulicaut, F. Esposito, F. Giannotti, & D. Pedreschi (Eds.), *Machine Learning: ECML 2004, 15th European Conference on Machine Learning, Pisa, Italy, 2004 Proceedings*, Lecture Notes in Computer Science (pp. 359–370). Berlin: Springer.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*(1–2), 1–39.

Saffari, A., Leistner, C., Santner, J., Godec, M., & Bischof, H. (2009). On-line random forests. In *2009 IEEE 12th international conference on computer vision workshops (ICCV workshops), Kyoto, Japan* (pp. 1393–1400). New York City, NY: IEEE.

Shahzad, R. K., & Lavesson, N. (2012). Veto-based malware detection. In *2012 seventh international conference on availability, reliability and security (ARES), Prague, Czech Republic* (pp. 47–54). New York City, NY: IEEE.

Shahzad, R. K., & Lavesson, N. (2013). Comparative analysis of voting schemes for ensemble-based malware detection. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *4*(1), 98–117.

Smyth, P., & Wolpert, D. (1999). Linearly combining density estimators via stacking. *Machine Learning*, *36*(1–2), 59–83.

Sun,Y.-A., & Dance, C. (2012). When majority voting fails: Comparing quality assurance methods for noisy human computation environment. *CoRR*. Retrieved from arXiv:1204.3516.

Tsymbal, A., Pechenizkiy, M., & Cunningham, P. (2006, September 18–22). Dynamic integration with random forests. In J. Fürnkranz, T. Scheffer, & M. Spiliopoulou (Eds.), *Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, 2006 Proceedings*, Lecture Notes in Computer Science (pp. 801–808). Berlin: Springer.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259.

Xu, B., Huang, J. Z., Williams, G., & Ye, Y. (2012). Hybrid weighted random forests for classifying very high-dimensional data. *International Journal of Data Warehousing and Mining*, *8*(2), 44–63.

Yan, W., & Goebel, K. F. (2004). Designing classifier ensembles with constrained performance requirements. In B. V. Dasarathy (Ed.), *Proceedings of the SPIE 5434, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2004, Orlando, FL* (pp. 59–68). Bellingham, WA: SPIE.

Zaklouta, F., Stanciulescu, B., & Hamdoun, O. (2011). Traffic sign classification using kd trees and random forests. In *The 2011 international joint conference on neural networks (IJCNN), San Jose, CA, USA* (pp. 2151–2155). New York City, NY: IEEE.