# Random Forest Classifiers :A Survey and Future Research Directions

Vrushali Y Kulkarni
PhD Student, COEP, Pune, India
Email: kulkarnivy@rediffmail.com

Dr Pradeep K Sinha
Senior Director,
HPC, CDAC, Pune, India

## ABSTRACT

Random Forest is an ensemble supervised machine learning technique. Machine learning techniques have applications in the area of Data mining. Random Forest has tremendous potential of becoming a popular technique for future classifiers because its performance has been found to be comparable with ensemble techniques bagging and boosting. Hence, an in-depth study of existing work related to Random Forest will help to accelerate research in the field of Machine Learning. This paper presents a systematic survey of work done in Random Forest area. In this process, we derived Taxonomy of Random Forest Classifier which is presented in this paper. We also prepared a Comparison chart of existing Random Forest classifiers on the basis of relevant parameters. The survey results show that there is scope for improvement in accuracy by using different split measures and combining functions; and in performance by dynamically pruning a forest and estimating optimal subset of the forest. There is also scope for evolving other novel ideas for stream data and imbalanced data classification, and for semi-supervised learning. Based on this survey, we finally presented a few future research directions related to Random Forest classifier.

**Keywords** - Data Mining, Ensemble, Classification, Random Forest, Supervised Machine Learning

## 1. INTRODUCTION

Random Forest is an Ensemble Supervised Machine Learning technique that has emerged recently. Machine learning techniques have applications in the area of Data mining. Data mining is broadly classified as Descriptive and Predictive. Descriptive data mining concentrates more on describing the data, grouping them into categories, and summarizing the data. Predictive data mining analyzes past data and generates trends or conclusions for future prediction. Predictive data mining has its roots in the classical model building process of statistics. Predictive model building works on the basis of feature analysis of predictor variables. One or more features are considered as predictors. Output is some function of the predictors, which is called hypothesis. The generated hypotheses are tested for their acceptance or rejection. Accuracy of this model is decided by following various error estimation techniques. Usually, descriptive data mining is implemented using unsupervised machine learning techniques, while predictive data mining is carried out using supervised machine learning techniques. Supervised machine learning uses labeled data samples; labels are used to classify samples into different categories. Predictive model learns using training dataset. Test dataset is used to estimate accuracy of the model. Decision tree is commonly used technique for supervised machine learning. Random Forest [11] uses decision tree as base classifier. Random Forest generates multiple decision trees; the randomization is present in two ways: (1) random sampling of data for bootstrap samples as it is done in bagging and (2) random selection of input features for generating individual base decision trees. Strength of individual decision tree classifier and correlation among base trees are key issues which decide generalization error of a Random Forest classifier [11]. Accuracy of Random Forest classifier has been found to be at par with existing ensemble techniques like bagging and boosting. As per Breiman [11], Random Forest runs efficiently on large databases, can handle thousands of input variables without variable deletion, gives estimates of important variables, generates an internal unbiased estimate of generalization error as forest growing progresses, has effective method for estimating missing data and maintains accuracy when a large proportion of data are missing, and has methods for balancing class error in class population unbalanced data sets. The inherent parallel nature of Random Forest has led to its parallel implementations using multithreading, multi-core, and parallel architectures. Random Forest is used in many recent classification and prediction applications due to its above mentioned features. In this paper, we have concentrated on the empirical research related to Random forest classifier rather than exploring and analyzing its theoretical background in detail.

This paper is organized as follows: Section 2 provides theoretical foundations of ensembles and Random Forest algorithm. Section 3 provides a survey of current status of research on Random Forest classifier. Based on this survey, we have evolved Taxonomy of Random Forest classifier which is also presented in this section. Section 4 includes Discussions and a Summary chart summarizing key features of the surveyed Random Forest classifiers in tabular form. Section 5 focuses few future research directions in the area of Random Forest. Section 6 gives concluding remarks.

## 2. THEORETICAL FOUNDATIONS

### 2.1 Ensemble Classifiers

An ensemble consists of a set of individually trained classifiers (such as neural networks or decision trees) whose predictions are combined for classifying new instances. Previous research has shown that an ensemble is often more accurate than any of the single classifiers in the ensemble [20], [22], [29]. Bagging [10] and Boosting [32] are two popular methods for producing ensembles. These methods use re-sampling techniques to obtain different training sets for each of the classifiers. Bagging stands for bootstrap aggregating which works on the concept of bootstrap samples. If original training dataset is of size N and m individual classifiers are to be generated as part of ensemble then m different training sets- each of size N, are generated from original dataset by sampling with replacement. The multiple classifiers generated in bagging are independent to each other. In case of boosting, weights are assigned to each sample from the training dataset. If m classifiers are to be generated, they are generated sequentially such that one classifier is generated in a single iteration. For generating classifier $C_i$, weights of training samples are updated based on classification results of classifier $C_{i-1}$. The classifiers generated by boosting are dependent on each other.

The theoretical and empirical research related to ensemble has shown that an ideal ensemble consists of highly correct classifiers that disagree as much as possible [18], [22], [26], [35]. Opitz and Shavlik [28] empirically verified that such ensembles generalize well. Breiman [10] showed that bagging is effective on unstable learning algorithms. In [23] Kuncheva presents four approaches for building ensembles of diverse classifiers:

1. Combination level: Design different combiners.

2. Classifier level: Use different base classifiers.

3. Feature level: Use different feature subsets.

4. Data level: Use different data subsets.

## 2.2 Random Forest

***Definition***: Random Forest is a classifier consisting of a collection of tree-structured classifiers $\{h(x, \Theta_k)\ \ k=1, 2, ....\}$, where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input *x* [11].

Random Forest generates an ensemble of decision trees. To achieve diversity among base decision trees, Breiman selected the randomization approach which works well with bagging or random subspace methods [10], [11], [29]. To generate each single tree in Random Forest Breiman followed following steps: If the number of records in the training set is N, then N records are sampled at random but with replacement, from the original data, this is bootstrap sample. This sample will be the training set for growing the tree. If there are M input variables, a number m << M is selected such that at each node, m variables are selected at random out of M and the best split on these m attributes is used to split the node. The value of m is held constant during forest growing. Each tree is grown to the largest extent possible. There is no pruning.

In this way, multiple trees are induced in the forest; the number of trees is pre-decided by the parameter Ntree. The number of variables (m) selected at each node is also referred to as mtry or k in the literature. The depth of the tree can be controlled by a parameter nodesize (i.e. number of instances in the leaf node) which is usually set to one.

Once the forest is trained or built as explained above, to classify a new instance, it is run across all the trees grown in the forest. Each tree gives classification for the new instance which is recorded as a vote. The votes from all trees are combined and the class for which maximum votes are counted (majority voting) is declared as classification of the new instance.

This process is referred to as Forest RI in the literature [11]. Here onwards, Random Forest means the forest of decision trees generated using Forest RI process.

In the forest building process, when bootstrap sample set is drawn by sampling with replacement for each tree, about 1/3rd of original instances are left out. This set of instances is called OOB (Out-of-bag) data. Each tree has its own OOB data set which is used for error estimation of individual tree in the forest, called as OOB error estimation. Random Forest algorithm also has in-built facility to compute variable importance and proximities [11]. The proximities are used in replacing missing values and outliers.

Illustrating Accuracy of Random Forest:

The Generalization error (PE*) of Random Forest is given as,

$$PE^* = P_{x,y}(mg(X,Y)) < 0$$

Where mg(X,Y)is Margin function. The Margin function measures the extent to which the average number of votes at (X, Y) for the right class exceeds the average vote for any other class. Here X is the predictor vector and Y is the classification.

The Margin function is given as,

$$mg\,(X,Y) = av_k\, I(h_k\,(X) = Y) - max_{\,j \neq Y}\, av_k\, I(h_k\,(X) = j)$$

Here I(.) is Indicator function.

Margin is directly proportional to confidence in the classification.

Strength of Random Forest is given in terms of the expected value of Margin function as,

$$S = E_{X,\,Y}\,(mg\,(X,\,Y))$$

The generalization error of ensemble classifier is bounded above by a function of mean correlation between base classifiers and their average strength (s) [33]. If ρ is mean value of correlation, an upper bound for generalization error is given by,

$$PE^* \leq \rho\ (1 - s^2)\,/\,s^2$$

## 3. CURRENT ONGOING WORK ON RANDOM FOREST

Research work in the area of Random Forest aims at either improving accuracy, or improving performance (reducing time required for learning and classification), or both. Some work aims at experimentation with Random Forest using online continuous stream data, which is essential today due to data streams getting generated by various applications. Random Forest being an ensemble technique, experiments are done with its base classifier, e.g. Fuzzy Decision Tree as base classifier of Random Forest. We have done systematic survey of current ongoing research on Random Forest and developed a "Taxonomy of Random Forest Classifier". In this section, we first elaborate in detail the work done and then present the Taxonomy.

### 3.1 Improvements in Random Forest Algorithm Based on Accuracy

To have a good ensemble, base classifiers are to be diverse (i.e. they predict differently), and accurate. Random selection of attributes makes individual trees weak. The improvements suggested are such that individual base classifiers are strong as well as diverse.

Meta Random Forest [7] is based on the concept of using random forest themselves as base classifiers for making ensembles, and the performance of this model is tested and compared with the existing Random Forest algorithm. Meta Random Forests are generated by both bagging and boosting approaches i.e. ensemble using Random Forest as base classifier with bagging approach, and ensemble using Random Forest as base classifier with boosting approach. Comparative study of both these techniques and original Random Forest technique has shown that Bagged Random Forest gives the best results among the three techniques.

In original Random Forest, Gini Index is used in decision tree for attribute split. Gini Index is not able to detect strong conditional dependencies among attributes [34]. ReliefF measure for attribute split gives better results in this case. Robnik and Sikonja [34] experimented with Random Forest using five different attribute measures; each fifth of the trees in the forest is generated using different split measure (Gini index, Gain ratio, MDL, ReliefF). This helped in decreasing correlation between the trees while retaining their strengths. The performance increase observed was not much significant.

As suggested by Breiman, Random Forest uses majority voting as voting mechanism for classification. Experiments are carried out related to the voting mechanism. For improving voting scheme, internal estimates are used. The process is as follows: for classifying a new instance, instances similar to this new instance are found. Then individual trees are given weights based on the strength they demonstrate on these selected instances. This is a kind of weighted voting. Research work related to Dynamic Integration demonstrates that performance of Random Forest is improved in some domains by replacing majority voting with Dynamic Integration, which is based on local prediction performances of base decision trees. Tsymbal, Pechenizkiy, and Cunningham [38] suggested three different techniques based on performance of local predictors: Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS).

Simon Bernard, Laurent Heutte, and Sebastien Adam [4] proposed a new Random Forest algorithm called Forest RK in which k, the number of features, is randomly selected at each node during tree induction process. In this paper it is stated that k is not a hyper-parameter, as it is not playing a crucial role in generating accurate Random Forest classifier. They used McNemar statistical test of significance to compare predictions generated by original Random Forest and Forest RK. They claimed that the two algorithms are statistically equivalent.

### 3.2 Improvements in Random Forest Algorithm based on Performance

Theoretical and empirical results have proved that above a certain number of trees, adding more trees in the forest does not improve accuracy [5]. There are specific methods suggested to find a sub-forest that can achieve prediction accuracy of a large random forest. Researchers have taken efforts in achieving smaller forests or shrink the forest. Most of these efforts are based on Overproduce-and-Chose strategy [36]. The approach taken to shrink the forest is as follows: First overproduce the forest to a fixed number decided a priori. Then calculate prediction accuracy of the forest. For every tree T in the forest, calculate the prediction accuracy of the forest that excludes T. Then find the difference (ΔT) between the prediction accuracies of the original forest and the forest without T. The tree with minimum ΔT is the least important one and it is removed from the forest [43]. The other approach is based on similarity between two trees. It works on the basis that a tree can be removed if it is similar to other trees in the forest. Another approach to limit the number of trees in the random forest works a priori and it is based on applying McNemar non-parametric test of significance between predictions of two subsets of the original forest [24].

The work which is add-on to the existing "Overproduce and Choose" paradigm is suggested in [41]. Here a new algorithm called BAGA is proposed which generates ensemble using combination of bagging and genetic algorithm techniques so that individual classifiers are determined at execution time. As bagging is to be treated as special case of Random forest, the BAGA approach suggested is also applicable to Random Forest, and hence included here as a development related to Random Forest.

Researchers have proposed a new concept called dynamic ensemble. The dynamic induction of Random Forest eliminates the Overproduce phase. In their work, Tripoliti, Fotiadis, and Manis [44] determine the number of decision trees in random forest dynamically during the growing process of forest. The method is based on on-line curve fitting. The forest is first built with 10 trees. At each next step, a new tree is added and tested if it is a best fit. For selection of best fit, eight polynomials are used. The termination of iterative process is based on predefined threshold for the fitted value and the accuracy curve. These threshold values are determined heuristically.

In Dynamic Random Forests [6], individual base trees are added in the dependent manner rather than the independent approach taken by Breiman. A new tree is added in the forest by taking into account the evaluation of the sub-forest already built and thus taking an adaptive approach. With this approach, an initial tree is generated in traditional way as it is done in original Random Forest. For generating every next tree, the weights of training instances are modified (as it is done in boosting), so that weights are increased for the instances those are wrongly classified by the initial tree and decreased for correctly classified instances. This approach generates dependent trees and nullifies the original inherent parallel nature of Random Forest. Here base trees are Random Trees rather than the Decision trees used by Breiman.

Many tasks in the data mining domain concern high-dimensional data. Consequently, these tasks are often complex and computationally expensive. A GPU-based implementation of Random Forest algorithm is developed, which is based on Compute Unified Device Architecture (CUDA). The algorithm is experimentally evaluated on NVIDIA GT 220 graphics card with 48 CUDA cores and 1 GB of memory. Both training phase and classification phase are parallelized in CUDA implementation. Performance is compared with two state-of-the-art implementations of Random Forest; sequential (LibRF) and parallel (FastRF) in Weka [19]. CudaRF outperforms both LibRF and FastRF for the specified classification task [17].

## 3.3 Improvements in Random Forest Algorithm for Online Data

Standard Random Forest algorithm works on off-line data. Many recent applications deal with data streams. Streams are conceptually end-less sequence of data records, real-time, and often arriving at high flow rates [1], [14]. The challenge with streaming data is that there cannot be multiple passes through the data for analysis. Streaming Random Forest is a classification algorithm that combines techniques used to build streaming decision trees with attribute selection techniques of Random Forest. The streaming version of Random Forest achieves classification accuracy comparable to the standard version on artificial and real data sets using only single pass through the data [1]. The limitation is that the algorithm handles only numerical or nominal attributes for which minimum and maximum values of each attribute are known. It also handles multi-class classification problem.

Online Random forest algorithm [37] generates on-line decision trees based on concepts from on-line bagging [30] and extremely randomized trees [15]. It also uses Temporal Weighting scheme to discard non performing trees based on their out-of-bag error performance. The algorithm is ported on NVIDIA GPU, which has shown ten times speed up.

Incremental Extremely Random forest algorithm is specially designed for small data streams [40]. The algorithm works on the basis of expanding the leaf nodes without reconstructing the whole trees. This approach avoids use of Hoeffding bounds which need large number of samples.

## 3.4 Data Specific Random Forest Algorithm

In many real world applications, the data to be dealt with is imbalanced. A classifier built using all data has a tendency to ignore minority class. There are two common approaches to deal with imbalanced data. The first is based on cost sensitive learning and the second is based on use of a sampling technique: either down sampling the majority class, or over sampling the minority class. Breiman has mentioned that Random Forest has methods for balancing error in class population unbalanced data [11]. Vladimir, McLachlan, and Shu Kay Ng proposed a large number of relatively small and balanced subsets where representatives from the larger pattern are to be selected randomly [27]. Another approach is ensemble learning based on repeated random sub-sampling [12]. This technique divides training data into multiple sub-samples while ensuring that each sub-sample is fully balanced. The results have shown that Random Forest ensemble outperformed SVM, bagging and boosting in terms of the area under receiver operating characteristics (ROC) curve (AUC) for Imbalanced data. It is suggested that Random Forest can be used as a base learner of ensemble for achieving better results with Imbalanced data [27].

One of the features of Random Forest is that it can handle thousands of input variables without variable deletion. The study of Gene data uses tens of thousands of gene expressions to predict an outcome using several tens or hundreds of subject. This is commonly referred to as “Large p (number of predictors) and Small n (number of samples)” problem. The “Large p Small n” paradigm arises in Microarray studies where expression levels of thousands of genes are monitored for a small number of subjects [20]. Random Forest works well for this paradigm.

The original Random Forest algorithm or its modified version (to suit the application) is used to solve classification problems in various areas. Some areas where Random Forest classifier is used are Handwritten digit recognition [2], Detection of hidden web search interfaces [42], Land cover classification [31], Prediction of fault-prone modules in software development process for effective detection and identification of defects [16], Multi-label classification [21], Analysis of Hyper-spectral data [13], etc. The survey of various application areas using Random Forest is given in summarized form in [39].

*This article is authorized for use only by Recent Science Journal Authors, Subscribers and Partnering Institutions*

### 3.5 Naïve Implementations of Random Forest Algorithm

Research work has been carried out for generating multi-class classifier using fuzzy decision trees i.e. Fuzzy Random Forest. Fuzzy Random Forests try to use the robustness of a tree ensemble, the power of the randomness to increase diversity of the trees in the forest, and the flexibility of fuzzy logic and fuzzy sets for data managing [8], [9].

Random Forests suffer from the same disadvantage as other popular discriminative learning methods: they need a huge amount of labeled data to achieve good performance. C Leistner, A Saffari, J Santner, M Godec, H Bischof [25] address this particular weakness by proposing a semi-supervised learning (SSL) approach for Random Forest allowing the algorithm to make use of both labeled and unlabeled training data. A problem with SSL methods is that they only focus on binary classification problems. Multi-class problems are often decomposed into a set of binary tasks with 1-Vs-all or 1-Vs-1 strategies. Considering the fact that most state-of-the-art SSL methods have high computational complexity, such a strategy can become a problem when dealing with a large number of samples and classes. Therefore, the ability of Random Forest algorithm to handle multi-class tasks makes it very attractive for SSL problem.

### 3.6 Taxonomy of Random Forest

Based on the above survey, we have developed Taxonomy of Random Forest Classifier which is presented in figure 1.

## 4. DISCUSSION

Ensemble methods aim at improving classification accuracy by aggregating predictions from multiple classifiers. More diverse the base classifiers and less are they correlated; the more is accuracy of the ensemble. Random Forest algorithm uses 1) Sub-sampling the examples/cases as in bagging, 2) Sub-sampling the features known as feature selection. Both these strategies are used in Random Forest to introduce randomization and achieve diversity. Also, there is no pruning in the base decision trees to ensure diversity among them.

Using the strong law of large numbers, Breiman has demonstrated that Random Forest always converges so that over-fitting is not a problem [11]. Survey of various papers shows that there is scope for work using important features of Random Forest, i.e. proximity based computation, and variable importance [39].

In case of accuracy improvement, research is done using different attribute split measures and combine functions. The survey has shown that experiments with attribute split measures has not shown significant improvement and further work in this direction need to be carried out. The weighted voting with Random Forest has shown significant improvements in accuracy. As compared to improvement in accuracy, there is less work done for improvement in performance. Performance improvement mainly concerns on reducing number of base decision trees in Random Forest so that learning and in turn, classification is faster. The survey shows that efforts are taken in suggesting different ways for finding subsets of Random forest, but no concrete work is done to find optimal subset of Random forest. Additionally, all efforts taken to find subsets of Random forest which will work with same accuracy as the original Random Forest are taking static approach. i.e. The entire forest is grown first and then step-by-step base decision trees are verified for being part of the subset. Major work of this kind uses “Overproduce and Choose” approach which is not cost effective. Though efforts are taken to generate dynamic ensemble, many of them are not eliminating the overproduce phase, i.e. generation of N classifiers at the start. Eliminating overproduce phase will truly generate dynamic ensemble. The Dynamic Random forest eliminates the overproduce phase and generates only the trees which are contributing to the better accuracy; but due to dependent way of tree generation it eliminates the inherent parallel nature of Random Forest induction. By reviewing all work done related to performance improvement of Random forest there is still an important issue which is unresolved is: what is the optimal number of base classifiers in Random Forest and how to select the optimal subset without growing the entire forest.

There are existing parallel implementations of Random Forest: PARF is parallel implementation of Random Forest using Fortran 90. FastRF is parallel implementation of Random Forest in Weka which uses multithreading. There exists GPU based parallel implementation of Random Forest using CUDA platform, based on multi-core architecture. R contains parallel cluster based Random Forest. Each implementation is specific to some language or platform.

There is lot of scope for experimentation with Random Forest using streaming data. Many recent applications like Internet traffic monitoring, Telecommunications billings, etc. produce huge amount of data and it is practically impossible to store this real-time stream. Also it is not possible to have multiple passes through this data. Many algorithms for stream data has problem with handling multi-class classification, which is not an issue in Random forest due to its inherent multi-class capability. A good amount of base research work is done related to classification of stream data using Random Forest, which can be used as fundamental work for further enhancement in this field.

Research is also going on for classifying Imbalanced data using Random forest. Results have shown that Random Forest outperforms other classification techniques for Imbalanced data and hence there is great scope for developing improved Random Forest algorithm for Imbalanced data.

Use of Fuzzy decision trees and Semi supervised learning with Random Forest is recent development. There is future scope for semi supervised learning with Random Forest due to capability of handling both labeled and unlabelled data; especially for scenarios where getting labeled data is a problem.

Most of the work done related to Random Forest follows parameter settings as mentioned by Breiman. The forest size is decided a priori and the default value used for number of trees is 100 in many cases. Weka and R are the commonly used tools for research using Random Forest. The applications implemented using Random Forest algorithms are compared with bagging and boosting. Almost all results have shown that Random Forest does either better or at least equivalent with these two techniques. Commonly used datasets for research work related to Random Forest Classifier are from UCI Machine Learning Repository. A few datasets are from Semi-supervised Benchmarks and LibSVM repository. Two synthetic datasets (Twonorm and Ringnorm) are used which are designed by Breiman.

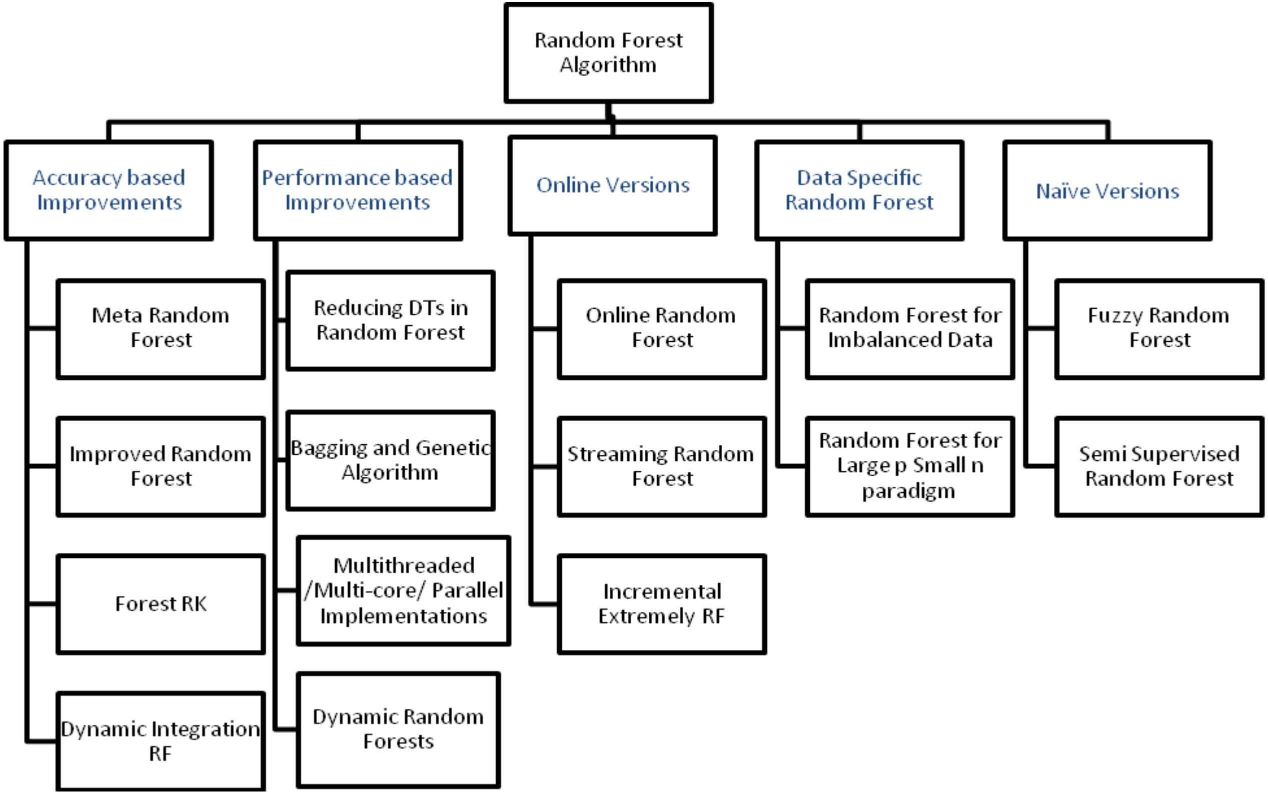
Random Forest Algorithm
Accuracy based Improvements
Performance based Improvements
Online Versions
Data Specific Random Forest
Naïve Versions
Meta Random Forest
Improved Random Forest
Forest RK
Dynamic Integration RF
Reducing DTs in Random Forest
Bagging and Genetic Algorithm
Multithreaded /Multi-core/ Parallel Implementations
Dynamic Random Forests
Online Random Forest
Streaming Random Forest
Incremental Extremely RF
Random Forest for Imbalanced Data
Random Forest for Large p Small n paradigm
Fuzzy Random Forest
Semi Supervised Random Forest

Fig.1 Taxonomy of Random Forest Classifier

Classification accuracy which is defined as percentage of correctly classified samples to the total number of samples is an important measure for evaluation of classifier. AUC (Area Under Receiver Operating Curve ROC) is used as measure of performance with Random Forest. F-measure is also used to evaluate performance of a classification [41]. Random Forest being ensemble classifier, different techniques are used to compare performances of individual base classifiers. Statistical tests, especially Wilcoxon Signed –rank test [34] and McNemar non-parametric test [3], [24] are commonly used with Random Forest.

We have systematically analyzed all the research efforts taken related to Random Forest and come up with the Comparison chart as given in figure 2. The parameters used for comparison are as follows:

1. Base Classifier: It describes the base classifier used in the Random Forest ensemble.
2. Split Measure: If base classifier of Random Forest is decision tree, then which split measure is used at each node of the tree to perform the splitting.
3. Number of Passes: For building Random Forest classifier, if single pass is sufficient or multiple passes through data are needed.
4. Combine Strategy: In Random Forest ensemble, all the base classifiers generated are used for classification. At the time of classification, how the results of individual base classifiers are combined is decided by the combine strategy.
5. Number of attributes used for base classifier generation ($M_{try}$): This parameter gives the number of how many attributes are to be used (which are randomly selected from the original set of attributes) at each node of the base decision tree.
6. Stopping Criterion: In Random Forest, multiple base classifiers are generated. The number of base classifiers is usually pre-decided or based on some estimate (usually accuracy). This is described by the stopping criterion.
7. Pruning of Forest: This parameter gives if there are steps / measures taken to perform pruning of Random Forest / to reduce the number of base classifiers from the Random forest.

8. Parallel Extension: It describes whether there is parallel extension exists for the associated approach related to Random Forest classifier.

9. Datasets used: This is the number showing how many datasets are used.

| | ***Parameters*** →<br>***Approach*** | **Base Classifier** | **Split Measure** | **No of Passes *** | **Combine Strategy** | **Mtry** | **Stopping Criterion** | **Pruning of Forest** | **Parallel Extension** | **Datasets Tested** | **Key Features** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Accuracy Improvement** | **Meta Random Forest** | Random Forest | Gini Index | MP | Majority Voting | √M | Fixed apriori | No | No | 10 | Random Forest as base classifier, Bagging and Boosting techniques for ensemble |
| | **Improved Random Forest** | Decision Tree | Gini, Info Gain, MDL ReliefF | MP | Weighted Voting | √M | Fixed apriori | No | No | 17 | Multiple split measures, Weighted voting |
| | **Forest RK** | Decision Tree | Gini Index | MP | Majority Voting | KЄ [1,M] | Fixed apriori | No | No | 10 | Features randomly selected at each node during tree induction, McNemar test for comparison |
| | **Dynamic Integration Random Forest** | Decision Tree | Info Gain | MP | Dynamic Integrated Voting | Log$_2$M+ 1 | Fixed apriori | Yes | No | 27 | Use of distance measures HEOM and Intrinsic similarity |
| **Performance Improvement** | **BAGA** | Decision Tree | Info Gain | MP | Majority / probabilistic voting | M | Fixed apriori | Yes | No | 2 | Overproduce, and Genetic algorithm strategy |
| | **Selection of DTs in RF** | Decision Tree | Gini Index | MP | Majority Voting | √M | Fixed apriori | No | No | 10 | Overproduce & Choose, SFS, SBS approach, McNemar nonparametric test for classifier comparison |
| | **Dynamic Random Forests** | Random Tree | Info Gain | SP | Majority Voting | KЄ [1,M] | Fixed apriori | No | No | 20 | Weighted training samples as in boosting, sequential process |
| **Naive Versions** | **Fuzzy Random Forest** | Fuzzy Decision Tree | Info Gain | MP | Majority Voting | √M | Fixed apriori | No | No | 2 | Use of Fuzzy partition to generate Fuzzy decision tree |
| | **Semi-supervised Random Forest** | Decision Tree | Info Gain / Gini index | MP | Majority Voting | √M | Fixed | No | Yes-GPU | 3 | Use of labeled and unlabelled data, Maximum margin approach using Deterministic Annealing |
| **Online Versions** | **Online Random Forest** | Extremely Randomized Tree | Info Gain | SP | Majority Voting | √M | Fixed apriori | Yes | Yes-GPU | 7 | Online bagging, Temporal weighting for forest pruning |
| | **Streaming Random Forest** | Decision Tree | Gini Index | SP | Majority Voting | Log$_2$M+ 1 | Fixed apriori | No | No | 2 | Use of Hoeffiding bound, Limited pruning of base tree |
| | **Incremental Extremely Random F** | Extremely Randomized Tree | Gini Index | SP | Majority Voting | √M | Fixed apriori | No | No | 7 | Small number of labeled examples, Used for video tracking |
| **Data Specific** | ***Random Forest for Imbalanced Data*** | Decision Tree | Gini Index | MP | Majority Voting /Weighted voting | √M | Fixed apriori | No | No | 5 | Weights on minority class, Down-sampling majority class |

Fig.2 Comparison Chart (* MP - Multiple Passes, SP - Single Pass)

RECENT SCIENCE Publishing Research

© RECENT SCIENCE PUBLICATIONS ARCHIVES | April 2013|$25.00 | 27702358 |
*This article is authorized for use only by Recent Science Journal Authors, Subscribers and Partnering Institutions*

10. Key Features: It describes the core ideas / concepts used in the approach related to Random Forest classifier.

THIS COMPARISON CHART WILL BE OF HELP TO THOSE WHO ARE ASPIRING TO TAKE UP RESEARCH RELATED TO RANDOM FOREST CLASSIFIER.

# 5 FUTURE RESEARCH DIRECTIONS

## 5.1 Based on Accuracy Improvement

Accuracy improvements in Random Forest are possible using different attribute split measures, using different combine functions, or using both. Achieving diversity in base classifiers is an ongoing quality improvement process which will improve accuracy. Hence, finding ways to achieve diversity definitely has future scope for research. It is possible to use OOB estimates, proximity computation, and variable importance features more prominently for improving accuracy of Random Forest classifiers.

## 5.2 Based on Performance Improvement

Random forest algorithm generates many classification trees and generation of each tree is independent of each other. Thus, Random Forest is by nature a suitable candidate for parallel processing. Additionally, data mining is usually performed on very large datasets, and Random Forest can work well on datasets with large number of predictors. As mentioned in Section 4, each parallel implementation of Random Forest is specific to some platform or language. Thus, there is scope for generalized Parallel Algorithm for Random Forest. With the geographical spread of business and the world getting connected with the Internet; business data is distributed at different locations. Hence, design of Distributed Random Forest algorithm is another important future research direction.

Theoretical and empirical results have proved that beyond a certain number, increasing the number of trees in the forest does not yield increase in accuracy. Previous research work in this direction takes static approach, i.e. first build a forest to its full extent and then shrink / prune it by deleting some of the trees which are not contributing towards increase in accuracy. This approach is not cost effective from the viewpoint of time and memory. Also, it reduces only time taken by classification and not by learning. There is scope to generate dynamic techniques to prune the forest size on the fly. Also, no research work has yet shown what will be the optimal subset of forest which will work with accuracy of the original forest.

## 5.3 Data Specific Improvements

Almost all classifiers have problem in classifying imbalanced data; they have a tendency to ignore minority classes. As there are many real life problems that deal with imbalanced data such as Fraud detection, Network intrusion, Rare disease diagnosing, etc; classifiers for imbalanced data are in demand. Earlier results have shown that Random Forest with suitable modification gives better results over other classifiers for imbalanced data sets. Hence, there is scope to propose a new modified Random Forest algorithm for Imbalanced data. Using Random Forest as a base learner can achieve good results in the domain of Imbalanced data.

As per Breiman, Random Forest can handle thousands of input variables without variable deletion. In case of applications where nature of data is such that number of samples available is less than number of predictors, i.e. n << p, Random Forest can work very well and there is scope for research in this direction.

## 5.4 Online versions of Random Forest

Online continuous and endless data stream processing is a challenge for machine learning community. Improvement in accuracy and performance for Random Forest with Stream data is a prominent field for research. Experimenting using different attribute split measures, combine functions, pruning of forest based on tree performance, proper handling of concept drifts, and parallel algorithm for Random Forest using stream data are some of the directions for future research in this area.

## 5.5 Naïve approach

Random Forests using Semi Supervised Learning (SSL) approach is an open field for research. With SSL approach, it is possible to construct a classifier using combination of labeled and unlabeled data. This approach is useful for both offline and online data problems. Especially with stream data where decision tree construction is based on Hoeffiding bound statistics, the number of data samples needed at each node for splitting is huge, and in this case SSL approach can be effective.

# 6. CONCLUDING REMARKS

The intension of this paper was to present a review of current work related to Random Forest classifier and identify future research directions in the field of Random Forest classifier. Random Forest classifier is an ensemble technique and hence is more accurate, but it is time consuming compared to other individual classification techniques. We mainly tried to review the work done for accuracy improvement and performance improvement of Random Forest. As a result of our survey, we have presented Taxonomy of Random Forest algorithm and performed analysis of various algorithms / techniques based on Random Forest algorithm. This analysis which is presented as Comparison chart will serve as a guideline for pursuing future research related to Random forest classifier.

# REFERENCES

[1] Abdulsalam H, Skillicorn B, Martin P, Streaming Random Forests, Proceedings of 11th International Database and Engineering Applications Symposium, Banff, Alta pp 225-232, (2007) .

[2] Bernard S, Heutte L, Adam S, Using Random Forest for Handwritten Digit Recognition, International Conference on Document Analysis and Recognition 1043-1047, (2007)

[3] Bernard S, Heutte L, Adam S, Towards a Better Understanding of Random Forests Through the Study of Strength and Correlation, ICIC Proceedings of the Intelligent Computing 5th International Conference on Emerging Intelligent Computing Technology and Applications, (2009)

[4] Bernard S, Heutte L, Adam S, Forest-RK : A New Random Forest Induction Method, Proceedings of 4th International Conference on Intelligent Computing: Advanced Intelligent Computing Theories and Applications – with Aspects of Artificial Intelligence, Springer-Verlag, (2008)

[5] Bernard S, Heutte L, Adam S, On the Selection of Decision Trees in Random Forest, Proceedings of International Joint Cobference on Neural Networks, Atlanta, Georgia, USA, June 14-19,302-307, (2009)

[6] Bernard S, Heutte L, Adam S, Dynamic Random forests, Pattern Recognition Letters, 33 (2012), 1580-1586

[7] Boinee P, Angelis A, Foresti G, Meta Random Forest, International Journal of Computational Intelligence 2, (2006)

[8] Bonissone P, Candenas J, Garrido M, Diaz R, A Fuzzy Random Forest: Fundamental for Design and Construction, Studies in Fuzziness and Soft Computing, Vol 249, 23-42, (2010)

[9] Bonissone P, Cadenas J, Garrido M, Diaz-Valladares R, A Fuzzy Random Forest, International Journal of Approximate Reasoning, 51, 729-747, (2010)

[10] Breiman L, Bagging Predictors , Technical report No 421, (1994)

[11] Brieman L, Random Forests, Machine Learning, 45, 5-32, (2001)

[12] Chain C, Liaw A, Breiman L, Using Random forest to Learn Imbalanced Data, Technical Report, Department of Statistics, U. C. Berkley (2004)

[13] Crawford M, Ham J, Chen Y, Ghosh J, Random Forests of Binary Hierarchical Classifiers for Analysis of Hyper-spectral Data, Advances in Techniques for Analysis of Remotely Sensed Data, 337-345, IEEE, (2003)

[14] Gaber M, Zaslavsky A, Krshnaswamy S, Mining Data Streams: A Review, SIGMOD Record, Vol 34 No 2, (2005)

[15] Geurts P, Ernst D, Wehenkel L, Extremely Randomized Trees, Machine Learning, volume 63, 3-42, (2006)

[16] Guo L, Ma Y, Cukic B, Singh H, Robust Prediction of Fault-Proneness by Random Forests, Proceedings of the 15th International Symposium on Software Reliability Engineering, IEEE, (2004)

[17] Grahn H, Lavesson N, Lapajne M, Slat D, A CUDA implementation of Random Forest – Early Results, Master Thesis Software Engineering, School of Computing, Blekinge Institute of Technology, Sweden

[18] Hansen L, Salamon P, Neural Network Ensembles, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol 12 No 10, (1990)

[19] I. H. Witten, E. Frank, Weka: Practical machine learning tools and techniques, Morgan Kaufmann publisher, (2005)

[20] Kosorok M, Ma S, Marginal Asymptotics for the Large p Small n paradigm: With Applications to Microarray Data, Ann Statist 35, 1456-1486, (2007)

[21] Kouzani A, Nasireding G, Multilabel Classification by BCH Code and Random forest, International Journal of Recent Trends in Engineering, Vol 2, No 1, (2009)

[22] Krogh A, Vedelsby J, Neural Network Ensembles, Cross Validation, and Active Learning, Advances in Neural Information Processing Systems Vol 7, MIT Press , 231-238, (1995)

[23] Kuncheva L, Diversity in Multiple Classifier Systems, Information Fusion, Vol 6, Issue 1, 3-4, (2005)

[24] Latinne P, Debeir O, Decastecker C, Limiting the number of trees in Random Forest, MCS, UK (2001)

[25] Leistner C, Saffari A, Santner J, Godec M, Bischof H, Semi-Supervised Random Forests, ICCV IEEE, Conference Proceedings, 506-513 (2009)

[26] Maudes J, Rodridugz J, Garcia-Osorio C, Disturbing Neighbors diversity for decision forests, Studies in Computational Intelligence, Vol 245, 113-133, (2009)

[27] Nikulin V, McLachlan G, Ng S, Ensemble Approach for Classification of Imbalanced Data, Proceedings of the 22nd Australian Joint Conference on Advances in Artificial Intelligence, Springer-Verlag (2009)

[28] Opitz D, Shavlik J, Generating Accurate and Diverse Members of a Neural-Network Ensemble, Advances in Neural Information Processing Systems Vol 8, MIT Press , (1996)

[29] Opitz D, Maclin R, Popular Ensemble Methods: An Empirical Study, Journal of Artificial Intelligence 11, 169-198, (1999)

[30] Oza, Russell S, Online Bagging and Boosting, Proceedings of Artificial Intelligence and Statistics, 105-112, (2001)

[31] Pal M, Random Forests for Land Cover Classification, Proceedings of Geoscience and Remote Sensing Symposium, IEEE, 3510-3512, (2003)

[32] Robert E Schapire, The Boosting Approach to Machine Learning an Overview, Nonlinear Estimation and Classification, Springer, 2003

[33] Prenger R, Lemmond T, Varshney K, Chen B, Hanley W, Class-Specific Error Bounds for Ensemble Classifiers, KDD'10, Washington DC, USA, (2010)

[34] Robnik M, Sikonja, Improving Random Forests, J F Boulicaut et al (eds): Machine Learning, ECML 2004 Proceedings, Springer, Berlin, (2004)

[35] Rodriguze J, Kuncheva L, Rotation Forest: A New Classifier Ensemble Method, IEEE Transaction on Pattern Analysis and Machine intelligence, Vol 28, N0 10, 1619-1630, (2006)

[36] Roli F, Giacinto G, Vernazza G, Methods for Designing Multiple Classifier Systems, Second International Workshop on Multiple Classifier Systems, Springer-Verlag, (2001)

[37] Saffari A, Leistner C, Santner J, Godec M, Bischof H, On-line Random Forests, ICCV IEEE, Conference Proceedings 1393-1400, (2009)

[38] Tsymbal A, Pechenizkiy M, Cunningham P, Dynamic Integration with Random Forest, ECML, LNAI, 801-808, Springer-Verlag (2006)

[39] Verikas A, Gelzinis A, Bacauskiene M, Mining data with random forests: A survey and results of new tests, Pattern Recognition 44 , 330 - 349, (2011)

[40] Wang A, Wan G, Cheng Z, Li S, An Incremental Extremely Random Forest Classifier for Online Learning and Tracking, 16th IEEE International Conference on Image Processing,1449-1452, (2009)

[41] Wu X, Chen Z, Toward Dynamic Ensemble: The BAGA Approach, Proceedings of the ACS/ IEEE International Conference on Computer Systems and Applications,(2005)

[42] Ye Y, Li H, Deng X, Huang J, Feature Weighting Random Forest for Detection of Hidden Web Search Interfaces, Computational Linguistic and Chinese Language Processing, Vol 13, No 4, 387-404, (2008)

[43] Zhang H, Wang M, Search for the smallest Random Forest, Statistics and Its Interface Volume.2, pp 381-388, (2009)

[44] E Tripoli, D Fotiadis, G Manis, " Dynamic Construction of Random Forests: Evaluation using Biomedical Engineering Problems", IEEE, 2010