A Novel Consistent Random Forest Framework: Bernoulli Random Forests

Yisen Wang, Shu-Tao Xia, Qingtao Tang, Jia Wu, *Member, IEEE*, and Xingquan Zhu, *Senior Member, IEEE*

***Abstract*— Random forests (RFs) are recognized as one type of ensemble learning method and are effective for the most classification and regression tasks. Despite their impressive empirical performance, the theory of RFs has yet been fully proved. Several theoretically guaranteed RF variants have been presented, but their poor practical performance has been criticized. In this paper, a novel RF framework is proposed, named Bernoulli RFs (BRFs), with the aim of solving the RF dilemma between theoretical consistency and empirical performance. BRF uses two independent Bernoulli distributions to simplify the tree construction, in contrast to the RFs proposed by Breiman. The two Bernoulli distributions are separately used to control the splitting feature and splitting point selection processes of tree construction. Consequently, theoretical consistency is ensured in BRF, i.e., the convergence of learning performance to optimum will be guaranteed when infinite data are given. Importantly, our proposed BRF is consistent for both classification and regression. The best empirical performance is achieved by BRF when it is compared with state-of-the-art theoretical/consistent RFs. This advance in RF research toward closing the gap between theory and practice is verified by the theoretical and experimental studies in this paper.**

***Index Terms*— Classification, consistency, random forests (RFs), regression.**

I. Introduction

RANDOM forest (RF) is one type of very popular ensemble learning method in which numerous randomized decision trees are constructed and combined to form an RF that is then used for classification or regression. It is extremely easy and efficient to train such RFs [1]. RFs are very powerful because of the vote/average mechanism, and they have achieved great success in many cross-domain applications (e.g., chemoinformatics [2], bioinformatics [3], [4], ecology [5], [6], computer vision [7], [8], and data mining [9], [10]).

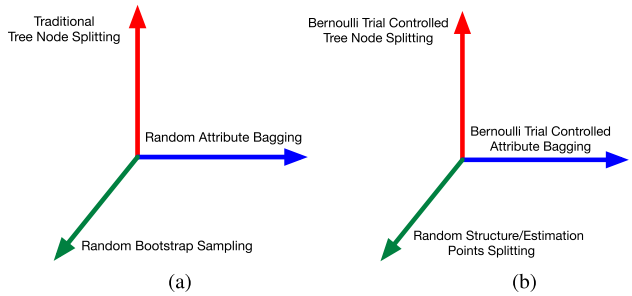


Fig. 1. Illustration of (a) Breiman RF and (b) BRF. Breiman RF has a deterministic tree node splitting process, which leads to highly data-dependent trees. In contrast, BRF introduces randomness via Bernoulli controlled tree construction. This kind of randomness makes the trees less data-dependent without sacrificing their learning performance.

In contrast to the attractive practical performance of RFs in many real-world applications, their theoretical properties have yet to be fully established and are still the subject of active research. For a learning algorithm, *consistency* is the most fundamental theoretical property because it guarantees convergence to optimum as the data grow infinitely large. RFs employ a randomized instance bootstrapping, a randomized feature bagging, and a deterministic tree construction, and thus it is not easy to prove the consistency of RFs. As shown in Fig. 1(a), instance bootstrapping and feature bagging are two random processes in Breiman RF whose goal is to construct a less data-dependent tree, whereas the traditional tree node splitting process of tree construction is determined by a data-driven criterion (such as Gini index [11]–[13]). Consequently, the above procedures result in data-dependent trees, which make it difficult to theoretically analyze RFs.

Given the difficulty of analyzing the consistency of RFs, several RF variants have been proposed [14]–[20] to incorporate more randomness and relax or simplify the deterministic tree construction process: 1) substituting random sampling of a single feature for feature bagging or 2) using a more elementary splitting criterion instead of the common complicated impurity-based one to split the tree node. A less data-dependent tree structure is the objective of both approaches and this also applies to the consistency analysis. Unfortunately, such approaches usually result in

Manuscript received July 2, 2016; revised May 27, 2017 and July 8, 2017; accepted July 10, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61371078 and Grant 61375054, in part by the R&D Program of Shenzhen under Grant JCYJ20140509172959977, Grant JSGG20150512162853495, Grant ZDSYS20140509172959989, and Grant JCYJ20160331184440545, in part by the Australian Research Council Discovery Projects under Grant DP140100545 and Grant DP140102206, and in part by the Program for Professor of Special Appointment (Eastern Scholar) at the Shanghai Institutions of Higher Learning. *(Yisen Wang and Qingtao Tang contributed equally to this work.) (Corresponding author: Jia Wu.)*

Y. Wang, S.-T. Xia, and Q. Tang are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: wangys14@mails.tsinghua.edu.cn; xiast@sz.tsinghua.edu.cn; tqt15@mails.tsinghua.edu.cn).

J. Wu is with the Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, NSW 2109, Australia (e-mail: jia.wu@mq.edu.au).

X. Zhu is with the Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: xzhu3@fau.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2729778

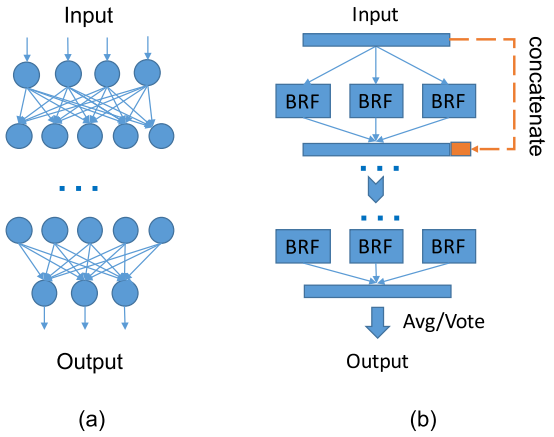


Fig. 2. Conceptual view of using RFs to build deep learning architecture that resembles deep neural networks. (a) Traditional deep neural network in which each node denotes a neuron. (b) Deep forest-like architecture [25] in which an RF serves as a "forest neuro" of the network. The "concatenate" and "vote" resemble the nonlinear transformation procedures in deep learning. The theoretical consistency of the proposed BRF "forest neuro" forms a clear base for understanding the theoretical properties of such deep architecture learning machines.

poor performance for classification or regression, even though they have theoretically analyzed properties. The dilemma between theoretical consistency and empirical soundness continually inspires active research in this field.

The above observations motivate us to propose a novel RF framework in this paper, named Bernoulli RFs (BRFs). It not only has proven theoretical consistency but also has comparable performance to Breiman RF. As illustrated in Fig. 1(b), the key factor lies in the two Bernoulli driven/controlled tree construction processes. A certain degree of randomness as well as the overall quality of the trees is ensured simultaneously. Because a probability value-controlled random process is involved in the Bernoulli trial, the tree construction in BRF can be either random or deterministic depending on the probability value. Therefore, a much less data-dependent tree structure is obtained by BRF compared with Breiman RF, yet BRF demonstrates much better performance than all the existing RFs with theoretical consistency.

The main **contributions** of this paper are threefold.

1) BRF has fully proven theoretical consistency, and it has the fewest simplification changes compared with Breiman RF.
2) We provide an approach for resolving the dilemma between theoretical consistency and empirical soundness through the Bernoulli distributions controlled splitting feature and splitting point selection.
3) A large number of experiments demonstrate the superiority of BRF over state-of-the-art theoretical/consistent RFs.

Our proposed BRF could also advance research in neural networks and learning systems. In traditional RF learning, trees are generated and combined through a single layer concatenation. Motivated by the recent success of deep architecture [21], several researchers have proposed the incorporation of RFs into deep neural networks [22]–[25]. One such architecture is shown in Fig. 2 where an RF serves as a "forest neuro," with such neuros being stacked in multiple layers in a deep learning fashion. One one hand, the performance of RFs can be boosted through deep representation learning. On the other hand, some issues of deep neural networks can be partially addressed by the "forest neuro." For example, deep neural networks have a huge number of hyperparameters that need to be tuned carefully, whereas forests have a very few hyperparameters and are not sensitive to these parameters. Therefore, this kind of deep forests could reduce burdensome parameter tuning. Moreover, the BRF "forest neuro" has proven theoretical properties compared to its peer neural network neuro, which provides a suitable approach to conduct the theoretical analysis of deep models.

The remainder of this paper is organized as follows. Section II reviews RF-related methodology as well as theoretical work on consistency. Section III describes the proposed BRF. Section IV outlines the consistency proof of BRF. Section V demonstrates the extension of BRF to the classification problem. Section VI discusses the differences between several consistent RFs, followed by empirical comparisons in Section VII. The conclusions of this paper are drawn in Section VIII.

II. Related Work

A. Methodology of Random Forests

Breiman [1] first proposed RFs two decades ago, inspired by primary work [26] in the feature selection technique [27], the random subspace method [28], and the random split selection approach [29]. Because of their ability to effectively handle various types of data, RFs have achieved huge success in numerous fields (see [2]–[10]). Below, we briefly introduce the RF framework. Interested readers can refer to [1] and [30] for comprehensive technical details.

Let us assume a data set $\mathcal{D}_n$ with n instances $(\boldsymbol{X}, Y)$, where $\boldsymbol{X} \in \mathbb{R}^D$. Breiman's approach combines numerous independently trained decision trees to form a forest. We can regard the tree construction procedure of each tree as a partition of data space. That is to say, if the full data space is $\mathbb{R}^D$, then a leaf is a partition of $\mathbb{R}^D$ and each node corresponds to a hyperrectangular cell of data space. The details of the RF algorithm are given below.

1) At the beginning of the tree construction, we randomly sample n data points from the given data set $\mathcal{D}_n$ with replacement [31]. These and only these bootstrap samples are used for constructing the current tree.
2) Classification and Regression Tree [11] is adopted. For each tree node, $mtry$ features ($mtry < D$) are first randomly sampled from the original D features, which are then used for selecting the splitting feature and splitting point. The criterion is the largest Gini impurity decrease or the largest mean squared error (MSE) reduction for classification and regression, respectively. The tree nodes are constructed one by one following the above procedure until the stopping condition is reached, e.g., the threshold of the instance size in leaf nodes.
3) RFs make predictions by averaging the result from each tree, i.e., a majority vote from Y for classification, or the average of Y for regression.

The above summary shows that there are three key aspects to RFs: 1) the method that injects randomness into the trees (bootstrap sampling); 2) the tree construction approach; and 3) the type of prediction from each tree.

B. Consistency of Random Forests

Despite the excellent practical performance of RFs, research on their theoretical analysis has been slow [32]. Breiman [1] offered the first theoretical result, noting that the generalization error is bounded by the strength of individual trees and the correlation of multiple trees. Lin and Jeon [33] subsequently highlighted the relation of RFs and a type of the nearest neighbor-based estimators. A further investigation of this direction can be found in [34].

A crucial theoretical breakthrough in the study of RFs was conducted in [15], which proved the consistency of two direct simplification models of Breiman RF—the random selection of feature and splitting points. The splitting feature for each tree node is selected uniformly and randomly from all original features. The splitting point is also chosen uniformly and randomly from the values of the selected feature.

Based on intuition and Breiman's mathematical heuristics technical report [14], Biau [18] proved another simplified version of Breiman RF, in which the simplification aspects are less than those in [15]. Each node of each tree is built by randomly selecting a subspace of features, and the splitting point is fixed at the midpoint of the data in the node for each candidate feature. To choose between candidate features, the splitting feature and splitting point with the greatest decrease in impurity are selected to grow the tree.

Denil *et al.* [19] analyzed a new variant of RFs that is quite similar to Breiman RF. The subspace of candidate features for each node is selected in a Poisson distribution. The main difference lies in how the splitting points are chosen. For each candidate feature, a subset (e.g., m) of data points is randomly selected and a search is conducted to find the best splitting point, which gives the maximal reduction of squared error (the regression problem) in the range defined by the preselected data points.

Although the above RF methods [15], [18], [19] enjoy their theoretical consistency, their empirical performance is, however, significant inferior to the original Breiman RF, mainly because these methods [15], [18], [19] employ too much randomness in the tree construction, which inherently reduce the superiority of the optimized tree construction of RFs. Accordingly, in [20], we have proposed a Bernoulli controlled tree construction in the forests under the classification task, but this approach is only proved to be consistent for classification. In addition, our previous work [20] only considers the following two cases: 1) randomly choosing a single feature and a data point as the splitting feature and splitting point with probabilities p_1 and p_2, respectively, and 2) using Breiman RF method to choose the splitting feature and splitting point with probabilities $1 - p_1$ and $1 - p_2$, respectively. In this paper, we propose a novel complete RF framework—named BRFs—that is useful for both regression and classification tasks. BRF considers all four probability combination cases, including $(p_1;\ p_2)$, $(p_1;\ 1 - p_2)$, $(1 - p_1;\ p_2)$, and $(1 - p_1;\ 1 - p_2)$. We also prove the theoretical consistency of BRF and evaluate its empirical performance on 23 regression tasks and 27 classification tasks, which confirm that BRF outperforms all the existing methods [15], [18], [19] on all tasks. Moreover, we assess the influence of all parameters in BRF, i.e., the ratio of structure points to entire points, the number of trees, and the probabilities p_1 and p_2. The computational cost of BRF is also analyzed in this paper.

In addition to generic RFs, several special RFs have also been demonstrated to be consistent. For example, Meinshausen proved the consistency of RFs for quantile regression in [35]. Random survival forests were proved to be consistent in [36]. An online version of RFs was proven to be consistent in [37].

III. Proposed Random Forests

To achieve theoretical consistency while retaining good performance, the proposed BRF differs in three ways from Breiman RF, as illustrated in Fig. 1. To simplify the structure of this paper, we first discuss the regression problem in this section and present the extension to the classification problem in Section V.

A. Training Data Set Partitioning

Given a training data set $\mathcal{D}_n$ with D-dimensional features and n sample points, for each sample point $(\boldsymbol{X}, Y)$, $\boldsymbol{X} \in \mathbb{R}^D$, where $\boldsymbol{X}$ represents the features and Y is a real value representing the target variable. The training data set $\mathcal{D}$ is first partitioned into a **Structure part** and an **Estimation part**. The structure and estimation parts play different roles when the trees are constructed, which is important for achieving the consistency property of the proposed BRF (shown in Lemma 3).

The **Structure part** is used when the trees are constructed. The best feature and splitting points in each splitting node are chosen only on the structure part, not on the estimation part, and the structure part is not used for prediction.

The **Estimation part** is only used for prediction (i.e., averaging Y in tree leaves), not for tree construction. Note that when a prediction is made, the estimation part is split by the rules created in the tree construction based on the structure part, but the estimation part has no effect on tree construction.

The training data set is partitioned randomly and independently when each tree is constructed. The ratio of the two parts is defined as Ratio = (No. of Structure part/No. of Entire points), the influence of which is assessed in the experiments.

B. Tree Construction

In the proposed BRF, unlike the classical RF, the training data set partitioning is adopted instead of the bootstrap technique. Two Bernoulli distributions are adopted when the features and splitting points are selected.

The first novelty of the proposed model is that instead of traversing all the candidate features, our candidate feature selection is based on a Bernoulli distribution. Assume that B_1 is an event choosing one value from 0 or 1 with probability;

therefore, we can say that B_1 follows a Bernoulli distribution with a probability of p_1 taking 1, or 0 otherwise. We define $B_1 = 1$ if one candidate feature is chosen and $B_1 = 0$ if $\sqrt{D}$ candidate features are uniformly randomly chosen. In each splitting, one candidate features are chosen with p_1 probability, and $\sqrt{D}$ candidate features are chosen with $1 - p_1$ probability. The single feature is to guarantee consistency and $\sqrt{D}$ is to maintain performance. In the RFs literature, setting the number of candidate features to $\sqrt{D}$ generally gives near optimum results [38], [39], so we adopt this value here.

The second novelty is that the splitting point selection is based on two different methods. Similar to B_1, $B_2 \in \{0, 1\}$ is assumed to satisfy a Bernoulli distribution which, with p_2 probability, takes 1. If $B_2 = 1$, the random sampling method is used, otherwise, we adopt the impurity criterion method. Therefore, with p_2 probability, the splitting point is selected through random sampling; with $1 - p_2$ probability, the splitting point is selected through the impurity criterion.

In a regression problem, the impurity decrease is based on MSE, denoted by

$$\mathrm{MSE}(\mathcal{D}^S) = \frac{1}{N(\mathcal{D}^S)} \sum_{(\boldsymbol{X},Y)\in\mathcal{D}^S} (Y - \bar{Y})^2 \tag{1}$$

where $N(\mathcal{D}^S)$ counts the number of structure part in $\mathcal{D}$ and $\bar{Y}$ is the sample mean of the structure part in $\mathcal{D}$.

Thus, the MSE reduction is

$$I(s) = \mathrm{MSE}(\mathcal{D}^S) - \mathrm{MSE}(\mathcal{D}^{lS}) - \mathrm{MSE}(\mathcal{D}^{rS}). \tag{2}$$

The best splitting point is selected by maximizing the above (2). $\mathcal{D}$ is the training data set in the parent node, including the structure part $\mathcal{D}^S$ and the estimation part $\mathcal{D}^E$. $\mathcal{D}^l$ and $\mathcal{D}^r$ are the training data sets in the child nodes that will be generated when $\mathcal{D}$ is split at s.

Through the two steps above, one feature and its corresponding splitting point are chosen to grow the tree. It is worth noting that the tree construction only uses the structure part while the prediction only involves the estimation part. The process is repeated until the given stopping criteria are satisfied.

Similar to classical RF, the proposed BRF's stopping condition is also related to minimum leaf size, but this restriction is on the estimation part rather than the whole training data set, i.e., for each leaf, the instance size of estimation part is bigger than k_n. k_n is the low-order infinity of the number of training instances n, i.e., $k_n \to \infty$ and $k_n/n \to 0$ when $n \to \infty$.

C. Prediction

After the trees in our method have been constructed by the structure part and the sample means have been estimated based on the estimation part, BRF can make predictions for a newly given $\boldsymbol{x}$ as follows.

Each tree can make predictions separately. f represents the base decision tree created by the proposed BRF. For any query point $\boldsymbol{x}$, the prediction of the tree is the average in the leaf

$$\hat{y} = \frac{1}{N(A^E(\boldsymbol{x}))} \sum_{(\boldsymbol{X},Y)\in A^E(\boldsymbol{x})} Y \tag{3}$$

where $N(A^E(\boldsymbol{x}))$ is the instance size of the estimation part in the corresponding leaf node in which the given $\boldsymbol{x}$ falls.

The prediction of the forests is the sample average of all the trees

$$\overline{\hat{y}} = \frac{1}{M} \sum_{j=1}^{M} \hat{y}_j \tag{4}$$

where M is a hyperparameter representing the tree number in the forests. Although the leaf contains both a structure part and an estimation part, only the estimation part is used for prediction.

D. BRF Algorithm

We summarize the proposed BRF framework in a pseudocode format in Algorithms 1 and 2. Algorithm 1 is a complete process for the prediction of an instance. Algorithm 2 is the detailed construction procedure of decision trees in BRF.

Algorithm 1 BRF Prediction Value at a Query Point $\boldsymbol{x}$

1: **Input:** Training data $\mathcal{D}_n$, number of trees $M \in \mathcal{N}^+$, the parameter k_n.
2: **Output:** Prediction of BRF at $\boldsymbol{x}$.
3: **for** $j = 1, 2, \ldots, M$ **do**
4: Structure part, Estimation part ← Partitioning training data $\mathcal{D}_n$
5: A tree in Bernoulli random forests ← BRF_Tree (Structure part, Estimation part, k_n) // Algorithm 2
6: Leaf predictor of individual tree ← Using Estimation part following Eq. (3).
7: **end for**
8: Compute the predicted value of Bernoulli random forests by averaging each tree result according to Eq. (4).
9: **Return:** Predicted value at $\boldsymbol{x}$

To sum up, the proposed BRF method introduces two independent Bernoulli distributions for tree construction and prediction, in contrast to Breiman RF. Because of the two Bernoulli distributions, the proposed BRF not only introduces a certain degree of randomness to the feature and splitting point selection, but also retains the sound performance of Breiman RF.

IV. Proof of Consistency

In this section, we first show the necessary preliminaries for the consistency proof, and then the detailed proof of consistency of our proposed method. Our proof adopts the following principles. Because RFs consist of decision trees, we transfer the consistency of an RF to its decision trees. The tree is well known as a partition of the data space. Intuitively, the consistency of the decision trees is transferred to the associated partition rule.

In the proof, a random variable $\mathcal{C}$ represents the randomness in the process of tree construction, including the randomness when we select features and splitting points. For space and clarity, only the essential proofs are included in this section. Other proofs are given in Appendix A–E.

Algorithm 2 BRF_Tree: BRF Decision Tree Construction

1: **Input:** Structure part, Estimation part, parameter k_n.
2: **Output:** The decision tree in Bernoulli random forests.
3: **while** stop condition is false **do**
4: Assume one node of the tree is $\mathcal{D}$, containing structure part $\mathcal{D}^S$ and estimation part $\mathcal{D}^E$.
// Tree Construction with structure part:
5: Select subspace of candidate features. Choose one candidate feature in probability p_1 or $\sqrt{D}$ candidate features in probability $1 - p_1$.
6: Select splitting point s using structure part $\mathcal{D}^S$. For each candidate feature, randomly sample a point to split with p_2 probability; or, with $1-p_2$ probability, a splitting point is selected by optimizing the given impurity criterion.
7: Obtain the optimal pair of splitting feature and splitting point following Eq. (2). Accordingly, create two child nodes $\mathcal{D}^l$ and $\mathcal{D}^r$. The structure and estimation parts are correspondingly cut into child nodes, called $\mathcal{D}^{l_S}$, $\mathcal{D}^{l_E}$ and $\mathcal{D}^{r_S}$, $\mathcal{D}^{r_E}$.
// Stop condition with estimation part:
8: **if** the number of estimation part $\mathcal{D}^{l_E}$ and $\mathcal{D}^{r_E}$ are both larger than k_n **then**
9: Go to line 3 for $\mathcal{D}^l$ and $\mathcal{D}^r$, recursively grow tree.
10: **else**
11: Stop condition is true.
12: **end if**
13: **end while**
14: **Return:** A decision tree in Bernoulli random forests

A. Preliminaries

Consistency is a fundamental theoretical property of a learning algorithm that guarantees that the output of the algorithm converges to optimum as the data size closes to infinity.

Definition 1: In regression, given the data set $\mathcal{D}_n$, for the distribution of $(\boldsymbol{X}, Y)$, a series of estimators $\{f\}$ have consistency when the risk function $R(f)$ satisfies

$$R(f) = \mathbb{E}[(f(\boldsymbol{X}, \mathcal{C}, \mathcal{D}_n) - f(\boldsymbol{X}))^2] \to 0, \quad n \to \infty \quad (5)$$

where the underlying unknown function $f(\boldsymbol{X}) = \mathbb{E}[Y|\boldsymbol{X}]$ is the target.

Lemma 1: Suppose a series of estimators $\{f\}$ have the consistency, then the empirical averaging estimator $\overline{f^{(M)}}$, which is the average of M copies of f with different randomness $\mathcal{C}$, has consistency.

From Lemma 1 [15], we only need to prove the consistency of the individual tree to prove the consistency of the forest.

Revisiting the tree construction of BRF, we add the data point partitioning procedure that partitions the training data set into a structure part and an estimation part. The following Lemma 2 proves that the consistency of the decision tree is sufficient to show its consistency on data point partitioning [19].

Lemma 2: Suppose, for the distribution of $(\boldsymbol{X}, Y)$, a series of estimators $\{f\}$ are conditionally consistent

$$\lim_{n\to\infty} \mathbb{E}[(f(\boldsymbol{X}, \mathcal{C}, \mathcal{D}_n) - f(\boldsymbol{X}))^2 | I] = 0 \quad (6)$$

where I represents randomness when we partition the training data set. If the training data set partitioning produces an acceptable structure part and an estimation part with probability 1, and f is bounded, then $\{f\}$ values are unconditionally consistent

$$\lim_{n\to\infty} \mathbb{E}[(f(\boldsymbol{X}, Z, \mathcal{D}_n) - f(\boldsymbol{X}))^2] \to 0. \quad (7)$$

Through the above Lemmas 1 and 2, we conclude that to prove the consistency of our proposed method, we only need to ensure the consistency of the individual tree. To do this, we employ Lemma 3 for partition rules as follows [40].

Lemma 3: Consider a partitioning regression function estimate that builds a prediction by averaging method in each leaf node. If the leaf predictors are fit by the data that are independent of the tree structure, and $\mathbb{E}[Y^2] < \infty$, then the consistency of the above estimate is ensured, provided that: 1) with $n \to +\infty$, the diameter of $\mathcal{N}(\boldsymbol{X}) \to 0$ in probability and 2) with $n \to +\infty$, $N(\mathcal{N}^E(\boldsymbol{X})) \to \infty$ in probability, where $\boldsymbol{X}$ falls into the leaf node $\mathcal{N}(\boldsymbol{X})$ and $N(\mathcal{N}^E(\boldsymbol{X}))$ represents the instance size of the estimation part in the leaf node.

Proof: Refer to [40, Th. 4.1] for more detail. □

It is well known that when decision trees are constructed, the original instance space is partitioned, which implies that the diameter of the leaf node $\mathcal{N}(\boldsymbol{X}) \to 0$ is equivalent to the $\mathcal{N}(\boldsymbol{X})$ corresponding hypercube size approaching 0.

Lemma 3 also provides support for data point partitioning because it requires that the leaf predictors are fit by the data that are independent of the tree structure. More importantly, Lemma 3 states that the consistency of the tree construction can be proven if the hypercubes/cells belonging to leaves approximate 0 but at the same time contain an infinite number of estimation part, when $n \to \infty$.

In summary, Lemmas 1 and 2 assert that the consistency of forests is implied by the consistency of individual trees. To prove the consistency of the individual trees, we employ the consistency condition in Lemma 3 for partition rules, because we need to prove that the decision trees in our proposed BRF satisfy the conditions of the partition rules. If the conditions are satisfied, consistency is proven. The conditions include: 1) the leaf predictors are fit by the data that is independent of the tree structure, and this is met by the data points partitioning procedure and 2) the hypercube corresponding to the leaf should be sufficiently small, but should contain an infinite number of data points, as proven in the following section.

B. Proof of Consistency Theorem

The consistency of our proposed method is proven with the lemmas given above, as follows.

Theorem 1: Assume the support of $\boldsymbol{X}$ is $[0, 1]^D$ and the density of $\boldsymbol{X}$ is not 0 almost everywhere on the support. The cumulative distribution function (CDF) of the selected splitting points is right-continuous at 0 and left-continuous at 1. Our proposed method has consistency when $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$.

To prove the consistency of BRF according to Section IV-A, we only need to ensure the consistency of the individual decision trees. This consistency is guaranteed

by proving the two conditions of Lemma 3. In summary, we only need to prove the diameter of $\mathcal{N}(\boldsymbol{X}) \to 0$ and $N(\mathcal{N}^E(\boldsymbol{X})) \to \infty$ in probability.

Proof: First, in the proposed BRF, since $N(\mathcal{N}^E(\boldsymbol{X})) \geq k_n$ is required, $N(\mathcal{N}^E(\boldsymbol{X})) \to \infty$ is trivial when $n \to \infty$.

Second, we only need to prove that the diameter of the leaf node $\mathcal{N}(\boldsymbol{X})$, $\text{diam}(\mathcal{N}(\boldsymbol{X}))$, approximates 0 in probability. Denoting $Size(a)$ as the size of the ath feature of $\mathcal{N}(\boldsymbol{X})$, we only need to show that $\mathbb{E}[Size(a)]$ approximates 0 for $a \in \{1, 2, \ldots, D\}$.

For a given a, we denote the largest size among its child nodes as $Size^*(a)$. Since the selected splitting point is created by a random sampling in $[0, 1]$ with p_2 probability, or by optimizing the impurity criterion with $1 - p_2$ probability, it is evident that

$$\begin{aligned}\mathbb{E}[Size^*(a)] &\leq (1 - p_2) \times 1 + p_2 \times \mathbb{E}[\max(U, 1 - U)] \\ &= (1 - p_2) \times 1 + p_2 \times \frac{3}{4} = 1 - \frac{1}{4} p_2 \end{aligned} \tag{8}$$

where U is a random sample from $Uniform[0, 1]$ and is generated by the random sampling in $[0, 1]$ with a probability p_2.

Recall that when we choose the candidate features, one candidate feature is selected with p_1 probability or $\sqrt{D}$ candidate features are selected with $1 - p_1$ probability. We now define C_1, C_2 as follows:

$C_1 =$ {One candidate feature is split}

$C_2 =$ {The ath one is exactly the splitting feature}.

We define $Size'(a)$ as the child node size for the ath feature, then

$$\begin{aligned}\mathbb{E}[\text{Size}'(a)] &= \mathbb{P}(C_1)\mathbb{E}[\text{Size}'(a)|C_1] + \mathbb{P}(\bar{C_1})\mathbb{E}[\text{Size}'(a)|\bar{C_1}] \\ &\leq p_1 \times \mathbb{E}[\text{Size}'(a)|C_1] + (1 - p_1) \times 1 \\ &= p_1 \times \big(\mathbb{P}(C_2|C_1)\mathbb{E}[\text{Size}'(a)|C_1, C_2] \\ &\quad + \mathbb{P}(\bar{C_2}|C_1)\mathbb{E}[\text{Size}'(d)|C_1, \bar{C_2}]\big) + (1 - p_1) \\ &\leq p_1 \times \left(\frac{1}{D}\mathbb{E}[\text{Size}^*(a)] + 1 - \frac{1}{D}\right) + (1 - p_1) \\ &\leq 1 - \frac{p_1 p_2}{4D}. \end{aligned} \tag{9}$$

We denote the number of layers from the root to the bottom as K. After K times iteration of (9), we have

$$\mathbb{E}[Size(d)] \leq \left(1 - \frac{p_1 p_2}{4D}\right)^K. \tag{10}$$

This is sufficient for the proof of consistency of our proposed BRF if $K \to \infty$ in probability, which will be shown in Lemma 4. □

Lemma 4: With $n \to \infty$, if the CDF of the splitting points is right-continuous at 0 and left-continuous at 1, each node in the tree of our proposed method, in probability, will be split infinite times.

Proof: Recall that the rule for selecting the splitting point is a random sampling method with a probability of p_2 or optimization of the impurity criterion with a probability of $1 - p_2$. Unlike the deterministic rule in the classical RF, the splitting point rule in our proposed BRF has randomness. Thus, from the root to the bottom, the splitting point in the ith splitting in our proposed BRF is a random variable $W_i (i \in \{1, 2, \ldots, K\})$, whose CDF is denoted as F_{W_i}.

Given a K and $\delta > 0$, the size of the root smallest child is denoted as $M_1 = \min(W_1, 1 - W_1)$, then we have, at least with the probability

$$\begin{aligned}\mathbb{P}(M_1 \geq \delta^{1/K}) &= \mathbb{P}(\delta^{1/K} \leq W_1 \leq 1 - \delta^{1/K}) \\ &= F_{W_1}(1 - \delta^{1/K}) - F_{W_1}(\delta^{1/K}). \end{aligned} \tag{11}$$

The values of features can be scaled to the range $[0, 1]$ for each node, without loss of generality. After K splits, the smallest child has the size of at least δ with the probability at least

$$\prod_{i=1}^{K} (F_{W_i}(1 - \delta^{1/K}) - F_{W_i}(\delta^{1/K})). \tag{12}$$

Equation (12) is on the condition that each splitting uses the same feature. However, even though different features are split, (12) also holds. Since the CDF of W_i, F_{W_i}, is right-continuous at 0 and left-continuous at 1, $\lim_{\delta \to 0} F_{W_i}(1 - \delta^{1/K}) - F_{W_i}(\delta^{1/K}) = 1$. Therefore, $\forall \epsilon_1 > 0$, $\exists \delta_1 > 0$, such that

$$\prod_{i=1}^{K} \left(F_{W_i}\left(1 - \delta_1^{1/K}\right) - F_{W_i}\left(\delta_1^{1/K}\right)\right) > (1 - \epsilon_1)^K. \tag{13}$$

In addition, $\forall \epsilon > 0$, $\exists \epsilon_1 > 0$, such that

$$(1 - \epsilon_1)^K > 1 - \epsilon. \tag{14}$$

Equations (13) and (14) show that after K splits, the size of each node is δ with at least $1 - \epsilon$ probability.

Recalling that the density of $\boldsymbol{X}$ is assumed to be nonzero almost everywhere on the support, all the nodes in our proposed method have a positive measure with respect to $\mu_{\boldsymbol{X}}$. If we define

$$p = \min_{l\text{: a leaf at } K\text{th level}} \mu_{\boldsymbol{X}}(l) \tag{15}$$

it is clear that $p > 0$ because each leaf contains a set of positive measures and the number of leaf nodes is finite.

Assuming the size of the training data set is n, we can denote the number of points falling into leaf $\mathcal{N}(\boldsymbol{X})$ as Binomial(n, p). Without loss of generality, we assume the $Ratio$ is 0.5 and the expectation of the number of sample points in the estimation part is $np/2$. From Chebyshev's inequality, we know that

$$\begin{aligned}&\mathbb{P}(N(\mathcal{N}^E(\boldsymbol{X})) < k_n) \\ &\quad = \mathbb{P}\left(N(\mathcal{N}^E(\boldsymbol{X})) - \frac{np}{2} < k_n - \frac{np}{2}\right) \\ &\quad \overset{(a)}{=} \frac{1}{2}\mathbb{P}\left(\left|N(\mathcal{N}^E(\boldsymbol{X})) - \frac{np}{2}\right| > \left|k_n - \frac{np}{2}\right|\right) \\ &\quad \leq \frac{1}{\left|k_n - \frac{np}{2}\right|^2} \end{aligned} \tag{16}$$

where (a) is from the fact that $\frac{k_n}{n} \to 0$ as $n \to \infty$. The right hand side of (16) $\to 0$ as $n \to \infty$. Therefore, we can conclude that the number of sample points in the estimation part of the leaf node is at least k_n, in probability. From the stopping condition, we know that the tree will stop only when the number of sample points in the estimation part of the node is less than k_n. Thus, $K \to \infty$ in probability. □

V. Extension to Classification

We have demonstrated the construction of BRF in Section III and proven its consistency for regression in Section IV. BRF is also consistent for the classification problem, which is discussed in this section.

Suppose we have a training data set $\mathcal{D}_n$ with D-dimensional features and n sample points, $(\boldsymbol{X}, Y)$ represents one sample point, where $\boldsymbol{X} \in \mathbb{R}^D$, and $Y \in \{1, 2, \ldots, C\}$ is the class.

Similar to, but not the same as, the regression problem, we modify the tree construction with two major steps to make BRF suitable for the classification problem. The first is the impurity criterion, which is based on Gini index, denoted by

$$I(v) = T(\mathcal{D}^S) - \frac{|\mathcal{D}^{lS}|}{|\mathcal{D}^S|} T(\mathcal{D}^{lS}) - \frac{|\mathcal{D}^{rS}|}{|\mathcal{D}^S|} T(\mathcal{D}^{rS}). \quad (17)$$

Here the function $T(\mathcal{D}^S)$ is the impurity criterion, which is computed based only on the structure part $\mathcal{D}^S$.

The second is the prediction procedure, which uses votes to replace averages. Assuming the classifier created by our proposed BRF is g and the unlabeled instance for test is $\boldsymbol{x}$, the probability belonging to c class is

$$\gamma^{(c)}(\boldsymbol{x}) = \frac{1}{N(\mathcal{N}^E(\boldsymbol{x}))} \sum_{(\boldsymbol{X},Y) \in \mathcal{N}^E(\boldsymbol{x})} \mathbb{I}\{Y = c\} \quad (18)$$

and the prediction is given by maximizing $\gamma^c(\boldsymbol{x})$

$$\hat{y} = \arg\max_c \{\gamma^{(c)}(\boldsymbol{x})\} \quad (19)$$

where $\mathbb{I}(\cdot)$ is 1 if $\cdot$ is true and is 0 if $\cdot$ is false. The prediction of the proposed BRF is

$$\overline{\hat{y}} = \arg\max_c \sum_{j=1}^{M} \mathbb{I}\{\hat{y}^{(j)}(\boldsymbol{x}) = c\}. \quad (20)$$

In the classification framework, consistency is defined as follows.

Definition 2: In classification, given the training data set $\mathcal{D}_n$, for a distribution of $(\boldsymbol{X}, Y)$, a series of classifiers $\{h\}$ has consistency if

$$\mathbb{E}[L] = \mathbb{P}(h(\boldsymbol{X}, \mathcal{C}, \mathcal{D}_n) \neq Y) \to L^*_{\text{Bayes}} \quad \text{as} \quad n \to \infty \quad (21)$$

where L^*_{Bayes} denotes the Bayes risk, i.e., the distribution of $(\boldsymbol{X}, Y)$ achievable minimum risk.

Corresponding to Lemma 1 in regression, the classification Lemmas 5 and 6 [37], which consider multiclass classification problem, is as follows.

Lemma 5: If a series of classifiers $\{h\}$ have consistency, the classifier $\overline{h^{(M)}}$, which is defined as taking the majority vote from different h values paired with randomness $\mathcal{C}$, also has consistency.

Lemma 6: Suppose that the maximum posterior estimation for class c is $\gamma^{(c)}(\boldsymbol{x}) = \mathbb{P}(Y = c \,|\, \boldsymbol{X} = \boldsymbol{x})$, which has consistency. Then, the classifier

$$h(\boldsymbol{x}) = \arg\max_c \{\gamma^{(c)}(\boldsymbol{x})\} \quad (22)$$

has consistency.

Lemma 5 shows that to prove the consistency of the proposed RFs, we only need to prove the consistency of the individual trees. From Lemma 6, we know that we only need to prove consistency of the maximum posterior estimation for each class to prove the consistency of multiclass models.

Similar to Lemma 2 in regression, the consistency of the decision tree in classification is also sufficient to show its consistency on data point partitioning, as shown in the following Lemma 7 [37].

Lemma 7: For a distribution of $(\boldsymbol{X}, Y)$, suppose a series of classifiers $\{h\}$ have consistency on the condition I

$$\mathbb{P}(h(\boldsymbol{X}, \mathcal{C}, I) \neq Y \,|\, I) \to L^*_{\text{Bayes}} \quad (23)$$

where I is the randomness when the training data set is partitioned. If acceptable structure and estimation parts are created with probability 1, then $\{h\}$ unconditionally has consistency

$$\mathbb{P}(h(\boldsymbol{X}, \mathcal{C}, I) \neq Y) \to L^*_{\text{Bayes}}. \quad (24)$$

In addition, the general consistency Lemma 8 in classification is almost the same as the regression problem (Lemma 3). The only difference is that the average in regression is replaced by a majority vote in classification [41].

Lemma 8: Consider a partitioning classification rule that builds a prediction by a majority vote method in all the leaf nodes. If the rule for classification is independent of the labels of data for voting, we have

$$\mathbb{E}[L] \to L^*_{\text{Bayes}}, \quad n \to \infty \quad (25)$$

provided that: 1) $n \to \infty$, the diameter of $\mathcal{N}(\boldsymbol{X}) \to 0$ in probability and 2) $n \to \infty$, $N(\mathcal{N}^E(\boldsymbol{X})) \to \infty$ in probability, where $\boldsymbol{X}$ falls into the leaf node $\mathcal{N}(\boldsymbol{X})$ and $N(\mathcal{N}^E(\boldsymbol{X}))$ represents the instance size of the estimation part in the leaf node.

Proof: Refer to [41, Th. 6.1] for more detail. □

Under the above Lemmas 5–8, Theorem 1 can be applied to the classification framework in a straightforward manner, because the conditions for consistency, which require that the hypercube corresponding to the leaf should be sufficiently small, but should contain infinite number of data points, are the same for both regression and classification problems.

VI. Further Comparison

In this section, we compare BRF with three consistent variants of RF, i.e., $Biau08$ [15], $Biau12$ [18], and $Denil14$ [19]. We also include Breiman's original RFs [1], denoted as $Breiman$ in this discussion.

If the tree construction procedure uses labels, it is essential to partition the training data set for the proof of consistency because Lemmas 3 and 8 require the leaf predictors to be fit by the data points, which has no effect on tree structure. Thus, our proposed methods, $Biau12$ and $Denil14$, partition the training data set, while $Biau08$ and $Breiman$ do not.

To ensure consistency according to Lemmas 3 and 8, each feature of the training data set must be selected in a probability as $n \to \infty$ when we choose the candidate features. A single feature and a fixed number of candidate features are randomly selected in $Biau08$ and $Biau12$, respectively. $\min(1 + \text{Poisson}(\lambda), D)$ candidate features are selected in $Denil14$ without replacement. In contract, our proposed BRF

TABLE I
BENCHMARK REGRESSION DATA SETS

DATA SETS	INSTANCES	FEATURES
SLUMP	103	10
SERVO	167	4
AUTOMOBILE	205	26
COMPUTER	209	9
YACHT	308	7
AUTOMPG	398	8
HOUSING	506	14
FORESTFIRES	517	13
STUDENT	649	33
ENERGY	768	8
WIKI	913	53
CONCRETE	1030	9
FLARE	1389	10
AIRFOIL	1503	6
COMMUNITIES	1994	128
SKILLCRAFT	3395	20
WINEQUALITY	4898	12
PARKINSONS	5875	26
INSURANCE	9000	86
AIRQUALITY	9358	15
CBM	11934	16
CASP	45730	9
CT SLICES	53500	386

TABLE II
BENCHMARK CLASSIFICATION DATA SETS

DATA SETS	INSTANCES	FEATURES	CLASSES
ZOO	101	17	7
BREAST	106	10	6
ECHOCARDIOGM	132	12	2
WINE	178	13	3
VERTEBRAL	310	6	3
CVR	435	16	2
BANDS	512	39	2
WDBC	569	32	2
LAND-COVER	675	148	9
CREDIT	690	15	2
TRANSFUSION	748	5	2
VEHICLE	946	18	4
MAMMO	961	6	2
CMC	1473	9	3
CAR	1728	6	4
IMAGE	2310	19	7
MADELON	2600	500	2
CHESS	3196	36	2
ADS	3279	1558	2
ABALONE	4177	8	29
SPAMBASE	4601	57	2
ISOLET	7797	617	26
GISETTE	13500	5000	2
INDOORLOC	21048	529	4
NAMAO	34465	120	2
ADULT	48842	14	2
CONNECT-4	67557	42	3

chooses one or $\sqrt{D}$ based on Bernoulli distribution B_1 without replacement. Last, in the classical RF, $Breiman$, a fixed number of candidate features are randomly selected without replacement.

When we select splitting points, it should be possible to select each candidate splitting point to guarantee consistency according to Lemmas 3 and 8. $Biau08$ randomly selects a point as the splitting point, while $Biau12$ selects the midpoint in each feature to split. $Denil14$ searches the best splitting point in the section ranged by m selected sample points. Our proposed method adopts a hybridized method of selecting the splitting point, based on Bernoulli distribution B_2. The strategy is either to randomly select a point as the splitting point or to search for the best splitting point. Last, $Breiman$ considers all the possible splitting points and selects the best splitting point.

It is clear from the above discussion that our proposed BRF is the closest to Breiman RF. The key difference is in the Bernoulli distributions adopted in our proposed method, which are used when we select features and split points. Another difference is that BRF includes the training data set partitioning procedure. All the strategies adopted by our proposed method are to ensure consistency, while at the same time maintaining sound performance.

VII. EXPERIMENTS

In this section, the performance of BRF is assessed on publicly available data sets [42] in both regression and classification problems.

A. Data Sets

Tables I and II report the 23 UCI data sets for regression and the 27 UCI data sets for classification, respectively, ranked from small to large instance size. The number of features and instances in these benchmark data sets varies. Binary and multiclass data sets are also considered for classification. They are, therefore, sufficiently representative to demonstrate and evaluate how well the proposed BRF behaves.

B. Baselines

The research on *consistency* for Breiman RF is a very challenging issue and has not so far been studied well. Our proposed BRF is compared with the following consistent RFs.

1) Biau08 [15] randomly and uniformly chooses a single feature and splitting point to grow the tree, without data point partitioning.
2) Biau12 [18] chooses a fixed number (e.g., $\sqrt{D}$) of candidate features and their corresponding midpoint as the splitting point, then uses the paired feature and splitting point, which achieves the largest decrease in impurity for growing the tree with data point partitioning.
3) Denil14 [19] chooses $min(1 + Poisson(\lambda), D)$ candidate features, and the best splitting points are optimized in a range that is defined by preselected m points (not the entire number of data points). The tree is also grown with data point partitioning.

TABLE III
MSE ON REGRESSION DATA SETS

DATA SET	*Biau*08[15]	*Biau*12[18]	*Denil*14[19]	BRF
SLUMP	62.30 •	62.31 •	60.19 •	**55.67**
SERVO	3.19 •	2.39 •	2.26 •	**2.07**
AUTOMOBILE	1.53 •	1.51 •	1.46	**1.41**
COMPUTER	57.13 •	56.85 •	56.44 •	**54.14**
YACHT	229.86 •	225.97 •	150.58 •	**128.85**
AUTOMPG	50.94 •	40.78 •	26.63 •	**26.01**
HOUSING	85.50 •	82.97 •	81.62 •	**77.81**
FORESTFIRES	6.32 •	5.39	5.37	**5.36**
STUDENT	9.83 •	9.81 •	9.38 •	**8.93**
ENERGY	64.11 •	40.71 •	24.53 •	**19.85**
WIKI	8.05	8.03	8.02	**8.01**
CONCRETE	279.13 •	279.70 •	278.64 •	**275.56**
FLARE	1.39 •	1.11 •	1.04 •	**0.70**
AIRFOIL	66.66 •	47.73 •	43.47 •	**38.57**
COMMUNITIES	0.07	0.06	**0.05**	**0.05**
SKILLCRAFT	2.68 •	2.57 •	2.41 •	**2.01**
WINEQUALITY	0.81 •	0.81 •	0.67 •	**0.51**
PARKINSONS	68.28 •	66.08 •	64.52 •	**61.61**
INSURANCE	0.09	0.07	0.07	**0.06**
AIRQUALITY	19.10 •	7.47 •	5.88 •	**3.32**
CBM	7.24 •	5.77 •	5.41 •	**4.51**
CASP	30.71 •	14.19 •	9.67 •	**6.24**
CT SLICES	7.13 •	4.88 •	2.71 •	**1.60**

• BRF performs significantly better against consistent random forests at a level of significance 0.05.

TABLE IV
CLASSIFICATION ACCURACY (*ACC*%) ON CLASSIFICATION DATA SETS

DATA SET	*Biau*08 [15]	*Biau*12 [18]	*Denil*14 [19]	BRF
ZOO	50.00 •	41.00 •	80.00 •	**85.00**
BREAST	13.00 •	11.00 •	52.00 •	**66.00**
ECHOCARDIOGRAM	66.15 •	67.69 •	78.46 •	**88.46**
WINE	40.59 •	41.18 •	96.47 •	**97.65**
VERTEBRAL	48.39 •	48.39 •	82.26	**82.58**
CVR	51.86 •	61.40 •	94.42 •	**95.58**
BANDS	57.96 •	57.78 •	68.15 •	**69.44**
WDBC	53.57 •	62.50 •	92.86 •	**95.36**
LAND-COVER	16.12 •	15.37 •	78.06 •	**82.99**
CREDIT	53.73 •	55.37 •	82.09 •	**86.72**
TRANSFUSION	68.92 •	70.27 •	72.97 •	**77.70**
VEHICLE	27.98 •	23.10 •	68.81 •	**71.67**
MAMMOGRAPHIC	54.17 •	53.75 •	79.17 •	**81.25**
CMC	42.72 •	42.65 •	53.60 •	**54.63**
CAR	70.06 •	70.00 •	88.02 •	**93.43**
IMAGE	12.42 •	13.29 •	95.45	**96.06**
MADELON	49.27 •	50.31 •	54.81 •	**69.23**
CHESS	55.64 •	54.95 •	61.32 •	**97.12**
ADS	86.12 •	86.06 •	85.99 •	**94.43**
ABALONE	16.05 •	16.52 •	26.23	**26.44**
SPAMBASE	60.59 •	60.59 •	92.04 •	**94.13**
ISOLET	3.39 •	3.98 •	87.24 •	**88.90**
GISETTE	50.08 •	50.27 •	84.97 •	**94.83**
INDOORLOC	26.61 •	25.12 •	34.39 •	**99.97**
NAMAO	75.85 •	71.44 •	93.33 •	**95.50**
ADULT	50.18 •	50.61 •	53.06 •	**57.57**
CONNECT-4	64.52 •	65.47 •	66.19 •	**76.75**

• BRF performs significantly better against consistent random forests at a level of significance 0.05.

C. Experimental Settings

Our comparisons are as fair as possible, even though each algorithm is parameterized slightly differently. BRF, $Denil14$, and $Breiman$ are parameterized by the number of instances in a leaf. Following [1], we set this number to 5. $Biau08$ and $Biau12$ specify a final leaf number of $n/5$, such that all the trees are constructed the same size. We set the forest size $M = 100$. The parameter ($Ratio$) is set as 0.5 for $Biau12$, $Denil14$, and BRF.

In $Denil14$, m structure points are first chosen. These structure points are used to determine a range from which the splitting point is selected. We set $m = 100$ as suggested in [19]. The probabilities in BRF are set as $p_1 = p_2 = 0.05$ for the Bernoulli distributions. On each data set, we conduct tenfold cross validation with the aim of alleviating the influence of randomness.

D. Learning Performance Analysis

Tables III and IV report the MSE and ACC of different algorithms, respectively. The statistical significance analysis is conducted at the 0.05 significance level, marked by "•." The highest learning performance (i.e., the smallest MSE or the highest ACC) among the consistent RF algorithms is marked in boldface for each data set.

1) Regression: Compared with the other consistent random forest algorithms, BRF achieves the lowest MSE, and the improvement is significant on almost all data sets. Of the four consistent RF algorithms, BRF employs the least simplification of Breiman RF. For example, with regard to the splitting point, $Biau12$ selects a fixed middle point, and $Denil14$ selects an optimized splitting point in a data subset. Both lose a certain amount of information during tree construction, while BRF uses a Bernoulli controlled tree construction, which attempts to use all the information from the entire data. The experimental results show that BRF outperforms $Denil14$, which is in turn better than $Biau12$. Through these comparisons, we can say that BRF's improvement is highly dependent on the two Bernoulli distribution processes for controlling the selection of splitting features and splitting points.

2) Classification: As expected, BRF achieves the highest accuracy of all consistent RF algorithms. For example, BRF achieves a remarkable improvement in accuracy on the INDOORLOC data set, i.e., up to 65.58%, over $Denil14$ which was previously the most consistent RF. The reason for this huge improvement is that some features in the INDOORLOC data set have numerous values. The preselected m data points are likely to cover several of the full feature values, which will influence splitting point selection, and further affect tree structure and performance. Similar to regression, the promotion of BRF is mainly beneficial to Bernoulli controlled tree construction, which maintains good tree structure quality and introduces a degree of useful randomness to guarantee consistency.

TABLE V
MSE DIFFERENCE BETWEEN CONSISTENT RFS (*Denil*14, BRF) AND *Breiman* (NONCONSISTENT)

DATA SETS	$Denil14 - Breiman$	$BRF - Breiman$
FORESTFIRES	0.04	**0.03**
WIKI	1.26	**1.25**
INSURANCE	0.02	**0.01**
AUTOMOBILE	1.18	**1.13**
WINEQUALITY	0.3	**0.14**
SERVO	2.00	**1.81**
FLARE	0.39	**0.05**
SKILLCRAFT	1.57	**1.17**
STUDENT	7.00	**6.55**
AUTOMPG	18.54	**17.92**
CBM	4.72	**3.82**
CT SLICES	1.67	**0.56**
COMPUTER	32.71	**30.41**
AIRQUALITY	5.87	**3.31**
PARKINSONS	51.41	**48.5**
CASP	6.40	**2.97**
HOUSING	71.17	**67.36**
SLUMP	46.13	**41.61**
ENERGY	18.94	**14.26**
AIRFOIL	39.09	**34.19**

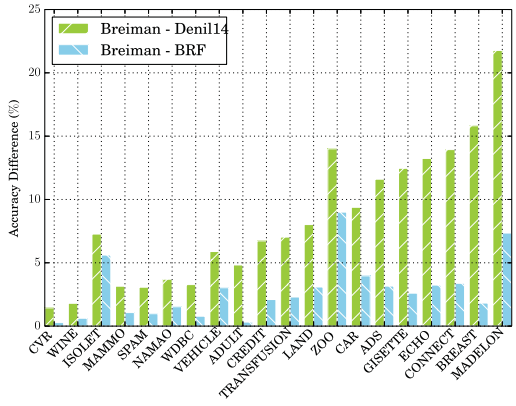


Fig. 3. *ACC* difference between consistent RFs (*Denil*14, BRF) and *Breiman* (nonconsistent).

E. Closing the Gap with Empirical Soundness

All consistent RF variants employ various levels of Breiman RF simplification to guarantee consistency. As a result, the performance of consistent RFs is not as effective as that of Breiman's version.

Table V and Fig. 3 report the gap between theory and practice on 20 regression and 20 classification learning tasks, respectively. The gap between *Denil*14, demonstrated to be an example of the best consistent RFs before [19], and the nonconsistent/empirical *Breiman* is the narrowest. Compared to *Denil*14, the proposed consistent BRF further narrows the gap between theoretical consistency and practical performance. For example, Fig. 3 shows that BRF narrows the gap in classification to around 3%, whereas *Denil*14 only narrows it by around 10%. The narrowing is caused by BRF's controlled tree construction using two Bernoulli distributions, which has a fewer simplifications than *Denil*14. The gap between BRF and *Breiman*, however, still exists, because the training data set partitioning procedure reduces the instance size for constructing the trees in BRF. Similar observations can be found in regression, as shown in Table V.

In summary, BRF is proven in this paper to be consistent. On one hand, BRF is superior to all other consistent variants on empirical performance. On the other hand, it is the closest one to *Breiman* compared to other theoretical versions of RFs.

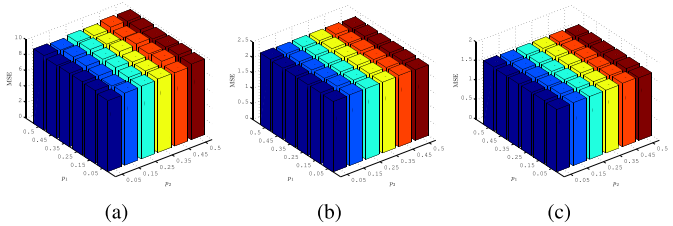


Fig. 4. MSE of BRF with different p_1 and p_2 values ($M = 100$, Ratio $= 0.5$). (a) STUDENT. (b) SKILLCRAFT. (c) CT SLICES.

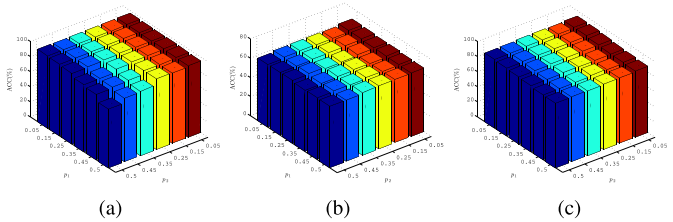


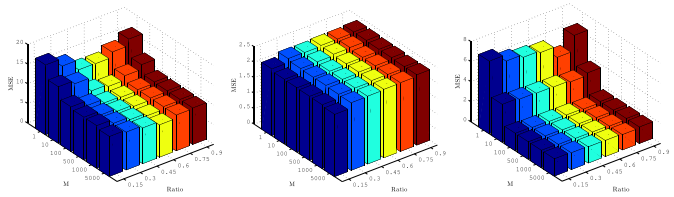
Fig. 5. ACC of BRF with different p_1 and p_2 values ($M = 100$, Ratio $= 0.5$). (a) CHESS. (b) MADELON. (c) ADS.

F. Parameter Analysis

Furthermore, a series of cross-test experiments in the BRF parameters are conducted, i.e., the number of trees M, the (Ratio), and the probabilities p_1, p_2 in the Bernoulli distributions.

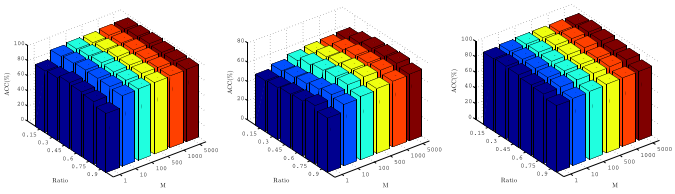
Here, three representative data sets are selected with a small, middle, and large number of instances or features, i.e., STUDENT, SKILLCRAFT, and CT SLICES for regression, and CHESS, MADELON, and ADS for classification. Parameters are tested in the following range: p_1, $p_2 \in \{0.05, 0.15, 0.25, 0.35, 0.45, 0.5\}$, $M \in \{1, 10, 100, 500, 1000, 5000\}$, Ratio $\in \{0.15, 0.3, 0.45, 0.6, 0.75, 0.9\}$.

When p_1 and p_2 have small values, their changes barely influence the MSE or ACC of BRF, i.e., $p_1, p_2 \leq 0.5$ in Fig. 4 or $p_1, p_2 \leq 0.25$ in Fig. 5. The reason is that p_1 and p_2 are two parameters that are set to balance consistency analysis and empirical performance in the tree construction procedure of BRF. When $p_1, p_2 \to 0$, BRF degenerates to *Breiman*, whereas BRF degenerates to *Biau*08 as $p_1, p_2 \to 1$. From this viewpoint, it is reasonable that the values of p_1 and p_2 should be small. The results in Figs. 6 and 7 show that MSE decreases or ACC increases gradually and tends to be constant as M increases. Meanwhile, a large *Ratio* value means that the estimation points are fewer, which leads to imprecise leaf predictors, while a small *Ratio* value means that the structure



(a) (b) (c)

Fig. 6. MSE of BRF with different M and $Ratio$ values ($p_1 = p_2 = 0.05$). (a) STUDENT. (b) SKILLCRAFT. (c) CT SLICES.



(a) (b) (c)

Fig. 7. ACC of BRF with different M and $Ratio$ values ($p_1 = p_2 = 0.05$). (a) CHESS. (b) MADELON. (c) ADS.

points are fewer, which results in a nonoptimal tree structure. To balance the structure and estimation parts, we usually set Ratio $= 0.5$ without favoring either element. There is no need to set the ensemble size very large, because it is necessary to consider the balance between computation complexity and performance gain.

G. Computational Costs

The RF model is an ensemble method whose complexity tends to be the summation of the complexities of constructing individual trees. Typically, the running time of constructing a balanced binary tree is $O(nD\log(n))$ and the query time is $O(\log(n))$. When building RFs, we need to consider two aspects–the number of trees in the forest M, and the size of the feature subspace $mtry$ for building each node of the tree. Thus, the complexity of Breiman RF should be $O(M*mtry*n*\log(n))$. Note that this calculation ignores the complexity involved in the random selection of the feature subspace at each node.

$Biau08$ uses two random processes to select the splitting feature and splitting point, so its complexity is $O(M*\log(n))$. For $Biau12$, it is $O(M*\sqrt{D}*\log(n))$. $Denil14$ is $O(M*\min(1+Poisson(\lambda), D)*n'*\log(n))$, where $n' < n$ is the search space defined by the preselected m points. Last, BRF is $O(M*(p_1*1+(1-p_1)*\sqrt{D})*(1-p_2)*n*\log(n))$, which is comparable to $Denil14$ and lower than $Breiman$.

VIII. Conclusion

In this paper, a novel RF framework named BRFs is proposed, which has nice practical soundness and proven theoretical consistency. We argued that Breiman RF has very good empirical performance because the data-driven tree construction procedure is highly sensitive; however, its theoretical consistency has not been confirmed. Several theoretically guaranteed RF variants are criticized for their inferior empirical performance. While two Bernoulli distributions are employed into the strategies of features and splitting points selection in BRF. Because a probability value-controlled random process is involved in the Bernoulli trial, the tree construction in BRF is random or deterministic with respect to a probability value. A much less data-dependent tree structure is, therefore, obtained by BRF compared with Breiman RF, yet it still achieves a much better performance than RFs with theoretical consistency. Experiments and comparisons show that significantly superior performance is achieved by BRF compared to all existing variants with theoretically guaranteed consistency, and this performance is also the closest one to Breiman RF. BRF takes a big step toward closing the gap between the theoretical consistency and practical performance of RFs.

Appendix A
Proof of Lemma 1

The risk function

$$\begin{aligned} R(\overline{f^{(M)}}) &= \mathbb{E}\left[\left(\frac{1}{M}\sum_{j=1}^{M} f(\boldsymbol{X}, \mathcal{C}^{(j)}, \mathcal{D}_n) - f(\boldsymbol{X})\right)^2\right] \\ &\overset{(a)}{\leq} \frac{1}{M}\sum_{j=1}^{M}\mathbb{E}[(f(\boldsymbol{X}, \mathcal{C}^{(j)}, \mathcal{D}_n) - f(\boldsymbol{X}))^2] \\ &= R(f) \to 0 \end{aligned} \tag{A.1}$$

where (a) is due to $(\sum_{j=1}^{M} a_j)^2 \leq M\sum_{j=1}^{M} a_j^2$ and the triangle inequality. □

Appendix B
Proof of Lemma 2

The risk function

$$\begin{aligned} R(f) &= \mathbb{E}_{\boldsymbol{X},\mathcal{C},I,\mathcal{D}_n}[(f(\boldsymbol{X}, \mathcal{C}, I, \mathcal{D}_n) - f(\boldsymbol{X}))^2] \\ &= \mathbb{E}_I[\mathbb{E}_{\boldsymbol{X},\mathcal{C},\mathcal{D}_n}[(f(\boldsymbol{X}, \mathcal{C}, I, \mathcal{D}_n) - f(\boldsymbol{X}))^2|I]]. \end{aligned} \tag{B.1}$$

Since

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{X},\mathcal{C},\mathcal{D}_n}[(f(\boldsymbol{X}, \mathcal{C}, I, \mathcal{D}_n) - f(\boldsymbol{X}))^2] \\ &\quad\leq \mathbb{E}_{\boldsymbol{X},\mathcal{C},\mathcal{D}_n}[(f(\boldsymbol{X}, \mathcal{C}, I, \mathcal{D}_n))^2] + \mathbb{E}_{\boldsymbol{X}}[f(\boldsymbol{X})]^2 \end{aligned} \tag{B.2}$$

because of the boundedness assumption of f, both terms are finite. Therefore, the dominated convergence theorem can be applied to exchange the expectation and the limit

$$\begin{aligned} &\lim_{n\to\infty} R(f) \\ &\quad= \mathbb{E}_I[\lim_{n\to\infty}\mathbb{E}_{\boldsymbol{X},\mathcal{C},\mathcal{D}_n}[(f(\boldsymbol{X}, \mathcal{C}, I, \mathcal{D}_n) - f(\boldsymbol{X}))^2|I]] \\ &\quad= 0. \end{aligned} \tag{B.3}$$

□

Appendix C
Proof of Lemma 5

Let $h^*(x)$ denote the Bayes classifier, then the lower bound of any classifier is Bayes risk, denoted by

$$L^* = \mathbb{P}(h^*(\boldsymbol{X}) \neq Y \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - \max_c\{\gamma_*^{(c)}(\boldsymbol{x})\}. \tag{C.1}$$

Define G, B as follows:

$$G = \{c \mid \gamma^{(c)}(\boldsymbol{x}) = \max_c\{\gamma^{(c)}(\boldsymbol{x})\}\} \tag{C.2}$$

$$B = \{c \mid \gamma^{(c)}(\boldsymbol{x}) < \max_c\{\gamma^{(c)}(\boldsymbol{x})\}\}. \tag{C.3}$$

Then

$$\begin{aligned}
&\mathbb{P}(\overline{h^{(M)}}(\boldsymbol{X}, \mathcal{C}, \mathcal{D}_n) \neq Y \mid \boldsymbol{X} = \boldsymbol{x}) \\
&= \sum_c \mathbb{P}(\overline{h^{(M)}}(\boldsymbol{X}, \mathcal{C}, \mathcal{D}_n) = c \mid \boldsymbol{X} = \boldsymbol{x}) \mathbb{P}(Y \neq c | \boldsymbol{X} = \boldsymbol{x}) \\
&\leq \left(1 - \max_c \left\{\gamma_*^{(c)}(\boldsymbol{x})\right\}\right) \sum_{c \in G} \mathbb{P}(\overline{h^{(M)}}(\boldsymbol{X}, \mathcal{C}, \mathcal{D}_n) = c \mid \boldsymbol{X} = \boldsymbol{x}) \\
&\quad + \sum_{c \in B} \mathbb{P}(\overline{h^{(M)}}(\boldsymbol{X}, \mathcal{C}, \mathcal{D}_n) = c \mid \boldsymbol{X} = \boldsymbol{x})
\end{aligned} \tag{C.4}$$

it is sufficient to show that $\mathbb{P}(\overline{h^{(M)}}(\boldsymbol{X}, \mathcal{C}, \mathcal{D}_n) = c \mid \boldsymbol{X} = \boldsymbol{x}) \to 0$ for all $c \in B$. For all $c \in B$, we have

$$\begin{aligned}
&\mathbb{P}(\overline{h^{(M)}}(\boldsymbol{X}, \mathcal{C}, \mathcal{D}_n) = c \mid \boldsymbol{X} = \boldsymbol{x}) \\
&= \mathbb{P}\left(\sum_{j=1}^M \mathbb{I}\{h(\boldsymbol{x}, \mathcal{C}^{(j)}) = c\} > \max_{t \neq c} \sum_{j=1}^M \mathbb{I}\{h(\boldsymbol{x}, \mathcal{C}^{(j)}) = t\}\right) \\
&\leq \mathbb{P}(\sum_{j=1}^M \mathbb{I}\{h(\boldsymbol{x}, \mathcal{C}^{(j)}) = c\} \geq 1) \\
&\leq \mathbb{E}\left[\sum_{j=1}^M \mathbb{I}\{h(\boldsymbol{x}, \mathcal{C}^{(j)}) = c\}\right] \\
&= M\mathbb{P}(h(\boldsymbol{x}, \mathcal{C}) = c) \to 0.
\end{aligned} \tag{C.5}$$

□

Appendix D
Proof of Lemma 6

By definition, we need to prove that the rule

$$h^*(\boldsymbol{x}) = \arg\max_c \left\{\gamma_*^{(c)}(\boldsymbol{x})\right\} \tag{D.1}$$

achieves the optimal risk, i.e., the Bayes risk. If $\gamma_*^{(c)}(\boldsymbol{x})$, $c = 1, 2, \ldots, C$, equal, the Bayes risk is obtained, because any choice has the same error probability. In other cases, assuming there is at least one class c that satisfies $\gamma_*^{(c)}(\boldsymbol{x}) < \gamma_*^{(h^*(\boldsymbol{x}))}(\boldsymbol{x})$, we can define

$$\psi^*(\boldsymbol{x}) = \gamma_*^{(h^*(\boldsymbol{x}))}(\boldsymbol{x}) - \max_c \left\{\gamma_*^{(c)}(\boldsymbol{x}) \,\middle|\, \gamma_*^{(c)}(\boldsymbol{x}) < \gamma_*^{(h^*(\boldsymbol{x}))}(\boldsymbol{x})\right\} \tag{D.2}$$

$$\psi(\boldsymbol{x}) = \gamma^{(h^*(\boldsymbol{x}))}(\boldsymbol{x}) - \max_c \left\{\gamma^{(c)}(\boldsymbol{x}) \,\middle|\, \gamma_*^{(c)}(\boldsymbol{x}) < \gamma_*^{(h^*(\boldsymbol{x}))}(\boldsymbol{x})\right\} \tag{D.3}$$

where $\psi^*(\boldsymbol{x}) \geq 0$ measures how much better the best class is than the second, not considering the ties for best. $\psi(\boldsymbol{x})$ is the margin of $h(\boldsymbol{x})$. If $\psi(\boldsymbol{x}) > 0$, $h(\boldsymbol{x})$ has the same probability of making mistakes as the Bayes classifier.

The aforementioned assumption ensures that there is a ϵ satisfying $\psi^*(\boldsymbol{x}) > \epsilon$. Denoting C as the number of classes, if n is large enough, we have

$$\mathbb{P}\left(\left|\gamma^{(c)}(\boldsymbol{X}) - \gamma_*^{(c)}(\boldsymbol{X})\right| < \epsilon/2\right) \geq 1 - \delta/C \tag{D.4}$$

since $\eta^{(c)}$ is a constant. Therefore

$$\begin{aligned}
&\mathbb{P}\left(\bigcap_{c=1}^C |\gamma^{(c)}(\boldsymbol{X}) - \gamma_*^{(c)}(\boldsymbol{X})| < \epsilon/2\right) \\
&\geq 1 - C + \sum_{c=1}^C \mathbb{P}\left(\left|\gamma^{(c)}(\boldsymbol{X}) - \gamma_*^{(c)}(\boldsymbol{X})\right| < \epsilon/2\right) \\
&\geq 1 - \delta \text{ (Bonferroni inequalities [43]).}
\end{aligned} \tag{D.5}$$

The following equation holds with at least $1 - \delta$ probability:

$$\begin{aligned}
\psi(\boldsymbol{X}) &= \gamma^{(h^*(\boldsymbol{X}))}(\boldsymbol{X}) - \max_c \left\{\gamma^{(c)}(\boldsymbol{X}) \middle| \gamma_*^{(c)}(\boldsymbol{X}) < \gamma_*^{(h^*(\boldsymbol{X}))}(\boldsymbol{X})\right\} \\
&\geq \left(\gamma_*^{(h^*(\boldsymbol{X}))} - \epsilon/2\right) \\
&\quad - \max_c \left\{\gamma_*^{(c)}(\boldsymbol{X}) + \epsilon/2 \middle| \gamma_*^{(c)}(\boldsymbol{X}) < \gamma_*^{(h^*(\boldsymbol{X}))}(\boldsymbol{X})\right\} \\
&= \gamma_*^{(h^*(\boldsymbol{X}))}(\boldsymbol{X}) \\
&\quad - \max_c \left\{\gamma_*^{(c)}(\boldsymbol{X}) \middle| \gamma_*^{(c)}(\boldsymbol{X}) < \gamma_*^{(h^*(\boldsymbol{X}))}(\boldsymbol{X})\right\} - \epsilon \\
&= m^*(\boldsymbol{X}) - \epsilon \\
&> 0.
\end{aligned} \tag{D.6}$$

Due to the fact that δ is arbitrary, it is clear that the risk of $h \to L^*_{\text{Bayes}}$ in probability. □

Appendix E
Proof of Lemma 7

Assuming $I \in \mathcal{I}$, and the distribution of I is ν, then

$$\begin{aligned}
&\mathbb{P}(h(\boldsymbol{X}, \mathcal{C}, I) \neq Y) \\
&\quad = \mathbb{E}[\mathbb{P}(h(\boldsymbol{X}, \mathcal{C}, I) \neq Y | I)] \\
&\quad = \int_{\mathcal{I}} \mathbb{P}(h(\boldsymbol{X}, \mathcal{C}, I) \neq Y | I) \nu(I) \\
&\qquad + \int_{\mathcal{I}^c} \mathbb{P}(h(\boldsymbol{X}, \mathcal{C}, I) \neq Y | I) \nu(I).
\end{aligned} \tag{E.1}$$

Due to the assumption that the training data set partitioning generates an acceptable structure part and an estimation part with probability 1, $\nu(\mathcal{I}^c) = 0$.

Since the probability is intrinsically bounded in $[0, 1]$, the dominated convergence theorem can also be applied

$$\begin{aligned}
&\lim_{n \to \infty} \mathbb{P}(h(\boldsymbol{X}, \mathcal{C}, I) \neq Y) \\
&\quad = \lim_{n \to \infty} \int_{\mathcal{I}} \mathbb{P}(h(\boldsymbol{X}, \mathcal{C}, I) \neq Y | I) \nu(I) \\
&\quad = \int_{\mathcal{I}} \lim_{n \to \infty} \mathbb{P}(h(\boldsymbol{X}, \mathcal{C}, I) \neq Y | I) \nu(I) \\
&\quad = L^*_{\text{Bayes}} \int_{\mathcal{I}} \nu(I) \\
&\quad = L^*_{\text{Bayes}}.
\end{aligned} \tag{E.2}$$

□

References

[1] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[2] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: A classification and regression tool for compound classification and qsar modeling," *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 6, pp. 1947–1958, 2003.

[3] A. Acharjee, B. Kloosterman, R. G. Visser, and C. Maliepaard, "Integration of multi-omics data for prediction of phenotypic traits using random forest," *Bioinformatics*, vol. 17, no. 5, p. 363, 2016.

[4] Y. Wang and S.-T. Xia, "A novel feature subspace selection method in random forests for high dimensional data," in *Proc. IJCNN*, 2016, pp. 4383–4389.

[5] A. M. Prasad, L. R. Iverson, and A. Liaw, "Newer classification and regression tree techniques: Bagging and random forests for ecological prediction," *Ecosystems*, vol. 9, no. 2, pp. 181–199, 2006.

[6] D. R. Cutler *et al.*, "Random forests for classification in ecology," *Ecology*, vol. 88, no. 11, pp. 2783–2792, 2007.

[7] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," *Proc. CVPR*, 2011, pp. 1297–1304.

[8] C. Lindner, P. A. Bromiley, M. C. Ionita, and T. F. Cootes, "Robust and accurate shape model matching using random forest regression-voting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1862–1874, Sep. 2015.

[9] C. Xiong, D. Johnson, R. Xu, and J. J. Corso, "Random forests for metric learning with implicit pairwise position dependence," in *Proc. KDD*, 2012, pp. 958–966.

[10] A. Verikas, A. Gelzinis, and M. Bacauskiene, "Mining data with random forests: A survey and results of new tests," *Pattern Recognit.*, vol. 44, no. 2, pp. 330–349, 2011.

[11] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.

[12] Y. Wang and S.-T. Xia, "Unifying attribute splitting criteria of decision trees by tsallis entropy," in *Proc. ICASSP*, 2017, pp. 2507–2511.

[13] Y. Wang, S.-T. Xia, and J. Wu, "A less-greedy two-term tsallis entropy information metric approach for decision tree classification," *Knowl.-Based Syst.*, vol. 120, pp. 34–42, Mar. 2017.

[14] L. Breiman, "Consistency for a simple model of random forests," Statistical Dept., Univ. California, Berkeley, CA, USA, Tech. Rep. 670, 2004.

[15] G. Biau, L. Devroye, and G. Lugosi, "Consistency of random forests and other averaging classifiers," *J. Mach. Learn. Res.*, vol. 9, pp. 2015–2033, Jun. 2008.

[16] R. Genuer. (Jun. 2010). "Risk bounds for purely uniformly random forests." [Online]. Available: https://arxiv.org/abs/1006.2980

[17] R. Genuer, "Variance reduction in purely random forests," *J. Nonparam. Statist.*, vol. 24, no. 3, pp. 543–562, 2012.

[18] G. Biau, "Analysis of a random forests model," *J. Mach. Learn. Res.*, vol. 13, pp. 1063–1095, Apr. 2012.

[19] M. Denil, D. Matheson, and N. de Freitas, "Narrowing the gap: Random forests in theory and in practice," in *Proc. ICML*, 2014, pp. 665–673.

[20] Y. Wang, Q. Tang, S. Xia, J. Wu, and X. Zhu, "Bernoulli random forests: Closing the gap between theoretical consistency and empirical soundness," in *Proc. IJCAI*, 2016, pp. 2167–2173.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[22] S. Bulò and P. Kontschieder, "Neural decision forests for semantic image labelling," in *Proc. CVPR*, Jun. 2014, pp. 81–88.

[23] P. Kontschieder, M. Fiterau, A. Criminisi, and S. Bulò, "Deep neural decision forests," in *Proc. ICCV*, 2015, pp. 1467–1475.

[24] G. Biau, E. Scornet, and J. Welbl. (Apr. 2016). "Neural random forests." [Online]. Available: https://arxiv.org/abs/1604.07143

[25] Z.-H. Zhou and J. Feng. (Feb. 2017). "Deep forest: Towards an alternative to deep neural networks." [Online]. Available: https://arxiv.org/abs/1702.08835

[26] S. W. Kwok and C. Carter, "Multiple decision trees," in *Proc. UAI*, 1990, pp. 327–338.

[27] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol. 9, no. 7, pp. 1545–1588, 1997.

[28] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[29] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, 2000.

[30] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Found. Trends Comput. Graph. Vis.*, vol. 7, nos. 2–3, pp. 81–227, Feb. 2012.

[31] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[32] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[33] Y. Lin and Y. Jeon, "Random forests and adaptive nearest neighbors," *J. Amer. Statist. Assoc.*, vol. 101, no. 474, pp. 578–590, Jun. 2002.

[34] G. Biau and L. Devroye, "On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification," *J. Multivariate Anal.*, vol. 101, no. 10, pp. 2499–2518, 2010.

[35] N. Meinshausen, "Quantile regression forests," *J. Mach. Learn. Res.*, vol. 7, pp. 983–999, Jun. 2006.

[36] H. Ishwaran and U. B. Kogalur, "Consistency of random survival forests," *Statist. Probab. Lett.*, vol. 80, nos. 13–14, pp. 1056–1064, 2010.

[37] M. Denil, D. Matheson, and N. de Freitas, "Consistency of online random forests," in *Proc. ICML*, 2013, pp. 1256–1264.

[38] L. Breiman, "Manual on setting up, using, and understanding random forests V3.1," Statistics Dept. Univ. California, Berkeley, CA, USA, Tech. Rep., 2002.

[39] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[40] L. Györfi, M. Kohler, A. Krzyżak, and H. Walk, *A Distribution-Free Theory of Nonparametric Regression*. Berlin, Germany: Springer, 2002.

[41] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, vol. 31. Berlin, Germany: Springer, 2013.

[42] M. Lichman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[43] J. Galambos, "Bonferroni inequalities," *Ann. Probab.*, vol. 5, no. 4, pp. 577–581, Aug. 1977.

Yisen Wang received the B.S. degree in information engineering from the South China University of Technology, Guangzhou, China, in 2014. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing, China.

His current research interests include machine learning and data mining.

Shu-Tao Xia received the B.S. and Ph.D. degrees in mathematics and applied mathematics from Nankai University, Tianjin, China, in 1992 and 1997, respectively.

In 2004, he joined the Graduate School at Shenzhen, Tsinghua University, Beijing, China, where he is currently a Professor with the Department of Computer Science and Technology. He has authored his main research paper in the IEEE TRANSACTIONS ON INFORMATION THEORY. His current research interests include coding theory, information theory, compressed sensing, and machine learning.

Qingtao Tang received the B.S. degree in statistics from the Renmin University of China, Beijing, China, in 2015. He is currently pursuing the master's degree with the Department of Computer Science and Technology, Tsinghua University, Beijing.

His current research interests include machine learning and data mining.

Jia Wu (M'16) received the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia.

He was with the Center for Artificial Intelligence, University of Technology Sydney. He is currently a Lecturer with the Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, NSW, Australia. He has authored more than 60 refereed journals (such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON CYBERNETICS, and the *ACM Transactions on Knowledge Discovery Data, Pattern Recognition* and conference papers (such as in the International Joint Conference on Artificial Intelligence, the AAAI Conference on Artificial Intelligence, the IEEE International Conference on Data Mining, the IEEE International Conference on Data Engineering, the SIAM International Conference on Data Mining, the ACM Conference on Information and Knowledge Management) in these areas. His current research interests include data mining and machine learning.

Dr. Wu was a recipient of the IJCNN'17 Best Student Paper Award and the ICDM'14 Best Student Paper Candidate Award.

Xingquan Zhu (SM'12) received the Ph.D. degree in computer science from Fudan University, Shanghai, China.

He was with the Center for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Australia. He is currently an Associate Professor with the Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA, and a Distinguished Visiting Professor (Eastern Scholar) with the Shanghai Institutions of Higher Learning, Shanghai, China. He has authored more than 210 refereed journal and conference papers in these areas. His current research interests include data mining, machine learning, and multimedia systems.

Dr. Zhu was a recipient of two best paper awards and one best student paper award. He was an Associate Editor from 2008 to 2012. He has been an Associate Editor since 2014. He is currently serving on the Editorial Board of the *International Journal of Social Network Analysis and Mining* (since 2010) and the *Network Modeling Analysis in Health Informatics and Bioinformatics Journal* (since 2014). He serves as a Program Committee Co-Chair for the 14th IEEE International Conference on Bioinformatics and BioEngineering (2014), the IEEE International Conference on Granular Computing (2013), the 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2011), and the 9th International Conference on Machine Learning and Applications (ICMLA-2010). He also served as a Conference Co-Chair for ICMLA-2012, the Program Area Chair for ICDM (2017, 2013, 2011), BigData-2017, ICTAI (2017, 2015), and CIKM (2013, 2011, 2010), and a Student Travel Award Chair for the 21th ACM SIGKDD-2015.