PNAS RESEARCH ARTICLE | STATISTICS OPEN ACCESS

Provable Boolean interaction recovery from tree ensemble obtained via random forests

Merle Behr[a,1], Yu Wang[a,1], Xiao Li[a], and Bin Yu[a,b,c,2]

Contributed by Bin Yu; received October 10, 2021; accepted March 24, 2022; reviewed by Gérard Biau and Lucas Mentch

Random Forests (RFs) are at the cutting edge of supervised machine learning in terms of prediction performance, especially in genomics. Iterative RFs (iRFs) use a tree ensemble from iteratively modified RFs to obtain predictive and stable nonlinear or Boolean interactions of features. They have shown great promise for Boolean biological interaction discovery that is central to advancing functional genomics and precision medicine. However, theoretical studies into how tree-based methods discover Boolean feature interactions are missing. Inspired by the thresholding behavior in many biological processes, we first introduce a discontinuous nonlinear regression model, called the "Locally Spiky Sparse" (LSS) model. Specifically, the LSS model assumes that the regression function is a linear combination of piecewise constant Boolean interaction terms. Given an RF tree ensemble, we define a quantity called "Depth-Weighted Prevalence" (DWP) for a set of signed features $S^{\pm}$. Intuitively speaking, DWP($S^{\pm}$) measures how frequently features in $S^{\pm}$ appear together in an RF tree ensemble. We prove that, with high probability, DWP($S^{\pm}$) attains a universal upper bound that does not involve any model coefficients, if and only if $S^{\pm}$ corresponds to a union of Boolean interactions under the LSS model. Consequentially, we show that a theoretically tractable version of the iRF procedure, called LSSFind, yields consistent interaction discovery under the LSS model as the sample size goes to infinity. Finally, simulation results show that LSSFind recovers the interactions under the LSS model, even when some assumptions are violated.

decision trees | interaction selection | ensemble methods | consistency | interpretable machine learning

Significance

Random Forests (RFs) are among the most successful machine-learning algorithms in terms of prediction accuracy. In many domain problems, however, the primary goal is not prediction, but to understand the data-generation process—in particular, finding important features and feature interactions. There exists strong empirical evidence that RF-based methods—in particular, iterative RF (iRF)—are very successful in terms of detecting feature interactions. In this work, we propose a biologically motivated, Boolean interaction model. Using this model, we complement the existing empirical evidence with theoretical evidence for the ability of iRF-type methods to select desirable interactions. Our theoretical analysis also yields deeper insights into the general interaction selection mechanism of decision-tree algorithms and the importance of feature subsampling.

Supervised machine learning (ML) algorithms have been proven to be extremely powerful in a wide range of predictive tasks from genomics to cosmology to pharmacology. Understanding how a model makes predictions is of paramount value in science and business alike (1). For example, when a geneticist wants to understand a particular disease—e.g., breast cancer—a black-box algorithm predicting the risk of breast cancer from genotype features is useful, but it does not offer biological insight.

That is, discovery of genes and gene interactions driving a particular disease provides not only understanding as a basic goal in science, but also opens doors for therapeutic treatments. It is a pressing task, in genomics and beyond, to interpret supervised ML models or algorithms and extract mechanistic information in addition to prediction.

Among many supervised ML algorithms, tree ensembles, such as those from Random Forests (RFs) (2) and gradient-boosted decision trees (3), stand out, as they enjoy both state-of-the-art prediction performance in a variety of practical problems and lead to relatively simple interpretations (4–8). To interpret a tree ensemble model, two questions are central:

- **Feature importance**: What features are important for the model's prediction?
- **Interaction importance**: What interactions among features are important for the model's prediction?

While many studies (refs. 4 and 6–8 and the references therein) focus on the RF feature importance, there are relatively few results on the second question. In genetics, Wan et al. (9) and Yoshida and Koike (10) seek (higher-order) gene interactions (or epistasis) by extracting genetic variant interactions from paths of ensembles of fitted decision trees. Wan et al. (9) use MegaSNPHunter based on boosting trees and interpret all groups of features that jointly appear on one of the decision paths as a candidate interaction. Yoshida and Koike (10) propose to rank interactions of genetic variants based on how often they appear together on decision paths in an RF tree ensemble. Recently, iterative RFs (iRFs) (11) were proposed to seek predictive, stable, and high-order nonlinear or Boolean feature interactions. Even though iRF uses the idea that the set of interacting features often appear

Author contributions: M.B., Y.W., X.L., and B.Y. designed research; M.B., Y.W., X.L., and B.Y. performed research; and M.B., Y.W., X.L., and B.Y. wrote the paper.

Reviewers: G.B., Sorbonne Universite; and L.M., University of Pittsburgh

The authors declare no competing interest.

[1]M.B. and Y.W. contributed equally to this work.

[2]To whom correspondence may be addressed. Email: binyu@berkeley.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2118636119/-/DCSupplemental.

Published May 24, 2022.

together on individual decision paths of a tree in an RF ensemble, as in Yoshida and Koike (10), it uses several other ideas. That is, iRF incorporates a soft dimension-reduction step via iterative reweighting of features in terms of their Gini importance, in order to stabilize individual decision paths in the trees. Using the random intersection trees (RIT) (12) algorithm, iRF extracts stable interactions of arbitrary order in a computationally efficient way, even when the number of features is large. There is very positive evidence that iRF extracts predictive, stable, and high-order Boolean interaction information from RF in genomics and other fields (11, 13, 14). While all the works mentioned above provide strong empirical evidence that interactions extracted from the ensemble of decision trees via RF or iRF are informative about underlying biological functional relationships, there are no theoretical results regarding interaction discovery using RF, iRF, or other tree-based methods. In this paper, as a first step toward understanding the interaction-discovery property of tree-based methods, we investigate a key idea in the previous works (9–11)—namely, that frequent joint appearance of features on decision paths in the RF tree ensemble suggests an interaction.

One of the most common assumptions made in previous theoretical analyses of RF is a family of smoothness conditions on the underlying mean regression function, such as the Lipschitz smoothness condition (see, e.g., refs. 15–17). However, many biological processes show thresholding or discontinuous interacting behavior among biomolecules (18, 19), which strongly violates the Lipschitz assumption. It is therefore necessary to introduce a model that can capture the thresholding behavior through discontinuous mean regression function.

The Locally Spiky Sparse Model. Motivated by this thresholding behavior of biomolecules and inspired by RF's predictive performance successes in genomics data problems (20–22), we consider the locally spiky sparse (LSS) model:* an additive regression model where the mean regression function is assumed to be a linear combination of Boolean interaction functions. The linear coefficients, as well as the threshold coefficients of the Boolean functions, are called "model coefficients." Via Boolean functions, the LSS model is able to capture discontinuous thresholding behavior in biology; hence, it can be more relevant for biologists than models with smoothness constraints. We believe the LSS model is suitable and useful as a benchmark model under which to evaluate theoretically (and computationally) interaction-discovery performance of tree-based ML algorithms, including RF.

Our Contributions. Assume that independent and identically distributed (i.i.d.) data samples from the LSS model are given and an RF is fit to these data.

1. For an RF tree ensemble, we first define "signed features." For a decision path of a set of signed features $S^{\pm}$ in the ensemble, we then define a quantity called "depth-weighted prevalence" (DWP). Intuitively speaking, DWP of $S^{\pm}$ measures how frequently the features in $S^{\pm}$ appear together in an RF tree ensemble. We show that DWP has a universal upper bound that depends only on the size of the set of signed features. Moreover, the upper bound is attained with high probability as the sample size increases if and only if the signed features represent a union of interactions in the LSS model. Based on DWP, we show that a simple algorithm—i.e., LSSFind, defined in Algorithm 1—can consistently recover interaction components in the LSS model, regardless of the model coefficients.
2. Our theoretical results imply that feature subsampling of RF is essential to recover interactions by the RF tree ensemble. When too few features are sampled at each node, the tree ensemble is close to extremely randomized trees, and DWP of any set of signed features is independent of the response, which means that it does not contain information on the LLS model; when too many features are sampled, all the trees in the ensemble will be very similar to one another, and that turns out to make it difficult to use tree structures to distinguish between interactions and noninteractions. More specifically, the ratio between the number of subsampled features m_{try} and the total number of features p should be a nonzero constant in order for our algorithm to learn higher-order interactions from tree paths.

Existing Theoretical Works on RF. Existing theoretical studies of RF and its variants belong to two categories. The first focuses on estimating the regression function under Lipschitz or related conditions on the underlying regression function via averaging the decision trees in the RF tree ensemble. The second category studies feature importance measures as an RF output. In contrast, we provide a study on feature interaction selection consistency under an LSS model using DWP extracted from the RF tree ensemble.

In particular, in the first category, Biau (15) considers "median forests" (23), originally considered as a theoretical surrogate by Breiman (24), and obtains the L_2 convergence rate under the Lipschitz continuous models. Scornet et al. (16) give the first consistency result for Breiman's original RF with subsampling instead of bootstrapping in the low-dimensional setting when data are generated via an additive regression model with continuous components. Wager and Athey (17) consider a variant of RF, called honest RF, in the causal inference setup and prove its point-wise consistency and asymptotic normality when the conditional mean function is Lipschitz continuous. Similarly, Mentch and Hooker (25) showed that, under some Lipschitz-type conditions, a moderately large number of trees approximate well the infinite number of trees. Based on these asymptotic normality results, ref. 26 derived hypothesis tests for the null hypothesis that the regression function is additive. Thus, if one defines features interaction as the deviation from a continuous additive regression function, then their results enable testing on a particular candidate. In contrast, in this work, we define feature interaction via the noncontinuous Boolean functions in the LSS model, and we derive consistent interaction selection via the RF tree ensemble, as opposed to a test for an individual interaction, as in ref. 26.

The second category focuses on theory regarding individual feature importance measures. Results in this line of work do not rely on Lipschitz conditions. However, to the best of our knowledge, these works study statistical properties of only noisy features, but do not provide results for signal features in finite samples. Louppe et al. (5) show that Mean Decrease in Impurity (MDI) feature importance for randomized trees has a closed-form formula with an infinite number of samples. Zhou and Hooker (6) use out-of-sample data to improve the MDI feature importance with unbiased theoretical guarantees. Li et al. (8) show that the MDI feature importance of noisy features is inversely proportional to the minimum leaf-node size and suggest a way to improve the MDI using out-of-bag samples. Loecher (7) gives a family of MDI feature importance via out-of-bag samples that are unbiased for the noisy features. Moreover, many studies focus on permutation-based feature importance measures—in particular, Shapley effects (27–33). Among these works, ref. 33 shows some conceptual similarities to the DWP approach considered in this paper, as the authors also consider the concept of joint appearance of features

*The LSS model was first introduced by the authors of ref. 11 (including one of us) and has already been used in simulations to evaluate the performance of iRF/siRF in ref. 13.

on decision paths in the RF tree ensemble. However, instead of using this concept to extract feature interactions, as done in this work, they use it to define an importance sampling scheme to estimate the Shapley effects.

Also related to our work is the recent work ref. 34, which analyzes the extraction of rule sets from an RF tree ensemble. This is very similar to interaction selection, as considered in this work, except that the extracted rules in ref. 34 also include specific estimated thresholds for the individual features. The theoretical analysis in ref. 34 focuses on the stability of the selected rules without specifying a particular data-generating model. In contrast, this paper obtains model-selection-consistency results for LSSFind to estimate signed interactions of signal features under the LSS model.

The rest of the paper is organized as follows: Section 1 introduces the LSS model and Boolean interactions in more detail. Section 2 reviews the RF algorithm and formally defines DWP for a given set of signed features relative to an RF tree ensemble. Section 3 presents our main theoretical results for DWP and introduces LSSFind, a theoretically inspired algorithm to detect interactions from RF tree ensembles via DWP. Section 4 contains simulation results. We conclude with a discussion in Section 5.

1. LSS Model to Describe Boolean Interactions

In this section, we introduce necessary notations and a precise mathematical definition of the LSS model. To this end, for an integer $N \in \mathbb{N}$, let $[N] := \{1, 2, \ldots, N\}$. For a set S of finite elements of $[N]$, let $|S|$ denote its cardinality or the number of elements in S. For any event A, let $\mathbf{1}(A)$ denote the indicator function of A. We assume a given dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ of n samples, with $\mathbf{x}_i = (x_{i1}, \ldots, x_{in}) \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. We say that the data $\mathcal{D}$ are generated from an LSS model when the following assumptions hold true.

LSS Model 1. *Assume $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ are i.i.d. samples from a distribution $P(X, Y)$, such that for some fixed constants $C_\beta > 0$, $C_\gamma \in (0, 0.5)$, the regression function takes the following form*:

$$E(Y|X) = \beta_0 + \sum_{j=1}^{J} \beta_j \prod_{k \in S_j} \mathbf{1}(X_k \gtrless \gamma_k), \qquad [1]$$

where $\gtrless$ in Eq. 1 means either $\leq$ or $\geq$, potentially different for every k. Coefficients β_j are bounded from below, i.e.,

$$\min_{j=1}^{J} |\beta_j| > C_\beta, \qquad [2]$$

and thresholds γ_j are bounded away from 0 and 1, i.e.,

$$\gamma_j \in (C_\gamma, 1 - C_\gamma), \qquad [3]$$

for $j = 1, \ldots, J$. $S_1, \ldots, S_J \subset [p]$ are sets of features called basic interactions. We associate $\leq$ in Eq. 1 with a negative sign (-1) and $\geq$ with a positive sign $(+1)$, such that a signed feature can be written as a tuple $(k, b_k) \in [p] \times \{-1, +1\}$. We call $S_1^{\pm}, \ldots, S_J^{\pm} \subset [p] \times \{-1, +1\}$ basic signed interactions with $S_j^{\pm} = \{(k, b_k) : k \in S_j\}$.

Note that for interactions with only one feature k, due to the sign ambiguity in the LSS model—i.e., $\mathbf{1}(X_k \leq a) = 1 - \mathbf{1}(X_k > a)$—both $\{(k, -1)\}$ and $\{(k, +1)\}$ are counted as an interaction.

The LSS model aims to capture interactive thresholding behavior, which has been observed for various biological processes (18, 35–39). For example, in gene regulatory networks, often a few different expression patterns are possible. Switching between those patterns can be associated with individual components that interact via a threshold effect (36–38). Such a threshold behavior is also observed for other signal-transduction mechanisms in cells—e.g., protein kinase (35) and cell differentiation (18). Another example of a well-studied threshold effect is gene-expression regulation via small RNA (39). Although for most biological processes, the precise functional mechanisms between different features and a response variable of interest are much more complicated than what the LSS model can capture, theoretical investigations of a particular learning algorithm, such as RF, are only feasible within a well-defined and relatively simple mathematical model and useful for practice when such a model is empirically relevant. Given the empirically observed interactive threshold effects in many real biological systems, the LSS model clearly provides a useful enrichment to the current state of theoretical studies of RF and related methods, since current theoretical models do not capture the often-observed interactive threshold behavior.

In order to prove our main Theorem 2, we further impose the following constraints on the LSS model.

Constraint 1 (C1) (Uniformity): X is uniformly distributed on $[0, 1]^p$.

This uniformity assumption implies that each feature is independent of each other. Because any decision tree remains invariant under any strictly monotone transform of an individual feature, the uniform distribution assumption of X can be relaxed to the assumption that individual features X_j, $j \in [p]$ are independent with a distribution that has Lebesgue density. We note that such an independence assumption might be violated in real-world problems. For example, for genetic data with single-nucleotide polymorphisms or gene expression as features X_j, there will typically be a strong correlation between features that are located close by on the chromosome. However, in many cases, it is feasible to restrict to a subset of features (e.g., those that are located sufficiently far apart on the genome) in order to obtain approximate independence. In Section 4, we also demonstrate in simulations that for sufficiently weak feature correlation, one can still obtain accurate interaction selection with LSSFind.

C2 (Bounded-Response): Y is bounded—i.e., $|Y| < 1$.

Note that although we assume $|Y| < 1$, the constant one can be changed to any constant, as we can scale Y by any positive number, and the conclusions in our main results will remain intact. This boundedness condition can be further relaxed so that the residue $Z := Y - E(Y|X)$ is independent of X and 1-subgaussian if we assume a slightly stronger assumption on p and n than the conditions in C4. See *SI Appendix*, Proposition S5 for more detail.

C3 (Nonoverlapping Basic Interactions): $S_1, \ldots, S_J$ do not overlap—i.e., $S_{j_1} \cap S_{j_2} = \emptyset$ for all $j_1 \neq j_2$.

The nonoverlapping assumption that different interactions S_{j_1}, S_{j_2} with $j_1 \neq j_2$ are disjoint might not always be justified in real-world problems. However, it is a crucial assumption for our theorem to hold. The general problem with overlapping interactions in the LSS model is that such models can be nonidentifiable, meaning that different forms of Eq. 1 can imply the same regression function $E(Y|X)$. For example, for the response $\mathbf{1}(X_1 < 0.5, X_2 < 0.5) + \mathbf{1}(X_1 > 0.5, X_2 > 0.5)$, by the definition of signed interactions in the LSS model, it has two basic signed interactions, $\{(1, -1), (2, -1)\}$ and $\{(1, +1), (2, +1)\}$. However, we can also write it as $1 - \mathbf{1}(X_1 < 0.5, X_2 > 0.5) - \mathbf{1}(X_1 > 0.5, X_2 < 0.5)$, which has two different basic interactions, $\{(1, -1), (2, +1)\}$ and $\{(1, +1), (2, -1)\}$. This means that a set of signed features that is an interaction in one of the representations is not an interaction

in the other. Due to this identifiability problem, overlapping features can lead to both false positives and false negatives in terms of interaction recovery with RF. One may try to define interaction more broadly to avoid this identifiability problem. For the previous example $\mathbf{1}(X_1 < 0.5, X_2 < 0.5) + \mathbf{1}(X_1 > 0.5, X_2 > 0.5)$, although the basic signed interactions are not unique, they always constitute both X_1 and X_2. Whether the coefficients $\{\beta_j\}_{j=0}^J$ are allowed to have different signs also affects the identifiability. The previous example is identifiable if we only allow positive coefficients. For domain problems, where interactions are believed to be overlapping, one should investigate different identifiability conditions, but as this depends on the precise application, we leave this for future work. Our work in this paper provides a pathway to investigate this in detail later. We demonstrate how overlapping features affect our results with a simulation study in Section 4.

In Section 3, we show that a simple algorithm, LSSFind, that takes an RF tree ensemble as input can consistently recover basic interactions $S_1, \ldots, S_J$ in the LSS model. Besides recovering $S_j \subset [p]$, LSSFind can also recover the signs of each feature $k \in \cup_{j=1}^J S_j$ in the LSS model, which indicates whether the corresponding threshold behavior in Eq. **1** is given by a $\leq$- or $\geq$-inequality. Without loss of generality, in the rest of the paper, we assume that all inequalities are $\leq$ in Eq. **1**—that is,

$$E(Y|X) = \beta_0 + \sum_{j=1}^{J} \beta_j \prod_{k \in S_j} \mathbf{1}(X_k \leq \gamma_k). \qquad [4]$$

We stress, however, that all our results also hold for the general case Eq. **1**. Because we assume that all the features in basic interactions have minus signs, we denote $S_1^-, \ldots, S_J^- \subset [p] \times \{-1, +1\}$ with $S_j^- = \{(k, -1) : k \in S_j\}$ as basic signed interactions of the LSS model. As our theoretical results will show, the RF tree ensemble can recover not only the basic interactions $S_j \subset [p]$, but also basic signed interactions $S_j^- \subset [p] \times \{-1, +1\}$. In other words, through DWP and under the LSS model, the RF tree ensemble can recover not only which features interact with each other in the LSS model, but also whether a particular feature in an interaction has to be larger or smaller than some threshold for this interaction to be active. Besides basic signed interactions, we also define a "union signed interaction" as a union of individual basic signed interactions, as made more precise in the following definition.

Definition 1 (Union Signed Interactions): In the LSS model with basic signed interactions $S_1^-, \ldots, S_J^- \subset [p] \times \{-1, +1\}$, a (nonempty) set of signed features $S^{\pm} \subset [p] \times \{-1, +1\}$ is called a union signed interaction, if

$$S^{\pm} = \bigcup_{j \in \mathcal{I}} S_j^- \bigcup_{j \in \mathcal{I}_s, k \in S_j, b_k \in \{-1,+1\}} \{(k, b_k)\}, \qquad [5]$$

for some (possibly empty) set of indices $\mathcal{I} \subset \{j \in [J] : |S_j| > 1\}$, $\mathcal{I}_s \subset \{j \in [J] : |S_j| = 1\}$.

In other words, a union signed interaction is a union of one or more basic signed interactions. For a single-feature signed interaction, its sign-flipped counterpart can also be added to the union. For example, for an LSS model with $E(Y|X) = \mathbf{1}(X_1 \leq 0.5) + \mathbf{1}(X_2 < 0.5, X_3 < 0.5)$, there are two basic signed interactions—namely, $\{(1, -1)\}$ and $\{(2, -1), (3, -1)\}$—and five union signed interactions—namely, $\{(1, -1)\}$, $\{(2, -1), (3, -1)\}$, $\{(1, +1)\}$, $\{(1, -1), (2, -1), (3, -1)\}$, and $\{(1, +1), (2, -1), (3, -1)\}$.

The theoretical results that we present in Section 3 are asymptotic, in the sense that they assume the sample size n to go to infinity. Denote the number of signal features $\cup_{j=1}^J S_j$ in the LSS model to be s—i.e., $\sum_{j=1}^J |S_j| = s$. We assume s is uniformly bounded, regardless of n and p. However, the overall number of features p or the number of noisy features $p - s$ can grow to infinity as n increases. Our theoretical results also assume

C4 (Sparsity): $s = O(1)$ and $\frac{\log(p)}{n} \to 0$.

This means that, in contrast to many theoretical works (16, 17, 40),[†] our results hold in a high-dimensional setting, as long as the overall number of signal features s is bounded. The limit $\frac{\log(p)}{n} \to 0$ is a common assumption for high-dimensional settings when analyzing consistency properties of Lasso (see, for instance, refs. 41–43).

2. DWP for an RF Tree Ensemble

In this section, we first review the RF algorithm and then define DWP for a given RF tree ensemble.

A. Review of RF. RF is an ensemble of classification or regression trees, where each tree T defines a mapping from the feature space to the response. Trees are constructed on a bootstrapped or subsampled dataset $\mathcal{D}^{(T)}$ of the original data $\mathcal{D}$. Note that each tree is conditionally independent of one another, given the data. Any node t in a tree T represents a hyper-rectangle R_t in the feature space. A split of the node t is a pair (k_t, γ_t), which divides the hyper-rectangle R_t into two hyper-rectangles $R_{t,l}(k_t, \gamma_t) = R_t \cap \mathbf{1}(X_{k_t} \leq \gamma_t)$ and $R_{t,r}(k_t, \gamma_t) = R_t \cap \mathbf{1}(X_{k_t} > \gamma_t)$, corresponding to the left child t_l and right child t_r of node t, respectively. For a node t in a tree T, $N_n(t) = |\{i \in \mathcal{D}^{(T)} : \mathbf{x}_i \in R_t\}|$ denotes the number of samples falling into R_t.

Each tree T is grown using a recursive procedure (denoted as the CART algorithm (2)), which proceeds in two steps for each node t. First, a subset $M_{\text{try}} \subset [p]$ of features is chosen uniformly at random. The size of M_{try} is m_{try}. Then, the optimal split $k_t \in M_{\text{try}}, \gamma_t \in \mathbb{R}$ is determined by maximizing impurity decrease defined in Eq. **6**:

$$\Delta_I^n(t) := I_n(t) - \frac{N_n(t_l)}{N_n(t)} I_n(t_l) - \frac{N_n(t_r)}{N_n(t)} I_n(t_r), \qquad [6]$$

where t_l (t_r) is the left (right) child of t, and for sample size n, $I_n(t)$ is the impurity measure defined in this paper as

$$I_n(t) = \text{variance of} \{y_i, i \in R_t\},$$

which is the variance of the response y_i's for all the samples in the region R_t. Note that the analysis of this paper holds only for the variance impurity measure, but it is possible to extend to other impurities measures, which is left as future work. The procedure terminates at a node t if two children contain too few samples—e.g., $\min\{N_n(t_l), N_n(t_r)\} \leq 1$—or if all responses are identical—e.g., $I_n(t) = 0$. For any tree T and any leaf node $t_{\text{leaf}} \in T$, denote $\mathfrak{p}(t_{\text{leaf}})$ to be a path to that leaf node.

Definition 2 (Depth of a Path): Given a path $\mathfrak{p}(t_{\text{leaf}})$ that connects root node t_1 and leaf node t_{leaf} in a tree T, we define the depth of the path $\mathfrak{p}(t_{\text{leaf}})$ to be the number of nonroot nodes contained in the path.

For any hyper-rectangle R_t, $\mu(R_t)$ denotes its volume. We make the following assumptions on an RF tree ensemble:

[†]Note that ref. 15 covers the high-dimensional setting, too, but their results only depend on s, and not p.

Assumption 1 (A1) (Increasing Depth of a Tree in the RF Ensemble). *The minimum depth of any path in any tree goes to infinity—i.e.*,

$$\min_{T} \min_{t_{\text{leaf}} \in T} D(t_{\text{leaf}}) \xrightarrow{p} \infty,$$

as $n \to \infty$.

A2 (Balanced Split in a Tree of the RF Ensemble). *Each split* (k_t, γ_t) *is balanced*: *for any node t*,

$$\min\left(\frac{\mu(R_{t,l}(k_t,\gamma_t))}{\mu(R_{t,r}(k_t,\gamma_t))}, \frac{\mu(R_{t,r}(k_t,\gamma_t))}{\mu(R_{t,l}(k_t,\gamma_t))}\right) > \frac{C_\gamma}{1-C_\gamma}.$$

Note that, without loss of generality, we use the same C_γ here as in the LSS model. Otherwise, we can always let C_γ to be the minimum of the two.

A3 (m_{try} Is of Order p). $C_m p + (1-C_m)s \le m_{\text{try}} \le (1-C_m)(p-s)$, *where* $C_m \in (0, 0.5)$ *is a constant.*

A4 (No Bootstrap or Subsampling of Samples). *All the trees in RF are grown on the whole dataset without bootstrapping or subsampling—i.e.*, $\mathcal{D}^{(T)} = \mathcal{D}$ *for any T.*

A4 is a technical assumption that simplifies our notation and analysis. We assume that each tree is grown using all of the samples, which is quite different from the assumptions on subsampling in recent theoretical works on RF (e.g., refs. 15 and 17). The subsampling rate plays a crucial role in the analysis of the asymptotic distribution of the RF predictor (15, 17), where it is assumed that the subsampling rate converges to zero at a desirable rate. However, since we focus on the features selected at each node, and not on the asymptotic distribution of the predictor, we do not require such assumptions on the subsampling rate.

A1 ensures that the length of any decision path in any tree tends to infinity. This assumption is reasonable as tree depths in RF is usually of order $O(\log n)$, which tends to infinity as $n \to \infty$. A2 ensures that each node split is balanced. Similar conditions are used commonly in other papers (17). A3 shows the important role of the parameter m_{try}. Roughly speaking, m_{try} cannot be too small or too big. When m_{try} is too small, there will be too many splits on irrelevant features, which makes the tree noisy. When m_{try} is too big, there will be too little variability in the tree ensemble. This motivation will be made rigorous in the proof of Theorem 2.

B. DWP. In this section, for a tree ensemble from RF, we formally introduce DWP. Given a decision tree T in an RF tree ensemble, we can randomly select a path $\mathcal{P}$ of T as follows: We start at the root node of T and then, at every node, randomly go left or right until we reach a leaf node. This is equivalent to selecting a path in T of depth D with probability 2^{-D} from all the paths in a decision tree. Denote the nodes in $\mathcal{P}$ to be $t_1, \ldots, t_D, t_{\text{leaf}}$. As such, any path $\mathcal{P}$ in a decision tree T can be associated with a sequence of signed features $(k_{t_1}, b_{t_1}), \ldots, (k_{t_D}, b_{t_D}) \in [p] \times \{-1, +1\}$, where D is the depth of the path, and for any inner node $t \in [D]$ on the path, the sign b_t indicates whether the path at node t followed the $\le$ direction ($b_t = -1$) or the $>$ direction ($b_t = +1$) for the split on feature $k_t \in [p]$. For a given RF tree ensemble depending on data $\mathcal{D}$, the randomly selected path $\mathcal{P}$ of tree T, and any fixed constant $\epsilon > 0$, we now define $\hat{\mathcal{F}}_\epsilon(\mathcal{P}, T, \mathcal{D})$ to be the set of signed features on $\mathcal{P}$, where the corresponding node in the RF had an impurity decrease of at least ϵ—that is,

$$\hat{\mathcal{F}}_\epsilon(\mathcal{P}, T, \mathcal{D}) := \{(k_t, b_t) \mid t \text{is an inner node of} \mathcal{P} \text{ with} \Delta_I^n(t) > \epsilon \text{and feature} k_t \text{appears first time on} \mathcal{P}\}. \quad [7]$$

We use $\hat{\mathcal{F}}_\epsilon$ as a shorthand for $\hat{\mathcal{F}}_\epsilon(\mathcal{P}, T, \mathcal{D})$ when the path $\mathcal{P}$ from tree T and the data $\mathcal{D}$ of interest are clear. Note that if a feature appears more than once on the path $\mathcal{P}$, its sign in $\hat{\mathcal{F}}_\epsilon$ is the sign when the feature appears the first time with the impurity decrease above the threshold. Our main theorem will be stated in terms of the DWP of a signed feature set $S^{\pm} \subset [p] \times \{-1, +1\}$ on the random path $\mathcal{P}$ within $\hat{\mathcal{F}}_\epsilon$. To formally define the DWP of $S^{\pm}$, we first need to identify the sources of randomness underlying $\hat{\mathcal{F}}_\epsilon$. There are three layers of randomness involved:

1. ($\mathcal{D}$ **: Data randomness)**: The randomness involved in the data generation;
2. (***T*****: Tree randomness)**: The randomness involved in growing an individual tree with parameter m_{try}, given data $\mathcal{D}$;
3. ($\mathcal{P}$ **: Path randomness)**: The randomness involved in selecting a random path $\mathcal{P}$ of depth d with probability 2^{-d}, given a tree T from an RF tree ensemble with parameter m_{try} based on data $\mathcal{D}$.

In the following definition of the DWP of signed feature sets, the probability is conditioned on data $\mathcal{D}$ and taken only over the randomness of the tree T and the randomness of selecting one of its paths, as in $\mathcal{P}$.

Definition 3 (DWP): Conditioning on data, for any signed feature set $S^{\pm} \subset [p] \times \{-1, +1\}$, we define the DWP of $S^{\pm}$ as the probability that $S^{\pm}$ appears on the random path $\mathcal{P}$ within the set $\hat{\mathcal{F}}_\epsilon$,—that is,

$$\text{DWP}_\epsilon(S^{\pm}) = P_{(\mathcal{P},T)}(S^{\pm} \subset \hat{\mathcal{F}}_\epsilon \mid \mathcal{D}). \quad \textbf{[8]}$$

We emphasize that the probability of selecting a path in a tree T is $P(\mathcal{P}|T) = 2^{-d}$, where d is the depth of the path $\mathcal{P}$.

While we only have a fixed sample size, which means that the data randomness is inevitable, the tree randomness and path randomness are generated by the algorithm and thus can be eliminated by sampling as many trees and paths as we like. Because the DWP in Eq. **8** is only conditioned on data, for any given $\epsilon > 0$ and set of signed features $S^{\pm}$, it can be computed with arbitrary precision from an RF tree ensemble with sufficiently many trees (recall that, conditioned on data $\mathcal{D}$, the different trees in an RF tree ensemble are generated independently).

3. Main Results

In this section, we present our main theoretical results, which are concerned with DWP, as introduced in the previous section. Our results show that LSSFind (Algorithm 1), which is based on DWP at an appropriate level ϵ described in Theorem 3, consistently recovers signed interactions under an LSS model. Before we state our main results in full detail, we want to illustrate it with a simple example.

Algorithm 1: LSSFind (m_{try},ϵ, η, $s_{\max}$)

Input: Dataset $\mathcal{D}$, RF hyperparameter m_{try}, impurity threshold $\epsilon > 0$, prevalence threshold $\eta > 0$, and maximum interaction size $s_{\max} \in \mathbb{N}$.

Output: A collection of sets of signed features. Train an RF using dataset $\mathcal{D}$ with parameter m_{try}; return $\{S^{\pm} \subset [p] \times \{-1, +1\}$ such that $|S^{\pm}| \le s_{\max}$ and $2^{|S^{\pm}|} \cdot \text{DWP}_\epsilon(S^{\pm}) \ge 1 - \eta\}$.

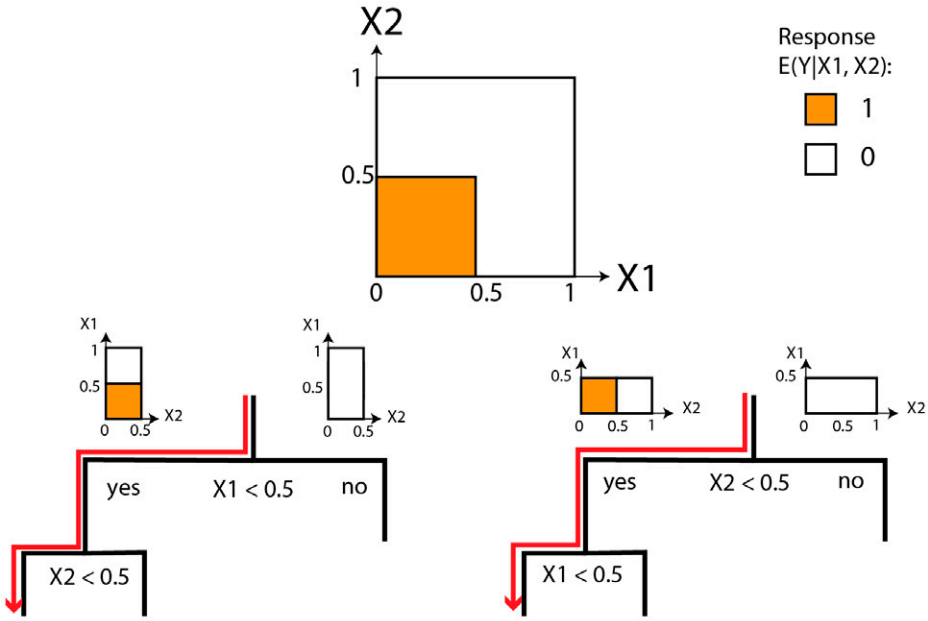

Fig. 1. Exemplary RF decision trees trained on data as in Eq. **9** to illustrate the results that will appear in Theorem 2. (*Upper*) Response surface of $E(Y|X_1, X_2)$, as in Eq. **2**, with $X_1 \in [0, 1]$ on the x axis and $X_2 \in [0, 1]$ on the y axis. (*Lower Left*) A decision tree that splits on feature X_1 at the root node with the respective regions and conditional response surfaces for the left and right child of the root node. (*Lower Right*) A decision tree that splits on feature X_2 at the root node. The red-marked decision paths contain all signed features from the basic signed interaction $S^- = \{(1, -), (2, -)\}$ from an LSS model, as in Eq. **9**. For both of the trees, if one starts at the root node and randomly goes left or right at every node, then the probability of the basic signed interaction to appear on the path is $\mathrm{DWP}_\epsilon(S^-) = 2^{-2} = 2^{-|S^-|}$. In contrast, for any other set of signed features $S^\pm \subset [p] \times \{-1, +1\}$, it holds that $\mathrm{DWP}_\epsilon(S^\pm) < 2^{-|S^\pm|}$. This provides a simple example for the more general result in Theorem 2.

Illustrative Example: Assume that $p = 2$, and there are just two features X_1 and X_2. Assume there is a single interaction $J = 1$, and the regression function is given by

$$E(Y|X_1, X_2) = \mathbf{1}(X_1 \leq 0.5) \cdot \mathbf{1}(X_2 \leq 0.5). \qquad [9]$$

The response surface of Eq. **9** is shown in Fig. 1, *Upper*. We consider the population case, where we have full access to the joint distribution $P(X, Y)$—that is, we have access to an unlimited amount of data ($n = \infty$). When we apply the RF algorithm as in Section 2, then, for each individual tree in the forest, the root node either splits on feature X_1 or on feature X_2. Since X_1 and X_2 are completely symmetric in the distribution $P(X, Y)$, thus, if the RF algorithm grows more and more trees, in the limit, half of them will split on X_1 at the root node and half of them split on X_2 at the root node. For infinite data, this 50/50 split is introduced by the CART algorithm, since the two splits have identical decreases of impurities. Furthermore, the split at any node will be at 0.5 for any of the two features, since the two splits corresponding to $X_1 \leq 0.5$ and $X_2 \leq 0.5$ maximize the impurity decrease given infinite data. This is illustrated in Fig. 1, where *Left Lower* shows a tree that splits on feature X_1 at the root node, and *Right Lower* shows a tree that splits on feature X_2 at the root node. As each tree in RF grows to purity, when the root node splits at feature X_1, then, for the path of the tree that follows the $(1, +1)$ direction—that is, the $X_1 > 0.5$ direction—the tree will stop growing, as the respective response surface is already constant. However, for the path of the tree that follows the $(1, -1)$ direction—that is, the $X_1 \leq 0.5$ direction—the tree will further split on the remaining feature X_2. Then, the tree will stop because the node reaches purity. Thus, we conclude that the forest consists of exactly the two different trees shown in Fig. 1 and in the limit, where the number of trees grows to infinity, each of the two trees appears equally often.

For each node t in these trees, the impurity decrease satisfies $\Delta_I^n(t) \geq 1/16$. Thus, for any $\epsilon < 1/16$, we can show that the DWP of the basic signed interaction $S^- = \{(1, -1), (2, -1)\}$ is $2^{-|S^-|}$. To show this, we can get:

$$\begin{aligned}
&\mathrm{DWP}_\epsilon(S^-) = P(S^- \subset \hat{\mathcal{F}}_\epsilon|\mathcal{D}) \\
&= \underbrace{P_T(T\text{'s root splits on feature1})}_{=0.5,\text{correspond to the left tree}} \\
&\cdot \underbrace{P(S^- \subset \hat{\mathcal{F}}_\epsilon|\mathcal{D},\, T\text{'s root splits on feature1})}_{=0.25,\text{only the red path satisfies this.}} + \\
&\underbrace{P_T(T\text{'s root splits on feature2})}_{=0.5,\text{correspond to the right tree}} \\
&\cdot \underbrace{P(S^- \subset \hat{\mathcal{F}}_\epsilon|\mathcal{D},\, T\text{'s root splits on feature2})}_{=0.25,\text{only the red path satisfies this.}} \\
&= 0.5 \cdot 2^{-2} + 0.5 \cdot 2^{-2} = 2^{-2} = 2^{-|S^-|}.
\end{aligned}$$

In the above example with infinite data, the tree depth is not going to infinity, which means it does not satisfy A1. A1 is needed only for the finite sample case because, for finite samples, internal nodes in a tree can never reach purity due to noise.

In Fig. 1, the paths that contain the basic signed interaction $S^- = \{(1, -1), (2, -1)\}$ are marked red. For all the other sets of signed features $S^\pm \subset [p] \times \{-1, +1\}$, it is easy to check that $\mathrm{DWP}_\epsilon(S^\pm) < 2^{-|S^\pm|}$. For example,

$$\mathrm{DWP}_\epsilon(\{(1, -1), (2, +1)\}) = 0.5 \cdot 2^{-2} + 0.5 \cdot 0 < 2^{-2},$$

and

$$\begin{aligned}
&\mathrm{DWP}_\epsilon(\{(1,-1)\})\\
&= \underbrace{P_T(\,T\text{'s root splits on feature1})}_{=0.5,\text{correspond to the left tree}}\\
&\cdot \underbrace{P(\{(1,-1)\}\subset\hat{\mathcal{F}}_\epsilon|\mathcal{D},\,T\text{'s root splits on feature1})}_{=0.5,\text{any path that goes left at the root satisfies this.}} +\\
&\underbrace{P_T(\,T\text{'s root splits on feature2})}_{=0.5,\text{correspond to the right tree}}\\
&\cdot \underbrace{P(S^-\subset\hat{\mathcal{F}}_\epsilon|\mathcal{D},\,T\text{'s root splits on feature2})}_{=0.25,\text{only the red path satisfies this.}}\\
&= 0.5\cdot 2^{-1}+0.5\cdot 2^{-2}<2^{-1}.
\end{aligned}$$

As we will formally state in the two theorems below, the same reasoning holds true asymptotically for any RF trained on the data from the LSS model—namely, the DWP of a set of signed features $S^\pm\subset[p]\times\{-1,+1\}$ is always upper-bounded by $2^{-|S^\pm|}$, and this upper bound is attained if and only if $S^\pm$ is a union-signed interaction. Recall that the DWP depends on the data $\mathcal{D}$. It turns out that the general upper bound follows directly from the construction of DWP and holds for any data $\mathcal{D}$—i.e., independent of the LSS model—as the following theorem shows.

Theorem 1. *For any impurity threshold $\epsilon>0$ and any set of signed features $S^\pm\subset[p]\times\{-1,+1\}$ for the RF algorithm from Section 2, it holds true that*

- (*General upper bound*) $\mathrm{DWP}_\epsilon(S^\pm)\le 2^{-|S^\pm|}$.

In addition, when the data $\mathcal{D}$ are generated from an LSS model, asymptotically (as the sample size increases), the general upper bound is attained if and only if $S^\pm$ is a union signed interaction, as the following theorem shows.

Theorem 2. *Assume that the data $\mathcal{D}$ are generated from an LSS model with uniformity, bounded-response, nonoverlap basic interactions, and sparsity constraints (see C1–C4). For any impurity threshold $\epsilon>0$, let*

$$b(\epsilon):=\left(4\epsilon/(C_\beta^2\,C_\gamma^{2s-1})\right)^{C_m^{2s}/\log(1/C_\gamma)}, \qquad [10]$$

with constants C_β as in Eq. 2, C_γ as in Eq. 3, s as in C4, and C_m as in A3. Given a set of signed features $S^\pm\subset[p]\times\{-1,+1\}$, for the RF algorithm from Section 2, it holds true that,

- (*Interaction lower bound*) *when $S^\pm$ is a union signed interaction as in Definition 1, we have*

$$\mathrm{DWP}_\epsilon(S^\pm)\ge 2^{-|S^\pm|}-b(\epsilon)-r_n(\mathcal{D},\epsilon);$$

- (*Noninteraction upper bound*) *when $S^\pm$ is not a union signed interaction, then,*

$$\mathrm{DWP}_\epsilon(S^\pm)\le 2^{-|S^\pm|}\left(1-\frac{C_m^s}{2}\right)+r_n(\mathcal{D},\epsilon),$$

with

$$r_n(\mathcal{D},\epsilon)\xrightarrow{p}0 \quad \text{as} n\to\infty,$$

where $\xrightarrow{p}$ denotes convergence in probability.

Proof Sketch: The detailed proof of Theorem 2 is deferred to *SI Appendix*, Section S1. It has two major parts: first, showing the assertion for the idealized population case and, second, extending the population case to the finite sample case.

In the first part, we define a population version of the set $\hat{\mathcal{F}}_\epsilon$, which we denote as $\mathcal{F}$. The set $\mathcal{F}$ only contains desirable features, which are features of a path $\mathcal{P}$ that correspond to a positive decrease in impurity if the RF gets to see the full distribution $P(X,Y)$ (not just a finite sample $\mathcal{D}$). Note that desirable/nondesirable features are different from signal/noisy features. The definition of desirable/nondesirable features depends on the concerned path in a tree. A noisy feature is always a nondesirable feature, but a signal feature can become a nondesirable feature when it has been split in the path. See *SI Appendix*, Definition S1. The set $\mathcal{F}$ is an oracle, in the sense that its construction depends on the true underlying LSS model. This is in contrast to the set $\hat{\mathcal{F}}_\epsilon$, which can be computed for any given path from a tree of RF. Given this definition of $\mathcal{F}$, a sketch of the proof of the major assertions of Theorem 1 and 2 is as follows:

1. When a set of signed features $S^\pm$ appears in $\mathcal{F}$, this implies that every time a signed feature $(k,b)\in S^\pm$ appears on the way from the root node to the leaf, the splitting direction implied by b was selected for $\mathcal{P}$, which gives rise to the general upper bound of $\mathrm{DWP}_\epsilon(S^\pm)\le 2^{-|S^\pm|}$ (Theorem 1).
2. If $S^\pm$ is a union interaction, then (assuming all leaf nodes of the tree are pure) a correct splitting direction for each of its features already implies that $S^\pm$ appears on $\mathcal{P}$ and, thus, $\mathrm{DWP}_\epsilon(S^\pm)\approx 2^{-|S^\pm|}$ (see first part of Theorem 3).
3. If $S^\pm$ is not a union interaction, then there will always be the possibility that, although every split for an encountered feature that is an element of $S^\pm$ was done in the correct direction, some of the features in $S^\pm$ were just never encountered, and, therefore, a correct splitting direction does not imply that $S^\pm$ appears on $\mathcal{P}$; hence, $\mathrm{DWP}_\epsilon(S^\pm)<2^{-|S^\pm|}$ (see second part of Theorem 3).

In the second part of the proof, we show that the observed set $\hat{\mathcal{F}}_\epsilon$ and the oracle set $\mathcal{F}$ are the same, with probability going to one as ϵ goes to zero and n goes to infinity. That would be nice and easy if a tree grown using finite samples will converge to a tree grown using the population in terms of the splitting features and thresholds when sample size tends to infinity. However, that is not true. The obstacle is that, when a node splits on a nondesirable feature, since all the thresholds yield the same impurity decrease in the population case, the threshold selected via finite samples can deviate from the threshold via the population, no matter how many samples are used. Thus, we need to carefully analyze desirable features and nondesirable features separately based on uniform convergence results. □

Remark 1: Theorems 1 and 2 demonstrate that recovery of interactions becomes exponentially more difficult as the size of an interaction increases. An interaction $S^\pm$ corresponds to a region of size $O(2^{-|S^\pm|})$, which means the sample size must be much larger than $2^{|S^\pm|}$ to have enough samples in that region. Also, the DWP at an appropriate level ϵ of a basic interaction $S^\pm$ is $2^{-|S^\pm|}$. To have a consistent estimate, the number of independent paths should be much larger than $2^{|S^\pm|}$. Thus, when one wants to recover an interaction of size s, the number of samples and the number of trees must be much larger than 2^s. That shows the intrinsic difficulty of estimating high-order interactions.

Using the conclusions in Theorem 2, one can show that LSSFind (Algorithm 1) can consistently recover all the basic interactions from the LSS model, as stated in Theorem 3.

Theorem 3. *Let the output of LSSFind (Algorithm 1) be* $\mathscr{S}(m_{\text{try}}, \epsilon, \eta, s_{\max})$. *Under the same settings as in Theorem 2,as long as* $m_{\text{try}}, \epsilon, \eta$ *satisfies the assumptions in Theorem 2 and the following condition*:

$$2^s \cdot b(\epsilon) < \eta < \frac{[C_m]^s}{2}, \qquad [11]$$

with $b(\epsilon)$ *defined in Eq.* **10** *and* C_m *in A3, then, with probability approaching one as* $n \to \infty$, $\mathscr{S}$ *is a superset of the basic signed interactions with size at most* $s_{\max}$ *and a subset of union signed interactions. In particular, if we define*

$$\mathscr{U} = \{S \in \mathscr{S} \mid \text{There is no set} S' \in \mathscr{S} \text{s.t.}, S \subsetneq S'\},$$

then $\mathscr{U}$ *equals the set of basic signed interactions of size at most* $s_{\max}$.

Note that to recover all the basic interactions, $s_{\max}$ needs to be larger than or equal to the order of all the basic signed interactions, but the latter is unknown, as we do not know the underlying LSS model.

Proof: If $S^{\pm}$ is not a union signed interaction, then it follows from the second part of Theorem 2 and $\eta < [C_m]^s/2$ that $2^{|S^{\pm}|} \cdot \text{DWP}_\epsilon(S^{\pm}) < 1 - \eta$, with probability approaching one as $n \to \infty$. Thus, $\mathscr{S}$ is a subset of union signed interactions. If $S^{\pm}$ is a basic signed interaction of size at most $s_{\max}$, then it follows from the first part of Theorem 2 and the fact that $2^s \cdot b(\epsilon) < \eta$ that $2^{|S^{\pm}|} \cdot \text{DWP}_\epsilon(S^{\pm}) \geq 1 - \eta$, with probability approaching one as $n \to \infty$. Thus, $\mathscr{S}$ is a superset of the basic signed interactions with size at most $s_{\max}$. □

Remark 2: One important assumption in our theorem is the sparsity of signal features. If there are many "weak" signal features, it is very hard for RF to work well. For RF, at each node of a tree, only one feature is used. That means the total number of features used along each path is limited by the depth of the tree, which is usually of order $O(\log n)$. For our assertions of Theorem 2, the hard threshold ϵ in the set $\hat{\mathcal{F}}_\epsilon$ has the purpose to select the signal features. Clearly, the choice of an appropriate value of ϵ is hard in practice. The fitting procedure in iRFs (11) (which uses joint prevalence on decision paths in RF to recover interactions, similar as suggested by Theorem 2) filters noisy features not with a hard, but with a soft thresholding procedure: It grows several RFs iteratively and samples features at each node, according to their feature importance from the previous iteration. In that way, one does not need to chose a single hard threshold, which leads to a much more practical algorithm. Unfortunately, such an iterative soft thresholding makes theoretical analysis much harder, which is why we restrict to the hard threshold for the theoretical analysis in this work.

One of the remarkable aspects of the result in Theorem 3 is that the range of η is independent of any model coefficients in the LSS model (that is, the linear β coefficients and the γ thresholds). For sufficiently small ϵ, it only depends on the number of signal features s and the bound of m_{try}—i.e., C_m—and nothing else. In a sense, this shows that the tree ensemble of RF contains the qualitative or discrete-set information of which features interact with each other, independently of the quantitative information about what are the numerical parameters or model coefficients in the LSS model.

Another interesting aspect about the results from Theorem 3 is that they shed some light on the influence of m_{try} on the interaction recovery performance of RF. For the third assertion in Theorem 2, we actually show that

$$\text{DWP}_\epsilon(S^{\pm}) \leq r_n(\mathcal{D}, \epsilon) + 0.5^{|S^{\pm}|}\left(1 - 0.5 \min_{k \in \cup_j S_j} P(\text{root node splits on feature} k)\right).$$

When m_{try} is too large, $\min_{k \in \cup_j S_j} P(\text{root node splits on feature } k)$ can get very small, as particularly strong features (large initial impurity decrease) can mask weaker features. As an extreme example, consider the situation where $m_{try} = p$, and, thus, the root node gets to see all the features. In that case, the single feature that has the highest impurity decrease, say, X_1, will always appear at the root node, and, hence, for $S^{\pm} = \{(1, -1)\}$ or $S^{\pm} = \{(1, +1)\}$, one will get $\text{DWP}_\epsilon(S^{\pm}) = 2^{-|S^{\pm}|} = 0.5$, independent of whether $S^{\pm}$ is an interaction or not. This shows that when m_{try} is too large, DWPs corresponding to false interactions can attain the universal upper bound $2^{-|S^{\pm}|}$, which leads to false positives in terms of interaction recovery. On the other hand, when m_{try} is too small, for a signal feature $k \in \cup_j S_j$, it can take a long time until it gets selected into the candidate feature set at a node. In particular, for a finite sample, it can happen that the tree reaches purity due to lack of samples without having split on any of the signal features. Hence, the reasoning of Theorem 2—namely, that correct split direction + pure path implies that a union interaction appears on the path does not hold anymore. This can lead to union interactions having significantly smaller DWP than the universal upper bound $2^{-|S^{\pm}|}$—i.e., false negatives in terms of interaction recovery.

4. LSSFind and Simulation Results

In this section, motivated by our theoretical results in the previous section, we evaluate LSSFind empirically in terms of its ability to recover interactions.[‡] Simulated experiments are carried out to assess the ability of LSSFind to correctly recover interactions from the LSS model, even when some of the LSS model assumptions are violated.

In LSSFind, one needs to search over all possible sets with size at most $s_{\max}$ to obtain the final result. That is computationally very intensive. One more efficient way is to only look for sets with size at most $s_{\max}$ and also with

$$\text{DWP}_\epsilon(S^{\pm}) \geq (1 - \eta) \cdot 2^{-s_{\max}}, \qquad [12]$$

which implies that we don't need to search over all possible sets with sizes at most $s_{\max}$; instead, we need to search only for sets whose DWP_ϵ's are larger than $(1 - \eta) \cdot 2^{-s_{\max}}$. Because many sets with sizes at most $s_{\max}$ are filtered out, this significantly reduces the search space. We use the FP-growth algorithm (44) to obtain those sets of signed features that have a DWP higher than some threshold. Note that DWP requires an infinite number of trees. To approximate DWP, we use 100 trees in the simulation. Since each tree contains thousands of paths, we have hundreds of thousands of paths to estimate the DWP for.

A. Simulated Data from LSS Models. In the following, we present simulation results, where we generated data $\mathcal{D}$ from the LSS model for different numbers and orders of basic interactions and different signal-to-noise ratios (SNRs). We find that LSSFind recovers the true interactions from the LSS model with high probability

[‡]Source code is available at GitHub (https://github.com/Yu-Group/interaction_selection).

whenever the overall number of basic interactions and their orders are small.

More precisely, we consider $p = 20$ features and $n = 1,000$ samples, where each feature X_j is generated from an uniform distribution $U([0, 1])$, independent from one another. The number of basic interactions is denoted as J, and the order of each interaction is denoted by L. We consider the same threshold τ for all features. The noise is Gaussian with variance σ^2, and the response is:

$$Y = \sum_{j=1}^{J} \prod_{k=(j-1)\cdot L+1}^{k\cdot L} \mathbf{1}(X_k < \tau) + \mathcal{N}(0, \sigma^2). \qquad [13]$$

We consider different values for J, L, and σ^2—namely, $J = 1, 2$, $L = 2, 3, 4$, and σ^2 s such that the SNR is 0.5, 1, 2, or 5. For a given J and L, the threshold τ is chosen such that about 50% of samples fall into the union of hyper-rectangles—that is, $\cup_{j=1}^{J} \cap_{k=(j-1)\cdot L+1}^{j\cdot L} \{X_k < \tau\}$. As we know that the number of samples falling into $\cup_{j=1}^{J} \cap_{k=(j-1)\cdot L+1}^{j\cdot L} \{X_k < \tau\}$, which can also be roughly thought as the label imbalance, has a high impact on the results, keeping this number the same across different simulation settings makes sure that the simulation outcomes are more comparable. The results are averaged across 40 independent Monte Carlo runs. We grow RF using the scikit-learn package with 100 trees. We apply LSSFind with parameters $\eta = 0.01$, $\epsilon = 0.01$, and $s_{\text{max}} = L + 1$. Recall that we use Eq. **12** to select candidate interactions. If s_{max} is set to L, the condition Eq. **12** would be too restrictive for challenging situations, such as when the LSS model is violated, and LSSFind can end up finding no interactions. Given a set $\mathcal{S}^*$ of K true basic signed interactions from the respective LSS model and output from LSSFind $\mathcal{S}$, we evaluate their proximity based on their Jaccard distance:

$$\text{score}(\mathcal{S}^*, \mathcal{S}) = \frac{|\mathcal{S}^* \cap \mathcal{S}|}{|\mathcal{S}^* \cup \mathcal{S}|}. \qquad [14]$$

Note that any element in $\mathcal{S}^*$ and $\mathcal{S}$ is a set of signed features. This score gives no credit for partial recovery: If one interaction $S^{\pm}$ in $\mathcal{S}^*$ is $\{(1, +1), (2, +1)\}$, there will be no credit for $\mathcal{S}$ if it contains subsets of $S^{\pm}$, such as $\{(1, +1)\}$, or same features with different signs, such as $\{(1, +1), (2, -1)\}$. While this score can be overly restrictive for practical problems, it is suitable for our simulation because we would like to evaluate whether LSSFind can consistently recover the interactions in the LSS model. The simulation results are shown in *SI Appendix*, Fig. S1. In general, the performance of LSSFind sharply degrades when the number of basic interactions and the order of interactions increases. For $K = 1$ and $L = 2, 3, 4$, LSSFind almost always recovers the correct basic signed interactions. For $K = L = 2$, it mostly recovers the correct basic signed interactions, except for small SNR. When $K = 2$ and $L = 3, 4$, LSSFind rarely recover the basic signed interactions for this simulation setup, resulting in a score of almost zero. Note that this is consistent with our results in Theorem 2, which indicates that the problem is much harder for more interactions and higher-order interactions. We also explored the high-dimensional case. When $p = 20, 50, 100, 200$ and $n = 1000 \cdot (1 + \log(p/20))$, the score for LSSFind is shown in *SI Appendix*, Fig. S5. The scaling of p and n is chosen to make sure $\log p/n \approx 0.001$, and also when $p = 20$, n will be 1,000, which corresponds to our previous numerical setting for better comparison. Recall that Theorems 2 and 3 require condition $\log(p)/n \to 0$, as stated in condition C4. We also note that $\log(p)/n \to 0$ is commonly imposed when analyzing lasso problems, too (41–43). As can be seen in *SI Appendix*, Fig. S5, the score increases and approaches to one as the dimension p increases. This is consistent with Theorem 3, which shows that LSSFind can recover the underlying interactions for the high-dimensional case.

B. Robustness to LSS Model Violations. In the following, we present simulation results for LSSFind when the data are generated from a misspecified LSS model, which means that some of the LSS model assumptions are violated. We find that LSSFind deteriorate when the LSS model is violated. We consider a misspecified LSS model with $SNR = 5$ and two order-2 interactions with $p = 20$ features and $n = 1,000$ samples, analog as in *SI Appendix*, Fig. S1, second row, first column, third bar. We consider the following violations of LSS model assumptions:

- **Overlapping interactions:** Different basic interactions have overlapping features. When overlap $= 1$, the basic interactions are $((1, -1), (2, -1))$, $((2, -1), (3, -1))$.
- **Correlated features:** Different features are correlated instead of independent. When corr $= \alpha$, the correlation between feature j_1 and j_2 is $\alpha^{|j_1 - j_2|}$.
- **Heavy-tail noise:** Tthe noise follows a Laplace or Cauchy distribution, which have heavier tails than (sub)Gaussian distributions. The noise is normalized such that the SNR is 50.

Results of LSSFind are shown in *SI Appendix*, Fig. S2. For heavy-tail noise, we observe a gradual drop in performance. For the correlated feature case, one can see that LSSFind has reasonable performance when the correlation is close to zero, but its performance deteriorates when the correlation is high. Similarly, for the overlapping feature case, the performance worsens.

C. Empirical Comparison between LSSFind and iRF. Our original motivation to study DWP in RF tree ensemble came from the strong positive empirical evidence of iRF (11, 13). There are three major reasons why the full iRF procedure is hard to analyze theoretically: First, the iterative reweighting in iRF is based on the feature importance metric of MDI. Analyzing MDI for the RF algorithm is a challenging task on its own. In particular, MDI of noisy features in deep trees are known to have a bias (6–8), which may propagate through various iterations in iRF and make a theoretical analysis very challenging. Second, the iRF procedure selects interactions from the paths of the RF tree ensemble via the RIT algorithm (12). Thereby, individual paths are weighted according to the number of observations which fall into their respective leaf nodes. This means that the selected feature interactions of iRF cannot be derived from the RF tree ensemble directly, but depend on the data in a more complex way. Third, the outer stability layer of iRF, where interactions are evaluated based on their consistent appearance among several bootstrap replications of the procedure, adds an additional layer of complexity for theoretical analysis.

In order to still analyze the major aspects of iRF theoretically, we proposed the related LSSFind algorithm. Instead of iterative reweighting via MDI, LSSFind introduces a single hard threshold on the impurity index at individual tree nodes. Moreover, instead of selecting interactions via RIT, LSSFind is based on DWP, which is derived from the tree ensemble directly, without an additional data-dependent sampling scheme. In other words, although a high DWP in LSSFind does not exactly correspond to the RIT interaction selection strategy employed in iRF, they both build on similar high-level quantities—namely, sets of stable features, which often appear together on decision paths in an RF tree ensemble. Therefore, our theoretical results on DWP and LSSFind

provide evidence that the general interaction discovery strategy of iRF is theoretically justified. In the following, we complement our theoretical findings about LSSFind with an empirical comparison between iRF and LSSFind.

We consider the same simulation setting as in Section A. However, we replace the very strict performance measure in Eq. **14** by a weaker one.[§] Specifically, given a set $\mathscr{S}^*$ of K true basic signed interactions from the respective LSS model and output from LSSFind and iRF, respectively, $\mathscr{S}$, we now evaluate their proximity based on:

$$\overline{\text{score}}(\mathscr{S}^*, \mathscr{S}) = \frac{|\{\cup_{S^- \in \mathscr{S}^*} S\} \cap \{\cup_{S^- \in \mathscr{S}} S\}|}{|\{\cup_{S^- \in \mathscr{S}^*} S\} \cup \{\cup_{S^- \in \mathscr{S}} S\}|}. \quad [15]$$

Note that Eq. **15** corresponds to the Jaccard distance on the set of unsigned features that appear in any of the detected interactions. While the stricter metric in Eq. **14** is more appropriate to evaluate finite sample validity of Theorem 2, the relaxed version in Eq. **15** is arguably of more practical interest. This is because it gives partial credit for interactions that are almost, but not perfectly, recovered. If $\overline{\text{score}}(\mathscr{S}^*, \mathscr{S})$ is high, it means that the features in the discovered interactions overlap with the features in the true interactions, which would greatly narrow down the interaction search space and save tremendous effort for subsequent analysis for a practical problem.

For the iRF algorithm, we used the signed iRF algorithm (siRF) from the Python iRF package iRF,[¶] with default parameter settings and a threshold on iRF's stability score of 0.5 for interaction selection, as recommended in ref. 13. Simulation results are shown in *SI Appendix*, Fig. S3. When the LSS model as in Eq. **13** is relatively simple—for example, when it has only a single signed interaction (K = 1) or only a single feature per signed interaction (L =1)—iRF and LSSFind perform comparably (first row and second row, first column, of *SI Appendix*, Fig. S3). However, when the LSS model gets more complex, with several additive interactions ($K > 1$) each having more than one signed features ($L > 1$) (second row, second and third columns in *SI Appendix*, Fig. S3), iRF outperforms LSSFind in terms of the metric Eq. **15**.[‖] In summary, we find that iRF outperforms LSSFind in situations where the underlying LSS model is more complex and when a flexible performance metric is chosen. This appears to be consistent with the fact that the iRF algorithm has witnessed empirical success on specific domain data problems (11), whereas LSSFind was specifically constructed in such a way that it reflects our result in Theorem 2.

5. Discussion

Relevant statistics theory starts with a model that is a good approximation to reality. Thus, it is important to derive theoretical results under a model that is scientifically motivated. Our proposed LSS model class provides such a family that reflects the biological phenomena of biomolecules interacting through thresholding. Also, analyzing RF-based algorithms under different models, rather than the smoothness classes in the literature, can give insights into their empirical adaptivity. Our results give a theoretical result that DWP of a set of features in an RF tree ensemble recovers high-order interactions under the LSS model and reasonable conditions on the RF hyperparameters. Moreover, the universality of interaction's DWP in LSS models gives insights into the general difference between quantitative (e.g., prediction accuracy) and qualitative (e.g., interaction recovery) information extraction. In scientific problems, often the latter is of higher interest. Thus, this work narrows the gap between theory and practice for Boolean interaction discovery and is of general interest to the fields of statistics, data science, ML, and scientific fields, such as genomics.

Our theoretical analysis also gives some insights of RF for tuning a crucial hyperparameter m_{try}: Given an interaction with a fixed size, the noninteraction DWP upper bound in Theorem 2 depends only on C_m, and C_m is only constrained by m_{try} (A3). Therefore, one can find an optimal m_{try} that minimizes this upper bound. The optimal choice of m_{try} turns out to be $m^\star_{\text{try}} = p \cdot (0.5 - s/(2(p-2)))$. If one-third of all features are signal features—that is, $s = p/3$ —$m^\star_{\text{try}}$ recovers the default choice in standard RF implementations for regression—namely, $m^\star_{\text{try}} \approx p/3$. However, when $p \gg s$, the optimal choice from our theoretical results corresponds to $m_{\text{try}} \approx p/2$, which suggests that with the presence of many noisy features, m_{try} should be larger than $p/3$, as in the default choice. Further investigations through data-inspired simulations and theoretical analyses are needed.

One might wonder whether the form of interaction defined by the LSS model constitutes a particularly difficult or a particularly easy form of feature interaction. In general, there appears to be no clear (mathematical) answer to this question, as one cannot define what is meant by feature interaction in a clear way for a generic (possibly discontinuous) regression function $f(X) = E(Y|X)$. For example, it is easy to check that for any multivariate function $f : [0,1]^p -> \mathbb{R}$, one can find (possibly discontinuous) univariate functions $g, h_1, \ldots, h_p$, such that $f(x_1, \ldots, x_p) = g(h_1(x_1) + \ldots + h_p(x_p))$. We stress that the reason why we considered the LSS model in this work was not because it defines a particularly easy form of interaction, but rather its biological relevance, as the thresholding relationships captured in the LSS model are observed in various biological data.

Although, the LSS model is motivated from biological phenomena, some of the assumptions that we made in order to derive our theoretical results might be difficult to justify directly in real-world problems, in particular, the independence condition C1 and the nonoverlapping interaction-set condition C3. Note, however, that in many application settings, it is possible to overcome these limitations by appropriate data preprocessing—e.g., decorrelating features (recall the discussion after condition C1). Nevertheless, for future work, it will be interesting to extend our results to a general LSS model (with possibly overlapping interaction sets and correlated features) or even interaction models beyond Boolean interactions, in order to further close the gap between theory and practice.

Finally, it will also be of interest to compare LSSFind and iRF with methods that, more generally, employ an ML black-box model to extract interactions. For example, when individual features are independent, as we assume in C1, one can use Monte Carlo methods (45) to estimate higher-order Sobol indices for the fitted ML model.

[§] The results of iRF for the set-wise Jaccard distance in Eq. **14** are shown in *SI Appendix*, Fig. S4.

[¶] See GitHub (https://github.com/Yu-Group/iterative-Random-Forest).

[‖] In contrast, for the stricter performance metric Eq. **14**, which precisely captures the interaction detection property of Theorem 2, we note that LSSFind outperforms iRF; *SI Appendix*, Fig. S4.

Data Availability. All source code to reproduce the simulation results and data of this paper is publicly available at GitHub, https://github.com/Yu-Group/interaction_selection. The Python iRF package which was used in the simulations is publicly available at GitHub, https://github.com/Yu-Group/iterative-Random-Forest.

ACKNOWLEDGMENTS. M.B. was supported by Deutsche Forschungsgemeinschaft (German Research Foundation) Post-Doctoral Fellowship BE 6805/1-1. M.B. acknowledges partial support from NSF Grant Big Data 60312. B.Y. acknowledges partial support from NSF Grants NSF-DMS-1613002, 1953191, 2015341, and IIS 1741340. This work was supported in part by the Center for Science of Information, an NSF Science and Technology Center, under Grant Agreement CCF-0939370; and by NSF and the Simons Foundation for the Collaboration on the Theoretical Foundations of Deep Learning through Awards DMS-2031883 and 814639, respectively. This work was supported in part by funding to B.Y. as a Chan Zuckerberg Biohub Investigator. Helpful comments of Sumanta Basu and Karl Kumbier are gratefully acknowledged.

Author affiliations: [a]Department of Statistics, University of California, Berkeley, CA 94720; [b]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720; and [c]Center for Computational Biology, University of California, Berkeley, CA 94720

1. B. Yu, K. Kumbier, Veridical data science. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 3920–3929 (2020).
2. L. Breiman, Random forests. *Mach. Learn.* **45**, 1–33 (2001).
3. J. H. Friedman, Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **29**, 1189–1232 (2001).
4. C. Strobl, A. L. Boulesteix, T. Augustin, Unbiased split selection for classification trees based on the Gini index. *Comput. Stat. Data Anal.* **52**, 483–501 (2007).
5. G. Louppe, L. Wehenkel, A. Sutera, P. Geurts, "Understanding variable importances in forests of randomized trees" in *Advances in Neural Information Processing Systems*, C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger, Eds. (Curran Associates, Inc., Red Hook, NY, 2013), vol. 26, pp. 431–439.
6. Z. Zhou, G. Hooker, Unbiased measurement of feature importance in tree-based methods. *ACM Trans. Knowl. Discov. Data* **15**, 26 (2020).
7. M. Loecher, Unbiased variable importance for random forests. *Commun. Stat. Theory Methods* **51**, 1413–1425 (2020).
8. X. Li, Y. Wang, S. Basu, K. Kumbier, B. Yu, "A debiased MDI feature importance measure for random forests" in *Advances in Neural Information Processing Systems*, H. Wallach *et al.*, Eds. (Curran Associates, Inc., Red Hook, NY, 2019), vol. 32, pp. 8047–8057.
9. X. Wan *et al.*, MegaSNPHunter: A learning approach to detect disease predisposition SNPs and high level interactions in genome wide association study. *BMC Bioinf.* **10**, 13 (2009).
10. M. Yoshida, A. Koike, SNPInterForest: A new method for detecting epistatic interactions. *BMC Bioinf.* **12**, 469 (2011).
11. S. Basu, K. Kumbier, J. B. Brown, B. Yu, Iterative random forests to discover predictive and stable high-order interactions. *Proc. Natl. Acad. Sci. U.S.A.* **115**, 1943–1948 (2018).
12. R. D. Shah, N. Meinshausen, Random intersection trees. *J. Mach. Learn. Res.* **15**, 629–654 (2014).
13. K. Kumbier, S. Basu, J. B. Brown, S. Celniker, B. Yu, Refining interaction search through signed iterative random forests. arXiv [Preprint] (2018). https://arxiv.org/abs/1810.07287 (Accessed 1 September 2019).
14. A. Cliff *et al.*, A high-performance computing implementation of iterative random forest for the creation of predictive expression networks. *Genes (Basel)* **10**, 996 (2019).
15. G. Biau, Analysis of a random forests model. *J. Mach. Learn. Res.* **13**, 1063–1095 (2012).
16. E. Scornet, G. Biau, J. P. Vert, Consistency of random forests. *Ann. Stat.* **43**, 1716–1741 (2015).
17. S. Wager, S. Athey, Estimation and inference of heterogeneous treatment effects using random forests. *J. Am. Stat. Assoc.* **113**, 1228–1242 (2018).
18. L. Wolpert, Positional information and the spatial pattern of cellular differentiation. *J. Theor. Biol.* **25**, 1–47 (1969).
19. M. M. Hoffman *et al.*, Integrative annotation of chromatin elements from ENCODE data. *Nucleic Acids Res.* **41**, 827–841 (2013).
20. P. Jiang *et al.*, MiPred: Classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic Acids Res.* **35**, W339–W344 (2007).
21. X. Chen, H. Ishwaran, Random forests for genomic data analysis. *Genomics* **99**, 323–329 (2012).
22. W. G. Touw *et al.*, Data mining in the Life Sciences with Random Forest: A walk in the park or lost in the jungle? *Brief. Bioinform.* **14**, 315–326 (2013).
23. R. Duroux, E. Scornet, Impact of subsampling and pruning on random forests. arXiv [Preprint] (2016). https://arxiv.org/abs/1603.04261 (Accessed 31 July 2020).
24. L Breiman, "Consistency for a simple model of random forests" (Tech. Rep., Statistical Department, University of California Berkeley, Berkeley, CA, 2004).
25. L. Mentch, G. Hooker, Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *J. Mach. Learn. Res.* **17**, 1–41 (2016).
26. L. Mentch, G. Hooker, Formal hypothesis tests for additive structure in random forests. *J. Comput. Graph. Stat.* **26**, 589–597 (2017).
27. H. Ishwaran, Variable importance in binary regression trees and forests. *Electron. J. Stat.* **1**, 519–537 (2007).
28. C. Strobl, A. L. Boulesteix, T. Kneib, T. Augustin, A. Zeileis, Conditional variable importance for random forests. *BMC Bioinformatics* **9**, 307 (2008).
29. S. Janitza, E. Celik, A. L. Boulesteix, A computationally fast variable importance test for random forests for high-dimensional data. *Adv. Data Anal. Classif.* **12**, 885–915 (2018).
30. S. Nembrini, I. R. König, M. N. Wright, The revival of the Gini importance? *Bioinformatics* **34**, 3711–3718 (2018).
31. S. Nembrini, Bias in the intervention in prediction measure in random forests: Illustrations and recommendations. *Bioinformatics* **35**, 2343–2345 (2019).
32. D. Debeer, C. Strobl, Conditional permutation importance revisited. *BMC Bioinformatics* **21**, 307 (2020).
33. C. Bénard, G. Biau, S. da Veiga, E. Scornet, "SHAFF: Fast and consistent SHApley eFfect estimates via random forests" in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics* (PMLR, 2022), vol. 151, pp. 5563–5582.
34. C. Bénard, G. Biau, S. Da Veiga, E. Scornet, SIRUS: Stable and Interpretable RUle Set for classification. *Electron. J. Stat.* **15**, 427–505 (2021).
35. J. E. Ferrell Jr., Tripping the switch fantastic: How a protein kinase cascade can convert graded inputs into switch-like outputs. *Trends Biochem. Sci.* **21**, 460–466 (1996).
36. J. W. Little, D. P. Shepley, D. W. Wert, Robustness of a gene regulatory circuit. *EMBO J.* **18**, 4299–4307 (1999).
37. O. Kobiler *et al.*, Quantitative kinetic analysis of the bacteriophage lambda genetic network. *Proc. Natl. Acad. Sci. U.S.A.* **102**, 4470–4475 (2005).
38. J. W. Little, Threshold effects in gene regulation: When some is not enough. *Proc. Natl. Acad. Sci. U.S.A.* **102**, 5310–5311 (2005).
39. E. Levine, T. Hwa, Small RNAs establish gene expression thresholds. *Curr. Opin. Microbiol.* **11**, 574–579 (2008).
40. M. Denil, D. Matheson, N. D. Freitas, "Narrowing the gap: Random forests in theory and in practice" in *Proceedings of the 31st International Conference on Machine Learning*, EP Xing, T Jebara, Eds. (Proceedings of Machine Learning Research, PMLR, 2014), vol. 32, pp. 665–673.
41. R. Tibshirani, Regression shrinkage and selection via the lasso. *J. Royal Stat. Soc. Ser. B (Methodological)* **58**, 267–288 (1996).
42. P. Zhao, B. Yu, On model selection consistency of lasso. *J. Mach. Learn. Res.* **7**, 2541–2563 (2006).
43. T. Hastie, R. Tibshirani, M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations* (Monographs on Statistics and Applied Probability, CRC Press, Boca Raton, FL, 2015), vol. 143.
44. J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation. *SIGMOD Rec.* **29**, 1–12 (2000).
45. A. Saltelli, Making best use of model evaluations to compute sensitivity indices. *Comput. Phys. Commun.* **145**, 280–297 (2002).