Supporting information for "Regression forest approaches to gravity wave parameterization for climate projection"

David S. Connelly[1], Edwin P. Gerber[1]

[1]Center for Atmosphere Ocean Science, New York University

Contents of this document

- Introduction
- Text S1 and S2
- Table S3
- Figure S4

Introduction

This document describes technical aspects of the construction of regression trees and forests (Text S1) and of the calculation of an alternate feature importance metric called the Gini importance (Text S2 and Figure S4). These sections are not novel contributions of the work, but may be informative for readers unfamiliar with regression tree modeling.

Also in the supporting information is Table S3, which enumerates the hyperparameters used to train the regression forest emulators discussed in the main text.

Text S1

Regression trees are built recursively, starting with the root node. Each potential combination of input feature and threshold — such combinations will hereafter be referred to as *splits* — is enumerated and scored. The best-scoring split across all features is assigned to the root node, and the training data are partitioned according to that split. Left and right child nodes are added to the root, each of which then repeats the process of selecting its own split using only the subset of the training data that falls to it.

Potential splits are scored using a quantity called *impurity*. The impurity of a subset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$ of input-target pairs in the training dataset is

$$\eta = \frac{1}{M} \sum_{i=1}^{M} \|\boldsymbol{y}_i - \bar{\boldsymbol{y}}\|_2^2 \tag{1}$$

where $\bar{\boldsymbol{y}} = M^{-1} \sum \boldsymbol{y}_i$ is the mean of the target vectors. In the single-output case where the y_i are scalars, η is simply the variance of the targets; in the multioutput case, η is the sum of the component-wise variances. The score of a potential split is the sum of the impurities of the left and right datasets that would result — the lower the better. In other words, better-scoring splits partition the training targets into subsets that are similar to each other, and thus well-approximated by their means.

Nodes are added to the tree in this manner until some stopping condition is met (typically a limit on the depth of the tree) at which point the node becomes a leaf. Instead of evaluating splits and further partitioning the data, the node stores the mean of the remaining training targets to return later. At prediction time, the tree returns the mean of the targets corresponding to training samples that fell to the same leaf node as the given input.

Text S2

The *Gini importance*, a feature importance metric unique to tree-based models, builds on the node impurity η defined in (1). Specifically, let η_n and M_n be the impurity and number of training samples, respectively, at node n. Moreover, let k_n be the feature used in the split at node n, and let ℓ_n and r_n be its left and right children. The Gini importance of feature k for a tree T is

$$g_k(T) \propto \sum_{k_n = k} M_n \left(\eta_n - \frac{M_{\ell_n}}{M_n} \eta_{\ell_n} - \frac{M_{r_n}}{M_n} \eta_{r_n} \right) \tag{2}$$

where the term in parentheses is the reduction in impurity node n achieves by splitting on feature k. The Gini importance is therefore the average impurity gain of all nodes splitting on feature k, weighted by the number of training samples passing through each node. Features have high Gini importance if they are used at many nodes, used at nodes that significantly reduce training impurity, or used at nodes through which many samples pass (e.g., those close to the root). The proportionality sign in (2) indicates that the importances are scaled to sum to unity over all features.

The value $g_k(T)$ defined in (2) is a single scalar describing the importance of feature k to the output of tree T. In this study, though, the targets $\boldsymbol{y}_i$ are vectors. Accordingly, we define a vector-valued impurity analogous to (1) by

$$\boldsymbol{\eta} = \frac{1}{M} \sum_{i=1}^{M} (\boldsymbol{y}_i - \bar{\boldsymbol{y}}_i) \odot (\boldsymbol{y}_i - \bar{\boldsymbol{y}}_i)$$

where $\odot$ indicates component-wise multiplication. We can then use $\boldsymbol{\eta}$ in (2) to obtain a vector-valued Gini importance $\boldsymbol{g}_k(T)$, each component of which gives the weighted average impurity reduction splitting on feature k achieves for the corresponding output component.

The Gini importance can be extended to regression forests by averaging the importances calculated for each constituent tree. However, because trees may make predictions of varying size, and so contribute differently to the forest's output, we define a weight vector $\boldsymbol{w}_T$ for each tree T by

$$\boldsymbol{w}_T = \sum_{i=1}^{M} |T(\boldsymbol{x}_i)|$$

where the absolute value is applied component-wise. Each component of the weight vector $\boldsymbol{w}_T$ is thus the average norm of the of the predictions of T on the training data in the corresponding target component. The Gini importance vector for the forest is then the weighted average

$$\boldsymbol{g}_k = \left(\sum_T \boldsymbol{w}_T\right)^{-1} \odot \left(\sum_T \boldsymbol{w}_T \odot \boldsymbol{g}_T(k)\right)$$

where the reciprocal operation is taken component-wise. This weighting is particularly important for boosted forests, wherein earlier trees make large predictions (and so have a greater influence on the final output), while later trees correspond to minor corrections, as suggested by the schematic in Figure 3b.

Table S3 Hyperparameters used by random and boosted forests in this work.

Value	**Hyperparameter**
300	number of trees in the ensemble
15	maximum depth of each tree
0.15	fraction of training dataset sampled to train each tree
0.5	fraction of features sampled at each node to choose the split
0.1	learning rate multiplying each tree's predictions (boosted forests only)

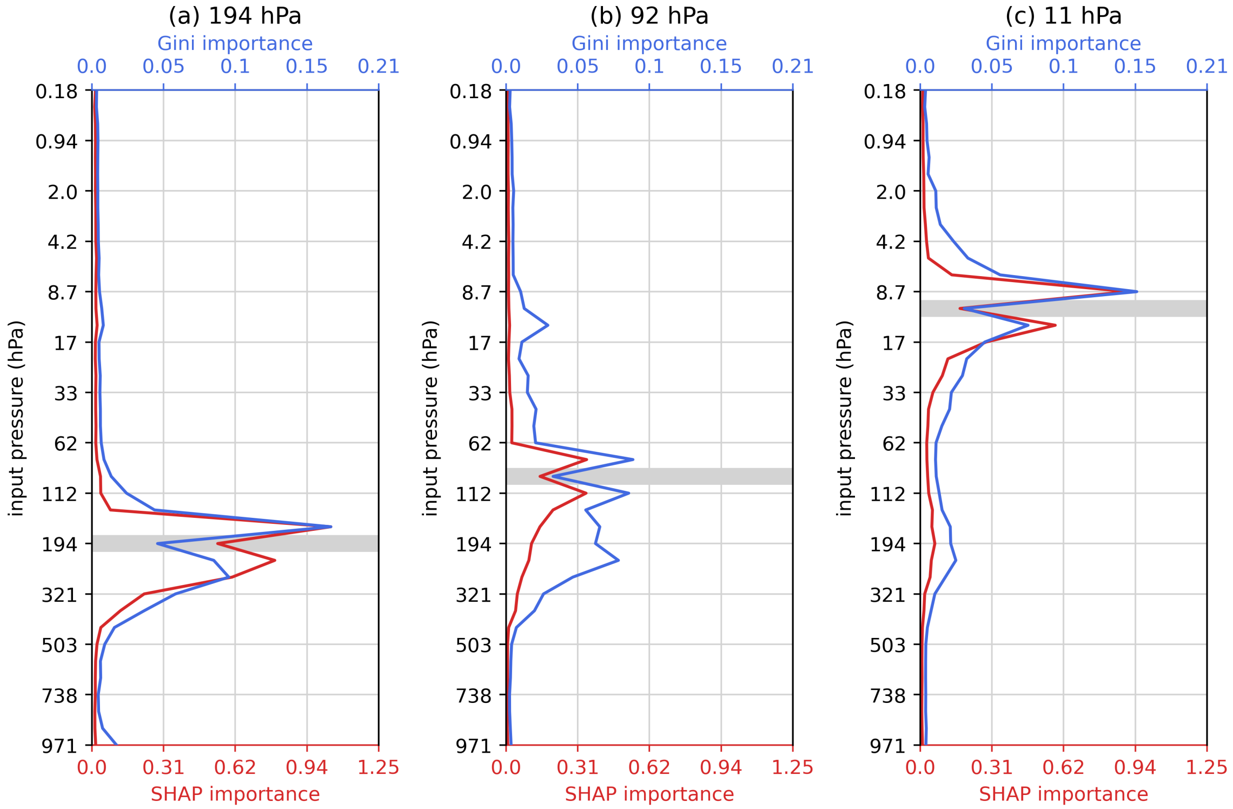

Figure S4 As in Figure 10, but for just the boosted forest with Gini importances in blue and SHAP values in red.