Tree-Based Ensemble Algorithm in SCALATION

Author: Dong Yu Yu
Professor: Dr. John Miller
Date: 12/16/2018

Abstract

Recent Techniques in Data Science have achieved huge success on application in regression and classification tasks using tree base ensemble approaches and artificial neural network. This work performed numerical results to the performance of these tools in SCALATION. The result is shown to improve accuracy of current version of SCALATION in AutoMPG data set, and slight better than the corresponding packages in R.

1. Introduction

In Data Science, techniques can be divided into two major parts: Supervised and Unsupervised Learning. Supervised Learning has known targets while unsupervised learning has no predefined labels. If the target is already known for some dataset used in training, there are quite a few ways to be used by research communities such as applied statistics and neural networks.

In applied statistics, tree based approach like CART and C4.5 are famous choices in both regression and classification problems. While CART and C4.5 shares similarity - both can be effective if the data can be divided into categories, they have a major difference: CART divides tree using Sum of square errors. C4.5 uses information gain.

Ensemble techniques can be used to largely improve the performance of the tree based approach. Two ways to do are Random Forest and Gradient Boosting. Random Forest gathers trees by randomly picking subsamples and showing some portion of features to increase variance within each trees, which can be done in parallel and increase efficiency by nature. On the other hand, Gradient Boosting includes one tree at a time by using derivative, making the tree tuned to fix the error the model made.

Recurrent Neural Network is another approach used for supervised tasks, note that it also has the capacity to deal with unsupervised tasks. Recently, there are considerable activities in artificial neural network. In this approach, a group of units form a layer, interconnecting to each other in capturing input to output. Recurrent Neural Network introduces the time stamp, allowing parameters of current to be the input for the future of the hidden layer for time serial problems.

In this work, instead of doing general overview on all the approaches discussed here, we focus on experiments and give numerical results in the particular tasks in Data Science using SCALATION, including plenty of dataset: AutoMPG, Wine quality classification, and Natural Language Processing. We found that each technique has advantage in different problem set.

2. Our Approach

For every technique we used in this section, we all compared with the package in R if it's available. The SCALATION version was also described in the sub sections.

2.1 Random Forest

Random Forest generates trees and gets the classification result by letting the trees vote. It has been proven to have ideal properties, like its simplicity to understand and implement, with the capacity to handle non-linearity, friendly to parallel training and large data set, and efficiency. We had our version using C4.5 tree as bases.

SCALATION version
param x the features part of samples
param y the class labels of samples
param nF the number of Trees
param bR bargain ratio (the portion of samples used in building trees)
param fS the number of features used in building trees
param nC the number of classes in samples

C4.5
C4.5 tree is the most popular induction tree algorithm and the later variants for ID3. It extends to capacity for continuous and discrete features, and can be used in regression and classification problems. C4.5, on contrary to CART, uses information gain to generate the tree.

2.2 Gradient Boosting

Gradient Boosting is an approach by gradient descent in function space in contrast to tradition way, which is used in parameters. After calculating the loss, one tree is included to reduces the loss. We used Regression Tree as bases.

SCALATION version
param x the data vecotrs stored as rows of a matrix
param y the depedent value
param n_iteration the iterations for training
param depth the max_depth for the base (regression tree)

Regression Tree (used in CART)
Regression Tree can be viewed as diving dataset into disjoint region according to the features. One common choice will be to use binary tree which chooses threshold to split the dataset to minimize total sum of square error. Here we want to find constant predictor C for every region defined by the tree. If region is fixed, it's easy to prove that the mean in the region will be the best constant we used by taking the derivative and set it to zero.

SCALATION version
param x the data vectors stored as rows of a matrix
param y the dependent value
param fn the names for all features/variables
param maxDepth the depth limit for tree
param curDepth current depth
param branchValue parameter used to record the branchValue for the tree node
param thres parameter used to record the threshold for the tree 's parent node
param feature parameter used to record the feature for the tree 's parent node

Fast Algorithm
A more efficient algorithm would be to calculate the sum of SSEs iteratively as gradually include data points when considering next possible threshold, ie, to maintain a running SSEs. For example, {1, 10, 11, 12} can be considered to have 5.5, 10.5, 11.5 as thresholds. When traversing from 5.5, 10.5, to 11.5, SSEs for 5.5 is first calculated. At the time we jump from 5.5 to 10.5, the points in [5.5, 10.5) are included form left(x) and deducted from right(x). Similarly from 10.5 to 11.5. In this way, the corresponding SSEs for every threshold can be obtained in linear time as traversing possible thresholds.

2.3 Recurrent Neural Network

Recurrent Neural Network consists of RNN layer which has three components: gate, activation, output. The derivative is computed with the components using backward propagation.

SCALATION Version
param data_dim the dimension of the data space
param hidden_dim the dimension of the hidden layer
param bptt_truncate truncate bptt, clip to constrain the dependency to avoid gradient vanish/explode

3. Experiment

3.1 Random Forest

Data set
We used famous WineQuality dataset, which has 4898 data with 11 features and 7 labels defining the level of the wine.

To exam the effects for randomness, we used 80% of dataset using for training, number of trees = 5, bagging ratio = 2/3, features for splitting = 7. We generated 11 RF with seed from 0 to 10, and confident interval as following.

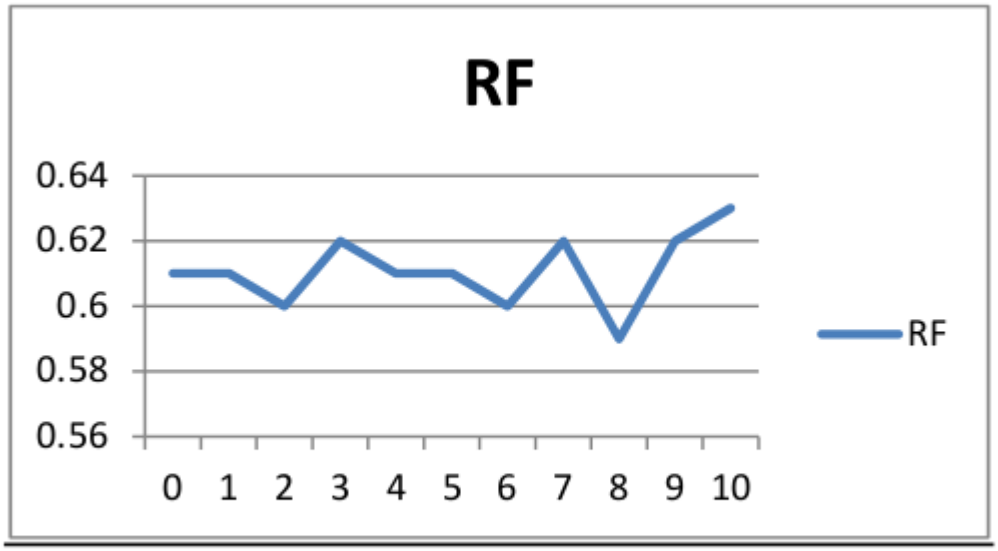

Fig.1

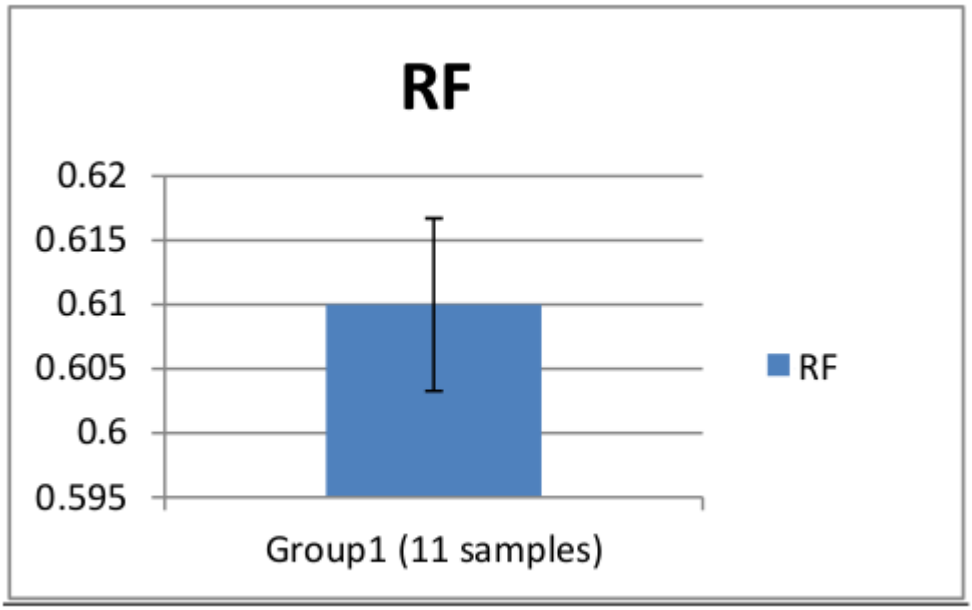

Fig.2

As Shown in Fig.1 and Fig.2, in 11 samples, the value of SCALATION version:
mean: 0.61, sample std=0.01, standard error of the mean(sem)= std/(square root of n)=0.0034
Build the 95% confident interval (Assuming Normal Distribution)
Upper limit = mean + 1.96*sem=0.6167
Lower limit = mean – 1.96*sem=0.6033
So SCALATION version is within 0.6033~0.6167 with 95% confidence.

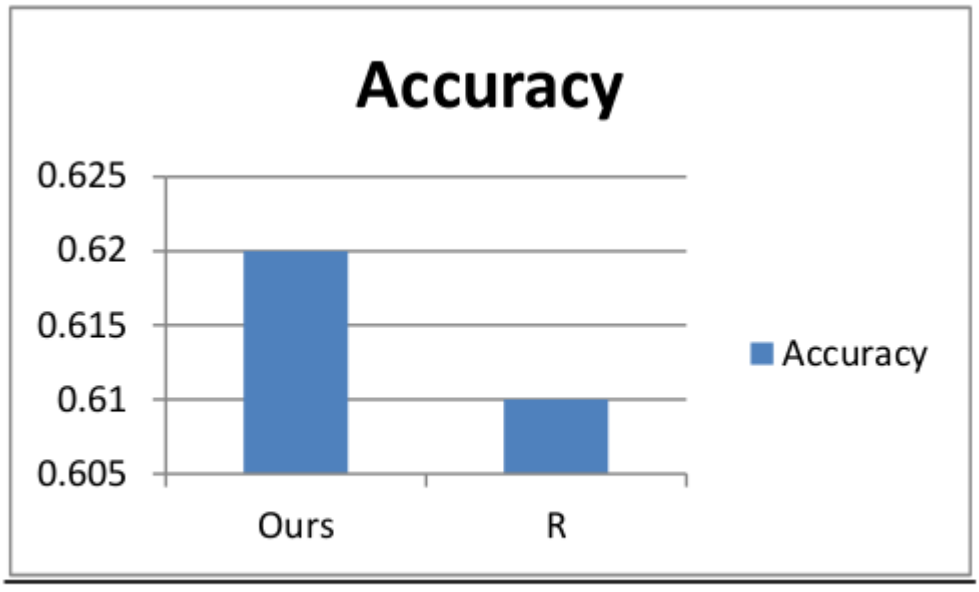

Fig.3

As Fig.3 shown, with specific seed(seed = 3), SCALATION version is slightly better than R.

3.2 Gradient Boosting

Data Set

We used AutoMPG dataset, which has 392 rows and with 8 features. We used the 8th column as the target and used the previous 7 features to predict the 8th.

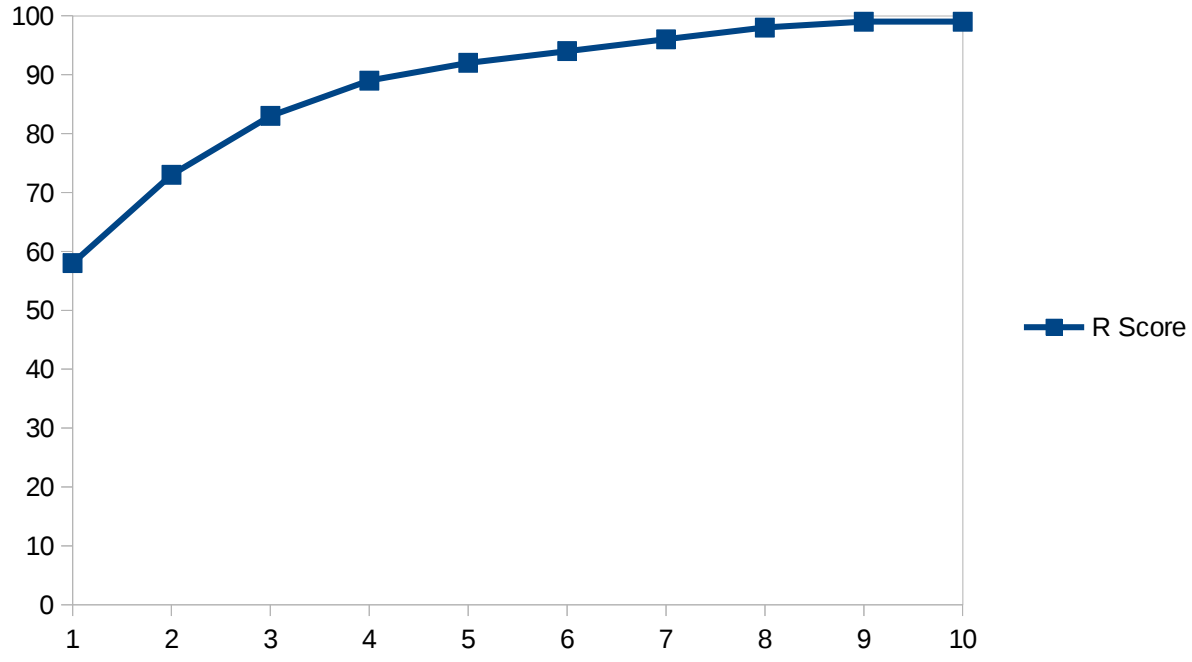

Fig. 4

As shown in Fig.4, the accuracy(R Score) increased with depth range from 1 to 10.

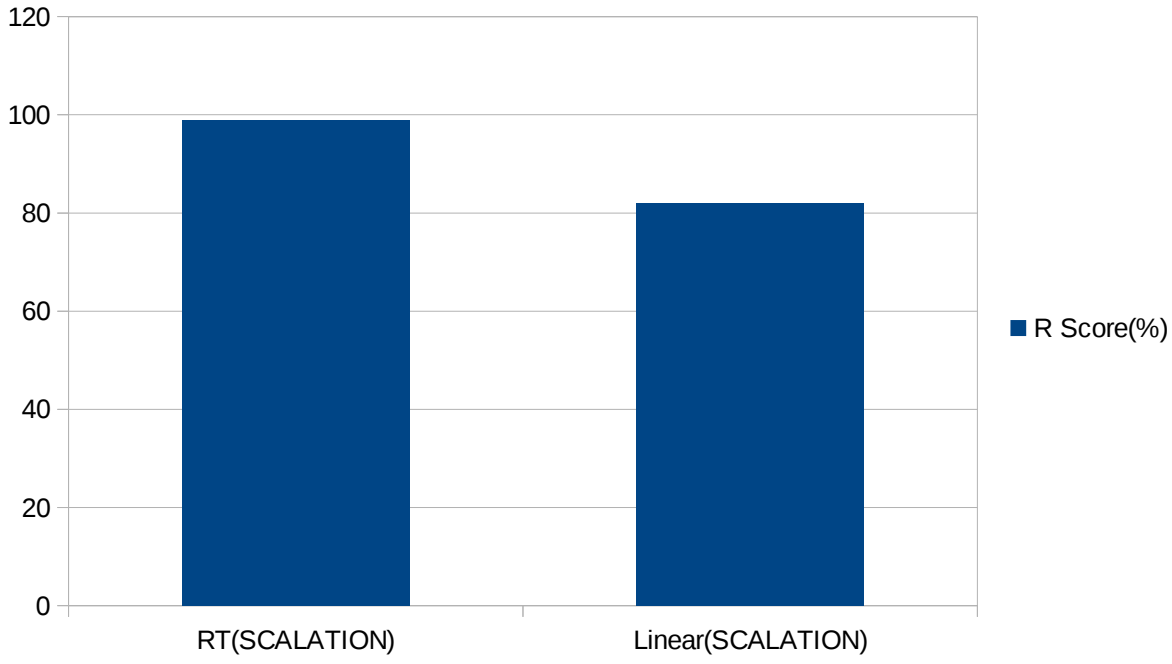

Fig.5

As shown in Fig.5, we compared the current models used in SCALATION, linear model to show the improvement for current capacity of the package (99% vs 82%).

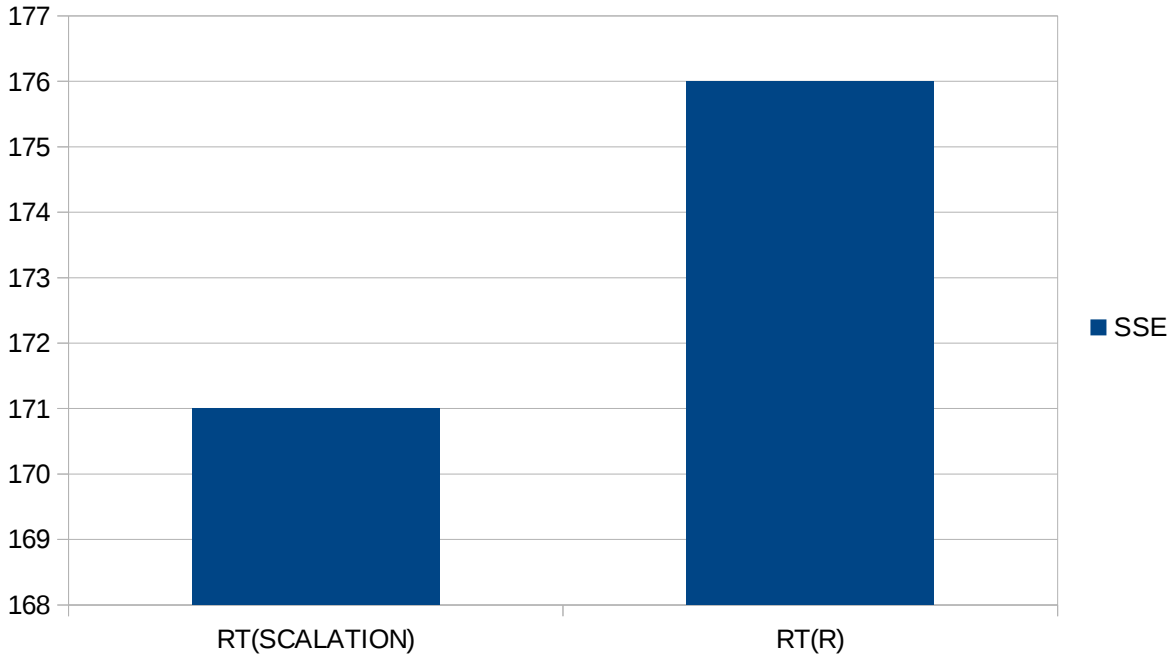

Fig. 6

The Fig. 6 showed the comparison between SCALATION and R, and our version had less SSE than R (171 vs 176).

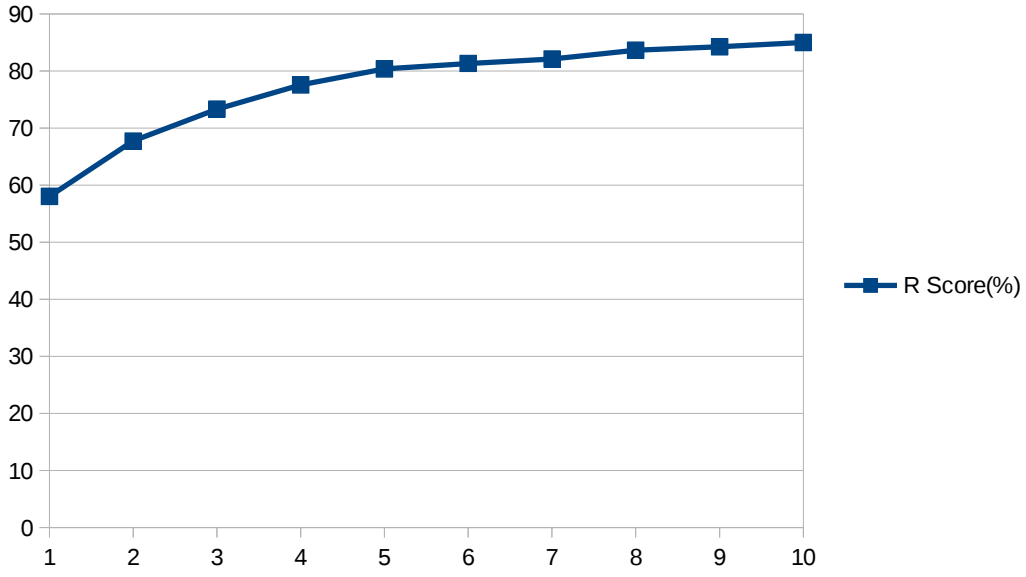

Fig. 7

Fig. 7 showed the improvement by inclusion of Regression Tree using Gradient Boosting. Note that we used weaker estimator (Tree Stump, Regression Tree with depth 1).

3.3 Recurrent Neural Network

Data Set
We used comments in reddit. After tokenizing, we had 80,000 examples. We used 8,000 most common words in the vocabulary. The hidden level dimension was 100. Due to the complexity, for every epoch, we trained RNN using 10 examples.

Epoch	Loss
1	8.987025183337305
2	8.987024838798146
3	8.987024468009622
4	8.987024061237044
5	8.987023615963706
6	8.98702312981069
7	8.987022600648743
8	8.987022026726518
9	8.987021406814893
10	8.987020740361299

Fig. 8

As Fig. 8 shown, every epoch decreased the loss function (use entropy).

4. Conclusions and Future Works

Our work over viewed recent techniques in Data Science and was proved by numerical results in contribution in for SCALATION on applications in regression and classification tasks. Regression Tree improves the current linear model in dealing with AutoMPG data set. Random Forest and Gradient Boosting are proven to have slight better accuracy on training sets in Wine Quality and AutoMPG data set than corresponding packages in R. Recurrent Neural Network was proved to function correctly in Natural Language Processing in complicate data set.

Gradient Boosting can be further extend to fit in classification tasks by choice of loss function, like logistic loss. Utilizing power like GPU and graph optimization can increase the efficiency of Recurrent Neural Network. Those are certainly an interesting direction for further study.

Reference

Akinsola, J E T. “Supervised Machine Learning Algorithms: Classification and Comparison.” *International Journal of Computer Trends and Technology (IJCTT)* 48 (June 8, 2017): 128–38. https://doi.org/10.14445/22312803/IJCTT-V48P126.

Breiman, Leo. “Arcing the Edge,” 1997.

Chen, Tianqi, and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System.” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*, 2016, 785–94. https://doi.org/10.1145/2939672.2939785.

Chung, Junyoung, Sungjin Ahn, and Yoshua Bengio. “Hierarchical Multiscale Recurrent Neural Networks.” *ArXiv:1609.01704 [Cs]*, September 6, 2016. http://arxiv.org/abs/1609.01704.

Denil, Misha, David Matheson, and Nando De Freitas. “Narrowing the Gap: Random Forests In Theory and In Practice.” In *International Conference on Machine Learning*, 665–73, 2014. http://proceedings.mlr.press/v32/denil14.html.

Domingos, Pedro. “A Few Useful Things to Know about Machine Learning.” *Communications of the ACM* 55, no. 10 (October 1, 2012): 78. https://doi.org/10.1145/2347736.2347755.

Donner, Reik V., Y. Zou, Jonathan F. Donges, Norbert Marwan, and Juergen Kurths. “Recurrence Networks - A Novel Paradigm for Nonlinear Time Series Analysis,” August 24, 2009. https://doi.org/10.1088/1367-2630/12/3/033025.

Freund, Yoav, and Robert E Schapire. “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.” *Journal of Computer and System Sciences* 55, no. 1 (August 1997): 119–39. https://doi.org/10.1006/jcss.1997.1504.

Gregor, Karol, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. “DRAW: A Recurrent Neural Network For Image Generation.” *ArXiv:1502.04623 [Cs]*, February 16, 2015. http://arxiv.org/abs/1502.04623.

Loh, Wei-Yin. “Classification and Regression Trees: Classification and Regression Trees.” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1, no. 1 (January 2011): 14–23. https://doi.org/10.1002/widm.8.

Mason, Llew, Jonathan Baxter, Peter Bartlett, and Marcus Frean. “Boosting Algorithms as Gradient Descent in Function Space,” n.d., 29.

Sathya, R., and Annamma Abraham. “Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification.” *International Journal of Advanced Research in Artificial Intelligence* 2, no. 2 (2013). https://doi.org/10.14569/IJARAI.2013.020206.

Schmidhuber, Jürgen. “Deep Learning in Neural Networks: An Overview.” *Neural Networks* 61 (January 1, 2015): 85–117. https://doi.org/10.1016/j.neunet.2014.09.003.

Scornet, Erwan. “On the Asymptotics of Random Forests,” September 7, 2014. https://arxiv.org/abs/1409.2090.

Scornet, Erwan, Gérard Biau, and Jean-Philippe Vert. “Consistency of Random Forests.” *The Annals of Statistics* 43, no. 4 (August 2015): 1716–41. https://doi.org/10.1214/15-AOS1321.

Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. “Global Refinement of Random Forest.” In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 723–30. Boston, MA, USA: IEEE, 2015. https://doi.org/10.1109/CVPR.2015.7298672.

Srivastava, Nitish, Elman Mansimov, and Ruslan Salakhutdinov. “Unsupervised Learning of Video Representations Using LSTMs.” *ArXiv:1502.04681 [Cs]*, February 16, 2015. http://arxiv.org/abs/1502.04681.

Su, Jiang, and Harry Zhang. “A Fast Decision Tree Learning Algorithm,” n.d., 6.
Tsoi, Ah Chung, and R A Pearson. “Comparison of Three Classification Techniques: CART, C4.5 and Multi-Layer Perceptrons,” n.d., 7.
Vijayapriya, Tamilmaran, and Dwarkadas Pralhadas Kothari. “Smart Grid: An Overview.” *Smart Grid and Renewable Energy* 02, no. 04 (2011): 305–11. https://doi.org/10.4236/sgre.2011.24035.