Improving the Robust Random Forest Regression Algorithm

MAJ John R. Brence, Ph.D.
Adjunct Assistant Professor
Department of Systems Engineering
United States Military Academy
West Point, NY 10996
845-304-6416
john.brence@usma.edu

Donald E. Brown, Ph.D.
Department Chair and Professor
Department of Systems and Information Engineering
University of Virginia
Charlottesville, VA 22903
434-924-5393
brown@virginia.edu

ABSTRACT

Improving the Robust Random Forest Regression (RRFR) Algorithm leads to the discovery of a new forest prediction method called Booming. In previous research, we determined that RRFR was more robust than Random Forest Regression (RFR) to unbounded outliers and heteroscedastic datasets using a DFfits style analysis; however, with other dirty datasets RFR outperformed RRFR with respect to prediction error. The study's goal is to show why the mean prediction method (RFR) dominated our previous experiment, why RRFR did not perform as well with these datasets, as well as a discussion of Booming and how this forest prediction method improves the prediction of RRFR. We compare the performance of these algorithms using test dataset mean squared error (MSE) and mean absolute deviation (MAD).

KEY WORDS

Random Forest, Outlier, Robust Statistics, Regression, Tree-based Methods, Booming

1 INTRODUCTION

Random Forests were introduced in 2000 by Breiman [2]. This algorithm creates a classification tree forest. A single tree is created by using a bootstrap sample of the training dataset and bagging predictors in order to facilitate more efficient node-splitting. Bagging (Bootstrap Aggregation) is the generation of multiple (random) versions of a predictor in order to combine into an aggregated prediction. Bagging creates the forest prediction by combining numerous trees in order to take advantage of their predictive power [1]. Use of the Strong Law of Large Numbers indicates that the trees always converge (only if the mean is finite [exists]); therefore there is no issue with overfitting the data [1].

In April of 2001, Breiman introduced code for Random Forest Regression (RFR). This method mirrored the previous classification algorithm (Random Forests); however, it predicted and calculated error using the terminal node means and mean squared error. The thesis of this method is that complex predictors provide a wealth of interpretable scientific information about

the data and how it is predicted. In addition, among the current prediction methods, the most complex methods are the most accurate [2].

Robust Random Forest Regression embeds the methodology of Random Forest Regression [1] with a major difference ~ the introduction of robust prediction and error statistics using the median and other robust measures for prediction, and the use of mean absolute deviation (MAD) to calculate both the in-node and overall error. By adapting this new strategy, we theoretically lessen the effects of outliers and heteroscedasticity while retaining the beneficial components of RFR as well. RFR currently uses the node mean for prediction and mean squared error (MSE) to derive the in-node and overall error.

Definition: A random forest (regression) is a predictor consisting of a collection of tree-structure predictors $\left\{h(\mathbf{x},\Theta_z)_\psi, z=1...Z\right\}$ in which each tree casts an equal valued vote for the prediction at input vector $\mathbf{x}$ and where Θ_z are independently identically distributed random vectors whose elements are counts on the number of times an input row appears in the bootstrap sample [1]. z is the index for the tree in the forest and Z is the total number of trees in the forest. ψ is a statistic for central tendency, which could be the mean, median, or other measure.

For the z^{th} tree, we generate a random vector Θ_z, independent of the previous random vectors $\Theta_1,...,\Theta_{z-1}$ but with the same distribution; and we grow a tree using the training set and Θ_z, resulting in the predictor $h(\mathbf{x},\Theta_z)_\psi$ where $\mathbf{x}$ is an input vector [1].

The prediction based on the vectors $\mathbf{x},\Theta_z$ is conducted in two steps. First, at the tree level, the prediction $h(\mathbf{x},\Theta_z)_\psi$ is based on the statistic ψ chosen to determine central tendency of the observations within each terminal node ~ mean for RFR and median, trimean, broadened median or trimmed mean for RRFR. Second, all the tree predictions are combined to create a forest prediction $\left\{h(\mathbf{x},\Theta_z)_\psi, z=1,...Z\right\}$ using *bagging* (bootstrap-aggregation).

2 PREVIOUS EXPERIMENTAL RESULTS

2.1 Experimental Setup

In our previous experiment, we varied five factors to compare RRFR and RFR: Prediction method, terminal node size threshold, number of trees in a forest, the percentage of independent variables tried at each split, and the dataset used to build the training and test datasets. Each run consists of ten replicates.

The prediction method factor had eight parameters for testing. The experiment included the mean, median, broadened median, trimean, and four trimmed means (α =0.1, 0.2, 0.3, 0.4). The mean parameter was tested using Breiman's RFR which uses MSE to create node splits. The other parameters were tested using RRFR that implements MAD to create node splits. In addition, both codes output MSE and MAD for model comparison.

The terminal node threshold size, $N_{m_{threshold}}$, is a factor that sets a terminal node's population size. For example, if the threshold is five, a node is terminal on the first instance

when the number of observations is five or less. This factor was tested at three levels: five, ten, and twenty. The default size used in Breiman's code was five.

The number of trees in a forest, Z, were varied in order to test algorithm performance for small, medium, and larger forest values. This is done by setting Z equal to 50, 100, and 200 respectively. Initially, the value of 300 was tested as the larger tree value, but due to the size of some of the datasets (especially those over 1,000 observations); running 300 trees per forest caused the computer to crash while doing simultaneous replicates or was very time consuming (over a week for some runs). Upon comparison of the results from a run with 200 trees versus 300 trees there was minimal improvement with the increase wait time; therefore, 200 trees were deemed reasonable for this study.

The number of independent variables tried at each split I_m determines the number of predictor variables to try for each parent node split. Since each dataset has a different number of independent variables, we chose to express this parameter in percentage at three different levels: 25%, 50%, and 75%, in order to capture low, medium, and high values for subset selection. For example, 25% indicates that a random selection of 25% of the possible predictor variables are tried for each parent node split.

The last factor identifies the different datasets used in the experiment. This factor has eleven different levels. The datasets used in this experiment are Brence1, Brence2, Brence3, Corrosion, Abalone, Boston Housing, Ozone, Servo, Friedman1, Friedman2, and Friedman3. The origins and descriptions of these datasets are set forth in [5].

The five factors at various levels (unbalanced design) results in a full factorial model with 2,376 trials (23,760 overall individual runs based on 10 replications of each parameter combination). Center points, where applicable, were manually input so that each factor has at least three levels. These center points are used in order to test for non-linearities. Traditional design of experiment methods using center points (e.g. central composite) are not used since this design is unbalanced. Table 1 translates the coded values of each of these factors.

Factor	Coded Name	Math Equivalent
A	Pred	Prediction Method
B	Nthsize	$N_{m_{threshold}}$
C	Jbt	Z
D	Mtry	I_m

Table 1: Experimental Factor Translation

Our evaluation criterion is based on numerical accuracy, defined as how well, measured in terms of error rate, the algorithm predicts the true answer for each observation. A lower error rate indicates better performance. For this experiment, *all* the prediction methods were compared using both MSE and MAD. Each of these summary error rate measures were calculated based on an average of the replicates reported for each run on the test dataset.

2.2 Experimental Results

In general, we found that the more trees in a forest, the greater the percentage of independent variables tried at each split, and smallest terminal node size threshold was the best combination of parameters to complement a prediction method.

Table 2 is a summary table for performance of the prediction methods on all the datasets used in this study. The areas shaded in green show the criterion where a prediction method from RRFR outperformed the mean prediction method used in the RFR algorithm. For the most part, RFR outperformed RRFR which seems counter-intuitive based on the noisy, outlier-ridden datasets presented in this study.

Best Performances Roll-up		
	MSE_Tst	MAD_Tst
Abalone	Mean	Mean
Boston	Mean	Mean
Brence1	Mean	Mean
Brence2	Mean	Trim Mean 0.3
Brence3	Mean	Mean
Corrosion	Mean	Mean
Friedman1	Mean	Mean
Friedman2	Mean	Mean
Friedman3	Mean	Trimean
Ozone	Mean	Mean
Servo	Mean	Mean

Table 2: Best Performance Roll-up

Factor A (prediction method) dominated this experiment. The prediction method generally explained at least 75 – 99% of the training and testing error in each experimental model. The prediction method was also able to explain 60 – 99% of the error related to the runtime and runtime variance experimental models except for the servo dataset where the error explanation was more balanced due the relatively small size of the dataset (very fast runs). For the most part, the other factors and interaction terms included in these models only explained minor additional error.

The following example shows the dominance of factor A (prediction method) and its ability to gain significance for other factors based on even the slightest of interactions. Table 3 shows the ANOVA table for the experimental model that describes the servo dataset MAD test set error. This table is used to shows the dominance of factor A over the rest of the factors in the model. The dominance of factor A may be interpreted from the much larger F Value compared to the other values even though the most of the significance levels are reported similarly at $p < 0.0001$, which is software package's default minimum. In addition, an experimental model with only factor A has a coefficient of determination (R^2) of 0.8846.

ANOVA for Factorial Model — Response: Avg Tst MAD

Analysis of variance table [Partial sum of squares]

Source		Sum of Squares	DF	Mean Square	F Value	Prob > F
Model		6.125	43	0.1424	1239.956	< 0.0001
	A	5.435	7	0.7765	6759.367	< 0.0001
	B	0.009	2	0.0043	37.289	< 0.0001
	D	0.001	2	0.0007	5.808	0.0036
	AB	0.330	14	0.0236	205.265	< 0.0001
	AD	0.335	14	0.0239	208.357	< 0.0001
	BD	0.014	4	0.0036	31.409	< 0.0001
Residual		0.020	172	0.0001		
Cor Total		6.144	215			

Std. Dev.	0.011	**R-Sq**	0.997
Mean	1.049	**Adj R-Sq**	0.996
C.V.	1.022	**Pred R-Sq**	0.995
PRESS	0.031	**Adeq Precsn**	150.141

Table 3: ANOVA Table for MAD_Tst (Servo)

Another way to view the dominance of factor A is to view the main factor and the interaction plots and look for significance. Figure 1 shows the main effects plot for this example. This plot indicates whether a factor is significant by itself versus the response. From the main effects plots it is not difficult to see the major differences in factor A (Pred) and minor differences if factor B (Nthsize) and D's (Mtry) plots. This shows significance in factor A and no significance in B and D when modeled individually with the response. This result was also validated using a multiple comparison analysis.

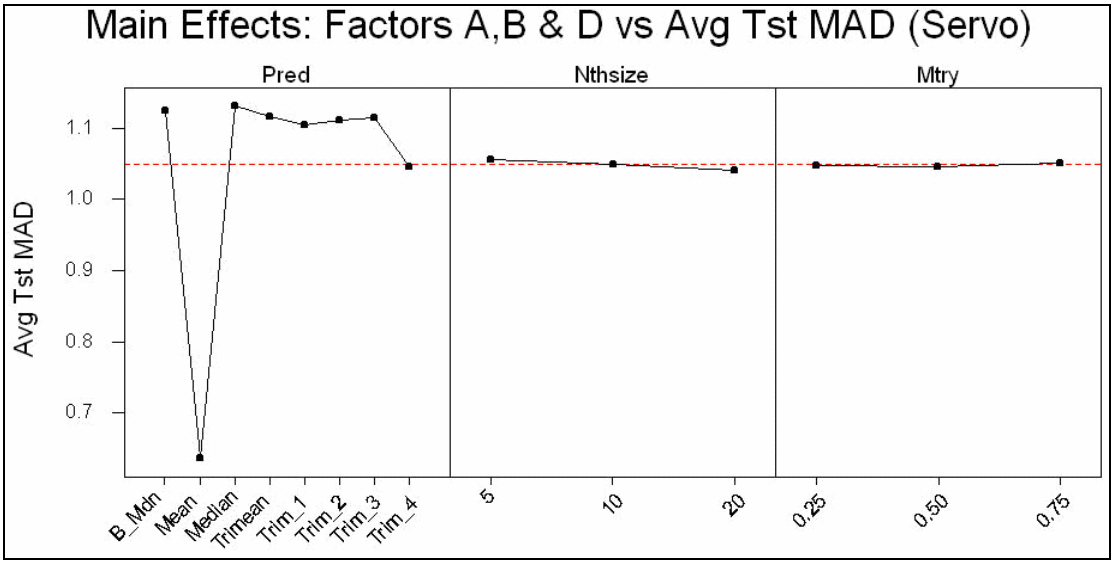

Figure 1: Main Effects Plots for Factors A, B & D for Servo Dataset

There is an interesting result when these three factors are included in the model with the interaction terms AB, AD, and BD. This happens mainly due to the dominance of factor A amplifying the minor differences in factors B and D. Figure 2 and Figure 3 show significance

(crossed lines) in interaction terms AB and AD, however slight that may be. When AB and AD are added to the model with A, B, and D, the factors B and D now become significant in the model. In addition, the interaction BD was found to be significant and assisted in describing even more model error. Figure 4 shows an obvious cross-over point from 0.5 to 0.75 for factor D.

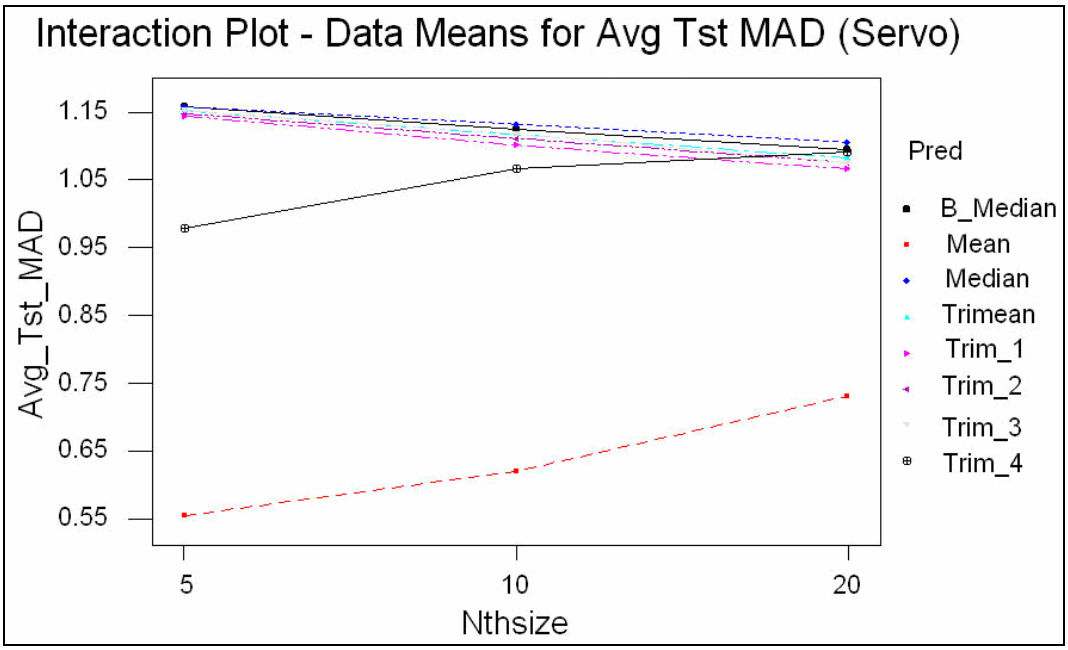

Figure 2: Interaction Plot: Factor AB for Servo Dataset

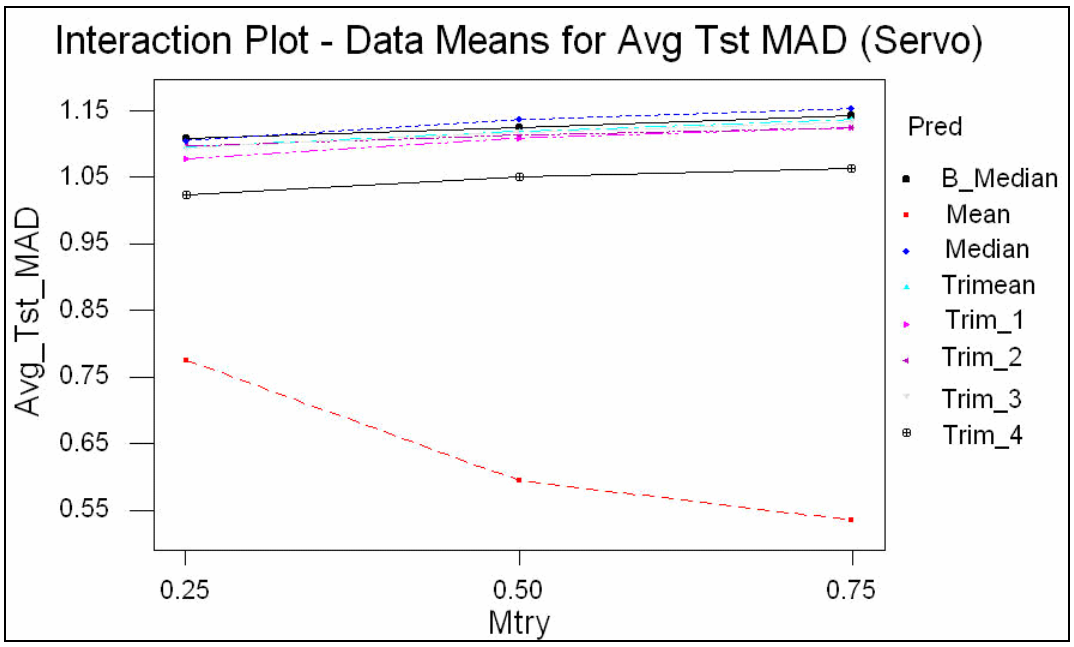

Figure 3: Interaction Plot: Factor AD for Servo Dataset

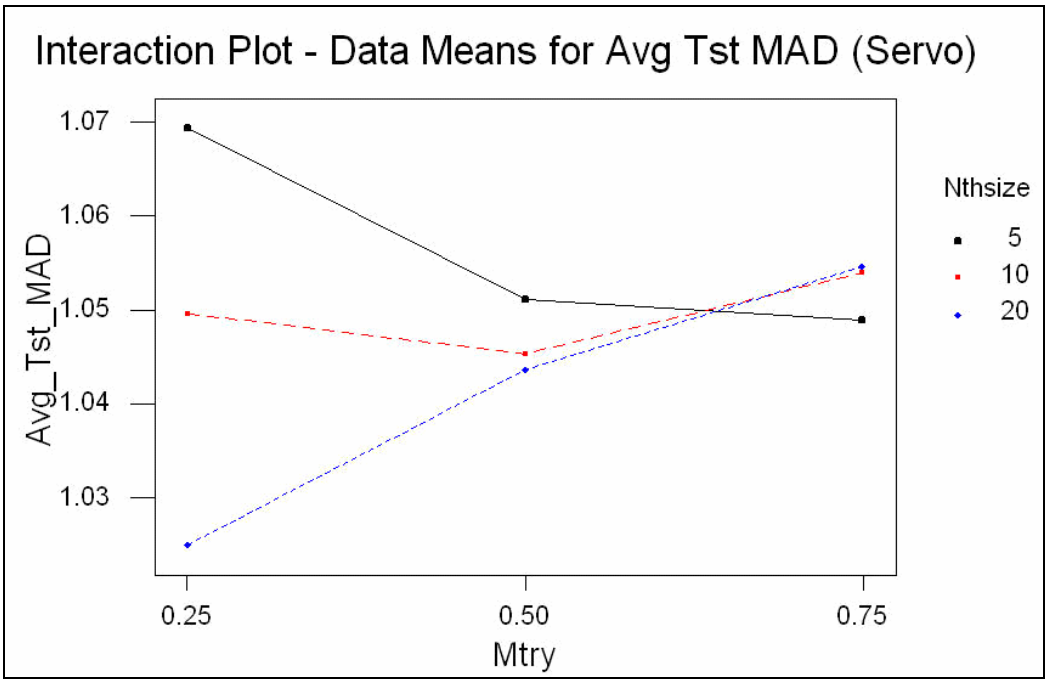

Figure 4: Interaction Plot: Factor BD for Servo Dataset

The real benefit of the additional factors (beyond just using factor A) in the experimental model is capturing the additional degrees of freedom coupled with the interactions' significance that assists in explaining the small remaining error. The increased degrees of freedom have a significant effect on the ANOVA table and its calculations, thereby increasing R^2 with minimal effect on the adjusted coefficient of determination (Adj-R^2). The adjusted R^2 typically decreases when unneeded variables are included in the model.

Figure 5 shows the progression in the predicted versus actual graph for three experimental model iterations. The first graph (a) shows the solo performance of factor A when modeling the MAD test error. This shows a decent explanation of the response since the experimental data points mostly adhere to the trendline. Graph (b) shows minimal improvement when factors B and D are introduced into the model. Graph (c) shows the real benefit of including the interaction terms AB, AD, and BD and why R^2 and adj-R^2 show excellent performance of this experimental model.

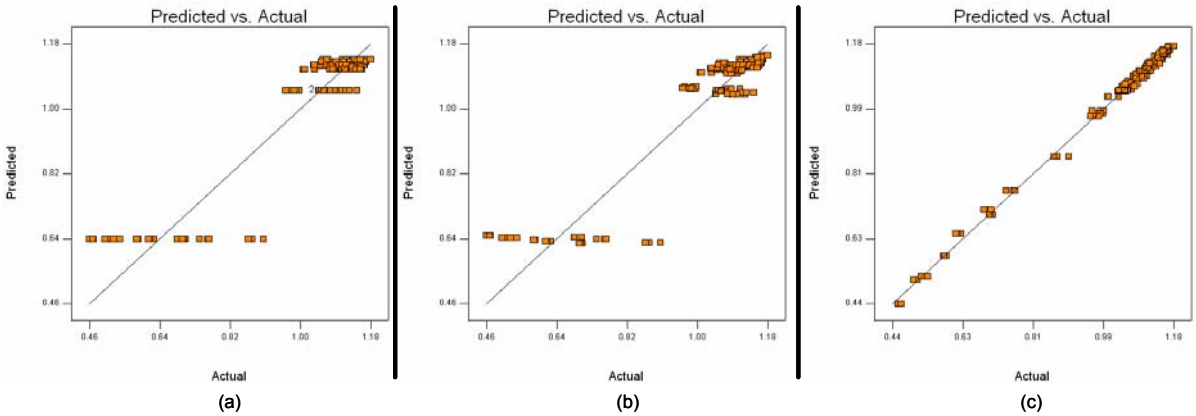

Figure 5: Prediction vs. Actual Progression for Servo Dataset

3 ROBUST PROPERTIES OF RRFR

This section provides support to the claim that robust measures are better suited to model data with extreme outliers. Random Forest Regression lacks robustness due to its use of the mean for prediction of a terminal node when an unbounded outlier is present. In contrast, Robust Random Forest Regression, using the median for prediction in a terminal node, is less influenced by an unbounded outlier.

Suppose that the training dataset contains an extreme unbounded outlier, y_r, then RFR, where $\psi = \overline{y}$, has node estimates that are not robust i.e. they are not bounded in the presence of this outlier as shown in theorem 1.

Theorem 1: For an observation y_r in the bootstrap sample, there exists a node m in the random forest with predictor $\left\{h(\mathbf{x},\Theta_z)_{\overline{y}}, z=1,...,Z\right\}$ such that $h_m(\mathbf{x},\Theta_z)_{\overline{y}} \to \infty$ as $y_r \to \infty$.

Proof. At each tree in the random forest with predictor $\left\{h(\mathbf{x},\Theta_z)_{\overline{y}}, z=1,...,Z\right\}$ for which the y_r is in the bootstrapped sample there exists a terminal node, m that contains y_r. The existence of this node follows from construction since this data point must reach a terminal node. Let the predictor at this node be $h_m(\mathbf{x},\Theta_z)_{\overline{y}}$. Then $h_m(\mathbf{x},\Theta_z)_{\overline{y}} = \frac{\sum^{N_m}\mathbf{y}}{N_m}$. But $y_r \to \infty$ so $h_m(\mathbf{x},\Theta_z)_{\overline{y}} \to \infty$.

QED

An important issue is existence of this extreme unbounded outlier in the bootstrapped training dataset. The probability a single outlier will be included in the bootstrap sample of a single tree is

$$P(y_r \mid T_z) = 1 - \left[\left(\frac{N-1}{N}\right)\right]^N \geq 0.63 \qquad (3.1)$$

where y_r is a single instance of an unbounded outlier, T_z is a single tree and $N \geq 1$ is the number of observations in the modeled dataset

As we increase the number of trees to build a forest, the probability that a single outlier is included in the bootstrap sample of a forest is

$$P(y_r \mid T_Z) = 1 - \left[\left(\frac{N-1}{N}\right)\right]^{N*Z} \approx 1 \qquad (3.2)$$

where y_r is a single instance of an unbounded outlier, T_Z is a forest of size $Z \geq 5$ trees and $N \geq 1$ is the number of observations in the modeled dataset.

Since we are building forests, we can see that y_r will impact the predictions based on the high probability of inclusion in the bootstrapping process.

In the presence of an unbounded outlier, y_r, RRFR, where $\psi = \tilde{y}$, is robust i.e. the predictor remains bounded.

Theorem 2: Assume y_r is in a single training set sample, then for all nodes, $N_m \geq 3$, in the forest $h_m(\mathbf{x}, \Theta_z)_{\tilde{y}} < \infty$.

Proof. For any terminal node m, where $N_m \geq 3$, containing y_r, $h_m(\mathbf{x}, \Theta_z)_{\tilde{y}} < \infty$ since $h_m(\mathbf{x}, \Theta_z)_{\tilde{y}} = \underset{j \in J_m}{median}\{y_j\}$ where $J_m =$ the set of indices of y_i in node m, $i = 1, ..., I$. **QED**

Theorem 2 holds when only one instance of y_r is in the node. However, the theorem fails, $h_m(\mathbf{x}, \Theta_z)_{\tilde{y}} \to \infty$, when more than one instance of y_r is bootstrapped and inhabits a node with $N_m \leq 4$. In the case when two instances of y_r are in the bootstrap and hence the node, the probability is $P(y_r : 2 \mid T_z) = 1/N^2$. To protect against this, we can set $N_m \geq 5$ and/or choose a very large training dataset. The probability of three y_r in the bootstrap further diminishes to $P(y_r : 3 \mid T_z) = 1/N^3$ which is very small for a large dataset.

4 WHY DOES RFR PERFORM WELL EVEN WITH OUTLIERS?

Throughout this empirical study, we found that the RFR algorithm, using the mean prediction method and the MSE splitting criterion, performed very well. This is an interesting result based on the background of many of the datasets used in this experiment (see [5]) and the presentation of Theorem 1 in section 3. Many of these datasets were extremely noisy or contained one or more potential outliers, in which some incorporated extreme outliers (much greater than $|\bar{y} \pm 3\sigma|$). This result goes against the thesis and results reported in much of the literature referenced in this study [10,11,12,13,15,16,17,18,19].

The key to this unfounded robustness is two-fold 1) Theorem 1 is dependent on an unbounded outlier, not found in these data and 2) the bagging (bootstrap-aggregation) heuristic. The first claim is taken at face value, while the second is further explained in the following sections.

The *bagging* heuristic combines bootstrapping of the training dataset to ensure different trees are created in the forest and the aggregation (averaging) of the predicted responses to produce the final prediction. Figure 6 through Figure 8 visually depict how RFR handles outliers with the bootstrapping heuristic. Figure 6 shows the original training dataset prior to bootstrapping. This figure includes two graphs: 1) the left one for readability and 2) the right one to show where there are multiple observations that overlap. Note in Figure 6 that there are synthetic outliers that are above and below the general shape of the graphical curve. These outliers are single instances as shown in the right portion of the figure.

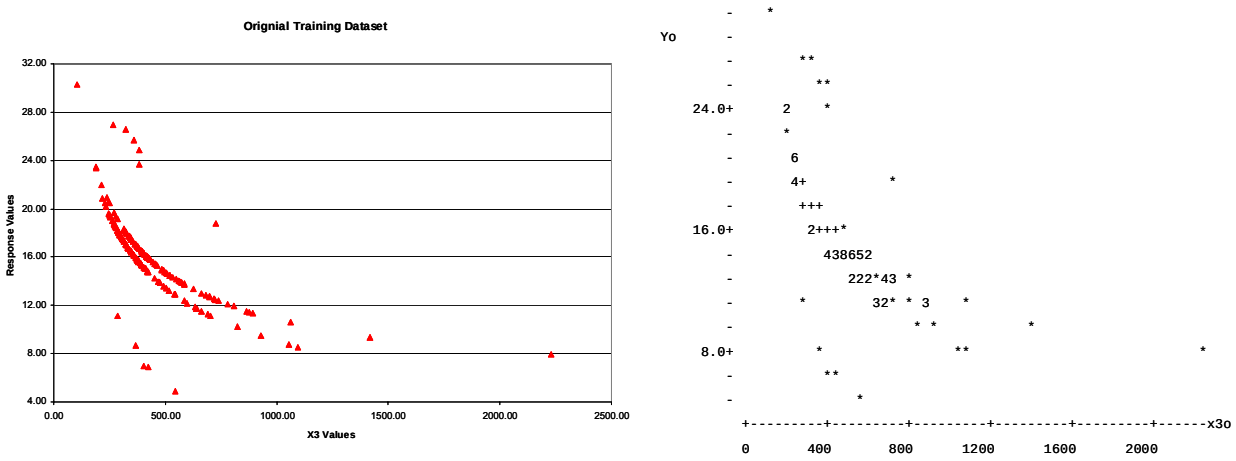

Figure 6: Original Brence3 Training Dataset Scatter Plots for the (*Y vs.* X_3)

Figure 7 shows the bootstrapped dataset that is modeled for the first tree in the forest. There are a few items to note in this dataset: 1) when comparing this graph to the original dataset, there are missing observations, 2) many of the outliers now have multiple observations, denoted by the number next to the observation in the right graph of the figure.

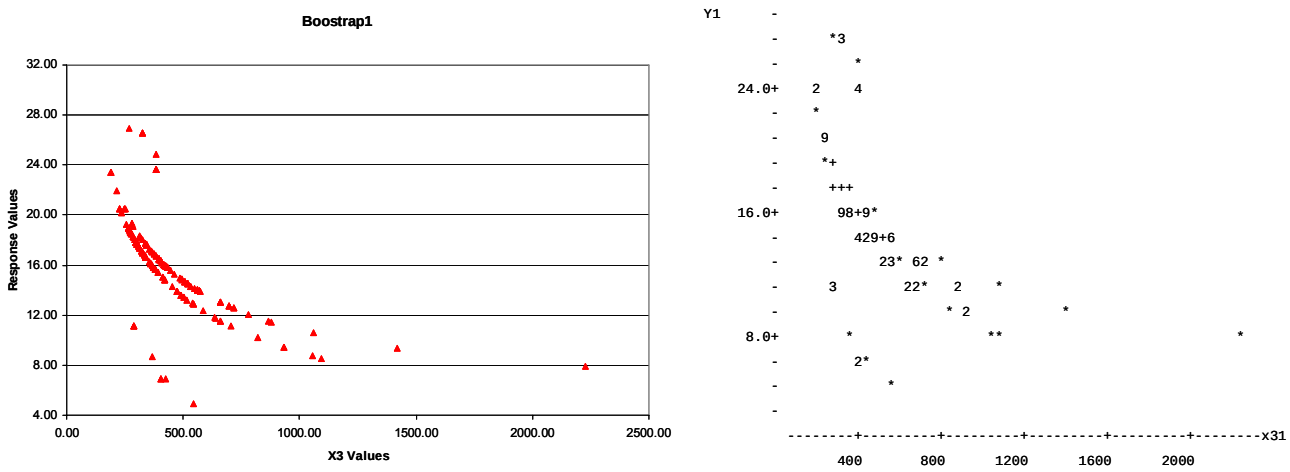

Figure 7: Bootstrap1 Sample for Brence3 Dataset Scatter Plot (*Yb_1 vs. Xb_{31}*)

A similar result from Figure 7 is found in Figure 8. It is especially clear that there are multiple observations for outliers when you focus your attention of the right-most observation in the right hand graph.

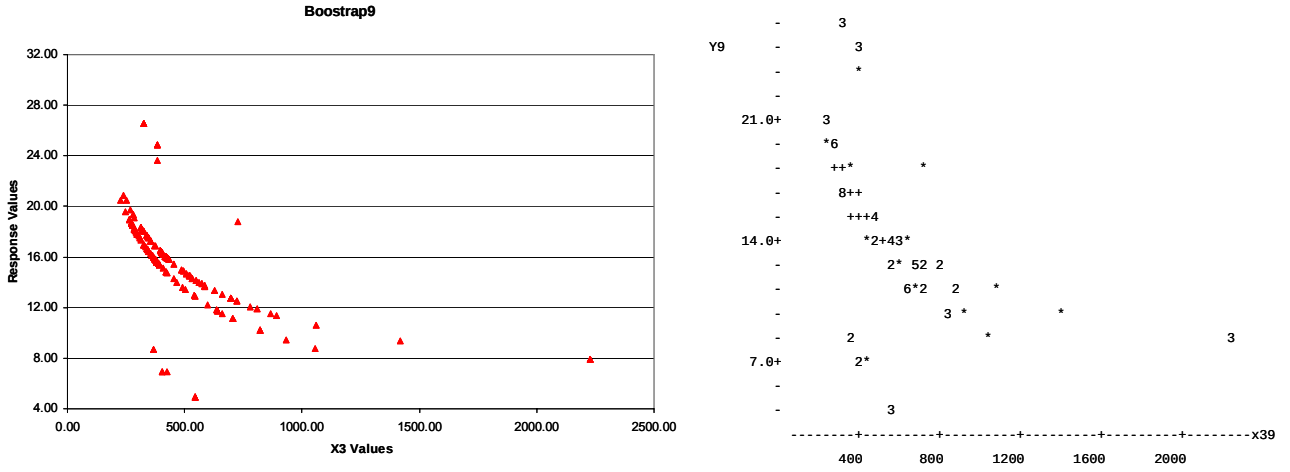

Figure 8: Bootstrap9 Sample for Brence3 Dataset Scatter Plot (*Yb_9 vs. Xb_{39}*)

These are a few snippets of what bootstrapping does for a dataset. This method essentially lessens the effect of outliers through random sampling of the training dataset whether it ignores an observation for a specific modeling run or multiplies it.

It is important to note that the iterative nature of RFR is very important to the effectiveness of bootstrapping. Without the multiple runs of trees and the aggregation of the responses, bootstrapping may not do very well. A single run with a bootstrapped sample generally provides a high error result. This error does not get confounded until multiple trees are included in the model. Figure 9 displays how the MSE training converges to a solution after approximately fifteen runs of a 200 run model, thereby highlighting the importance of bootstrap iteration.

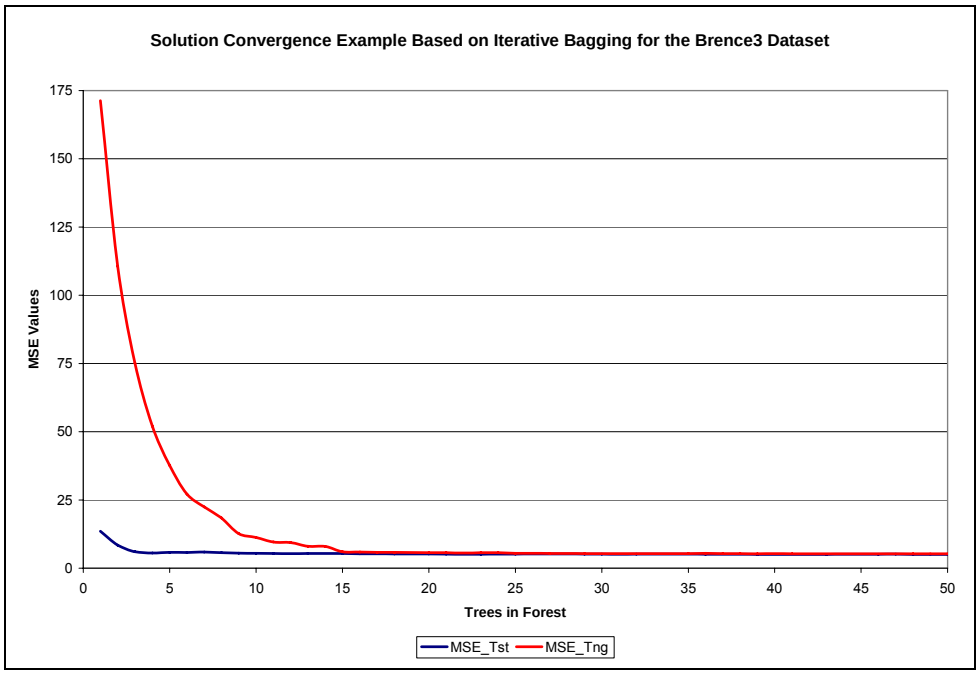

Figure 9: Example Convergence for Brence3 Dataset

The last reason that RFR performs well is answered in the aggregation of the *out-of-bag* predicted responses. The mean tends to perform better because even when forced higher or lower than the true value of the response, it tends to work its way back to that solution, while the median is slower to return to the true response (shown as the horizontal lines in Figure 10). The counter-argument may reveal, that in the figure presented, the median performs better after bootstrap 2 and appears to start its descent to the response line at bootstrap 10. The critical issue to point out is that this may be the case; however, the median tends to be a more erratic estimate and the mean does a better job to balance its predictions. In response to the argument at bootstrap 2, Figure 9 and the previous discussion explains why it is better to create forests much greater than two trees (2 trees = 2 bootstrap samples).

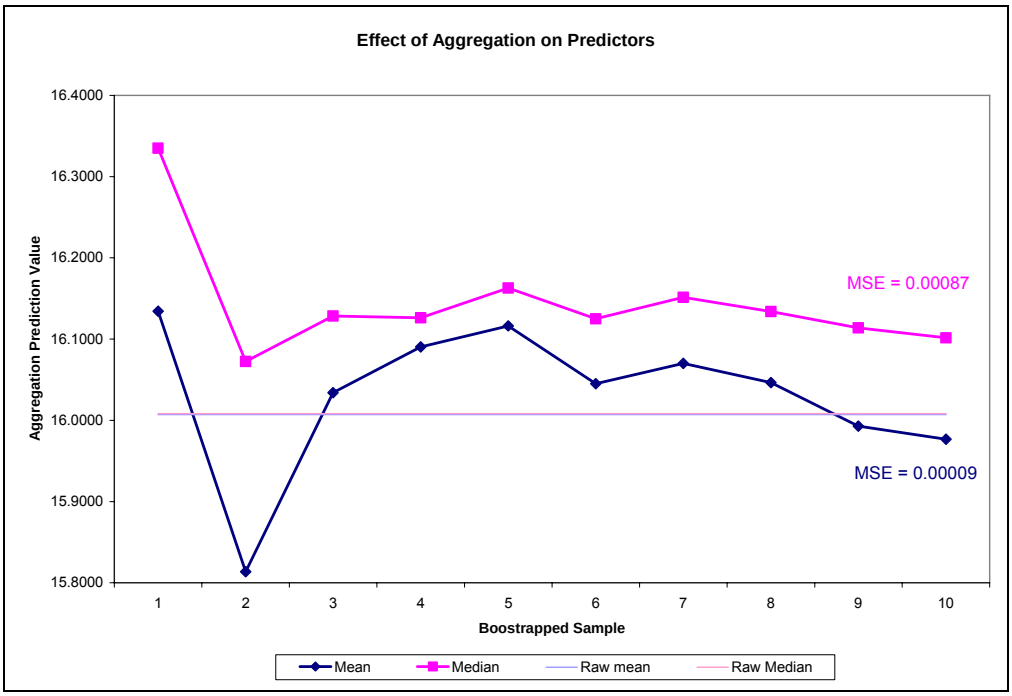

Figure 10: Example of the Effect of Prediction Aggregation on Mean and Median

This result may be because of the nature of aggregation. The mean prediction method is an aggregative measure, taking into account all of the values of the distribution for prediction, whether they are good or bad. The median, however, will mask the effects of much of the data, only taking into account the middle-ranked value of the distribution. For this same reason, some of the robust measures that more closely relate to the mean often outperform the median prediction i.e. broadened mean, trimean and trimmed means. The trimmed mean is very dependent on good choice for the α value in this instance.

5 A CURSORY EXPLORATION INTO WHAT WE CALL BOOMING

After reviewing the theorems presented in this paper and the empirical results from section 2.2 we returned to an initial algorithm concept we previously dismissed early in our research. The concept, which we refer to as *booming* or bootstrap-mediation, was dismissed because it requires a serious overhaul of the current code and most likely a change of coding language to something more efficient. *Booming* follows the same construct as *bagging* except instead of aggregating the tree votes, *booming* takes the median of the tree votes (Figure 11). Mathematically, the *booming* algorithm when coupled with RRFR (median) has the intent to

$$E_{Y\mathbf{X}}\left|Y - median_{Z} h(\mathbf{X}, \Theta_{Z})_{\tilde{y}}\right| \rightarrow E_{Y\mathbf{X}}\left|Y - Median_{\Theta} h(\mathbf{X}, \Theta)_{\tilde{y}}\right| \quad (5.1)$$

For a count of 1 to N_{Tng} (number of training observations)
    If that sample observation was NOT bootstrapped then
    Calculate forest estimate based on median of *out-of-bag* tree estimates
    Maintain count for number of trees used
    End (Calculation If)
  Calculate the current error of the estimates
End (Observation Loop)
    Calculate overall error of estimates

Figure 11: *Booming* Pseudo-Code

The motivation in resurrecting this method was spurred by the occurrence of 30 or so outlier influences per run found in RRFR in [4]. We suspected that the outlier influence prediction in RRFR stemmed from the aggregation of tree votes from rare-occurring terminal nodes; nodes that that choose the outlier as the median prediction. Aggregation, used in *bagging*, simply calculates the mean of the tree predictions, a method that an outlier may easily influence.

We were able to conduct a cursory test on the effectiveness of the *booming* algorithm by running RFR and RRFR (median) with a minor alteration; the tree predictions were collected in worksheet and the forest prediction was calculated manually.

We ran one iteration of both RFR and RRFR (median) on data that included only one extreme outlier. The observation is an outlier due to an enormously large response term in comparison to the other responses. The outlier response is on the order of 10^{38} compared to values of 10^{3}. When the dataset is examined using Cook's D (Figure 12) and DFfits (Figure 13) the outlier is clearly identified.

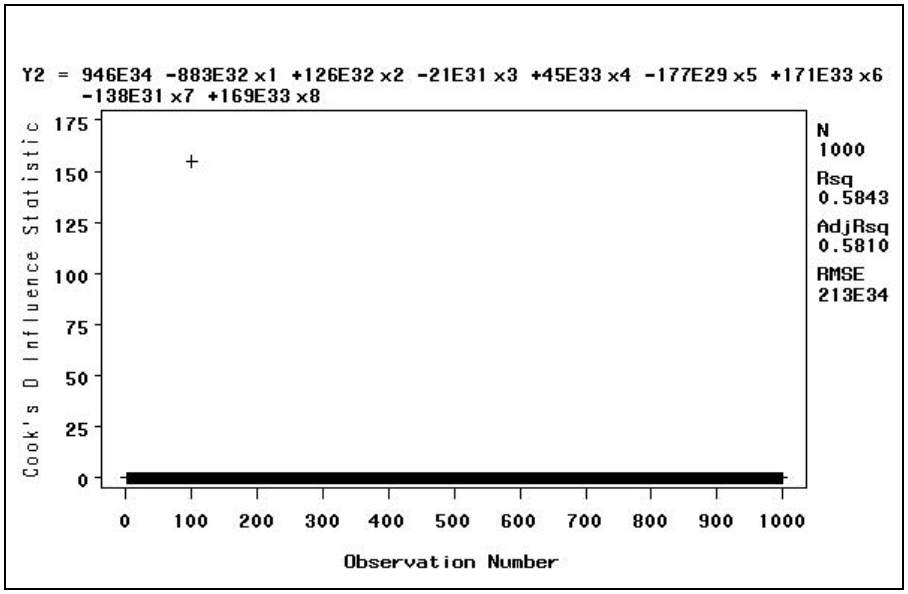

Figure 12: Cook's D for Extreme Outlier Case

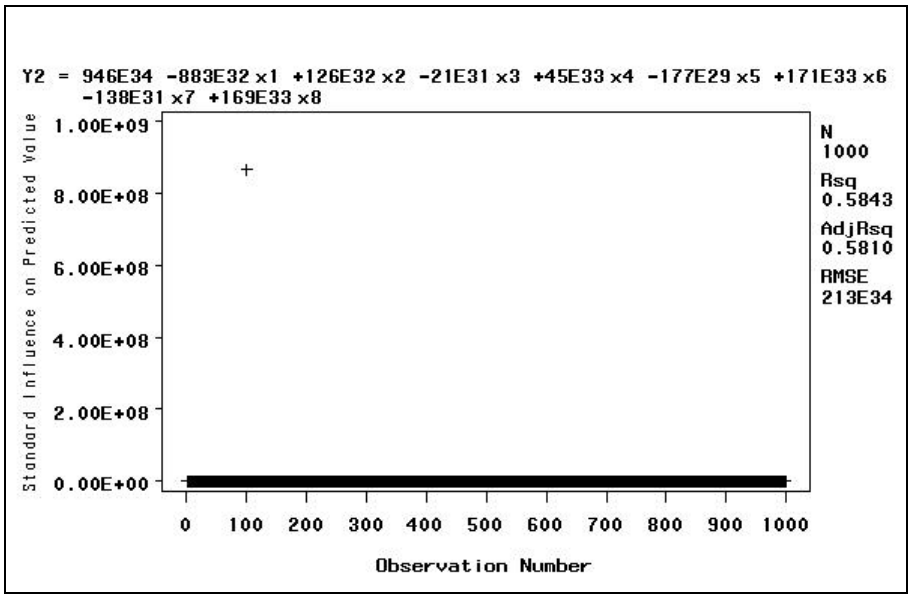

Figure 13: DFfits for Extreme Outlier Case

Table 4 and Table 5 shows the results of using *booming* versus *bagging* in one run of RFR and RRFR (median). The *Outlier Inf* row displays the number of times that the prediction was far greater than the actual response value. Influence was easy to detect since the prediction values were much greater than the true observations or $>>10^3$. The *MAD* row is the calculation of the mean absolute deviation for all 1000 observations, including the extreme outlier. The *MAD*** row is the mean absolution deviation calculation that does not consider the deviation for the outlier. MAD** is calculated because even if the prediction is small, the deviation is quite large because of the true value of the outlier response.

RFR	Bagging	Booming
Outlier Inf	879	608
MAD	1.94E+35	1.81E+35
MAD**	9.04E+34	7.68E+34

Table 4: Booming vs. Bagging ~ RFR

The RFR code, for this dataset, shows improvement in all areas when the *booming* algorithm is used. The outlier influence is reduced by 271 observations, and both the MAD and MAD** are reduced by 1.36×10^{34}. This is a significant improvement in dealing with outliers; however, RFR is still plagued by the lack of robustness to an extreme outlier by the mean prediction method and MSE.

RRFR	Bagging	Booming
Outlier Inf	30	0
MAD	1.57E+35	1.04E+35
MAD**	5.35E+34	4.82

Table 5: Booming vs. Bagging ~ RRFR

An intriguing and more promising result is found when running the *booming* algorithm with the RRFR (median) code. The *booming* algorithm is more robust to the extreme outlier influence that *bagging*. For booming, all outlier influence in the predictions is rejected, the MAD is reduced by 5.34×10^{34} and the MAD** is a mere 4.82, a reduction of similar order to the entire *bagging* MAD** at 5.35×10^{34}.

In addition, *booming* seems to work well with heteroscedastic datasets. We ran a similar experiment with a heteroscedastic (displaying non-constant variance) dataset and found some promising results. To construct our heteroscedastic dataset, we used a mean value of 10,000 for both distributions; however, distribution 1 (blue in Figure 14) had 100 observations and a standard deviation of 1,000 while distribution 2 (pink in Figure 14) had 900 observations and a standard deviation of 10.

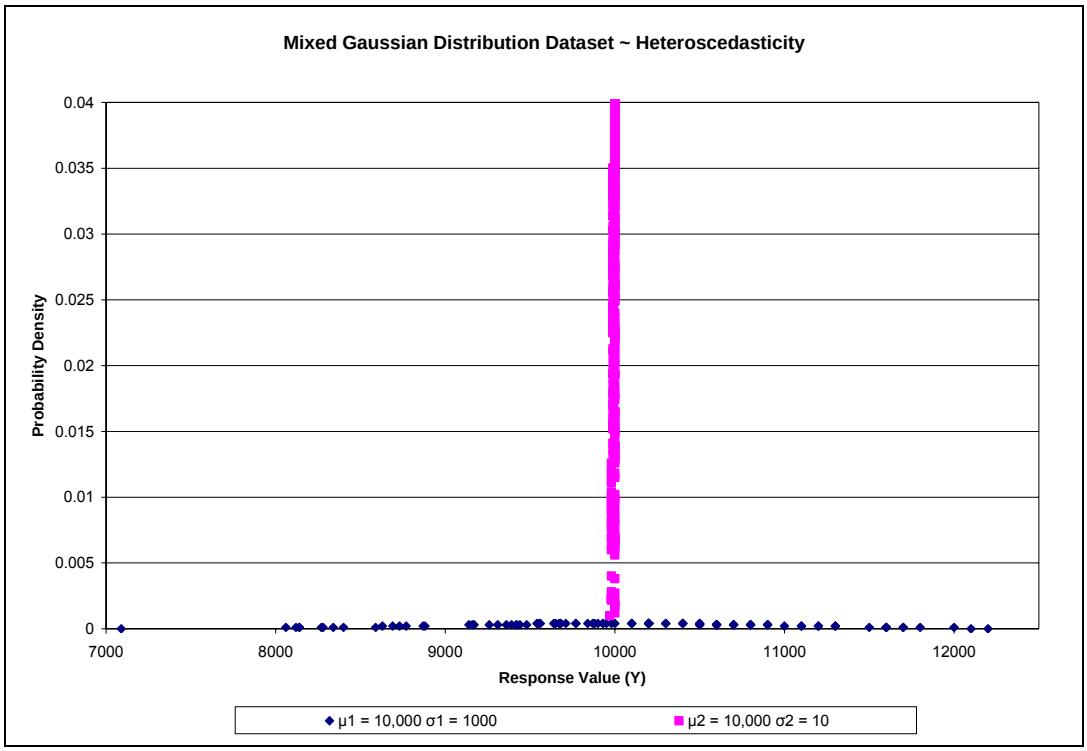

Figure 14: Mixed Gaussian Distribution Dataset ~ Heteroscedasticity Experiment

Table 6 shows that in both the RFR and RRFR runs the *booming* algorithm performs better than *bagging*. In both cases, *booming* does a better job in weeding out the influence from the large variance data.

RFR	Bagging	Booming
MAD	131.43	86.04
RRFR	Bagging	Booming
MAD	116.05	83.82

Table 6: Booming vs. Bagging ~ Heteroscedastic Dataset

In conclusion, we suggest that the *booming* algorithm, when used with extreme datasets, presumably does better with RRFR (median) due to the ability of RRFR to ignore outliers when building the nodes of the tree and *booming's* ability to ignore those rare-occurring terminal nodes with the outlier-valued prediction.

6 CONCLUSION

The literature reviewed for this research explains that robust measures should outperform the mean and MSE when modeling noisy and outlier-ridden datasets. In this study, we found that this thesis did not apply mainly due to the application of the *bagging* subroutine within the algorithm. As explained in section 4, the mean and MSE are a better statistic combination for the central tendency when coupled with the *bagging* subroutine. With the prediction power gained from this application, it seems unnecessary to add robust measures when predicting the central tendency of a distribution. In addition, the experimental results (section 2.2) show that the prediction method was the most influential factor used in most every modeling run. More often than not, the best predictor was the mean coupled with MSE in the RFR algorithm.

We then introduced a new algorithm to consider for use with RFR and RRFR. *Booming* (bootstrap-mediation) is similar to *bagging* except that it takes the median of tree predictions rather than the mean as the forest prediction. This method, on cursory review, has shown promise when modeled against an extreme outlier and a heteroscedastic dataset. The results from *booming* suggest that this algorithm will add to the robustness against noisy, outlier-ridden, irregular datasets especially if coupled with an RRFR style algorithm.

The application of *booming* is the most promising area of research uncovered in this study. An extensive exploratory analysis of the *booming* algorithm and the determination of its validity would be a beneficial step in this area of research. This requires a large effort to re-code RRFR in a more efficient/flexible coding environment (better than FORTRAN 77/90) with the *booming* algorithm. This is the basis for our future research.

7 REFERENCES

1. Breiman, L. [2001] *Random Forests*, Statistics Department, University of California, ftp://ftp.stat.berkeley.edu/pub/users/breiman/.
2. Breiman, L. [2000] *Understanding Complex Predictors*, Statistics Department, University of California. ftp://ftp.stat.berkeley.edu/pub/users/breiman/.
3. Brence, John R. & Brown, Donald E. [2002] Data Mining Corrosion from Eddy Current Non-Destructive Tests, *Computers and Industrial Engineering: Data Mining and Knowledge Discovery in Industrial Engineering*. 43, pp. 821- 840.

4. Brence, John R. & Brown, Donald E. [2004] Analysis of Robust Measures in Random Forest Regression, submitted to *Data Mining and Knowledge Discovery*, Kluwer Academic Publishers.
5. Brence, John R. & Brown, Donald E. [2006] Comparative Analysis of Two Forest-based Regression Algorithms, Technical Report sie06_0003, Department of Systems and Information Engineering, University of Virginia.
6. Christensen, Ronald [1989] Data Distributions: A Statistical Handbook. 2 ed., Entropy Limited, Lincoln, MA.
7. Friedman, J. [1991] Multivariate Adaptive Regression Splines, *The Annals of Statistics*, 19, 1, pp. 1-67.
8. Forsyth, D.S. [2000] *Nondestructive Inspections of Calibration Specimens and KC 135 Aircraft Specimens*. Institute for Aerospace Research, Canada.
9. Gumbel, E.J. [1958] Statistics of Extremes, Columbia University Press, New York, NY.
10. Hampel, Frank R., Ronchetti, Elvezio M., Rousseeuw, Peter J., & Stahel, Werner A. [1986] Robust Statistics: The Approach Based on Influence Functions, John Wiley and Sons, New York.
11. Hastie, Trevor, Tibishirani, Robert, & Friedman, Jerome [2002] The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York.
12. Hoaglin, D. C., Mosteller, F., and Tukey, J. W. (eds.). [1983] Understanding Robust and Exploratory Data Analysis, New York: John Wiley & Sons.
13. Hoaglin, D. C., Mosteller, F., and Tukey, J. W. (eds.). [1985] Exploring Data Tables, Trends, and Shapes, New York: John Wiley & Sons.
14. Kotz, S, & Nadarajah, S. [2001] Extreme Value Distributions: Theory and Applications, Imperial College Press, World Scientific Publishing Company, Portland, Oregon.
15. Martin, Doug. [2002] *Robust Statistics and Data Mining for Outliers with S-Plus*. Insightful Corporation. www.insightful.com.
16. Mosteller, F. and Tukey, J. W. [1977] Data Analysis and Regression, Cambridge, MA: Addison-Wesley.
17. Neter, J., Kutner, M., Nachtsheim, C., & Wasserman, W. [1996] Applied Linear Statistical Models, 4 ed., Boston, MA: McGraw Hill.
18. NIST, *Engineering Statistics Handbook*. http://www.itl.nist.gov .
19. Velleman, P. F. and Hoaglin, D. C. [1981] Applications, Basics, and Computing of Exploratory Data Analysis, Duxbury Press.