A comparison of single and multiple response machine learning algorithms for species distribution modeling

Julie A. Lapidus[1] and Eli L. Moss[2]

[1]*Scripps College, 1030 Columbia Avenue Claremont, CA 91711 USA; Eco-Informatics Summer Institute, PO Box 300, Blue River, OR 97413 USA*
[2]*Brown University, 45 Prospect Street Providence, RI 02912 USA; Eco-Informatics Summer Institute, PO Box 300, Blue River, OR 97413 USA*

Abstract. Species distribution modeling has previously been used to determine the geographic distributions of species by relating environmental covariates to each species independently. However, from an ecological standpoint, it seems more informative to use species co-occurrence as a factor in determining spatial distribution. For this reason, we compare the results of both single response (elastic net logistic regression and single-response decision trees) and multiple response machine learning algorithms (neural networks and multiple-response decision trees). These were each tested on presence-absence datasets of 100 Southeastern Australian plant species and 606 moth species in the H. J. Andrews Experimental Forest in the North American central Cascade Mountain Range. In addition, an auxiliary-task model algorithm, using a multiple decision tree model to predict each species individually, is implemented for the moth data. We present our results along with a context in which to analyze them and suggest that including more species-specific covariates would improve the accuracy of presence-absence predictions for the moth dataset. Furthermore, we demonstrate that independently thresholding for cutoff for each species results in more accurate predictions. We also create a grid layer for the moth dataset that can be used in future studies to predict moth species for the entire H. J. Andrews Experimental Forest. We suggest the implementation of a four-part hybridization model that would model each species using the algorithm that is most accurate at predicting that species, as we believe this would produce more accurate predictions than any of the algorithms alone.

Keywords: species distribution models; presence-absence; multiple response algorithms; single response algorithms; multiple response decision trees; single-response decision trees; elastic net logistic regression; single hidden-layer feedforward neural networks; auxiliary-task models; R.

1. INTRODUCTION

Species distribution modeling can have important applications for conservation biology practices. Species distribution models have been used to help determine the causal relationships between species and their environment and the relative importance of various environmental factors in their distributions [1], [2]. With increased interest in global and local climate change and the effects that it may have on various species and their global and local distribution

patterns, the need for accurate and effective species distribution models has risen greatly [1], [2], [3]. In recent studies, species distribution models have been created by relating environmental covariates to each species independently [1], [2]**.** However, it is a well-known fact that certain sub-communities tend to co-exist due to shared environmental preferences. For this reason, it seems that from an ecological perspective, it would be more informative to use species co-occurrence as a factor in determining spatial distributions. It is our belief that predicting species simultaneously based on the same environmental covariates in this way may thus provide a more accurate picture of the distribution of species.

In order to test this hypothesis, four different machine learning algorithms were implemented on two different datasets to determine the respective accuracy of each algorithm. The two single-response algorithms—elastic net logistic regression and single-response decision trees—related environmental covariates to each species individually, while the two multiple-response algorithms—single hidden-layer feedforward neural networks and multiple-response decision trees—related environmental covariates to all species simultaneously. Each of these algorithms was tested on two different presence-absence datasets: 1) one of Southeastern Australian vegetation species [4] and 2) one of moth species of the North American central Cascade Mountain Range [5].

R, an open-source programming language for statistical analysis, was used to implement each of these four algorithms. Each algorithm was implemented by using a different package in the R environment. R version 2.9.1 was used at the time of the implementation of these models. The packages used for each algorithm were as follows: 1) glmnet [6] for elastic net logistic regression, 2) nnet [7], [8] for single hidden-layer feedfoward neural networks, 3) rpart for single-response decision trees [9], and 4) mvpart for multiple-response decision trees [10]. The results of these four algorithms were analyzed and compared for the accuracy of species occurrence prediction for each species for both datasets using Cohen's kappa as the main performance measurement, although more extensive analysis was performed on the results from the North American moth dataset.

In addition, an auxiliary-task model algorithm—a multiple-response model tuned to predict a single species—was implemented for multiple-response decision trees on the North American moth dataset. Furthermore, a grid layer was developed for the H. J. Andrews Experimental Forest that can be used for presence-absence prediction for the entire region in future studies. In this paper, we present our findings from each model implementation as well as a means of and context in which to analyze our results and introduce a new approach to implementing these algorithms.

2. EXPERIMENTAL METHODS

2.1 Data Sources

2.1.a. Southeastern Australian Vegetation Data

One of the datasets used in our study consisted of 5,605 plant species of Victoria, Australia measured at over 113,000 sites. The data were collected by and are owned by the Arthur Rylah

Institute in Melbourne, Australia [4]. We used a subset of this dataset containing the 100 most abundant plant species over 15,328 sites (see Figure 1) for our models. The data contain 81 covariates that were used as inputs for the modeling algorithms. The raw data were provided as discretized abundances that were then converted to presence/absence data. That is, all abundance codes "+" and all numeric codes were treated as present. The data was then formatted in the following manner: A 1 kilometer by 1 kilometer grid was imposed on the data, and all sites (plots) that fell within a single grid cell were combined as follows: All species occurrences within the cell were combined (i.e., if a species was present at any one of the sites, then it was recorded as being present in the grid cell). Then, the covariates (predictor variables) were copied from the site that was closest (in Euclidean distance) to the center of the grid cell.

The grid cells were then grouped into an arrangement of alternating 2500 square kilometer blocks in a checkerboard pattern. The data were then separated by odd and even vertical stripes across the grid to split the data into two halves for use as the training and testing data sets. For the purpose of our models, each dataset was then sorted by highest abundance per species. The eighty-one covariates used from this dataset as predictors in the models include attributes of rainfall amount per month, various temperature values, slope, and aspect, amongst many others. For the purposes of this paper, the Southeastern Australian vegetation dataset will be referred to as "plant data."

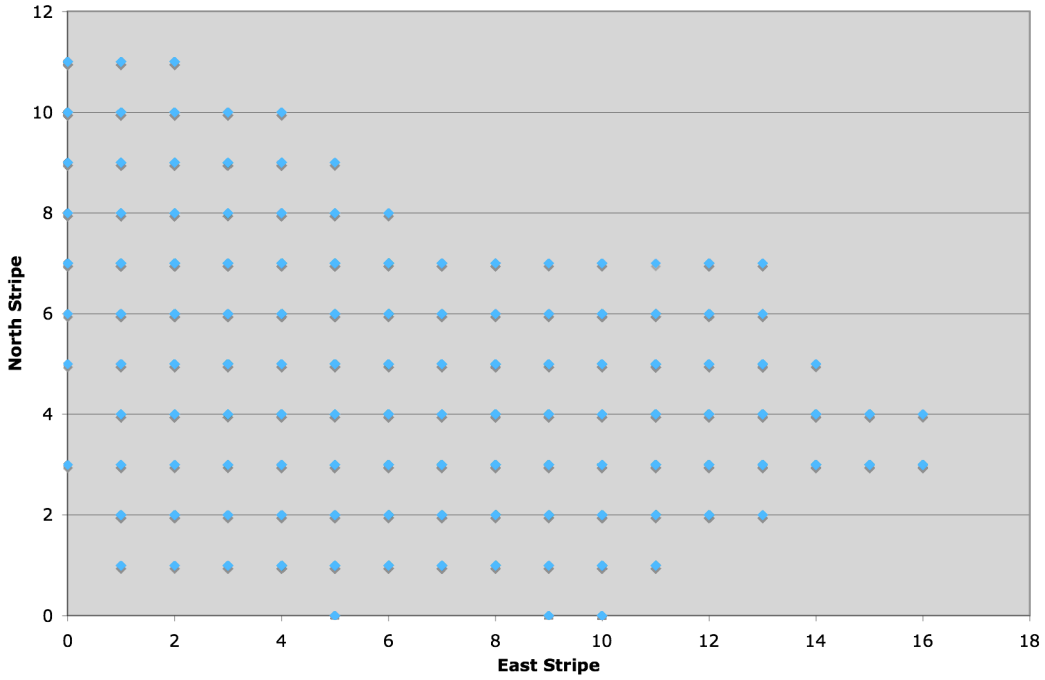


Figure 1. A visual representation of plant dataset site locations plotting the northing vs. easting.

2.1.b. Northwestern North American Moth Data

The second dataset used in this study is from a database of moth traps throughout the H. J. Andrews Experimental Forest in the Northwestern United States. These data were collected and compiled by Dr. Jeffrey Miller, an ecologist at Oregon State University, between the years of

1986 and 2008. An accompanying set of environmental covariates for each trap site was created using ArcGIS software using a digital elevation model provided by the state of Oregon. Each trap site was matched to 4 different environmental covariate values. The covariates used for this dataset were slope, aspect, elevation, and the vegetation type of each trap site. Because this study involves predicting species occurrence related to the environment, the quantity of moths recorded at individual traps was discarded, and a 1 was recorded if the species occurred at that trap site over the course of the entire 23-year trapping period. Thus a vector of zeroes (absences) and ones (presences) was created for each species, with each position corresponding to a trap site. These were assembled as columns in a matrix, preceded by columns corresponding to each covariate recorded for each trap site. Each row then contained first the value for each covariate at that trap site, followed by a 1 or a 0 for each species of moth (see Table 1). The data used consisted of 256 traps (see Figure 2) and 606 different moth species.

Table 1. An excerpt from the moth dataset illustrating the data format used.

Trap_ID	Elevation	Slope	Aspect	Vegetation Type	Acerra normalis	Behrensia conchiformis	Cerastis enigmatica
3C	4754	44.44786835	225.96533203	OPEN FOREST	1	1	1
3G	4685	27.00290489	171.19337463	OPEN FOREST	0	0	0
3B	4521	44.94692993	124.53629303	SHRUB/VERY OPEN FOREST	1	0	1
3D	4639	55.5788002	166.04974365	SHRUB/VERY OPEN FOREST	0	0	1

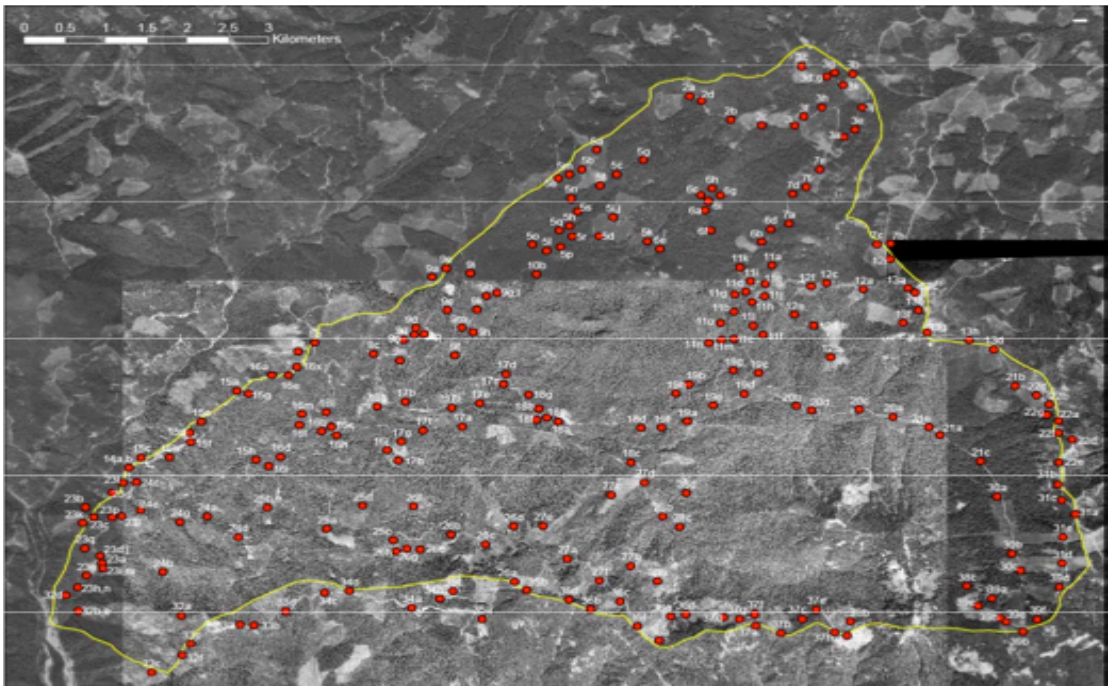


Figure 2. A map of the H. J. Andrews Experimental Forest (outlined in yellow) with the 256 trap sites marked by red dots.
Note: The various "holes" in the data are due to low accessibility by lack of roads, which made it unfeasible for Dr. Jeff Miller and his team to trap in those areas.

**Image provided by Dr. Tom Dietterich.*

After compiling the data into the described format used for modeling, it was split into training and test sets. This was done by sampling the data set without replacement until a division was found with approximately equal distributions of species. The original data contains 32,352 individual moth records, however, it is important to note that, more than half of the included species occurred fewer than 18 times over the course of the entire trapping period, while one sixth of the species account for over half of the recorded moth occurrences. The low resolution of the majority of the moth data was one motivation for attempting multiple response modeling. As, we hypothesized that overall habitat trends would influence the accuracy of predictions for species for which there is comparatively little known (i.e. almost all present or all absent) if there was a trend of covariance between species. For the purposes of this paper, the Northwestern North American moth dataset will be referred to as "moth data."

2.2. Experimental Design

The four different algorithms elastic-net logistic regression, single-response decision trees, multiple-response decision trees, and single hidden-layer feedforward neural networks were implemented on the plant and moth datasets. For all algorithms, a common testing procedure was used, consisting of parameterization and testing steps. The parameterization step may be further subdivided into trial and validation steps. The data is first split into two halves: parameterization and test sets. And the parameterization set is further divided into halves of train and validation sets.

For the trial step, models are created for every parameter combination using the 'train' subset of the parameterization set as training data. Each parameterized model version is then used to generate a prediction on the predictors from the 'validation' subset, and the prediction is then compared to the responses from 'validation'. If only one response was predicted, a single kappa score is generated. A kappa score [11] is calculated for each response, and this is recorded with the parameter values used in the model that generated it. The optimal parameters are then selected based on kappa scores. These optimal parameters determined in the trial step are then used in the testing step.

The tuned parameter sets thus created are used to judge the accuracy of the algorithm in the testing step. For each species, the parameterized model is used to make a prediction on covariates within the 'test' subset, which is compared to the response in the 'test' subset.

In addition, a suite of analysis tools was developed to analyze trends in algorithm success. These tools allowed us to identify the species for which a given algorithm predicted most successfully, compile the covariate values under which those species occurred, and calculate the average abundance of those species. They were used to search for trends influencing algorithm performance. These tools also provide easy generation of various visualizations of the data and include: the best kappa scores across algorithms color-coded by algorithm, the connection of corresponding kappa scores for individual species across algorithms, kappa scores per algorithm sorted or unsorted with or without mean lines.

2.3. Algorithm Implementations

2.3.a. Glmnet

The R package "glmnet" [6] relates real-valued or categorical predictor variables to a response vector with an elastic-net logistic regression. This is achieved by using training data to tune an array of ß coefficients within a logit function, each weighting the contribution of a predictor. Overfitting is controlled by a regularization term $(1-\alpha)/2||\beta||_2^2+\alpha||\beta||_1$, which penalizes high ß coefficients with a balance of L_1 and L_2 penalties. The term is itself weighted with a λ coefficient controlling the strength of regularization. L_1, also known as Manhattan distance or lasso penalty, equals the sum of the absolute values of the ß coefficients, and tends to zero less informative coefficients. L_2, known as the ridge regression penalty, equals the sum of the squares of the coefficients and thus penalizes coefficients greater than one more strictly. The elasticity of the elastic net regression derives from the alpha term controlling the balance between these two forms of penalty; not only is the severity of the penalty modulated, but the form as well [6].

Once an elastic net regression model is trained, its continuous output must be thresholded into discrete presence or absence values. This cutoff, along with α and λ, is optimized during the parameterization stage using a subset of the dataset reserved for training, which is further subdivided into training and validation subsets. Models trained on the train subset for twenty values of alpha between 0 and 1, each with 100 different lambda values. For each model, predictions are thresholded on twenty-one different cutoff values between 0 and 1, and the predictions compared to the validation subset. Cohen's Kappa [11] is calculated as a performance measure, and an optimal parameterization for each species is selected based on this value.

The set of coefficients found to most accurately predict the presence or absence of each species based on environmental covariates is then used to predict a reserved testing subset, and its success in predicting this heretofore-unseen subset is used as a measure of the effectiveness of the algorithm. A limitation of glmnet is its inability to learn a uniform response, whether totally absent or present. For this reason, kappa scores are reported as 'skipped' when such responses are encountered.

2.3.b. Rpart

The implementation of single response decision trees used in this study is called rpart, and was written by Terry M. Therneau and Beth Atkinson, ported to R by Brian Ripley [9]. It operates by creating a tree comprised of decision nodes that recursively split on predictor values, with more influential splits occurring closer to the root. At each node, an inequality decides which of its children a certain set of covariates will follow next, eventually terminating at a leaf node that contains a real-valued prediction of occurrence likelihood.

Overfitting in this case is controlled in two steps. In the first step, the tree complexity is scaled to control how closely to the training data the tree fits. Tree complexity is a measure of the informativeness of additional branches; beyond a certain level, branches contributing no additional refinement of prediction are omitted. Thus, in the first step, an initial tree is created

with unbounded complexity to fit the data as closely as possible, and then, in the second step, pruned to a range of complexity levels.

The real-valued predictions made at leaf nodes must then be thresholded into discrete presence/absence values. This cutoff is selected on each pruned tree by comparing the thresholded output against the validation prediction. This is similar to our elastic-net logistic regression implementation. As with the other algorithms, the parameterized decision trees are then used to predict an unseen test subset of the data, and the results are kept as a measure of predictive accuracy.

2.3.c. Mvpart

Mvpart [10] performs multivariate decision tree modeling, enabling a simultaneous prediction of all responses based on all sets of predictor covariates. This method is similar to the rpart method, moths may choose their habitats based on environmental characteristics beyond those that we consider, but those that prefer the same or similar habitats will predictably co-occur in those places satisfying their shared needs. We hoped that in addition to refining the accuracy of distribution prediction in general, the simultaneous method would also aid in modeling species with rare occurrence, which pose a serious problem to the single response algorithms.

Similar to the single response tree method, an initial tree is created with unbounded complexity and then pruned to a series of complexity limits. These trees are then used to predict from a training subset to a validation subset, which is used to optimize thresholds for the continuous prediction just as in rpart and glmnet. The model is then trained with the optimized parameterization on the training subset, and used to predict a reserved test subset.

2.3.d. Nnet

2.3.d.i. Overview

Neural networks are a multiple-response non-linear statistical data modeling algorithm originally inspired by human neural networks that are often used to model complex relationships. A complete and adequate discussion of neural networks is beyond the scope of this paper, but, for those interested, more information can be found in the book *Pattern Recognition and Neural Networks* written by B. D. Ripley [7]. The implementation of neural networks used was a single hidden-layer feedforward neural network algorithm via the nnet package of the R library [7], [8]. This model is a stochastic model because the starting values for the network weights and biases are chosen randomly, which means that the results from each run of nnet usually differ, although the overall kappa scores should hopefully be similar amongst runs [12]. R's nnet implementation fits a neural network to data using an iterative procedure [7], [8], [12]. Before nnet was implemented on a dataset, the dataset covariates were rescaled using the following formula:

$$x=(x-\text{mean}(x))/(\text{standard deviation}(x)). \quad (1)$$

Where x is a covariate and is replaced by the mean of its values across all sites subtracted from its value and divided by the standard deviation of its values across all sites. One may note that the mean value of (x-mean(x)) will be zero, but that it will have the same standard deviation, and thus by dividing by the standard deviation, it is ensured that the standard deviation is 1.

Rescaling the covariates in this way ensures that all covariates have approximately the same magnitudes and thus will be similarly weighted by the neural network algorithm [12]. In short, this helps the training algorithm run more efficiently.

2.3.d.ii. Parameterization

The nnet package provides the option to change over 17 different parameters, but for our purposes, only 6 different parameters were changed from their default values. These parameters were: 1) entropy (which when set to true uses entropy, the maximum conditional likelihood, fitting), 2) maxit (which restricts the maximum number of iterations of the numerical procedure), 3) size (which determines the number of nodes in the hidden layer), 4) maxNWts (the maximum allowable number of weights with a default value set to 1000), 5) rang (initial random weights on a range from [-rang, rang]) and 5) decay (the weight decay which can be used to penalize overfitting) [7], [12]. Of these parameters, the entropy parameter was set to true, maxNWts was calculated using a formula that will be described in greater detail below, and rang was set to 0.5 (an optimal value for rang suggested by the nnet package documentation). And the values for maxit, size, and decay, along with cutoff, were determined using a four-step parameterization procedure (described in greater detail below).

The total number of weights of each neural network is determined by the formula:

$$W=H*(P+1) + K*(H+1) \quad (2)$$

where W is the total number of weights, P is the number of input (predictor) variables, K is the number of output (response) variables, and H is the number of hidden units. Thus, the total number of weights of the neural network is dependent on the number of hidden units, the number of response variables, and the number of predictor variables. Because the maxNWts parameter must be at least as great as the total number of weights of each fitted neural network, the maximum number of weights is set to a value slightly larger than the number of weights (MaxNWts=W+10).

The parameterization procedure used for nnet consisted of a 4-step sequential process in which multiple values for each parameter being optimized (p) were tested in a given optimum range of n different values while the other parameters (x,y,z) were held constant for each different fitting of the nnet model (n times) to determine the best value for p using the maximum sum of kappa scores across all species as the means of comparison. Because the parameters for nnet are fairly dependent and could thus affect the overall performance of each parameter value, the parameters were optimized in an order that attempted to minimize the effects of the dependency of each parameter. Thus, the parameter that was believed to be the least dependent on other parameters was calculated first and the one that was believed to be the most dependent on the other parameters was calculated last. The best parameter value determined in each prior step was then used as a constant in each successive step. The optimal parameter values were determined in the following order: 1) size (number of hidden units), 2) maxit (the maximum number of iterations), 3) weight decay, and 4) cutoff thresholded for each species. Because the neural networks algorithm is a stochastic model, a seed used for generating random numbers was set to a value (s) and the model was then parameterized for s in each step. For each dataset, the parameterization values tested for each dataset are summarized in the following table (Table 2):

Table 2. Parameter values tested in each step

Parameter being Optimized	Range of values	Increment of values	Number of Values Tested
Size	10-100	10	10
Maxit	100-1000	100	10
Decay	10^{-8}-10^{-2} *	Factors of 10	7**
Cutoff	0-1	0.05	21

Note: This table quantifies the range of parameter values tried for each parameterization step with the amount each value was incremented by. For example, size values were 10-100 by increments of 10: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100.

** The range of values tested for decay were actually from 10^{-6} to 10^{-2} for the plants dataset, but were the range defined above for the moth dataset.*

*** The number of values tested for weight decay for the plant dataset was actually 5.*

2.3.d.iii. Independent Thresholding

As was the case with the other multivariate response algorithm (multiple-response decision trees), because the prediction values outputted by the neural networks model are expressed in a range of real-valued numbers from 0 to 1, instead of integer values of 0 and 1 to indicate presence or absence, the values are thresholded by cutoff values ranging from 0 to 1. These cutoff values were independently thresholded for each species by maximum kappa score across cutoff values to obtain higher kappa scores for each species.

The method of determining the optimal size to be used in the test step was later changed to simply the maximum size value that could be handled by the computer (as each successive size value is more computationally expensive and requires more computer memory). This parameter value, as well as the other parameter values determined in the original parameterization process, was then used for the testing step (see Table 3 for exact values).

Table 3. Optimal Parameters Determined for each Dataset

Moth Dataset				*Plant Dataset*			
Seed	Size	Maxit	Decay	Seed	Size	Maxit	Decay
100	29	1000	10^{-6}	1000	100	400	10^{-2}

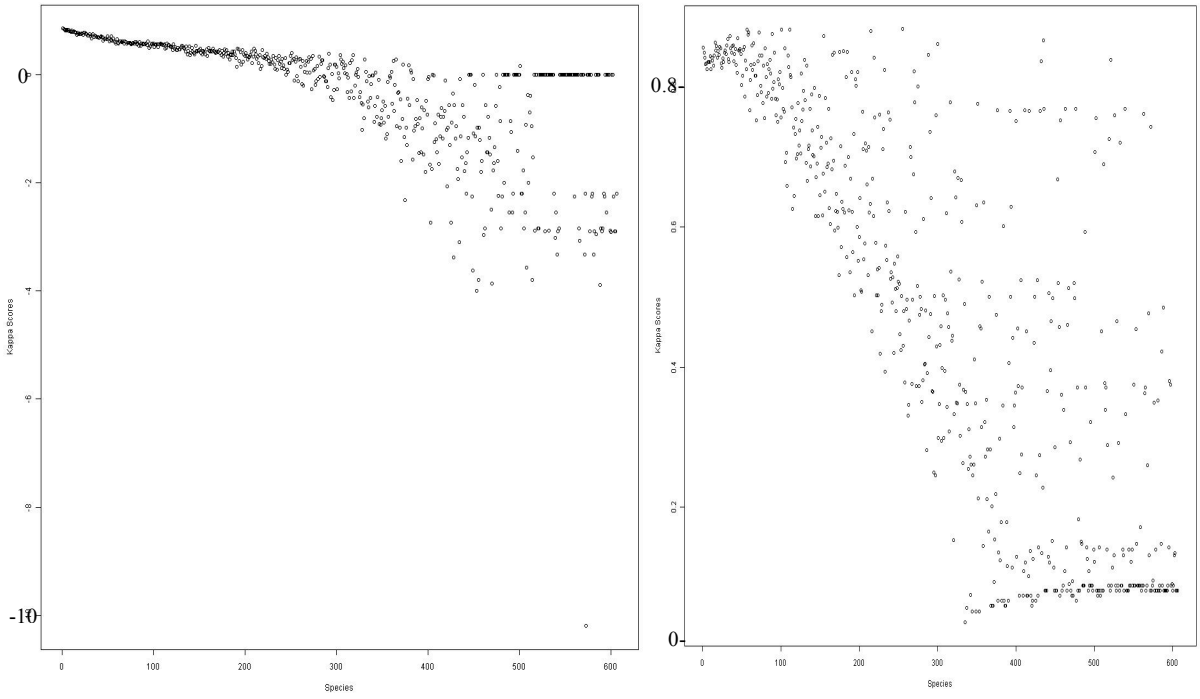
Note: This table shows the optimal parameter values determined for each dataset in the parameterization procedure that were later used for the testing step.

2.3.d.iv. Early Stopping

There are 3 general ways of controlling overfitting in neural networks: 1) network size (number of hidden units), 2) number of iterations of training and 3) weight decay. Of these, the first method is mostly avoided, the second method is the most practical, and the third method is theoretically the best method. For the purposes of this study, the second method (using the number of iterations of training) was implemented. In order to use early stopping for nnet using the number of iterations of training, the best iteration for each species (k) was chosen for k using the maximum kappa score (and optimal threshold) for each k. This was done by capturing the results of each iteration and then feeding those learned weights back into nnet as the initial values of the weights for the next iteration for 1000 iterations. In this way, we could train for 1000 iterations with the weight vector that was computed after each iteration and thus choose an

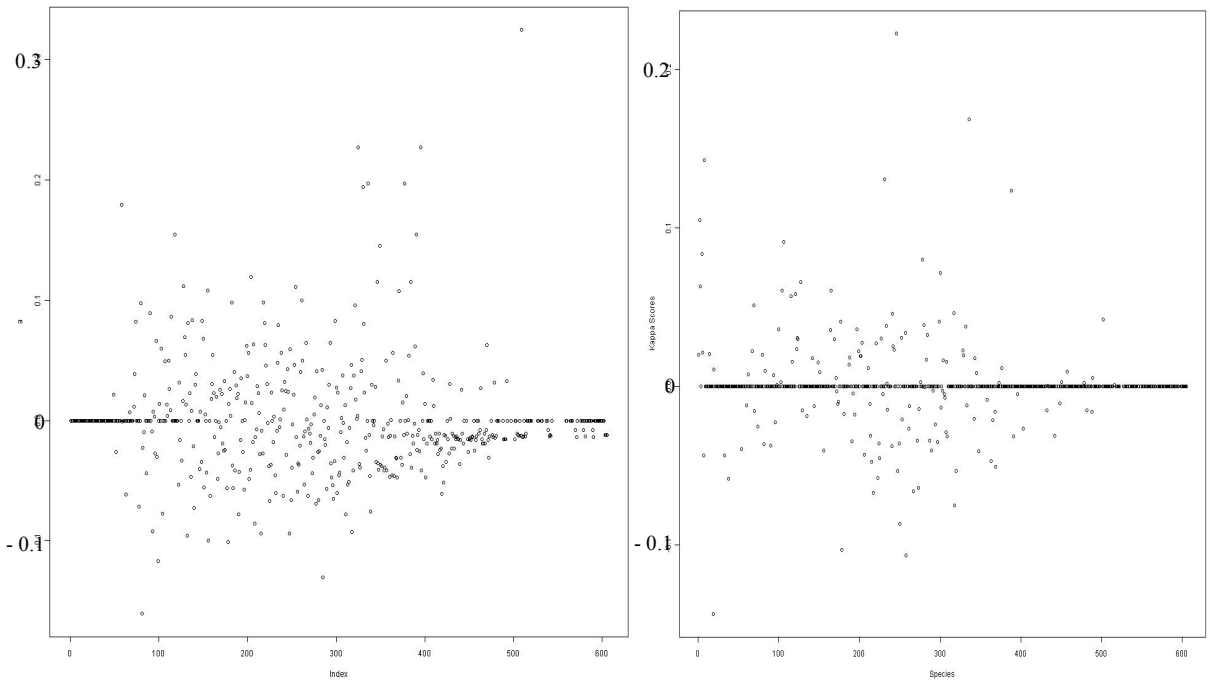
optimal stopping point (maximum iterations) without having to rerun nnet every time. In other words, each iteration was predicted for on the holdout (validation set) data and the iteration that gave the best performance (using maximum kappa score) was chosen for each species. Unfortunately, because one hundred iterations of this method took about one day to complete on the equipment being used (for a total of 10 days for one thousand iterations), it was not feasible to implement this computationally expensive method fully due to time constraints. However, this method was run for approximately 100 iterations (112 for the moth dataset) and preliminary results indicated much higher kappa scores using this method as compared to not implementing early stopping (Figure 3). This early stopping method is loosely based on Rich Caruana's multitask learning method [13].

As can be seen in the figures below (Figures 3 and 4), although the parameterization step suggested that early stopping would improve the accuracy of predictions, it is not so clear whether early stopping really improved the accuracy of predictions in the final testing step. However, since early stopping was not fully implemented due to time constraints (only run for 112 iterations out of 1000 on the moth dataset), it may well be that fully implementing early stopping would improve the accuracy of the neural networks algorithm. Due to time limitations, the final implementation of neural networks used for comparison purposes in the results section of this paper, did not reflect the use of early stopping as a means of overfitting control.



(a) (b)

Figure 3. Plot of kappa scores for moth data parameterization step (a) without and (b) with early stopping. The x-axis is the species in order of most to least abundant while the y-axis is the kappa scores.
Note: As can be seen on this graph, there were quite a few negative kappa scores when early stopping wasn't implemented for nnet, but only positive kappa scores when implemented with early stopping.



(a) (b)

Figure 4. Plot of kappa scores for moth data testing step (a) without and (b) with early stopping. The x-axis is the species in order of most to least abundant while the y-axis is the kappa scores.

The testing step was implemented for the moth dataset for neural networks and double-cross validation was implemented as well. Unfortunately, due to time constraints, the testing step was not implemented for the plants dataset for neural networks (although the script for this has been written); however, optimal parameters were determined for the plant dataset.

2.4. Prediction Grid for Moth Data

A 13-kilometer by 13-kilometer grid that covered the entire H. J. Andrews Experimental Forest was created and divided into squares of 100 meters by 100 meters (see Figure 5). Each of 16,900 coordinate points was matched up with corresponding covariate values of slope, aspect, elevation, and vegetation type. This dataset can be used to predict moth species occurrence for the entire H. J. Andrews Experimental Forest. Due to time constraints, a prediction for the entire H. J. Andrews Experimental Forest was not executed, however, we believe that this dataset may easily be used in future studies for this purpose.

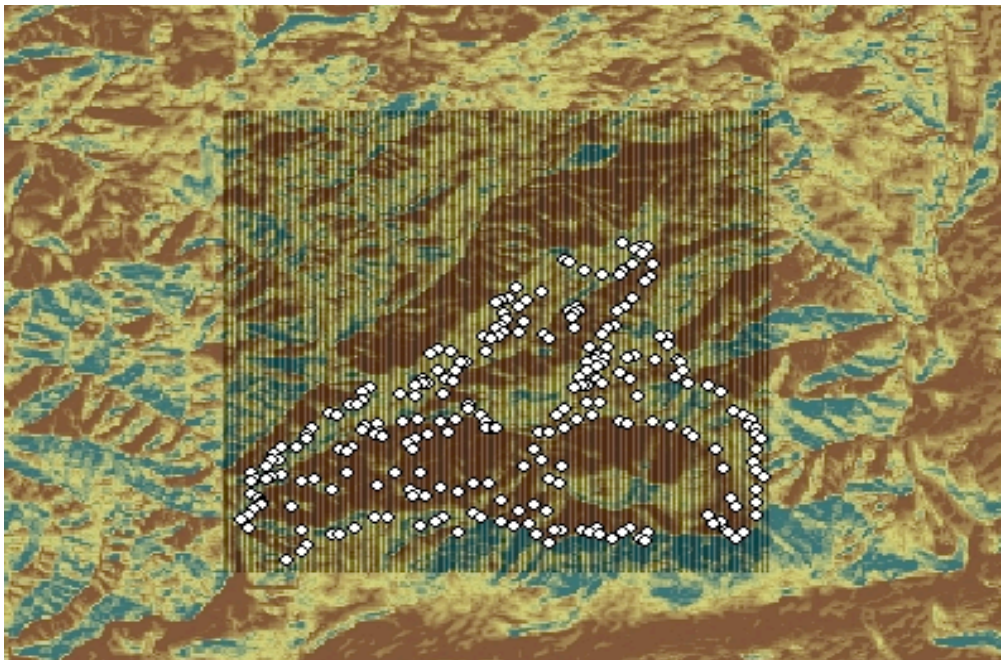


Figure 5. A 169 square kilometer grid layer (in black) split into 10,000 square meter squares overlaid on top of the H. J. Andrews Experimental Forest. The white dots are the trap sites in the data. This image was created using ArcGIS software.

3. RESULTS

Our algorithms were much less successful at accurately predicting moth occurrence than Australian plant species occurrence. Whereas predicting Australian plant occurrence yielded average kappa scores upwards of 0.2 with maxima well above 0.4, the mean and mode of moth prediction kappa scores hovered very near 0 (see Figures 6 and 7). Auxiliary-task modeling, using a multiple response model to predict each species individually, did not fare much better, with a mean kappa score of 0.003 (see Table 4).

Table 4. Average kappa values for each algorithm for both datasets

Dataset	Glmnet	Rpart	Mvpart	Nnet	Auxiliary Task mvpart
Moth Data	0.001	0	0.001	0	0.003
Plant Data	0.26	0.183	0.152	Not implemented	Not implemented

We surmise that this is due to the quality of the moth data; as described in the data section, the moth data are not entirely amenable to machine learning. As, half of the total individual moth records occur in the top 16% most abundance species, and more than 50% of the species occur fewer than 18 times over the trapping period of 23 years. Additionally, moths are highly mobile

organisms, and could reasonably be expected to stray beyond their optimal habitat, creating spurious data points. Plants have the advantage, from the modeling standpoint, of being sessile organisms and therefore much more reliably found in consistent environmental surroundings.

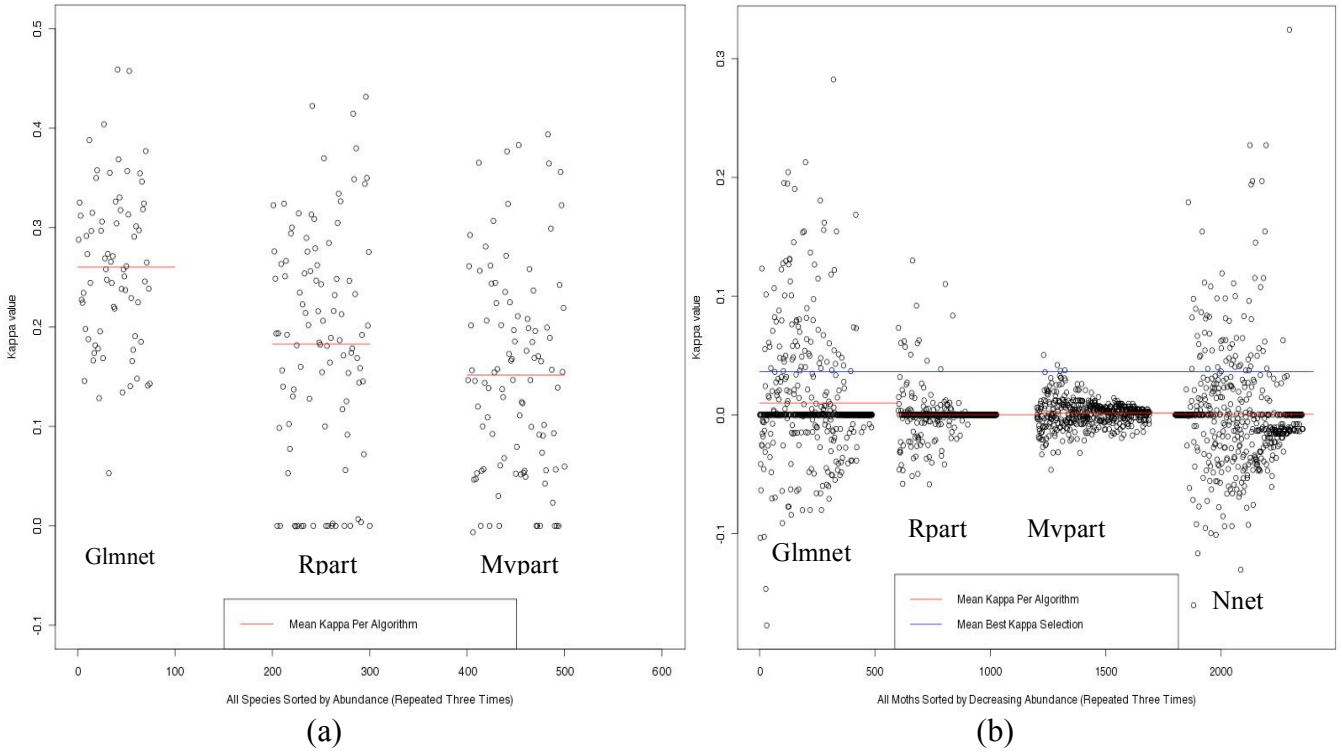


Figure 6. Results for each algorithm for the (a) plant data and (b) moth data.

Table 5. Mean and standard deviation of abundances of moth species that were predicted on most successfully by each algorithm

	Glmnet	Rpart	Mvpart	Nnet
Mean	0.145	0.165	0.113	0.103
Standard deviation	0.016	0.022	0.011	0.008

Note: These values were calculated by assembling the list of species for which each algorithm predicted most successfully and then by calculating the mean and standard deviation of the moth species abundances within each list for each algorithm.

Because the moth data were noisier and less broadly sampled than the plant data, they offered an interesting opportunity for prediction refinement in the face of poor model tuning. Each algorithm was most successful on a specific range of moth species abundance and covariate values, suggesting that certain moth species are best suited for specific modeling techniques (Table 5).

As can be seen in Figure 7, the four algorithms performed most accurately on species with abundance of 13.15% on average, nearest the third quartile of abundances across all species of 14.06%. This indicates that above the third abundance quartile species occurrence is too widespread to be informative, and below it they are too sparse to provide an informative occurrence pattern.

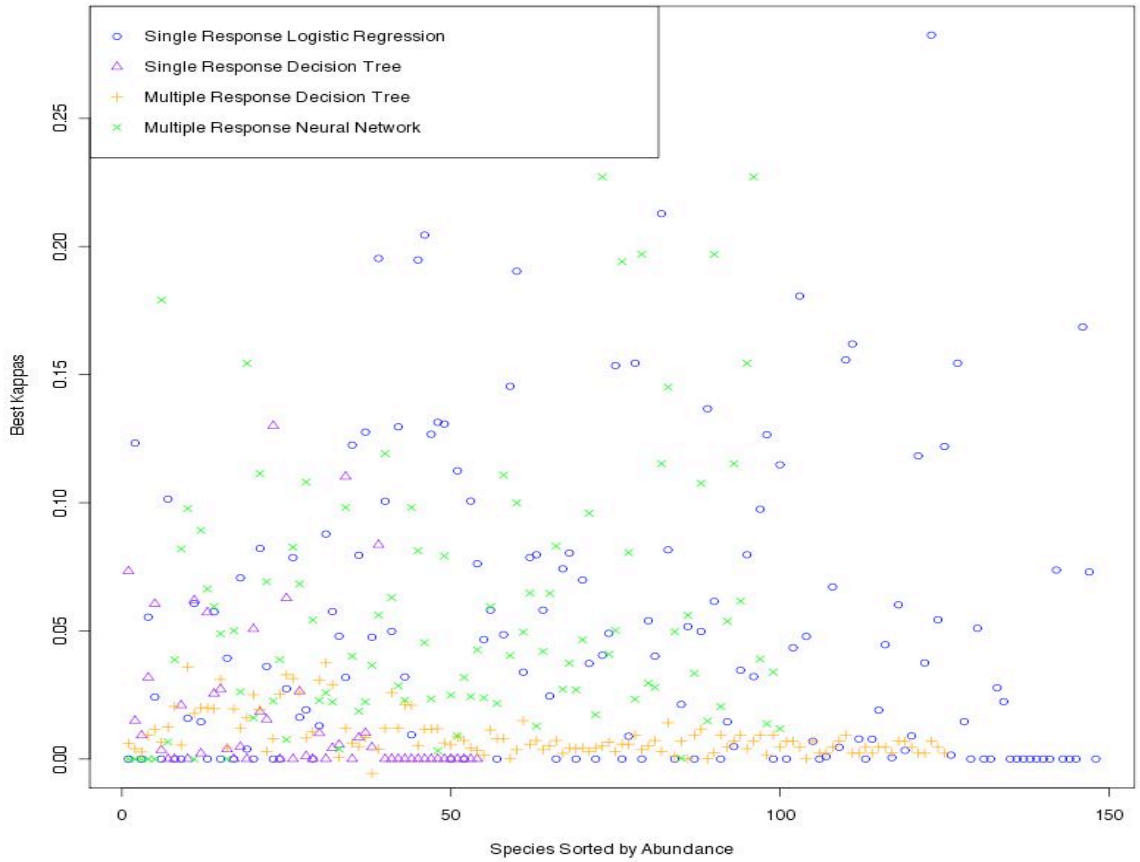


Figure 7. This graph shows which algorithm made the most accurate prediction for each species for the moth dataset. Species are color-coded by winning algorithm, and sorted across the x-axis by increasing abundance. Abundance is calculated by averaging the response vector for each species, giving the percent of sample sites at which that species occurred.

The availability, abundance distribution, and quality of alternative kappa scores between predictions were visualized on a plot of all kappas for the four algorithms (Figure 8). Improved alternatives were found for the majority of species, indicating that the ranges of "specialty" for each algorithm cover the majority of the overall species abundance range. For illustrative purposes, the following diagram plots only the largest kappa improvements. We observe that although single response logistic regression fails for species with very high abundance, those species are accurately predicted by the two tree methods. Similarly, the neural networks algorithm predicts a group of low-abundance species predicted poorly by the first three methods very successfully.

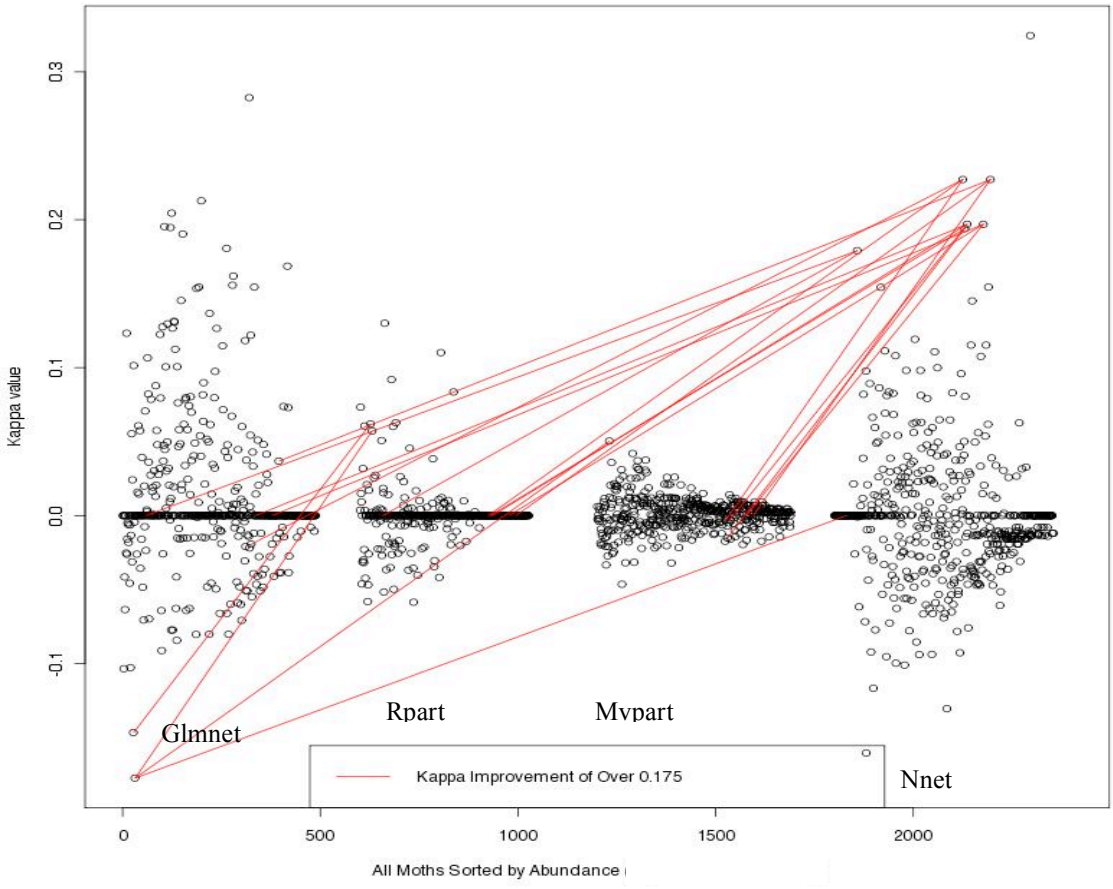


Figure 8. Best kappa alternatives between algorithms for the moth data. The red lines on this plot indicate kappa improvement of over 0.175 (a fairly significant increase) between algorithms.

4. DISCUSSION

The marked disparity between predictive performance on plants and moths indicates several things. First, the predictive models do produce high quality predictions, given data amenable to extracting correlations between predictor and response. The plant dataset includes many more environmental covariates than the moth dataset, which gives the model more opportunity to find a determinant correlation between covariate and response. In addition, the sample size of the plant dataset is almost 60 times that of the moth dataset (15,328 sample sites vs. 256 trap sites), which reduces the impact of misinformative occurrences. With large sample size and fidelity, the predictive models were able to infer relationships between covariates and response with a high degree of success.

From the differential success on the two datasets, we may also infer that, the moth dataset poses a considerable challenge from a modeling standpoint. Moths, being very mobile organisms, may not neatly occur only in traps set in their preferred habitat. For moths with very high abundances, this makes prediction of preferred habitat very challenging, especially on an

individual basis. For moths with very low abundance, it is difficult to distinguish between informative and misinformative occurrences, and with few records to consult; a prediction is difficult at best. The distribution of the data in the moths dataset compounds these problems, focusing most of the dataset into a tiny portion of very saturated moth occurrences, and leaving the majority of species without much presence in the data. In addition, because the environmental covariates for the moth dataset were determined from a static digital elevation model, they likely did not reflect the habitat preferences of moths. Since moths are mobile organisms, they are sensitive to dynamic factors in their environments, something which elevation, slope, aspect, and vegetation type did not account for. Moreover, because slope is measured in continuous units (0-360 degrees) for which 0° and 360° are the same, the fact that the models treated this covariate as non-continuous also affected the accuracy of the models' predictions on the moth dataset. Thus, in future studies, we recommend that a Euclidian transformation be performed on the slope and aspect covariates to improve the accuracy of the species distribution models.

We have attempted to exploit the "areas of expertise" of the various algorithms in order to extract a more successful prediction. Each of the four algorithms implemented in this study has a slightly differing range of abundance and covariate proportions on which it predicts more successfully than the other methods; by exploiting this; it is possible to create a conglomerate prediction model that uses the best model for each species. This hybrid method would produce a prediction many times better than individual methods, although still not as good as the individual methods on a dataset with greater data quality. The four methods may be integrated under a "controller" algorithm, which learns the "areas of expertise" of each of the four, and then uses it to predict for those species for which it is suited. Unfortunately, we must leave the "conglomerate prediction" implementation to subsequent research.

5. CONCLUSION

In order for species distribution modeling to be used as a tool by conservation biologists, the accuracy of these models is of chief concern. After having implemented four different algorithms on two types of datasets (one with mobile organisms and less data and one with sessile organisms and more data) for presence/absence, we determined that informative covariates are important in order to have accurate predictions. In fact, we believe that more species-specific covariates should be included when modeling the occurrence of mobile organisms. Although it was unclear whether modeling species simultaneously resulted in a significant improvement from the traditional modeling method, we observed that different algorithms have differing success at modeling species with different abundances or covariate values. For this reason, we believe that an integrated predictive model that selects one of the four algorithms based on observed performance specialty has great potential as a solution to noisy data. We also discovered that independently thresholding for different cutoff value for each species proves more successful than using the same cutoff value for all species. Furthermore, we developed a dataset of covariate values for the entire H. J. Andrews Experimental Forest in 50 meters by 50 meters squares for a surface area of 169 square kilometers that could be used in subsequent studies for moth species distribution modeling for the entire forest.

We believe that a number of transformations could be performed on the moth dataset in order to improve prediction results. First, a Euclidian transformation should be done on the aspect and slope covariates of the moth dataset to improve the accuracy of predictions using this data. Second, the moth species could be grouped based on common feeding preferences or habitat preferences. Third, the vegetation type covariate could be transformed into continuous values instead of the current discrete values to facilitate the algorithms' use of this covariate for predictions. Using more covariates for the moth dataset may also improve the accuracy of predictions on this dataset.

If these changes are implemented in future studies, we believe that species occurrence predictions will be more accurate on the moth dataset and thus that we may be closer to being able to accurately predict moth species occurrence in the H. J. Andrews Experimental Forest. In addition, subsequent researchers will gain further insight on how to create more accurate species distribution models for mobile organisms. As a result, the increasing accuracy of these species distribution models for different types of data could enable conservation biologists to more reliably employ species distribution models for their practices.

ACKNOWLEDGEMENTS

We would like to thank the following people and organization: Our advisors Dr. Tom Dietterich and Dr. Weng-Keen Wong, two computer science professors at Oregon State University, for their extensive help and support throughout our research project; Oregon State University computer science post-doc Rebecca Hutchinson, who took time out of her schedule to attend our meetings and provided us with support; Oregon State University ecological engineering professor Dr. Desiree Tullos for having served as the director for the Eco-Informatics Summer Institute; the director of the Eco-Informatics program at Oregon State University Dr. Julia Jones for her support and help throughout our research experience; Oregon State University PhD candidate Steven Highland for having served as the graduate student teacher assistant in our program and for his assistance with acquiring the moth data covariates using ArcGIS; the Oregon State University computer science PhD candidates working on our project Arwen Lettkeman and Paul Wilkins for their support and continued research on this project; and Dr. Jeff Miller and the Arthur Rylah Institute of Melbourne, Australia for providing us with our data.

The NSF provided support and funding for this project through the Eco-Informatics Summer Institute Research Experience for Undergraduates, at which this research was conducted.

REFERENCES

[1] A. Guisan and N. E. Zimmermann, "Predictive habitat distribution models in ecology," *Ecological Modeling,* vol. 135, nos. 2-3, pp. 147–186, 5 December 2000. [Online]. Available: ScienceDirect, http://www.sciencedirect.com. [Accessed August 18, 2009].

[2] M.P. Austin, "Spatial prediction of species distribution: an interface between ecological theory and statistical modeling," *Ecological Modeling,* vol. 157, nos. 2-3, pp. 101-118, pp. 101-118, 30 November 2002. [Online]. Available: ScienceDirect, http://www.sciencedirect.com. [Accessed August 18, 2009].

[3] W. Thuiller, "Patterns and uncertainties of species' range shifts under climate change," *Global Change Biology*, vol. 10, no. 12, pp. 2020-2027, 2004. [Online], Available: Wiley InterScience, http://www.interscience.wiley.com. [Accessed July 10, 2009].

[4] M. White, Arthur Rylah Institute, Personal Communication, 2008.

[5] J. Miller, 2005. "Spatial and temporal distribution and abundance of moths in the Andrews Experimental Forest." H. J. Andrews Experimental Forest. Available: Forest Science Data Bank, http://andrewsforest.oregonstate.edu/data/abstract.cfm?dbcode=SA015. [Accessed August 18, 2009].

[6] J. Friedman, T. Hastie, and R. Tibshirani, "*Regularization Paths for Generalized Linear Models via Coordinate Descent,"* 2008. Available: http://www.stat.stanford.edu/ ~hastie/Papers/glmnet.pdf. [Accessed August 18, 2009].

[7] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press, 1996.

[8] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S.* 4th ed. New York: Springer, 2002.

[9] L. Brieman, J. Friedman, R. A. Olshen, C. J. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth International Group, 1984.

[10] G. De'ath. "Multivariate Regression Trees: A New Technique for Constrained Classification Analysis," *Ecology,* vol. 83, no. 4, pp. 1103-1117, 2002.

[11] Cohen, Jacob. "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement,* vol. 20, no.1, pp. 37–46, 1960.

[12] K. Velten. *Mathematical Modeling and Simulation: Introduction for Scientists and Engineers.* Weinheim: Wiley-VCH, 2009. pp. 87-99.

[13] R. Caruana, "Multitask Learning," *Machine Learning*, v. 28, no. 1, pp. 41-75, July 1997.