Calculating the Density of Georgia Forests

Maniketh Aley[1], **Jared Dunham**[2], **and Samuel St John**[3]

[1, 2, 3]Central Michigan University, Department of Computer Science, Mt. Pleasant, MI, USA
[1]manik1a@cmich.edu
[2]dunha2j@cmich.edu
[3]stjoh1sr@cmich.edu

ABSTRACT

Managing forests is an important task because over 800 million acres of the United States contain various forest environments[1]. A key factor in managing these forests is knowing their tree density. This metric can help predict whether a forest should be thinned or is remaining healthy. Additionally, knowing the tree density in a forest can help to predict how climate change is affecting the environment. The two reasons why we believe this is a good problem for machine learning to solve are that populations of trees are slower to change than populations of other living things, and there are many variables at play that can be missed by human computation. Our main sources of data are the US Forest Service's Inventory and Analysis Program and the Climate Research United Gridded Time Series data set[2]-[3]. Our approach is different from other experiments because we use factors outside of the knowledge about the trees themselves, such as temperature and precipitation data. Throughout this experiment, we developed several high-performing deep learning models using recurrent neural networks to address this task. Between the six models that were tested, the GRU RNN model with dropout layers achieved the best performance with an average MAE of 0.02540 and MAPE of 31.98 on our test dataset (averaged over training 10 models).

Introduction

As a parameter of biological health, tree density is important in assessing the ecology of an area. This is especially relevant in light of environmental climate change. Additionally, overly dense forests can increase the chances of "pest insect infestations, wildfires, and other disturbances," and out of the total forest area in the US, the percentage of dense forests has risen from one to six percent between 1999 to 2020[4]-[5]. Since tree canopy coverage determines how much light reaches the forest floor and lower-height plants, tree density can be used to predict several biological characteristics of a forest. These characteristics include biodiversity, nutrient prevalence, and species composition[6]. Therefore, predicting the tree density of an area while taking into account environmental parameter changes can improve our understanding of the ecology of a forest. Furthermore, tree density is closely related to carbon sequestration, and thus is an important metric for any potential solutions to excess atmospheric carbon[7]. If predicted over large scales, tree density could act as a progress marker for how well tree-based carbon solutions are implemented. Monitoring the health of forests, projecting the consequences of climate change, and improving forest management methods all depend on having accurate predictions of the forest biomass and tree density.

Machine learning has been utilized to estimate tree parameters in several past studies. In many cases, these studies have relied on satellite imagery to predict the tree density of forests. For instance, Lui et al. developed a deep convolutional neural network called TreeCountNet in order to predict the total count of trees over a map of remote-sensing images of China. TreeCountNet uses a hybrid loss scheme to reach a promising start to estimating density, though training on a wider data set would be needed for large-scale applicability[8]. Neural Networks were also used with tree predictions in Brazil, which estimated the volume and growth rate of Eucalyptus trees at plantations. Using Adaptive Neuro-Fuzzy Inference Systems (ANFIS), and Random Forests (RF), they were quite successful in their predictions[9]. Outside of common machine learning techniques, researchers have also calculated the growth of tree populations using a bayesian framework and computations without any programming[9].

Additionally, a universal data-driven model has been used to estimate forest gross primary productivity and net ecosystem exchange globally using deep learning networks and seven ecological and climatic parameters as inputs[10]. Artificial Intelligence, Big data, and Remote sensing technology have advanced significantly in recent years, and the availability of high-resolution data sets has made it possible to collect comprehensive data on forest ecosystems, enabling researchers to train machine learning models on massive volumes of data. The created model integrates historical data obtained from permanent sample plots with information on tree growth produced by a process-based model[11]-[12]. A case study states Nova Scotia is expected to experience climate change with an increase in temperature and growing degree days that will impact the forests covering most of the province. To predict future forest dynamics under a changing climate, a practical and simple growth and yield model that can be easily operated by forest managers was developed[12]. Another case study by Tanaka and Nishii proposes a machine learning approach that uses SVM and SVR to predict forest coverage. Additionally, the approach classifies areas into three categories based on their deforestation status: completely-deforested, fully-forest-covered, and partly-deforested. By predicting the forest coverage ratio and identifying areas that have been deforested, the proposed approach can help in monitoring and managing forest density and growth[13].

While these existing studies are set in a limited geographical area, they still represent a potential launching point for further investigation into estimating tree density. As we've seen, prior experiments have used image-based approaches[8, 14], while another focused on using the tree basal area and age to predict the growth rate or amount of trees[15]. This study intends to take into account numeric external variables that may affect tree density, which allows for a wide range of applicability of the final mode. The external environmental features that we implemented into the training dataset were average latitude, longitude, below ground carbon, above ground carbon, below ground dry biomass, precipitation, temperature, diurnal temperature range, and water vapor. Our goal for this project was to predict the average tree net cubic foot volume per acre of each county in Georgia. We did this for each month between the years 1901 to 2021, and believe that there was some success in our methods despite not being able to compare our results to any similar applications.

Methods

Data

The data used for this project was taken from several sources. The main data set in the study, which originated from the United States Forest Service Forest Inventory and Analysis National Program, is hosted by Google[2]. The United States Forest Service Inventory and Analysis program is a long-term federal program of data collecting on the forests of the United States, including biological, geographic, and environmental information[16]. Supplemental county geographic information came from the U.S. Environmental Protection Agency[17]. Finally, climate data used in this study originated from the Climate Research United Gridded Time Series (CRU TS) data set from the University of East Anglia[18]. The fourth version of the CRU TS data consists of grid-based information for precipitation, temperature, and water vapor from 1901-2021 at monthly intervals[3]. These three data sources were combined to create or otherwise process data intended to be used during the model creation and evaluation stages.

To ensure a maintainable scope for the study, the decision was made to focus on a limited geographic area encompassing the state of Georgia. Georgia was chosen because it had the highest amount of sampled trees per square mile when compared to other states. From the Forest Service Inventory and Analysis data, records were filtered to include only measurements from Georgia and were grouped by county, year of measurement, and month of measurement. Additionally, the features selected from the data set were limited to county code, measurement year, measurement month, total tree sample count, total net cubic foot volume, average latitude and longitude (on a county-by-county basis), average elevation, average tree diameter, average tree height, average seedlings per acre, average tree stocking, average below-ground carbon, average above-ground carbon, and average below-ground biomass. This made for 15 features total originating from the U.S. Forest Service. During processing, any records that were missing any of these features were dropped from the data. Afterward, county acreage was added from the Environmental Protection Agency data set, totaling 16 features. Our target variable, average tree net cubic foot volume per acre, was generated by dividing the total net cubic foot volume by the county acreage.

Each county was then manually assigned to a grid box corresponding to the 0.5 degrees by 0.5-degree grid schema used by the CRU TS data set, which added grid center point latitude and longitude features, which brought the total number of features to 17. A Python-based algorithm was used to crawl through web-hosted text file grid-by-grid and data type by data type. This allowed for the inclusion of average temperature, total precipitation, average diurnal temperature range, and water vapor data on a monthly basis. Because this data includes consistent monthly records over the past century, there was no need to handle missing climate information. With the inclusion of the climate data, the data frame contained 21 features. No sort of encoding was necessary for these features, since the intention was to drop out the only non-numeric feature (county code) before proceeding to model training.

After all the data was merged together, the next step was working on segmenting the entries. Before splitting, the data was ordered for each county by the year and month of the sampling. Then, the total data set was split into 60% of the entries being used for training, 20% for validation, and 20% for testing. This was done by doing a 60-20-20 split for each county individually, which allowed keeping the year and month of each measurement sequential per county. Additionally, the data was not shuffled because it was in a time series and would risk a temporal leak. Once split, the excess columns of each set of data were dropped to slim it down to 16 features, removing the following: 'total_net_cubicfoot_vol', 'measurement_year', 'measurement_month', 'county_code', 'CRUT_Grid_Center_Lat', and 'CRUT_Grid_Center_Long'. Lastly, the data was normalized for each split by using Z-Score calculated using scaling from the training data.

Table 1. Features Used in Final Models

Feature	Data Type	Origin
Tree sample count	numeric	USFS FIA (summed by county, year, month)
Average latitude	numeric	USFS FIA (averaged by county, year, month)
Average longitude	numeric	USFS FIA (averaged by county, year, month)
Average elevation	numeric	USFS FIA (averaged by county, year, month)
Average tree diameter	numeric	USFS FIA (averaged by county, year, month)
Average tree height	numeric	USFS FIA (averaged by county, year, month)
Average seedlings per acre	numeric	USFS FIA (averaged by county, year, month)
Average tree stocking	numeric	USFS FIA (averaged by county, year, month)
Average below-ground carbon	numeric	USFS FIA (averaged by county, year, month)
Average above-ground carbon	numeric	USFS FIA (averaged by county, year, month)
Average below-ground dry biomass	numeric	USFS FIA (averaged by county, year, month)
County land area acres	numeric	US EPA (by county)
Average tree net cubic-foot vol per acre*	numeric	USFS FIA (averaged by county, year, month)
Precipitation	numeric	CRU (by month and year)
Temperature	numeric	CRU (by month and year)
Diurnal temperature range	numeric	CRU (by month and year)
Water vapor	numeric	CRU (by month and year)

* Average tree net cubic-foot vol per acre was the target variable.

Development Environment

The development environment for this project was Google Colaboratory, primarily using Keras[19]. At times, computations were performed using a graphical processing unit (GPU) virtually allocated through Google Colaboratory's runtime options.

Model Architecture

Once the data was prepared and separated by the features and targets, the train, validation, and test data sets were ready. Then, six models were implemented, including a Simple Deep NN model, a Simple RNN model, an LSTM RNN model, two GRU RNN models with differing settings, and a Bidirectional LSTM RNN model with k-fold validation. For the baseline, the Simple Deep NN model was chosen, under the assumption that all other models would perform better and because it would be challenging to create a naive predictor for this particular problem.

The layers and amount of nodes that were utilized changed depending on the model, as they trained differently from each other. However, for all models, ReLu was used as the activation function for all layers minus the output, which was the linear activation function. ReLu outperformed tanh, which is a popular activation function for RNN layers. It was found that Tanh allowed the model to train for a longer period of time, but was not as accurate. For the optimizer, rmsprop was selected, which performed better than the Adam and Adagrad optimizers. In terms of metrics, mean absolute error (MAE) and mean squared error (MSE) were used as the primary performance measurements. Relative root mean squared error (RRMSE) was also added because Silva et al. used it as a metric in previous works[15]. This was under the assumption that it may allow for comparing results with previous discoveries.

When optimizing the models for performance, the models with RNN layers would commonly train too fast and not result in great performance due to how fast the loss of the models plateaued. This issue was tackled in different ways depending on the model. Layer Normalization was added to the LSTM and one of the GRU models, Dropout was added to the other GRU model, and the Bidirectional model used cross-fold validation. When comparing the two GRU models, Layer Normalization generally performed better than Dropout. A batch size of 40 with 40 epochs was typically used. Additionally, a callback was utilized when the model stopped performing well. The RNN models were generally inconsistent with the number of epochs they could run effectively so the callbacks were useful in this regard. Lastly, for the k-fold validation, 10-fold was the most optimal and surprisingly allowed the models to train for longer periods of time than 5-folds.

Results

For the model performance on the validation data, the model with the lowest MAE was the GRU RNN model with dropout, with a value of 0.04278. The next lowest MAE values were the LSTM Bidirectional RNN model, at 0.04847, and the GRU RNN model with layer normalization, at 0.05003. For MSE, the three models with the lowest values were in a similar order - the GRU RNN model with dropout at 0.00709, followed by the GRU RNN model with layer normalization at 0.00785, followed by the LSTM RNN model at 0.00865. Unexpectedly, the Deep NN model performed nearly in line with the recurrent networks. In general, however, the recurrent networks generally performed slightly better.

Table 2. Model Comparison over Validation Data

Architecture	MSE	MAE	MAPE	RRMSE
Deep NN	0.01338	0.06117	42.93	0.03527
Simple RNN	0.00955	0.05610	36.93	0.01120
LSTM RNN	0.00865	0.05196	37.10	0.01048
GRU RNN (LayerNormalization)	0.00785	0.05003	35.85	0.01078
GRU RNN (Dropout)	0.00709	0.04278	31.3	0.00806
LSTM Bidirectional RNN (Fold Validation)	0.00895	0.04847	32.90	0.01920

*Metrics were averaged after training each model over ten iterations.

For the testing data metrics, the GRU RNN model with dropout performed the best in terms of mean absolute error (MAE) with a value of 0.02540, followed closely by the LSTM RNN model at 0.02970 and the GRU RNN model with layer normalization at 0.00282. In terms of mean squared error (MSE), the GRU RNN model with a dropout at 0.00162 edged out the LSTM RNN model with 0.00218. As we mentioned previously, we added the RRMSE to our metrics to compare with Silva et al[15]. and realized later that the values were likely going to be very different since we normalized our data beforehand. However, they may still be useful for future studies with a similar target prediction.

Table 3. Model Comparison over Testing Data

Architecture	MSE	MAE	RRMSE
Deep NN	0.00250	0.03260	0.14482
Simple RNN	0.00282	0.03504	0.04717
LSTM RNN	0.00218	0.02970	0.03421
GRU RNN (LayerNormalization)	0.00282	0.03386	0.03870
GRU RNN (Dropout)	0.00162	0.025407	0.02955
LSTM Bidirectional RNN (Fold Validation)	0.00443	0.03849	0.00497

*Metrics were averaged after training each model over ten iterations.

By most metrics, the GRU RNN model appears to perform the best on the training and testing data. The figure below provides an illustration of the validation and training loss over time for one iteration of training the model.

Discussion

After curating our dataset and training several models, there are several points to discuss about our approach to solving this problem. Regarding the advantages of our methods, some are based on how we created the dataset, while others are about how our approach is unique to this problem. First, this experiment can be used to predict tree mass density on an area level rather than individual trees. Second, it incorporates external data factors - weather (including precipitation, temperature, diurnal range, vapor pressure), elevation, and underground biomass, for example - whereas other models do not. Additionally, this allows our models to train with more features than other similar machine learning applications to this problem[15]. Finally, our data set is over a long period of time (approximately 50 years) and is relatively large, with 7999 entries in total. We believe that increasing the number of features in our dataset, and a large amount of data has significantly increased our when compared to other models.

In contrast, there were several disadvantages to our methods. While the field of machine learning moves quickly and cutting-edge approaches may lead to margin improvements in model performance, we elected not to incorporate any of these in our models. By using well-established machine learning techniques, we may have circumvented additional performance gains. Additionally, our approach to data cleaning may have limited the potential of the models. To handle entries with missing or null values, we chose to drop the records entirely. This shrunk our dataset significantly, which may have affected the final performances of our models. Furthermore, in estimating the density on a county level, we do not account for the distribution of

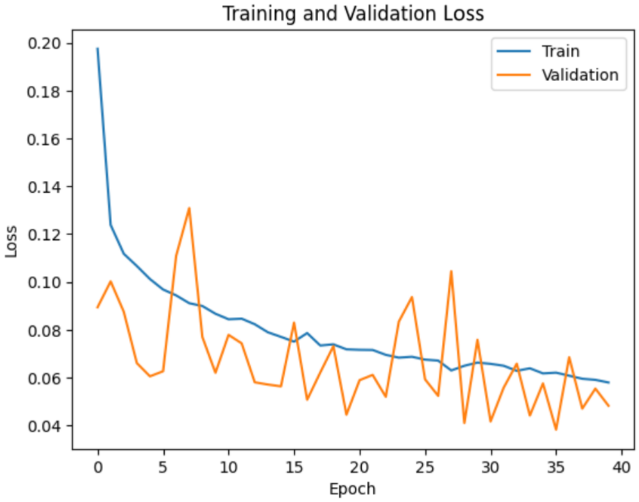

Figure 1. Training and validation loss over time for one iteration of the GRU RNN model.

density within the county. So, for example, a county may have its distribution of trees concentrated in a specific area of the county, which this model cannot account for. This prevents deeper analysis of density in smaller areas than at the county level.

Comparing our approach to other similar experiments can be somewhat difficult as there are few solutions to the problem we are attempting to solve. However, based on the articles we read, we believe that our solution is more complex. One reason why is that our models incorporate more features into the final calculation. Additionally, many other approaches have used manually designed mathematical models or object detection in images[8]-[9] to predict forest population or density, however, these solutions could likely be less mathematically complex than the operations done during deep learning.

When initially training our models, we had no real expectations of what the performance would be, this is due to not having other solutions to reasonably compare to. Additionally, we spent a large amount of time finding/referencing data from other databases to create our own and had no idea whether it would be successful. However, our final performance in this application to the problem was highly accurate, which was determined by the deep neural network model that we used as a baseline. Referencing Table 1, the Deep NN MAE on validation data was 0.06117, compared to the GRU RNN with Dropout layers, which earned an MAE of 0.04278. As mentioned before, we did not have similar applications that we could draw a baseline predictor from, so this classic deep Neural Network was our next best option.

Though our models performed very well on the curated dataset, there are a number of improvements that we believe could be implemented to further the accuracy of this approach. Utilizing smaller geographic granularity than counties would likely provide improved results. Additionally, using the same location measurements to determine the area that each entry covers (county) and the weather (0.5 x 0.5-degree grid in the CRU database), could make training more efficient. Another possible idea we had was to gather the weather information for each tree individually and then average that information when grouping the trees into one entry. Opposed to the method we are using right now which gets the average latitude and longitude of all the tree samples in an entry and then determines the weather information based on that latitude and longitude and the time of the measurements. Lastly, we also believe that researching and implementing more cutting-edge methodologies could have improved this experiment.

Conclusion

The original goal of this project was to create deep learning models to accurately predict the tree density via net cubic foot volume of tree mass per acre, specifically of counties in the state of Georgia. This paper outlines the results of the final models built for this purpose. The GRU RNN model with Dropout layers performed the best under most metrics across both validation and testing datasets with a test MAE of 0.02540 and MAPE of 31.98. Other models, such as the LSTM RNN and Bidirectional RNN, also performed well for the task. Future studies may consider the use of more sophisticated, cutting-edge machine learning techniques to tackle this issue. Additionally, if data availability improves for weather data, developing a future approach that takes into account climate information for smaller parcels of land could be helpful for extracting detailed predictions on the sub-county level. In the meantime, we believe that our approach has generated the highest-performing models for tree density on the county level in Georgia.

References

1. Oswalt, S. The state of the forest (2019).

2. Forest inventory analysis dataset.

3. Harris, I. A., Jones, P., Jones, P. & Lister, D. G. Version 4 of the CRU TS monthly high-resolution gridded multivariate climate dataset. *Sci. Data* **7**, DOI: 10.1038/s41597-020-0453-3 (2020).

4. McLeish, T. Forest density a growing concern: Spring 2022: Discoveries. *North. Woodlands* (2022).

5. Thinning the forest for the trees. *US For. Serv.* .

6. Paulson, A. K. *et al.* Understory plant diversity and composition across a postfire tree density gradient in a siberian arctic boreal forest. *Can. J. For. Res.* **51**, 720–731, DOI: 10.1139/cjfr-2020-0483 (2021). https://doi.org/10.1139/cjfr-2020-0483.

7. Rautiainen, A., Wernick, I., Waggoner, P. E., Ausubel, J. H. & Kauppi, P. E. A national and international analysis of changing forest density. *PLOS ONE* **6**, 1–7, DOI: 10.1371/journal.pone.0019577 (2011).

8. Liu, T. *et al.* A deep neural network for the estimation of tree density based on high-spatial resolution image. *IEEE Transactions on Geosci. Remote. Sens.* **60**, 1–11, DOI: 10.1109/TGRS.2021.3101056 (2022).

9. Lamonica, D., Pagel, J. & Schurr, F. M. Predicting the dynamics of establishing tree populations: A framework for statistical inference and lessons for data collection. *Methods Ecol. Evol.* **12**, 1721–1733, DOI: https://doi.org/10.1111/2041-210X.13656 (2021). https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.13656.

10. Wu, W., Gong, C., Li, X., Guo, H. & Zhang, L. An online deep convolutional model of gross primary productivity and net ecosystem exchange estimation for global forests. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **12**, 5178–5188, DOI: 10.1109/JSTARS.2019.2954556 (2019).

11. Jianjun, D., Chunqiao, S., Ruifan, L. & Guohua, Z. Spatial prediction of population based on random forest. In *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, vol. 10, 1360–1363, DOI: 10.1109/ITAIC54216.2022.9836834 (2022).

12. Ashraf, M. I., Meng, F.-R., Bourque, C. P.-A. & MacLean, D. A. A novel modelling approach for predicting forest growth and yield under climate change. *PLOS ONE* **10**, 1–18, DOI: 10.1371/journal.pone.0132066 (2015).

13. Nishii, R. & Tanaka, S. Unified modeling based on svm and svr for prediction of forest area ratio by human population density and relief energy. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2552–2555, DOI: 10.1109/IGARSS.2015.7326332 (2015).

14. Timilsina, S., Aryal, J. & Kirkpatrick, J. B. Mapping urban tree cover changes using object-based convolution neural network (ob-cnn). *Remote. Sens.* **12**, DOI: 10.3390/rs12183017 (2020).

15. Pereira Martins Silva, J. *et al.* Prognosis of forest production using machine learning techniques. *Inf. Process. Agric.* **10**, 71–84, DOI: https://doi.org/10.1016/j.inpa.2021.09.004 (2023).

16. Forest Inventory and Analysis National Program - About Us (2022).

17. Ozone County Population (2015).

18. Harris, I. High-Resolution Gridded Datasets (2022).

19. Keras documentation: About keras (2023).