Predicting Soil Carbon Content to Inform Soil Preservation

Matt Reed
Department of Computer Science
Stanford University
Stanford, CA 94305
mattreed@stanford.edu

Grant Sheen
Department of Computer Science
Stanford University
Stanford, CA 94305
gsheen@stanford.edu

Peng Hao Lu
Department of Computer Science
Stanford University
Stanford, CA 94305
penglu@stanford.edu

1 Introduction

Soil is one of the largest carbon reservoirs on Earth, containing approximately 75% of the carbon stored on land — three times more than the amount stored in living plants and animals. Soils perform carbon sequestration, the long term storage of carbon dioxide, which plays a significant role in maintaining a balanced and harmonious global carbon cycle.

However, soils are increasingly being destroyed through harmful practices such as deforestation, urban development, pollution, etc. This leads to the stored carbon being emitted into the atmosphere as CO_2 and contributing to the negative feedback loop of climate change.

As a result, it is paramount for us to understand which areas of soil store the most amount of carbon and thus are the most important to preserve. In order to do this, we need to know the soil organic carbon content (SOC) within a unit of soil. However, SOC is difficult to measure: dry combustion, the most accurate form of measurement, requires expensive equipment and trained personnel. Therefore, we need to predict SOC based on metrics that are easier to measure or that we know of already.

Our project predicts SOC from other known soil parameters. The input to our model is the known properties of a given unit of soil. We then used a neural network to output a predicted SOC for that unit of soil. Additionally, our model is able to take any given soil attributes as input and predict SOC as accurately as possible. We designed our model to be robust to missing feature values by implementing a various imputation layers and a dropout layer.

2 Related Work

Although not specific to peatlands, Tramontana et al. (2016) shows how regression algorithms have a high capability to predict carbon fluxes using other environmental variables. Rafat et al. (2021) expands on this approach by demonstrating its viability specifically on predicting carbon emissions from peatlands. Rafat et al. was able to identify that seven such variables are needed, including air and soil temperatures, wind speed, soil moisture content, and net radiation above the canopy. This approach extends to areas outside of carbon as well, with Hikouei et al. (2023) predicting groundwater levels in peatlands through the same method.

While the previous papers used data from sensors, Habib et al. (2024) has a different approach to the issue, where they use satellite data from Google Earth Engine and Sentinel-2 as the basis for their model. They were able to achieve a high accuracy, showing the potential of satellite imaging data. Ingle et al. (2023) also utilizes this approach with PlanetScope imagery for methane fluxes and was able to achieve success. Although we are not using satellite imaging in our study, these studies

show that remote sensing technology such as satellite imaging is a very promising direction for future research within the space.

3 Dataset and Features

Our model uses the Harmonized World Soil Database, which is the world's most comprehensive collection of soil data. It aggregates information across seven source databases and standardizes them across around 40 soil properties (both continuous and categorical).

We performed Exploratory Data Analysis on the training set and found 3 notable properties. First, some parameters had a significant number of missing values as seen in Figure 1. A result, we decided to implement data imputation. Second, there were some strongly predictive features. For continuous features, we produced the correlation matrix in Figure 2. The most predictive features were Bulk Density, pH in water, and Cation Exchange Capacity. For categorical features, we created box plots of the carbon content distribution for each feature value. Some of the predictive features we found were FAO90, WRB + Phases, and WRB2, which are different classification systems for soil. The plots for SOC distributions are in Figure 3.

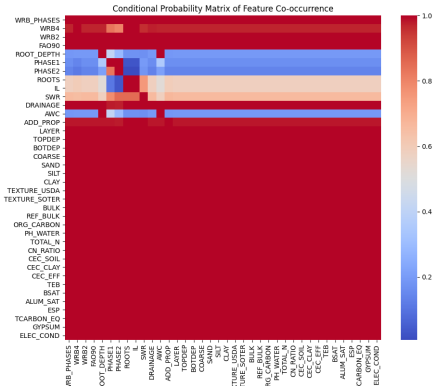


Figure 1: Co-Occurrence Probabilities

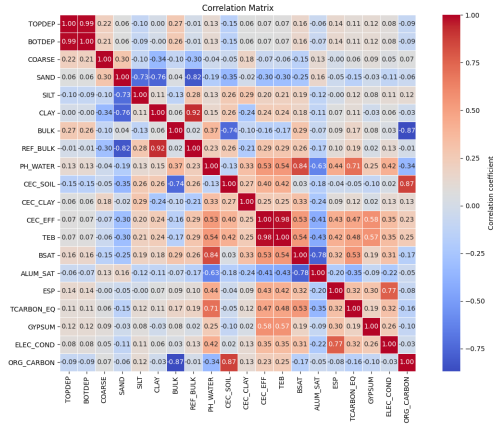
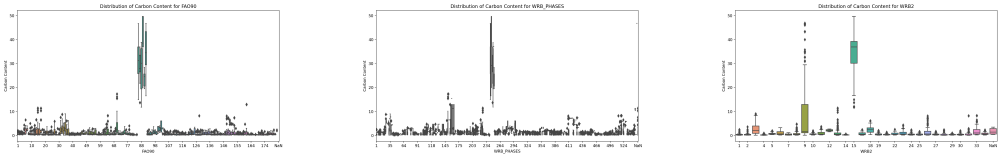


Figure 2: Correlation Matrix



(a) FAO90 SOC (b) WRB + Phases SOC (c) WRB2 SOC

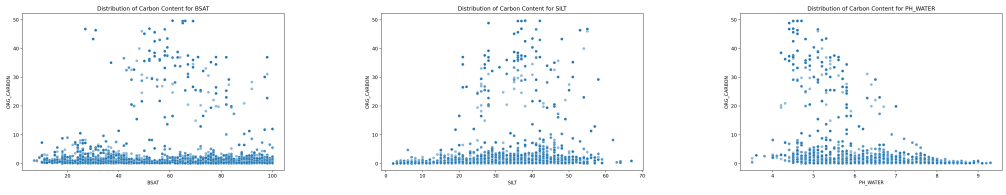
Figure 3: Predictive Categorical Features

Third, we explored the linearity of our features and found that they were predominantly non-linear. Some examples of non-linear continuous features are shown in Figure 4. The prevalence of non-linearity in our features informed our decision to implement a neural network.

4 Methods

4.1 Baseline Model

Our baseline model is a simple linear regression model with LASSO regularization that predicts a soil sample's organic carbon content using a linear combination of the other features of the sample. Due to the immense size of the dataset, we could not tractably compute the closed form solution to the mean squared error loss objective, so we resorted to using gradient descent.



(a) BSAT SOC (b) Silt SOC (c) pH Water SOC

Figure 4: Non-Linear Continuous Features

Due to the noisiness of the data, we found it difficult to make the model converge using stochastic gradient descent, so we used full-batch gradient descent. After 30,000 steps, the algorithm converged based on the stopping criteria that the change in loss was sufficiently small. The predicted values for the test set are shown against their actual values in Figure 6a. As can be seen in the figure, the model captures the correlation of the data well, but there is room for improvement which can be seen as the distance from the data points to the ideal line $\hat{y} = y$.

The linear model had many drawbacks that we mitigated in our improved model. Firstly, it is unclear whether the residuals shown in the figure are due to irreducible noise in the data or if adding non-linearities into the model will allow these differences to be explained by the various features. Secondly, since our dataset has a large majority of data-points with nearly-zero SOC, the linear model does really well on those points but less well on the points we really care about, the high SOC samples (as seen in Figure 6a). Thirdly, the linear model has no way of intelligently imputing values for missing features in the dataset. This ability is crucial since our dataset is missing a significant number of values, and more importantly, this model is intended to make it easy to predict SOC, not harder. Ideally, one should be able to input some subset of the features and still get an accurate prediction about the SOC.

4.2 Non-Linear Model

To solve these problems, we implemented the model architecture as shown in Figure 5. It has two fully-connected layers with ReLU activation functions to capture the nonlinear nature of the dataset and clamp the output values to the range $(0, \infty]$. It has a One-Hot Encoder for the categorical features and normalization layer for more consistent learning. Crucially, it has an imputation layer for predicting missing values as well as a dropout layer for simulating missing features. We trained it using mini-batch gradient descent which we found had the best naturally regularizing effect. Finally, the loss function we optimized was a weighted mean squared error which was weighted by the SOC for the data point (see Equation 1).

Imputation Layer: The role of the imputation layer is to predict the missing values of features from the dataset before it is passed through the rest of the model. This helps with the stability and quality of predictions. We tried three different imputation methods and compared them. *Naive Imputation* (used in our linear model as well) substitutes missing values with zero. This is not especially problematic as the model has access to missing-values mask as well and can learn a bias term for when the value is not present, but it also makes training more difficult and less successful. A slightly more sophisticated method was sampling these features from a *Gaussian Distribution* with a mean and standard deviation defined by the data-points that contain that value. For training this added noise helps with regularization, and during testing (when we only use the mean instead of sampling), the model is more stable and less overfit. Our final approach was to use *K-Nearest Neighbors Imputation*. This is theoretically the most accurate method since it draws missing values from similar samples, however, due to the large size of our dataset and huge dimensionality of our feature space (after creating one-hot-encodings), this method proved incredibly slow.

Dropout Layer: Although our model is already missing some features from a large number of data-points, we wanted the model to be robust to missing features, so we built in this layer that simulates removing data-points. This incentivizes the model to use all of the features to makes its predictions instead of a few.

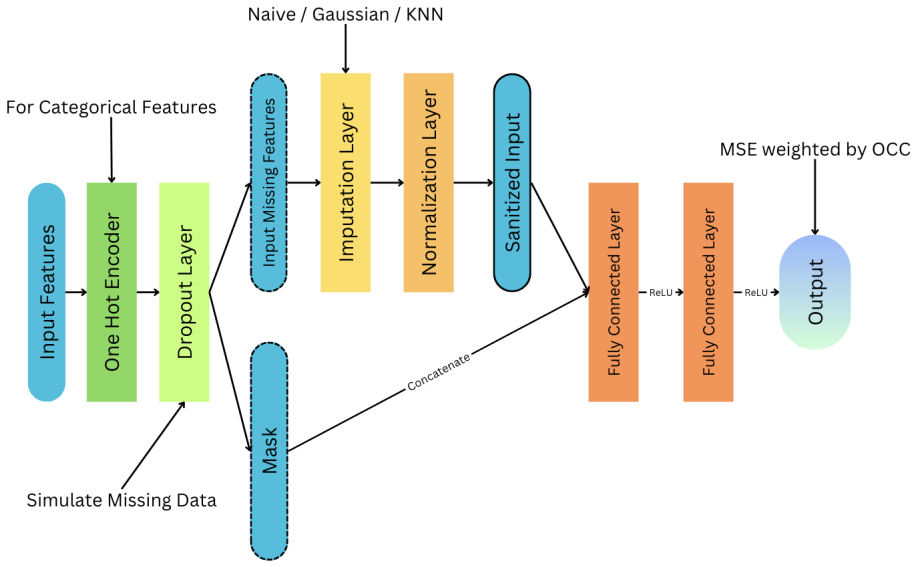


Figure 5: NN Model Architecture

Loss Function The loss function we used is based on the mean squared error, but is weighted by the actual carbon content of each data-point. This is because we care most about the accuracy of predictions on the crucial carbon sinks and less so about the samples without a lot of carbon (we'd rather have a false positive than a false negative), as well as because the high-SOC samples are a minority of the dataset. Our weighted mean squared error objective function for a single data-point is written as:

$$\ell_{weighted}(M|x^{(i)}, y^{(i)}) = (M(x^{(i)}) - y^{(i)})^2 \frac{\lambda y^{(i)} + 1}{\frac{1}{n}\sum_{j=1}^{n} \lambda y^{(j)} + 1} \tag{1}$$

The scaling factor, λ, controls how much we prioritize accuracy on higher SOC datapoints. We add 1 so that data-points with $y^{(i)}$ close to zero are still weighted. Finally, we have the normalization constant in the denominator so that the magnitude of the weighted MSE is comparable to the standard MSE objective.

5 Experiments

Our primary metric for evaluation was the weighted mean squared error objective shown in Equation 1. This objective was effective at prioritizing high-SOC samples as the MSE for samples with SOC > 15 g/Kg was 3.1769 when trained using weighted MSE objective whereas it was 4.3831 when trained using traditional MSE. We used 3-fold cross validation to choose our hyper-parameters to minimize validation error. We found that learning rate of .001, 150 epochs, $\lambda = 0.1$, and stochastic gradient descent with mini batches of 32 worked best. Note that we chose different values for the dropout rate depending on our objective. Our first experiment used no dropout, and the second was with dropout.

5.1 Experiment 1: Model Without Dropout

Our first experiment was to try to build a model that would perform best on the test set using all features (except N, CN_RATIO, and BULK due to their reliance on SOC measurements). We used the model shown in Figure 5 with a dropout rate of 0.0 so the only missing values were values that were actually missing in the data. For this task, the model performed exceptionally well, almost a 10x improvement from the baseline model in terms of test error and test weighted error (see Table 1).

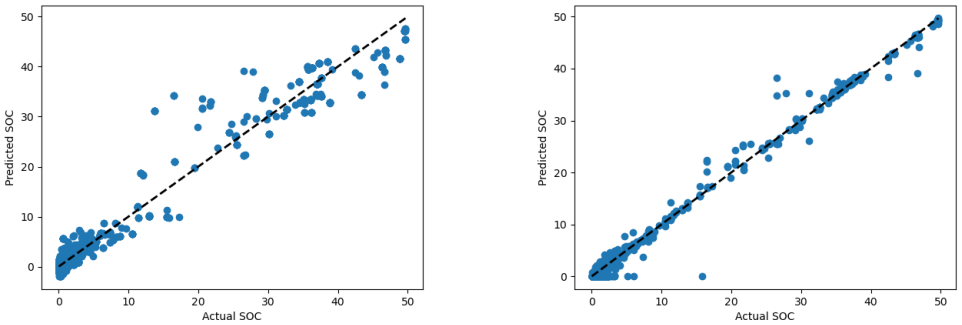
Table 1: MSE of Linear and Non-Linear (No Dropout) models on Test Set

Model	Training ℓ	Training $\ell_{weighted}$	Test ℓ	Test $\ell_{weighted}$
Linear	1.809	-	2.251	5.518
Non-Linear (No Dropout)	-	**0.227**	**0.413**	**0.690**
Non-Linear (Dropout)	-	1.514	2.921	3.803

Table 2: Weighted Random Subset Loss

Features Missing (n)	0	5	10	15	20	25	30
Linear	5.518	25.037	49.329	75.666	98.349	130.646	167.927
Non-Linear (No Dropout)	**1.026**	**2.183**	5.773	11.446	20.256	33.769	48.507
Non-Linear (Dropout)	3.803	3.211	**3.817**	**6.116**	**11.221**	**22.344**	**40.606**

This confirmed our hypothesis that the features provided have predictive power for SOC that is not purely linear. Further experiments increasing model complexity showed a decrease in validation loss which demonstrates that our model is about as low bias as it can be without over fitting and the test error is likely a substantial amount of noise. The fit of this model on the test set is shown in Figure 6b which is visibly much more accurate than the baseline linear model.



(a) Linear Model

(b) Non-Linear Model (no dropout)

Figure 6: Predictive Ability of Models on Test Set

5.2 Experiment 2: Model With Dropout

Our second experiment was to try to build a model that was especially robust to missing features. Ultimately, the goal of this project was to create a model that makes predicting SOC for preservation purposes as easy as possible, so we attempted to create a model that was able to predict SOC based on a subset of the total input features. This is different than standard feature selection or LASSO regression which selects a static group of best features; instead, we wanted the model to be able to predict SOC based on whatever measurements are accessible, regardless of which ones they are. Adding a dropout layer was our solution to allowing the model to rely on all different kinds of inputs and still providing reasonable predictions. Using cross-validation, we determined a dropout rate of 0.1 was best for random subsets that were missing between 5-15 features of the total 37 features (which is mathematically consistent with the median 10 being roughly 0.3 of 37). As seen in Table 2, the model with dropout did slightly worse on full and most observability (0, 5 features missing), but did much better than the non-dropout model for random subsets with a large number of features missing. These metrics were calculated by making copies of the test set with n random features missing from each data-point, then calculating weighted MSE on the predictions.

6 Conclusion / Future Work

In conclusion, we created a Neural Network model that predicts Soil Organic Carbon content (SOC) from various soil attributes with a weighted test MSE of 0.69. This was a significant improvement from our baseline lasso regression result of 5.52. We found that the non-linearity of our model led to a more accurate fit of the predominantly non-linear soil features. Additionally, incorporating Gaussian data imputation and a dropout layer into the model enabled it to be more robust to missing values. We hope that our model will allow climate scientists to gain insight into which regions of soil are the most important to preserve through accurate prediction of SOC from known soil attributes. In the future, we hope to expand upon our work by experimenting with different forms of data such as satellite imaging as well as repurposing our model for similar tasks such as predicting ocean acidification.

Our code can be found at this repo:
https://github.com/gsheen11/soil-carbon-content

References

FAO and IIASA. 2023. Harmonized world soil database. https://doi.org/10.4060/cc3823en. Accessed: 2024-02-15.

Wahaj Habib, Ruchita Ingle, Matthew Saunders, and John Connolly. 2024. Quantifying peatland land use and co2 emissions in irish raised bogs: mapping insights using sentinel-2 data and google earth engine. *Scientific Reports*, 14(1):1171.

Iman Salehi Hikouei, Keith N Eshleman, Bambang Hero Saharjo, Laura LB Graham, Grahame Applegate, and Mark A Cochrane. 2023. Using machine learning algorithms to predict groundwater levels in indonesian tropical peatlands. *Science of the Total Environment*, 857:159701.

Ruchita Ingle, Wahaj Habib, John Connolly, Mark McCorry, Stephen Barry, and Matthew Saunders. 2023. Upscaling methane fluxes from peatlands across a drainage gradient in ireland using planetscope imagery and machine learning tools. *Scientific Reports*, 13(1):11997.

Arash Rafat, Fereidoun Rezanezhad, William L Quinton, Elyn R Humphreys, Kara Webster, and Philippe Van Cappellen. 2021. Non-growing season carbon emissions in a northern peatland are projected to increase under global warming. *Communications Earth & Environment*, 2(1):111.

Gianluca Tramontana, Martin Jung, Christopher R Schwalm, Kazuhito Ichii, Gustau Camps-Valls, Botond Ráduly, Markus Reichstein, M Altaf Arain, Alessandro Cescatti, Gerard Kiely, et al. 2016. Predicting carbon dioxide and energy fluxes across global fluxnet sites with regression algorithms. *Biogeosciences*, 13(14):4291–4313.