AI assisted prediction of biochar carbon stability

David Mlčoch

Master of Science
School of Informatics
University of Edinburgh
2022

Abstract

Stable biochar enables the effective removal of carbon from the atmosphere. Biochar is produced through pyrolysis of biomass. It then stores carbon for long periods and mitigates climate change. But to date, there is no universally accepted way of predicting biochar carbon stability. In this thesis, we used machine learning techniques to predict biochar carbon stability from biomass characteristics and pyrolysis conditions. This has never been attempted in previous studies. The dataset included biomass properties, pyrolysis conditions and biochar carbon stability measurements. Biochar carbon stability methods included Proximate analysis, Chemical oxidation and H: C molar ratios. We have implemented and evaluated the following models: Linear Regression, Random Forests, Support Vector Regression and Gaussian Processes. Consequently, we performed feature importance analysis and partial dependence analysis.

Our findings show that predicting biochar carbon stability is possible with the best mean absolute percentage error (MAPE) of $13.6\% \pm 4.9\%$ achieved with Gaussian Processes when predicting all carbon stability methods together. Support Vector Regression model achieved MAPE $= 15.6\% \pm 5\%$ for Chemical oxidation method. Random forests proved the best for modelling Proximate analysis (MAPE $= 16.4\% \pm 3.4\%$) and H:C molar ratio (MAPE $= 19.8\% \pm 4.3\%$). The developed models can help domain experts to guide laboratory experiments and better understand the underlying pyrolysis processes. We believe the errors could be reduced with a more comprehensive dataset, including a wider variety of feedstock types and pyrolysis conditions such as residence time and heating rate. Residence time with contents of nitrogen and carbon were the most important features for Proximate analysis, with pyrolysis temperature being dominant for other stability methods.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*David Mlčoch*)

Acknowledgements

I would like to express my gratitude to my supervisors Dr. Nigel Goddard and Dr. Ondřej Mašek for their guidance and support throughout this project. Furthermore, I would like to thank my family and friends (BB).

Table of Contents

Chapter 1

Introduction

1.1 Motivation

With an increasing amount of carbon dioxide in the atmosphere, carbon sequestration helps to reduce global climate change by capturing and storing atmospheric carbon dioxide. One of the possible carbon storages is biochar. Biochar is a solid product of the thermochemical conversion of biomass called pyrolysis. The biomass contains atmosphere-derived carbon, which is stored stably in biochar after pyrolysis. Through pyrolysis the unstable carbon in biomass is converted into stable carbon [1]. For biochar to deliver on its potential, producing highly stable biochar is necessary. Stable biochar contains carbon that does not decompose and releases carbon back into the atmosphere for centuries and millennia [2]. Biochar has other positive usages, such as soil fertilizer. Therefore, biochar production reduces atmospheric greenhouse gas concentration and also improves soil or marine environment.

1.2 Problem statement

As we can't wait hundreds of years to measure carbon decomposition, we need to use other metrics to act as carbon stability proxies. Many experiments were conducted on the aspect of biochar stability, yet to date, there is no universally accepted way of carbon stability prediction [3]. In this work, we test machine learning methods for predicting biochar carbon stability for the first time. As numerous studies showed [4, 5, 6], it is possible to apply machine learning methods to model complex processes happening during pyrolysis. Nevertheless, results varied, and as suggested by [7], more in-depth work should be carried out investigating machine learning methods to predict biochar properties. Predicting biochar's long-term stability is challenging due to many chemical factors affecting it.

1.3 Contributions

In this project, we have developed machine learning models for predicting biochar carbon stability. We have tested Linear regression, Support Vector Regression, Random Forests and Gaussian processes. We have compared results across various models and performed feature importance and dependence analysis. Three biochar carbon stability measures were modelled: **Chemical oxidation** [8], **Proximate analysis** [9], and **H: C molar ratios** [10]. The inputs to our models were data from experimental observations gathered by the UK Biochar Research Centre. These included biomass characteristics (e.g. contents of carbon, hydrogen etc.) and pyrolysis conditions (e.g. pyrolysis temperature or residence time).

1.4 Thesis structure

This thesis is structured as follows: Chapter 2 introduces the background and related work relevant to our problem. We introduce biochar, carbon stability methods and studies using machine learning for biochar properties prediction. Chapter 3 describes our dataset. Chapter 4 covers the methodology defining our evaluation metrics, machine learning methods and feature importance methods. We show and discuss our results in Chapter 5. We conclude and suggest future directions in Chapter 6.

Chapter 2

Background

The ultimate goal is to create stable biochar. This work contributes to this goal by creating biochar carbon stability prediction models. If we manage to predict biochar stability, there will not be a need to conduct as many physical experiments in the laboratory as we will only conduct experiments with promising parameters validated by the model. We first need to understand what biochar is, how it is created and how its carbon stability can help with climate change. We introduce and define the most important concepts in this chapter. We also present the most recent papers that used machine learning to predict biochar properties. Parts of Sections 2.1, 2.2 and 2.3 are based on my IPP report.

2.1 Biochar and Pyrolysis

Biochar is a charcoal-like substance rich in carbon that is created from biomass (organic material) by a process called pyrolysis [11]. Figure 2.1 shows a picture of biochar. Most of the biochar composition is carbon (usually more than 70%). The rest is made of nitrogen, hydrogen and oxygen, among other elements [12]. Various types of biomass can be used to produce biochar, such as waste feedstock, agricultural waste, or municipal sewage sludge [11]. Pyrolysis transforms the biomass into biochar through a thermochemical process that undergoes either in the complete absence of oxygen or in limited supply [1]. Usually, the operating temperature of pyrolysis is in the range of 600–900°C. The other important parameters of pyrolysis are heating rate and residence time [1]. The output of the pyrolysis process includes the biochar, pyrolysis oil and synthesis gas [13].

Biochar has many applications. It has been produced and utilized for several thousand years [12]. Apart from our main interest in sequestrating carbon, biochar is also used for soil fertilization, gas and water purification, or metallurgical applications [12]. Burning biomass through pyrolysis also generates heat and power. Recently, biochar has gained popularity for its potential to reduce greenhouse gas emissions by replacing fossil carbon carriers [12]. The Figure 2.2 indicates the biochar carbon cycle. Carbon dioxide is captured from the air and bound in biomass through photosynthesis [14]. The biomass is then processed through pyrolysis, which produces biochar and other by-products. Biochar is subsequently disseminated into the ground, where it permanently stores carbon.

Figure 2.1: Biochar [15]

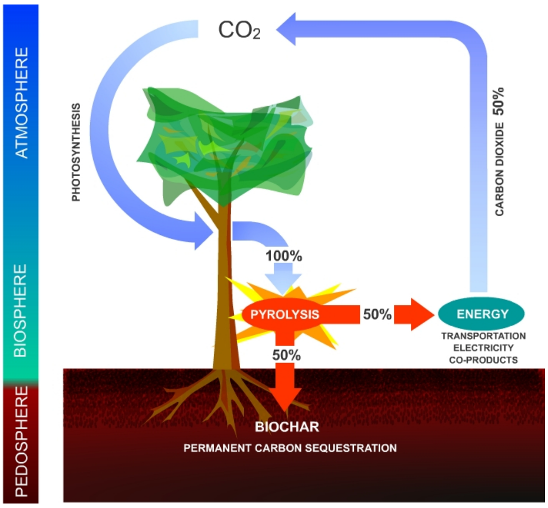


Figure 2.2: The biochar carbon cycle [16].

2.2 Biochar stability assesment methods

Stable biochar does not decompose and releases carbon back into the atmosphere for centuries and millennia. But waiting centuries to test the stability is not feasible. Therefore, we use approximate methods to estimate carbon stability. This text will sometimes refer to them as biochar carbon stability proxies.

The most common methods of stability determination were reviewed by [3]. These are Proximate analysis, Chemical Oxidation and Elemental molar ratios (also called Ultimate analysis). All methods are correlated, and we will use the same methods in this study. As found by [3], a strong correlation ($R>0.79$) is between the Chemical oxidation and Proximate analysis. A slightly weaker correlation is between the Chemical oxidation and H:C ratio ($R=0.65$).

We briefly introduce the measurement procedures here. To measure the Proximate analysis the sample is first heated. The air is then added to the system, and the sample is combusted. Finally, the fixed carbon that approximates the carbon stability is calculated by subtracting moisture, volatile and ash values from the original mass [3]. To calculate the elemental ratios, we can determine the quantity of each element using an elemental analyser. With the Chemical oxidation method, the milled sample is treated with H_2O_2 initially at room temperature and then at 80°C for 48 h. Stable C is then expressed as the percentage of the carbon that remains after oxidation [8].

2.3 Machine learning for biochar properties prediction

To our knowledge, no previous study has attempted to use machine learning techniques to predict biochar stability. Some studies have used machine learning methods on similar problems. We summarize the most similar studies with the input and predicted features in Table 2.1. The prediction of the chemical properties was usually defined as a regression task. Three methods have been most widely used: Random Forests (RF), Support Vector Regression (SVR) or Support Vector Machines (SVM), and Artificial neural networks (ANN). Random Forests and Support Vector Regression generally proved superior to Artificial neural networks [5, 17, 18]. Data for these tasks were usually acquired by costly and time-consuming laboratory experiments. Therefore, the sizes of these datasets were fairly small. ANNs usually failed as they are more suitable for tasks with more data.

Studies [4] and [5] predicted biochar yield from biomass characteristics and pyrolysis conditions, where [4] also predicted the carbon contents of biochar. Using Random forests, the authors achieved $R^2 = 0.85$ on both tasks. Both R^2 and MAPE was used to evaluate models in [5]. Least-squares SVR (LS-SVR) achieved R^2 of 0.96 and MAPE = 4.9% on the test set. ANN showed worse results on the test set with R^2 of 0.80 and MAPE = 9.6%. Both studies [5, 17] achieved good performance, proving that machine learning methods can effectively perform predictions for similar tasks. Feature importance analysis was also performed on the Random forest approach taken by [4]. It was shown that pyrolysis conditions were more important than biomass characteristics. The most influential pyrolysis conditions to predict the biochar yield from the cattle manure in [5] were pyrolysis temperature and moisture content, especially sample mass.

Methods	Input features	Predicted feature(s)	Publication
RF	Biomass characteristics, Pyrolysis conditions	Biochar yield, Carbon contents	[4]
ANN, SVR	Cattle manure characteristics, Pyrolysis conditions	Biochar yield	[5]
RF, ANN	Biochar characteristics	Metal sorption onto biochars	[17]
RF, SVM	Temperature, Equivalence ratio, Fuel flow rate	Syngas composition	[19]
RF	Oxidation experiment data	Coal spontaneous combustion	[20]
RF, SVR	Biomass characteristics	Fuel properties of hydrochar and pyrochar	[21]
RF, ANN, SVR	Proximate analysis, Ultimate analysis contents	Biomass higher heating value	[18]
ANN	Biomass characteristics	Kinetic parameters of biomass pyrolysis	[6]

Table 2.1: Related works

Another similar problem investigated by [19] is a prediction of syngas composition for downdraft biomass gasification. Even though the model's output is continuous, the authors considered a different approach, transforming the regression problem into a classification problem. The output concentration values were quantized to the nearest integers before the classification. Random Forests and LS-SVMs achieved 89% and 96% classification accuracy scores, which both outperformed the stoichiometric and non-stoichiometric models previously used for the gasification product estimation [19].

Chapter 3

Dataset

In this chapter, we introduce our dataset. Section 3.1 describes model inputs that consist of pyrolysis parameters and feedstock properties, while Section 3.2 describes model outputs which are biochar carbon stability metrics. We show how the dataset was created, the data distribution and the correlations.

The data for this project were provided by Dr Ondrej Masek from the UK Biochar Research Centre. There were two types of measurements conducted in the laboratory. First, the feedstock properties were measured for different biomass types, sometimes repeatedly for the same sample. Then the feedstock sample was processed using pyrolysis under measured conditions, and the newly created biochar was analyzed with biochar stability methods. In some cases, multiple experiments were conducted with identical pyrolysis parameters and on the same feedstock. Therefore, we have two major groups of input features - Pyrolysis parameters and Feedstock properties. The outputs of our models are the biochar carbon stability measurements.

3.1 Pyrolysis parameters and Feedstock properties (model inputs)

Input to our models are Pyrolysis and Feedstock properties listed in Table 3.1. We have 13 input features for Proximate analysis and 10 for other methods. Table 3.1 also indicates how much data are missing for each input feature and method. As mentioned later, because of the amount of missing data, we do not always use all the features in our models.

Input feature	**Chemical oxidation**	**Proximate analysis**	**H: C molar ratio**	**All methods**
Pyrolysis temperature (°C)	0%	0%	0%	0%
Residence time (min)	-	0%	-	0%
Volatiles (%, daf)-Feedstock	13.4%	55.3%	11.3%	8.90%
Fixed C (%, daf)-Feedstock	13.4%	55.3%	11.3%	8.90%
% C, ave-Feedstock	13.4%	35.6%	11.3%	8.90%
% H, ave-Feedstock	13.4%	35.6%	11.3%	8.90%
% N, ave-Feedstock	13.4%	35.6%	11.3%	8.90%
Remainder (%)-Feedstock	13.4%	35.6%	11.3%	8.90%
O:C mol/mol-Feedstock	13.4%	35.6%	11.3%	8.90%
H:C mol/mol-Feedstock	13.4%	35.6%	11.3%	8.90%
Biochar C stability (%)-Feedstock	13.4%	68.1%	11.3%	8.90%
Heating rate (°C/min)	-	44.9%	-	-
Volatile content (db%)	-	80.4%	-	-

Table 3.1: Dataset features with a proportion of missing values indicated (0% meaning no data missing)

Figure 3.1 shows the Pearson correlation matrix. It shows the pairwise correlation between the input variables. Volatiles (%, daf) and Fixed C (%, daf) are perfectly negatively correlated. This is not surprising as one is calcuated from the other, and their relationship is inversely proportional: Volatiles $\propto \frac{1}{\text{Fixed C}}$. Other highly correlated features are O:C mol/mol with % C, ave (-0.99), and O:C mol/mol with Remainder (%) (0.99). We omit Volatiles (%, daf) and O:C mol/mol further in our models.

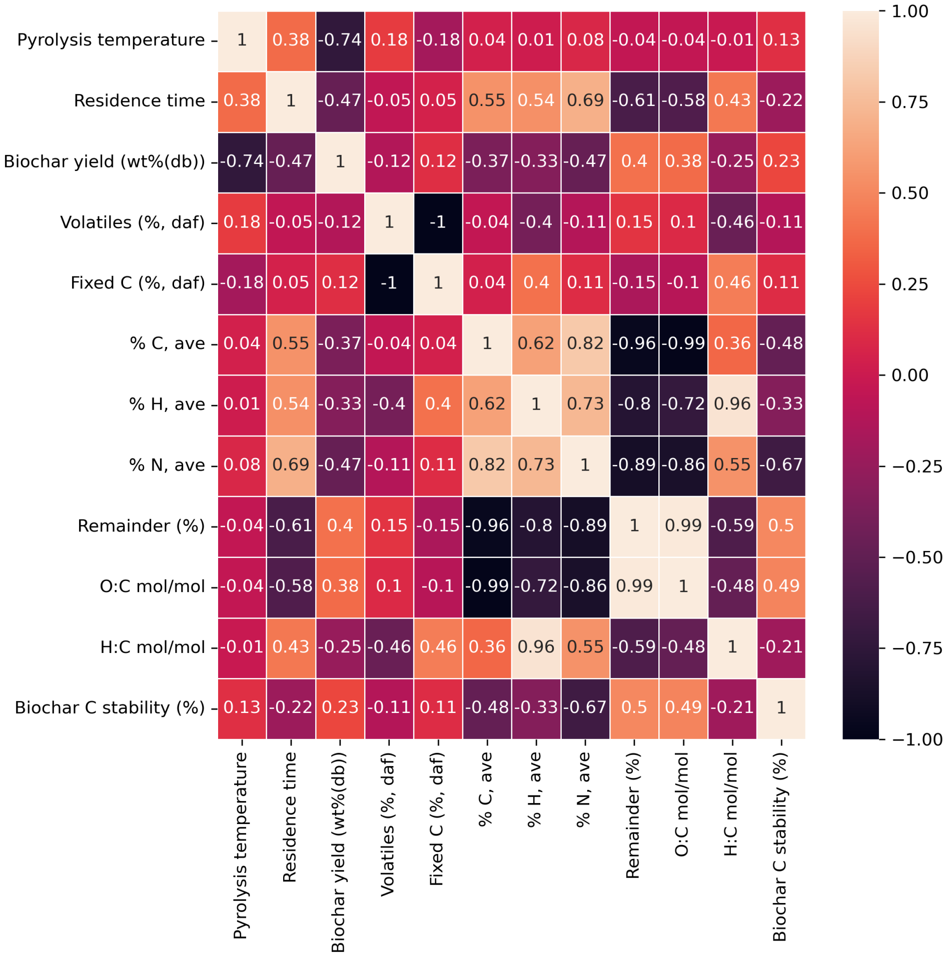


Figure 3.1: Pearson correlation coefficient - Pyrolysis and Feedstock properties

3.2 Biochar carbon stability methods (model outputs)

We can't measure biochar carbon stability directly, and there is not yet a universally accepted method for biochar carbon stability determination. We use proxies to estimate the biochar carbon stability. In this work, we use three methods to compare the relative stability of different biochar materials. These proxies are: **Chemical oxidation** [8], **Proximate analysis** [9], and **H: C molar ratios** [10]. We show the Pearson correlation coefficients for these methods in Table 3.2 . It is visible that the proxies are highly positively or negatively correlated. Still, since they are not perfectly correlated, each proxy might possess different predictive information. In Figure 3.2, we show the data distribution histograms. We see that for Proximate analysis measurements, we have two areas with samples. H:C molar ratio and Chemical oxidation data are distributed more uniformly with exception of a few outlier areas.

	Proximate analysis	Chemical oxidation	H:C molar ratio
Proximate analysis	1	0.80	-0.77
Chemical oxidation	0.80	1	-0.91
H:C molar ratio	-0.77	-0.91	1

Table 3.2: Biochar stabilitiy proxies - Pearson correlation coefficients

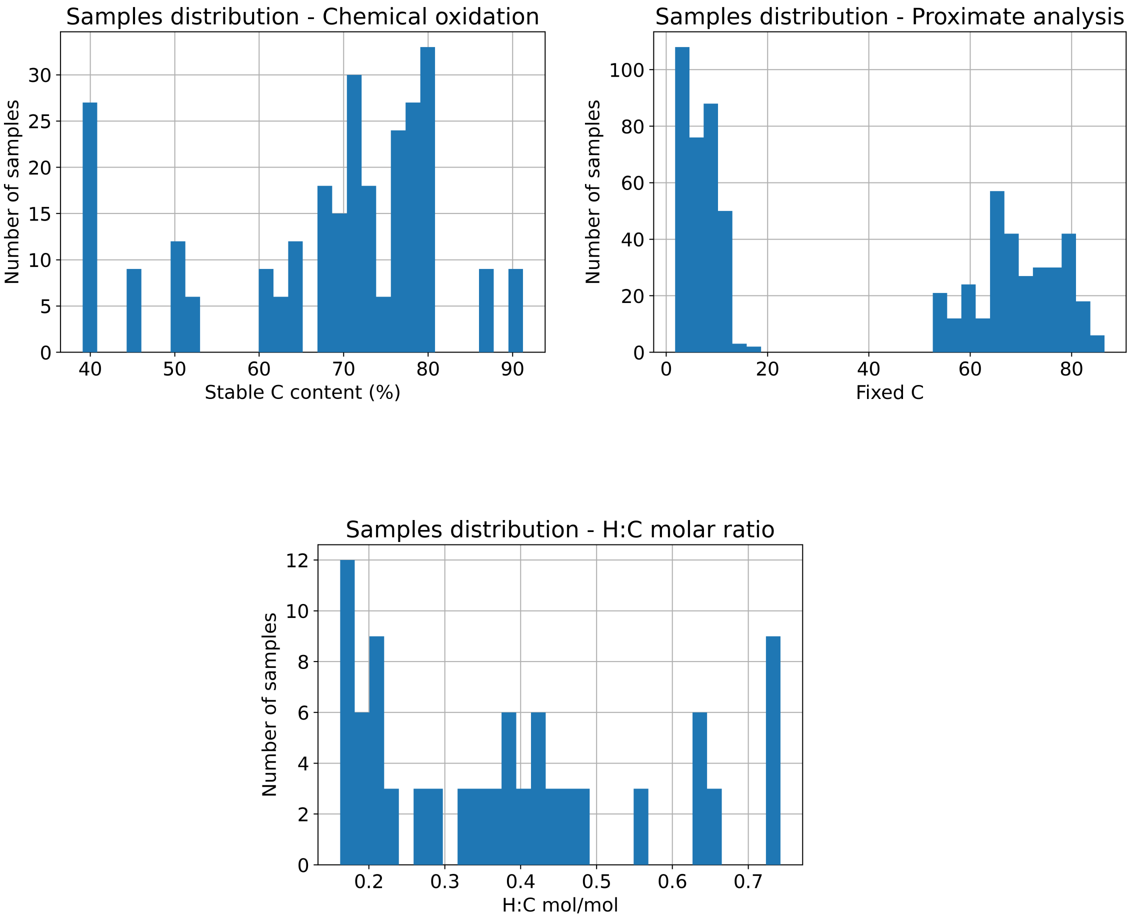


Figure 3.2: Biochar proxies - Data distribution histogram

For each method, we have a different amount of data, different sample distribution and slightly varying input features. Some input features contain missing values, as some experiments did not measure every property of the biomass or did not track some pyrolysis process conditions.

The amount of data and proportion of missing values for all Biochar carbon stability methods are summarized in Table 3.3. The column *Distinct samples* shows the number of pyrolysis experiments, each with unique pyrolysis parameters. As sometimes the experiments were repeated with the same parameters for the same sample, the next column shows the total number of measurements for multiple runs. The column *Samples multiple runs with Feedstock* shows the number of samples after we perform the left join of the Feedstock table on the Pyrolysis properties table. We have more samples than in the previous column, as for each feedstock type, we might also have multiple feedstock measurements. This left join then increases the final table size with many values repeated. This is important to keep in mind when performing the cross-validation split. We will split the data grouped by the Pyrolysis parameters. This prevents information leakage to the test set.

We also combine all stability methods together as a three-dimensional output (All proxies row in Table 3.3). We merge the tables for all three proxies and only keep samples where we have data for all of them. The last two columns in Table 3.3 show that we miss values for all methods, most notably 39% for Proximate analysis. Most of these data are missing from the Feedstock properties. From Table 3.3, it is visible that our dataset is really small, with only dozens to hundreds of samples. We have most data for Proximate analysis, but with many missing values. Proximate analysis is followed by Chemical oxidation, and we have the least data for the Molar ratio method.

Biochar carbon stability method	**Distinct samples**	**Samples multiple runs**	**Samples multiple runs with Feedstock**	**Missing values overall**	**Missing values on Feedstock**
Chemical oxidation	41	123	312	10%	10%
Proximate analysis	291	596	1007	39%	31%
H: C molar ratio	39	39	102	8%	8%
All proxies	39	359	1007	6%	6%

Table 3.3: Dataset size

Chapter 4

Methodology

This chapter outlines methods and evaluation. We start with introducing necessary data preprocessing in Section 4.1. Evaluation metrics are described in Section 4.2 and cross validation procedure in Section 4.3. Section 4.4 outlines machine learning methods that we used for modelling biochar stability. Feature importance methods are shown in Section 4.5.

We implemented the described methods and evaluation in Python. Linear regression, Support Vector Regression and Random forests were modelled using scikit-learn [22] and Gaussian processes with GPyTorch [23]. To track our experiments we used Weights & Biases [24].

4.1 Data prepossessing

Before feeding the data into our models, we perform data preprocessing which includes Data imputation to account for the missing values and Data scaling to account for different scales of the features.

4.1.1 Data imputation

As mentioned in previous chapter, we have a considerable amount of missing values in our inputs. We can either remove the samples that include at least one missing feature or impute the data. We only impute data for our models' inputs, not the outputs.

4.1.1.1 Removing data

Removing the samples results in even smaller datasets. For Chemical oxidation, Proximate analysis, H:C molar ratio outputs, removing missing values decrease the number of samples by 40 %, 36%, and 29% respectively. For proximate analysis, we slightly changed the procedure. If we removed all samples with at least one missing value, we would end up with zero samples, as every sample has at least one missing value. Hence for proximate analysis, we use a combined approach, where we drop features with the most missing values: Volatile content, Biochar C stability, and Heating rate. We also remove each datapoint, where more than 3 out of 9 features are missing. For the remaining data points, we use kNN imputation.

4.1.1.2 kNN imputation

kNN imputation has been found to perform well in practice [25, 26], where [25] compared kNN to six other imputation methods on five different numeric datasets with kNN resulting in the best approach.

K-Nearest Neighbors method finds *k* most similar neighbours that don't have the particular feature value missing. The similarity is measured as an Eucledian distance between the non-missing features of both samples. All our features are numeric, hence calculating similarity as Euclidean distance is possible. The imputed value is a mean of the *k* neighbour's values. We assume that our data are missing at random and can be explained by other variables. For example, if we miss one of the Feedstock features, we know that this value will be similar to another datapoint with the remaining Feedstock features the same as for the similar biomass.

kNN Imputation is a Single Imputation Method because it produces a point estimate. There are also Multiple Imputatnions Methods such as MICE [27] that can produce multiple estimates with uncertainty information. While multiple estimates might be useful, integrating multiple estimates with uncertainty into our consequent models is more complicated than point estimates, and hence we will not consider these methods in this work.

4.1.2 Scaling

Before we input our data into the models, we perform data scaling. This is necessary as our input features have different scales. Some methods, such as SVR, rely on calculating distances between observations. The distance between two observations differs for non-scaled and scaled cases. We test two types of scaling for our models: Standard scaling and Min-Max scaling. Standard scaling scales the values to have a zero mean and a standard deviation of 1:

$$z = \frac{x - \mu}{\sigma} \tag{4.1}$$

where μ is the mean of the data and σ is the standard deviation. Min-max scaling maps the samples into the [0, 1] interval:

$$z = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.2}$$

4.2 Evaluation Metrics

To evaluate our models, we used the mean absolute percentage error (MAPE) and coefficient of determination (R^2). Mean absolute percentage error is calculated as:

$$\mathrm{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{\max(\varepsilon, |y_i|)} \tag{4.3}$$

where n is the number of data points, y_i are observed values, $\hat{y}_i$ are predicted values, and ε is an arbitrary small yet strictly positive number to avoid undefined results when y_i is zero [22]. MAPE has an intuitive interpretation as a relative error, which is very useful for our prediction problem. We express MAPE as a percentage. It is also not influenced by a global scaling of the target variable.

Coefficient of determination is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.4}$$

where $\bar{y}$ is the mean of the observed data. R^2 measures how changes in the dependant variable can be explained by the change in the independent variable(s). It assesses how strong the linear relationship is between the two variables [28]. It ranges from negative values to 1, where 1 corresponds to a perfect fit and 0 to just predicting the mean. Negative values mean that predicting the mean of the observed data would give a better fit than the fitted function values.

We measure R^2 as it was used by previous literature, and it allows us to partially compare our models with similar papers. Unfortunately, there are issues with R^2 as it can be misleading. As R^2 scales the error with $\bar{y}$ in the denominator, it punishes fewer errors that are further from the mean. Also, as the variance is dataset dependent, R^2 may not be meaningfully comparable across different datasets [22]. Therefore, we should interpret R^2 results mainly across different methods and in combination with MAPE.

4.3 Cross validation

Cross-validation estimates how accurately our predictive model will perform in practice and if it can generalize on unseen data. Unfortunately, our dataset is relatively small. We still need to test our models on unseen data to validate our models. We split our dataset into training (85% of our data) and testing (15%) sets. We split the data carefully to be able to test the generalization performance of our models properly. It is essential to split the data based on groups of pyrolysis parameters. As mentioned in Chapter 3, we merged biochar stability proxies and pyrolysis conditions with feedstock properties when creating the dataset. By doing this, we increased the dataset size and duplicated some of the output features. Before we split the data, we group the data by pyrolysis parameters and then perform the split. If we performed the splitting without grouping, it would constitute training example leakage. We would be testing on almost identical data to the training set and achieve artificially good results. We also split the data so that all splits have the same amount of missing data proportionally.

k-fold cross validation

To find optimal hyperparameters, we perform a 5-fold cross-validation on the training set (85% of data) with different hyperparameter configurations. With k-fold cross-validation, the training set is split into k smaller same-sized folds. The model is trained using each combination of $k-1$ folds as training data. The resulting model is validated on the remaining part of the data. The final performance is then the average of the values computed for each k [29]. K-fold cross-validation is more computationally expensive but does not waste too much data. This is a significant advantage for our small dataset. For the k-fold split, we also perform the splitting on the grouped dataset. To search through the hyperparameters options, we use random search as it was shown to find good results and be more efficient [30]. For each method, we test our best scoring model from validation data on testing data to see the generalization performance. To choose the best-performing model, we use MAPE. We also refit the best-performing model on the entire training dataset before testing on the test set.

As our dataset is very small, the results can highly oscillate depending on the data split. The test set doesn't have to be a representative sample of the original distribution. To account for this randomness in data splitting, we test the final models 5-times each time with different random seeds used for data splitting. We report the mean with the standard error of MAPE and R^2. Standard error is calculated as $\sigma_e = \frac{\sigma}{\sqrt{n}}$, where σ is the standard deviation and n is the number of samples.

4.4 Methods

This section introduces regression methods that we used to predict biochar carbon stability. These methods include Linear regression, Support Vector Regression, Random forests and Gaussian processes. We predicted a single proxy or all proxies at once. Therefore, the output's dimensions were either one-dimensional or multi-dimensional. Description of the methods is for the multi-output case, as it is the generalization of the single-output case.

4.4.1 Linear regression with L2 regularization

As a simple baseline, we use multi-variate linear least squares with L2 regularization. The objective is to minimize the following objective function:

$$\min_{W} ||Y - XW||_2^2 + \alpha||W||_2^2 \tag{4.5}$$

where Y is the output matrix, X is the input matrix, W are learned weights, and α is the regularization parameter. The objective function represents the linear least squares function with regularization given by the L2 norm. α controls how big a penalty the model imposes on the size of the coefficients. Smaller coefficients mean less complex models. To perform well, linear regression assumes a linear relationship between the input and output variables.

4.4.2 Support Vector Regression

Support Vector Regression (SVR) is a regression counterpart of the more often used Support Vector Machines classification method. SVR fits a hyperplane to the data within a threshold value, the distance between the hyperplane and a boundary line. The Boundary line is formed from Support Vectors, which are data points on either side of the hyperplane that is closest to the hyperplane. The best fit line is the hyperplane with the maximum number of points. The hyperplane is a subspace whose dimension is one less than the original space. To be able to solve non-linear regression, the kernel function transforms the data to a higher dimension before performing the linear fit [31].

The SVR has three primary hyperparameters: kernel, gamma and C. The kernel transforms the data from the original dimension to a dimension where it is easier to fit the data with a hyperplane. Using a kernel trick, we can even go up to an infinite number of dimensions using kernels. This is because kernel functions return the inner product between two points in a suitable feature space [31]. By using the dot product, kernels measure similarity between two points with a little computational cost, even in high-dimensional spaces. The most widely used kernel is Radial Basis Function (RBF). RBF values can be interpreted as a similarity measure as they decrease with squared Euclidean distance between the two feature vectors. The values range between zero (in the limit) and one (when $\mathbf{x} = \mathbf{x'}$). RBF kernel:

$$K(\mathbf{x},\mathbf{x'}) = \exp(-\frac{||\mathbf{x}\text{-}\mathbf{x'}||^2}{2\sigma^2}) \tag{4.6}$$

Other popular kernels are linear, polynomial or sigmoid kernels.

To control the influence distance of each training example, the SVR uses the gamma parameter. High values represent the close influence, and low values far influence. For the mentioned RBF kernel, gamma parameter is: $\gamma = \frac{1}{2\sigma^2}$ [22].

C is a regularization parameter. It controls the effect of the squared L2 penalty. The effect of the regularization is inversely proportional to C [22].

4.4.3 Random forests

Random forests (RF) are an ensemble method that constructs a multitude of decision trees and returns the average prediction of the individual trees. The most crucial idea is that many relatively uncorrelated trees should outperform any of the individual trees [32].

To create uncorrelated trees, RFs employ two techniques: **Bagging (Bootstrap Aggregation)** and **Feature Randomness**. Bagging is a technique of randomly sampling the dataset with replacement when creating each individual tree. As each tree is built with a different subset of data, the tree structures will be different. A tree is built from the tree's root by splitting the training data with learned rules that produce the most separation between the data points in the children's nodes. To force more variation among the trees, each tree can select only from a random subset of features (Feature Randomness).

Hyperparameters that we consider for Random forests are [22]:

- Number of estimators - The number of trees in the forest.
- Maximum depth - The maximum depth of the tree.
- Minimum samples split - The minimum number of samples required to split an internal node.
- Minimum samples leaf - The minimum number of samples required to be at a leaf node.
- Maximum features - The maximum number of features to consider when looking for the best split.
- Maximum samples - The maximum number of samples to draw from input to train each base estimator.

4.4.4 Gaussian processes

With Gaussian processes, we can take a more Bayesian approach to regression. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [33]. Gaussian processes can be seen as an infinite-dimensional generalization of multivariate normal distributions. They are completely specified by their mean function and covariance function. We can write the Gaussian process as:

$$f(\mathbf{x}) \sim \mathrm{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x'})) \tag{4.7}$$

where $m(\mathbf{x})$is the mean function, $k(\mathbf{x}, \mathbf{x'})$ is the covariance function and $f(\mathbf{x})$ is the function we are approximating. We select the mean function and especially the covariance function based on our prior beliefs about the target distribution. It is usual to set the mean of the Gaussian process to be zero. This doesn't mean that the posterior of GP will have a zero mean. We will use a constant mean in our implementation.

Kernels

Most of the behaviour of the function specified by the Gaussian process is defined by the covariance function, which describes how variables affect each other. For example, data points closer to each other should have higher covariance. Covariance between two function values is defined using a kernel function $k(\mathbf{x}, \mathbf{x'})$. The kernel function needs to be positive definite, meaning that it would produce a positive definite matrix K if each element is set using the kernel function: $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. One kernel that is positive definite and proportional to a Gaussian is an RBF, also called the Gaussian kernel, which is in Equation (4.6). This kernel is parameterized by a length scale describing the function's smoothness. Small lengthscale means frequent changes, and large lengthscale characterizes functions that change slowly [34]. Another group of kernels are Matern kernels that generalize the RBF kernel. They have an additional parameter ν that controls the smoothness of the function. Larger ν increases the smoothness [33]. Kernels can be scaled by more parameters and combined. The shape of the modelled function greatly depends on the choice of the hyperparameters.

Gaussian process conditional distribution

The measured data usually include noise. We can model this noise directly with a noise term added to our kernel function. In particular, our dataset has multiple runs with different results for the same parameter configuration. Hence, we can estimate the noise by the standard deviation across the multiple runs. In our models, we will use additive independent identically distributed Gaussian noise with variance σ^2. For the new covariance matrix, we can write:

$$\operatorname{cov}(\mathbf{y}) = K(X,X) + \sigma^2 I \tag{4.8}$$

As said before, the Gaussian process is just a high-dimensional multivariate Gaussian distribution which we define for both known target values $\mathbf{y}$ and function values at test locations $\mathbf{f}_*$ This joint distribution under the prior can be written as [33]:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{4.9}$$

Finally, we just need to express the conditional distribution from the joint. Using the properties of Gaussian distribution, the conditional distribution then becomes:

$$\mathbf{f}_* | X, y, X_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, \operatorname{cov}(\mathbf{f}_*)), \text{ where} \tag{4.10}$$

$$\bar{\mathbf{f}}_* = K(X_*,X)[K(X,X) + \sigma^2 I]^{-1}\mathbf{y} \tag{4.11}$$

$$\operatorname{cov}(\mathbf{f}_*) = K(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma^2 I]^{-1} K(X,X_*) \tag{4.12}$$

Equations 4.10, 4.11 and 4.12 give us the final prediction equations. We can see that the Gaussian process is a parameter-free model. The only parameters to learn are the hyperparameters (denoted by θ), which mostly include the kernel parameters and noise variance σ^2. We set the hyperparameters by using maximum likelihood, which means finding values that make the observations seem probable. The marginal log-likelihood of the standard multivariate Gaussian pdf can be evaluated as [33]:

$$\log p(\mathbf{y}|X,\theta) = -\frac{1}{2}\mathbf{y}^T M^{-}1\mathbf{y} - \frac{1}{2}\log|M| - \frac{N}{2}\log 2\pi \tag{4.13}$$

where $M = K + \sigma^2 I$ is the kernel matrix evaluated at the training inputs plus the variance of the observation noise. We will use gradient-based optimizers to find optimal hyperparameters as the marginal likelihood is easy to differentiate. To prevent overfitting, we will regularize the model by enforcing noise variance and lengthscale constraints. We use the GPytorch library to implement our models [23].

To summarize, we predict a Gaussian distribution for every point with Gaussian processes. The Gaussian mean can be used as a point estimate and standard deviation as an uncertainty estimate. We can find some predictions to be more confident as there might be very similar datapoint present in the train set. On the contrary, the predictions will have higher uncertainty for areas where we did not have enough training points. To define the Gaussian process, we need to specify prior on the mean and covariance function. Gaussian processes scale poorly with the amount of data ($O(N^3)$ computational cost). This isn't an issue for us as our dataset is relatively modest; hence, Gaussian Processes can be evaluated explicitly without needing approximation methods.

4.5 Feature importance

One of the most significant drawbacks of machine learning techniques is their black-box nature and the lack of interpretability. We can get some insight into the models and dataset using feature importance. Feature importance is a class of techniques that calculate scores for input features to a predictive model. This score indicates the relative importance of each feature. A higher score indicates a more significant effect on the model's predictive ability.

4.5.1 Permutation feature importance

Model-agnostic approach to determine which features are most predictive is permutation feature importance. The permutation feature importance is defined as the decrease in a model score when we randomly shuffle a single feature [32]. The decreased score indicates that the feature is important, and the model makes more errors when the feature is shuffled. This technique can be used for any model, as we just permute the input data. It can also be calculated separately for training and testing data. We can then inspect which features are important for the model to generalize. Important features on the training set but not on the testing set might cause the model to overfit. Permutation feature importance can even be negative. This indicates that predictions on the shuffled data are more accurate than on the actual data [32].

4.5.2 Mean Decrease in Impurity

Another option to calculate importance is Mean Decrease in Impurity (MDI). MDI can only be applied to trees. We will apply this technique to the Random Forests. Sometimes also called Gini importance, this technique calculates each feature's importance as the sum over the number of splits (across all tress) that include the feature, proportionally to the number of samples it splits [35]. As described in [36], the importance of a variable is equal to zero if and only if the variable is irrelevant to the predictive model. Also, the non-zero MDI importance of a relevant variable is invariant with respect to the removal or the addition of other irrelevant variables.

Chapter 5

Results and Discussion

In this chapter, we present and discuss the results of our experiments. In Section 5.1, we show and interpret the results on the validation and test sets for each biochar carbon stability proxy and each machine learning method. We compare to the literature and discuss modelling limitations. Each subsection gives details about optimal settings and results for individual machine learning methods. The Section 5.2 analyzes feature importance and Section 5.3 feature dependence.

5.1 Machine learning methods evaluation

We summarize the main results in Table 5.1. For each biochar stability method and every regression method, we report the validation and test mean with standard error of MAPE and R^2. As explained in Chapter 4 about methodology, we mainly focus on the more intuitive and robust MAPE metric. We highlight the best test result for each biochar stability proxy. We achieved the best overall result using Gaussian processes for all stability methods combined with test MAPE $= 13.6\% \pm 4.9\%$. This result can be interpreted that the average difference between the forecasted values and the actual values is 13.6%, which we find to be a good result. Support Vector Regression proved to be the best for predicting stability using Chemical oxidation (MAPE $=$ $15.6\% \pm 5.0\%$). Random forests achieved the best result for the Proximate analysis (MAPE $= 16.4\% \pm 3.4\%$), and also for H:C molar ratio, with MAPE $= 19.8 \pm 4.3$. All of the results have a relatively high standard error suggesting that models' performances can vary depending on the train-test split. This makes it more difficult to say which machine learning method is the best conclusively. Even though SVR was best for Chemical oxidation, other methods achieved similar results. The same applies to

Gaussian processes with All proxies. Random forests had similar performance to Gaussian processes for H:C molar ratio but significantly outperformed other methods for Proximate analysis.

Biochar stability method	**Method**	**Val MAPE (%)**	**Test MAPE (%)**	**Val R^2**	**Test R^2**
Chemical Oxidation	LR	20.8±0.9	18.6±3.1	-4.961±2.921	0.051±0.26
	SVR	19.7±3.6	**15.6±5.0**	-2.025±1.038	-0.611±0.584
	RF	21.3±1.7	17.5±4.1	-1.196±0.291	-0.307±0.313
	GP	10.8±0.6	16.7±7.6	0.485±0.017	-1.57±1.60
Proximate analysis	LR	264.1±3.9	223.2±15.3	-0.107±0.068	0.287±0.022
	SVR	47.3±0.7	35.3±1.7	0.791±0.023	0.92±0.026
	RF	23.1±0.6	**16.4±3.4**	0.916±0.003	0.978±0.005
	GP	27.3±2.5	28.8±5.7	0.822±0.141	0.888±0.038
H:C molar ratio	LR	43.7±2.7	25.1±2.4	-0.457±0.115	-6.0±5.138
	SVR	36.0±4.3	32.6±8.0	-0.374±0.302	-1.241±0.951
	RF	42.9±3.0	**19.8±4.3**	-0.634±0.553	0.114±0.344
	GP	23.6±12.3	21.3±8.6	0.198±0.552	0.192±0.328
All proxies	LR	25.8±2.8	15.4±3.5	-1.017±0.341	0.098±0.101
	SVR	25.2±1.4	14.4±1.4	-1.78±0.475	0.254±0.169
	RF	24.8±0.7	18.6±2.7	-3.021±1.479	0.08±0.056
	GP	20.3±13.2	**13.6±4.9**	-0.487±0.416	0.084±0.396

Table 5.1: MAPE and R^2 results

Literature comparison

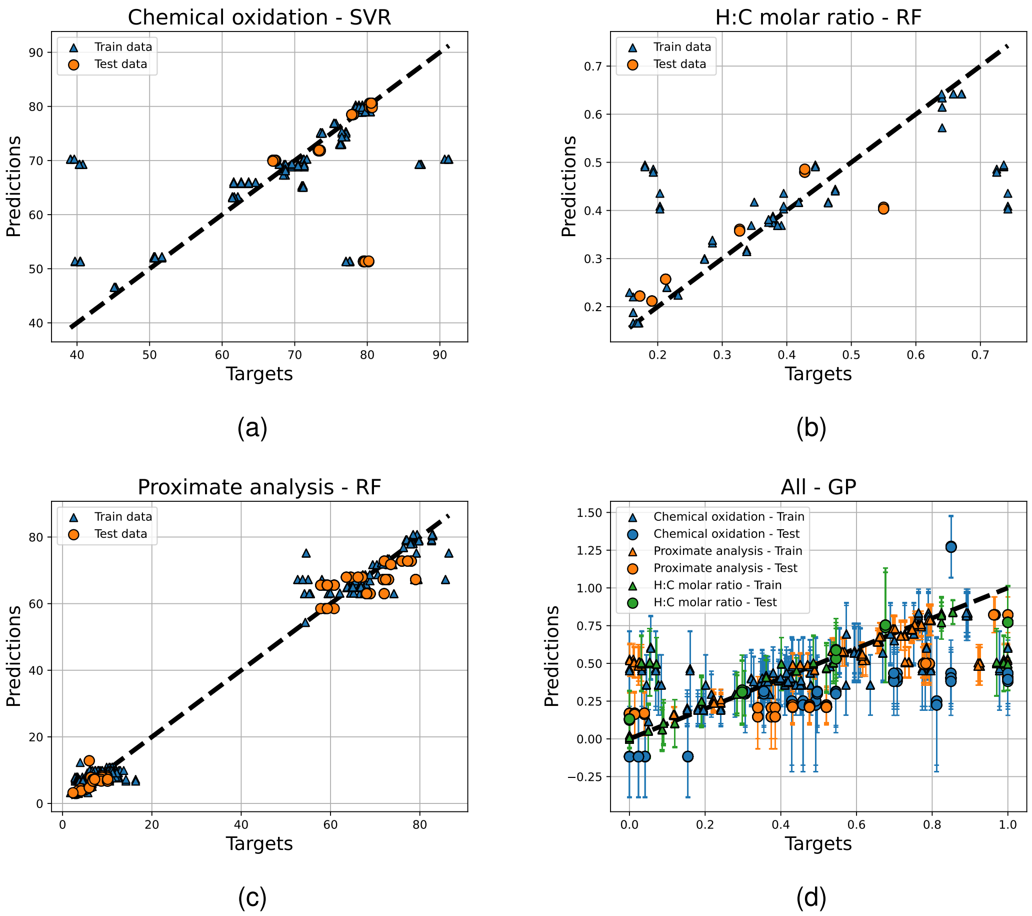
It is hard to compare results with previous studies, as no study predicted biochar carbon stability, and similar studies usually used R^2 (which is dataset dependent) or RMSE (scale dependent) for evaluation. An exception is [5], where authors also used MAPE for evaluation. Their task was a prediction of biochar yield, which is by domain experts considered to be an easier task than the prediction of biochar stability. The best achievement was MAPE = 4.9% using LS-SVM [5]. This is considerably better than our best MAPE of 13.6%, showing that carbon stability prediction is a more challenging task. Another similar study to compare to is [4], where authors predicted biochar yield and c-content of biochar. The dataset was relatively similar, with input features also mainly being biomass feedstock and pyrolysis conditions. The best achievement was R^2 of nearly 0.85 on both regression tasks [4]. Hence our best R^2 result of 0.978 ± 0.005 for Proximate analysis using Random forests would appear to be an outstanding result. But it contrasts with quite high MAPE loss, highlighting the possible inadequacies of R^2. Also, apart from Proximate analysis, other methods haven't achieved as high R^2 results. We believe that too optimistic R^2 for Proximate analysis is caused by the underlying high variance data distribution. From the samples distribution histogram in Figure 3.2, we see no samples are centred around the mean ($\mu = 30$). But we have two areas further from the mean on both sides. Errors for these samples will be considered smaller, resulting in higher and possibly too optimistic R^2.

The main result is that we can predict any biochar carbon stability method with an average absolute percentage error of approximately 13-19%, depending on the method.

Cross plots

To see the relationship between the test predictions and true targets, we show cross plots for the best performing models for each proxy in Figure 5.1. The cross plots show predictions against the true targets. Ideally, all points should be on the diagonal line representing a perfect fit. As stability measurements have different ranges, we scale them to the $[0,1]$ range for all proxies plot. For Gaussian processes, we plot both the mean and standard deviation of the predicted Gaussian distribution. One standard deviation indicates that the prediction should lie within that range with approximately 68% probability.

For all plots, we see that the models sometimes struggle even with predicting the training data (e.g. triangles in the top two plots 5.1a and 5.1b). This suggests that either our models' capacity is low or our data aren't predictive enough. To test this, we tried to overfit our training data by increasing the models' complexity. For example, by increasing the number of trees, depth etc. for RFs or decreasing regularization for SVR. When trying to overfit, the best train MAPE for Proximate analysis was 11.4% and R^2 of 0.986. For H:C molar ratio, MAPE was 13.2% and R^2 of 0.75. We can consider these values to be a lower bound on the achievable error of our models on the testing data. With a quite high train MAPE values, this indicates that our data doesn't include enough distinctive features so that our models would be able to predict every data point accurately. A contributing factor is probably that our samples contained multiple runs with identical parameters, but sometimes the corresponding output values were quite different. No deterministic function can predict all samples correctly with identical inputs but varying outputs. Therefore, we believe that more complex models would not help. Gaussian processes can at least assign higher variance for the corresponding predictions. Aware of this limitation, we still try to find the best model possible.



(a)

(b)

(c)

(d)

Figure 5.1: Cross plots for best models on the training and test data

Looking at the cross plots in Figure 5.1, Random forests in plot 5.1c predicting Proximate analysis appear to model the samples well. We have considerably more data points for Proximate analysis than for other methods. When investigating wrongly predicted outliers visible in Chemical oxidation and H:C molar ratio plots, we found that these samples had a type of feedstock not present in training data. Feedstock types are summarized in A.1. Proximate analysis data contain the widest variety of feedstock. This suggests that many types of feedstock are important for training the models.

In the All - GP subplot, we show all predicted proxies separately. With standard deviation indicating the model's confidence in the prediction, we can see that predictions closer to the true predictions usually have lower deviations indicating higher confidence. This agrees with our previous hypothesis that Gaussian processes can give us a confidence estimate. Standard deviation does not affect MAPE as it is measured from the mean. Depending on our application, we could select a threshold for the standard deviation and discard predictions that exceed the chosen threshold, only keeping the most confident predictions.

5.1.1 Linear regression

We first tried Linear regression without regularization, which resulted in extreme coefficients and deficient performance of the models. High coefficients for some variables were caused by some highly negatively correlated features, where coefficients for these features grew together and cancelled each other out. To account for the problem, we added L2 regularization, which improved the models. We also tested models on just subsets of features. The results are in Table 5.1 and the optimal hyperparameters used are in Table 5.2.

Using a subset of input variables did not improve the performance. Linear regression achieved results close to but slightly worse than other methods. The exception is Proximate analysis, where Linear regression performed considerably worse. This might suggest that Proximate analysis is a more non-linear process. Linear regression, in some cases, achieved even slightly better results than SVR or RF. Noticing not negligible linear relationship, we explored combinations of linear and other kernels for GPs and a linear kernel for SVR.

Biochar stability proxy	**All proxies**	**Chemical Oxidation**	**Proximate analysis**	**H:C ratio**
Input subset	-	-	-	-
α	1.15	2.2	0.1	1.15
Scaler	MinMax	Standard	MinMax	Standard
Imputation	drop	drop	drop	kNN

Table 5.2: Linear regression best hyperparameters

5.1.2 Support Vector Regression

Support Vector Regression performed better than other methods for Chemical Oxidation (MAPE=15.6 ± 5) and almost as good as Gaussian Processes for All proxies (MAPE=14.4 ± 1.4).

We performed hyperparameter optimisation for C, Gamma and kernel. We tested RBF, Sigmoid, Linear and Polynomials kernels. As expected, the best kernel for all methods was RBF. For C and Gamma, we first performed grid search with exponential factors 100, 10, 1,0.1, 0.01, 0.001 as suggested in [37]. We then narrowed the search to more promising areas using random search. For SVR, it was essential to scale the input features first as SVR calculates the distance between the observations. Best configurations found are summarized in Table 5.3.

Biochar stability proxy	**All proxies**	**Chemical Oxidation**	**Proximate analysis**	**H:C ratio**
Kernel	RBF	RBF	RBF	RBF
C	0.1	50	25	50
Gamma	0.05	0.05	2	0.005
Scaler	Standard	MinMax	MinMax	MinMax
Imputation	drop	drop	drop	drop

Table 5.3: SVR best hyperparameters

5.1.3 Random forests

Random forests were the best method both for H:C molar ratio (MAPE=19.8 ± 4.3) and Proximate analysis (MAPE=16.4 ± 3.4), for which it considerably outperformed second best gaussian processes (MAPE=28.8). We can probably attribute the success of random forests to their ability to handle linear and non-linear relationships well and not being susceptible to outliers.

Random forests have many hyperparameters to optimize. Fortunately, they are fast to train with our small dataset, and we can explore many configurations. We used a random search with 800 iterations. Optimal hyperparameters found on the validation set are in Table 5.4. RFs achieved the best results for Proximate analysis with only 9 trees but a maximum depth of 40, deeper than for any other proxy.

Biochar stability proxy	**All proxies**	**Chemical Oxidation**	**Proximate analysis**	**H:C ratio**
Estimators	8	8	9	11
Maximum depth	20	15	40	15
Minimum samples split	7	9	8	9
Minimum samples leaf	2	3	1	2
Maximum features	7	auto	6	auto
Maximum samples	0.8	0.8	0.9	0.9
Scaler	Standard	MinMax	Standard	Standard
Imputation	drop	kNN	drop	kNN

Table 5.4: Random forest best hyperparameters

5.1.4 Gaussian processes

Gaussian processes performed best on All proxies with MAPE of 13.6 ± 4.9, indicating that Gaussian processes might better learn the relationships between the stability methods. Gaussian processes predict Gaussian distribution for each output. We calculate MAPE and R^2 from the mean. As Gaussian processes are non-parametric models, the training includes finding the optimal hyperparameters. Those are mainly kernels with their parameters, noise and lengthscale priors. As with other methods, we first tested various configurations of hyperparameters on the validation set and then tested the best on the hold-out set. Optimal Gaussian process hyperparameters are in Table 5.5.

Kernels

We tested various configurations of RBF, Matern and Linear kernels. Using Linear regression, we found that some linear relationship exists between the inputs and targets. This made us more focused on testing Linear and RBF or Matern kernel combinations. Matern and RBF kernels define more complex covariance functions that are able to capture non-linear relationships, while the linear kernel can easily learn the linear patterns. Matern kernel ($\nu = 2.5$) combined with Linear kernel proved to model best the Chemical oxidation and H:C ratio. ν indicates the complexity of the covariance function with $\nu = 2.5$ corresponding to a twice differentiable function and $\nu = 1.5$ to a less smooth once differentiable function. Matern kernel ($\nu = 1.5$) combined with Linear kernel performed best for Proximate analysis, indicating that the underlying chemical process is more complex and hence Matern kernel with $\nu = 1.5$ is needed. On the other hand, a sum of RBF and Linear kernel was better for All proxies combined, showing that a simpler kernel can perform well when more data are available.

Noise and lengthscale constraints

Because Gaussian processes would quickly overfit our small dataset, we used noise and lengthscale constraints to regularize the model. With no constraint on the noise, the gradient-based optimizer would set the noise to zero, and the model would overfit. We measured the noise on our output data and set the noise constraint near this value. As mentioned in the Data Chapter 3, we have multiple measurements for the same configuration parameters. Therefore, we measured the variance across these runs for training data. After scaling, these values are 0.002, both for Chemical oxidation and Proximate analysis. We don't have multiple runs for the H:C molar ratio, so we set the noise constraint to a smaller value of 0.0001. We test values for the constraint close to the measured noise. We know that the Gaussian process noise values should not be considerably smaller than the measured noise. The possibility of including noise prior in the model is another advantage of GPs. For the lengthscale, we experimented with values in the range of [0,1] to limit the influence of each data point. The best values are in Table 5.5, but we found that lengthscale constraints usually didn't significantly influence the model's performance.

We also experimented with modelling the uncertainty of our training data. This is possible as we also have multiple measurements for the feedstock properties. If we explicitly know the type of uncertainty in our inputs, we can pass that into our kernel [38] using GPyTorch [23]. But initial runs did not show performance improvement. Nevertheless, we find modelling uncertainty of inputs as a possible direction for future investigation.

Biochar stability proxy	**All proxies**	**Chemical Oxidation**	**Proximate analysis**	**H:C ratio**
Kernel	RBF + Linear	MK(2.5) + Linear	MK(1.5) + Linear	MK(2.5) + Linear
Noise constraint	>0.05	>0.002	>0.003	>0.0001
Lengthscale constraint	>0.3	>0.7	>0.7	>0.3
Learning rate	0.1	0.1	0.15	0.1
Scaler	MinMax	Standard	Standard	Standard
Imputation	drop	kNN	drop	drop

Table 5.5: Gaussian process parameters

5.2 Feature importance

We have performed permutation importance analysis on all features for SVR, RF and GP models, both for training and testing sets. The results are in Figure 5.3. For Random forest models, we have also calculated the Mean Decrease in Impurity with MDI results in Figure 5.2. For each biochar carbon stability proxy, we focus on describing the test importance for the method that achieved the best result, as shown in Table 5.1

Pyrolysis temperature is dominantly the most important feature for All proxies, Chemical oxidation and H:C molar ratio. This is expected as the importance of pyrolysis conditions, especially of Pyrolysis temperature, has been shown in [4, 5]. Fixed C is the only other feature at least partially significant for All proxies. As we are predicting carbon stability, the original amount of Fixed C is naturally an important factor. For Chemical oxidation, SVR also focuses on C. Fixed C and N importance values are actually negative. This indicates that shuffled data are more accurate than the real data meaning the feature does not contribute to predictions.

Interestingly, all methods attend to the features more uniformly for Proximate analysis than for other biochar proxies. For other proxies, most of their predictive power is based on just a few most important features. For random forests modelling Proximate analysis, the most important features were Residence time, N %, C % and Remainder %. This agrees both for the permutation importance in Figure 5.3 and MDI importance in Figure 5.2. Residence time data were available only for Proximate analysis. But other methods focused on different features than random forests and, on the contrary, did not give any importance to Residence time. Their performance was lower possibly because they failed to model the Residence time. Gaussian processes weighted other features similarly, putting most importance on Fixed C. Fixed C (measured before pyrolysis) was expected to be important as Proximate analysis estimates carbon stability by the proportion of Fixed C in the biochar (after pyrolysis). The order of importance for C, H, N, and Remainder, which varies among the methods, isn't vital as these features are highly correlated (0.8). Overall, as for Proximate analysis, we have a wider variety of feedstock data, models were perhaps forced to learn more patterns because the temperature or residence time wouldn't be sufficient indicators. The models then may generalize better to new feedstock data.

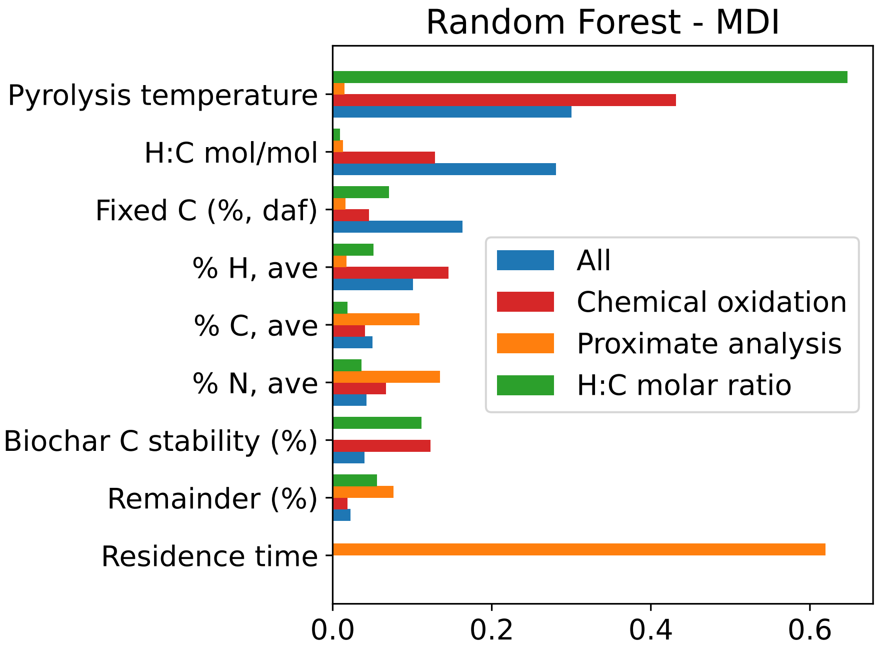


Figure 5.2: Mean Decrease in Impurity of Random forests for every biochar stability method

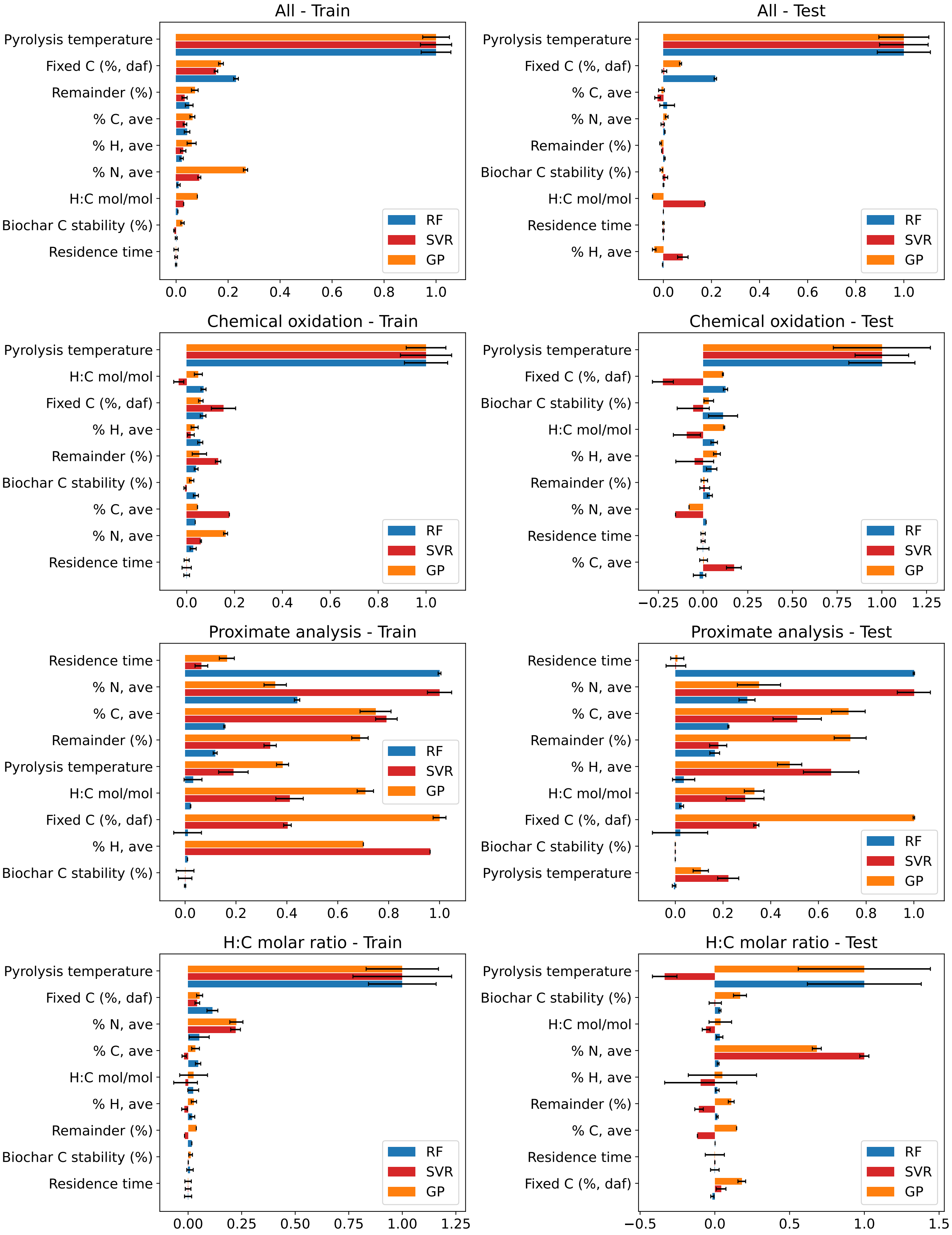


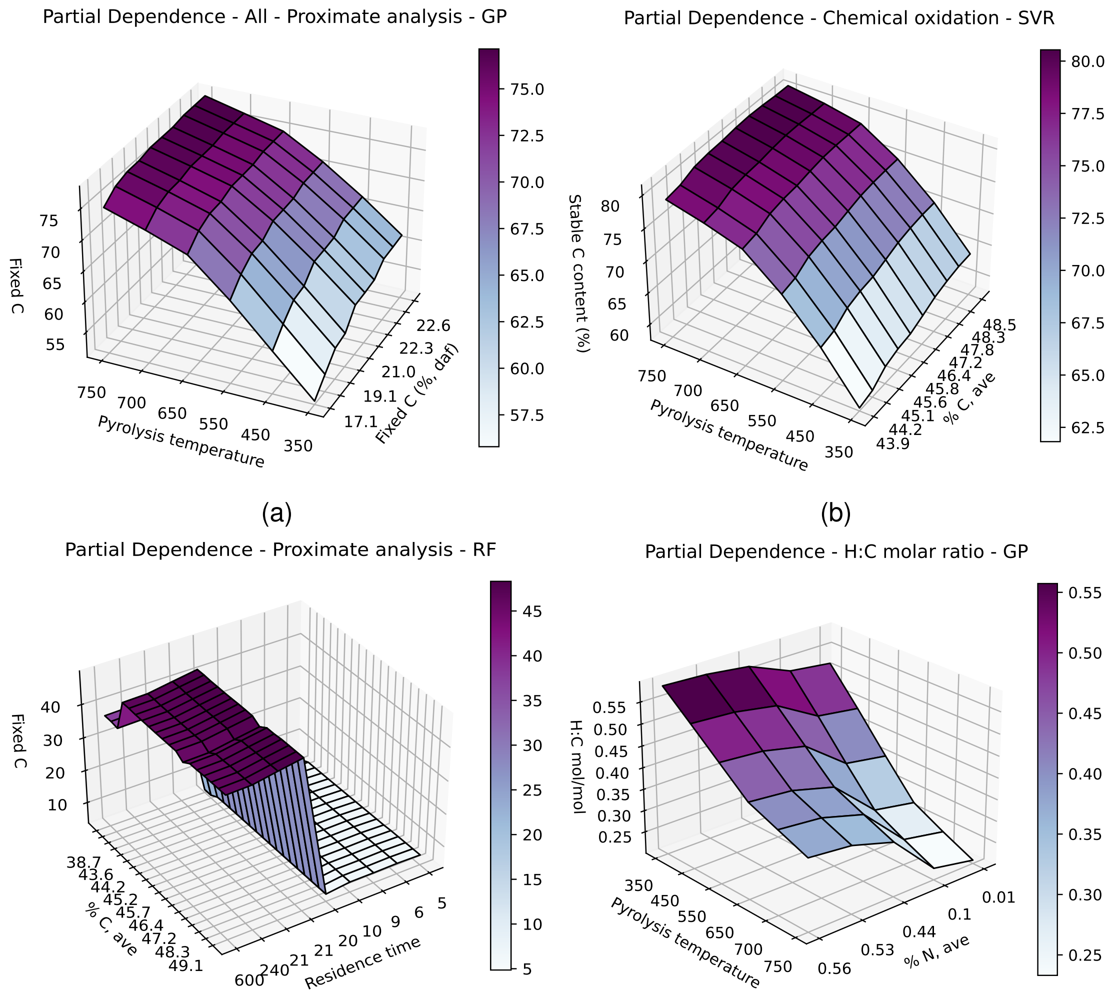
Figure 5.3: Permutation importance for each machine learning method and biochar stability method (left column on training data and right column on the testing data

5.3 Feature dependence

In this section, we plot Partial dependence plots (PDP) for some of the most indicative methods and features to learn their effect on biochar stability. We focus on methods that achieved good results as summarized in Table 5.1.

Partial dependence plots show the dependence between biochar stability proxy and a set of input features. They marginalize over the values of all other input features. We plot multiple PDPs for different pairs of input features in Figure 5.4. For every biochar stability method, we plot a partial dependence plot of the two most important features for the best model. We only plot Proximate analysis in subfigure 5.4a for all stability methods combined.

We show effect of Pyrolysis temperature in the subfigures 5.4a, 5.4b, and 5.4d. A clear linear trend is visible in each plot. Increasing Pyrolysis temperature increases the biochar stability. In the subplot 5.4a, we also see that higher Fixed C of the feedstock contributes to higher Fixed C of the biochar. Both of these results agree with results from [3].



(a)

(b)

(c)

(d)

Figure 5.4: Partial dependence plots of the most important features for the best performing methods for each biochar carbon stability method - z axes show carbon stability metric, x and y axes show input features: 5.4a Pyrolysis temperature (℃) vs. Fixed C (%, daf), 5.4b Pyrolysis temperature (℃) vs. C average (%), 5.4c C average (%) vs. Residence time (min), 5.4d Pyrolysis temperature (min) vs. N average (%)

From the plot 5.4c, it is visible that a short residence time is not enough to create stable biochar. We see a big transition at residence time of 20 minutes. For extreme values of 240 and 600 minutes, the stability is similar. This suggests that a minimal residence time of around 20 minutes is needed, but a considerably longer time does not help.

Subplots 5.4b and 5.4c show that C content has a small effect with higher values slightly increasing stability both measured by Chemical oxidation and Proximate analysis. Higher nitrogen content also appears to increase stability in the plot 5.4d.

Chapter 6

Conclusion

6.1 Main results

This was the first study that used machine learning methods to predict biochar carbon stability. We conclude that predicting biochar carbon stability is possible but more challenging than other tasks such as predicting biochar yield. Our best overall result on the test set was for all biochar carbon stability measures combined with MAPE $= 13.6\% \pm 4.6\%$ using Gaussian processes. The most similar previous work [5] achieved MAPE $= 4.9\%$ using LS-SVM, but on the biochar yield prediction task. Their result is considerably better, but yield prediction is considered to be much easier by the domain experts. We achieved MAPE $= 16.4\% \pm 3.4\%$ using random forests for Proximate analysis (most data with 291 distinct samples, or 1007 samples including multiple runs), For Chemical oxidation and H:C molar ratio, we achieved MAPE $= 15.6 \pm 5.0$ and MAPE $= 19.8 \pm 4.3$, respectively.

Overall, random forests proved to work best on 2 out of 3 individual biochar carbon stability methods, with Support Vector Regression working the best on the remaining one. This agrees with findings of [4], where random forest showed good prediction ability for biochar yield and carbon contents. Gaussian processes were best in modelling all stability methods combined. Best performing kernels were combinations of RBF and Matern kernels with Linear kernels. Gaussian processes' advantage was the ability to input measurement noise into the models and give a confidence estimate that can guide future laboratory experiments. Using confidence estimates, researchers can guide their experiments by deciding when the model is sufficient or when the real experiment should be conducted.

The most predictive feature for Chemical oxidation, H:C molar ratio and All methods combined was pyrolysis temperature. Increasing pyrolysis temperature increased biochar carbon stability. This is supported by previous findings in [3, 4, 5]. For Proximate analysis, features were more uniformly important, with the most important features being Residence time, followed by N, C, and Remainder. We saw from the partial dependence analysis that a minimum residence time of approximately 20 minutes is needed to achieve higher carbon stability.

6.2 Limitations

The main limitation was the amount and distribution of data available, where for H:C molar ratio method, we had only 102 samples, 312 samples for Chemical oxidation, and 1007 samples for Proximate analysis (Figure 3.3). Due to a considerable amount of missing data (8-39%), we had to remove or impute data. Some important features were missing completely for some methods, e.g. residence time proving very important for Proximate analysis but missing for other methods. We saw that even very complex models couldn't achieve better MAPE values than 11.4 % on the training data, suggesting this being a lower bound on the error for our data.

We have also noted the limitations of using R^2 metric, widely used by previous works [4, 5, 17]. We achieved $\mathrm{R}^2 = 0.978 \pm 0.005$ for Proximate analysis, much higher R^2 than for other methods (e.g. H:C molar ratio $\mathrm{R}^2 = 0.114 \pm 0.344$), but similar MAPE. Too optimistic R^2 was caused by the high variance of the data (Figure 3.2) as errors in R^2 calculation are scaled by the dataset mean. Therefore, we used Mean absolute percentage error (MAPE) as the primary metric. We argue that it is more interpretable as a relative error and comparable across methods and datasets.

6.3 Future work

For future work, we emphasise the need for a more comprehensive dataset with more pyrolysis parameters, feedstock types and fewer missing values. From pyrolysis conditions, we had extensive data only for the pyrolysis temperature. We couldn't properly evaluate the effect of heating rate and residence time as we didn't have enough samples with different values. We also stress the need for a wide variety of Feedstock types. As found with feature importance analysis, our models could only use feedstock properties for prediction when many feedstock types were used for training (48 types). The necessity of having a comprehensive dataset was also highlighted in [4].

We recommend using Random forests or Gaussian processes. Random forests proved useful in our and previous works [4, 5, 17] and appeared to work well on a wide range of problems. We have also tested a novel approach using Gaussian processes that performed well. Employing a probabilistic framework seems sensible due to many underlying uncertainties resulting from the chemical measurements. Domain experts can use the Gaussian process model to guide their experiments. As we achieved reasonably good results with linear regression, Bayesian linear regression might also be a good baseline to try. Future works can also investigate propagating the uncertainty through the models, different priors on the input noise or other types of kernels for the Gaussian processes.

Bibliography

[1] Man Kee Lam, Adrian Chun Minh Loy, Suzana Yusup, and Keat Teong Lee. Chapter 9 - biohydrogen production from algae. In Ashok Pandey, S. Venkata Mohan, Jo-Shu Chang, Patrick C. Hallenbeck, and Christian Larroche, editors, *Biohydrogen (Second Edition)*, Biomass, Biofuels, Biochemicals, pages 219–245. Elsevier, second edition edition, 2019.

[2] A Budai, AR Zimmerman, AL Cowie, JBW Webber, BP Singh, B Glaser, CA Masiello, D Andersson, F Shields, J Lehmann, et al. Biochar carbon stability test method: An assessment of methods to determine biochar carbon stability. *International biochar initiative*, pages 1–10, 2013.

[3] Kyle Crombie, Ondřej Mašek, Saran P Sohi, Peter Brownsort, and Andrew Cross. The effect of pyrolysis conditions on biochar stability as determined by three methods. *Gcb Bioenergy*, 5(2):122–131, 2013.

[4] Xinzhe Zhu, Yinan Li, and Xiaonan Wang. Machine learning prediction of biochar yield and carbon contents in biochar based on biomass characteristics and pyrolysis conditions. *Bioresource technology*, 288:121527, 2019.

[5] Hongliang Cao, Ya Xin, and Qiaoxia Yuan. Prediction of biochar yield from cattle manure pyrolysis via least squares support vector machine intelligent approach. *Bioresource technology*, 202:158–164, 2016.

[6] Sasithorn Sunphorka, Benjapon Chalermsinsuwan, and Pornpote Piumsomboon. Artificial neural network model for the prediction of kinetic parameters of biomass pyrolysis from its constituents. *Fuel*, 193:142–158, 2017.

[7] Liuwei Wang, Yong Sik Ok, Daniel CW Tsang, Daniel S Alessi, Jörg Rinklebe, Hailong Wang, Ondřej Mašek, Renjie Hou, David O'Connor, and Deyi Hou. New trends in biochar pyrolysis and modification strategies: feedstock, pyrolysis

conditions, sustainability concerns and implications for soil amendment. *Soil Use and Management*, 36(3):358–386, 2020.

[8] Andrew Cross and Saran P Sohi. A method for screening the relative long-term stability of biochar. *Gcb Bioenergy*, 5(2):215–220, 2013.

[9] Michael Jerry Antal and Morten Grønli. The art, science, and technology of charcoal production. *Industrial & engineering chemistry research*, 42(8):1619–1640, 2003.

[10] Kurt A Spokas. Review of the stability of biochar in soils: predictability of o: C molar ratios. *Carbon management*, 1(2):289–303, 2010.

[11] Jianlong Wang and Shizong Wang. Preparation, modification and environmental application of biochar: A review. *Journal of Cleaner Production*, 227:1002–1022, 2019.

[12] Kathrin Weber and Peter Quicker. Properties of biochar. *Fuel*, 217:240–261, 2018.

[13] James G. Speight. 12 - synthesis gas and the fischer–tropsch process. In James G. Speight, editor, *The Refinery of the Future (Second Edition)*, pages 427–468. Gulf Professional Publishing, second edition edition, 2020.

[14] Douglas Harper. Etymology of photosynthesis. *Online Etymology Dictionary*. `https://www.etymonline.com/word/photosynthesis`; accessed 16 August, 2022.

[15] Stefanie Spears. What is biochar? *Regeneration International*, 2018. `https://regenerationinternational.org/2018/05/16/what-is-biochar`; accessed 16 August, 2022.

[16] Christoph Steiner. Biochar carbon sequestration. *University of Georgia, Biorefining and Carbon Cycling Program, Athens, GA*, 30602, 2008.

[17] Xinzhe Zhu, Xiaonan Wang, and Yong Sik Ok. The application of machine learning methods for prediction of metal sorption onto biochars. *Journal of hazardous materials*, 378:120727, 2019.

[18] Jiangkuan Xing, Kun Luo, Haiou Wang, Zhengwei Gao, and Jianren Fan. A comprehensive study on estimating higher heating value of biomass from proximate

and ultimate analysis with machine learning approaches. *Energy*, 188:116077, 2019.

[19] Ali Yener Mutlu and Ozgun Yucel. An artificial intelligence based approach to predicting syngas composition for downdraft biomass gasification. *Energy*, 165:895–901, 2018.

[20] Changkui Lei, Jun Deng, Kai Cao, Li Ma, Yang Xiao, and Lifeng Ren. A random forest approach for predicting coal spontaneous combustion. *Fuel*, 223:63–73, 2018.

[21] Jie Li, Lanjia Pan, Manu Suvarna, Yen Wah Tong, and Xiaonan Wang. Fuel properties of hydrochar and pyrochar: Prediction and exploration with machine learning. *Applied Energy*, 269:115166, 2020.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

[24] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[25] Anil Jadhav, Dhanya Pramod, and Krishnan Ramanathan. Comparison of performance of data imputation methods for numeric dataset. *Applied Artificial Intelligence*, 33(10):913–933, 2019.

[26] Gustavo EAPA Batista and Maria Carolina Monard. An analysis of four missing data treatment methods for supervised learning. *Applied artificial intelligence*, 17(5-6):519–533, 2003.

[27] Ian R White, Patrick Royston, and Angela M Wood. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, 30(4):377–399, 2011.

[28] Jay L Devore. *Probability and Statistics for Engineering and the Sciences*. 2011.

[29] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133, 1974.

[30] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[31] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[32] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[33] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[34] Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

[35] Leo Breiman. Manual on setting up, using, and understanding random forests v3. *Statistics Department University of California Berkeley*, 2002.

[36] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. *Advances in neural information processing systems*, 26, 2013.

[37] Koen Smets, Brigitte Verdonk, and Elsa M Jordaan. Evaluation of performance measures for svr hyperparameter selection. In *2007 International Joint Conference on Neural Networks*, pages 637–642. IEEE, 2007.

[38] Agathe Girard and Roderick Murray-Smith. Learning a gaussian process model with uncertain inputs. *Department of Computing Science, University of Glasgow, Tech. Rep. TR-2003-144*, 2003.

Appendix A

Data - Feedstock types

Feedstock name	H:C molar ratio count	Chemical oxidation count	Proximate analysis count
Untreated Misconthus	18	54	78
Miscanthus (1% potassium doped)	9	27	36
Miscanthus (2% potassium doped)	9	27	36
Miscanthus (Cs+ doped)	9	27	36
Miscanthus (Na+ doped)	9	27	36
Untreated Willow Chip	9	27	39
Washed Miscanthus	9	27	9
Washed Willow Chip	9	27	9
Willow chip (potassium doped)	9	27	39
Miscanthus Chip	3	9	12
Miscanthus Chip with Spray Quenching	3	9	3
Potassium Doped Miscanthus Chip	3	9	12
Willow Chip with Spray Quenching	3	9	3
Mixed Willow:Bonemeal (4:1)		6	0
Wheat Straw			154
Coffee Grounds MA - I			75
Softwood Pellets			64
Anaerobic Digestate			48
Sewage Sludge			42
Coffee Grounds MA - II			36
Rice Husk			30
Coffee Grounds SC			18
Pure Pine			17
MB			13
SBP			12
Oilseed Rape Straw Pellet (OSR)			12

Mix Straw Pellets	11
Softwood pellets (SWP)	11
Macrocystis Pyrifera-MBL Residues	9
Ascophyllum Nodosum-MBL Residues	9
Laminaria Hyperborea-MBL Residues	9
Slaughterhouse waste	9
Miscanthus Pellets	8
Palm Kernel Meal	6
Scottish Mule Wash Sheep fleece	6
Palm Kernel Shell	6
Empty Fruit Bunch	6
Wood Pellets	6
GS	4
Scottish Black Face Sheep Fleece	3
Texel Wash Sheep fleece	3
Miscanthus (potassium doped)	3
Coffee Grounds	3
Blue Face Sheep Fleece	3
W+B20%	2
Wheat Pellets	2
Bone meal alone	2
Willow alone	2

Table A.1: Dataset - Feedstock count