# Mapping soil organic carbon using convolutional neural networks and global data

*MSc Thesis Soil Geography and Landscape*

R.J.M. van Heumen

Wageningen, August 2019

# Mapping soil organic carbon using convolutional neural networks and global data

MSc Thesis for the partial fulfillment of the MSc Earth and Environment, specialisation: Soil Geography and Earth Surface Dynamics

Wageningen, August 2019

**Thesis course code:**

SGL-80436 - MSc Thesis Soil Geography and Landscape

**Student:**

R.J.M. (Rik) van Heumen
Registration number: 950529332090

**Supervisors:**

Prof. dr. ir. G.B.M. (Gerard) Heuvelink
Soil Geography and Landscape Group, Wageningen University and Research
ISRIC world soil information

dr. ir. Laura Poggio
ISRIC world soil information

**Examiners:**

dr. ir. V.L. (Titia) Mulder
Soil Geography and Landscape Group, Wageningen University and Research

dr. ir. J.J. (Jetse) Stoorvogel
Soil Geography and Landscape Group, Wageningen University and Research

# Abstract

Soil organic carbon is an important regulator of global climate and soil quality and an important aspect in the current climate change mitigation debate. To assess the current state of soil organic carbon accurate maps of the distribution are needed. Recently machine learning replaced more traditional regression kriging techniques in the application of digital soil mapping. While the machine learning models often outperform the traditional methods on accuracy, most of them do not take spatial context into account. To overcome this limitation convolutional neural networks were recently introduced for digital soil mapping. These models were originally designed for automatic image recognition and have the added benefit that they use the spatial structure of the input. This research continues the development of this very new technique. A common critique of machine learning methods is that they are black boxes, which makes it difficult to understand their functioning. The Local Interpretable Model agnostic Explanations algorithm was proposed to open this black box and let the user judge the model performance. LIME is mainly designed for classification, but also works for regression tasks. This research is the first application of LIME for digital soil mapping.

The goal of this research was to implement a convolutional neural network for the spatial modeling of soil organic carbon, compare the accuracy with a random forest and assess the usefulness of LIME to open the black box. To obtain this goal a selection of covariates was made using a recursive feature elimination approach, after which several convolutional neural network model architectures were run in a 10-fold cross validation to select the most suitable one. At last the same selection of covariates was also fed to the LIME algorithm.

The results of this research show that the convolutional neural network is outperformed by a random forest when modeling SOC concentrations in Argentina. However, some interesting relations between the chosen activation function, number of convolutional layers, window sizes and the accuracy of the model were found. These findings can be used in future research to design a more effective convolutional neural network. The LIME algorithm did unfortunately not work with the convolutional neural network, because of issues with the structure of the input. The algorithm did, however, work for the random forest model. Some minor differences in covariate importance between a selection of points could be seen. But as the algorithm is mostly used for classification tasks, the interpretation of the results proved challenging for a regression task.

Convolutional neural networks are shown to be a promising technique for digital soil mapping, but their implementation needs further improvements before they can be considered fully operational for digital soil mapping applications. The LIME algorithm did unfortunately not prove very useful in explaining the model performance for this digital soil mapping task.

# Preface

I could not have finished this thesis without the help of several people. First of all I would like to thank my supervisors Laura Poggio and Gerard Heuvelink for their continuous support, advice, help and time. Their critical feedback was exactly what I needed to move forward at times and to push me in the right direction when I needed a push. A special thanks goes to Laura for helping me with the modeling and IT-related difficulties.
I also would like to thank ISRIC as an organization for providing my with all the facilities I used during my thesis and would like to thank the ISRIC staff and guest researchers for welcoming me into the team. Having these facilities and a nice working environment helped me in focusing my attention to this thesis research. At last I would like to thank Titia Mulder and Jetse Stoorvogel for opening up time in their agendas to be an examiner for this thesis.

Even though at times it was a difficult process, I am very grateful for the opportunity that I received. I have learned tremendous amounts about modeling, digital soil mapping and data analysis. All skills that I hopefully can use for the rest of my career.

Thanks!

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

### 1.1.1 The importance of soil organic carbon

Globally 1500 Pg of soil organic carbon (SOC) are stored in the first meter of the soil. This massive storage of SOC plays a major role in the regulation of the global climate (Jobbágy and Jackson, 2000). The flux of $CO_2$ coming from the soil is one of the biggest fluxes of the global carbon cycle (Schlesinger and Andrews, 2000). SOC is also an important indicator of soil quality, agronomic productivity and sustainability as it influences important physical, chemical and biological soil quality processes (Reeves, 1997). Climate change, land use change and mismanagement of agricultural land can degrade the SOC pool and consequently reduce soil quality and release large amounts of $CO_2$ into the atmosphere (Lal, 2004a). SOC is currently receiving a lot attention for its role in mitigating climate change. In 2015 the "4 per 1000" initiative started to demonstrate the potential of SOC to mitigate climate change and ensure food securite (Minasny et al., 2017), a recent article promoted eight different ways of stimulating carbon sequestration in soils to meet the Paris climate agreements (Rumpel et al., 2018) and a new index of calculating how a change in land use contributes to the global capacity to store carbon was proposed (Searchinger et al., 2018). A global increase of the carbon content in soils by just a few parts per thousand would remove around 3-4 gigatonnes of $CO_2$ from the atmosphere, which is equivalent to the total annual fossil-fuel emission of the European Union (Chabbi et al., 2017). By changing to recommended management practices (RMP) the influx of SOC into the soil can exceed the outflux of $CO_2$ to the atmosphere. Changing from conventional plow tillage to conservation tillage is one of the most proposed RMPs (e.g. Follett, 2001; Lal, 2004b; West and Post, 2002; Lal, 2005). These changes will mitigate climate change and maintain soil quality (Follett, 2001).

### 1.1.2 Digital Soil Mapping

Because of the crucial role of SOC in both climate regulation and soil quality status there is a need for more accurate maps of SOC carbon storage in global soils. Modeling studies can help provide this information. Mechanistic models such as RothC provide information on soil organic carbon storage and dynamics by modeling the turnover rate of SOC (Coleman and Jenkinson, 1996). Statistical methods such as digital soil mapping (DSM) focus on the correlation between SOC and environmental explanatory variables, i.e. covariates. Most of the DSM methods are built upon the CLORPT framework of soil forming factors (Jenny, 1994):

$$S = f(cl, o, r, p, t)$$

where S is the soil property of interest, cl is climate, o are organisms, r is relief, p is parent material and t is time. Once the function $f$ is chosen and its parameters are calibrated, predictions of S are made for each location in the area of interest. Later a new framework was proposed that adds soil type, spatial location and spatial errors into the equation. This is called the scorpan framework (McBratney et al., 2003). An example of a commonly used DSM techniques techniques is multiple linear regression (Lamichhane et al., 2019). Linear correlations between the covariates and the variable of interest are calculated and used to derive a trend that predicts SOC from the covariates. The residuals of the trend have a zero mean but a non-zero variance. If these residuals are spatially correlated, as characterized by a variogram, then multiple linear regression may be extended to regression kriging (Hengl et al., 2004).

A lot of the methods used in digital soil mapping are based on linear relations (Webster and Oliver, 2007). Recent developments in computer science have, however, made it possible to implement machine learning techniques for DSM (Lamichhane et al., 2019). These machine learning methods can deal with complex correlations and non-linear relations, which makes them excellent for tracing relations between dependent and explanatory variables and making predictions (Goodfellow et al., 2016). Some examples of these techniques are random forests, neural networks and support vector machines (Heung et al., 2016). Random forests are found to outperform multiple linear regression and even other machine learning techniques in most studies (Lamichhane et al., 2019).

### 1.1.3 Artificial Neural Networks

An artificial neural network is one of the many machine learning models. The first concepts of artificial neural networks (ANN) already date back to 1943 as a method to replicate the human brain (McCulloch and Pitts, 1943). Neural networks are built up from connected neurons. The most basic ANN consists of one input layer, one hidden layer and one output layer (Figure 1.1).
A hidden layer in a neural network makes a decision using a non-linear activation function. Each connection between the individual neurons gets assigned a weight and bias that are fed to the activation function. The combination of the non-linear activation function with these weights and biases of the input defines the signal that a neuron outputs. These weights and biases are recurrently updated during the training of the model to end up with the most accurate results at the output layer. The model does this by minimizing a difference function between model predictions and observations, e.g the mean squared error, in a validation set of the data (Goodfellow et al., 2016).

An extension of ANN models is called deep learning. A neural network gets deep when it used more than one hidden layer between the input and the output layer (Goodfellow et al., 2016). Examples of such deep learning models are convolutional neural networks. Convolutional neural networks are specifically developed to process data that have a grid-like topology and are often

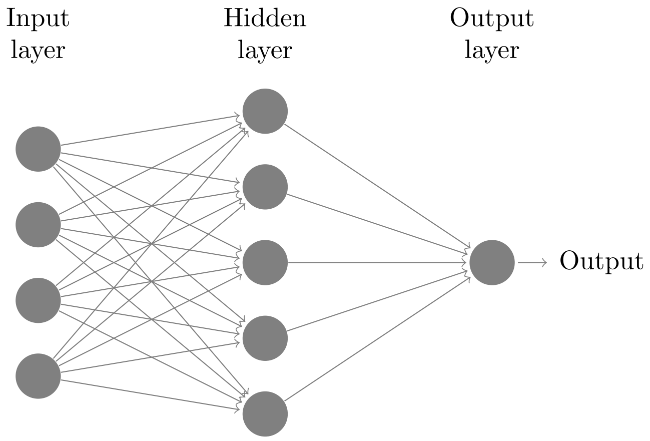


Figure 1.1: Simple representation of a neural network.

used for image classification tasks (Goodfellow et al., 2016). A convolutional neural network has several convolution layers that make use of a filter convolving (sliding) over the image to detect and store patterns in the data. Pooling layers are often added after a convolution layer to merge similar features into one. This reduces the dimension of the representation and reduces noise (LeCun et al., 2015). Recently it is shown that CNN's can also be applied for digital soil mapping, as spatial data also come in the form of grids (Padarian et al., 2019; Wadoux et al., 2019; Wadoux, 2019). To use convolutional neural networks for digital soil mapping windows, i.e. small subsets, of the covariates are created. Within these windows the convolution layers look for spatial information and patterns. The convolutional neural network then predicts the point value of the center point of the window from that information.

### 1.1.4 Opening the black box

In general ML models, and artificial neural networks in particular, are considered 'black boxes'. This means that while the accuracy of the model can be evaluated and may be higher than that of other models, the exact functioning of the model remains unknown. We do not learn about the underlying mechanisms behind the predictions, as the model does not give much insight about how input and output are related. Insight in the functioning of the model can provide information to the user to explain more about the modelled processes from these mechanisms. Insight in model processes at a local scale provides the opportunity to evaluate if the model predictions make sense when evaluating them with expert knowledge. In other words, is it logical that certain covariates are more important at one point compared to a other point. Ribeiro et al. (2016) recently developed the 'local interpretable model-agnostic explanations' (LIME) algorithm to provide insight into the structure of machine learning models. The LIME algorithm uses the assumption that every model can locally be approximated by simpler models. Analyzing these simpler models might provide the insight in machine learning models that we are after.

## 1.2 Problem statement

Kriging methods are proven techniques for spatial modeling, which take spatial correlation into account. However, these techniques do have some limitations and are often suboptimal to machine learning techniques when it comes to the achieved prediction accuracy. Recently machine learning techniques are therefore more often used, but while they perform well they lack the use of spatial correlation and structure in their predictions. Another issue with machine learning techniques is that they are black boxes. Their functioning is difficult to understand and it is difficult to learn something about the underlying processes from these models.

## 1.3 Objective and research questions

This study aims to implement a convolutional neural network for the spatial modeling of soil organic carbon concentrations, using global data with a resolution of 250 m. After this, the added added value of a CNN compared to other machine learning techniques will be investigated and an attempt will be made to learn about the mechanistic processes behind machine learning models by opening the ‘black box’. To achieve this objective Argentina will be used as a case study and a total of three research questions will be answered:

- What is the most suitable convolutional neural network architecture to map the spatial distribution of soil organic carbon on a country scale using global data?
- How does the prediction accuracy of a convolutional neural network compare to that of a random forest when applied to mapping the spatial distribution of topsoil SOC?
- How can the ’Local Interpretable Model-agnostic Explanations’ algorithm be used to provide insight into the structure, functioning and decisions of machine learning soil organic carbon prediction models?

# Chapter 2

# Argentina case study data

## 2.1 Description of study area

Argentina was used as a case study in this research. Argentina is the $8^{th}$ largest country in the world, it has a large variety of climate, geology and ecology and thus a large variety of soils (Heuvelink et al., 2018). This variety is also reflected in the spatial distribution of soil organic carbon (Figure 2.1). Large parts of Argentina are arid and semi-arid. The soils found in these areas, mostly Aridisols and Entisols, contain low SOC concentrations due to the dry conditions. The eastern part of the country receives more precipitation and elatively high concentrations of SOC can be found there. The dominant soil type that can be found in this region are Mollisols. Mollisols are also found in the mountainous area in the South-West and far South of Argentina. A lot of SOC is built up in the soils of this region due to low temperatures and high precipitation (Heuvelink et al., 2018). Argentina has a total land surface area of 2.78 million km$^2$. Of this area 60% is taken up by natural and semi-natural terrestrial vegetation, 20% by cultivated and managed land and 20% by natural or semi-natural aquatic or regularly flooded vegetation. However, the country is heavily affected by land use change, with a loss of more than 30,000 km$^2$ of forest between the years 2007 and 2016 (Heuvelink et al., 2018).

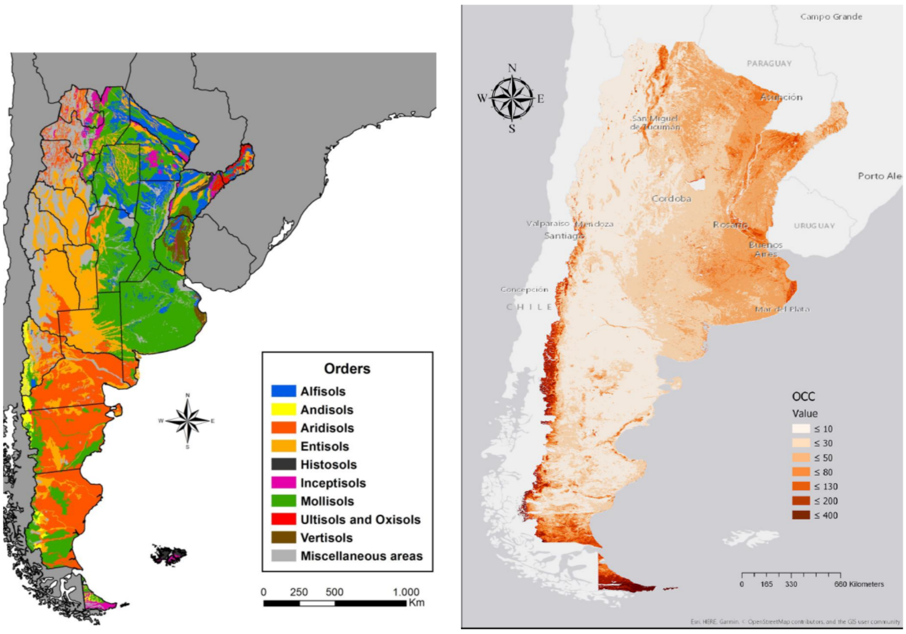


Figure 2.1: Soil types of Argentina (modified from Rodríguez et al., 2019) (left). SoilGrids map of the distribution of SOC in Argentina (map derived from SoilGrids (Hengl et al., 2017)) (right).

## 2.2 data description and pre-processing

### 2.2.1 Soil data

A large set of soil observations (n >17,000) was available for Argentina. The observations were collected during a relatively long period (1955 - 2017) and cover almost the entire country. The highest density of observations is found in the area South of Buenos Aires. Because soil organic carbon carbon concentrations change over time (Mann, 1986) and covariates derived from MODIS (see section below) are only available after 2000 a shorter period from 2002-2015 was used in this study. The point dataset was cleaned by filtering for the years 2002-2015, profiles that did not contain any organic carbon data were removed and the weighted average for the top 30 cm of the soil was calculated. For the convolutional neural network buffers have to be drawn around the point observations into which the covariate data are extracted (Chapter 3). If points are to close to the border these buffers will stretch over the borders, which can cause problems with the modeling. To tackle this problem a few points that were to close to the border were removed from the dataset. At the end a dataset of 1892 point observations was available for the modeling (Figure 2.2).

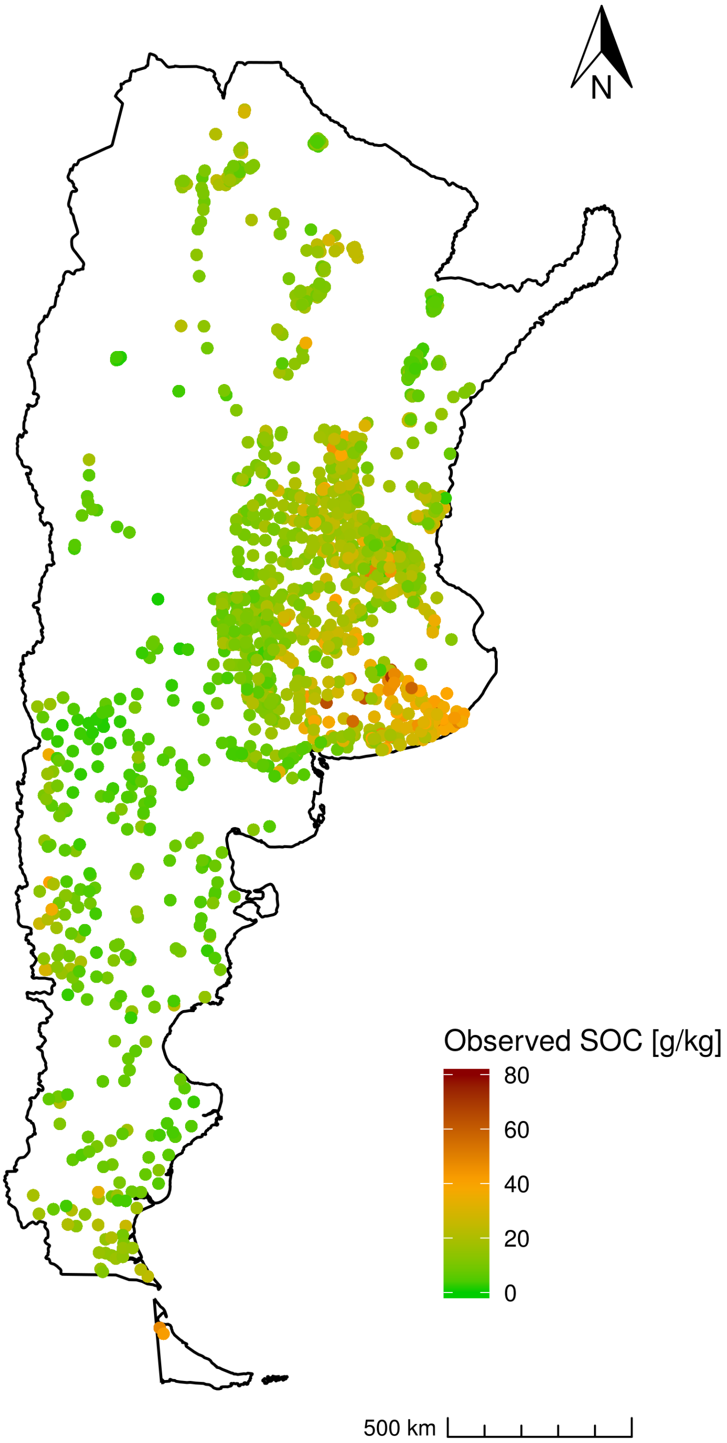


Figure 2.2: Spatial distribution of the topsoil (0 – 30 cm) organic carbon measurements used in this research (g/kg), n = 1892.

The distribution of the 0 - 30 cm soil carbon concentrations is heavily skewed to the right (Figure 2.3). The SOC data have a mean value of 16.16 (g/kg), a median value of 14.36 (g/kg), a minimum value of 0.36 (g/kg) and a maximum value of 81 (g/kg) (Table 2.1). The SOC values were log-transformed so that they better match a normal distribution. After this the values were normalized between 0 and 1. This transformation and normalization helps with model performance of neural networks (Sola and Sevilla, 1997). After predictions the values were back transformed for more easy interpretation. However, a back transformation of the log normal predictions gives the median instead of the mean back which can cause some bias in the results (Yamamoto, 2007).

Table 2.1: Summary statistics of 0-30 cm soil organic carbon concentrations [g/kg].

| Summary statistics | Value |
|---|---|
| Number of samples | 1892 |
| Minimum | 0.36 |
| 1st quantile | 9.30 |
| Median | 14.36 |
| Mean | 16.16 |
| 3rd quantile | 20.60 |
| Maximum | 81.00 |
| St. dev. | 10.08 |

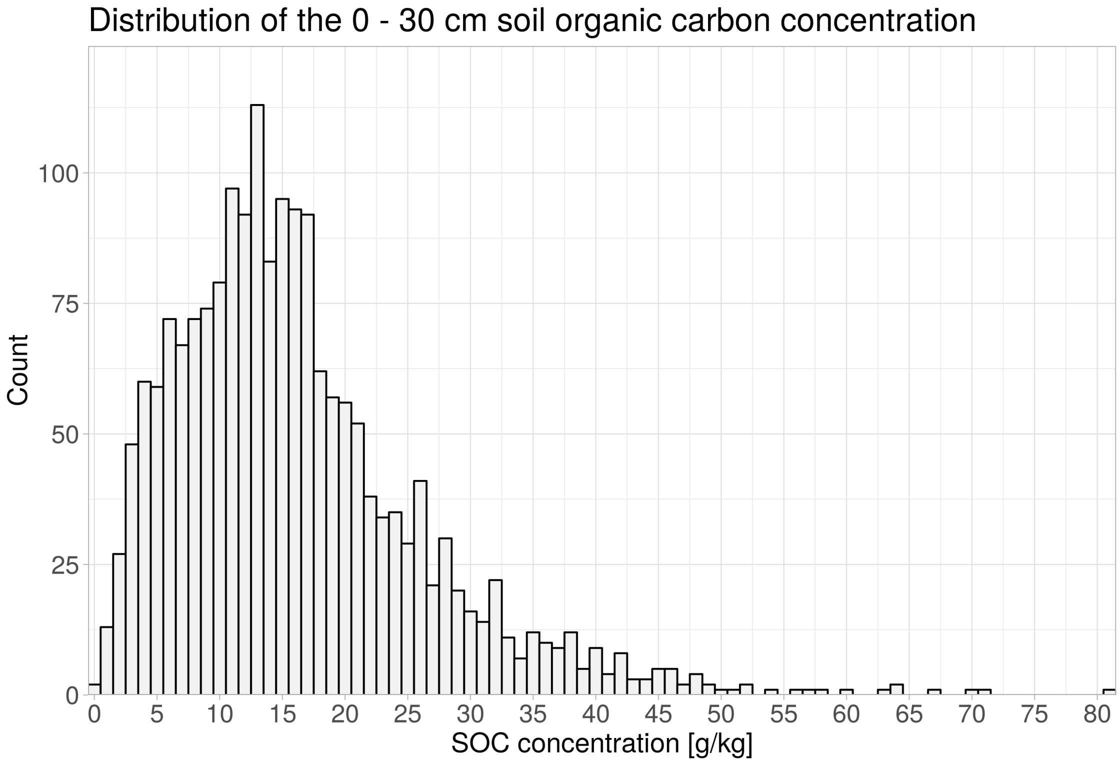


Figure 2.3: Histogram of the topsoil (0 – 30 cm) SOC concentrations (g/kg) for Argentina. The distribution is skewed to the right

### 2.2.2 Covariates

A large set of covariates related to the soil forming factors was available at a spatial resolution of 250m. This set contains data on global climate, geology, land use/cover, net primary productivity, reflectance, soil, terrain and vegetation indices. The main source of the data was ISRIC's 'WorldGrids' covariate repository. This repository contains covariates, e.g. MODIS satellite data, that were used to produce SoilGrids250m (Hengl et al., 2017). Heuvelink et al. (2019) further supplemented this dataset with global data from the GIMMS (Global Inventory Modeling and Mapping Studies) NDVI data set that is derived from imagery obtained from the Advanced Very High Resolution Radiometer (AVHRR) instrument on board the NOAA satellite series (Tucker et al., 2005; Vermote et al., 2014; Pizon et al., 2005). Global Inventory Modeling and Mapping Studies Several land use layers from The HYDE (History Database of the Global Environment) were also added to the dataset (Goldewijk et al., 2017). In total 352 different covariate layers were available for the modeling.

The covariates were already preprocessed by ISRIC. All layers were resampled to a resolution of 250 m and reprojected to an equal-area projection (Kempen et al., 2018). The NDVI data was averaged in four groups of average NDVI values of three months for each year (Heuvelink et al., 2019) and were further averaged over the entire period from 2002 to 2015. To efficiently handle the 352 large raster layers a dataset containing these layers was created in GRASS GIS (GRASS Development Team, 2019). This bypassed the problem of having to load all raster layers into R, which is relatively slow and memory intensive. To further lower the computational intensity, the data that had a global extent were clipped to the extent of Argentina. GDAL (Geospatial Data Abstraction Library) was used to convert the raster layers to text files, where the extent could be set with relatively low computational intensity (Greenberg and Mattiuzzi, 2018). These GDAL text files were then loaded into the GRASS database.

# Chapter 3

# Methodology

## 3.1 Covariate selection

A total of 352 covariate layers were available for the modeling. To reduce the number of covariates and select covariates to be used in the modeling, two steps were performed. The first step was to perform a Pearson correlation between all covariates and remove some of the covariates a with high correlation, because of the likelihood that they provide very similar information. Each pair of variables with a pair-wise correlation higher than 0.9 was selected. Of this pair, the variable with the largest mean absolute correlation with all other covariates was removed. This process was repeated iteratively until there were no more pairs of covariates with an absolute correlation greater than 0.9. This was done using the *findCorrelation* function from the *caret* package in R (Jed Wing et al., 2019). After this initial reduction, the remaining covariates were used in a recursive feature elimination algorithm (Guyon et al., 2002). The recursive feature elimination selected the most important predictors out of a large group of covariates. It gave the optimal number and selection of covariates, to provide the highest possible accuracy. Because the RFE is purely statistical and optimized for the random forest model it might not select the best performing covariates for the convolutional neural network. In addition, a selection of covariates was therefore made based on pedological knowledge and factors influencing soil organic carbon concentrations in soils (Jobbágy and Jackson, 2000; Parton et al., 1987). Table 3.1 shows the selected pedological covariates.

Table 3.1: List of manually selected covariates based on pedological knowledge.

| Code | Description |
|---|---|
| NDVI1 | Normalized Difference Vegetation Index January, February, March |
| NDVI4 | Normalized Difference Vegetation Index April, May, June |
| NDVI7 | Normalized Difference Vegetation Index July, August, September |
| NDVI10 | Normalized Difference Vegetation Index October, November, December |
| MOR ENV DEME | Digital elevation model |
| MOR MRG SLP | Terrain slope based on DEMMRG5 derived in SAGA GIS and expressed in radians x 100. |
| MOR MRG TPI | Topographic Position Index |
| MOR MRG TWI | Topgraphic Wetness Index |
| VEG MOD EVI01AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months January and February |
| VEG MOD EVI03AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months March and April |
| VEG MOD EVI05AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months May and June |
| VEG MOD EVI07AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months July and August |
| VEG MOD EVI09AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months September and October |
| VEG MOD EVI11AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months November and December |
| CLM MOD LSTDYRAVG | Long-term averaged mean annual surface temperature (daytime) MODIS |
| CLM CHE PYRSUM | Total annual precipitation at 1 km (based on CHELSA climate surfaces) |

## 3.2 Building the convolutional neural network

A convolutional neural network mostly consists of four major layers: convolution, pooling, flatten and fully connected (also called dense) layers. The convolution layers look for certain features and patterns in the input data and pass their results on to the pooling layers. These layers join features to optimize model performance and reduce noise. The flatten layers convert the matrix output of the convolutional and pooling layers to a vector of values that is then fed to a fully connected layer, which gives a weighted sum of the input and pass this on (Goodfellow et al., 2016). Also several dropout layers can be used. These dropout layers randomly disconnect certain connections between the neurons, to reduce the risk of overfitting (Srivastava et al., 2014).

All machine learning models need to be trained. During this step the model will "learn" to achieve the best prediction possible. An optimizer, e.g. the *Adam* optimizer, is used to minimize a loss function by updating the weights used within the model (Kingma and Ba, 2014). Before training a neural network the entire dataset is split into a training and a test set. The training set is fed to the neural network where it is internally split in 80% training and 20% validation set. The training data is used to optimize the weights of the model. It does so by updating the weights every epoch to minimize the loss function. The validation set is used to detect overfitting and help set the hyperparameters. The training runs for a certain number of epochs. One epoch is one training iteration over the entire training data set. During one epoch the data can be put in to the model at once or in minibatches. Another parameter that effects the training is the learning rate of the optimizer. The learning rate determines how big the changes in weights are (Goodfellow et al., 2016). After the model is trained the independent test set is used to assess the accuracy of the model on data that were not previously used in the model.

### 3.2.1 Preparing the input

A spatial convolutional neural network takes a four dimensional array of shape n x w x h x b as input, where *n* is the number of data points, *w* is the width of the window, *h* is the height of the window and *b* is the number of covariates. To create this array first all covariates were scaled to have a mean of 0 and a standard deviation of 1. Next, square buffers of *w* by *h* pixels were drawn around each point. The values of the selected covariate layers, were extracted to these buffers to form a 3D matrix of size w x h x b for each point. These 3D matrices where then joined together to form the final 4D matrix. The extraction of the raster data was done using the velox package in R (Hunziker, 2017). This package runs the extraction in C++, which has significant performance benefits over internal R speeds.

### 3.2.2 Optimal model selection

To implement the convolutional neural network the R package 'Keras' was used (Allaire and Chollet, 2019). Keras is based on the extensive tensorflow machine learning library and provides functions that are necessary to efficiently build a CNN (Allaire and Chollet, 2019). The convolutional neural network as used in Wadoux et al. (2019) was used as a starting point for the convolutional neural networks in this research. Many parameters can be tuned when building a convolutional neural network. Firstly, an initial hyperparameter optimization using the build in keras hyperparameter optimization was used to set appropriate setting for the number of neurons a layer, the size of the filter used in convolutional layers and the dropout rates in the dropout layers. For this the model was run with different combinations of settings for these hyperparameters (Allaire and Chollet, 2019). The combination of the initial model, with the optimized hyperparameters is given in table 3.2. Further a batch size of 32 was used, the learning rate was set to 0.0001 and the *Adam* optimizer (Kingma and Ba, 2014) was used to minimize the mean squared error during training.

After this initial hyperparameter optimization 24 models were designed with different combinations of several major parameters (Table 3.3): Three window sizes were tested, these are the windows within which the CNN looks for patterns. The data were either augmented or not augmented. Data augmentation is often used in machine learning to get more data for model training and should in general increase the accuracy. Augmenting was in this case done by rotating the covariate matrices in the array by 90, 180 and 270 degrees (Padarian et al., 2019). Two different activation functions were used. These activations function determine the signal that a neuron sends to the next neuron. Relu (Rectified Linear Unit) is a widely used activation function for neural networks. Selu (Scaled Exponential Linear Unit) is a more recently developed variation of Relu (Klambauer et al., 2017). Padarian et al. (2019) proposes adding a extra convolutional layer

Table 3.2: Initial convolutional neural network architecture. The values in between brackets are the dropout rates.

| Layer | Filter size | Number of neurons | Activation function |
|---|---|---|---|
| Convolutional | 3 x 3 | 64 | ReLu |
| Max Pooling | 2 x 2 | – | – |
| Convolutional | 2 x 2 | 28 | ReLu |
| Dropout (0.5) | – | – | – |
| Flatten | – | – | – |
| Dense | – | 40 | ReLu |
| Dropout (0.2) | – | – | – |
| Dense | – | 50 | ReLu |
| Dropout (0.2) | – | – | – |
| Dense | – | 1 | Linear |

when dealing with larger window sizes. To see this effect models with two convolutional layers and three convolutional layers were used.

Table 3.3: Major hyperparameters

| Major parameters | settings |
|---|---|
| Window size | 15x15; 21x21; 27x27 |
| Data augmentation | Yes or No |
| Activation function | RELU or SELU |
| Number of convolutional layers | 2 or 3 |

A 10-fold cross-validation approach was used to select which of the 24 models obtained the highest average accuracy. For this cross-validation the 1892 points were split in ten folds that were used as test sets to calculate the accuracy for each model (Appendix A). Each model is trained ten times on the remaining training data sets and its accuracy is calculated on the ten test sets. In the end the final accuracy of each model was computed over all the folds. The model architecture with the highest accuracy was selected to be used for further modeling and comparison with a random forest model. During the training phase of the model, an automated stopping algorithm was applied to stop the model when it started overfitting. Each model therefore runs for a different number of epochs, which indirectly introduced an extra major parameter: the number of epochs.

A random forest model was used as a reference model as it has already often been used for digital soil mapping and is shown to perform well (Lamichhane et al., 2019). The random forest model was built with 500 trees and used the exact same input folds as the convolutional neural network. However the random forest model takes tabular data as input and does not need the high-dimensional arrays.

### 3.2.3 Accuracy

The accuracy of the models was computed from the prediction errors on the test sets.

$$e(x) = z_{s_i} - \hat{z}_{s_i} \tag{3.1}$$

where $z_{s_i}$ is the observed value at location s, and $\hat{z}_{s_i}$ is the predicted value at that location. To quantify the accuracy of the model predictions several measurements were used. The root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_{s_i} - \hat{z}_{s_i})^2} \tag{3.2}$$

The amount of variance explained by the model (AVE):

$$AVE = 1 - \frac{\sum_{i=1}^{n}(z_{s_i} - \hat{z}_{s_i})^2}{\sum_{i=1}^{n}(z_{s_i} - \bar{z})^2} \tag{3.3}$$

The mean error (ME) was used to asses the bias of the predictions:

$$ME = \frac{\sum_{i=1}^{n}(z_{s_i} - \hat{z}_{s_i})}{n} \tag{3.4}$$

The concordance correlation coefficient ($\rho$) was used to study the agreement of the predictions to the measurements with respect to the 1:1 line (Lin, 1989):

$$\rho = \frac{2\rho^{'}\sigma_z\sigma_{\hat{z}}}{\sigma_z^2 + \sigma_{\hat{z}}^2 + (\mu_z - \mu_{\hat{z}})^2}, \tag{3.5}$$

where n is the number of independent test samples, $\mu$, $\sigma$ and $\sigma^2$ are the mean, variance and standard deviation of either the predicted or observed values and $\rho^{'}$ is the correlation between the observed and predicted mean.

### 3.2.4 Final predictions maps

The convolutional neural network architecture that had the highest accuracy was trained again using all 1892 data points. This final model was then used for the predictions. Creating predictions with a convolutional neural network required considerable computational resources. The final predictions were therefore made for a subsection of Argentina that runs East to West across approximately the center of Argentina (Figure 3.1). This region covers a wide range of SOC values and contains large differences in climate and soils. To efficiently manage the prediction process, the area was split in 444 tiles of 100 by 100 cells, i.e. 25 * 25 km. After a prediction was made for each tile these were mosaiced together using GDAL to create the final prediction maps.

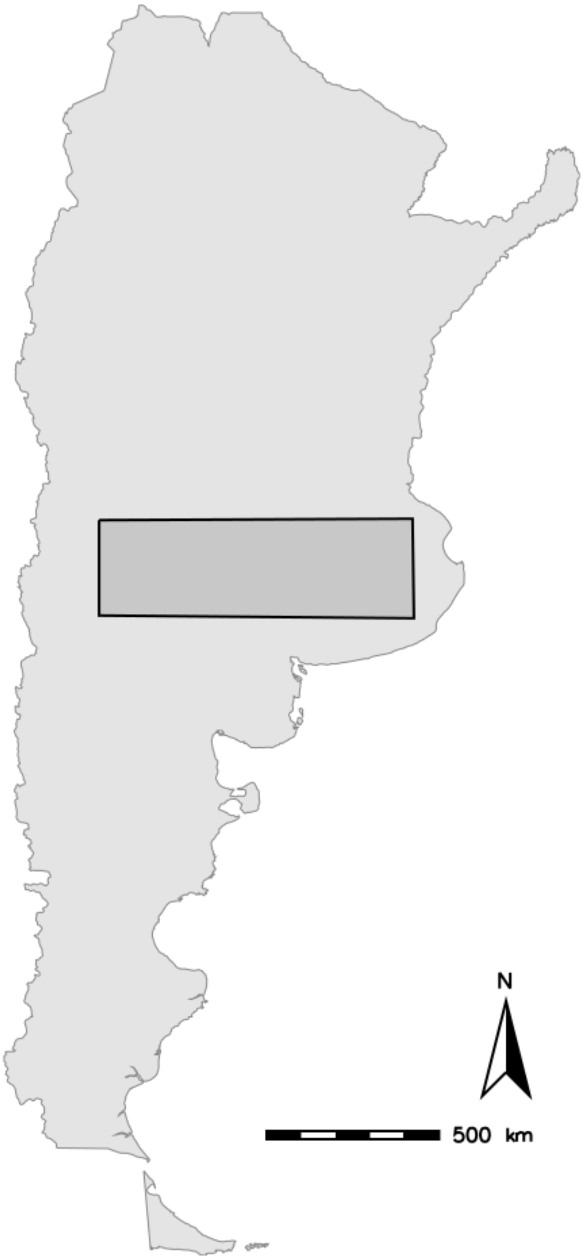


Figure 3.1: Region of Argentina that was used to create the final prediction maps

## 3.3 Opening the black box with LIME

To get insight into the decisions and functioning of the convolutional neural network the LIME algorithm was used (Ribeiro et al., 2016). LIME follows the principle that every complex model can locally be approximated by simpler models and fits sparse linear models for a selection of predicted points (Figure 3.2). Ribeiro et al. (2016) provides a detailed explanation of the algorithm.

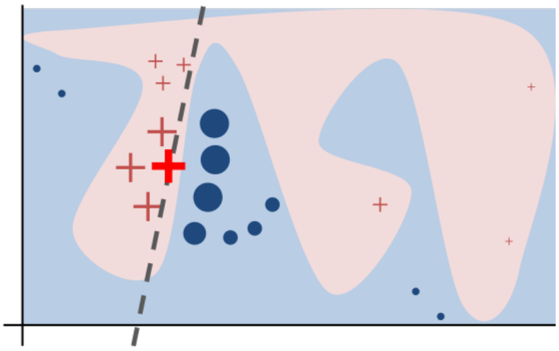


Figure 3.2: Graphical illustration of the functioning of LIME for a classification task. The blue/pink background represents a model's complex function $f$, which is unknown to LIME, and cannot be approximated well by a linear model. To predict at the bold red cross, LIME samples instances, gets predictions using $f$, and weighs these by the proximity to the instance being explained. These weights are represented by the sizes of the red cross and dark blue dots. The learned locally valid explanation is represented by the dashed line. Figure from Ribeiro et al. (2016)

This example and most uses of LIME are for classification tasks, for which the model outputs one outcome with the highest possibility. This study presents the first implementation of LIME in a digital soil mapping, where the outcome is a continuous numerical value instead of a classification. The algorithm has not yet often been used for regression problems and little reference was available for how to optimize the use of LIME for these tasks. To implement lime the R-package 'lime' was used (Pedersen and Benesty, 2018). The functions in this package take a machine learning model and a selection of points as input and run the lime algorithm on these. It is adapted to work with random forests as it can take a dataframe as input. However, the package had only recently been updated to work with convolutional neural networks and images as input. Trying to get it to work for the CNN would involve tricking the model into thinking it gets a real image as input. As the data in this research are not standard RGB images several adjustments were tried to get it to work with the covariates. Two groups of point were selected: six points that were close together in predicted SOC value, around 14 g/kg, and five points that covered more of the range of the SOC values (Figure 3.3). This way the differences in variable importance could be studied for similar SOC values and the extremes.

The lime package produced two types of plots. Detailed feature plots that show the weights attributed to the different covariates along with the values of the covariates. However, the values of the covariate had to be placed in a specified number of bins before the feature plots could be made, which causes the data values to be shown as thresholds. This in combination with data that is scaled to a mean of zero and standard deviation of 1, made for a difficult interpretation of these plots as the values no longer represent the actual value of the covariate. Next to these

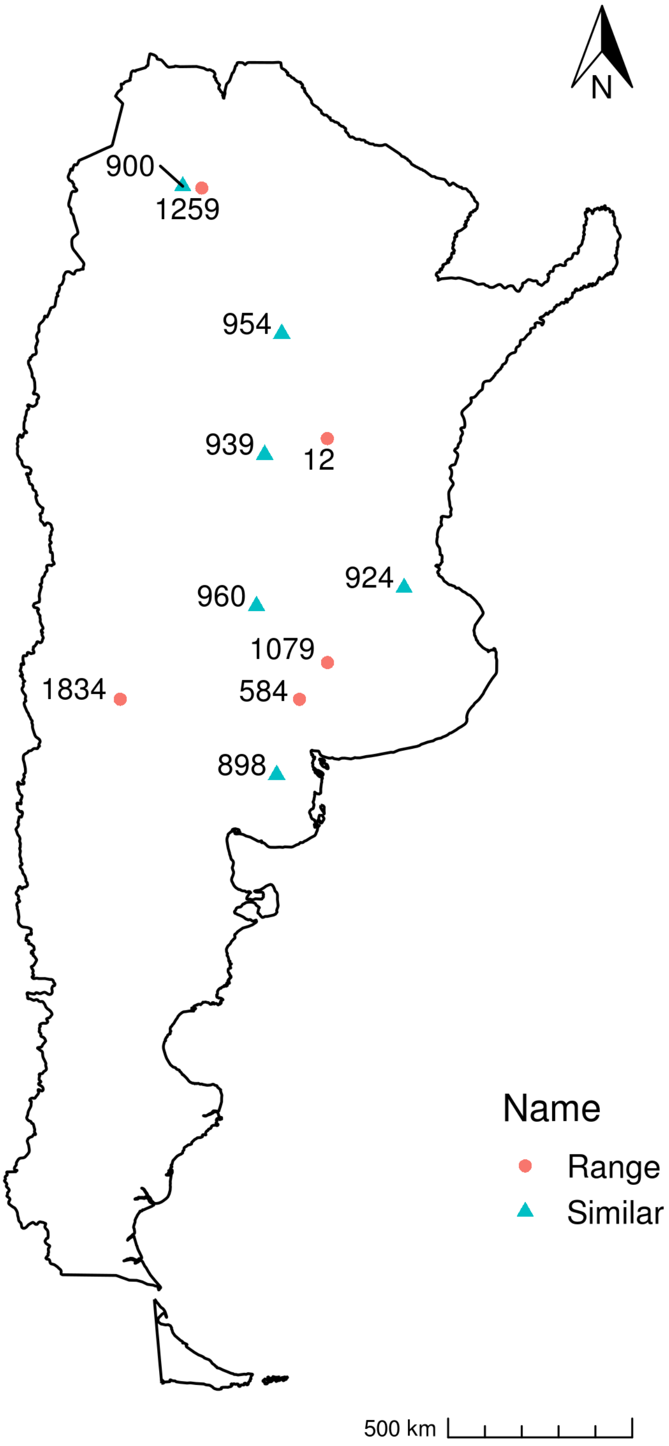


Figure 3.3: Points used with LIME. The blue triangles are points with predicted SOC values that are close together. The red dots are points taken from the full range of SOC values.

detailed plots more general overviews could also be plotted to review many explanations at once and look at the relative difference in importance of the covariates for the different points.

## 3.4 Software and computations

All statistical analyses and modeling were performed in the statistical programming language R running version 3.4.4 (R Core Team, 2019). R offers a wide range of packages to aid in statistical analysis, furthermore it is strong in working with spatial data and visualization. Besides the earlier mentioned R packages several other packages were also used. The main packages are 'raster' (Hijmans, 2019) to handle spatial data, 'ggplot2' (Wickham, 2016) for visualizations of both graphs and maps and 'dplyr' (Wickham et al., 2019) to efficiently handle data and dataframes. The random forest was implemented using the ranger package (Wright and Ziegler, 2017). After testing on a simple desktop the full computations were performed on a 12-core linux server. Working with raster layers in R can be quite memory intensive. To avoid running into memory issues most of the raster processing was done outside of R using GRASS GIS (GRASS Development Team, 2019) and GDAL (Greenberg and Mattiuzzi, 2018). Raster layers were only loaded into R when absolutely necessary.

# Chapter 4

# Results

## 4.1 Covariate selection

A selection of highly correlated covariates were removed after checking the pearson correlation coefficient. This reduced the covariates set from 352 layers to 132 layers. This set of covariates was further reduced using a recursive feature elimination (X. Chen and Jeong, 2007). A quick initial increase in accuracy is visible up to around 20 variables. After this initial increase, the increase gets minimal and the accuracy stabilizes (Figure 4.1). The RMSE values represent the normalized data and do therefore not show the true RMSE, but the same increase in accuracy is visible as for the AVE.

The RFE selected an optimal value of 55 variables when using random forest. However, because

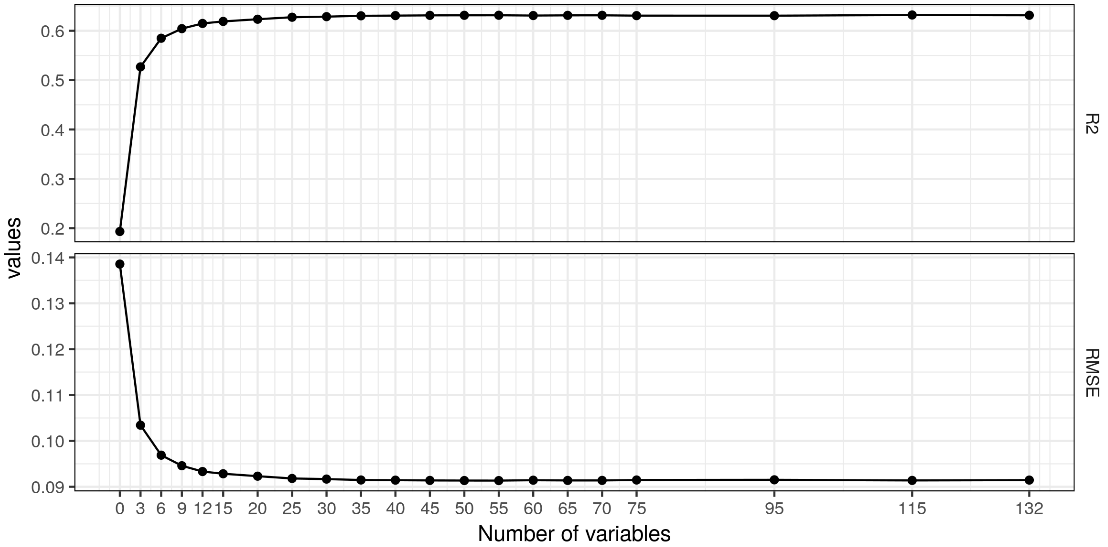


Figure 4.1: R
ecursive feature elimination results]Results of the recursive feature elimination run using 132 covariates. The best performance is obtained when using 55 covariates. However, after 20 variables the increase in accuracy is already minimal.

convolutional neural networks tend to be more computation-intensive than random forests only the 20 most important variables were selected for further modelling. Increasing the number of covariates any further would likely only give a small increase in accuracy. This increase would possibly not weigh up against the increase in computation time. The 20 selected variables are given in Table 4.1 along with a check for the direct Pearson correlation between these covariates and the topsoil organic carbon. The correlation with SOC is relatively low for all covariates. The selected covariates include EVI and NDVI layers that represent vegetation intensity. Climate is represented by surface temperatures, precipitation, precipitable water vapor and cloud cover layers. The highest found direct correlation is 0.49 for the long-term averaged mean monthly enhanced vegetation index (EVI) for September and October. The long-term averaged mean monthly surface temperature (nighttime) in February has the lowest direct (absolute) correlation with a negative correlation of -0.11.

Table 4.1: The twenty covariates that were selected by the RFE including their direct correlation with SOC

| Covariate | correlation |
|---|---|
| Mean yearly MODIS Enhanced Vegetation Index (EVI) | 0.49 |
| Normalized Difference Vegetation Index November, October, December | 0.49 |
| Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months September and October | 0.49 |
| Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months March and April | 0.44 |
| Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months November and December | 0.44 |
| Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months January and February | 0.44 |
| Long-term s.d. of the monthly MODIS Enhanced Vegetation Index (EVI) for months March and April | 0.41 |
| Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months July and August | 0.39 |
| Long-term s.d. of the monthly MODIS Enhanced Vegetation Index (EVI) for months July and August | 0.39 |
| Normalized Difference Vegetation Index January, February, March | 0.37 |
| Standard Deviation yearly MODIS Enhanced Vegetation Index (EVI) | 0.37 |
| Precipitation of wettest month | 0.34 |
| Long-term averaged mean cloud cover | 0.28 |
| Long-term averaged mean monthly MODIS Precipitable Water Vapor in cm for months July and August | 0.25 |
| Long-term averaged monthly cloud cover July | 0.16 |
| Long-term averaged mean monthly surface temperature (nighttime) MODIS February | -0.11 |
| Long-term averaged mean monthly surface temperature (daytime) MODIS March | -0.29 |
| Long-term averaged mean monthly surface temperature (daytime) MODIS November | -0.31 |
| Long-term s.d. of the monthly surface temperature (nighttime) MODIS Yearly | -0.31 |
| Long-term averaged mean monthly surface temperature (daytime) MODIS February | -0.53 |

## 4.2 Optimal model selection

A 10-fold cross validation was run to select the best performing convolutional neural network architecture out of combination of 4 major parameters (Table 3.3). For each architecture the accuracy was calculated. The architecture combination with the overall highest accuracy is a model with two convolutional layers, activated with SELU activation functions, no data augmentation and a window size of fifteen by fifteen cells, i.e. 3750 by 3750 meters (Table 4.2). This architecture achieved an AVE of 0.48, a RMSE of 7.27 (g/kg), a concordance correlation coefficient of 0.67 and a mean error of 0.38 (g/kg) (Table 4.3). The difference between the 24 architecture combinations are relatively small, especially within the top 10 (Table 4.3). The AVE is ranging from 0.472 to 0.439, the RMSE from 7.253 to 7.499 (g/kg) and the CCC from 0.667 to 0.643. The ME is a bit more variable and ranges from 0.377 to 1.709 (g/kg).

Table 4.2: Model architecture of the best performing model.

| Layer | Filter size | Number of neurons | Activation function |
|---|---|---|---|
| Convolutional | 3 x 3 | 64 | SELU |
| Max Pooling | 2 x 2 | | |
| Convolutional | 2 x 2 | 28 | SELU |
| Dropout | 0.5 | | |
| Flatten | | | |
| Dense | | 40 | SELU |
| Dropout | 0.2 | | |
| Dense | | 50 | SELU |
| Dropout | 0.2 | | |
| Dense | | 1 | Linear |

Table 4.3: Top 10 of the model and input architectures with the highest accuracy

| Activation function | N-convolutional layers | Window size [pixels] | Augmentation | AVE | RMSE [g/kg] | CCC | ME [g/kg] |
|---|---|---|---|---|---|---|---|
| Selu | 2 | 15 x 15 | No | 0.481 | 7.266 | 0.672 | 0.377 |
| Selu | 3 | 21 x 21 | Yes | 0.470 | 7.339 | 0.671 | 0.446 |
| Selu | 3 | 27 x 27 | No | 0.470 | 7.343 | 0.652 | 0.548 |
| Selu | 2 | 15 x 15 | Yes | 0.468 | 7.357 | 0.649 | 1.163 |
| Relu | 2 | 27 x 27 | Yes | 0.458 | 7.425 | 0.619 | 1.485 |
| Selu | 2 | 21 x 21 | No | 0.455 | 7.442 | 0.655 | 0.650 |
| Selu | 3 | 21 x 21 | No | 0.453 | 7.455 | 0.643 | 0.989 |
| Relu | 3 | 27 x 27 | Yes | 0.447 | 7.501 | 0.614 | 1.709 |
| Selu | 2 | 27 x 27 | No | 0.443 | 7.527 | 0.648 | 0.685 |

Figure 4.2 shows the relations between the major parameters and the RMSE. Models that use Selu activated neurons dominate the lower parts of the graphs and seem to get better accuracies on general. A window size of 15 shows the largest spread in RMSE compared to a window size of 21 and 27, but also contains the lowest RMSE value. When the window size increases the Relu activated models do show an increase in accuracy, but still do not obtain as low of a RMSE as the Selu activated models. The spread in RMSE values is more or less the same for either two or three

convolutional layers, but the values are generally lower for models with two convolutional layers. Also here Selu outperforms Relu for most combinations. A large gap between Selu and Relu is visible in the RMSE values when using no data augmentation. When the data are augmented both Relu and Selu activated models get relatively low RMSE values, but not as low as with no augmentation.

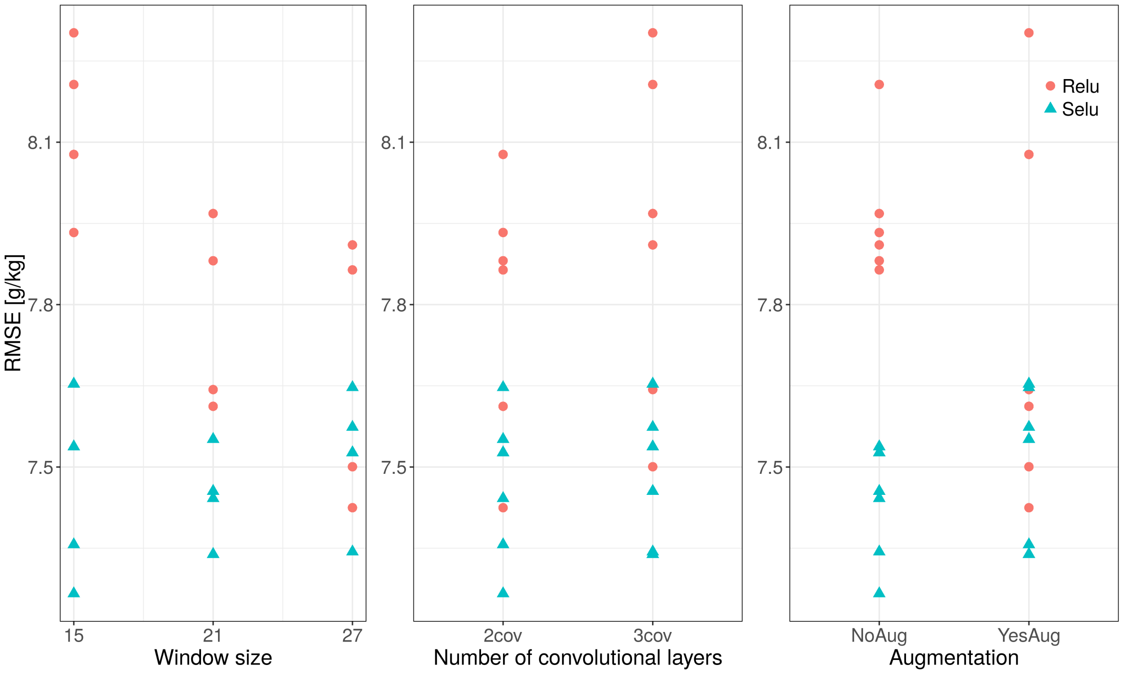


Figure 4.2: Response of the model accuracy to some major parameters.

## 4.3 Accuracy of CNN versus random forest

The random forest obtained an AVE of 0.53, RMSE of 6.91 (g/kg), a concordance correlation coefficient of 0.69 and a mean error of 1.09 (g/kg). When compared to the convolutional neural network, the random forest achieved better results for each accuracy measurement, except for the mean error. Table 4.4 gives an overview of the achieved accuracy for both models.

Table 4.4: 10-fold cross validation results

| Model | AVE | RMSE [g/kg] | CCC [-] | ME [g/kg] |
|---|---|---|---|---|
| Convolutional neural network | 0.48 | 7.27 | 0.67 | 0.38 |
| Random Forest | 0.53 | 6.91 | 0.69 | 1.09 |

Figure 4.3 gives plots of the observed versus predicted soil organic carbon concentrations for both the convolutional neural network and the random forest. The trend of both models generally follows the 1:1 line, but there is quite a lot of scatter visible. Both models seem to underpredict the high observed values.

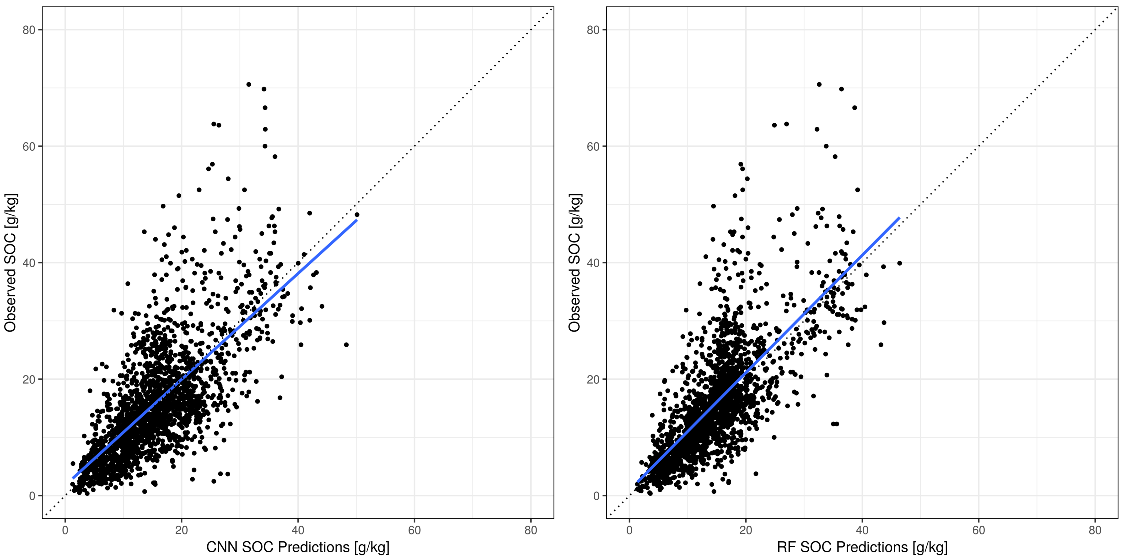


Figure 4.3: Observed vs. predicted soil organic carbon concentrations (g/kg) for the convolutional neural network (left) and the random forest (right)

Both models show a large spread in accuracy over the ten folds (Figures 4.4 & 4.5). For the optimal CNN model the AVE values are ranging from 0.34 to 0.58 and the RMSE is ranging from 7.56 to 6.36 (g/kg). For the random forest the AVE values are ranging from 0.41 to 0.68 and a RMSE of 5.58 (g/kg). The trends of both the CNN and the random forest model generally follow the 1:1 line, but still quite some scatter around the lines is visible.

During the modeling the impression rose that the neural network might be heavily influenced by setting the seed in the R-script, which also could cause the big difference between the folds. But it turned out that this was only caused by the selection of the input data for the model. To test this the optimal CNN model was run with the data from fold number 7, with four different seeds. The AVE ranges from 0.59 to 0.61, the RMSE from 6.11 to 6.32 (g/kg) and the CCC from 0.78 to 0.79. So there is a slight influence of the seed, but the influence is minimal.

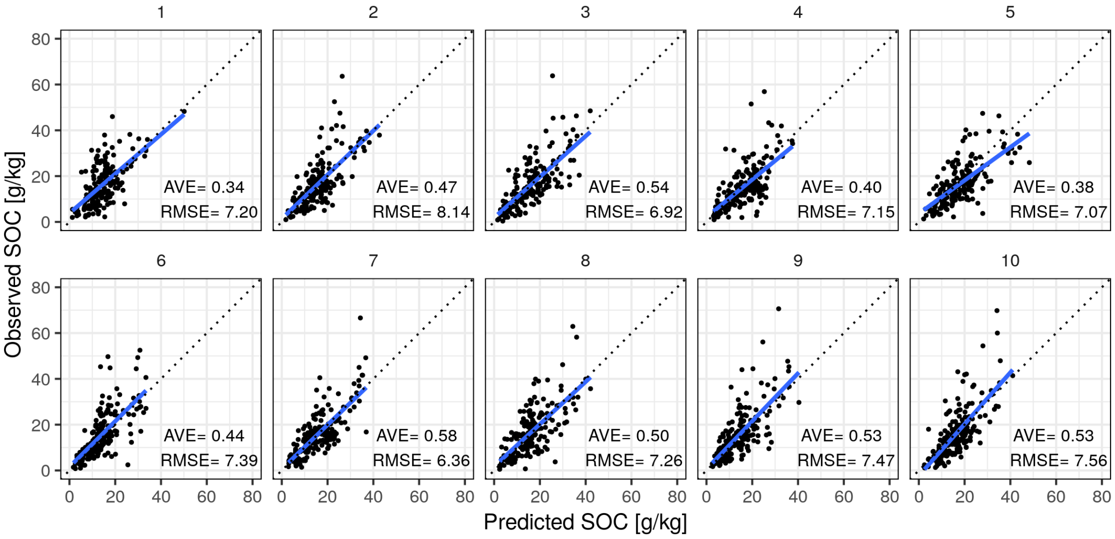


Figure 4.4: Results of the 10-fold cross validation per fold for the convolutional neural network. The dotted line represents the 1:1 relation. A trend line of the points is shown in blue.

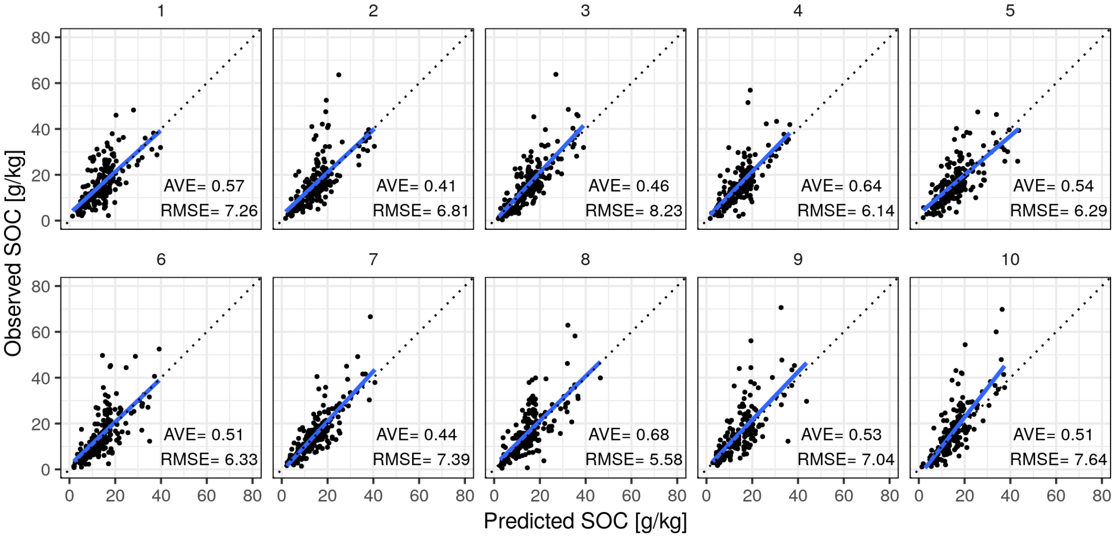


Figure 4.5: Results of the 10-fold cross validation per fold for the random forest. The dotted line represents the 1:1 relation. A trend line of the points is shown in blue.

### 4.3.1 K-fold predictions maps

Figure 4.6 shows the predictions on the test sets for the convolutional neural network and the random forest. In general the predictions show the same spatial pattern as is visible in the observed data (Figure 2.2). Most points are fairly close to the observed values, but for some points the difference between predicted and observed goes up to almost 50 g/kg (Figure 4.7 & 4.8). These figures again show the underfitting of both models where they severly underpredict some of the points, but the figures also show the models strongly overpredict some points.

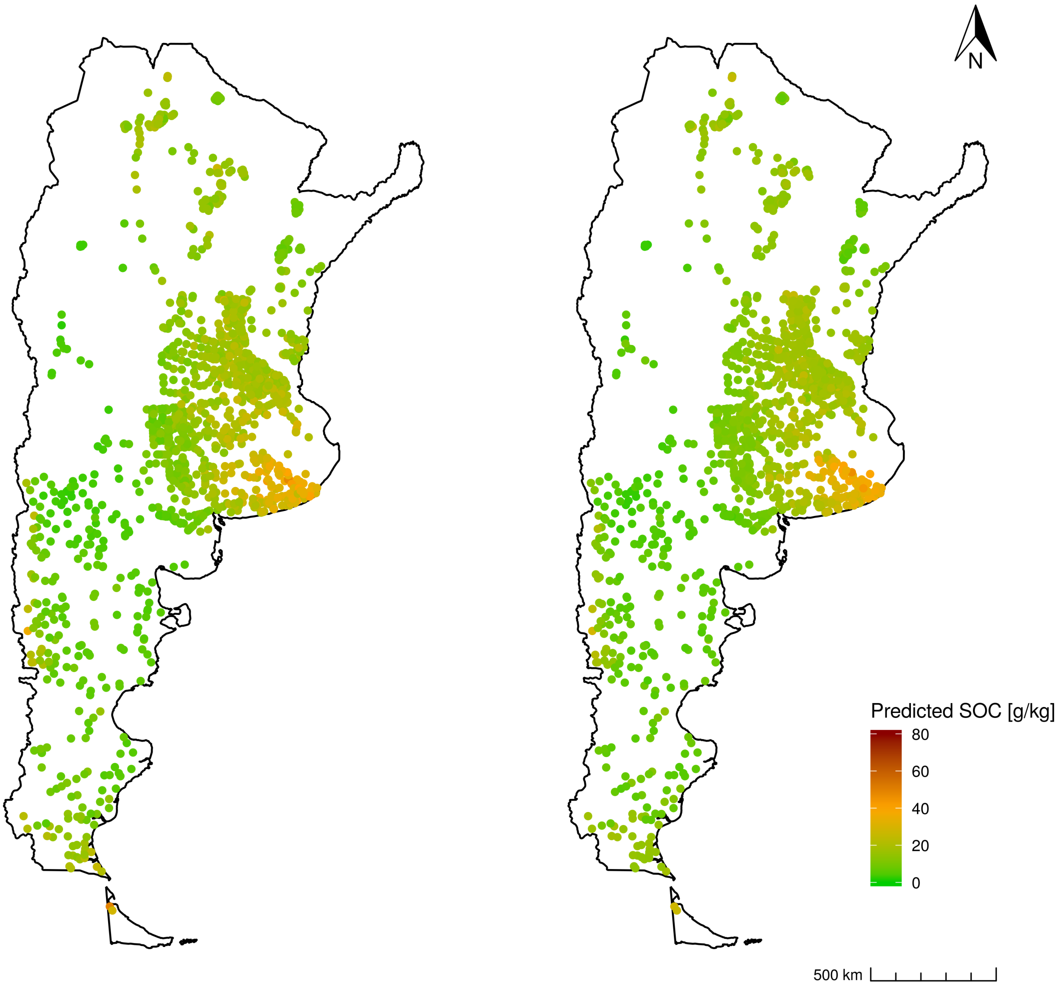


Figure 4.6: Points predictions on the test data sets from the convolutional neural network (left) and the random forest (right)

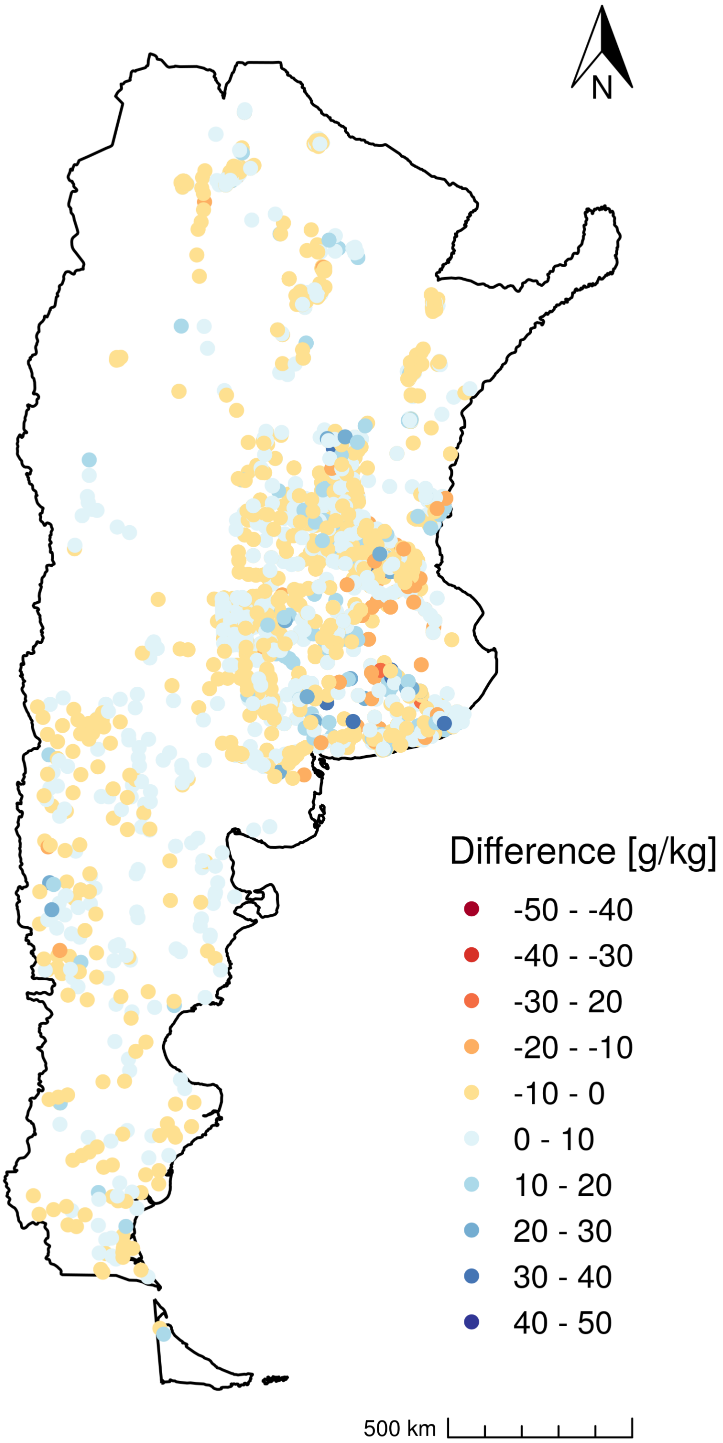


Figure 4.7: Difference between the observed SOC concentrations at the point locations and the predicted concentration using the convolutional neural network.

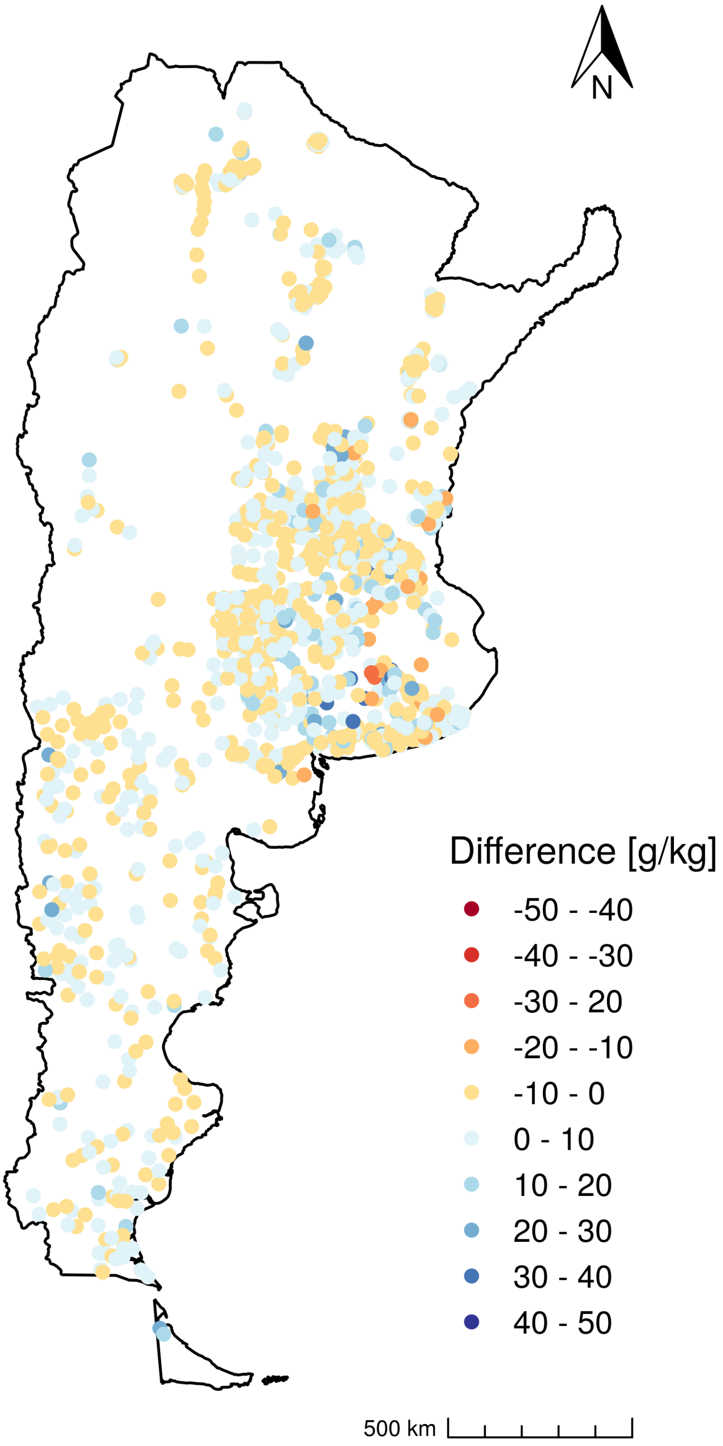


Figure 4.8: Difference between the observed SOC concentrations at the point locations and the predicted concentration using the random forest.

### 4.3.2 Pedological data

The accuracy of the convolutional neural network and the random forest was also assessed with covariates selected using pedological knowledge, including several terrain covariate layers. After the 10-fold cross validation with this data the same convolutional neural network architecture was selected as the optimal model. However, the rest of the top ten looks different (Table 4.5). The most notable results are that more models with Relu activation functions were performing well and only two models without data augmentation made the top 10.

Table 4.5: Top 10 of the model and input architectures with the highest accuracy using the pedologically selected data

| Activation function | N-convolutional layers | Window size [pixels] | Augmentation | AVE | RMSE [g/kg] | CCC | ME [g/kg] |
|---|---|---|---|---|---|---|---|
| Selu | 2 | 15 x 15 | No | 0.448 | 7.493 | 0.654 | 0.422 |
| Selu | 2 | 21 x 21 | Yes | 0.447 | 7.501 | 0.647 | 0.756 |
| Relu | 2 | 21 x 21 | Yes | 0.446 | 7.508 | 0.610 | 1.715 |
| Relu | 2 | 27 x 27 | Yes | 0.444 | 7.521 | 0.607 | 1.343 |
| Selu | 3 | 27 x 27 | Yes | 0.444 | 7.522 | 0.639 | 0.562 |
| Relu | 2 | 15 x 15 | Yes | 0.444 | 7.523 | 0.609 | 1.455 |
| Relu | 3 | 21 x 21 | No | 0.443 | 7.526 | 0.605 | 1.791 |
| Selu | 2 | 27 x 27 | Yes | 0.437 | 7.569 | 0.621 | 0.413 |
| Selu | 3 | 15 x 15 | Yes | 0.432 | 7.601 | 0.632 | 0.717 |
| Relu | 3 | 15 x 15 | Yes | 0.425 | 7.644 | 0.591 | 2.062 |

The obtained accuracy of the CNN models and the random forest model is consistently lower than for the models using the RFE data, but the differences are very small. The best performing convolutional neural network obtained an AVE of 0.45, a RMSE of 7.49 (g/kg), a concordance correlation coefficient of 0.65 and the mean error of 0.42. The random forest model obtained an average AVE of 0.52, a RMSE of 6.95 (g/kg), a CCC of 0.68 and a mean error of 1.21 (g/kg).

### 4.3.3 Prediction maps

Using the best performing convolutional neural network and the random forest, predictions were made for a subsection of Argentina (Figure 4.9). These prediction maps show some differences in the spatial patterns. The range of predicted values is much higher for CNN than for RF. The CNN predictions range from 0 g/kg to 50 g/kg. The RF predictions range from 5 g/kg to 35 g/kg. Therefore less details are visible in the random forest plots. Both models show the same trend of low values in the West of the subsection and higher values in the East. But the locations of the highest values is different for the two plots. The CNN predicts a small area with extremely high values in the South-Eastern corner of the section, while this region is not visible on the RF predictions.

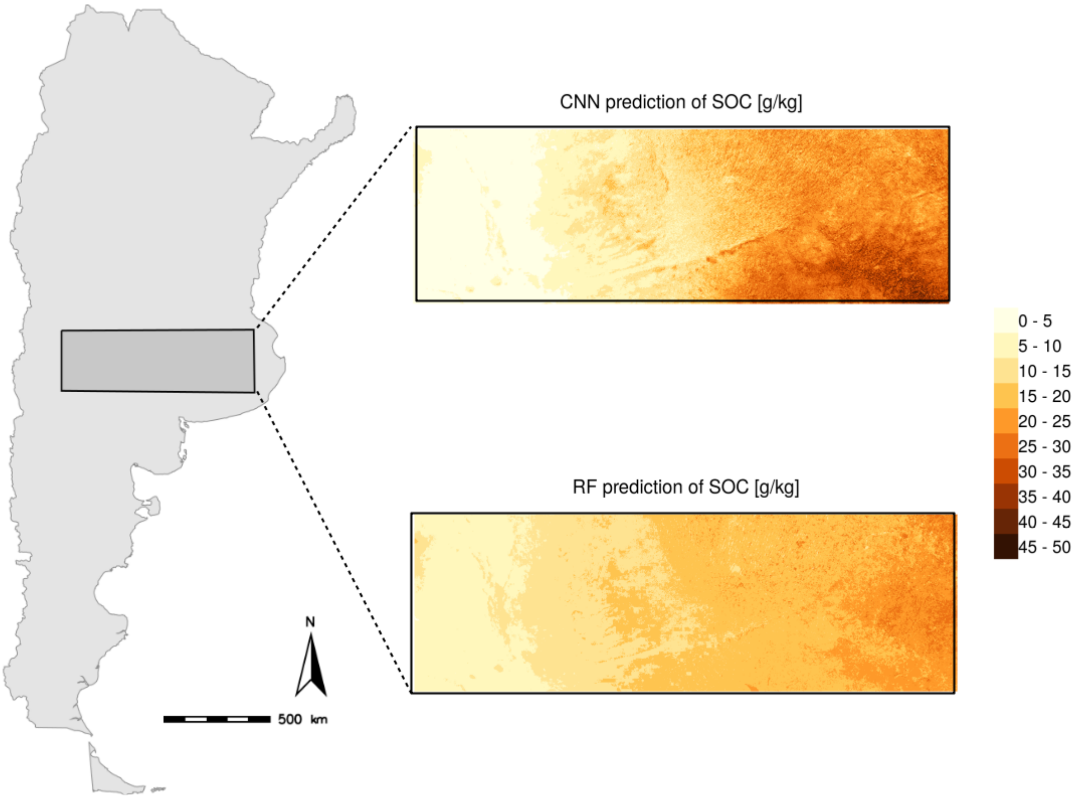


Figure 4.9: Prediction maps made using the convolutional neural network (top) and the random forest (bottom). For visibility reasons two different legends are used for the two maps.

## 4.4 Opening the black box with LIME

Unfortunately it was not possible to get the lime package working for the convolutional neural network. The algorithm would not accept the input in the form as it was used for the CNN. However, the algorithm did work for the random forest. The LIME algorithm was run for a selection of six points that were close together in predicted SOC concentration and a selection of five points that capture a large part of the prediction range (Figure 3.3). The lime package gave two feature plots as output (Appendix B). However, these proved difficult to interpret and seem to be more useful for classification tasks and for studying each individual point. Using continuous and scaled data as input makes these figures hard to read when two bins are defined (which is compulsory). These plots thus show results in relative terms, e.g. a relatively high ndvi versus a relatively low ndvi.

More useful were the explanation plots (Figures 4.10 & 4.11). A summary of the covariate codes and their meaning is given in appendix C. These plots give a general comparison of covariate importance between the points. In figure 4.10 all cases, i.e. points, have a predicted SOC value of around 14 g/kg. Only small differences in covariate weights are visible when looking at this figure. For figure 4.11 the points with a large difference in predicted SOC concentration were used. More difference in the weights are visible in between the points. For instance for the average daytime temperature in November (CLM_MOD_LSTD11AVG) the negative weight is severely higher for case 1079 (14.01 SOC (g/kg)) compared to for instance case 1834 (2.86 SOC (g/kg)). In general the figures show that the average ndvi for the months October, November and December gets the highest weight for all point observations and the daytime average temperature in November gets the strongest negative weight. It is interesting to see that the yearly average enhanced vegetation index gets almost 0 weight for all covariates, while this is the covariate with the highest score from the recursive feature elimination.

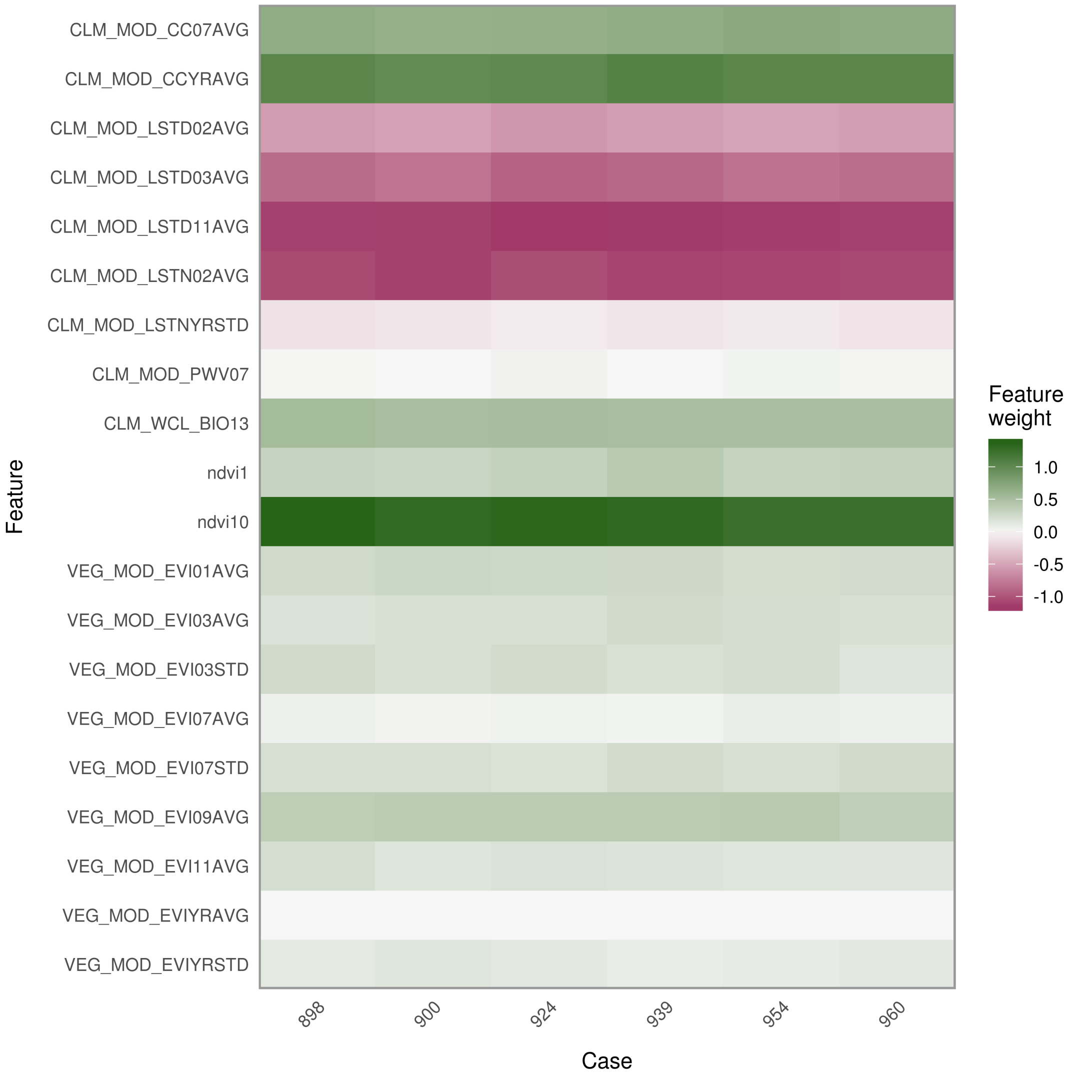


Figure 4.10: General explanation plot of the LIME prediction, that shows the weights assigned to each covariate for six points that have more or less the same SOC value.

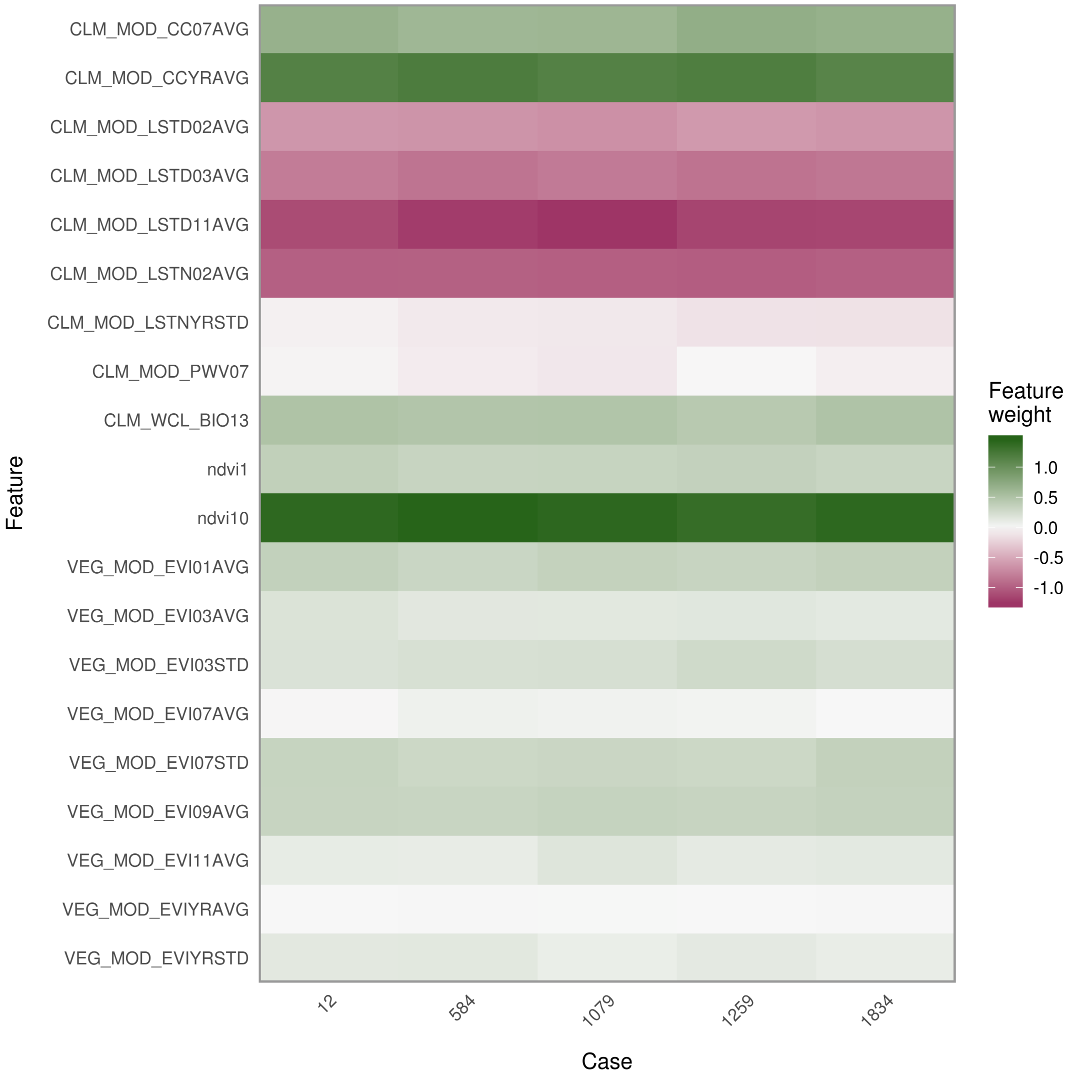


Figure 4.11: General explanation plot of the LIME prediction, that shows the weights assigned to each covariate for five points that cover a large range of SOC values.

# Chapter 5

# Discussion

## 5.1 Covariate selection

### 5.1.1 Optimal number of covariates

A recursive feature elimination (RFE) method was used to select the most important covariates out of the total list of covariates. RFE has shown promising results since its development. However, removing all 'weak' features might prove negative for the model performance as these features might be powerful when used together (Guyon and Elisseeff, 2003). X. Chen and Jeong (2007) shows this improvement of accuracy when 'weak' features are used together for a classification task using support vector machines. It might thus be that, in this research, features that could together have had a major positive effect on the modeling results were not selected. The recursive feature elimination gave an optimum of 55 covariates to obtain the highest possible accuracy when modeling with a random forest (Figure 4.1). To reduce computation times when training and predicting with the convolutional neural network the decision was made to use twenty variables. These twenty variables were still able to achieve a high accuracy as the increase in accuracy, when using more than twenty variables is minimal.

Selecting fewer variables than optimal for the random forest might have favored the convolutional neural network. Random forest generally respond well to large amounts of covariates, while the performance is quickly limited when the number of covariates is minimized (Nussbaum et al., 2018). CNN's might be able to deal with a small number of covariates very well, as they create a large number of hyper-covariates from the original input covariates (Wadoux, 2019). This possible effect is, however, not shown in this research as the accuracy of the CNN is lower than that of the random forest. In other digital soil mapping studies that only use a small number of covariates in combination with a convolutional neural network, the CNN outperforms the random forest (Wadoux et al., 2019; Padarian et al., 2019). A further decrease in the number of covariates, e.g. from twenty to three, might thus push the results in the favor of CNN compared to random forest. If this decrease in number of covariates also increases the overall accuracy of the convolutional neural networks has to be further investigated.

### 5.1.2 Selected covariates

**Vegetation indices**

The RFE selected a large number of vegetation indices as important features when predicting soil organic carbon (Table 4.1). The vegetation indices are all positively correlated with soil organic carbon, so more vegetation will in general lead to more organic material that can be added to the soil. This relation seems to be strong enough for the RFE to have selected several of these layers.

Next to vegetation indices, land use or land cover are also often used as important covariates for the mapping of soil organic carbon (e.g. Wadoux, 2019; Heuvelink et al., 2019). However, no land use or land cover layers were selected by the covariate selection. Most likely the influence of vegetation on soil organic carbon is already strongly covered by the NDVI and EVI layers. Multiple studies show this correlation between the vegetation indices and land cover (e.g. J. Chen et al., 2015; Wang et al., 2005).

**Climate data**

Several surface temperature layers were selected by the RFE. They all have a negative correlation with SOC which indicates a higher temperature will lead to lower soil organic carbon. The lower soil organic carbon concentrations can indeed be found in the large areas of Argentina that are arid or semi-arid and thus produce little organic matter (Chapter 2). These are also the regions with the highest mean monthly surface temperature in February.

Two covariates that are related to the water availability for the plants were selected. The precipitation of the wettest month and the the precipitable water vapor both show a positive correlation with SOC. In general more water availability will lead to a higher productivity in plants, which leads to a possible higher influx of plant material to the soil. The RFE also selected two cloud cover layers, these are the long-term averaged cloud cover and the average cloud cover in July. A direct relationship is not to be expected between cloud cover and soil organic carbon. It is likely that cloud cover is a proxy for the relation between precipitation and soil organic carbon concentrations.

**Terrain parameters**

The recursive feature elimination selected multiple layers of several covariates, but did not select any terrain and relief covariates within the top 20. This is suprising, as these factors are important aspects of the SCORPAN digital soil mapping framework (McBratney et al., 2003) and are often used in soil organic carbon modeling studies (e.g. Wadoux et al., 2019; Padarian et al., 2019; Mishra et al., 2010). Most likely their relationship with SOC was masked by the large number of climate and vegetation data available for the RFE.

## 5.2 Optimal major parameters

The 10-fold cross validation with 24 model architecture combination revealed several relations between the accuracy of the models and the parameters (Figure 4.2).

### 5.2.1 Relu versus Selu activation

Models with Selu activated neurons clearly outperform the models with Relu activations functions as is seen by the almost constantly lower RMSE. The reason that Selu activated neurons outperform the Relu activated neurons is because neurons that use Relu activation functions can get stuck at a zero gradient. When the neurons get stuck at a zero gradient they do not contribute to the learning of the model anymore and become ”dead” neurons. Selu activated neurons were designed to overcome this problem and can not go to a zero gradient (Klambauer et al., 2017). This effect of the dead neurons is probably also visible in the number of epochs the models where trained for (Figure 5.1). Because neurons of the Relu models can stop learning the models train less smoothly and have a risk of overfitting quite quickly, which causes them to stop training early when automated early stopping is active. It is visible that in general the Relu activated models stop after fewer epochs than the Selu models. By stopping early the model most likely has not yet reached its optimal training, which explains their lower accuracy. From this results it seems that training the convolutional neural network for more epochs generally leads to a lower RMSE and thus a better accuracy. Although Selu models on average train longer than Relu models, all models still stopped under 200 epochs. This is low compared to other convolutional neural networks used for SOC mapping (Wadoux et al., 2019; Padarian et al., 2019). However, Latifovic et al. (2018) obtained results deemed reasonable on a classification task to create a geological map of a region of Canada, while using only 100 epochs.

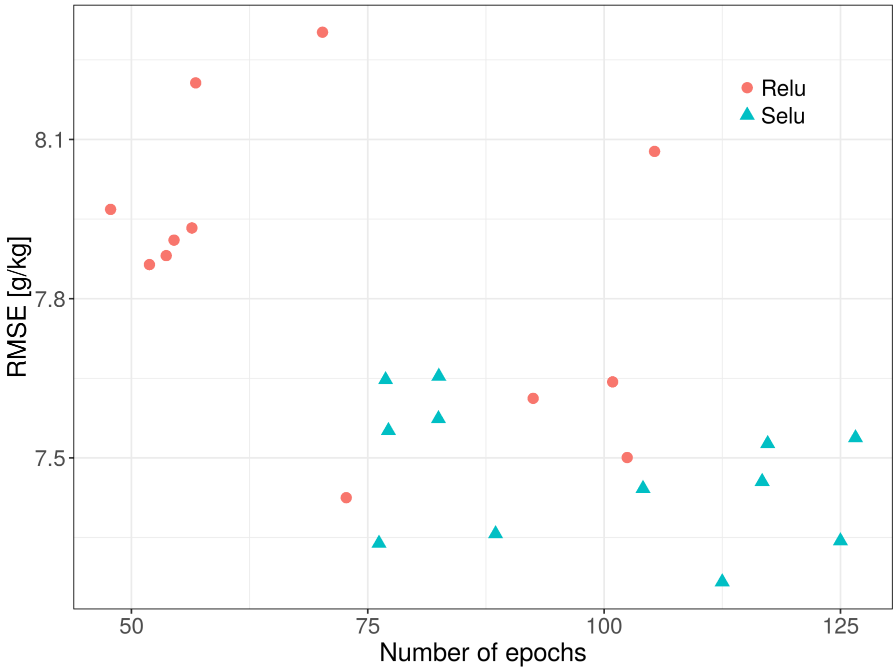


Figure 5.1: Relation between the number of epochs and the RMSE

### 5.2.2 Window size

The window size has a strong effect on model accuracy (Table 3.3). If the size of a window exceeds the size of the spatial correlation in environmental factors, the patterns that are observed might no longer effect the soil organic carbon concentration. This negative effect on the accuracy by increasing the windowsize is shown by (Padarian et al., 2019). Wadoux et al. (2019) shows an initial increase in accuracy with an increase in windowsize, but also shows a decrease in accuracy when the windowsize gets too large. Both of these studies use covariates with a smaller resolution than this research: 100m and 25m respectively. Wadoux et al. (2019), Padarian et al. (2019) and Wadoux (2019) all found very different scales of spatial correlation for SOC, ranging from several hundred meters to several kilometers. Wadoux (2019) used data with a resolution of 1 km and found that there is a spatial correlation up to 3 km for organic carbon. That finding coincides with the optimal windowsize found in this research of 15 x 15 cells, i.e. 3750 m by 3750 m. These differences between the different researches most likely show that resolution is an important aspect when determining the optimal window sizes. This makes sense as data of different resolutions represents different spatial patterns, e.g. global climate data shows a different scale of temperature patterns than a regional model. Adding a analysis of the sample variogram of soil organic carbon could be a valuable addition to future research, to more effectively determine the windowsize.

### 5.2.3 Convolutional layers

Increasing the window size increases the number of parameters in the model and therefore increases the risk of overfitting. Padarian et al. (2019) adds extra convolutional layers, and subsequently pooling layers, to minimize this risk. But increasing the complexity of a deep learning model by itself also increases the chance of overfitting on the training data, especially when the amount of data is limited (Srivastava et al., 2014). In this research the RMSE is consistently higher for models with three layers than for models with two convolutional layers. The increased complexity thus indeed causes a less stable model. Changing the dropout rate based on the number of layers in the network instead of setting it to a fixed number, might have a positive effect on the performance of the deeper models (Srivastava et al., 2014).

### 5.2.4 Data augmentation

Models with no augmentation, i.e. models that have less input data, seem to perform better in general than the models with data augmentation (Table 3.3). One would expect the opposite because in general neural networks depend on large amounts of training data to obtain accurate results (Simard et al., 2003). Convolutional neural networks are therefore often trained with tens of thousands of training images, when they are used for image classification (Krizhevsky et al., 2012). However, in the field of digital soil mapping those amounts of data are of often not available. Wadoux et al. (2019) used 2962 samples and Padarian et al. (2019) augmented the data used to obtain 1744 samples. Padarian et al. (2019) shows a decrease in modeling error of about 10% when modeling with 1744 samples instead of 436. In this research data augmentation, from 1892 original samples to 7568 samples after augmentation, did not lead to an overall increase in accuracy (Table 3.3). Also in studies from other domains it is found that not all data augmentation techniques prove effective (e.g. Schlüter and Grill, 2015). It could be that the data augmentation lets the CNN learn patterns that simply are not to be found elsewhere in Argentina and therefore inhibit the generalization of the neural network. In further research several different data augmentation techniques could be used to come up with the optimal augmentation method for digital soil mapping.

## 5.3 Accuracy

### 5.3.1 Evaluation of model accuracy

On average the random forest obtains a better accuracy than the convolutional neural network (Table 4.4). This better performance of the random forest is most likely caused by the selection of the covariates, as the recursive feature elimination optimized this selection for random forest. The random forest obtains significantly better results for the AVE and RMSE. The CCC is more or less the same for both models, but again the random forest has a slightly better score. Only for the mean error the random forest performs worse, with a mean error of 1.09 (g/kg), compared to a mean error of 0.38 (g/kg) for the convolutional neural network. The mean errors show that both models are on average underpredicting, but that the underprediction is larger for the random forest. This bias is most likely due to the underestimation of the extremely high SOC values (Figure 5.2).

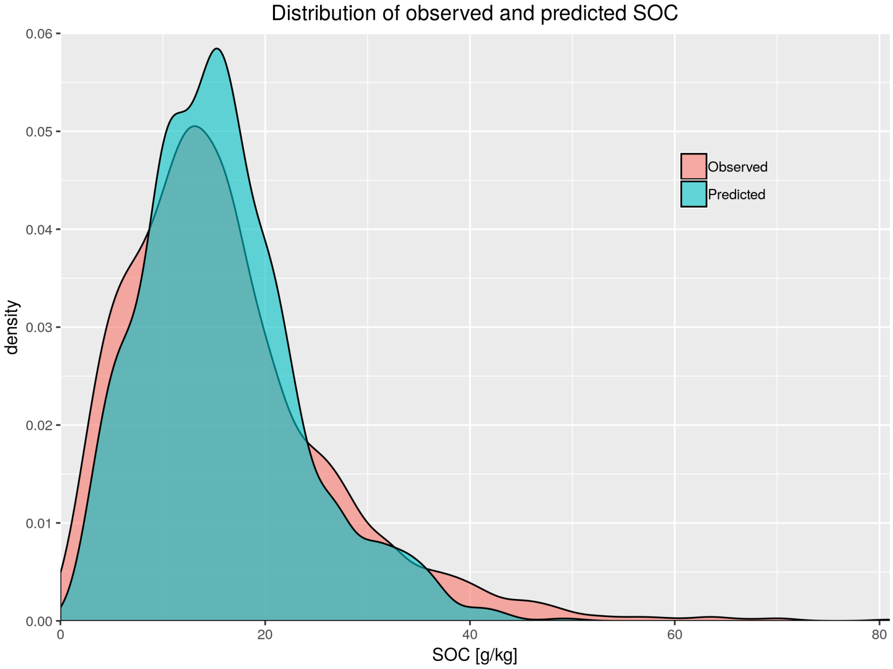


Figure 5.2: Density plot of the observed SOC [g/kg] and predicted SOC distributions.

A standard k-fold cross validation approach was used in this research to calculate the accuracies and select the model with the highest accuracy. However, this standard approach might produce overoptimistic estimates of the true prediction error (Krstajic et al., 2014). Krstajic et al. (2014)

therefore proposed the nested cross validation approach to avoid these overoptimistic estimates. This method could have been used in this research and would have produced more independent and accurate accuracy measures. The nested cross validation approach would involve splitting the data one extra time, which reduces the amount of data available for training. In this specific case this could have lead to a overall lower accuracy of the model, but this was not further investigated.

### 5.3.2 Point predictions

The underpredictions and overpredictions are also visible, when looking at the difference plots (Figures 4.7 & 4.8). Overall the error is relatively small, as most points are predicted within a -10 to 10 (g/kg) range around the observed value. But especially in the region South of Buenos Aires, the errors are larger. This is also the region in which there is more variation in SOC concentrations (Figure 2.2). The general spatial pattern in the point predictions (Figure 4.6) is comparable with the observed SOC values (Figure 2.2). With a smooth increase in SOC values from the low SOC concentrations in the arid regions in the center of the country towards the higher concentrations in the region south of Buenos Aires. The predictions, however, show a smoother pattern than the observed values. This is a general smoothing effect that regression models have when predicting, where high values tend to be underpredicted and low values tend to be overpredicted.

### 5.3.3 Comparison to other studies

The accuracy of the random forest is comparable to the accuracy obtained by a modeling study for topsoil SOC predictions in Argentina (Poggio et al., 2018). It is interesting to see that Poggio et al. (2018) obtained these results with very different covariates. Covariates used in Poggio et al. (2018) are the seasonal averaged NDVI's and several terrain parameters, i.e. elevation, slope and topographic position index. Wadoux et al. (2019) obtains an AVE of 0.55 using a convolutional neural network for the prediction of soil organic carbon, while a random forest achieved an AVE of 0.35. The CNN model thus performs slightly better than the CNN in this research and clearly outperforms a random forest. As they only use three covariates this might be explained by the fact that random forest generally favors a large number of covariates (Nussbaum et al., 2018), while the convolutional neural can already get a lot of extra information out of three layers (Wadoux, 2019). In a study that uses 37 covariates with a 1 km resolution to predict multiple soil properties, Wadoux (2019) obtained an AVE of 0.15 and a CCC of 0.46. So the overall accuracy is lower than in this research for the CNN, but again the CNN outperforms the random forest in that study for soil organic carbon. But, when looking more carefully at their results the random forest outperforms the neural network for most of the other soil properties that were modeled. Padarian et al. (2019) found the CNN to be outperforming a Cubist model, but does not show the actual accuracy measures and is therefore difficult to compare.

**Pedological data selection**

To address the fact that RFE is optimized for random forest a selection of covariates was also made based on pedological knowledge to see how the models would respond. Both the random forest and the convolutional neural network obtained lower accuracy when using pedological data, compared to using the RFE selected data, and the convultional neural network was again outperformed by the neural network. The difference in accuracy, between models using RFE data and the models using pedologically selected data, was more or less the same for RF and CNN. Adding terrain covariates was expected to increase the accuracy of the model, as they are commonly used in digital soil mapping studies (e.g. Wadoux et al., 2019; Wadoux, 2019; Padarian et al., 2019; Poggio et al., 2019). It is unclear why this was not the case in this research, but it might have something to do with the resolution of the data and that the patterns in terrain parameters are too local compared to the large patterns found in the climate and vegetation covariates. But as is explained earlier further research is needed on the relation between the resolution of the covariates, covariate selection and the window sizes.

The k-fold run for the convolutional neural networks with the pedological data resulted in exactly the same model architecture as the optimal one that was selected using the RFE data (Table 4.5). As the rest of the ten best performing architectures is different compared to the k-fold run with the RFE selected data, the optimal architecture seems to not only obtain the highest accuracy but also seems to be the most stable model.

### 5.3.4 Prediction maps

The final prediction maps show similarities but also large difference in predictions between the convolutional neural network and random forest predictions. It is clearly visible that the random forest predicted lower values compared to the convolutional neural network. This stronger underfitting was also shown by the relatively high mean error of the random forest. It therefore seems that the CNN is better in predicting the extremes than the random forest. Both prediction maps show the trend from low values in the West to high values in the East, which coincides with the trend observed SOC values (Figure 4.9). The prediction maps show a scattering look in the Eastern side of the region, which is especially visible in the CNN map. These patterns are most likely caused by the yearly average EVI covariate, that shows this same pattern. As this is the most important predictor according to the RFE, it makes sense that these patterns are visible in the final predictions.

## 5.4 Opening the black box with LIME

Unfortunately the LIME algorithm could not be made working for the convolutional neural network. The R-packages has recently been optimized for working with images but it is very specific about the input. It was possible to trick the algorithm to think the covariate data was an actual image, but it would only accept images with three bands (RGB) instead of the 20 covariates that were used as bands in the CNN. When trying to use only three covariates, as to resemble a three band image, it still did not recognize the values as being logical image values. This might be fixed by rescaling the data values to logical values for RGB images, values between 0 and 255 (Ribeiro et al., 2016). However, the model then would no longer resemble the model that was used to make the predictions. The lime package did work with the random forest and gave two general plots as output for the two sets of points (Figure 4.10 & 4.11). Interpreting the results of LIME proved rather difficult and examples were very limited. The original paper (Ribeiro et al., 2016) has over 1500 citations on google scholar, but when searching in the literature that cites LIME it was difficult to find papers that actually used it. Most of them just propose LIME as a possible way to learn from the machine learning models. A couple of papers could be found that actually use it, but these were all for classification tasks (e.g. Ghafouri-Fard et al., 2019). So no reference material was found interpreting the LIME results of a regression task.

The two general explanations plots that were created gave a overview of the weight assigned to each covariate for the different points. They show some surprising results. As LIME is based on local linear models you would expect quite some overlap with the direct correlation between the covariates and SOC (Table 4.1). However, on general EVI layers get assigned quite a low weight compared to climate layers, while the correlation for most EVI layers is stronger. The yearly average EVI even gets almost 0 weight assigned, while according to the RFE this was one of the most important predictors and it also has the highest direct positive correlation with SOC. What causes this difference is unclear and needs further research. Some quite clear differences in attributed weight can be visible when comparing the soil organic carbon points with a large difference in predicted value (Figure 4.11). However, what causes these differences is very difficult to explain.

## 5.5 General discussion and limitations of the research

### 5.5.1 Model

Creating a neural network requires a lot of tests and changes to the structure in order to get the most appropriate architecture for the available data. The model created by Wadoux et al. (2019) was chosen as a starting point, as it showed a promising increase in prediction accuracy compared to a random forest. Using a model architecture from another research allowed for a more efficient start of the project, but might not have been the optimal choice as the used covariates and data were completely different. Building a convolutional neural network from scratch might have yielded better results. However, because the implementation of convolutional neural networks in the field of environmental sciences is still evolving and the available literature is limited it can be difficult to find the right information you need to design your own starting architecture.

#### Improvements of the model

Several improvements could be applied to improve the model. The step-wise approach of building the model used in this research allowed to follow the process and better understand the importance of the model parameters especially during the optimization of the model. Other techniques like Bayesian optimization, might remove this trial and error part and could possible provide a more stable model optimization (Snoek et al., 2012). Further tweaking of the parameters could have had a positive effect on the accuracy, but was not feasible within the time frame of this research. One of the tweaks that could have been made was to do tests with a neural network, that has more neurons in the fully connected/ dense layers, like in Wadoux (2019). This could lead to higher accuracy but would also increase the training time due to an increase in parameters and would increase the risk of overfitting given the limited size of data available (Srivastava et al., 2014).

### 5.5.2 Data

The final accuracy of the convolutional neural network is in general rather low but comparable with other studies in digital soil mapping. This indicates that the accuracy is dependent on the the availability of the data, their spatial distribution and the correlation with available covariates. For the available data and covariates the accuracy thus seems adequate. A convolutional neural network is based on finding patterns in input data (LeCun et al., 2015). The CNN's are often used for image classification tasks in which certain features and boundaries are clearly visible (Krizhevsky et al., 2012). When using low resolution spatial data it might be that the patterns in the data are not always very clear, especially inside a window, where for instance the large scale climate data could be almost homogeneous distributed. This effect might even be stronger when using data that are downscaled from a 1 kilometer resolution to 250 meter resolution. Wadoux et al. (2019) uses data with 25 meter resolution and obtains a higher accuracy than this research, while Wadoux (2019) uses data with 1 kilometer resolution and obtains lower accuracy than this research. This might indicate a relation between the resolution of the data and the maximum obtainable accuracy, but further research is needed to prove this assumption.

**Improvements of the data**

Using more data points that have a better coverage over the entire country would most likely increase the accuracy. Using more data is generally shown to reduce prediction errors (Breiman, 2001). Even when splitting into training, validation and test subsets, each dataset would then still capture most of the variation. As the CNN is based on spatial patterns and these patterns can have many shapes in nature, it is important to capture as much of the patterns as possible. Otherwise the model would find new prediction locations with patterns it does not recognize.

# Chapter 6

# Conclusions

## 6.1 Research questions

**What is the most suitable convolutional neural network architecture to map the spatial distribution of soil organic carbon on a country scale using global data?**

After a 10-fold cross-validation run for 24 models with different combinations of several major model parameters and differences in input, a convolutional neural network that has two convolutional layers and uses Selu activated neurons obtained the highest accuracy. The input data were not augmented and a window size of 15 x 15 cell, i.e. 3750 x 3750 meters seemed optimal. The accuracy of this model was slightly higher than that of the models with other parameter combinations. This was both the case when training with covariates that were selected by a recursive feature elimination and when training with covariates manually selected using pedological knwoledge. Therefore this model seems to be the most stable model out of the 24 tested in this study.

**How does the prediction accuracy of a convolutional neural network compare to that of a random forest when applied to mapping the spatial distribution of top soil SOC?**

The convolutional neural network had a lower accuracy compared to the random forest. The CNN achieved an Amount of Variance Explained of 0.48, a RMSE of 7.27 (g/kg) and a concordance correlation coefficient of 0.67. The random forest obtained better accuracy with an AVE of 0.53, a RMSE of 6.91 (g/kg) and a concordance correlation coefficient of 0.69. In general the point predictions on the test sets reproduce the spatial pattern that is also visible in the observed data, but both models produce a smoothed result. They both underpredict some of the extreme SOC values, but the underprediction of the random forest is stronger than for the convolutional neural network. This is also shown by the mean error (ME), with a ME of 0.38 for the CNN and a ME of 1.09 for the RF. The final prediction maps for CNN and RF produce the trend of low values in the West of the selected subregion to higher values in the East of the section, that is also observed in the measured value. However, the range of values predicted for this region are rather different. The CNN seems to predict the extremes better and produces a range from 0 up to 60 g/kg SOC, while random forest produces a range of values from 5 up to 30 g/kg.

**How can we use the 'Local Interpretable Model-agnostic Explanations' algorithm to provide insight into the structure, functioning and decisions of machine learning soil organic carbon prediction models?**

The local interpretable model-agnostic explanations algorithm did not yet work for the convolutional neural network. The implementation of the algorithm into user-friendly code is still ongoing and for the moment the code does not accept spatial data. The issue was not solved by using three covariates to mimic the three RGB bands of an image as input. However, the algorithm worked for the random forest model. The algorithm is mainly aimed at and used for classification tasks. Interpreting the results for a regression task proved rather difficult. The resulting plots show some patterns of relative importance of the predictors, from which you could defer the relation between the values of the covariates and the SOC predictions for certain point locations. However, linking this to why these differences in weights actually led to the predicted outcome proved very difficult.

## 6.2 General conclusion

Using the spatial structure of the covariates surrounding a point observation by using a convolutional neural network is a promising method to increase the prediction accuracy for several soil properties when comparing it with other machine learning techniques. Even though the convolutional neural network used in this research produced an adequate accuracy, it did not show the expected increase in accuracy compared to the random forest.

Implementing a convolutional neural network not only requires a lot of testing and optimization, but also a clear understanding of computer science to be able to understand some of the underlying mechanisms. Next to that, advanced data handling skills are needed to prepare the input into the correct shape. When building the CNN is compared to implementing a random forest algorithm, CNN takes much more time and is more difficult to understand and implement. Because convolutional neural networks still take a lot more effort than other techniques and the expected increase in accuracy is not always obtained, convolutional neural networks are still far from becoming the go to method for each soil scientist and digital soil mapper.

The Local Interpretable Modeling agnostic Explanations algorithm was promising at first, but it could unfortunately not be implemented for the convolutional neural network. It did work with the random forest, but the results based on continuous predictions instead of a classification task proved difficult to interpret.

# References

Allaire, J. and Chollet, F. (2019). *keras: R Interface to 'Keras'*. R package version 2.2.4.1.

Breiman, L. (2001). “Random forests”. In: *Machine Learning* 45.1, pp. 5–32.

Chabbi, A., Lehmann, J., Ciais, P., Loescher, H. W., Cotrufo, M. F., Don, A., SanClements, M., Schipper, L., Six, J., Smith, P. and Rumpel, C. (2017). “Aligning agriculture and climate policy”. In: *Nature Climate Change* 7.5, pp. 307–309. ISSN: 17586798.

Chen, J., Chen, J., Liao, A., Cao, X., Chen, L., Chen, X., He, C., Han, G., Peng, S., Lu, M., Zhang, W., Tong, X. and Mills, J. (2015). “Global land cover mapping at 30m resolution: A POK-based operational approach”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 103. Global Land Cover Mapping and Monitoring, pp. 7–27. ISSN: 0924-2716.

Chen, X. and Jeong, J. C. (2007). “Enhanced recursive feature elimination”. In: *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pp. 429–435.

Coleman, K. and Jenkinson, D. S. (1996). “RothC-26.3-A Model for the turnover of carbon in soil”. In: *Evaluation of soil organic matter models*. Springer, pp. 237–246.

Follett, R. F. (2001). “Soil management concepts and carbon sequestration in cropland soils”. In: *Soil and Tillage Research* 61.1, pp. 77–92. ISSN: 0167-1987.

Ghafouri-Fard, S., Taheri, M., Omrani, M. D., Daaee, A., Mohammad-Rahimi, H. and Kazazi, H. (2019). “Application of Single-Nucleotide Polymorphisms in the Diagnosis of Autism Spectrum Disorders: A Preliminary Study with Artificial Neural Networks”. In: *Journal of Molecular Neuroscience* 68.4, pp. 515–521. ISSN: 1559-1166.

Goldewijk, K. K., Beusen, A., Doelman, J. and Stehfest, E. (2017). “Anthropogenic land use estimates for the Holocene–HYDE 3.2”. In: *Earth System Science Data* 9.1, pp. 927–953.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge.

GRASS Development Team (2019). *Geographic Resources Analysis Support System (GRASS GIS) Software, Version 7.6*. Open Source Geospatial Foundation.

Greenberg, J. A. and Mattiuzzi, M. (2018). *gdalUtils: Wrappers for the Geospatial Data Abstraction Library (GDAL) Utilities.*

Guyon, I. and Elisseeff, A. (2003). “An introduction to variable and feature selection”. In: *Journal of machine learning research* 3.Mar, pp. 1157–1182.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002). “Gene Selection for Cancer Classification using Support Vector Machines”. In: *Machine Learning* 46.1, pp. 389–422. ISSN: 1573-0565.

Hengl, T., Heuvelink, G. B. M. and Stein, A. (2004). “A generic framework for spatial prediction of soil variables based on regression-kriging”. In: *Geoderma* 120.1, pp. 75–93. ISSN: 0016-7061.

Hengl, T. et al. (2017). “SoilGrids250m: Global gridded soil information based on machine learning”. In: *PLOS ONE* 12.2, pp. 1–40.

Heung, B., Ho, H. C., Zhang, J., Knudby, A., Bulmer, C. E. and Schmidt, M. G. (2016). “An overview and comparison of machine-learning techniques for classification purposes in digital soil mapping”. In: *Geoderma* 265, pp. 62–77. ISSN: 0016-7061.

Heuvelink, G. B. M., Poggio, L., Olmedo, G. F., Angelini, M. E., Bai, Z., De Sousa, L., Batjes, N. H. and Sanderman, J. (2019). *Space-time statistical modelling of soil organic carbon concentrations and stocks. Space-time machine learning SOC predictions and comparison with Tier 1 results.* Tech. rep. ISRIC - World Soil Information (Netherlands); National Agricultural Institute (Argentina) and Woods Hole Research Center (USA), pp. 1–27.

Heuvelink, G. B. M., Kempen, B., Angelini, M. E., Olmedo, G. F., Turdukulov, U. and Ruiperez Gonzalez, M. (2018). *Space-time statistical modelling of soil organic carbon concentration. Description and summary statistics of available soil data for Argentina.* Tech. rep. Wageningen: ISRIC - World Soil Information (Netherlands) and National Institute of Agricultural Technology (Argentina) & Woods Hole Research Center (USA), pp. 1–15.

Hijmans, R. J. (2019). *raster: Geographic Data Analysis and Modeling.* R package version 2.9-5.

Hunziker, P. (2017). *velox: Fast Raster Manipulation and Extraction.* R package version 0.2.0.

Jed Wing, M. K. C. from et al. (2019). *caret: Classification and Regression Training.* R package version 6.0-84.

Jenny, H. (1994). *Factors of soil formation: a system of quantitative pedology.* Courier Corporation.

Jobbágy, E. G. and Jackson, R. B. (2000). “The vertical distribution of soil organic carbon and its relation to climate and vegetation”. In: *Ecological Applications* 10.2, pp. 423–436. ISSN: 1051-0761.

Kempen, B., Ruiperez Gonzalez, M., Angelini, M. E., Oldemo, G. F., Bai, Z., Heuvelink, G. B. M. and Sanderman, J. (2018). *Space-time statistical modelling of soil organic carbon concentration. Covariates for selected pilot area and time period.* Tech. rep. Wageningen: ISRIC - World Soil Information (Netherlands) and National Agricultural Institute (Argentina), pp. 1–6.

Kingma, D. P. and Ba, J. (2014). “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980.*

Klambauer, G., Unterthiner, T., Mayr, A. and Hochreiter, S. (2017). “Self-Normalizing Neural Networks”. In: *Advances in Neural Information Processing Systems 30.* Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett. Curran Associates, Inc., pp. 971–980.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*, pp. 1097–1105.

Krstajic, D., Buturovic, L. J., Leahy, D. E. and Thomas, S. (2014). “Cross-validation pitfalls when selecting and assessing regression and classification models”. English. In: *Journal of Cheminformatics* 6.

Lal, R. (2004a). “Soil Carbon Sequestration Impacts on Global Climate Change and Food Security”. In: *Science* 304.5677, pp. 1623–1627. ISSN: 0036-8075.

Lal, R. (2004b). “Soil carbon sequestration to mitigate climate change”. In: *Geoderma* 123.1, pp. 1–22. ISSN: 0016-7061.

Lal, R. (2005). “Soil carbon sequestration for sustaining agricultural production and improving the environment with particular reference to Brazil”. In: *Journal of Sustainable Agriculture* 26.4, pp. 23–42.

Lamichhane, S., Kumar, L. and Wilson, B. (2019). “Digital soil mapping algorithms and covariates for soil organic carbon mapping and their implications: A review”. In: *Geoderma.* ISSN: 0016-7061.

Latifovic, R., Pouliot, D. and Campbell, J. (2018). “Assessment of Convolution Neural Networks for Surficial Geology Mapping in the South Rae Geological Region, Northwest Territories, Canada”. In: *Remote Sensing* 10.2, p. 307.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). “Deep learning”. In: *Nature* 521, p. 436.

Lin, L. I.-K. (1989). “A Concordance Correlation Coefficient to Evaluate Reproducibility”. In: *Biometrics* 45.1, pp. 255–268. ISSN: 0006341X, 15410420.

Mann, L. K. (1986). “Changes in soil carbon storage after cultivation”. In: *Soil Science* 142.5, pp. 279–288.

McBratney, A. B., Santos, M. L. M. and Minasny, B. (2003). “On digital soil mapping”. In: *Geoderma* 117.1, pp. 3–52. ISSN: 0016-7061.

McCulloch, W. S. and Pitts, W. (1943). “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133. ISSN: 1522-9602.

Minasny, B. et al. (2017). “Soil carbon 4 per mille”. In: *Geoderma* 292, pp. 59–86. ISSN: 0016-7061.

Mishra, U., Lal, R., Liu, D. and Van Meirvenne, M. (2010). "Predicting the Spatial Variation of the Soil Organic Carbon Pool at a Regional Scale". English. In: *Soil Science Society of America Journal* 74, pp. 906–914.

Nussbaum, M., Spiess, K., Baltensweiler, A., Grob, U., Keller, A., Greiner, L., Schaepman, M. E. and Papritz, A. (2018). "Evaluation of digital soil mapping approaches with large sets of environmental covariates". In: *SOIL* 4.1, pp. 1–22.

Padarian, J., Minasny, B. and McBratney, A. B. (2019). "Using deep learning for digital soil mapping". In: *SOIL* 5.1, pp. 79–89.

Parton, W. J., Schimel, D. S., Cole, C. V. and Ojima, D. S. (1987). "Analysis of factors controlling soil organic matter levels in Great Plains Grasslands 1". In: *Soil Science Society of America Journal* 51.5, pp. 1173–1179.

Pedersen, T. L. and Benesty, M. (2018). *lime: Local Interpretable Model-Agnostic Explanations.* R package version 0.4.1.

Pizon, J., Brown, M. and Tucker, C. (2005). "Satellite time series correction of orbital drift artifacts using empirical mode decomposition". In: *Hilbert-Huang transform: introduction and applications*, pp. 167–186.

Poggio, L., Angelini, M., Bai, Z., Batjes, N. H., Heuvelink, G. B. M., Kempen, B., Olmedo, G. F., Ruiperez Gonzalez, M. and Sanderman, J. (2018). *Space-time statistical modelling of soil organic carbon concentration and stocks. escription of the space-time machine learning model.* Tech. rep. Wageningen: ISRIC - World Soil Information (Netherlands) and National Agricultural Institute (Argentina), pp. 1–6.

Poggio, L., Lassauce, A. and Gimona, A. (2019). "Modelling the extent of northern peat soil and its uncertainty with Sentinel: Scotland as example of highly cloudy region". In: *Geoderma* 346, pp. 63–74. ISSN: 0016-7061.

R Core Team (2019). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria.

Reeves, D. W. (1997). "The role of soil organic matter in maintaining soil quality in continuous cropping systems". In: *Soil and Tillage Research* 43.1, pp. 131–167. ISSN: 0167-1987.

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *CoRR* abs/1602.0. arXiv: 1602.04938.

Rodríguez, D., Schulz, G. A., Aleksa, A. and Vuegen, L. T. (2019). "Distribution and Classification of Soils". In: *The Soils of Argentina.* Ed. by G. Rubio, R. S. Lavado and F. X. Pereyra. Cham: Springer International Publishing, pp. 63–79. ISBN: 978-3-319-76853-3.

Rumpel, C., Amiraslani, F., Koutika, L.-S., Smith, P., Whitehead, D. and Wollenberg, E. (2018). "Put more carbon in soils to meet Paris climate pledges". In: *Nature* 564, pp. 32–34.

Schlesinger, W. H. and Andrews, J. A. (2000). “Soil respiration and the global carbon cycle”. In: *Biogeochemistry* 48.1, pp. 7–20.

Schlüter, J. and Grill, T. (2015). “Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks.” In: *ISMIR*, pp. 121–126.

Searchinger, T. D., Wirsenius, S., Beringer, T. and Dumas, P. (2018). “Assessing the efficiency of changes in land use for mitigating climate change”. In: *Nature* 564.7735, pp. 249–253. ISSN: 1476-4687.

Simard, P. Y., Steinkraus, D., Platt, J. C. et al. (2003). “Best practices for convolutional neural networks applied to visual document analysis.” In: *Icdar*. Vol. 3. 2003.

Snoek, J., Larochelle, H. and Adams, R. P. (2012). “Practical bayesian optimization of machine learning algorithms”. In: *Advances in neural information processing systems*, pp. 2951–2959.

Sola, J. and Sevilla, J. (1997). “Importance of input data normalization for the application of neural networks to complex industrial problems”. In: *IEEE Transactions on nuclear science* 44.3, pp. 1464–1468.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Tucker, C. J., Pinzon, J. E., Brown, M. E., Slayback, D. A., Pak, E. W., Mahoney, R., Vermote, E. F. and El Saleous, N. (2005). “An extended AVHRR 8-km NDVI dataset compatible with MODIS and SPOT vegetation NDVI data”. In: *International Journal of Remote Sensing* 26.20, pp. 4485–4498.

Vermote, E., Justice, C., Csiszar, I., Eidenshink, J., Myneni, R., Baret, F., Masuoka, E., Wolfe, R. and Claverie, M. (2014). “NOAA Climate Data Record (CDR) of normalized Difference Vegetation Index (NDVI), Version 4”. In: *NOAA Natl. Clim. Data Cent.*

Wadoux, A. M. J. C., Padarian, J. and Minasny, B. (2019). “Multi-source data integration for soil mapping using deep learning”. In: *SOIL* 5.1, pp. 107–119.

Wadoux, A. M. J. C. (2019). “Using deep learning for multivariate mapping of soil with quantified uncertainty”. In: *Geoderma* 351, pp. 59–70. ISSN: 0016-7061.

Wang, J., Rich, P. M., Price, K. P. and Kettle, W. D. (2005). “Relations between NDVI, Grassland Production, and Crop Yield in the Central Great Plains”. In: *Geocarto International* 20.3, pp. 5–11. eprint: `https://doi.org/10.1080/10106040508542350`.

Webster, R. and Oliver, M. A. (2007). *Geostatistics for environmental scientists*. John Wiley & Sons.

West, T. O. and Post, W. M. (2002). “Soil Organic Carbon Sequestration Rates by Tillage and Crop Rotation”. English. In: *Soil Science Society of America Journal* 66, pp. 1930–1946.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York. ISBN: 978-3-319-24277-4.

Wickham, H., François, R., Henry, L. and Müller, K. (2019). *dplyr: A Grammar of Data Manipulation*. R package version 0.8.1.

Wright, M. N. and Ziegler, A. (2017). "ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R". In: *Journal of Statistical Software* 77.1, pp. 1–17.

Yamamoto, J. K. (2007). "On unbiased backtransform of lognormal kriging estimates". In: *Computational Geosciences* 11.3, pp. 219–234. ISSN: 1573-1499.

# Appendix A

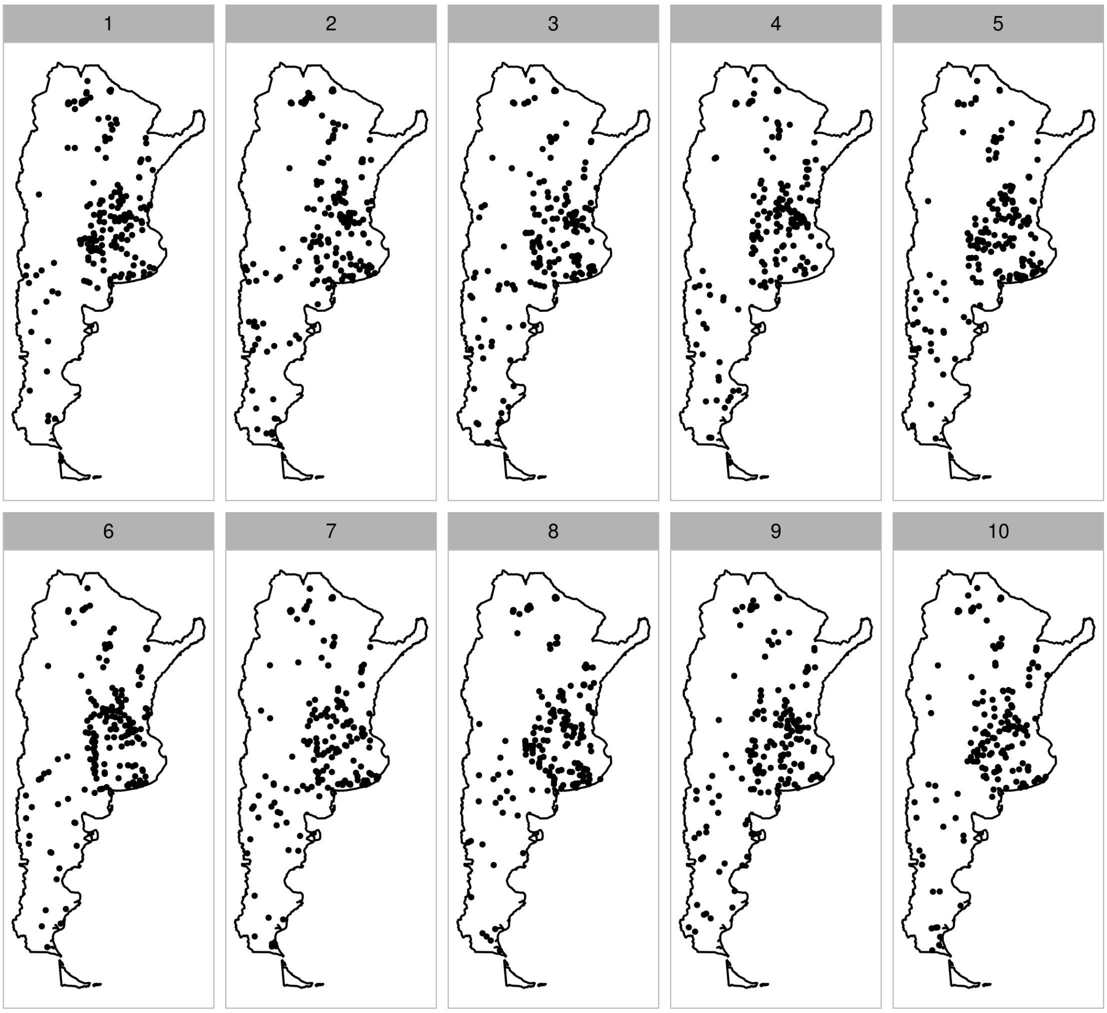
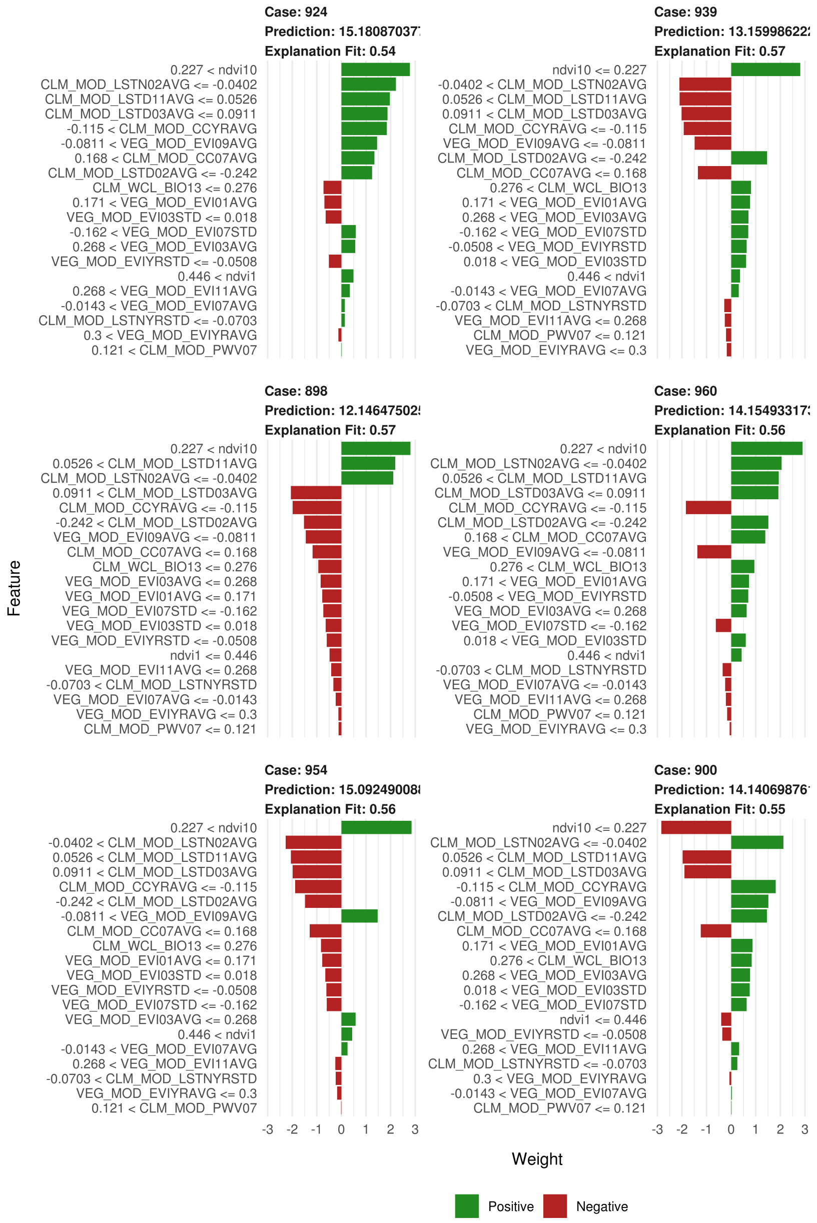


Figure A.1: The 10 test sets used for accuracy assessment in the k-fold cross validation. They are randomly selected from the total dataset and cover the entire country. As is expected the highest density is found in the area South of Buenos Aires. All other data is used for model training.

# Appendix B



Case: 924
Prediction: 15.18087037
Explanation Fit: 0.54
0.227 < ndvi10
CLM_MOD_LSTN02AVG <= -0.0402
CLM_MOD_LSTD11AVG <= 0.0526
CLM_MOD_LSTD03AVG <= 0.0911
-0.115 < CLM_MOD_CCYRAVG
-0.0811 < VEG_MOD_EVI09AVG
0.168 < CLM_MOD_CC07AVG
CLM_MOD_LSTD02AVG <= -0.242
CLM_WCL_BIO13 <= 0.276
0.171 < VEG_MOD_EVI01AVG
VEG_MOD_EVI03STD <= 0.018
-0.162 < VEG_MOD_EVI07STD
0.268 < VEG_MOD_EVI03AVG
VEG_MOD_EVIYRSTD <= -0.0508
0.446 < ndvi1
0.268 < VEG_MOD_EVI11AVG
-0.0143 < VEG_MOD_EVI07AVG
CLM_MOD_LSTNYRSTD <= -0.0703
0.3 < VEG_MOD_EVIYRAVG
0.121 < CLM_MOD_PWV07
Case: 939
Prediction: 13.15998622
Explanation Fit: 0.57
ndvi10 <= 0.227
-0.0402 < CLM_MOD_LSTN02AVG
0.0526 < CLM_MOD_LSTD11AVG
0.0911 < CLM_MOD_LSTD03AVG
CLM_MOD_CCYRAVG <= -0.115
VEG_MOD_EVI09AVG <= -0.0811
CLM_MOD_LSTD02AVG <= -0.242
CLM_MOD_CC07AVG <= 0.168
0.276 < CLM_WCL_BIO13
0.171 < VEG_MOD_EVI01AVG
0.268 < VEG_MOD_EVI03AVG
-0.162 < VEG_MOD_EVI07STD
-0.0508 < VEG_MOD_EVIYRSTD
0.018 < VEG_MOD_EVI03STD
0.446 < ndvi1
-0.0143 < VEG_MOD_EVI07AVG
-0.0703 < CLM_MOD_LSTNYRSTD
VEG_MOD_EVI11AVG <= 0.268
CLM_MOD_PWV07 <= 0.121
VEG_MOD_EVIYRAVG <= 0.3
Case: 898
Prediction: 12.14647502
Explanation Fit: 0.57
0.227 < ndvi10
0.0526 < CLM_MOD_LSTD11AVG
CLM_MOD_LSTN02AVG <= -0.0402
0.0911 < CLM_MOD_LSTD03AVG
CLM_MOD_CCYRAVG <= -0.115
-0.242 < CLM_MOD_LSTD02AVG
VEG_MOD_EVI09AVG <= -0.0811
CLM_MOD_CC07AVG <= 0.168
CLM_WCL_BIO13 <= 0.276
VEG_MOD_EVI03AVG <= 0.268
VEG_MOD_EVI01AVG <= 0.171
VEG_MOD_EVI07STD <= -0.162
VEG_MOD_EVI03STD <= 0.018
VEG_MOD_EVIYRSTD <= -0.0508
ndvi1 <= 0.446
VEG_MOD_EVI11AVG <= 0.268
-0.0703 < CLM_MOD_LSTNYRSTD
VEG_MOD_EVI07AVG <= -0.0143
VEG_MOD_EVIYRAVG <= 0.3
CLM_MOD_PWV07 <= 0.121
Case: 960
Prediction: 14.15493317
Explanation Fit: 0.56
0.227 < ndvi10
CLM_MOD_LSTN02AVG <= -0.0402
0.0526 < CLM_MOD_LSTD11AVG
CLM_MOD_LSTD03AVG <= 0.0911
CLM_MOD_CCYRAVG <= -0.115
CLM_MOD_LSTD02AVG <= -0.242
0.168 < CLM_MOD_CC07AVG
VEG_MOD_EVI09AVG <= -0.0811
0.276 < CLM_WCL_BIO13
0.171 < VEG_MOD_EVI01AVG
-0.0508 < VEG_MOD_EVIYRSTD
VEG_MOD_EVI03AVG <= 0.268
VEG_MOD_EVI07STD <= -0.162
0.018 < VEG_MOD_EVI03STD
0.446 < ndvi1
-0.0703 < CLM_MOD_LSTNYRSTD
VEG_MOD_EVI07AVG <= -0.0143
VEG_MOD_EVI11AVG <= 0.268
CLM_MOD_PWV07 <= 0.121
VEG_MOD_EVIYRAVG <= 0.3
Case: 954
Prediction: 15.09249008
Explanation Fit: 0.56
0.227 < ndvi10
-0.0402 < CLM_MOD_LSTN02AVG
0.0526 < CLM_MOD_LSTD11AVG
0.0911 < CLM_MOD_LSTD03AVG
CLM_MOD_CCYRAVG <= -0.115
-0.242 < CLM_MOD_LSTD02AVG
-0.0811 < VEG_MOD_EVI09AVG
CLM_MOD_CC07AVG <= 0.168
CLM_WCL_BIO13 <= 0.276
VEG_MOD_EVI01AVG <= 0.171
VEG_MOD_EVI03STD <= 0.018
VEG_MOD_EVIYRSTD <= -0.0508
VEG_MOD_EVI07STD <= -0.162
VEG_MOD_EVI03AVG <= 0.268
0.446 < ndvi1
-0.0143 < VEG_MOD_EVI07AVG
0.268 < VEG_MOD_EVI11AVG
-0.0703 < CLM_MOD_LSTNYRSTD
VEG_MOD_EVIYRAVG <= 0.3
0.121 < CLM_MOD_PWV07
Case: 900
Prediction: 14.14069876
Explanation Fit: 0.55
ndvi10 <= 0.227
CLM_MOD_LSTN02AVG <= -0.0402
0.0526 < CLM_MOD_LSTD11AVG
0.0911 < CLM_MOD_LSTD03AVG
-0.115 < CLM_MOD_CCYRAVG
-0.0811 < VEG_MOD_EVI09AVG
CLM_MOD_LSTD02AVG <= -0.242
CLM_MOD_CC07AVG <= 0.168
0.171 < VEG_MOD_EVI01AVG
0.276 < CLM_WCL_BIO13
0.268 < VEG_MOD_EVI03AVG
0.018 < VEG_MOD_EVI03STD
-0.162 < VEG_MOD_EVI07STD
ndvi1 <= 0.446
VEG_MOD_EVIYRSTD <= -0.0508
0.268 < VEG_MOD_EVI11AVG
CLM_MOD_LSTNYRSTD <= -0.0703
0.3 < VEG_MOD_EVIYRAVG
-0.0143 < VEG_MOD_EVI07AVG
CLM_MOD_PWV07 <= 0.121
Feature
-3 -2 -1 0 1 2 3
-3 -2 -1 0 1 2 3
Weight
🟩 Positive
🟥 Negative

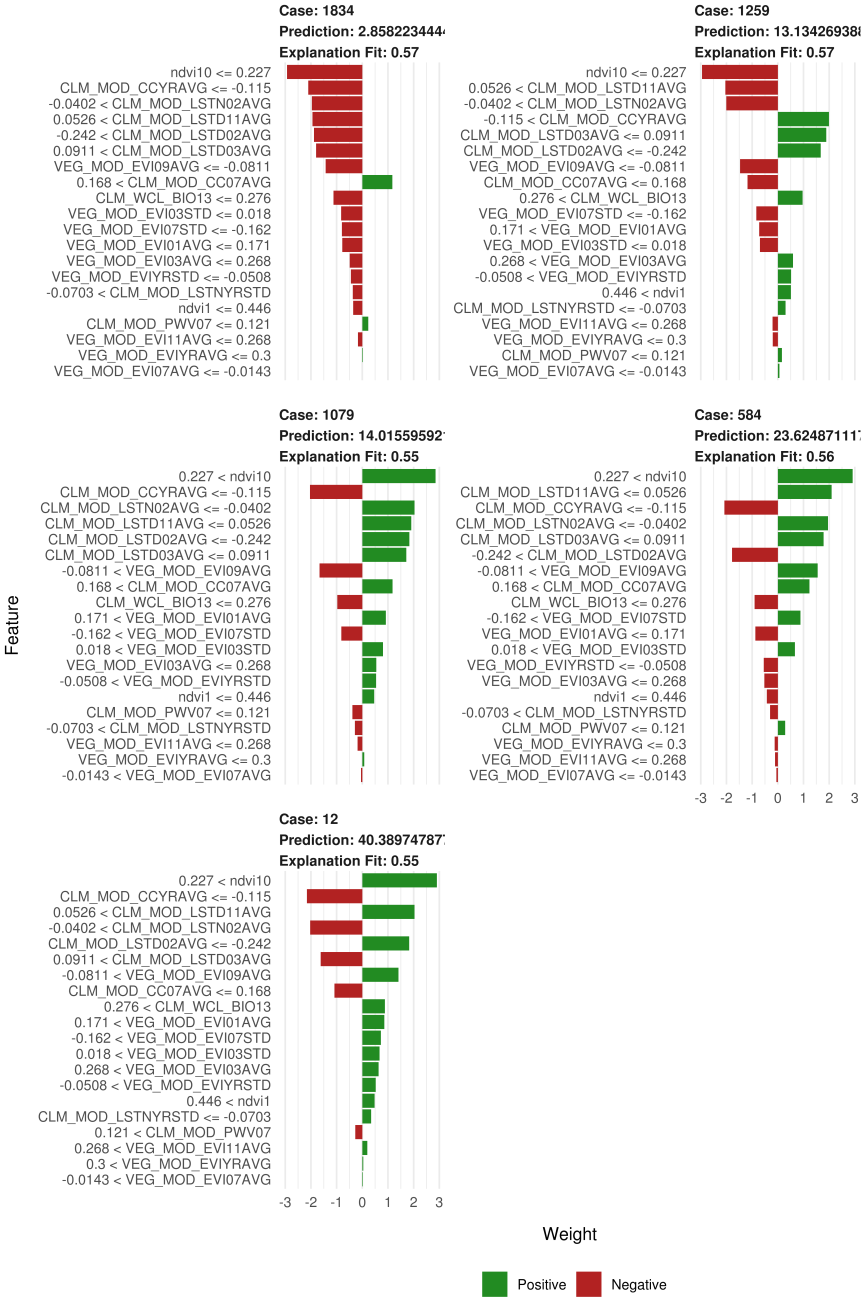


Figure B.1: Feature plots of LIME, that show the contribution of a covariate to the final predictions, expressed in weight. Upper plot shows the results for six points with a predicted SOC concentration around 14 g/kg. The lower plot gives the results for the five points that cover a large range of SOC values

# Appendix C

Table C.1: The 20 RFE selected covariates including the name codes

| Code | Description |
|---|---|
| VEG MOD EVIYRAVG | Long-term yearly average MODIS Enhanced Vegetation Index (EVI) |
| NDVI10 | Normalized Difference Vegetation Index October, November, December |
| VEG MOD EVI09AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months September and October |
| VEG MOD EVI03AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months March and April |
| VEG MOD EVI11AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months November and December |
| VEG MOD EVI01AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months January and February |
| VEG MOD EVI03STD | Long-term s.d. of the monthly MODIS Enhanced Vegetation Index (EVI) for months March and April |
| VEG MOD EVI07AVG | Long-term averaged mean monthly MODIS Enhanced Vegetation Index (EVI) for months July and August |
| VEG MOD EVI07STD | Long-term s.d. of the monthly MODIS Enhanced Vegetation Index (EVI) for months July and August |
| NDVI1 | Normalized Difference Vegetation Index January, February, March |
| VEG MOD EVIYRSTD | Standard Deviation yearly MODIS Enhanced Vegetation Index (EVI) |
| CLM WCL BIO13 | Precipitation of Wettest Month |
| CLM MOD CCYRAVG | Long-term averaged mean cloud cover |
| CLM MOD PWV07 | Long-term averaged mean monthly MODIS Precipitable Water Vapor in cm for months July and August |
| CLM MOD CC07AVG | Long-term averaged monthly cloud cover July |
| CLM MOD LSTN02AVG | Long-term averaged mean monthly surface temperature (nighttime) MODIS February |
| CLM MOD LSTD03AVG | Long-term averaged mean monthly surface temperature (daytime) MODIS March |
| CLM MOD LSTD11AVG | Long-term averaged mean monthly surface temperature (daytime) MODIS November |
| CLM MOD LSTNYRSTD | Long-term s.d. of the monthly surface temperature (nighttime) MODIS Yearly |
| CLM MOD LSTD02AVG | Long-term averaged mean monthly surface temperature (daytime) MODIS February |