High resolution mapping of topsoil properties in the Western Balkans using the LUCAS soil survey

Authors:

Cristina Arias-Navarro[1], Bexhet Islamaj[2], Dragana Vidojevic[3], Pandi Zdruli[4], Felipe Yunta-Mezquita[1], Panos Panagos[1], Piotr Wojda[1], Arwyn Jones[1]

[1] European Commission, Joint Research Centre (JRC), Ispra, Italy

[2] Geoinformation Centre for the Western Balkans, Tirana, Albania

[3] Environmental Protection Agency, Republic of Serbia

[4] Mediterranean Agronomic Institute of Bari, Italy

Correspondence email:

Cristina.ARIAS-NAVARRO@ec.europa.eu

SUPPLEMENTARY INFORMATION: Methodology, Analysis and Maps generated

Introduction

The Supplementary Information document presents a comprehensive description of the methods and kernel functions implemented in the accompanying codebase. It details the optimisation procedures employed and explains the rationale behind the selection of specific kernels to obtain optimal predictive performance. In addition, the document includes five well-documented Python scripts: (i) a script that specifies the chosen kernel and its associated parameters; (ii) a script for predicting chemical properties; (iii) a script that generates predictions over a defined grid of points; (iv) a script that conducts spatial and statistical analyses of the results; and (v) a script devoted to quality-assurance/quality-control (QA/QC) of the predicted data and maps.

Map production

Gaussian Process Regression (GPR) is a powerful statistical method for data modeling and forecasting. It offers flexibility for nonlinear modeling and allows for the accounting of forecast uncertainty, making it a preferred choice in fields as diverse as environmental sciences, geostatistics, and machine learning (Rasmussen, 2004; Ballabio *et al.*, 2019; Kuok *et al.*, 2025). This document describes the methods and kernels applied in the given code, explaining the optimization processes and the choices for kernels used to achieve optimal results.

Data Inputs

955 Sample Points collected in the field.

NUTS Boundaries (Shapefile).

Country Names derived from spatial analysis using country boundaries(Shapefile).

Climate Data (Climatic zone names and related data – Shapefile).

Pedoclimate Data (Shapefile).

Primary Land Cover from CLC 2018

Digital Elevation Model from Copernicus for elevation of predicted sample points.

Data Outputs:

Grid Points spaced at 500m intervals

CSV file with Predicted Chemical Properties for each grid point

TIFF Raster Files for each soil property with a resolution of 200m x 200m

Analysis and Statistics.

Data Prediction and Maps preparations

The preparation of the final product, which includes the points generated with predicted chemical analyses, raster images, and maps, followed several key steps:

Grid Point Generation

Script Development & Model Training - Gaussian Process Regression, Kernel and Parameter selection

Map & Raster Image Preparation

Statistical Analysis & Reporting.

Grid Point Generation – Creating points on a 500m grid

To generate points, a grid of points spaced 500 meters apart was selected, which provides sufficient density to create polygons/maps and reflect land qualities at each location. Using the specified data inputs and a spatial join operation with the Digital Elevation Model (DEM), CLC, Country Boundaries, NUTS, Climate Shapes, and Pedoclimate Shapes, attributes were retrieved for each point as follows:

Categorical Features:
['Climatic_zones_name', 'LC1_code', 'SOIL_ID', 'TrimWRB', 'S01_WRB', 'S02_WRB', 'pedoCLIM02', 'new_pedcli', 'NUTS_2_country_code']

Numerical Features:
['POINT_Z']

Script Development - GPR, Kernel and Parameter selection.

To generate high-resolution maps of top-soil property distribution across the Western Balkan countries, we employ Gaussian Process Regression (GPR) as described in Ballabio et al. (2019). The authors demonstrate that GPR can be instantiated with a Matérn covariance function, which extends the standard Radial Basis Function (RBF) kernel by introducing a smoothness parameter v that governs the differentiability of the process. In our workflow we evaluate both the RBF and Matérn kernels; the comparative performance of each kernel is discussed in the subsequent chapters.

The repository contains five fully documented Python scripts:

- Kernel specification – defines the selected kernel (RBF or Matérn) and its hyper-parameters and performs hyper-parameter optimisation.

- Chemical-property prediction – fits the GPR model to the training dataset and produces point-wise predictions of the target soil attributes.
- Grid prediction – extrapolates the fitted model onto a regular spatial grid to generate continuous property surfaces.
- Spatial and statistical analysis – calculates descriptive statistics, variograms and other spatial diagnostics for the predicted fields.
- QA/QC – validates the predictions against independent observations, assesses uncertainty, and produces quality-controlled maps.

These scripts collectively implement the methodological pipeline required to obtain robust, spatially explicit estimates of top-soil characteristics.

GPR was used to predict the Chemical values, and the whole process included five steps.

1. Create the Grid of Points which meets the requirements to create baseline topsoil chemical property maps at a 250-meter resolution for the five countries.
2. Predict the chemical properties of the points generated in Step 1 using Matern Kenel
3. Predict the chemical properties of the points generated in Step 1 using RBF+White Kernel
4. Performs a spatial comparison and statistical analysis between actual and predicted soil property values.
5. Assess the accuracy and reliability of Gaussian Process Regression predictions by comparing the predicted results created with Different GPR Kernels and Parameters.

Each of the steps is automated by developing python scripts which are presented below and from this set of scripts is possible to find in the corresponding annex of the Scripts as well a detailed document for each of them.

Script to create Predicted Points.py

- Gaussian_process_soil_prediction_Matern_Kernel.py
- Gaussian_process_soil_prediction-RBF_plus_White_kernels.py
- Check_Predicted-vs-Measured_Chemical_Properties.py
- Make_Stats_for_the_Predicted_Chemical_Properties.py

To predict the results for the generated points, categorised attributes were used and processed through Gaussian Process Regression (GPR) as follows:

Categorical Features:
['Climatic_zones_name', 'LC1_code', 'SOIL_ID', 'TrimWRB', 'S01_WRB', 'S02_WRB', 'pedoCLIM02', 'new_pedcli', 'NUTS_2_country_code']

Numerical Features:
['POINT_X', 'POINT_Y', 'POINT_Z']

Based on these categorical and numerical features, as well as trained data from 955 sample points, GPR was applied to predict the following target variables:

['PHH2O', 'PHCaCl2', 'ElectricalConductivity', 'OrganicCarbon', 'Carbonate', 'Phosphorus', 'TotalNitrogen', 'ExtractablePotassium', 'CationExchange', 'Clay', 'Sand', 'Silt', 'CoarseFragments']

Gaussian Process Regression Training Process

To ensure the most accurate results, approximately 100 different kernels and parameter configurations were tested. Among them, the Matérn Kernel demonstrated superior predictive performance for specific physic-chemical parameters, namely: Organic Carbon, Cation Exchange Capacity, Sand, Silt, and Coarse Fragments. On the other hand, the RBF + White Kernel produced more accurate predictions for pH in H_2O (pH_{H_2O}), pH in $CaCl_2$ (pH_{CaCl_2}), Electrical Conductivity, Carbonate, Phosphorus, Total Nitrogen, Extractable Potassium, and Clay. Gaussian Process Regression (GPR) using the RBF + White Kernel provided significantly improved precision for these parameters. These conclusions were reached through extensive testing using Python scripts developed specifically for this purpose. The scripts were used to evaluate predictions across multiple kernel configurations, ultimately identifying the RBF + White Kernel combination and the Matérn Kernel, as suggested in the referenced paper, as the most effective approaches. The experimentation/elaboration involved testing over 100 different kernel types, kernel combinations, and parameter settings. Only two methods concluded in two Python scripts were selected to process two csv files containing the respective predicted results from Matérn Kernel and RBF + White Kernel combinations.

Kernel Prediction Analysis

Predictive analysis results of both methods are shown in Table S1 and Table S2. These findings were derived by analysing predicted values from both the RBF and Matérn kernels. This table is available at the Annex (Statistical_Analysis_20250130.xlsx), as a spreadsheet or can be reproduced in csv file (Statistical_Analysis_20250130.csv) by running the python scripts.

It is important to note that using RBF, it is possible to make predictions for points spaced at intervals as small as 200m, whereas Matérn can make predictions only for a larger distance of points at the level of 1,000 m. For finer spatial resolutions, a high-performance server—rather than a standard workstation—would be required to efficiently compute the predicted results.

Table S1. Analysis of the predicted results using RBF+White Kernels

RBF+White Kernel													
Chemical Property	GPR_Error_Pred_Max	GPR_Error_Pred_Min	Max_Measured_value	Max_Predicted_value	Points between Max(Measured-Predicted)	Between Max %	Max Range %	Min_Measured	Min_Predicted	Points between Mins(Measured-	Between Mins %	Min Range %	TotalPct %
pH H_2O	0.93	0.67	8.89	8.12	14	1.47	8.66	3.77	4.34	14	1.47	6.41	2.94
pH $CaCl_2$	0.93	0.69	8.2	7.61	10	1.05	7.2	3.3	3.98	14	1.47	8.29	2.52
Electrical Conductivity (dS/m)	1.07	0.91	251	51.98	20	2.09	79.29	2.13	6.45	135	14.14	1.72	16.23
Organic Carbon (g/kg)	0.96	0.69	195.1	120.71	22	2.3	38.13	1.7	0.2	0	0	-0.77	2.3
Carbonate (g/kg)	1.05	0.88	764	199.09	36	3.77	73.94	0	0	0	0	0	3.77

Phosphorus (mg/kg)	1	0.91	245.7	37.02	82	8.59	84.93	0	0	0	0	0	8.59
Total Nitrogen (g/kg)	0.96	0.74	19.6	9.1	22	2.3	53.57	0.2	0.58	20	2.09	1.94	4.39
Extractable Potassium	1.02	0.89	3536.4	664.64	15	1.57	81.21	26.5	69.83	88	9.21	1.23	10.78
Cation Exchange Capacity	1	0.8	104.9	46.24	57	5.97	55.92	0	6.2	61	6.39	0	12.36
Clay %	0.97	0.78	60	43.49	52	5.45	27.52	1	10.52	46	4.82	15.87	10.27
Sand %	0.96	0.71	94	57.63	40	4.19	38.69	2	6.3	115	12.04	4.57	16.23
Silt %	1	0.76	70	60.98	126	13.19	12.89	4	31.23	53	5.55	38.9	18.74
Coarse Fragments	0.99	0.86	79	23.87	52	5.45	69.78	0	0	0	0	0	5.45

Table S2 Analysis of the predicted results using Matern Kernel

	Matern Kernel												
Chemical Property	GPR_Error_Pred_Max	GPR_Error_Pred_Min	Max_Measured_value	Max_Predicted_value	Points between Max(Measured-	Between Max %	Max Range %	Min_Measured	Min_Predicted	Points between Mins(Measured-	Between Mins %	Min Range %	TotalPct %
pH H_2O	0.92	0.66	8.89	8.09	16	1.68	9	3.77	4.4	18	1.88	7.09	3.56
pH $CaCl_2$	0.92	0.69	8.2	7.65	10	1.05	6.71	3.3	3.97	14	1.47	8.17	2.52
Electrical Conductivity (dS/m	1.08	0.91	251	51.99	20	2.09	79.29	2.13	6.63	142	14.87	1.79	16.96
Organic Carbon (g/kg)	0.96	0.68	195.1	122.79	19	1.99	37.06	1.7	0	0	0	-0.87	1.99
Carbonate (g/kg)	1.02	0.88	764	193.56	37	3.87	74.66	0	0	0	0	0	3.87
Phosphorus (g/kg)	1	0.91	245.7	37.16	82	8.59	84.88	0	1.07	266	27.85	0	36.44
Total Nitrogen	0.95	0.73	19.6	8.88	24	2.51	54.69	0.2	0.32	5	0.52	0.61	3.03
Extractable Potassium	1.02	0.89	3536.4	660.06	16	1.68	81.34	26.5	77.39	123	12.88	1.44	14.56

Cation Exchange Capacity	1	0.8	104.9	45.52	59	6.18	56.61	0	6.46	67	7.02	0	13.2
Clay %	0.97	0.77	60	43.45	52	5.45	27.58	1	10.6	46	4.82	16	10.27
Sand %	0.97	0.67	94	60.02	31	3.25	36.15	2	5.75	73	7.64	3.99	10.89
Silt %	1	0.74	70	61.26	93	9.74	12.49	4	30.93	41	4.29	38.47	14.03
Coarse Fragments	0.99	0.85	79	25.07	46	4.82	68.27	0	0	0	0	0	4.82

Prediction uncertainty

In this section we analyse and compare the results of the RBF and Matérn kernels in terms of prediction uncertainty, as presented in the Table S3.

Table S3. Prediction uncertainty between RBF and Matérn kernels models

RBF+White Kernel			Matern Kernel		Max Error Prediction Comparison	Min Error Prediction Comparison
Chemical Property	GPR_Error Pred_Max	GPR_Error	GPR_Error	GPR_Error Pred_Min		
pH H_2O	0.93	0.67	0.92	0.66	Matern Performs Better	Matern Performs Better
pH $CaCl_2$	0.93	0.69	0.92	0.69	Matern Performs Better	RBF Performs Better
Electrical Conductivity (dS/m)	1.07	0.91	1.08	0.91	RBF Performs Better	RBF Performs Better
Organic Carbon (g/kg)	0.96	0.69	0.96	0.68	RBF Performs Better	Matern Performs Better
Carbonate (g/kg)	1.05	0.88	1.02	0.88	Matern Performs Better	RBF Performs Better
Phosphorus (g/kg)	1	0.91	1	0.91	RBF Performs Better	RBF Performs Better
Total Nitrogen (g/kg)	0.96	0.74	0.95	0.73	Matern Performs Better	Matern Performs Better

Extractable Potassium (g/kg)	1.02	0.89	1.02	0.89	RBF Performs Better	RBF Performs Better
Cation Exchange Capacity	1	0.8	1	0.8	RBF Performs Better	RBF Performs Better
Clay %	0.97	0.78	0.97	0.77	RBF Performs Better	Matern Performs Better
Sand %	0.96	0.71	0.97	0.67	RBF Performs Better	Matern Performs Better
Silt %	1	0.76	1	0.74	RBF Performs Better	Matern Performs Better
Coarse Fragments	0.99	0.86	0.99	0.85	RBF Performs Better	Matern Performs Better

The GPR_Error_Pred_Max and GPR_Error_Pred_Min columns indicate the predicted error levels for both kernels. After data checking and validation was realized that they are nearly identical, with a negligible difference of approximately 0.01. However, RBF appears to perform better for 15 physic-chemical characteristics, while Matérn performs better for the remaining 11 parameters.

Range of Predicted Maximum and Minimum Values

Next was to analyse the range of predicted maximum and minimum values, as presented in the Table S4.

Table S4. Range of predicted maximum and minimum values

RBF+White Kernel	Matern Kernel	Comparison of Maximum	Comparison of Minimum

Chemical Property	Max Predicted Value	Min Predicted Value	Max Predicted Value	Min Predicted Value	Property Value Predictions	Property Value Predictions
pH H_2O	8.12	4.34	8.09	4.4	RBF Performs Better	Matern Performs Better
pH $CaCl_2$	7.61	3.98	7.65	3.97	Matern Performs Better	Matern Performs Better
Electrical Conductivity (dS/m)	51.98	6.45	51.99	6.63	Matern Performs Better	RBF Performs Better
Organic Carbon (g/kg)	120.71	0.2	122.79	0	Matern Performs Better	Matern Performs Better
Carbonate (g/kg)	199.09	0	193.56	0	RBF Performs Better	Equal
Phosphorus (g/kg)	37.02	0	37.16	1.07	Matern Performs Better	RBF Performs Better
Total Nitrogen (g/kg)	9.1	0.58	8.88	0.32	RBF Performs Better	Matern Performs Better
Extractable Potassium	664.64	69.83	660.06	77.39	RBF Performs Better	RBF Performs Better
Cation Exchange Capacity	46.24	6.2	45.52	6.46	RBF Performs Better	RBF Performs Better
Clay %	43.49	10.52	43.45	10.6	RBF Performs Better	RBF Performs Better
Sand %	57.63	6.3	60.02	5.75	Matern Performs Better	Matern Performs Better
Silt %	60.98	31.23	61.26	30.93	Matern Performs Better	Matern Performs Better

Coarse Fragments	23.87	0	25.07	0	Matern Performs Better	Equal

As observed from the comparison, Matérn performs better for 13 properties, while RBF performs better for 11. This indicates that both methods deliver nearly identical performance, with Matérn showing a slight advantage.

If the maximum predicted value for one method is higher than that of the other, then the first method covers a broader range of predicted values. A broader range suggests that the method captures more of the measured values, making it closer to reality. The opposite holds true: if a method predicts a lower minimum, its range is wider. A broader range suggests that the method captures more of the measured values, making it closer to reality.

Measured Points Outside the Predicted Range

The table S5 below represents the measured points that fall outside the predicted range. In other words, it indicates the number of measured points for which no predicted values exist in the final results.

Table S5. Measured points that fall outside the predicted range

RBF+White Kernel			Matern Kernel			
Chemical Property	Points between Max	Points between Mins	Points between Max	Points between Mins	Comparison of points between Max Measured And Max Predicted Values	Comparison of points between Min Measured And Min Predicted Values

pH H_2O	14	14	16	18	RBF Performs Better	Matern Performs Better
pH $CaCl_2$	10	14	10	14	RBF Performs Better	Equal
Electrical Conductivity (dS/m)	20	135	20	142	Equal	Matern Performs Better
Organic Carbon (g/kg)	22	0	19	0	Equal	Equal
Carbonate (g/kg)	36	0	37	0	Matern Performs Better	Equal
Phosphorus (g/kg)	82	0	82	266	RBF Performs Better	Matern Performs Better
Total Nitrogen (g/kg)	22	20	24	5	Equal	RBF Performs Better
Extractable Potassium (g/kg)	15	88	16	123	RBF Performs Better	Matern Performs Better
Cation Exchange Capacity	57	61	59	67	RBF Performs Better	Matern Performs Better
Clay %	52	46	52	46	RBF Performs Better	Equal
Sand %	40	115	31	73	Equal	RBF Performs Better
Silt %	126	53	93	41	Matern Performs Better	RBF Performs Better
Coarse Fragments	52	0	46	0	Matern Performs Better	Equal

It was observed that RBF performs better, as it has fewer points that fall outside the predicted value range in the final results. Specifically, RBF outperforms Matérn in 9 physic-chemical characteristics, while Matérn performs better in 8. This indicates that the predicted results from RBF are more representative compared to those from the Matérn

Kernel. In other words, RBF provides a better fit for the dataset, capturing a broader range of measured values.

Comprehensive Analysis of Prediction Comparison Results

Here we examine the total results to understand how representative the predicted results are for both kernels. The table S6 presents the percentage of LUCAS measured points (training data) that are not included in the final predicted results.

Table S6. Percentage of LUCAS measured points (training data) that are not included in the final predicted results

RBF+White Kernel		Matern Kernel	Who Performs Better
Chemical Property	TotalPct %	TotalPct %	
pH H_2O	2.94	3.56	RBF Performs Better
pH $CaCl_2$	2.52	2.52	Equal
Electrical Conductivity (dS/m)	16.23	16.96	RBF Performs Better
Organic Carbon (g/kg)	2.3	1.99	Matern Performs Better
Carbonate (g/kg)	3.77	3.87	RBF Performs Better
Phosphorus (g/kg)	8.59	36.44	RBF Performs Better
Total Nitrogen (g/kg)	4.39	3.03	Matern Performs Better
Extractable Potassium (g/kg)	10.78	14.56	RBF Performs Better
Cation Exchange Capacity	12.36	13.2	RBF Performs Better
Clay %	10.27	10.27	Equal
Sand %	16.23	10.89	Matern Performs Better

Silt %	18.74	14.03	Matern Performs Better
Coarse Fragments	5.45	4.82	Matern Performs Better

From these results it becomes clear that the RBF+White Kernel performs better in 6 chemical parameters**,** while Matérn performs better in 5**.**

RBF+White Kernel delivers superior performance in nearly all major parameters, including pH, EC, Carbonate, Phosphorus, Potassium, and CEC**.** Matérn Kernel, on the other hand, performs better for Organic Carbon (OC), Nitrogen (N). For the remaining two parameters**,** **(**pH $CaCl_2$ and Clay), both kernels yielded identical predicted results.

GPR Utilising Matern Kernel vs RBF+White Kernels

First, it was analyzed how Matern kernels act upon while performing the predictions.

Matern Kernel

The following section represents the L-BFGS-B Optimization Process (Table S7). L-BFGS-B (Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Box constraints [1] [1]) is an optimization algorithm used for minimizing functions.

Table S7. L-BFGS-B Optimization Process

Running L-BFGS-B (Scipy implementation) Code:			
runtime	**i**	**f**	**|g|**
02s24	10	1.05E+03	1.20E+00
08s85	40	1.05E+03	8.47E+01
RuntimeWarning:overflow encountered in divide			
RuntimeWarning:invalid value encountered in multiply			
RuntimeWarning:invalid value encountered in cos			
23s79	105	1.05E+03	1.23E-01
25s80	113	1.05E+03	1.43E-06
Runtime: 25s80			

[1] [1] A LIMITED MEMORY ALGORITHM FOR BOUND CONSTRAINED OPTIMIZATION by Richard H Byrd Peihuang Lu Jorge Nocedal and Ciyou Zhu
https://users.iems.northwestern.edu/~nocedal/PDFfiles/limited.pdf

Optimization status: Converged			
Running L-BFGS-B (Scipy implementation) Code:			
runtime	**i**	**f**	**|g|**
01s16	5	1.08E+03	8.68E+01
03s48	15	1.08E+03	1.18E+01
12s79	55	1.08E+03	2.39E+01
15s79	68	1.07E+03	5.40E-06
Runtime: 15s79			
Optimization status: Converged			

The table columns represent:

runtime: Time elapsed since the optimization started.

i: Iteration number.

f: Function value (objective function being minimized).

|g|: Gradient norm (indicating how much the function is changing).

After ~2.24 seconds, the optimization reached iteration 10, with a function value of 1053.709 and a gradient norm of 1.2. Several RuntimeWarnings appear in the output, indicating numerical issues. These errors occurred in functions within the GPy package, specifically in the periodic kernel computations. The optimization continued despite the warnings successfully converged. The second run of L-BFGS-B also converged, it generates warnings, indicating a higher likelihood of errors during execution.

RBF+WHITE Kernels.

Then the Matérn kernel analysis was performed during the prediction process to explore how the GPR process is optimized using the RBF+White Kernel. This output shows the execution of the L-BFGS-B optimization algorithm using SciPy's implementation. Table S8 presents a breakdown of the key elements.

Table S8. Execution of the L-BFGS-B optimization algorithm

R unning L-BFGS-B (Scipy implementation) Code:			
runtime	**i**	**f**	**|g|**

00s14	1	1.35E+03	2.18E+04
01s19	8	1.05E+03	7.62E+00
02s49	16	1.05E+03	1.04E-07
Runtime: 02s49			
Optimization status: Converged			
Running L-BFGS-B (Scipy implementation) Code:			
runtime	**i**	**f**	**|g|**
01s27	7	1.08E+03	7.72E+01
02s75	16	1.08E+03	8.80E-09
Runtime: 02s75			
Optimization status: Converged			

Table Columns Explained:

runtime: The elapsed time since the optimization started.

i: The iteration number.

f: The function value (objective function being minimized).

|g|: The norm of the gradient, which indicates how close the solution is to an optimal point. A very small gradient norm suggests convergence.

First Optimization Run: Status**:** Converged**;** Second Optimization Run: The L-BFGS-B algorithm also converged**,** indicating that it successfully found an optimal solution. This process runs without errors or warnings, which suggests that the RBF+White Kernels perform better than the Mattern Kernel in terms of optimization stability.

GPR method Conclusions

While both kernels enable GPR to perform well**,** the RBF+White Kernel demonstrates superior performance overall. As a result, the maps and final predictions presented in this study are based on the RBF+White Kernel combination**.**

Data analysis

Another aspect of the analysis involves comparing the predicted values with the trained values. While these values are not expected to be identical, they should converge. To verify this, a spatial join was performed between each predicted point and the nearest trained point (955 points from the LUCAS database). This allows for an assessment of how the predicted values compare to the training data. The analysis reveals that most predicted values are closely aligned with their corresponding trained values, with some variations. Then the dataset presented in Table S5 was examined, which has been generated by Script 4 – Check Predicted-vs-Measured_Chemical-Properties.py. The data in Table S9 originates from a CSV file produced by the script. However, for better readability, the columns have been transposed into rows.

Table S9. Statistical comparisons for each predicted point against the closest measured (field-sampled) point.

	PH H_2O	PH $CaCl_2$	Electrical Conductivity	OrganicCarbon	Carbonate	Phosphorus	Total Nitrogen	Extractable Potassium	CationExchange	Clay	Sand	Silt	Coarse Fragments
Difference 1	-3.05 to -2.28	-2.92 to -2.15	-238.14 to -202.80	-173.14 to -138.13	-779.93 to -657.80	-238.75 to -204.34	-17.63 to -14.53	-3464.09 to -2957.30	-80.37 to -65.88	-36.11 to -27.07	-84.66 to -68.87	-30.70 to -20.59	-77.58 to -65.49
%	**0.5**	**0.8**	**0.0**	**0.1**	**0.3**	**0.0**	**0.0**	**0.1**	**0.1**	**0.2**	**0.1**	**0.3**	**0.3**
Difference 2	-2.28 to -	-2.15 to -	-202.80 to -167.47	-138.13 to -103.11	-657.80 to -535.68	-204.34 to -169.93	-14.53 to -	-2957.30 to -2450.51	-65.88 to -	-27.07 to -	-68.87 to -	-20.59 to -	-65.49 to -

	PH H_2O	PH $CaCl_2$	Electrical Conductivity	OrganicCarbon	Carbonate	Phosphorus	Total Nitrogen	Extractable Potassium	CationExchange	Clay	Sand	Silt	Coarse Fragments
	1.52	1.39					11.43		51.39	18.03	53.08	10.48	53.40
%	**5.2**	**7.0**	**0.0**	**0.7**	**0.5**	**0.0**	**0.1**	**0.0**	**0.3**	**1.9**	**0.0**	**7.0**	**0.7**
Difference 3	-1.52 to -0.75	-1.39 to -0.62	-167.47 to -132.13	-103.11 to -68.10	-535.68 to -413.55	-169.93 to -135.52	-11.43 to -8.33	-2450.51 to -1943.71	-51.39 to -36.89	-18.03 to -8.99	-53.08 to -37.29	-10.48 to -0.37	-53.40 to -41.31
%	**15.6**	**19.2**	**0.0**	**1.5**	**0.3**	**0.1**	**0.6**	**0.0**	**1.3**	**12.6**	**0.8**	**44.2**	**1.1**
Difference 4	-0.75 to 0.01	-0.62 to 0.15	-132.13 to -96.79	-68.10 to -33.08	-413.55 to -291.42	-135.52 to -101.11	-8.33 to -5.24	-1943.71 to -1436.92	-36.89 to -22.40	-8.99 to 0.05	-37.29 to -21.50	-0.37 to 9.74	-41.31 to -29.23
%	**33.9**	**35.0**	**0.0**	**4.0**	**0.9**	**0.5**	**1.1**	**0.0**	**3.3**	**33.8**	**6.6**	**35.9**	**1.5**
Difference 5	0.01 to 0.78	0.15 to 0.92	-96.79 to -	-33.08 to 1.93	-291.42 to -	-101.11 to -	-5.24 to -	-1436.92 to -	-22.40 to -	0.05 to 9.09	-21.50 to -	9.74 to 19.85	-29.23 to -

	PH H_2O	PH $CaCl_2$	Electrical Conductivity	OrganicCarbon	Carbonate	Phosphorus	Total Nitrogen	Extractable Potassium	CationExchange	Clay	Sand	Silt	Coarse Fragments
			61.46		169.29	66.70	2.14	930.12	7.91		5.71		17.14
%	**26.6**	**22.1**	**0.2**	**40.3**	**1.1**	**0.8**	**5.3**	**0.1**	**14.2**	**37.2**	**19.8**	**10.5**	**1.8**
Difference 6	0.78 to 1.55	0.92 to 1.69	-61.46 to -26.12	1.93 to 36.95	-169.29 to -47.16	-66.70 to -32.29	-2.14 to 0.96	-930.12 to -423.33	-7.91 to 6.58	9.09 to 18.14	-5.71 to 10.08	19.85 to 29.96	-17.14 to -5.05
%	**14.5**	**12.6**	**2.3**	**48.3**	**7.0**	**3.1**	**67.5**	**1.0**	**51.3**	**12.6**	**55.3**	**2.0**	**5.8**
Difference 7	1.55 to 2.31	1.69 to 2.46	-26.12 to 9.22	36.95 to 71.96	-47.16 to 74.96	-32.29 to 2.11	0.96 to 4.06	-423.33 to 83.46	6.58 to 21.08	18.14 to 27.18	10.08 to 25.87	29.96 to 40.07	-5.05 to 7.04
%	**3.4**	**3.1**	**83.5**	**5.0**	**85.2**	**33.2**	**24.4**	**74.6**	**27.7**	**1.5**	**16.6**	**0.1**	**74.6**
Difference 8	2.31 to 3.08	2.46 to 3.23	9.22 to 44.55	71.96 to 106.97	74.96 to 197.09	2.11 to 36.52	4.06 to 7.15	83.46 to 590.26	21.08 to 35.57	27.18 to 36.22	25.87 to 41.66	40.07 to 50.18	7.04 to 19.13

	PH H_2O	PH $CaCl_2$	Electrical Conductivity	OrganicCarbon	Carbonate	Phosphorus	Total Nitrogen	Extractable Potassium	CationExchange	Clay	Sand	Silt	Coarse Fragments
%	**0.3**	**0.2**	**14.0**	**0.2**	**4.7**	**62.4**	**1.0**	**24.1**	**1.7**	**0.1**	**0.8**	**0.0**	**14.3**

Table S9 contains statistical comparisons for each predicted point **against the closest** measured (field-sampled) point**.** The analysis provides a clear insight into the predicted values by comparing them with the nearest reference point. In other words, it evaluates the prediction accuracy of GPR (Gaussian Process Regression) using the specified kernel and parameter settings in the script.

For example, Column 2 represents the pH in H_2O.

The first row contains the description.

The second row presents the percentage values.

The third row provides another description.

The fourth row displays numerical values, and so on.

It was observed that Difference 3, Difference 4, Difference 5, and Difference 6 collectively account for 90.6% of the points. This indicates that 90.6% of the predicted points have values within a difference range of -1.52 to 1.55. In other words, nearly 90% of the predictions have values very close to their nearest trained point, with the maximum deviation between them reaching 1.55. This statistical analysis validates the accuracy of the predicted values. Similarly, the same approach is applied to assess values for pH in $CaCl_2$, EC, and other parameters.

Map Continuity

Another key analysis is Map Continuity**.** This involves a visual inspection of the generated maps to ensure consistency. For example, maps typically end at national borders, whereas for neighboring European countries, LUCAS 2015 maps already exist. Therefore, continuity

is checked across national borders and NUTS boundaries. A comparison with existing European maps was conducted for each chemical element to ensure visual consistency (color coherence) **and** geographical continuity**,** except in areas explicitly excluded (urban areas, water bodies, transport networks). The maps (Figure S1 – S8) show a rather harmonized coherence between Western Balkans and the rest of Europe.

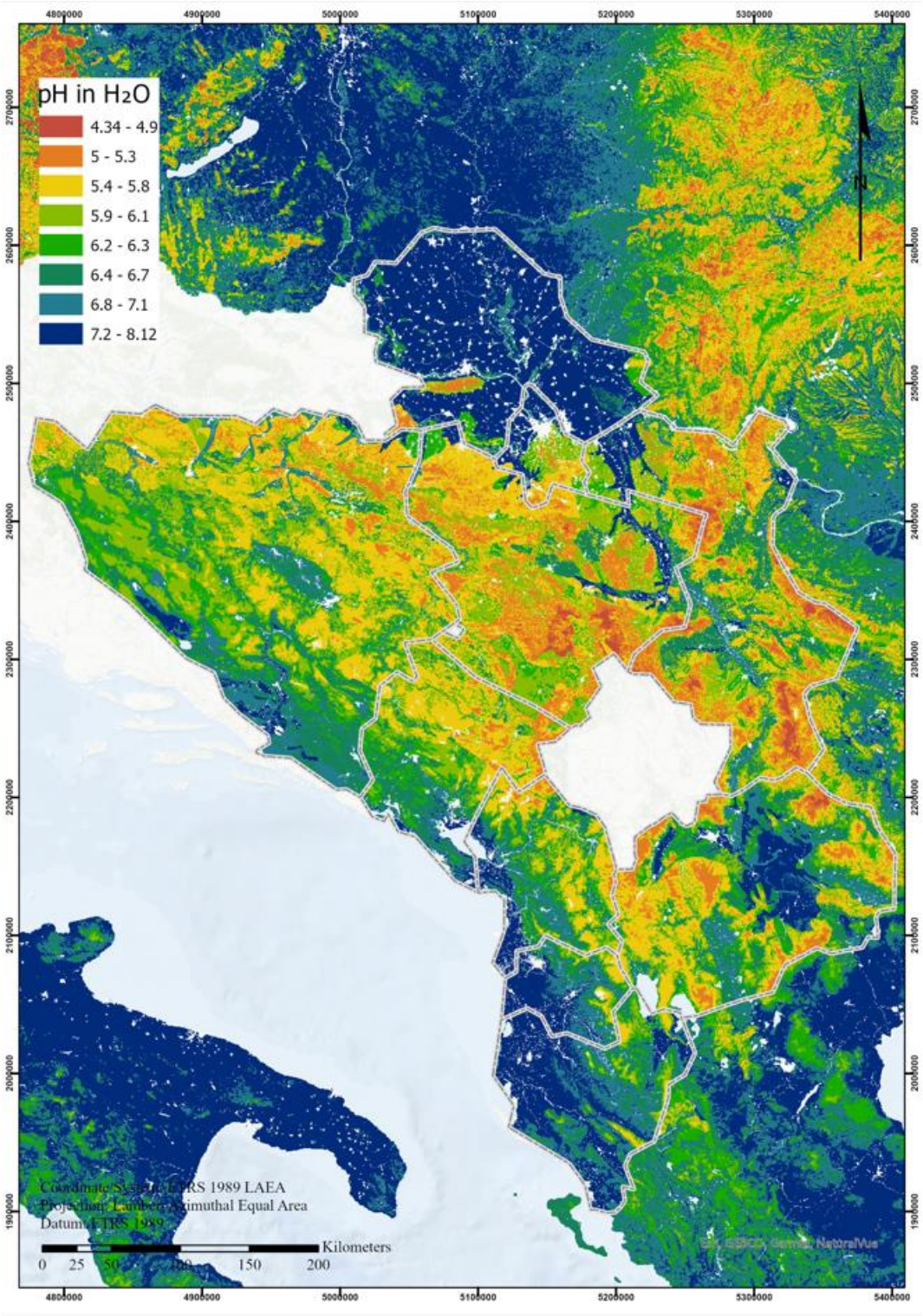

Figure S1. pH in H_2O map Continuity. The base map was designed and developed by Esri | Powered by Esri.

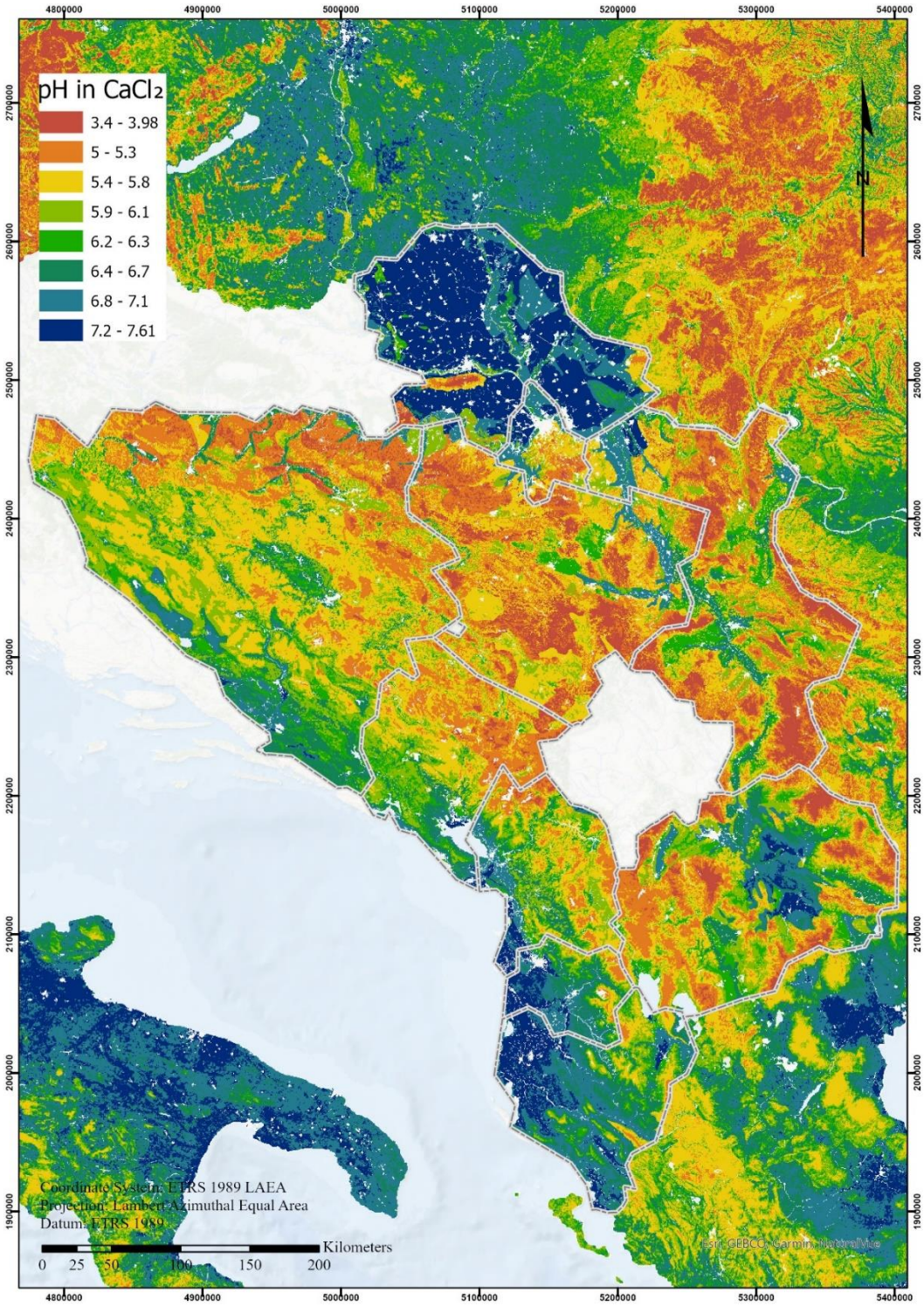

Figure S2. pH in $CaCl_2$ map Continuity. The base map was designed and developed by Esri | Powered by Esri.

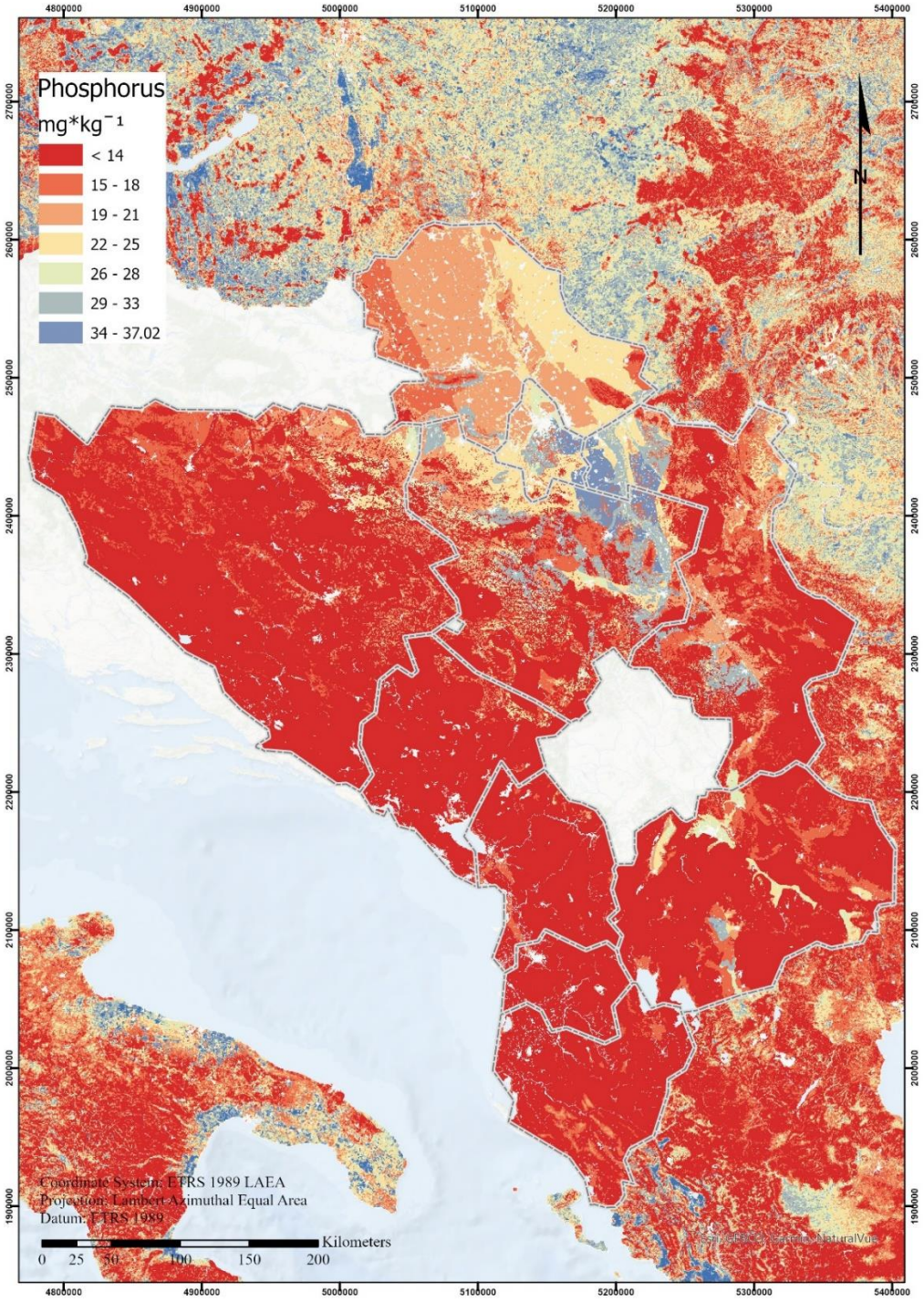

Figure S3. Phosphorus Map Continuity. The base map was designed and developed by Esri | Powered by Esri.

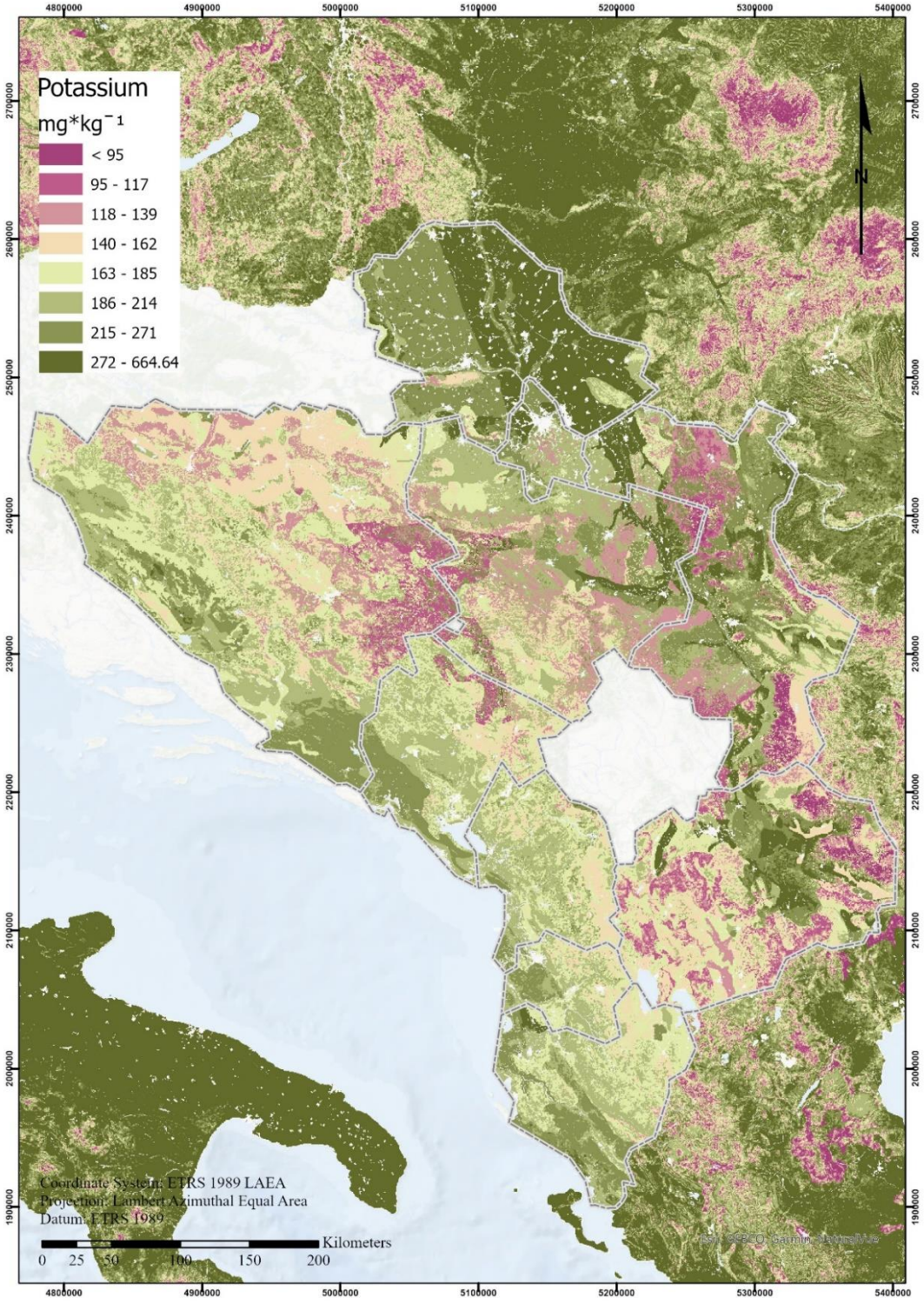

Figure S4. Potassium Map Continuity. The base map was designed and developed by Esri | Powered by Esri.

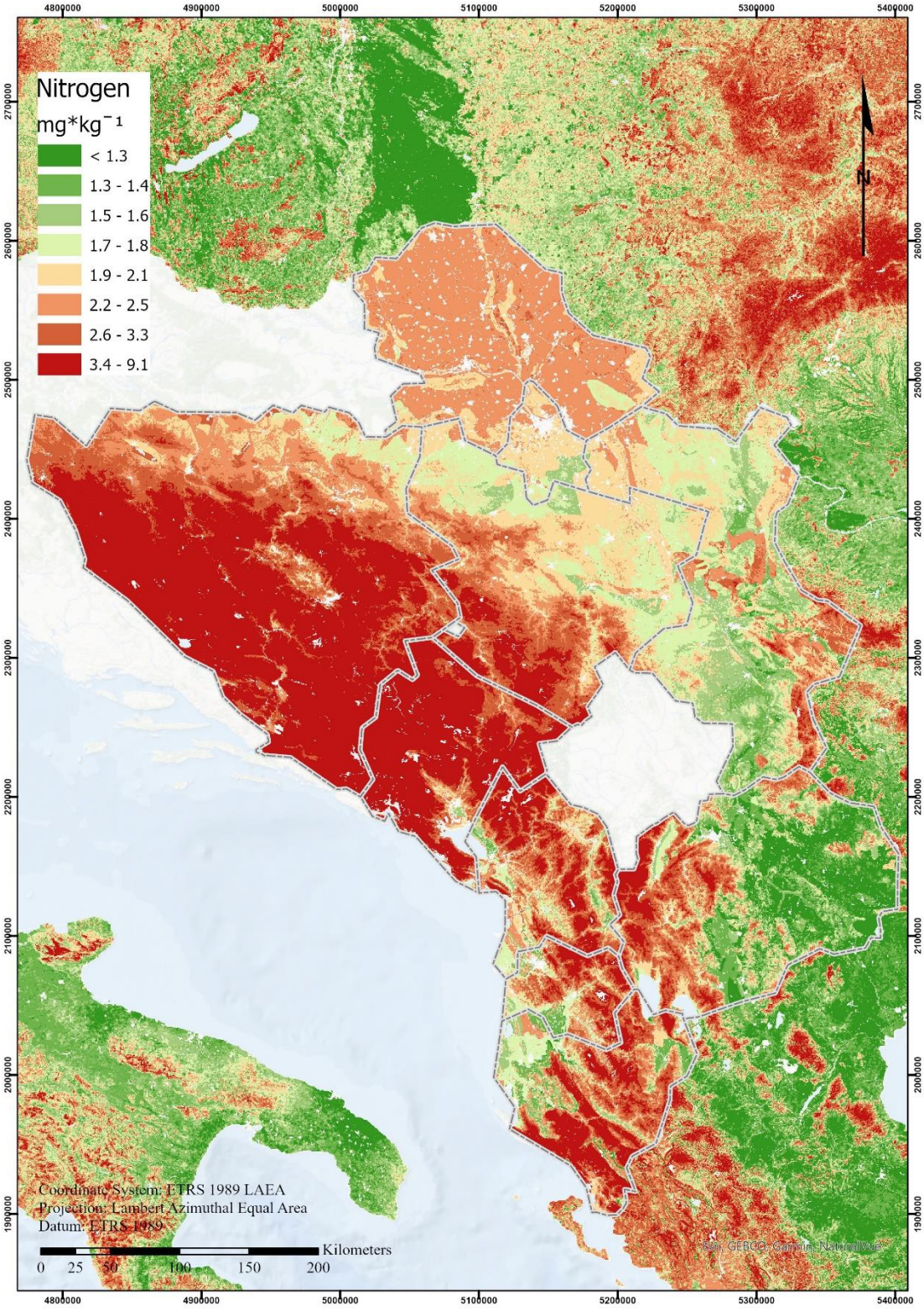

Figure S5. Nitrogen Map Continuity. The base map was designed and developed by Esri | Powered by Esri.

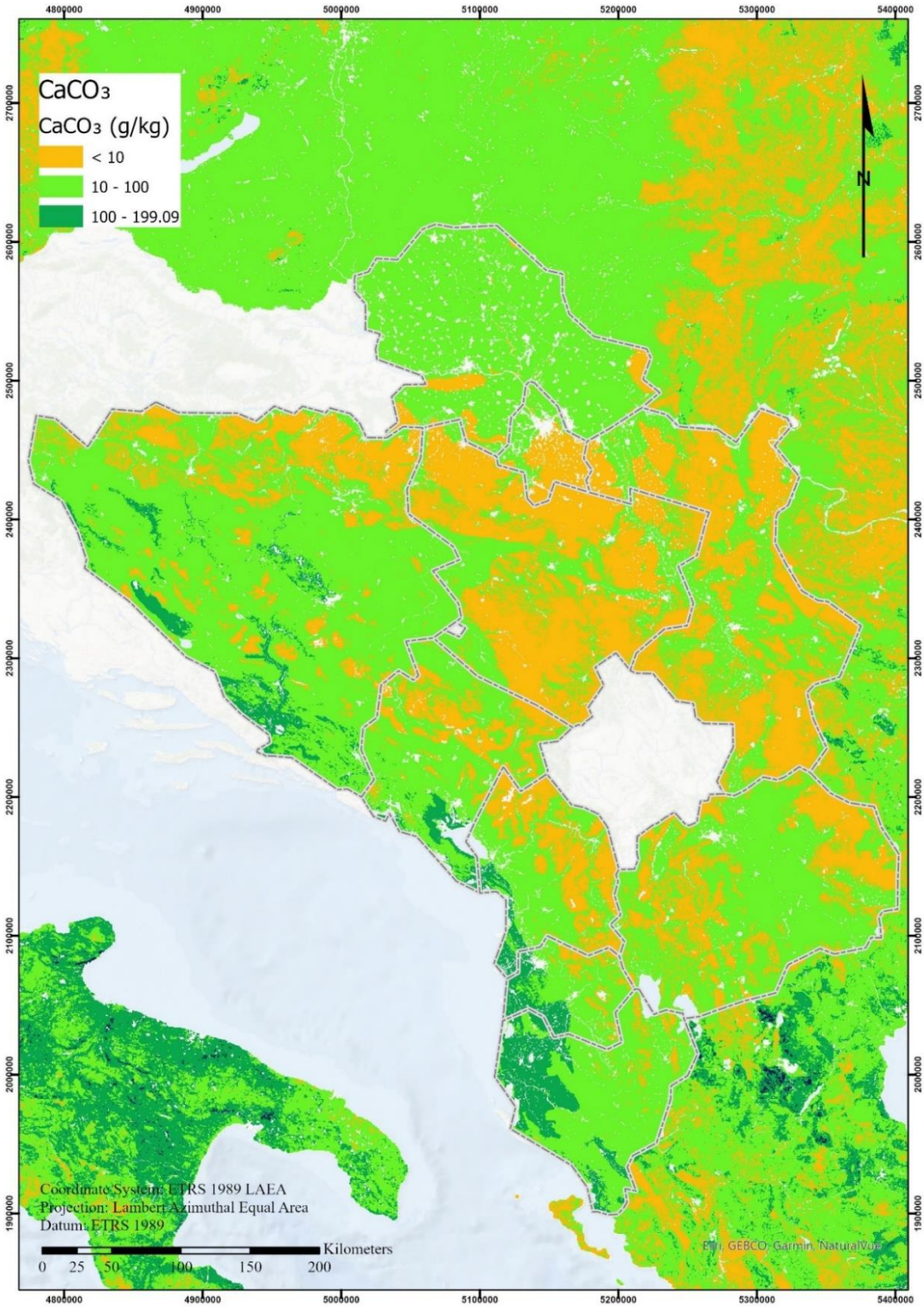

Figure S6. Carbonate Map Continuity. The base map was designed and developed by Esri | Powered by Esri.

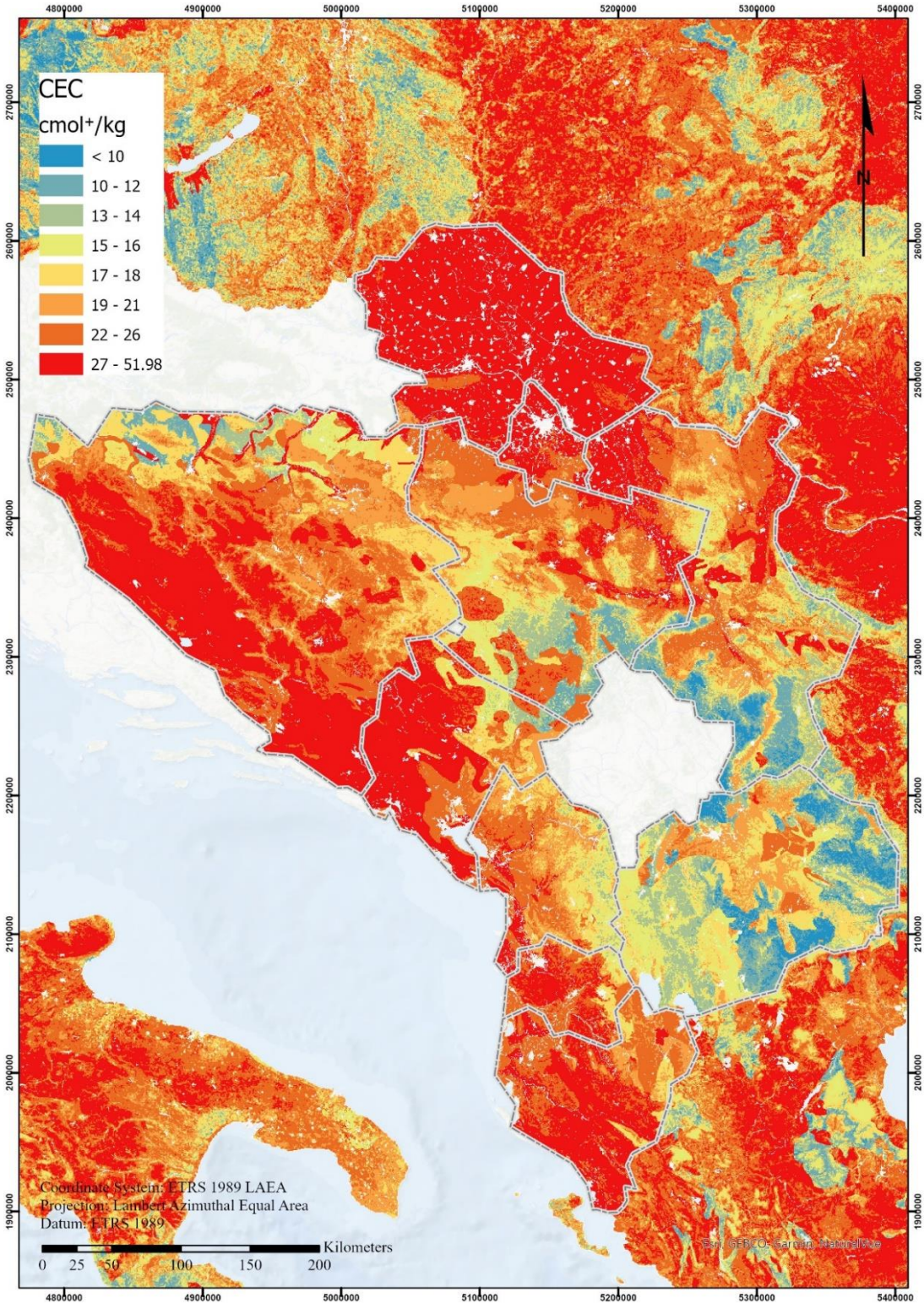

Figure S7. CEC Map Continuity The base map was designed and developed by Esri | Powered by Esri.

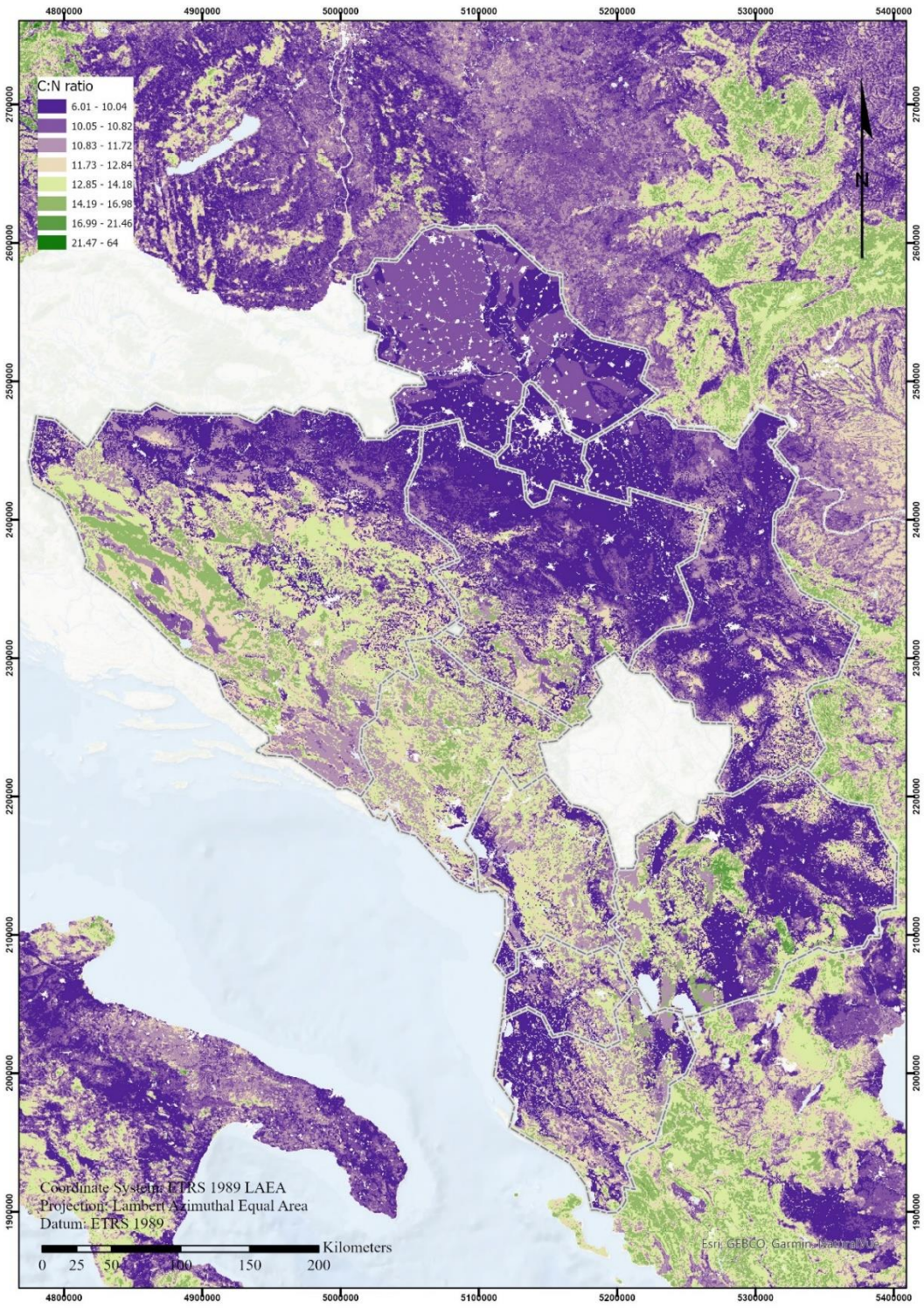

Figure S8. C: N Ration Map Continuity. The base map was designed and developed by Esri | Powered by Esri.

Effect of Missing Categories on Prediction Reliability in GPR

During prediction, some categories not found in the training data were encountered, likely due to the limited dataset of only 955 points. Unknown categories in column TrimWRB: 'CH', 'AB', 'HS'}. This indicates that there are no measured points (or, in GPR terminology, no trained points) for the WRB codes {'CH', 'AB', 'HS'}. Consequently, the prediction results for these categories are less reliable. The WRB values were sourced from the pedoclimate zones shapefile. GPR relies on trained data, the more attributes available in the training data, the better GPR performs, resulting in predictions closer to reality. The pedoclimate zones shapefile contains several attributes, but only complete ones can be used by GPR. Attributes with missing values can also be used if the missing values are replaced (e.g., with the median, minimum, maximum, standard deviation, etc.). In this case, we selected attributes with complete data, as training the model without knowing what values to assign for missing data isn't feasible.

Similarly:

> S01_WRB: {'HSeu', 'CHch', 'CHcc', 'HSdy', 'ABgl'} — these missing values indicate that GPR would perform more accurately if the training points included soil types according to the above WRB codes.
>
> S02_WRB: {'LVgl', 'CHcc'}
>
> pedoCLIM02: {3010.0, 8004.0, 3012.0, 8005.0, 7012.0, 8012.0, 8016.0, 8017.0, 6001.0, 6005.0, 5023.0}
>
> new_pedcli: {71.0, 81.0, 85.0, 56.0, 31.0}

Certainly, these attributes could be excluded from GPR training when using the Matérn kernel. However, doing so may lead to less accurate results.

Creating Maps and Raster Images

Based on the generated points and the predicted chemical properties, polygons were created. These generated points also include trained points, as they have field-measured chemical properties and, therefore, represent the data more accurately than the predicted values. The creation of polygons follows the Thiessen Polygon method. After generating the polygons, areas that are not part of the study such as regions outside the WB6 boundaries, water bodies, urban, industrial, and transport areas (roads, rail networks) are clipped. Each polygon created in this manner inherits the chemical properties of its respective point. Following this step, the next phase involved verifying the accuracy of the data generated. Once all the previous steps are completed, map creation becomes a matter of cartographic composition and operations. To generate the maps, the shapefiles are

converted into raster image format**,** specifically GeoTIFF**.** The resulting grid values store each chemical analysis. Each GeoTIFF file has a spatial resolution of 200m x 200m**,** ensuring the required level of accuracy.

Python Script for property prediction and data QA/QC assurance.

We have developed five Python scripts utilising machine learning techniques to predict top soil properties. The following list presents the scripts along with a brief description of each.

- **Script to create Predicted points.py**
- **Gaussian_process_soil_prediction_Matern_Kernel.py**
- **Gaussian_process_soil_prediction-RBF_plus_White_kernels.py**
- **Check Predicted-vs-Measures_Chemical-Properties.py**
- **Make-Stats-for-the-predicted-chemical-properties.py**

1-Script to create Predicted points.py

This Python script processes geospatial data to generate a grid of points within defined polygon areas, integrating various environmental attributes. The script is well documented and includes comments on each line of the code. The key steps are as follows:

1. **Load Geospatial Data**: The script reads points, polygons, climatic, soil, and administrative data from shapefiles and geodatabases.
2. **Ensure CRS Consistency**: All datasets are converted to a common Coordinate Reference System (CRS) using EPSG:3035.
3. **Create Grid System**: A grid of squares is generated across the study area, based on predefined grid size and step values.
4. **Clip Polygons**: The grid squares are intersected with land cover polygons to identify the relevant areas for further analysis.
5. **Generate Points**: A fine-resolution grid of points is created within the clipped polygon areas, with a spacing of every 500 meters.
6. **Assign Attributes**:
    - Land cover and administrative region data are assigned through spatial joins.
    - Elevation values are extracted from a Digital Elevation Model (DEM) raster.

    - Climatic and soil classification attributes are assigned from external datasets.
7. **Filter Duplicates**: Any points already existing in the original dataset are excluded from the output.
8. **Save Output**: The generated points are stored in both CSV and ESRI Shapefile formats for further use.
9. **Performance Logging**: The script tracks the execution time and the total number of created points, providing performance insights.
10. **Beep Alerts**: An audible alert is played at the start and end of the execution process to indicate progress.

2-Gaussian_process_soil_prediction_Matern_Kernel.py

This script leverages **Gaussian Process Regression (GPR)** to predict soil properties at unsampled locations. The script is well documented and includes comments on each line of the code. It employs **cross-validation** to assess model performance and optimize hyperparameters, ensuring accurate and reliable predictions.

Key Features:

- Allows users to specify which **soil properties** to predict.
- Enables customization of the **number of neighbors** considered during predictions.
- Utilizes **GPy** for Gaussian Process Regression, **scikit-learn** for preprocessing, and **pandas** for data manipulation.
- Trains the model using the **LUCAS 2015 soil database**.

Prediction Variables:

✓ Soil Properties:

- pH in H_2O
- pH in $CaCl_2$
- Electrical Conductivity
- Organic Carbon
- Carbonate

- Phosphorus
- Total Nitrogen
- Extractable Potassium
- Cation Exchange Capacity
- Clay, Sand, Silt
- Coarse Fragments

✓ Categorical Variables:

- Climatic Zone
- Land Cover
- Soil ID
- Soil Texture (General, Type 1, Type 2)
- Pedological Classification

✓ Numeric Variables:

- X, Y, Z Coordinates

Kernel Configuration:

- **Main Kernel: Matérn 5/2**
- **Periodic Kernel: Periodic Matérn 5/2**

Hyperparameters:

- **Variance = 1.0** → Controls the overall signal strength.
- **Lengthscale = 8.0** → Defines smoothness (higher values = smoother predictions).
- **White Kernel Variance = 0.05** → Adds a small noise component for better generalization.

The kernel parameters are **optimized during training** to maximize model fit and enhance predictive accuracy.

3-gaussian_process_soil_prediction-RBF_plus_White_kernels.py

This script employs **Gaussian Process Regression (GPR)** to predict soil properties at unsampled locations. The script is well documented and includes comments on each line of the code. It is a version of script 2: 2-Gaussian_process_soil_prediction_Matern_Kernel.py, but it is set up to use different kernels and parameters. It leverages **cross-validation** to assess model performance and optimize hyperparameters for improved accuracy.

Key Features:

- Allows users to specify which **soil properties** to predict.
- Enables customization of the **number of neighbors** considered during predictions.
- Utilizes **GPy** for Gaussian Process Regression, **scikit-learn** for preprocessing, and **pandas** for data manipulation.
- Trains the model using the **LUCAS 2015 soil database**.

Predictable Variables

Soil Properties

- **pH** (H_2O, $CaCl_2$)
- **Electrical Conductivity**
- **Organic Carbon**
- **Carbonate**
- **Phosphorus**
- **Total Nitrogen**
- **Extractable Potassium**
- **Cation Exchange Capacity**
- **Clay, Sand, Silt, Coarse Fragments**

Categorical Variables

- Climatic Zone
- Land Cover
- Soil ID
- Soil Texture (1, 2)

- Pedological Classification

Numeric Variables

- **X, Y, Z Coordinates**

Gaussian Process Regression (GPR) Configuration

The script applies a combination of the **Radial Basis Function (RBF) Kernel** and **White Kernel** to model spatial dependencies and account for noise.

Kernel Parameters

- **Variance** = **1.0** → Controls overall signal strength.
- **Lengthscale** = **8.0** → Determines smoothness (larger values yield smoother predictions).
- **White Kernel Variance** = **0.05** → Introduces noise to prevent overfitting.

During training, these parameters are optimized for better model performance. Initially, the script sets:

- Variance = **1.0**
- Lengthscale = **1.0**
- White Kernel Variance = **0.5**

These values are automatically refined through training to enhance the model's fit to the data.

4-Check Predicted-vs-Measures_Chemical-Properties.py

This script performs a spatial comparison and statistical analysis between actual and predicted soil property values using **Gaussian Process Regression (GPR)**. The script is well documented and includes comments on each line of the code

Key Steps in the Script

1. Load Data

- Reads actual soil data from **pikat_csv_path** (CSV).
- Reads GPR-predicted soil values from **gpr_csv_path** (CSV).

- Converts both datasets into **GeoDataFrames** with an **EPSG:3035** projection for spatial analysis.

2. Nearest Neighbor Matching

- Uses a **KDTree** for efficient spatial matching.
- Finds the nearest actual soil data point for each GPR-predicted point.
- Merges the actual and predicted datasets based on spatial proximity.

3. Calculate Differences

- Computes the difference between actual and predicted values for multiple soil parameters, including:
    - **pH** (H_2O, $CaCl_2$)
    - **Electrical Conductivity**
    - **Organic Carbon**
    - **Carbonate**
    - **Phosphorus**
    - **Total Nitrogen**
    - **Extractable Potassium**
    - **Cation Exchange Capacity**
    - **Clay, Sand, Silt, Coarse Fragments**
- Stores these differences in new columns for further analysis.

4. Classify Differences into 8 Ranges

- Determines **minimum, maximum, mean, and standard deviation** of the differences for each soil parameter.
- Categorizes the differences into **8 statistical classes**.
- Counts the number of data points in each class.

5. Save Results

- Saves the processed dataset (**joined_df**) to **output_csv_path**.
- Saves summary statistics (**stats_df**) to **stats_output_csv_path**.

Purpose of the Script

This script evaluates the **accuracy of GPR predictions** by comparing them to real soil data, classifying the differences, and summarizing key statistical trends. The results provide insights into the reliability and performance of GPR in soil property estimation.

5-Make-Stats-for-the-predicted-chemical-properties.py

This script Assess the accuracy and reliability of **Gaussian Process Regression** predictions by comparing the predicted results created with Different GPR Kernels and Parameters. It identifies discrepancies between predicted and observed values, offering insights into how well the GPR model captures the data distribution.

Key Steps in the Script

1. Load Datasets

- Reads actual soil property data from **pikat_csv_path**.
- Reads GPR-predicted soil values from **gpr_csv_path**.

2. Define Soil Properties for Analysis

- Selects **13 key soil parameters**, such as **pH, Organic Carbon, Clay, Sand, and Silt**.

3. Statistical Comparisons

- Determines **minimum and maximum values** for both actual and predicted datasets.
- Computes **prediction error bounds** using associated **sigma (uncertainty) values**.
- Counts actual values that **exceed the predicted maximum** or **fall below the predicted minimum**.

4. Percentage Analysis & Interpretation

- Calculates the **percentage of actual values** outside the predicted range.
- Assesses **how much of the data range is underrepresented** in GPR predictions.
- Provides interpretative insights, identifying potential **prediction failures** or **biases due to data sparsity**.

5. Results Export

- Saves the statistical summary to **output_csv_path**.
- Displays a confirmation message upon completion.

Purpose of the Script

This script evaluates the **effectiveness of GPR predictions** by analyzing their alignment with real-world soil measurements. It helps identify **model limitations** and highlights regions where GPR predictions may be **unreliable due to insufficient training data**.