Machine Learning Models for Soil Parameter Prediction Based on Satellite, Weather, Clay and Yield Data

Calvin Kammerlander, Viola Kolb, Marinus Luegmair[1], Lou Scheermann, Maximilian Schmailzl,
Marco Seufert, Jiayun Zhang, Denis Dalić[2], Torsten Schön[3]

[1] orcid: 0000-0001-9022-8428
[2] MI4People, Munich, Germany
[3] Technische Hochschule Ingolstadt, Ingolstadt, Germany, orcid: 0000-0001-5763-3392

March 31, 2025

Abstract—**Efficient nutrient management and precise fertilization are essential for advancing modern agriculture, particularly in regions striving to optimize crop yields sustainably. The AgroLens project endeavors to address this challenge by developing Machine Learning (ML)-based methodologies to predict soil nutrient levels without reliance on laboratory tests. By leveraging state of the art techniques, the project lays a foundation for actionable insights to improve agricultural productivity in resource-constrained areas, such as Africa. The approach begins with the development of a robust European model using the LUCAS Soil dataset and Sentinel-2 satellite imagery to estimate key soil properties, including phosphorus, potassium, nitrogen, and pH levels. This model is then enhanced by integrating supplementary features, such as weather data, harvest rates, and Clay AI-generated embeddings. This report details the methodological framework, data preprocessing strategies, and ML pipelines employed in this project. Advanced algorithms, including Random Forests, Extreme Gradient Boosting (XGBoost), and Fully Connected Neural Networks (FCNN), were implemented and fine-tuned for precise nutrient prediction. Results showcase robust model performance, with root mean square error values meeting stringent accuracy thresholds. By establishing a reproducible and scalable pipeline for soil nutrient prediction, this research paves the way for transformative agricultural applications, including precision fertilization and improved resource allocation in under-resourced regions like Africa.**

Index Terms—**Machine Learning, Soil Prediction, Satellite Data, Weather Data, Clay Model, Yield Data, Fertilization Recommendation**

I. INTRODUCTION

Efficient and sustainable agricultural practices are pivotal in addressing global challenges such as food security, resource scarcity, and environmental degradation. With the increasing demand for higher agricultural productivity, precise fertilization and effective nutrient management have become critical components in modern farming systems [1]. ML has emerged as a transformative technology in agriculture, providing robust decision-support tools to optimize resource utilization. ML-driven solutions enable the analysis of complex datasets, uncovering insights that traditional methods often overlook. As highlighted by [2], nutrient management plays a central role in ensuring optimal crop development, requiring precise knowledge of key soil nutrients such as nitrogen, phosphorus, and potassium, and others.

A. Motivation

Traditional soil testing methods are expensive and inaccessible to many farmers, particularly in under-resourced regions like Africa. By harnessing readily available satellite imagery and advanced ML algorithms, the AgroLens project seeks to democratize soil nutrient estimation, making precision agriculture more accessible and impactful.

B. Objectives

The overarching goal of the AgroLens project is to develop a scalable and accurate methodology for predicting soil nutrients using non-invasive data sources. This involves:

- Constructing ML models capable of estimating the key soil property scores for pH level, Phosphorus, Nitrogen, and Potassium.
- Enhancing model performance through the integration of diverse data sources, including local weather data and harvest rates.

C. Reasons for the Use of Machine Learning

ML techniques are well suited to agricultural applications due to their ability to handle large, heterogeneous datasets and uncover complex, non-linear relationships between variables. For instance, ML models can integrate satellite-derived spectral indices, climate variables, and other geospatial data to produce reliable predictions of soil nutrient levels. This capability makes ML a powerful tool for addressing the variability and unpredictability inherent in agricultural systems.

D. Overview of the Approach

The AgroLens project employs a structured two-phase methodology:

1) **Model Development**: Leveraging European datasets (e.g., LUCAS Soil and Sentinel-2 imagery) to create a baseline model for soil nutrient estimation.
2) **Feature Enrichment**: Incorporating supplementary data sources such as weather data, harvest rates, and Clay embeddings to improve model accuracy.

By following this approach, AgroLens aims to establish a reproducible and scalable framework for soil nutrient prediction, laying the groundwork for future work in precision agriculture.

II. METHODOLOGY

In this study, ML is used to predict soil properties using different types of data. The input data includes satellite images, weather data, harvest rates and Clay embeddings, while nutrient scores form the target data. The training pipeline includes data pre-processing, feature selection and model evaluation using Root Mean Square Error (RMSE).

A. Machine Learning for Soil Prediction

ML is widely applied across numerous domains, including agriculture, where it has shown particular promise for soil prediction—especially in African contexts [3], [4]. Various ML techniques, such as Random Forest, XGBoost, FCNN, Convolutional Neural Networks, and Support Vector Machines, have been explored for nutrient management in soils. Typical input data include soil properties, climatic variables, and satellite imagery, enabling accurate estimations of nutrient levels [2].

B. Data Types

This section provides an overview of the data types used in the project, beginning with the input datasets and concluding with the target variables.

1) Input Data:

a) Satellite Data: Satellite imagery serves as a primary source of spatial information in AgroLens, capturing reflectance values across multiple spectral bands. By analyzing these reflectance patterns, one can infer soil conditions, vegetation health, and other environmental variables.

The Sentinel-2 mission was launched in 2015 and includes satellites that revisit a given location every five days. It offers 13 spectral bands with resolutions ranging from 10 to 60 m, covering portions of the visible, near-infrared, and shortwave infrared spectrum. This project uses Sentinel-2 Level 2A images, which undergo atmospheric correction to reduce distortions and provide more accurate surface reflectance values. Using Level 2A data reduces the number of available bands to 12 [5]. Sentinel-2 is chosen for its high revisit frequency and data quality, improving the chances of obtaining cloud-free images aligned with the soil-sample collection dates.

Landsat 7 and Landsat 8 were launched in 1999 and 2013, respectively, each following a 16-day repeat cycle. This longer interval limits the frequency of observations compared to Sentinel-2, making time-sensitive analyses more challenging. Landsat 7 provides eight spectral bands with spatial resolutions between 15 and 60 m, whereas Landsat 8 adds three more bands (for a total of 11). However, the two thermal infrared bands of Landsat 8 are excluded here to match the Sentinel-2 band set, leaving nine bands with 15–30 meter resolutions. Both Landsat 7 and Landsat 8 also offer Level 2 atmospheric-corrected products. [6], [7]

b) Neighboring Pixels: In addition to the central pixel patch, neighboring pixels around each sampled location are considered to capture more spatial context. The premise is that pixel-to-pixel variations in reflectance might reveal subtle gradients related to soil properties, vegetation cover, or micro-topography.

c) BigEarthNet v2.0: BigEarthNet v2.0 is a benchmark dataset derived from Sentinel-1 and Sentinel-2 satellite imagery [8], published in conjunction with the work of Clasen et al. [9]. It is designed for Earth observation, focusing on land-cover classification and environmental monitoring. The dataset contains annotated image pairs from several European countries, encompassing diverse land-use and land-cover classes (e.g., agricultural fields, forests, urban areas, water bodies) [9]. Additionally, the BigEarthNet initiative provides pre-trained models [10], which can be used to classify land-use and land-cover types in regions beyond those originally included in BigEarthNet v2.0. However, this paper concentrates on predicting soil nutrients rather than classifying land use. Because the training data exclusively represent agricultural areas, land-use classification was deemed unnecessary. Nonetheless, incorporating such classification could be a useful extension in future enhancements of this system, ensuring that predictions are only generated for valid agricultural zones.

d) Weather Data: Aligning weather data to the same temporal window as satellite imagery ensures that the satellite reflectance can be interpreted in the context of concurrent weather conditions.

e) Harvest Rates: The Food and Agriculture Organization (FAO) publishes comprehensive yield statistics that can be leveraged to approximate crop productivity and, indirectly, soil fertility indicators across different regions. These datasets generally include average yield estimates for major crops (e.g., wheat, maize, rice) at subnational or national levels. Harvest rates can serve as a proxy for soil fertility when more direct soil measurements are unavailable or incomplete. Highly productive regions typically correlate with better soil conditions and nutrient availability. Additionally, integrating harvest rates can help contextualize local satellite pixel values, as known crop outputs may offer insights into whether observed reflectance patterns align with high- or low-fertility conditions.

f) Data Inspection: Several measures are taken during data preprocessing to ensure input quality and consistency. Rather than discarding rows with missing values, only potassium measurements that fall below the limit of detection are imputed (see Section III-A2a). Potential outliers are identified by generating histograms for each feature. Although a small fraction (less than 0.1% of the total dataset) lies noticeably outside the main distribution, these points are retained under the

assumption that a robust model can handle such rare extremes. Lastly, the Pearson product-moment correlation coefficient is computed,

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}},$$

where C is the covariance matrix. Correlation values approaching 1.0 suggest strong linear relationships, while those near 0.0 indicate minimal or no linear association. This analysis assists in detecting redundant features and gauging overall data quality.

2) Target Data: As part of the AgroLens project, the primary soil properties identified for initial prediction are pH levels, phosphorus, potassium, and nitrogen. An evaluation of soil data sources is carried out following [11], which mainly relies on AfSIS, and is extended to include other soil data sources beyond Africa, such as LUCAS [12] and WoSIS [13]. The resulting findings, along with assessments of their suitability for mapping against satellite data, appear in Table I. A key requirement for mapping soil data to satellite imagery is the availability of timestamps for the soil sampling event that are sufficiently accurate (within approximately one month), as well as overlap with the relevant satellite-data coverage. Although this threshold may seem self-evident, it poses a significant challenge in the case of many African soil samples.

TABLE I
SOIL DATA SETS WITH TIMESTAMP ATTRIBUTE AND LANDSAT 8 AND SENTINEL-2 COVERAGE

Data set	Time-frame	Time-stamp	Profiles	Landsat 7/8 ($>$ 03/2008)	Sentinel-2 ($>$ 06/2015)
LUCAS [12]	2018	Yes	18,984	18,471	18,471
AfSIS [14]	2009-2018	No	20,704	0	0
WoSIS [13]	1920-2023	Partially	228,000	3,641	0

C. Machine Learning Models

The prediction of soil nutrient levels using satellite data combined with additional information, such as weather data, represents a typical regression problem. Consequently, three model variants are selected for each nutrient type to utilize the specific strengths of each model in predicting the corresponding soil nutrient level and to perform a comparative performance analysis.

1) XGBoost: XGBoost is a tree boosting system, published by [15]. XGBoost builds trees sequentially, with each new tree aiming to correct the errors of the previous tree. The final prediction is obtained by combining all trees. The method is known for its high accuracy and computational efficiency. Furthermore. it is robust to large and noisy datasets. The built-in regularization techniques help mitigate overfitting.

2) Fully Connected Neuronal Network: A FCNN is a deep learning architecture designed to model non-linear interactions among input features [16]. By fully connecting every neuron in each layer to all neurons in adjacent layers, FCNNs can capture a wide range of multivariate relationships between soil properties and environmental factors, making them particularly well-suited for tasks requiring complex pattern discovery.

3) Random Forest: Random Forest is an ensemble method comprising multiple decision trees, initially introduced by [17]. It is recognized for its robustness and interpretability, offering reliable predictions while maintaining relatively low computational demands compared to more complex models. This approach is especially suitable for scenarios with limited training data.

D. Training Pipeline

The machine learning models are run on a server equipped with 8 CPU cores, 32 GB of RAM, and an NVIDIA RTX 3060 GPU with 6 GB of VRAM. Remote collaboration is facilitated through isolated Docker containers that provide GPU access, ensuring an efficient and organized workflow. Preprocessed datasets and resulting models are stored and versioned on the same server, while all code is published to a Git repository for collaboration and version control (https://github.com/cvims/AgroLens).

1) Data Preprocessing: In this section the mandatory data preprocessing is described.

a) Data Collection: The training, test, and validation data for this project is collected from multiple sources. LU-CAS 2018 TOPSOIL data is obtained by submitting an official request form to the European Soil Data Centre (ESDAC), which then provides the dataset as structured CSV tables [12]. This data is further preprocessed through Python scripts. Satellite imagery from Sentinel-2 and Landsat 8 is acquired via the Copernicus Data Space API through automated requests spanning several days. Additionally, weather and climate information is sourced from the OpenWeatherMap API, which requires a paid subscription for full access to its datasets. These diverse data sources are carefully integrated to support the project objectives. Weather data is retrieved using the OpenWeather API's Time Machine endpoint, which delivers historical weather information for specified geographic locations and dates. For each row in the dataset, latitude, longitude, and the date are extracted and used to construct API queries. A redundancy check ensures that API calls are only made for rows lacking weather data, thus minimizing costs. The returned data is parsed to extract key metrics such as temperature, humidity, wind speed, and sunrise/sunset times, which are then appended to the dataset. For details, see www.openweathermap.org/. Harvest rates are downloaded from the FAO GAEZ portal by selecting the "Theme 5: Actual Yields and Production" dataset and obtaining the relevant GeoTIFF files. These georeferenced files are then opened in a GIS environment to extract pixel values for each set of

latitude and longitude coordinates. In this manner, yield scores become available for spatially explicit analysis of agricultural productivity. Refer to https://gaez.fao.org/ for more information. Finally, WoSIS 2023 snapshot data is openly available as a zipped dataset from the web, offering structured CSV and TSV tables that are further processed through Python scripts [18].

b) Normalization: To ensure consistency and improve the performance of ML models, data normalization is implemented as part of the preprocessing pipeline. The AgroLens project adopts a min-max normalization strategy, which scales the input features to a range of [0, 1]. This method preserves the relationships between values while standardizing the dataset for computational efficiency. The normalization process is designed to retain the integrity of the original data by maintaining separate tables for raw and normalized values:

- **Original Data Table**: Contains the unmodified, raw data as collected from various sources.
- **Normalized Data Table**: A transformed version of the original data, where each feature value x is scaled using the formula:

$$x_{\text{normalized}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

  Here, $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the feature, respectively.

The target features, which represent soil nutrient levels, are not normalized as the goal is to predict these values on their original scale, ensuring meaningful and interpretable outputs. This approach ensures that the original dataset remains untouched for verification and comparison purposes, while the normalized dataset is utilized for training and validating ML models. The separation also provides flexibility for further analysis and troubleshooting during model development and evaluation. By applying this normalization technique to the input features, the project mitigates the risk of bias introduced by features with different scales or units, thereby enhancing the stability and accuracy of the predictive models.

c) Spatial Cross-Validation: Spatial Cross-Validation (CV) is an approach for evaluating model performance in projects involving spatial data. It is used to take spatial autocorrelation of the dataset into account and mitigates overestimation of model accuracy [19].
Traditional validation methods, which randomly split data into sets for training and testing, often result in spatially proximate samples appearing in both sets. This can result in overly optimistic performance metrics, as similar observations in both sets can artificially inflate model accuracy [19]. Therefore, it is recommended to account for spatial dependence when validating a model using spatial data. There are various approaches for spatial CV. One such method is Grid-based spatial CV. In this approach the dataset is divided into separate, spatial grid cells. The grids can be divided into training, validation and test grids [20]. The choice of the grid cell size is crucial. Smaller grids may result in test datasets sharing similar characteristics with the training datasets, while larger blocks increase the risk that test data is not spatially similar,

which could lead to better validation scores for the model [19].

2) Used ML Tools: Open-source software is employed throughout this research, except where commercial products are explicitly mentioned. Most computations are performed in Python [21], a widely recognized and state-of-the-art language for ML. The following Python libraries are also utilized:

- **numpy**—"The fundamental package for scientific computing with Python" [22], [23].
- **scipy**—Offers advanced mathematical functions for scientific computing [24], [25].
- **optuna**—Facilitates efficient hyperparameter tuning, supporting methods like grid search, random sampling, genetic algorithms, Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [26], Gaussian Processes, and Tree-structured Parzen Estimator (TPE) [27]. A dashboard feature provides real-time visualization of optimization progress and parameter sensitivity [28]–[30].
- **pandas**—Handles data-frame preparation and manipulation [31]–[33].
- **pytorch**—Provides a flexible, high-performance framework for deep learning tasks, used here for the FCNN [34].
- **scikit-learn**—A widely adopted library offering an assortment of common machine learning algorithms [35], [36].
- **xgboost**—Implements a distributed gradient boosting framework for efficient model training [15], [37].
- **Additional Tools**—Includes `boto3`, `gdal`, `geopandas`, `joblib`, `jupyter`, `OpenCV`, `rasterio`, `requests`, and `shapely`, among others, for specific data processing needs.

3) Visualization Tools: Visualization tasks are handled by the following Python libraries:

- **matplotlib**—Provides fundamental plotting capabilities for charts, graphs, and figures [38].
- **seaborn**—Extends matplotlib with advanced statistical visualizations, such as heat maps and correlation matrices [39], [40].

III. MODEL FOR EUROPE

This section presents a predictive modeling approach that uses satellite images and field-collected soil measurements to estimate key soil parameters. The model integrates Sentinel-2 satellite imagery as input data and soil property measurements from the LUCAS 2018 TOPSOIL dataset as target data. The concept of the model for Europe is visualized in figure 1.
To ensure robust predictions, three ML models, including XGBoost, FCNN and Random Forest, are implemented and evaluated. Furthermore, different data splitting techniques, such as single split and spatial cross-validation, are explored to assess the impact of spatial dependencies on model performance. The following sections detail the data preprocessing, model selection, hyperparameter optimization, and performance analysis.
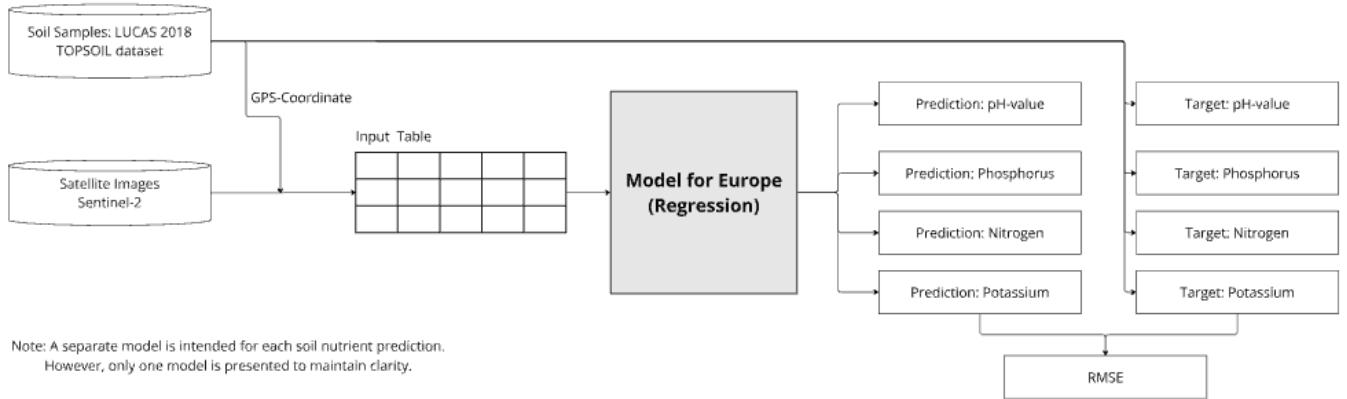
Fig. 1. Schematic Concept of the Training Process for Model and Data Usage for the Europe Model

A. Data Set

The Model for Europe requires input data and target data as detailed in the following section. Additionally, strategies for splitting the dataset are provided.

1) Input Data:

a) Sentinel-2: Sentinel-2 image data for each soil sample is obtained through multiple steps. First, all available satellite datasets matching each soil sample's GPS coordinates within a 29-day window (14 days before and 14 days after the sample date) are identified by the Copernicus API. Next, only the cloud probability masks for these datasets are downloaded to assess the pixel-level cloud probability within a small radius of the target coordinate. Once the nearest cloud-free dataset is identified, the corresponding 100km × 100km tile is downloaded as a ZIP file, then extracted and placed into a fixed directory structure. Each single-band grayscale image is cropped to 101 × 101 pixels, ensuring that the center pixel aligns precisely with the soil sample's location. Finally, the dataset folder is stored in the input data directory on the server, systematically organized by date and by GPS latitude and longitude. This procedure is repeated for every LUCAS SOIL dataset sample, resulting in the download and processing of approximately 20TB of raw image data and thus requiring optimization. Three distinct Python libraries are benchmarked on Sentinel-2 datasets to determine the most efficient method for cropping the images. In this case, OpenCV outperforms both Pillow and scikit-image, completing the task nearly twice as quickly. To reduce resource bottlenecks, a multi-threaded approach is implemented, with each thread dedicated to a specific task (e.g., image cropping, cloud detection, API calls, or data downloads). Temporary files are stored on RAM-disks to improve performance and minimize wear on the servers' SSDs. With the combined resources of three available servers, the process takes multiple days to complete and ultimately produces approximately 120GB of processed Sentinel-2 images. Afterwards, depending on the specific project model, the required pixels are extracted and saved into a data table using a separate command line script.

b) Correlation of the Input Data: In Figure 2, the correlation coefficient matrix for the input data is presented. A pronounced linear correlation is observed among Sentinel-2 bands 1–5, 6–9, and 11–12. This is likely due to all bands scanning the same spatial location at different wavelengths. Despite these high correlations, none of the Sentinel-2 bands are removed to preserve the maximum amount of information, given the relatively small size of the training dataset. Additionally, a strong correlation is evident between the POINTID and the longitudinal position of the measurement point, presumably because the longitudinal value was used to index the dataset. Since Index and POINTID are not used as training variables, this correlation does not influence the model's training process.
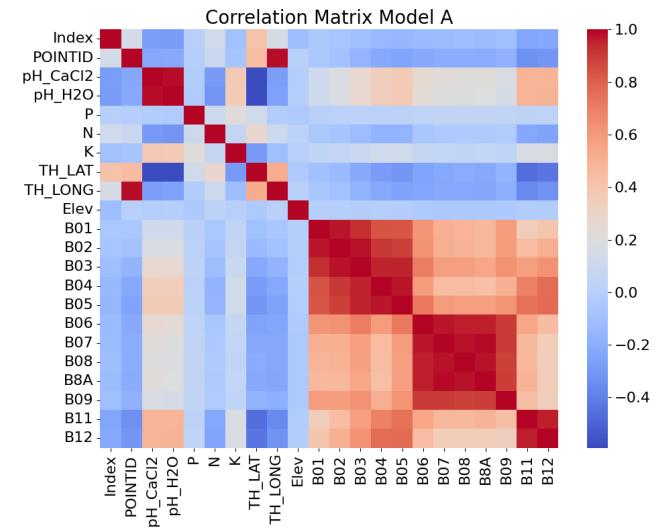

Fig. 2. Correlation coefficient matrix for the input data from Sentinel of the Europe model

2) Target Data:
The Model for Europe target dataset and its information is detailed below.

a) LUCAS 2018 TOPSOIL Dataset: The LUCAS Programme is an area-frame statistical survey organized and

managed by Eurostat (the Statistical Office of the European Union) to monitor changes in land use (LU) and land cover (LC) over time across the EU [41]. Since 2006, Eurostat has conducted LUCAS surveys every three years, relying on visual assessments of environmental and structural landscape features at georeferenced control points. These points are located at intersections of a 2km × 2km regular grid spanning the EU, generating approximately one million georeferenced points. In each survey, a subsample of these points is selected for collecting field-based information. The LUCAS 2018 TOPSOIL dataset comprises detailed soil property measurements from 18,984 samples collected throughout the European Union and the UK. It includes data on pH ($CaCl_2$ and H_2O), organic carbon, $CaCO_3$, nitrogen, phosphorus, potassium, electrical conductivity (EC), and oxalate-extractable iron and aluminum [12]. Figure 3 provides a geographical overview of the data points included in the dataset. As noted previously, the goal is to predict values for pH ($CaCl_2$ and H_2O), nitrogen, phosphorus, and potassium. For pH, two measurements are available depending on whether $CaCl_2$ or H_2O is used. Each of the four target nutrients has a distinct limit of detection (LOD), defined as the lowest quantity that can be measured with sufficient reliability. Table II summarizes these limits of detection. Out of the complete LUCAS 2018 TOPSOIL

Fig. 3. Locations of Soil Sample Collection of the LUCAS 2018 TOPSOIL Dataset

TABLE II
DESCRIPTION OF THE SOIL FIELDS IN THE LUCAS 2018 TOPSOIL DATASET

Field	Description	Unit	LOD
pH(CaCl2)	pH measured in a CaCl2 solution	-	2-10
pH(H2O)	pH measured in a suspension of soil in water	-	2-10
N	Total nitrogen content	g/kg	0.2
P	Phosphorus content	mg/kg	10
K	Extractable potassium content	mg/kg	10

dataset, 4,945 potassium values fall below the 10mg/kg limit of detection (LOD). These values are imputed by assigning an average estimate between 0 and 10mg/kg, resulting in a constant replacement of 5mg/kg. After imputing potassium, data cleansing and matching with Sentinel-2 imagery reduce the dataset to 18,471 soil samples. Figure 4 shows histograms of the target nutrients. While pH is relatively evenly distributed, nitrogen, phosphorus, and potassium exhibit a pronounced left skew. Along with min–max normalization, log normalization is evaluated for the skewed nutrients (N, P, and K); however, since it does not improve model performance, the min–max normalization approach is ultimately maintained for all four nutrients.

b) Correlation of the Target Data: Figure 2 also presents the correlation coefficient matrix for the target data. The very high correlation between pH_H2O and pH_CaCl2 arises because both values measure pH, albeit via two different methods. This strong correlation reflects the reliability of the pH measurements and suggests that training separate models for each pH variant may not be necessary. A notable linear correlation between input features and target values only emerges for the pH values and latitude (negative correlation), as well as for Sentinel bands B11 and B12 (positive correlation). These findings indicate that latitude
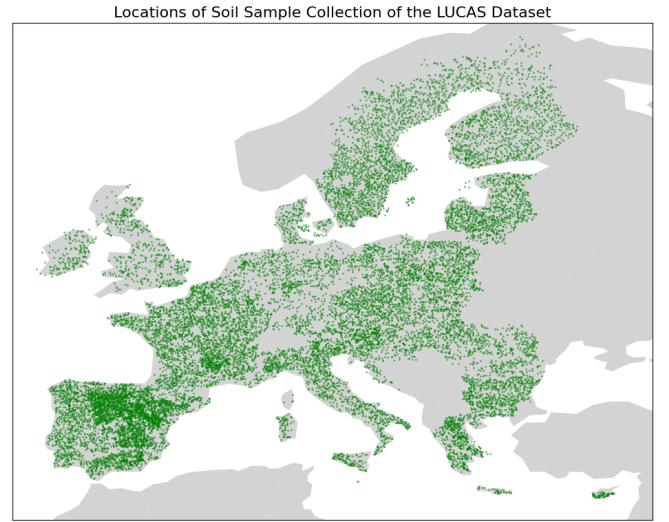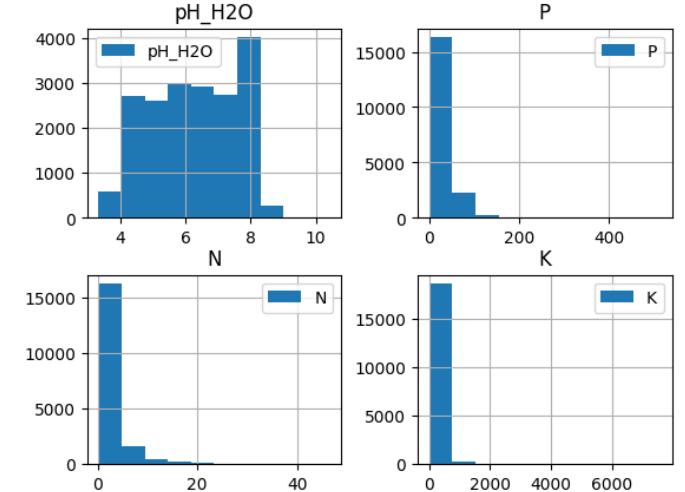

Fig. 4. Histogram for the target data nutrients pH, nitrogen, phosphorus and potassium in the LUCAS 2018 TOPSOIL dataset

and bands B11 and B12 are likely to be significant predictors of pH.

3) Dataset Split: The following section describes the procedure for partitioning the dataset, which is used for training and validating the model.

a) Single Split: The dataset, containing the previously described input and target data, includes 18,471 samples and is visualized in Figure 3. For training, it is randomly split into an 80:20 ratio, resulting in 14,776 samples in the training set and 3,695 samples in the test set. It is important to note that a single random split does not account for potential spatial dependencies in the data.

b) Spatial Cross Validation: As described in Section II-D1c, a single split does not account for spatial dependencies. Therefore, in addition to the single-split approach, spatial cross-validation (CV) is applied and its effect on performance

is evaluated using the XGBoost model. The corresponding results are discussed in Section III-C1a. To implement spatial CV, data points are grouped into grid cells based on their geographical location. The grid size affects both the total number of grid cells and the number of soil samples within each cell. Multiple grid sizes were considered during this project; this paper focuses on a $4° \times 4°$ grid. Each grid cell is used to split its contained data into training, validation, and test subsets. The overall goal is an approximate 60:20:20 split of the dataset. Because the grid cells contain different numbers of samples, the ratio can only be approximated. For a $4° \times 4°$ grid, the test dataset includes 3,567 samples, while 15,173 samples remain for training. The training set is then further divided via a 5-fold CV approach, where each fold consists of distinct grid cells. Based on the 5-fold CV, the $RMSE_{Average}$ is computed for the validation data. Additionally, the $RMSE_{Test}$ is calculated using the unseen test dataset, which does not appear in any training or validation folds. Figure 5 illustrates how the $4° \times 4°$ grid is partitioned into training, validation, and test cells.
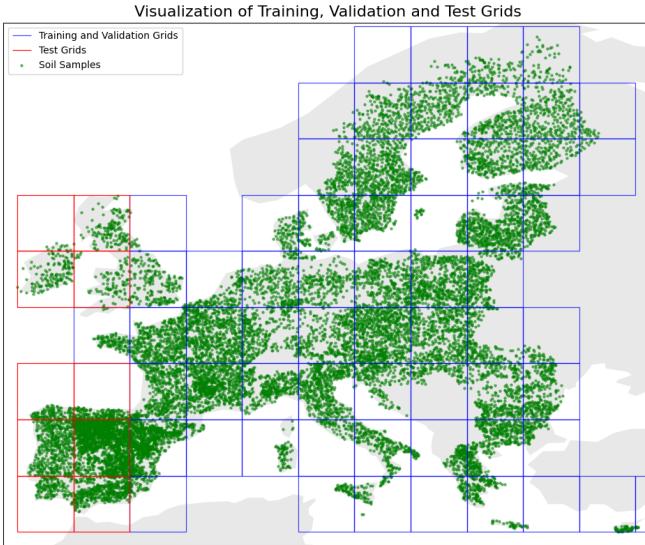

Fig. 5. Visualization of Training, Validation and Test Grids (Grid size: $4°\times 4°$)

B. Used Model Details

This section provides a detailed model overview for Europe, including hyperparameter optimization for XGBoost, FCNN and Random Forest. It explains which parameters are optimized for each model to achieve the best performance.

1) XGBoost: The hyperparameter tuning for the XGBoost model is performed using Optuna, as introduced in Section II-D2. The following two properties are pre-defined and not subject to hyperparameter optimization

- Evaluation Metric: The RMSE is the selected evaluation metric for model performance.
- Tree Construction Method: The hist algorithm is selected to accelerate the tree construction based on histograms for improved large dataset efficiency.

The following hyperparameters are optimized during the hyperparameter tuning process:

- Maximum Tree Depth (max_depth): The value for the maximum depth of the decision trees is optimized in the range from 3 to 12.
- Learning Rate (learning_rate): The learning rate is varied between 0.01 and 0.3 to control the model adaptation rate.
- Subsampling Rate (subsample): The fraction of the training data used for model fitting is optimized between 0.6 and 1.0.
- Column Sampling (colsample_bytree): The fraction of features randomly selected for each tree is optimized between 0.6 and 1.0.
- Gamma: The regularization term controlling the minimum reduction of the loss for each split is optimized between 0 and 1.
- L1 Regularization (reg_alpha): The L1 regularization parameter is set in the range from 0 to 1 to promote sparser models.
- L2 Regularization (reg_lambda): The L2 regularization parameter is also varied between 0 and 1 to prevent overfitting.

An overview of the selected hyperparameters for each XGBoost model is shown in table III. These optimized

TABLE III
XGBoost: Selected Hyperparameter for Model for Europe

	pH_CaCl2	pH_H2O	Phosphorus	Nitrogen	Potassium
max_depth	9	8	4	6	3
learning_rate	0.2413	0.2380	0.2398	0.1945	0.2573
subsample	0.8793	0.8366	0.8571	0.6997	0.6758
colsample_bytree	0.8405	0.8939	0.6542	0.7399	0.6788
gamma	0.7073	0.4134	0.8846	0.5221	0.5713
reg_alpha	0.3075	0.9372	0.6006	0.4994	0.5544
reg_lambda	0.7443	0.0988	0.5232	0.9292	0.5435

parameters highlight that model complexity and regularization has to be tailored to the specific characteristics of the target nutrition. The models for pH have deeper trees (8 and 9), indicating that predicting pH values requires greater model complexity- possibly because pH is influenced by multiple non-linear interactions. Phosphorus, nitrogen, and potassium, on the other hand, have moderate values, suggesting a balanced regularization approach. To assess the model performances of the trained XGBoost models, the test dataset is applied to the models to calculate the RMSE. The results are presented in table VI.

2) Fully Connected Neuronal Network: The FCNN serves as a foundational architecture in this study, leveraging multiple hidden layers to capture complex relationships in the data. To ensure an optimal model configuration, a hyperparameter search was conducted using Optuna, allowing for dynamic adjustments based on performance criteria. Various architectural and training parameters were explored, including the number of hidden layers, neurons per layer, dropout rates, learning rates, optimizers, and batch sizes. These parameters are fine-tuned to balance model complexity and generalization, minimizing the risk of overfitting while ensuring efficient

learning. The key hyperparameter settings are detailed as follows:

- Number of Hidden Layers: The network depth is optimized over a range of 1 to 5 layers.
- Number of Neurons per Layer: For each hidden layer, the number of neurons is determined individually within the range of 8 to 128 (step size 4).
- Dropout Rate: To avoid overfitting, the dropout functionality is used with a rate between 0.1 and 0.5.
- Learning Rate: The learning rate is varied between 0.0001 and 0.01 to control model adaptation.
- Optimizer: Only SGD and Adam are considered in this project.
- Batch Size: Three options were evaluated—16, 32, or 64.

The selected hyperparameters vary across models to optimize performance, as shown in Table IV. The number of hidden layers ranges from 8 to 18, with Adam as the chosen optimizer for all models. Learning rates are adjusted per target variable, with phosphorus requiring the lowest (0.00057) and pH in $CaCl_2$ the highest (0.00568). Batch sizes differ, with 16 or 32 for pH models and 64 for nutrients. These variations highlight the need for tailored configurations to ensure optimal training and generalization.

TABLE IV
FCNN: Selected Hyperparameter for Model for Europe

	pH_CaCl2	pH_H2O	Phosphorus	Nitrogen	Potassium
# hidden_layer	13	8	9	18	8
learning_rate	0.00568	0.00031	0.00057	0.00164	0.00136
optimizer	Adam	Adam	Adam	Adam	Adam
batch_size	16	32	64	64	64

3) Random Forest: As for the other models, optuna is used to optimize the hyperparameters for determining the best performing model.

The following hyperparameters are optimized during the hyperparameter tuning process:

- Number of Estimators: A range of 50 to 500 is given to select the number of estimators.
- Max Depth: The value for the maximum depth of the decision trees is optimized in the range from 3 to 30.
- Minimum Sample Split: The minimum sample split starts at 2 and ends up to 20.
- Minimum Sample Leafs: The minimum sample leafs starts at 1 and ends up to 20.
- Maximum Features: The maximum feature range is between 0.1 to 1.0.

An overview of the selected random forest hyperparameters for each target value is shown in table V.

C. Results

This chapter presents the achieved model performance and discusses feature importance. The analysis is done for the five key soil parameters: pH (measured in $CaCl_2$ and H_2O), phosphorus, nitrogen, and potassium.

TABLE V
Random Forest: Selected Hyperparameter for Model for Europe

	pH_CaCl2	pH_H2O	Phosphorus	Nitrogen	Potassium
estimators	352	447	402	451	331
max_depth	14	17	10	9	19
min_samples_split	14	17	7	14	15
min_samples_leaf	6	7	11	10	17
max_features	0.7323	0.4630	0.7243	0.9900	0.1101

1) Model Performance: In the following section, the results obtained by the three models are presented. As a performance metric, the Root Mean Squared Error (RMSE) of the final models is calculated and displayed in Table VI. The RMSE is determined using the test dataset from the single-split method described in Section III-A3a. Each of the three model variants is trained for each of the five nutrients.

TABLE VI
Model for Europe: Performance Results Test Dataset

Model Variant	Nutrient	Unit	(Mean ± StdDev)	RMSE (Test)
XGBoost	pH in CaCl2	-	5.71 ± 1.40	1.09
	pH in H2O	-	6.26 ± 1.32	1.03
	Phosphorus, extractable (P)	mg/kg	26.95 ± 27.02	26.53
	Nitrogen, extractable (N)	g/kg	3.15 ± 3.70	3.63
	Potassium, extractable (K)	mg/kg	204.83 ± 208.25	216.48
FCNN	pH in CaCl2	-	5.71 ± 1.40	1.12
	pH in H2O	-	6.26 ± 1.32	1.08
	Phosphorus, extractable (P)	mg/kg	26.95 ± 27.02	25.50
	Nitrogen, extractable (N)	g/kg	3.15 ± 3.70	3.44
	Potassium, extractable (K)	mg/kg	204.83 ± 208.25	178.20
Random Forest	pH in CaCl2	-	5.71 ± 1.40	1.09
	pH in H2O	-	6.26 ± 1.32	1.02
	Phosphorus, extractable (P)	mg/kg	26.95 ± 27.02	26.50
	Nitrogen, extractable (N)	g/kg	3.15 ± 3.70	3.63
	Potassium, extractable (K)	mg/kg	204.83 ± 208.25	216.06

a) Results XGBoost: When comparing the RMSE for pH in $CaCl_2$ and pH in H_2O, similar values are observed, indicating comparable accuracy in predicting pH values in both media. Because the pH range extends from 0 to 14, these errors are regarded as moderate. The RMSE of 26.53mg/kg for phosphorus and 3.15g/kg for nitrogen are both close to the standard deviations within the dataset, suggesting that the prediction accuracy for these nutrients aligns well with the observed variability but could still be improved. The RMSE of 216.48mg/kg for potassium is higher yet remains close to the dataset's standard deviation (208.25mg/kg), indicating that the XGBoost model captures variations in potassium levels reasonably well. Additionally, a comparison between the single-split and spatial CV approaches is conducted for the XGBoost model. The model is trained using both the single-split approach (with separate training and test datasets) and spatial CV (with training, validation, and test datasets). The results of this comparison are displayed in Table VII. In column three, the results of the single-split method, described earlier in this section, are presented for comparison. Columns four and five show the metrics $RMSE_{Average}$ and $RMSE_{Test}$ for the spatial CV. The results for the two pH values exhibit only minor differences between the single split and spatial CV, suggesting that no significant geographical dependencies

TABLE VII
COMPARISON XGBOOST RESULTS: SINGLE SPLIT VS. SPATIAL CV

Nutrient	Unit	$RMSE_{Test}$ (Single Split)	Spatial CV	
			$RMSE_{Average}$	$RMSE_{Test}$
pH in CaCl2	-	1.09	1.09	1.15
pH in H2O	-	1.03	1.03	1.10
Phosphorus	mg/kg	26.53	26.98	26.32
Nitrogen	g/kg	3.63	3.72	2.46
Potassium	mg/kg	216.48	207.68	177.52

Fig. 6. Compare Training and Test loss of potassium

exist for pH predictions. A similar observation applies to phosphorus, where model performance does not notably differ across the two methods. A more pronounced difference is observed when nitrogen and potassium results are compared between the single split and spatial CV. The RMSE$_{Test}$ for the spatial CV is lower than for the single-split method, which is unusual for spatial CV. Analysis indicates that the distribution of data points in the spatial CV test dataset is more favorable for the model, with many test data points resembling those in the training set and containing fewer extreme or rare values. Consequently, the lower RMSE$_{Test}$ may be misleading, as the test dataset is not fully representative of the overall data distribution. This issue arises because the test set in spatial CV is defined by geographic grids rather than random selection, potentially reducing the diversity of cases included in the test subset. Since the XGBoost-based analysis does not reveal substantial differences between the two validation methods (except for nitrogen and potassium), the remaining models are trained using the less complex single-split method. Nevertheless, spatial CV remains a promising avenue for future work. The aforementioned limitations of grid-based spatial CV should be taken into account, and more sophisticated approaches to data partitioning should be explored.

b) Results FCNN: The FCNN initially underperforms compared to XGBoost and Random Forest in predicting phosphorus, with an RMSE of 27.12, while XGBoost achieves 26.53. Investigation into the model's performance indicates potential overfitting or underfitting issues. Despite attempts to mitigate overfitting through regularization, improvements remain marginal. However, an intermediate FCNN model with two hidden layers and 152 neurons unexpectedly surpasses XGBoost, achieving an RMSE of 26.22. Further optimization using Optuna, with the number of hidden layers restricted between three and nine, yields an optimized FCNN featuring nine hidden layers and 708 neurons, which achieves a notably lower RMSE of 25.50. The final FCNN model also outperforms XGBoost and Random Forest in predicting nitrogen and potassium, with RMSE values of 3.44 and 178.20, respectively, compared to XGBoost's 3.63 and 216.48. To verify generalization, training and test loss trends are examined. The initial epoch (error = 198.13) is excluded for clarity, revealing a consistent decline in both losses over the first ten epochs. From epoch 15 onward, the test loss remains stable while training loss continues to drop, suggesting the onset of overfitting. Figure 6 illustrates the loss progression, confirming the model's robustness.

c) Results Random Forest: The performance results of the Random Forest model are presented in Table VI. For pH
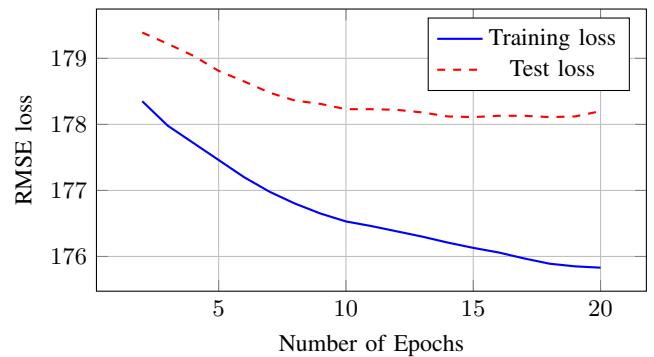
measurements, the model achieves a RMSE of 1.09 for pH in $CaCl_2$ and 1.02 for pH in H_2O, demonstrating relatively low error values. Regarding phosphorus, the RMSE is 26.50 mg/kg, closely aligning with the mean and standard deviation of the dataset, suggesting a moderate predictive performance. Similarly, for nitrogen, the model produces an RMSE of 3.63 g/kg, which is comparable to its standard deviation, indicating a consistent error margin. The highest RMSE is observed for potassium, with a value of 216.06 mg/kg, reflecting a greater variability in the dataset. Overall, the Random Forest model provides competitive results, particularly for pH prediction, while showing larger errors in nutrient predictions, likely due to the high variability within the dataset.

d) Comparison of Results: The performance evaluation of the three model types, XGBoost, FCNN, and Random Forest, highlights their predictive capabilities across different soil nutrients (Table VI). XGBoost demonstrates strong overall performance, achieving an RMSE of 26.53mg/kg for phosphorus, 3.63 g/kg for nitrogen, and 216.48mg/kg for potassium. Its RMSE for pH prediction remains low, with values of 1.09 for pH in $CaCl_2$ and 1.03 for pH in H_2O, indicating a reliable predictive ability for soil acidity. While XGBoost performs well for pH and phosphorus, its errors for nitrogen and potassium are comparable to those of Random Forest, suggesting challenges in capturing variability in nutrient concentrations. The Random Forest model delivers similar performance, particularly for pH prediction, with RMSE values of 1.09 and 1.02 for pH in $CaCl_2$ and H_2O, respectively. For phosphorus and nitrogen, it achieves RMSE values of 26.50mg/kg and 3.63g/kg, closely aligning with those of XGBoost. However, its largest error is observed for potassium, with an RMSE of 216.06mg/kg, reflecting high variability within the dataset. Despite competitive performance, especially for pH prediction, its accuracy in nutrient estimation remains limited.

In contrast, the optimized FCNN model surpasses both XGBoost and Random Forest for phosphorus, nitrogen, and potassium prediction. After hyperparameter tuning using Optuna, the FCNN achieves an RMSE of 25.50mg/kg for phosphorus, improving upon XGBoost and Random Forest. Additionally, it reduces the RMSE for nitrogen to 3.44g/kg and for potassium to 178.20mg/kg, outperforming the other models. pH predictions, however, remain slightly

less accurate, with RMSE values of 1.12 for pH in $CaCl_2$ and 1.08 for pH in H_2O. Training and test loss evaluations confirm the model's robustness, as illustrated in Figure 6. Overall, the FCNN demonstrates superior performance for nutrient prediction, particularly for phosphorus and potassium, while maintaining competitive accuracy for pH measurements.

2) Feature Importance: In the following, feature importance is illustrated for the XGBoost model. The evaluation is based on the average gain across all splits in which a given feature is utilized. For potassium (see Figure 7), there is no single dominant feature. Instead, three Sentinel-2 bands (B12, B07, and B05) emerge as the most influential for predicting this target variable. In figure 8 the most important features for

Fig. 9. Feature Importance for P - XGBoost Model for Europe

pH reveals some notable observations. Because two target values are available—one measured in H_2O and another in $CaCl_2$—it is possible to compare feature importance for each measurement method (Figures 10 and 11). In both cases, B12 emerges as a highly influential feature, serving as the most important feature for $CaCl_2$ and ranking second for H_2O. For pH measured in H_2O, band B05 takes on a slightly higher importance. Although this may seem to conflict with the high linear correlation between pH in H_2O and $CaCl_2$, as detailed in Paragraph III-A1b, these differences in feature importance are not surprising given the non-linear and multi-split nature of tree-based models.
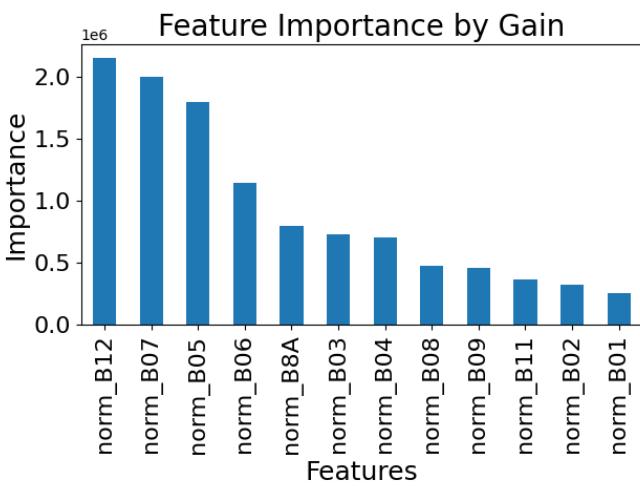

Fig. 7. Feature Importance for K - XGBoost Model for Europe

nitrogen are the bands B04 and B12. All other bands have a minor contribution. For phosphorus, the XGBoost feature

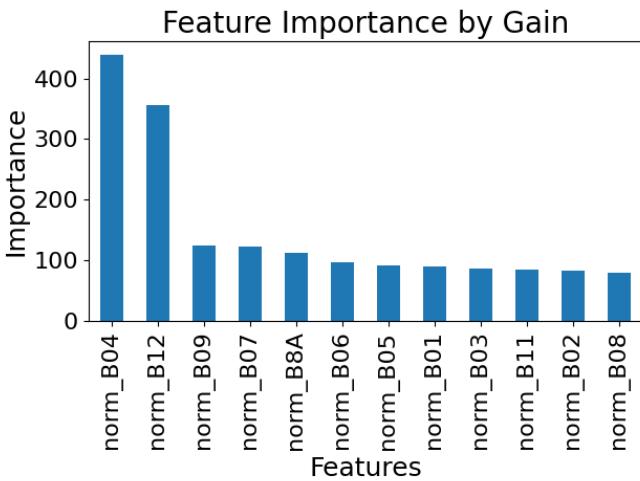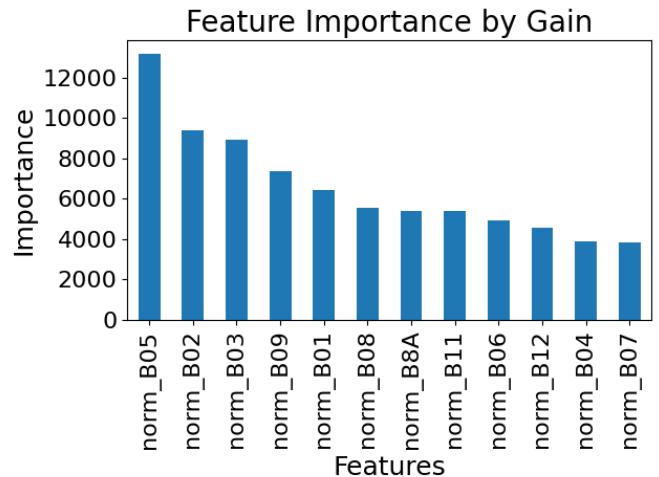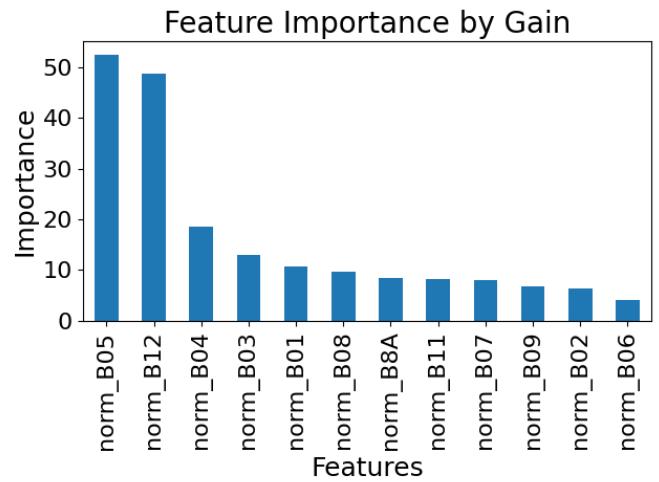Fig. 10. Feature Importance for pH from H2O - XGBoost Model for Europe

Fig. 8. Feature Importance for N - XGBoost Model for Europe

importance illustrated in Figure 9 indicates that band B05 has the greatest influence, while bands B02 and B03 also contribute significantly to the model. Finally, the analysis for

To summarize the feature importance results, band 12 is identified as the most influential feature for the model's predictions. In four out of the five cases, band 12 is ranked among the top two features—it is the most important feature in two cases and the second most important in another two, with the only exception being the phosphorus prediction where it plays a minor role. Overall, band 05 emerges as the next
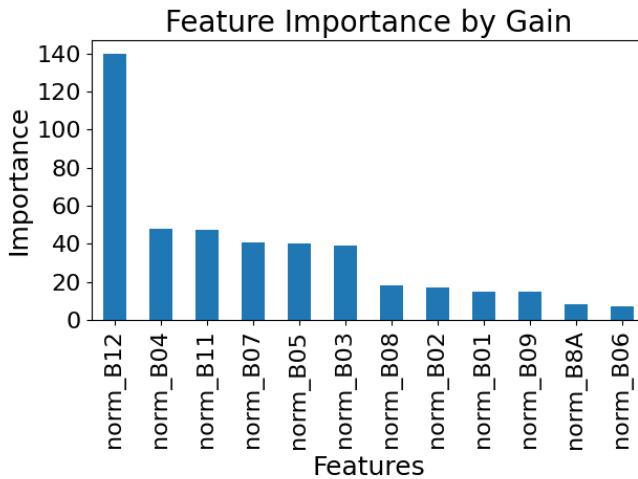
Fig. 11. Feature Importance for pH from CaCl2 - XGBoost Model for Europe

most influential feature, achieving the top rank twice.

IV. EXTENDED MODEL FOR EUROPE

Based on the Model for Europe in the previous section III, we decided to extend the scope to train the model by using additional features. We therefore expect increased precision and reduced prediction error with this extended model compared to the previous, simpler input model.

A. Data Set

In addition to the single pixel from Sentinel-2 satellite image bands, we consider the following data as additional features for model training.

- 8 neighbor pixels: 3 x 3 = 9 pixels instead of the single pixel
- Weather data: 9 features
- Crop yield scores: 27 features
- Clay model embeddings from Masked Auto Encoder (MAE): 1024 features

1) Neighbor Pixels of the Sentinel Data: Satellite data are, by nature, image data. For ML tasks involving images, neighboring pixels are often used—such as in Convolutional Neural Networks—to capture local spatial correlations [16]. In a soil prediction task, the use of neighbor pixels must be balanced between more neighbor pixels to get more information and less neighbor pixels to reduce unwanted noise. This noise occurs due to the large size of a pixel (10 m x 10 m up to 60m x 60m) compared with the small size of a field in most areas of the world. Using too many pixels can increase noise from adjacent buildings, streets, and non-agricultural areas. Therefore only direct neighbor pixels are used.

An investigation of the linear correlation coefficient matrix shown in figure 12, shows the very similar fundamental correlation between bands like in figure 2, but with a higher "resolution". This is caused by the very high correlation of

the additional neighbor pixels to the central pixel. Given that a change in information over a spatial step of 10 m to 60 m is unlikely, it is inferred that the additional neighbor pixels do not provide much extra information to the model.
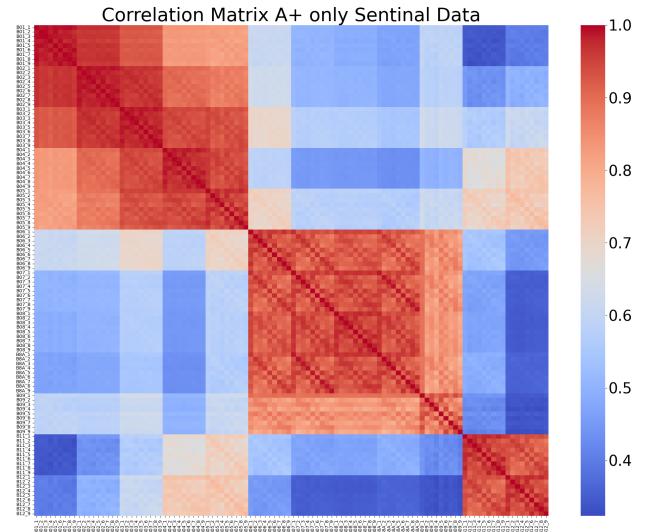

Fig. 12. Correlation coefficient matrix for the neighbor pixel of the Sentinel bands for the extended Europe model

2) Weather Data: Weather data is integral to understanding soil nutrient dynamics, as climatic variables strongly influence moisture availability and nutrient leaching. In this project, weather data (e.g., temperature, precipitation, dew point) is aligned with the satellite image timestamps to ensure that each soil sample's spectral reading corresponds to concurrent weather conditions. Access to this historical data incurs costs after 1,000 API calls per day.

Figure 13 presents the correlation coefficient matrix for the extracted weather variables. As expected, temperature (`OW_temp`) shows a nearly perfect correlation with the temperature-feels-like metric (`OW_feels_like`). Additionally, dew point (`OW_dew_point`) is highly correlated with both `OW_temp` and `OW_feels_like`, illustrating their shared underlying physical principles.

While such high correlations can reduce feature diversity, they also reaffirm that these variables capture significant climate-driven effects that potentially influence topsoil properties.

By incorporating these weather variables into the extended model, a modest improvement in RMSE scores was observed for certain nutrients (e.g., phosphorus and potassium). This improvement suggests that specific climatic conditions—such as rainfall patterns or heat stress—can provide meaningful context beyond raw reflectance signals alone.

3) Crop Yield Scores: Crop yield data serves as a proxy for soil fertility when direct soil measurements are either unavailable or incomplete. In this study, the project team integrated yield metrics derived from the FAO GAEZ portal, which provides production estimates for various crops (e.g.,
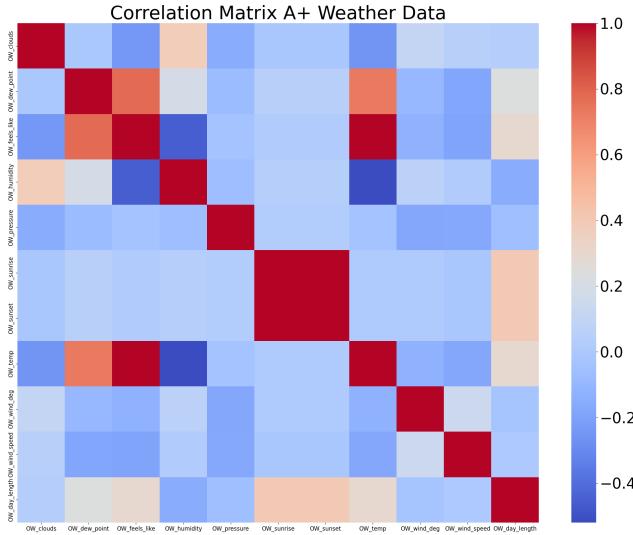
Fig. 13. Correlation coefficient matrix for the weather data for the extended Europe model

Fig. 14. Correlation coefficient matrix for the World Yield data for the extended Europe model

wheat, maize, barley, and oilseeds) across different regions. These yield values, normalized to ensure comparability, were aligned with the spatial coordinates of soil sampling.

a) Rationale for Including Yield Data: High or low crop yields typically reflect underlying soil characteristics, climatic conditions, and farm management practices. By incorporating crop yield information, the model gains an indirect measure of long-term soil productivity, bridging potential gaps in direct soil property data. Furthermore, it provides additional context information for satellite images since it may reveal characteristic crop reflectance patterns at the pixel level.

b) Correlation Analysis: A correlation matrix of the yield variables (Figure 14) revealed that certain crops share strong linear relationships (for instance, barley (`brl`) and wheat (`whe`)), reflecting similar agronomic requirements. Meanwhile, crops not commonly cultivated in Europe lacked data points in this region, leading to sparse or zero values. Despite these gaps, the inclusion of yield scores contributed to capturing broader agricultural contexts, thereby enhancing model performance for certain nutrients (notably nitrogen and phosphorus). However, excessive reliance on yield data could introduce biases if confounding factors (e.g., irrigation, fertilization regimes) are not adequately represented.

4) Clay Model: The Clay model is an open-source foundation model for Earth observation [42]. The goal of Clay is to make data cheaper, easier to use, and more accessible to communities and everyone working on climate and nature. Clay's model uses satellite imagery, as well as information about location (longitude, latitude) and time, as input. As output we obtain embeddings, which can be considered as mathematical representations of a given area at a certain time on the Earth's surface. Vision Transformer architecture is implemented for understanding geospatial and temporal relations on Earth Observation data. A Masked Autoencoder is used as a self-supervised learning method to train the models. In our project, we decided to use a pre-trained Clay model
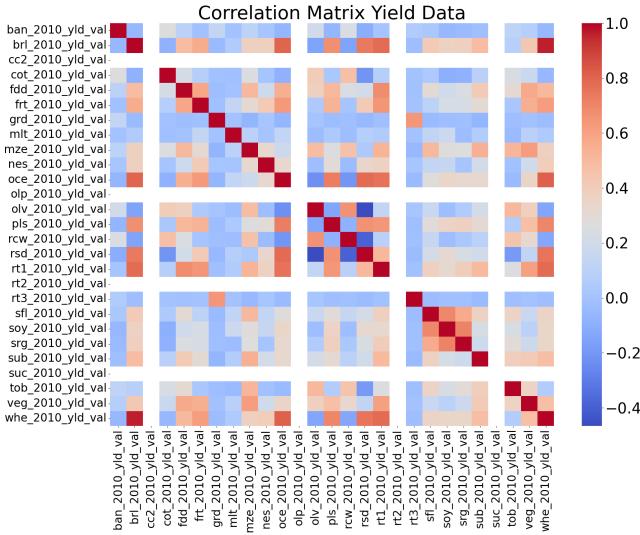
to generate semantic embeddings as extra features for training our own ML models for the soil prediction.

The Clay model expects a dictionary with the following keys of certain dimensions [42]:

- pixels: batch x band x height x width - normalized values
- time: batch x 4 - horizontally stacked week_norm and hour_norm
- latlon: batch x 4 - horizontally stacked lat_norm and lon_norm
- waves: list[:band] - wavelength of each band of the sensor from sentinel-2
- gsd: scalar - ground sample distance of the sensor from sentinel-2

The raw data will be preprocessed into the required dimension of the input data as listed above. Clay expects normalized pixel data coming from the satellite images. 9 x 9 pixels of each band are used as input. We set all the values of the parameter of the key "time" to zero, since we are much more interested in the location of the soil data and we don't transform the timestamp to weeks and hours as Clay requires. The values of longitude and latitude are normed using sine and cosine functions, respectively. That is the reason, why 4 values are needed for the key "latlon" for each data set. The ground sample distance is 60 m. It is pre-defined in order to use information of all the available 12 bands of the Sentinel-2 satellite.

The checkpoint we use, in which the pre-trained weights and bias are stored, can be downloaded from: https://huggingface.co/made-with-clay/Clay/resolve/main/v1.5/clay-v1.5.ckpt. The Clay model itself offers different model sizes as options: "tiny", "small", "base", and "large". The parameters from the checkpoint were trained with the "large" model, which uses a kernel size of 8 x 8. Decision has to be made by choosing the pixel resolution of the satellite images, so that the "large" model can be applied to the data set. Usage of other size of

models leads to mismatching errors, because the dimension of weights and bias from the pre-trained model checkpoint and the current model don't match.

After preprocessing, we have a dictionary with the following keys and dimensions:

- pixels: batch x 12 x 9 x 9
- time: batch x 4
- latlon: batch x 4
- waves: list[:band] 1 x 12
- gsd: 60

A data cube of all the keys and values as described above is prepared to feed the Clay model. The Clay model is imported to generate the embeddings with a dimension of 1024. It means that additional 1024 features are created to train our own models for soil prediction. For the illustration, the embeddings of the first 16 data sets are plotted as shown in figure 15 using a "bwr" (blue-white-red) color map. We transformed 1024 to a 32 x 32 matrix for better visualization.
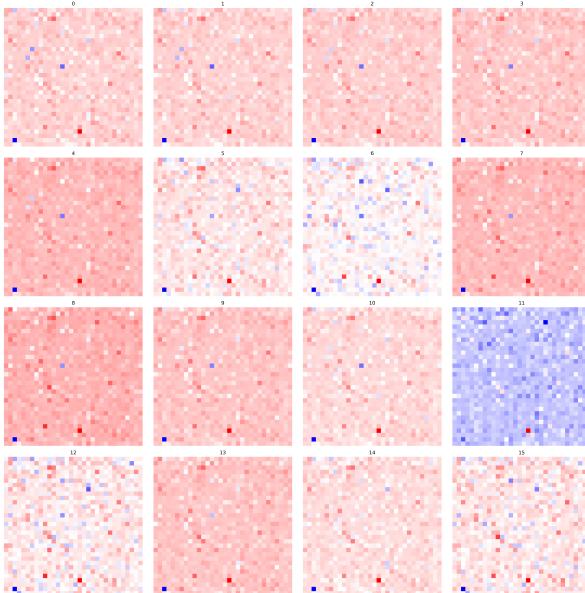

Fig. 15. Visualization of Clay embeddings

To complete the analysis of the Clay embeddings the linear correlation coefficients are shown in figure 16. As to expect for the abstract embeddings, no tendency towards a high positive or negative correlation can be observed.

5) Data Combinations for Training: Because multiple data sources are introduced in the extended model, the AgroLens project evaluates several feature configurations to determine which combination delivers the most accurate soil nutrient predictions:

- **Neighbor Pixels (SURR):** Expands each single-pixel input to include an adjacent 3×3 region, capturing local spatial variation.
- **Weather Data (WTHR):** Incorporates climate variables (e.g., temperature, precipitation) aligned with satellite imagery timestamps, providing environmental context.
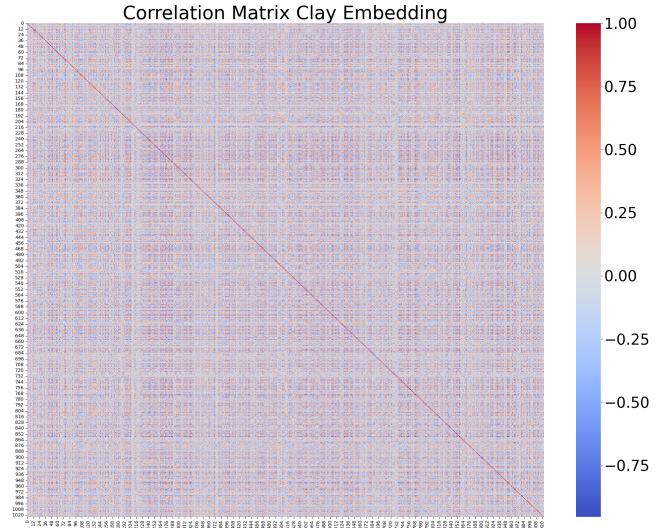

Fig. 16. Correlation coefficient matrix for the Clay embeddings for the extended Europe model

- **Crop Yield Scores (CRY):** Integrates FAO yield estimates for various crops, serving as a proxy for soil productivity.
- **Clay Embeddings (CLAY):** Adds a 1024-dimensional representation from a pre-trained Masked Autoencoder.

Two main configurations are tested within the Extended Model for Europe:

1) *SURR + WTHR + CRY*: A mid-level expansion of features, hypothesized to enhance predictions by capturing local spatial detail, climate factors, and crop performance.
2) *SURR + WTHR + CRY + CLAY*: A further extension, adding the Clay embeddings to the previous set. While these embeddings can encode high-level geospatial patterns, they substantially increase the dimensionality of the feature space and resulted in slower trainable and most often less performant models.

B. Model Adaption and Training

The primary distinction between the initial model for Europe and the extended model is an expanded set of input features. Consequently, only a change in the input vector size of the ML models is required. Furthermore, the feasibility of the parameter ranges used for hyperparameter optimization must be verified. Due to the increased volume of input data, it is possible that the models may require additional complexity to effectively process the larger amount of information. As demonstrated in the next section, *Results*, the models handle the expanded input data well, although the computational time for model training increases by approximately a factor of ten. As a quality check, the hyperparameters of the best model from the optimization run are analyzed. This analysis is performed for all models and target values. In Figure VIII, the parameters of the best XGBoost models without Clay embedding are presented. It is observed that no hyperparameter reaches the limits of the parameter ranges provided in

Section III-B, confirming that these ranges remain feasible for the extended model. Moreover, the hyperparameters vary significantly across the different target values, which further validates the decision to train separate models.

TABLE VIII
XGBOOST: SELECTED HYPERPARAMETER FOR EXTENDED MODEL WITHOUT CLAY EMBEDDINGS

	pH_CaCl2	pH_H2O	Phosphorus	Nitrogen	Potassium
max_depth	11	8	5	6	8
learning_rate	0.2652	0.2505	0.2079	0.2658	0.2317
subsample	0.9767	0.9288	0.9629	0.8898	0.6218
colsample_bytree	0.9477	0.9190	0.9319	0.7511	0.9473
gamma	0.6916	0.9914	0.7509	0.6356	0.1918
reg_alpha	0.5380	0.5489	0.3902	0.6018	0.3999
reg_lambda	0.4387	0.7022	0.5748	0.1716	0.8859

For the FCNN, the parameters of the best model without Clay data are presented in Figure IX. Significant differences between these models, as well as differences compared to the initial European model, are observed. In this comparison, the extended model without Clay requires fewer layers and a smaller learning rate than the initial European model.

TABLE IX
FCNN: SELECTED HYPERPARAMETER FOR EXTENDED MODEL WITHOUT CLAY EMBEDDINGS

	pH_CaCl2	pH_H2O	Phosphorus	Nitrogen	Potassium
# hidden_layer	8	9	4	4	7
learning_rate	0.00030	0.00018	0.00019	0.00019	0.00048
optimizer	Adam	Adam	Adam	Adam	Adam
batch_size	64	64	64	16	32

Finally, for the extended model without Clay embedding using Random Forest (RF), the best parameters are shown in Figure X. Here, the same conclusions can be drawn.

TABLE X
RANDOM FOREST: SELECTED HYPERPARAMETER FOR EXTENDED MODEL WITHOUT CLAY EMBEDDING

	pH_CaCl2	pH_H2O	Phosphorus	Nitrogen	Potassium
estimators	324	185	274	475	429
max_depth	16	19	23	27	23
min_samples_split	19	15	17	6	15
min_samples_leaf	13	4	7	17	6
max_features	0.7472	0.6249	0.4504	0.4401	0.6167

To complete the hyperparameter analysis, the best model parameters for the input data with Clay embedding are presented. For the XGBoost model, the best parameters are shown in Figure XI. Once again, no hyperparameter reaches the limits of the defined parameter ranges, confirming the validity of the optimization. Additionally, the parameters differ among the models for the various target values, as observed previously. In comparison to the model without Clay embedding, slight differences in the parameters are also noted.

Analysing the parameters for the FCNN model in figure XII leads to similar results.

The aforementioned effects are also observed for the Random Forest model parameters in Figure XIII. In this case, the differences between models with and without Clay embedding are even more pronounced than before.

TABLE XI
XGBOOST: SELECTED HYPERPARAMETER FOR EXTENDED MODEL WITH CLAY EMBEDDING

	pH_CaCl2	pH_H2O	Phosphorus	Nitrogen	Potassium
max_depth	7	8	4	5	3
learning_rate	0.2352	0.2503	0.2421	0.1945	0.2705
subsample	0.9334	0.8159	0.9515	0.6997	0.6012
colsample_bytree	0.8708	0.8837	0.9161	0.7399	0.6783
gamma	0.0418	0.1945	0.1861	0.5221	0.1975
reg_alpha	0.3183	0.3099	0.4329	0.4994	0.3617
reg_lambda	0.1657	0.9632	0.0112	0.8032	0.2392

TABLE XII
FCNN: SELECTED HYPERPARAMETER FOR EXTENDED MODEL WITH CLAY EMBEDDING

	pH_CaCl2	pH_H2O	Phosphorus	Nitrogen	Potassium
# hidden_layer	9	6	7	3	5
learning_rate	0.00031	0.00159	0.00065	0.00042	0.00111
optimizer	Adam	Adam	Adam	Adam	Adam
batch_size	64	16	16	16	32

To summarize the analysis, it is concluded that varying models for the different target values are necessary and should be considered in further developments. The differences observed for the same ML model when using different input data indicate that retraining is required whenever the input features are extended or modified. Nevertheless, the changes in the hyperparameters are not drastic, suggesting that the models are likely to yield feasible results for other features or datasets without the need for retraining.

C. Results

In the following section the achieved results of the three extended models for Europe - XGBoost, FCNN and Random Forest - are presented and discussed. Table XIV, XV, and XVI each show the overall performance of their respective model for Europe (BASE) compared to the extended models for Europe. Additionally, we give an overview of the mean together with the standard deviation to get a better evaluation of RMSE values.

1) XGBoost: Table XIV shows an overview of the extended models for Europe's performance with XGBoost.

a) SURR + WTHR + CRY: For pH the values are similar in $CaCl_2$ and H_2O, highlighting similar accuracy between both test methods with values of 0.86 and 0.81 and thus being 22% better than in BASE configuration. Also Phosphorus (24.88), Nitrogen (3.40), and Potassium (200.42) show a better RMSE than the BASE configuration with 7% improvement. It can also be mentioned, that all RMSE values are slightly to moderately better than the standard deviation.

b) SURR + WTHR + CRY + CLAY: The values of RMSE for pH (0.90 and 0.85) are 18% lower than compared to the BASE model. Phosphorus (25.05), Nitrogen (3.44), and Potassium (202.03) are 6% lower compared to the BASE model. Additionally, also with this configuration all RMSE values are slightly to moderately better than the corresponding standard deviation.

TABLE XIII
RANDOM FOREST: SELECTED HYPERPARAMETER FOR EXTENDED
MODEL WITH CLAY EMBEDDING

	pH_CaCl2	pH_H2O	Phosphorus	Nitrogen	Potassium
estimators	275	319	464	167	458
max_depth	15	26	29	27	18
min_samples_split	18	8	3	19	13
min_samples_leaf	3	2	17	15	20
max_features	0.6019	0.9486	0.4297	0.1507	0.9337

c) Effects of CLAY: When comparing *SURR + WTHR + CRY* against *SURR + WTHR + CRY + CLAY*, it is clear that *SURR + WTHR + CRY* performs slightly better than *SURR + WTHR + CRY + CLAY* across all nutrients. Hence for the XGBoost approach Clay may attribute negative effects to the models' performances due to the large number of extra features (1024) compared to other features all together.

TABLE XIV
EXTENDED MODEL FOR EUROPE: XGBOOST PERFORMANCE

Nutrient	Unit	Mean ± StdDev	RMSE		
			BASE	Previous+ SURR, WTHR, CRY	Previous+ CLAY
pH in CaCl2	-	5.71 ± 1.40	1.09	**0.86**	0.90
pH in H2O	-	6.26 ± 1.32	1.03	**0.81**	0.85
Phosphorus	mg/kg	26.95 ± 27.02	26.53	**24.88**	25.05
Nitrogen	g/kg	3.15 ± 3.70	3.63	**3.40**	3.44
Potassium	mg/kg	204.83 ± 208.25	216.48	**200.42**	202.03

2) Fully Connected Neural Networks: Table XV shows an overview of the Extended Model for Europe's performance with FCNNs.

a) SURR + WTHR + CRY: The pH values in $CaCl_2$ and H_2O are very similar with values of 0.93 and 0.87 resulting in 17% and 23% better performance compared to the BASE configuration. Also Phosphorus (24.37), Nitrogen (3.27), and Potassium (159.03) achieve a better RMSE than the BASE configuration with 11%, 5%, and 15% improvement. All RMSE values are slightly to moderate better than the standard deviation for each nutrient.

b) SURR + WTHR + CRY + CLAY: The values for pH (0.91 and 0.90) are 19% lower in comparison to the BASE configuration. For Phosphorus the RMSE (23.06) is 15% lower compared to BASE, whereas Nitrogen's RMSE (3.35) is only reduced by 3%. Potassium's RMSE (166.36) is reduced by 11% and all RMSE values are again slightly to moderate better than the standard deviation.

c) Effects of CLAY: We compare *SURR + WTHR + CRY* against *SURR + WTHR + CRY + CLAY* and the results are not as clear as previously with XGBoost. For the FCNN architecture *SURR + WTHR + CRY* as well as *SURR + WTHR + CRY + CLAY* both reach top RMSE values. *SURR + WTHR + CRY* performs better for pH in H_20, Nitrogen, and Potassium, whereas *SURR + WTHR + CRY + CLAY* performs better for pH in $CaCl_2$, and Phosphorus. Hence for the FCNN, Clay cannot be attributed a constant positive or negative effect to the models' performances.

TABLE XV
EXTENDED MODEL FOR EUROPE: FCNN PERFORMANCE

Nutrient	Unit	Mean ± StdDev	RMSE		
			BASE	Previous+ SURR, WTHR, CRY	Previous+ CLAY
pH in CaCl2	-	5.71 ± 1.40	1.12	0.93	**0.91**
pH in H2O	-	6.26 ± 1.32	1.12	**0.87**	0.90
Phosphorus	mg/kg	26.95 ± 27.02	27.12	24.37	**_23.06_**
Nitrogen	g/kg	3.15 ± 3.70	3.44	**_3.27_**	3.35
Potassium	mg/kg	204.83 ± 208.25	185.31	**_159.03_**	166.36

3) Random Forest: Table XVI shows an overview of the extended models for Europe's performance with Random Forest.

a) SURR + WTHR + CRY: The RMSE values for pH in $CaCl_2$ and H_2O (0.85 and 0.80) are 22% better than the BASE configuration. Phosphorus with an RMSE of 24.60 and Nitrogen with an RMSE of 3.37 are 7% better than BASE. Potassium's RMSE (192.01) is 12% better compared to BASE. While observing the standard deviation all RMSE values are slightly to moderately better than the standard deviation.

b) SURR + WTHR + CRY + CLAY: For pH in $CaCl_2$ and H_2O the RMSE (0.89 and 0.84) are 18% better than the BASE RMSE values. Phosphorus (24.80) and Nitrogen (3.42) have a 6% better RMSE than BASE. Potassium with an RMSE of 197.36 is improved by 9% in comparison to BASE. All RMSE values are slightly to moderate better than the standard deviation.

c) Effects of CLAY: A comparison of *SURR + WTHR + CRY* to *SURR + WTHR + CRY + CLAY* shows, that for all nutrients *SURR + WTHR + CRY* performs better than *SURR + WTHR + CRY + CLAY*. Hence for the Random Forest approach Clay can be attributed with a slightly negative effect to the model's performance.

TABLE XVI
EXTENDED MODEL FOR EUROPE: RANDOM FOREST PERFORMANCE

Nutrient	Unit	Mean ± StdDev	RMSE		
			BASE	Previous+ SURR, WTHR, CRY	Previous+ CLAY
pH in CaCl2	-	5.71 ± 1.40	1.09	**_0.85_**	0.89
pH in H2O	-	6.26 ± 1.32	1.02	**_0.80_**	0.84
Phosphorus	mg/kg	26.95 ± 27.02	26.50	**24.60**	24.80
Nitrogen	g/kg	3.15 ± 3.70	3.63	**3.37**	3.42
Potassium	mg/kg	204.83 ± 208.25	216.06	**192.01**	197.36

4) Model Performance Comparison: We compare all three models to each other to identify the best performing model and configuration for each nutrient.
XGBoost with both *SURR + WTHR + CRY* as well as *SURR + WTHR + CRY + CLAY* configuration performs less across all nutrients when compared to FCNN and Random Forest. The FCNN approach outperforms the Random Forest approach

negative effect to the models' performances.

for Phosphorus, Nitrogen, and Potassium. Interestingly for Phosphorus *SURR + WTHR + CRY + CLAY* performs better than *SURR + WTHR + CRY*. This implies that Clay has positive effect to reduce the RMSE at least for Phosphorus. Hence *SURR + WTHR + CRY + CLAY* with the FCNN should be picked for Phosphorus. For both Nitrogen and Potassium the *SURR + WTHR + CRY* configuration performs better without Clay by achieving its best RMSE with the FCNN approach. Hence, for the nutrients Nitrogen and Potassium *SURR + WTHR + CRY* with the FCNN approach should be selected.

For pH in $CaCl_2$ and H_2O the Random Forest *SURR + WTHR + CRY* without clay performs the best and should be used for prediction.

5) Feature Importance: Further, we want to achieve an insight into the influence of the input data on the model performance. The feature importance for the extended model based on the XGBoost is shown in the following figures. Due to the high number of input features, only the top 20 most important features are plotted.

In this analysis for potassium (Figure 17), the feature 'Clay embedding_691' is the most important (highest gain), followed by two other Clay embeddings; a Sentinel band appears at rank 4, and a yield feature ranks 6th.
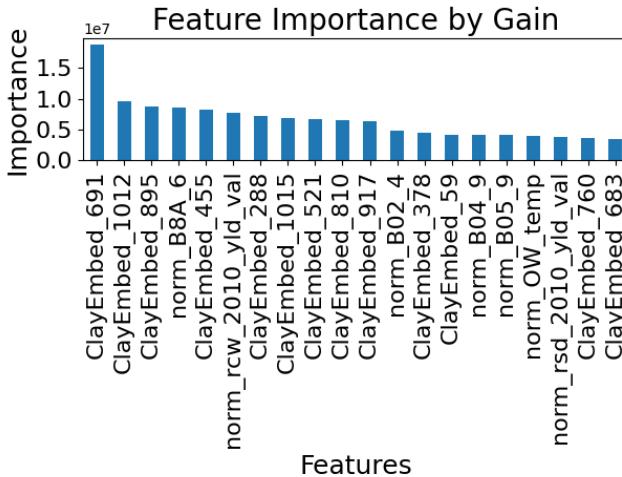

Fig. 17. Comparison of the 20 most important features of the extended data sets for K, based on the extended XGBoost model

Nitrogen as next target value has a Yield input data as the most important feature, namely norm_cot_2010_yld_val, which is the actual yield for cotton, as can be seen in figure 18. The next important features, with a relatively lower gain, are Clay embeddings and Sentinel bands.

The resulting feature importance for phosphorus in figure 19 shows features with a very similar gain. Starting with norm_frt_2010_yld_val (fruits) a yield feature is most important. Followed by a Sentinel band, two Clay embeddings, and an additional yield input.

For the two pH predictions, the feature importance for the H_2O-based model is presented in Figure 20 and for the $CaCl_2$-based model in Figure 21. It is observed that the most
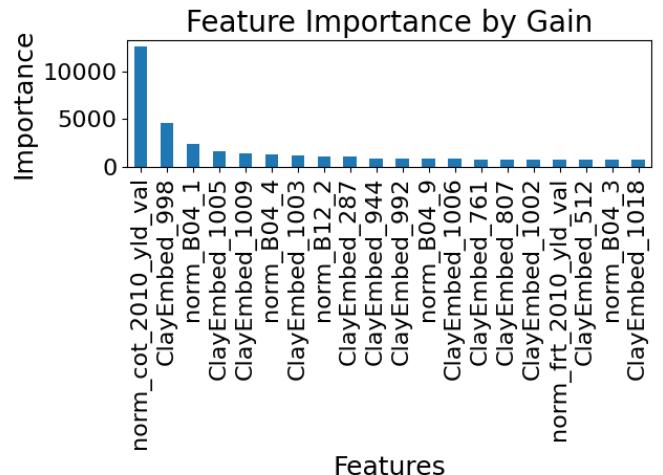

Fig. 18. Comparison of the 20 most important features of the extended data set for N, based on the extended XGBoost model
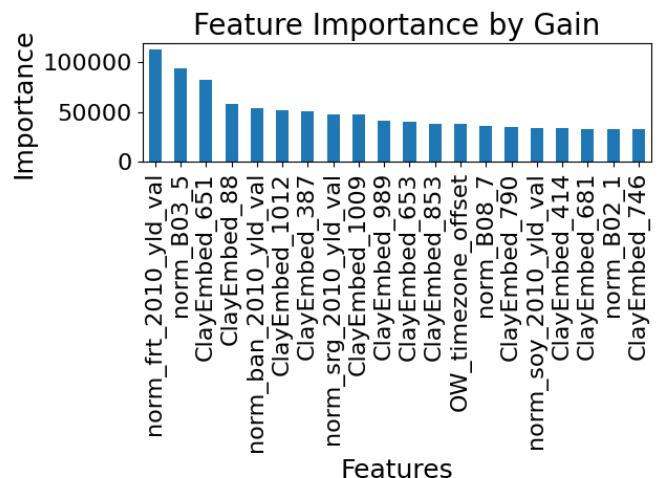

Fig. 19. Comparison of the 20 most important features of the extended data set for P, based on the extended XGBoost model

and second most important features for both pH values are the same clay embeddings. The third-ranked feature is a yield input, although not identical between the two models. The subsequent two features are Sentinel bands in both cases. Overall, the feature importance for both pH predictions is very similar, which aligns with the high correlation between the two pH target values.

In conclusion, it is clear that Clay embeddings have a high impact on the feature importance, even if they don't always improve the prediction quality as discussed in the previous section.

V. CHALLENGES IN EXTENDING ON AFRICA

As described in Section II-B2, a key problem with African soil data is the absence of timestamps, which makes it impossible to map a soil sample to a timely, accurate satellite image.
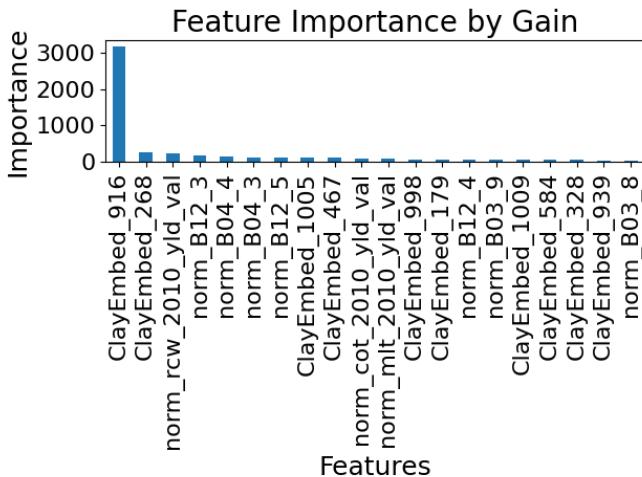
Fig. 20. Comparison of the 20 most important features of the extended data set for pH from H2O, based on the extended XGBoost model
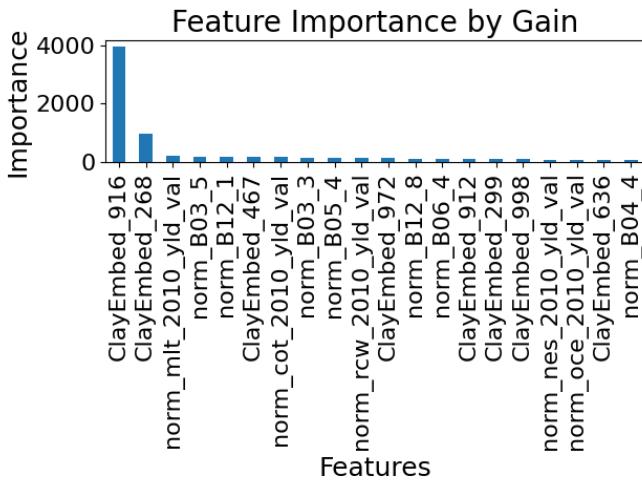

Fig. 21. Comparison of the 20 most important features of the extended data set for pH from CaCl2, based on the extended XGBoost model

A. Input Data

The currently available WoSIS dataset predates the launch of Sentinel-2, making it necessary to search for an alternative satellite imagery source. Only around 200 WoSIS data points outside of Europe fall within the timeframe since the launch of Landsat 8; therefore, we shift our focus to using Landsat 7 images. However, additional challenges come along, as Landsat 7 suffers from a Scan Line Corrector (SLC) failure since 2003, causing gaps in the resulting satellite images [6]. Obtaining historical data for Landsat 7 was also difficult, as the Copernicus Data Space provided only a limited number of datasets. In order to address this, the existing data acquisition scripts are extended to use the USGS M2M API, the official source of Landsat data [43]. Unfortunately, while the API allows searching and retrieving metadata, its download endpoint is disabled, and no further information is provided regarding the reason. The last remaining free source for Landsat 7 images is Google Earth Engine, but integrating

it would have required significant modifications to the existing data collection scripts, which would be beyond the scope of the project [44]. Due to the challenges mentioned above, the following approaches could not be validated. Nonetheless we hope that the following project continues from this point once adequate satellite data become available..

B. Target Data

The global soil data retrieved from WoSIS is shown in TABLE I. WoSIS (World Soil Information Service) [13] is a global initiative led by the International Soil Reference and Information Centre (ISRIC) with the primary purpose of facilitating the harmonization, collection, and dissemination of soil data. It has more than 228,000 worldwide profiles available which have been collected between 1920 and 2020. However, over 74% of WoSIS profiles were collected at depths greater than 30 cm. For our analysis, only soil profiles with a maximum depth of 25 cm are considered, since satellite imagery can only capture information from the topsoil. Additionally, almost 27% of WoSIS soil profiles don't have timestamps. After filtering based on these criteria, ensuring Landsat 7/8 data availability (after March 2008), and excluding European data (to avoid overlap and bias with the LUCAS 2018 TOPSOIL dataset), only 3,641 soil profiles remain. Unfortunately, within this subset, only one soil dataset is located in Africa.

TABLE XVII
WoSIS 2023 SNAPSHOT PROFILES WITH TIMESTAMP AND LANDSAT 7 AND 8 COVERAGE BY CONTINENT

Continent	pH	P	N
Africa	1	0	1
Asia	34	26	34
North America	2,708	806	1,615
Oceania	366	2	155
South America	21	3	21

C. Model Validation

Due to the scarcity of African soil data and the lack of timestamps, two approaches are suggested to validate whether our extended model for Europe generalizes for Africa. Both approaches require soil data by continents and therefore we separated the previously processed WoSIS profiles by continent as shown in TABLE XVII.

1) Extended Model for Europe with World Data (without Africa): The idea for the future project group would be to cross validate the extended model for Europe - which is trained with European data only - with data from other continents around the world, once satellite data becomes available. As part of this comparison, the test RMSE values of European data should be compared to the test RMSE values, when the model is run with data from other continents. Our hypothesis is that if the test RMSE values are similar for each continent, then the extended model for Europe generalizes well to other regions, including Africa.

2) Extended Model for Europe with African Soil Data:
Another approach for the future project group could be to wait for African soil data that has accurate timestamps available and use them to further train and tune our existing extended models for Europe. One very promising project to solve this missing soil data gap is the *Soils4Africa* project by the European Commission, which runs until 31/05/2025. At the time of this project, no data have been published yet from the Soils4Africa project [45].

VI. Discussion and Summary

A. Result Discussion

The research demonstrates that splitting the modeling process by training separate models for each target soil parameter is feasible. This approach results in lower training times, reduced model complexity, and improved prediction accuracy with lower error. In contrast, a single model trained on all target values would face additional missing-value challenges, since not all input data are available for every target, which would ultimately reduce overall performance.

Although spatial data are used as input, our investigation of the train, validation, and test splits shows that the models are not highly sensitive to the splitting method. However, spatial cross-validation is considered the preferred approach, as it better accounts for spatial dependencies.

The evaluated and optimized models exhibit very comparable prediction quality across most nutrients. The only noticeable difference occurs for potassium, where the FCNN outperforms the other models.

The expansion of the database with additional features clearly improves model quality, as evidenced by the feature importance maps of the extended input data. Therefore, incorporating more data is generally recommended. Nonetheless, an excessive number of input features can increase model complexity and prolong training times. For example, while Clay embeddings multiply the number of input features, only a few of these embeddings appear to be highly influential. Future improvements might include compressing the embeddings to a lower dimension using an autoencoder or eliminating less important features.

Regarding prediction quality, the achieved RMSE values for all target variables (except pH) are in a similar range as the corresponding means. Although these errors may seem high, they can be partly explained by the strong skew in the data toward lower values rather than a normal distribution. Moreover, the fact that the RMSE values are within the range of the standard deviation indicates that the prediction quality is not yet fully satisfactory. However, the error levels are comparable to those reported in [11], suggesting that the model quality is at least on par with existing references—or possibly that the reference errors are themselves high, as noted by [46]. Despite these challenges, the results justify continued pursuit of this project.

B. Project Summary

In summary, the machine learning methods applied in this project indicate significant potential for predicting soil quality.

Numerous types of data can be used as features for prediction, and while there is room for further model improvement, any extension must be carefully managed to keep the input scope reasonable and the training and inference times feasible.

The next step is to determine what range of errors is tolerable for a realistic prediction, and whether the observed errors are low enough to support crucial fertilization recommendations. If not, targeted model improvements and refined data selection will be necessary based on the insights gained so far.

As discussed in Section V, one of the biggest challenges remains the collection of African soil data with accurate timestamps corresponding to the available satellite data. Without resolving this issue, there will always be a quality gap for soil predictions in Africa, where inexpensive and easily accessible fertilizer recommendations are critically needed.

Further recommendations and ideas for improvements are discussed in the subsequent section.

VII. Outlook

During the research for this project, new ideas and opportunities for further improvement have emerged, which are outlined below.

A. Further Data Improvements

As is inherent in machine learning, the quality of the data exerts a strong influence on the performance of any prediction system. This principle applies to soil prediction in general and, specifically, to the task presented here. In this context, the following suggestions are made:

- One of the most serious challenges for an African prediction model is the absence of soil data with accurate timestamps. This limitation makes it difficult to match soil samples with temporally relevant satellite imagery, and it may account for critical views on such approaches [46]. Addressing this issue would represent a major improvement over the research in [11]. An enhanced database is expected from the Soils4Africa project, which is scheduled for completion in 2025 [45].
- The time interval between soil sample collection and the acquisition of the corresponding satellite image—especially for surface soil properties that are highly sensitive to short-term weather effects such as rain—should be minimized to improve prediction quality.
- The LUCAS soil data, particularly for phosphorus, contain many values below the limit of detection. Employing a random imputation strategy within the range from zero to the detection limit, rather than using a constant value, might further enhance prediction accuracy.
- The team responsible for the open weather data source used in this study has expressed interest in the research and in pursuing further projects. This opens up the possibility of gaining access to additional or more detailed weather data.
- Performing spatial cross-validation using clusters determined by, for example, a k-nearest-neighbor algorithm

rather than the current grid-based clustering might improve prediction performance by better capturing spatial dependencies.

- Numerous features have been extracted from satellite data for other applications (e.g., NDVI [47]). These features, or the algorithms used to derive them, could potentially be leveraged for soil prediction.

- Additional and more recent data sources—such as satellite data from future measurement campaigns like LSTM on temperature (Land Surface Temperature Monitoring, scheduled to start in 2029, [48]) and CIMR on climate change (Copernicus Imaging Microwave Radiation, scheduled to start in 2028, [49]) as recommended in [11]—could be incorporated.

- The amount of sunlight during satellite image acquisition significantly influences result quality. Therefore, an algorithm that filters out satellite images captured under low sunlight conditions would be beneficial. Low sunlight conditions might occur, for example, from September to March in northern European regions or in small valleys.

- To extend the range of input data, the use of local plant images for plant health estimation—as recommended in [2]—could be explored. For instance, users of a future app could directly upload images of their own plants to the prediction system, providing an additional, highly localized input feature.

- Additional information provided by users, such as details on crop rotation, previous fertilization practices, and the application of artificial irrigation, may further improve predictions.

- Crop rotation history might also be inferred from satellite data, as suggested in [2].

- The current world yield data could be further refined by accounting for the use of artificial irrigation.

- Open-source satellite data with higher resolution (up to 30 cm per pixel) are currently lacking; however, such data could lead to additional improvements.

- Higher-resolution satellite data could also be utilized in the clay model to obtain improved embeddings as input.

- Embeddings could be generated as a separate preprocessing step using autoencoders [50], particularly for neighboring pixels in satellite data. Additionally, autoencoders could be used to reduce the dimensionality of the clay embeddings, resulting in a more proportional representation of the input data and potentially leading to improved performance and reduced training times.

- The FAO GAEZ portal's "Theme 5: Actual Yields and Production" dataset may be supplemented by data from "Theme 6: Yield and Production Gaps," which provide estimates of how much additional yield could be achieved under optimal conditions. Initial experiments suggest that these yield gap figures can further improve nutrient predictability. However, it is important to confirm the independence of the Theme 6 dataset from ground-truth observations; if it already incorporates actual soil measurements, it may inflate model performance without providing genuinely new information.

B. Possible Machine Learning Enhancements

The analysis of feature importance presented for the XGBoost model reveals that, due to the nonlinearity and ambiguous behavior inherent in tree-based models, these importance results can vary significantly with different data and training runs. This variability is partially reflected in the plots shown earlier. Consequently, a more sophisticated and less model-biased method for determining feature importance is desirable. The use of permutation feature importance [51] is recommended, as it can provide a more robust assessment. With this measure, reducing the number of input features in the extended model becomes conceivable, which would not only speed up training times but also facilitate future predictions on handheld devices.

To realize a soil prediction app, the implementation of an ML-Ops strategy [52] is advisable. Although ML-Ops was not implemented in the current project due to the lack of database updates, future stages of app development—especially upon rollout—should include tracking and monitoring of datasets, deployments, and additional processes. Accordingly, the use of MLflow [53] as a framework in combination with Python is suggested. As an alternative, albeit not open source, Neptune [54] is also recommended.

C. Further Topics

Beyond the aforementioned ideas for improvements in data and models, broader topics must also be considered for the successful implementation of a fertilization recommendation system. In particular, the method by which information regarding fertilization needs is communicated to farmers is crucial. As stated by [2], "The real problem is understanding the right path to the best nutrient management recommendations and making them accessible and understandable to farmers."

DISCLAIMER

Throughout the development of this report, the project team utilized ChatGPT as a supplementary tool to streamline tasks such as technical description drafts and text refinement. Final responsibility for the research design, analysis, and conclusions remains entirely with the authors. All content generated by ChatGPT underwent careful review, validation, and, where necessary, revision to ensure it met academic standards and reflected the team's original work.

REFERENCES

[1] Raymond Weil and Nyle Brady. *The Nature and Properties of Soils. 15th edition*. Pearson and Prentice Hall, 2017.

[2] Oumnia Ennaji, Leonardus Vergütz, and Achraf El Allali. Machine learning in nutrient management: A review. *Artificial Intelligence in Agriculture*, 9:1–11, 2023.

[3] Abderahman Rejeb, Karim Rejeb, Suhaiza Zailani, John G. Keogh, and Andrea Appolloni. Examining the interplay between artificial intelligence and the agri-food industry. *Artificial Intelligence in Agriculture*, 6:111–128, 2022.

[4] Gezahagn Kudama, Mabiratu Dangia, Hika Wana, and Bona Tadese. Will digital solution transform sub-sahara african agriculture? *Artificial Intelligence in Agriculture*, 5:292–300, 2021.

[5] European Space Agency. Sentinels toolboxes - open source development. https://sentiwiki.copernicus.eu/web/sentinels-toolboxesl.

[6] United States Geological Survey. Landsat 7. https://www.usgs.gov/landsat-missions/landsat-7.

[7] United States Geological Survey. Landsat 8. https://www.usgs.gov/landsat-missions/landsat-8.

[8] BigEarthNet Project. Bigearthnet - a large-scale sentinel benchmark archive. https://bigearth.net/#download.

[9] Kai Norman Clasen, Leonard Hackel, Tom Burgert, Gencer Sumbul, Begüm Demir, and Volker Markl. reben: Refined bigearthnet dataset for remote sensing image analysis. *arXiv*, 2024.

[10] BigEarthNet Project. Bifold bigearthnet v2.0. https://huggingface.co/BIFOLD-BigEarthNetv2-0.

[11] T. Hengl, M.A.E. Miller, J. Križan, and et al. African soil properties and nutrients mapped at 30 m spatial resolution using two-scale ensemble machine learning. *Nature - Scientific Reports 11*, 27(6130), 2021.

[12] Fernandez Ugalde O, Scarpa S, Orgiazzi A, Panagos P, Van Liedekerke M, Maréchal A, and Jones A. Lucas 2018 soil module. https://publications.jrc.ec.europa.eu/repository/handle/JRC129926, 2022.

[13] N. H. Batjes, L. Calisto, and L. M. de Sousa. Providing quality-assessed and standardised soil data to support global mapping and modelling (wosis snapshot 2023). *Earth System Science Data*, 16(10):4735–4765, 2024.

[14] Tor-Gunnar Vågen, Leigh Ann Winowiecki, Luseged Desta, Ebagnerin Jérôme Tondoh, Elvis Weullow, Keith Shepherd, and Andrew Sila. Mid-Infrared Spectra (MIRS) from ICRAF Soil and Plant Spectroscopy Laboratory: Africa Soil Information Service (AfSIS) Phase I 2009-2013. https://doi.org/10.34725/DVN/QXCWP1, 2020.

[15] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794. ACM, August 2016.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[17] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001.

[18] L. Calisto, L.M. de Sousa, and Batjes N.H. Standardised soil profile data for the world (wosis snapshot – december 2023). https://doi.org/10.17027/isric-wdcsoils-20231130, 2023.

[19] David Roberts, Volker Bahn, Simone Ciuti, Mark Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, Jose Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, David Warton, Brendan Wintle, Florian Hartig, and Carsten Dormann. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40, 12 2016.

[20] Kai Sun, Yingjie Hu, Gaurish Lakhanpal, and Ryan Zhenqi Zhou. *Spatial Cross-Validation for GeoAI*, chapter 14, page 14. CRC Press, 1st edition edition, 2023. eBook.

[21] Python Software Foundation. Python programming language. https://www.python.org/.

[22] Numpy Team. Numpy, the fundamental package for scientific computing with python. https://numpy.org/.

[23] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, sep 2020.

[24] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[25] SciPy Team. Scipy - fundamental algorithms for scientific computing in python. https://scipy.org/.

[26] Ryoki Hamano, Shota Saito, Masahiro Nomura, and Shinichi Shirakawa. Cma-es with margin: lower-bounding marginal probability for mixed-integer black-box optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, page 639–647, New York, NY, USA, 2022. Association for Computing Machinery.

[27] James Bergstra, Remi Bardenet, Yoshua Bengio, and Balazs Kegl. Algorithms for hyper-parameter optimization. In *Proceedings of NEURIPS*, 2011.

[28] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[29] Inc. Preferred Networks. Optuna open source hyperparameter optimization framework to automate hyperparameter searc. https://optuna.org/.

[30] Plotly Technologies Inc. Random forests. https://plotly.com/.

[31] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[32] The pandas development team. pandas-dev/pandas: Pandas. *Zenodo*, February 2020.

[33] Python Software Foundation. Pandas - python data analysis library. https://pandas.pydata.org/.

[34] The PyTorch Foundation. Pytorch. https://pytorch.org/.

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

[36] scikit-learn Team. scikit-leanr, machine learning in python. https://scikit-learn.org/stable.

[37] XGBoost Developers. Xgboost. https://xgboost.readthedocs.io/en/stable//.

[38] The Matplotlib Development Team. Matplotlib: Visualization with python. https://matplotlib.org/.

[39] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[40] Michael Waskom. seaborn: statistical data visualization. https://seaborn.pydata.org/index.html.

[41] Panos Panagos, Marc Van Liedekerke, Pasquale Borrelli, Julia Köninger, Cristiano Ballabio, Alberto Orgiazzi, Emanuele Lugato, Leonidas Liakos, Javier Hervas, Arwyn Jones, and Luca Montanarella. European soil data centre 2.0: Soil data and knowledge in support of the eu policies. *European Journal of Soil Science*, 73(6):e13315, 2022.

[42] Clay AI for earth. Clay open source foundation model. https://clay-foundation.github.io/model/index.html.

[43] United States Geological Survey. Usgs machine-to-machine api. https://m2m.cr.usgs.gov/api/docs/json/.

[44] Google Earth Engine Team. Usgs landsat 7 level 2. https://developers.google.com/earth-engine/datasets/catalog/.

[45] European Commission. Soils4africa. https://doi.org/10.3030/862900.

[46] A.G.T. Schut and K.E. Giller. Soil-based, field-specific fertilizer recommendations are a pipe-dream. *Geoderma*, 380:114680, 2020.

[47] Freie Universität Berlin. Ndvi - sentinel 2. https://www.geo.fu-berlin.de/en/v/geo-it/gee/2-monitoring-ndvi-nbr/2-2-calculating-indices/ndvi-s2/index.htm.

[48] European Space Agency. Copernicus lst - land surface temperature monitoring. https://d-copernicus.de/daten/satelliten/satelliten-details/news/lstm-land-surface-temperature-monitoring/.

[49] European Space Agency. Copernicus cimr - copernicus imaging microwave radiometry. https://www.d-copernicus.de/daten/satelliten/satelliten-details/news/cimr-copernicus-imaging-microwave-radiometry/.

[50] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[51] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 04 2010.

[52] Yaganteeswarudu Akkem, Saroj Kumar Biswas, and Aruna Varanasi. Smart farming monitoring using ml and mlops. In Aboul Ella Hassanien, Oscar Castillo, Sameer Anand, and Ajay Jaiswal, editors, *International Conference on Innovative Computing and Communications*, pages 665–675, Singapore, 2023. Springer Nature Singapore.

[53] MLflow Project. Build better models and generative ai apps on a unified, end-to-end, open source mlops platform. https://mlflow.org/.

[54] Neptune Labs. Neptune for tracking foundation model training. https://neptune.ai/.