Contributions for Soil Nutrient Estimation Using Earth Observation Data: Datasets, Toolkits and Semi-Supervised Learning

RAFAEL ANTÓNIO FERNANDES E COSTA MARQUES CANDEIAS, Instituto Superior Técnico, Portugal

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

CCS Concepts: • **Software and its engineering → Application specific development environments**.

Additional Key Words and Phrases: Semi-Supervised Learning, Nutrient, Machine-Learning, Earth-Observation-Satellite, Python-Toolkit, Datasets; Spectral-Vegetation-Indices

ACM Reference Format:
Rafael António Fernandes e Costa Marques Candeias. 2024. Contributions for Soil Nutrient Estimation Using Earth Observation Data: Datasets, Toolkits and Semi-Supervised Learning. *ACM Trans. Graph.* 37, 4, Article 111 (August 2024), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Context

Soil is essential for the economy, society, and environment, providing materials, food, and carbon retention[1]. Soil over-exploitation due to increasing demand for food and resources is depleting fertile land [2], and given projections of global population increase of 33% by 2050 [Alexandratos and Bruinsma 2012], soil usage must be taken carefully. With unsustainable practices like excessive fertilizer use, intensive farming deepens these outcomes. Furthermore, Food and Agriculture Organization (FAO) notes that topsoil formation is slow, with significant loss occurring annually[3], which further escalates this issue. Despite such claims, others dispute the severity of the situation, but undeniably defend it as an important topic, as they mention that 6% of soils have less than a century of viability[4].

Soil is also critical for plant growth and carbon sequestration. Rising carbon dioxide levels threaten ecosystems and human livelihoods[5], with projected impacts on weather patterns, species loss, and poverty[6]. Mismanagement of soil exacerbates these threats, leading to erosion, desertification and climate change.

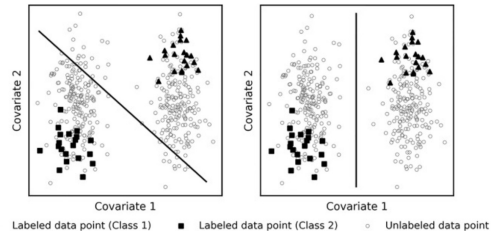

Fig. 1. An illustration of the usefulness of unlabeled data. Left one shows the optimal classification boundary only based on labeled data, right one shows the optimal boundary with considering both labeled and unlabeled data [Zhang et al. 2021]

1.2 Motivation

To protect our soils, it is essential to develop accessible methods for landowners to avoid over-fertilization. Soil fertility is assessed by the concentration of sixteen elements, but this alone is insufficient. Other properties, such as pH levels, also influence nutrient absorption. Conventional soil analysis, though reliable, is complex and requires laboratory access, making it impractical for many farmers. This method, dating back to at least 1970, remains the standard for soil testing, as seen with the *"Bray-P1 and Olsen soil tests"* which is still the most frequent in 2021[Dewis et al. 1970]. Spectral Vegetation Indices (SVI) offer a practical alternative. Derived from infrared wavelengths, SVI can estimate plant and soil properties, which unlike traditional methods, is fast and inexpensive[Bünemann et al. 2018].

Machine Learning (ML) in remote sensing has shown promise for soil nutrient prediction[Trontelj ml and Chambers 2021], but it typically requires extensive labeled data, which is scarce due to the high costs of soil data extraction. Semi-Supervised Learning (SSL) can address this issue by combining labeled and unlabeled data to improve model performance[Zhang et al. 2021].

TerraSenseTK (tstk) was created to streamline the process of using Earth Observation Satellites (EOS) data for ML models, covering everything from data acquisition to model fitting. However, it needed improvements to enhance usability and generality[Manuel Pereira [n. d.]]. Despite the abundance of toolkits, none adequately simplify the extraction and application of EOS data in ML models, highlighting the relevance of tstk for this purpose[Manuel Pereira [n. d.]].

1.3 Objectives & Contributions

This thesis aims to achieve three major objectives:

[1]The World Bank, https://tinyurl.com/vkdvzru7
[2]Overpopulation Environment, https://tinyurl.com/4n2szetn
[3]Soil Degradation, https://tinyurl.com/2p9e9bnf
[4]OWID 60 left, https://tinyurl.com/5xyrveue
[5]Plants climate change, https://tinyurl.com/y3j8dd6d
[6]Global warming, https://tinyurl.com/5cp764bf

Author's Contact Information: Rafael António Fernandes e Costa Marques Candeias, rafaelmcandeias@tecnico.ulisboa.pt, Instituto Superior Técnico, Lisbon, Lisbon, Portugal.

ACM 1557-7368/2024/8-ART111
https://doi.org/10.1145/nnnnnnn.nnnnnnn

(1) **Identify relevant datasets**: Discover and evaluate EOS and soil nutrient datasets suitable for preparing ML models.
(2) **Enhance the tstk toolkit**: Improve its functionality and clarity, integrate SSL methods, and ensure performance is not compromised.
(3) **Compare SSL and Supervised Learning (SL) algorithms**: Evaluate their effectiveness in estimating soil nutrients using EOS data.

To achieve these goals, the investigation is divided into three key steps:

(1) **Discover and evaluate EOS datasets**: Due to the scarcity of large soil nutrient datasets, this step involves assessing public datasets and creating a high-quality one. This will support the Artificial Intelligence (AI) community by providing reliable data for ML research.
(2) **Enhance tstk**: Analyze the toolkit's strengths and weaknesses, implement necessary improvements, and integrate ML models. This ensures the toolkit remains accessible and useful for the community. As noted in a NASA ML workshop, there is a limited number of open-source tools for developing and evaluating ML models [Manuel Pereira [n. d.]].
(3) **Compare SSL and SL algorithms**: Implement various algorithms in tstk and conduct tests to determine which models best predict soil nutrients from EOS data. This research aims to validate whether SSL methods outperform SL models in data-scarce environments, contributing significantly to the AI community.

By addressing these objectives, this thesis not only advances the functionality of tstk but also pioneers research in using SSL for soil nutrient estimation with EOS data.

2 RELATED WORKS

2.1 Datasets

2.1.1 AfSis Dataset. The Africa Soil information Service (AfSis) dataset aims to provide detailed soil nutrient maps for sub-Saharan Africa to enhance agricultural productivity. It aggregates data from eight organizations, totaling 105,074 labeled data points, and uses random forest and gradient boosting models for predictions [Hengl et al. 2017]. Despite limitations such as 250m resolution and spatial clustering, a 2021 update improved predictions using the Sentinel-2 (S2) satellite with a 30m resolution [Hengl et al. 2021]. However, the dataset should only be used for sub-Saharan Africa due to underrepresentation of other land types.

2.1.2 Big Earth Net Dataset. The Big Earth Net (BEN) dataset consists of 590,326 image patches from the S2 satellite, covering various regions and land cover types, to support land cover classification and land use analysis [Sumbul 2021]. Developed by the Remote Sensing Image Analysis (RSiM) and Database Systems and Information Management (DIMA) groups at Technische Universität Berlin (TU-B), it includes multi-label annotations from the CORINE Land Cover database of 2018 [Sumbul et al. 2019]. Despite some inconsistent labeling and potential cloud cover issues, it is a valuable resource for remote sensing research.

2.1.3 National Cooperative Soil Survey Dataset. The National Cooperative Soil Survey (NCSS) Soil Characterization Database, maintained by the Kellogg Soil Survey Laboratory, provides comprehensive soil data across the United States [Reinsch et al. 2010]. It supports various applications, such as agriculture and environmental management. A model trained on this dataset predicted soil bulk density with good accuracy [Ramcharan et al. 2017]. Despite potential data collection inaccuracies and large-scale variations, NCSS remains invaluable for soil-related studies.

2.1.4 LUCAS Topsoil & LUCAS Copernicus. Land Use and Coverage Area frame Survey (LUCAS) Topsoil [Orgiazzi et al. 2018] collects soil properties across the European Union (EU), with 18,984 locations surveyed in 2018. LUCAS Copernicus [d'Andrimont et al. 2021] focuses on EOS applications, surveying land cover around each point, covering 63,364 points and 26 land cover classes. Despite potential data quality variations, these datasets are crucial for land management and ML training for nutrient prediction.

2.2 Semi-Supervised Learning

2.2.1 SSL research. No studies have applied SSL models to predict soil nutrient quantities, likely due to the computational demands of Deep Learning (DL) models and dataset scarcity. Most research focuses on SL models like Support Vector Machine (SVM), Random Forest (RF), and Multiple Linear Regression (MLR). SSL in other fields shows promise: [Han et al. 2018] achieved 92% accuracy in land use classification with limited labels, and [Staccone 2020] improved sea-ice classification accuracy by 5% using SSL. These studies suggest SSL excels with limited labeled data.

2.2.2 State of the Art. [Zhang et al. 2021] compared SL and Self-training models for soil classification using MLR, K-Nearest Neighbours (KNN), and RF. Despite SSL's higher performance, accuracies were low (below 0.6) due to limited data diversification. Another study [Trontelj ml and Chambers 2021] used SL models to predict soil nutrients from UV-VIS electromagnetic waves. With precision above 80% for certain models, the study highlighted the benefit of incorporating additional soil components and using Principal Component Analysis (PCA) for better prediction.

2.3 Libraries & Toolkits

2.3.1 TerraSenseTK. tstk structure follows the figure above, being made out of 4 Python models: Dataset, Algorithms, Performance and Experiment. Firstly, Dataset was built to download EOS, and to represent the data created according to the user's specification. Secondly, the Algorithms module wraps all Python files with classes representing a predictive ML model, or some other useful ML algorithm. The Performance module is used to encapsulate a series of classes useful for estimating a model's performance and finally, the Experiment module is *"responsible for the execution of an (...) experiment*[Manuel Pereira [n. d.]] from which users can select models and other algorithms to include in the workflow.

2.3.2 Sickit-learn & others. Scikit-learn (sklearn) is a popular open-source ML library for Python, providing efficient tools for data analysis and modeling. It is widely used for classification and is well-documented, making it a favored choice among experienced

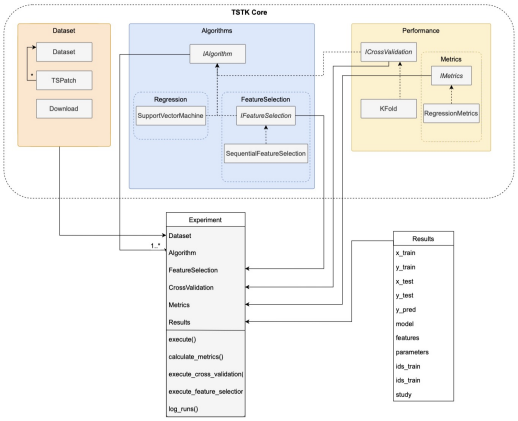

Fig. 2. Component diagram of the original version of TerraSenseTK [Manuel Pereira [n. d.]].

ML practitioners. It serves as a solid foundation for building and experimenting with ML models. Two other toolkits examined for implementing SSL models into tstk are Unified Semi-Supervised Learning Benchmark (USB) [Wang et al. 2022] and LAMDA-SSL (LAMDA) [Jia et al. 2022].

USB is a framework incorporating 14 state-of-the-art DL SSL classifiers, as mentioned in [Van Engelen and Hoos 2020], [Ouali et al. 2020], and [Yang et al. 2022]. It is an optimal approach as it significantly reduces training costs compared to TorchSSL, from 279 GPU days to 39 GPU days [Wang et al. 2022].

LAMDA is an extensive open-source Python toolkit specifically for SSL, including both statistical and DL algorithms for regression and classification. It offers 30 unique models, including those in USB, and a Data module for data management and transformation. According to the authors, it *"outperforms USB in various aspects, including the number of algorithms, data types, task types, and functionality"* [Jia et al. 2022].

Both toolkits provide potential sources of optimized software for tstk to expand its range of SSL models. However, USB's interface may not align with tstk's structure, and LAMDA lacks validation from external sources.

3 DATASET EXPLORATION

3.1 Dataset 1: AfSis

3.1.1 Introduction & Potential Applications. AfSis is a dataset with 105.074 soil samples with *ppm* values for nearly all of the sixteen core nutrients required for soil fertility collected from several locations spread through sub-Saharan Africa. Soil nutrient values are scarce and even though SSL makes use of X_u, it must contain a sufficient amount of X_l to learn the data distribution. So, it seems clear that ignoring such an extensive amount of labeled data is erroneous.

3.1.2 Results & Discussion. Even though it outshines many other possibilities for its large amount of X_l, it demonstrates a lack of reliance on its own prediction by having a R-squared average value of 0.728[Hengl et al. 2021], and only three of the sixteen nutrients have a R-squared value above 0.8[Hengl et al. 2021]. Moreover, even when discarding its performance values, the fact that it is composed of synthetically generated data provides little confidence, and also lacks on the time of nutrient prediction, which must be given to extract EOS data.

3.2 Dataset 2: BEN

3.2.1 Introduction & Potential Applications. As previously mentioned, BEN is a vast collection of 590.326 image patches from the S2 satellite labeled with land cover classes. Spread through ten European countries, it offers a vast amount of EOS data that can be easily used as X_u.

3.2.2 Results & Discussion. All sea, snow, and cloud image patches are removed from this dataset, totaling around 394.330 data points that can be used as X_u. However, one must take into consideration the dimensions each EOS patch has ($1.2 \times 1.2km$), which covers an area of $1.44km^2$. Such information is crucial since it is unrealistic to assume that all that area contains the same amount of nutrient values, and so, to be used, one must split each image patch into a $50 \times 50m$ square. Unfortunately, BEN only associates each land cover class to the full image patch (justifying the number of multi-labels one image patch can have), becoming impossible to identify through BEN what land cover class must be associated with the filtered $50 \times 50m$ square. Thus, one can conclude that even though BEN seems highly promising and can indeed be utilized, it should be temporarily discarded to focus on the main task.

3.3 Dataset 3: NCSS

3.3.1 Introduction & Potential Applications. NCSS has a huge potential to be used as X_l for the enormous quantity of accurate soil data that the authors mention to contain: *"500,000 hectares have been identified and mapped in the United States"*[Reinsch et al. 2010], and as stated before, it is a hard characteristic to come across. Therefore, the same principle from the prior explored datasets applies to NCSS: it must be considered for being a rare source of detailed soil data.

3.3.2 Results & Discussion. The first immediate conclusion that we can take is the enormous reduction of data. NCSS started as a promising large dataset full of nutrient values, perfectly suitable to be X_l, and even though all filtering procedures were reasonably justified, it still went from 417.652 data points, to 32.583, and eventually just 5.083. Such reduction, around 98.78% to be exact, diminishes the dataset's potential and more so its confidence, when mentioning the incomprehensible large amount of missing value percentages. Secondly, not all 5.083 soil locations are usable for X_l, as only a small percentage can be associated to EOS. In fact, true X_l potential only comes from the 1.107 Landsat 1-5 MSS (LMSS) data patches, or the 1.018 extracted by Landsat 4-5 TM (LTM), being the other two too small for ML creation.

3.4 Dataset 4: LUCAS

3.4.1 Introduction & Potential Applications. The LUCAS Topsoil potential comes from it containing soil nutrient data. On the other hand, LUCAS Copernicus contains a high detailed land cover/usage label, which has been proven to increase performance when filtered [Manuel Pereira [n. d.]]. Moreover, it associates these properties to

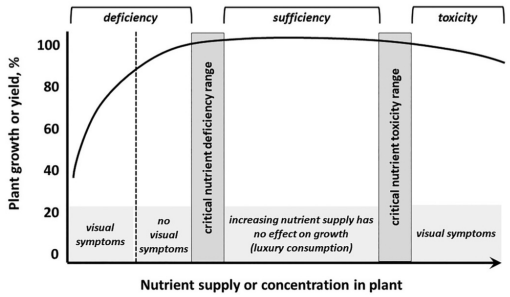

Fig. 3. *"Relationship between plant nutrient concentration and plant growth. The critical nutrient deficiency range represents the nutrient concentration below which nutrients should be added; however, nutrients added beyond this level increases plant nutrient concentration without a response in plant growth or yield."*[Havlin 2020]

an area, which smoothens the EOS data extraction and increases precision. Allied to this, being from the same organization, conducted during the same time periods, and mostly executed on the same sites, they can be merged to form a LUCAS dataset that maps land cover/usage data to nutrient values, and fits perfectly as X_1. Moreover, as mentioned before, these datasets purposely collect data from locations where S2 satellites capture image patches, and so one can hope that most of their data can be mapped to EOS data (unlike NCSS). So, having a considerable amount of 18.984 soil nutrients, an additional land cover and usage class, association with EOS data and an area delimitation, these datasets strike with a strong probability to become X_1.

3.4.2 Results & Discussion. LUCAS dataset was considered to be stable, where most of its data is compacted in an interval of values. Moreover, solemnly 2.846 data points were unsuccessfully associated with EOS data, leaving an interesting amount of 16.130 data points ready to build the ML models. However, even though Nitrogen (N) and Potassium (K) data are ultimately present, it is true that many Phosphorus (P) values are absent, but since our ML models are SSL and utilize unlabeled data, such absence does not raise an alarm. Thus, this dataset strikes as an improved version of NCSS. Furthermore, it presents a satisfactory amount of data, and when discarding all lands that are unrelated to soils, them being classified with "LC0_Desc" values "Artificial land", "Wetlands" and "Water", it still outputs 16.033 data points. Therefore, even if it is a small amount of data, this dataset offers the possibility to test the conclusions made by [Manuel Pereira [n. d.]] that models preform better when trained on crop filtered land and more importantly, to evaluate the SSL expected gain in these conditions.

3.5 From Regression to Classification

According to a book on soil fertility and nutrient management[Havlin 2020], crop yield depends on nutrient concentration ranges, not precise amounts. Within certain ranges, increasing nutrient supply has no effect on growth. Thus, nutrient concentrations can be categorized into deficiency, sufficiency, and toxicity.

Given the goal of identifying fertile soil using EOS data, soil can be classified as fertile or infertile, with infertility covering deficiency and toxicity. The conversion process involves determining nutrient concentration ranges that support plant growth. Research shows these ranges vary by plant species. For example, critical Olsen-P values differ for maize, wheat, and rice. General critical values for nutrients are N: 10-50 mg/kg, P: 10.9-21.4 mg/kg, and K: 40-80 ppm . However, for specific crops like *Triticum spelta*, these values can differ significantly. To avoid discarding data, two conversion methods are proposed: one using general critical points and another using values specific to *Triticum spelta*. However, the fertility concentrations revealed incoherent, and so we make use of general quantities. Despite the variability and challenges in general classification, this approach allows for better utilization of the dataset while accounting for species-specific nutrient requirements.

3.6 Final Dataset

The final dataset derived from the LUCAS dataset, excluding rows labeled as "Artificial land", "Wetland" or "Water." It comprises 16,033 data points and was converted to classification. The NCSS data was excluded due to its small size (1,018 data points) and incompatibility of satellite resolution compared to the S2 satellites used for LUCAS. Therefore, dataset refers to the adjusted LUCAS data.

Nutrient	Count	infertile%	fertile%
N	16.021	95.51	4.49
P	11.740	66.20	33.78
K	15.999	84.46	15.54

Table 1. Final dataset's N P K balance according to general classification.

Nutrient	Count	infertile%	fertile%
N	1.350	99.78	0.22
P	1.218	74.47	25.53
K	1.350	94.96	5.04

Table 2. Final dataset composed only of wheat N P K balance according to general classification.

The dataset reveals significant imbalances in nutrient distribution, particularly for N and K. This imbalance suggests the need to exclude N and K from further analysis due to their potential negative impact on classifier performance.

4 TERRASENSETK: TOOLKIT IMPROVEMENTS

4.1 Introduction

tstk is a python toolkit built to facilitate ML computation over EOS data. It combines all major steps required to conduct these types of experiments, by gathering all band data for each pre-defined location of interest, encapsulating them in a single optimized Dataset object, feeding it to a ML model of choice, and then computing evaluations. It is made of four key modules: 1) dataset, 2) algorithms, 3) performance, and 4) experiment, and 5) results. Each component was updated in this new version, and the modifications can be found online [7].

[7] tstk code, https://tinyurl.com/2kr8djb8

4.2 Class: Downloader

The Downloader class extracts EOS data from the Sentinel-Hub API[8]. Several issues, such as lack of generalization and resource inefficiencies, prompted a restructure. Following this, the Downloader class became abstract, with five sub-classes for different satellites. These sub classes specify attributes like name, collection variable, and available bands. Five GeoDataFrame variables were merged into one (dataset). Area association was streamlined. Instead of requiring get_bbox_with_data() and get_bbox(), the constructor now uses shapely.loads() to create shapely.Polygon directly from the file given as source of data. For rows with missing areas, calculate_area() was created to transform geometries from shapely.Point() to shapely.Polygon() objects, enabling band data extraction. This method uses WGS84 to UTM conversion and shapely.buffer() for calculations. A function was added for users to visualize polygons used for data extraction from Sentinel-Hub API, enhancing usability. The download_images() method was optimized. Initially, it used five EOTasks to create EOPatch objects, which was inefficient and lacked generalization. The new method saves bbox in EOPatch, uses a modified SaveTask() to skip patches with no band data, and avoids creating GeoDataFrames in the main loop. Only specified soil properties (keep) are saved, and users can set resolution, bands, cloud coverage, and time intervals. EOPatch objects now have a band name-channel index mapping, and are saved based on user-specified identification values.

4.3 Class: TSPatch

The TSPatch class simplifies EOPatch representation and manipulation, implementing useful functions. Due to changes in EOPatch creation, some attributes were modified. The patch attribute was renamed to EOPatch to clarify it holds an EOPatch. Added attributes include id for the identification string, location for geographical recognition, bands for band-channel mapping, and indices for index names. These are assigned in the constructor. Methods were updated due to new attributes and EOPatch changes. Methods like get_indices() were added to execute low-level tasks of returning these new attributes. get_masked_region_values() originally returned a matrix of 0's and 1's for valid band pixels. Since the VectorToRasterTask() feature was removed, band data validity changed. If no filter is present, it returns a matrix full of 1's. The represent_image() method displays EOS data in Red-Green-Blue (RGB). It retained its purpose with a light code restructure. EOPatch band data can have multiple EOS per temporal interval. Thus, _get_index_nearest_to_collection_date() was extended to validate band data. This method now returns the valid EOS data closest to the ground-truth extraction date, raising an error if all possibilities are invalid.

4.4 Class: Dataset

The Dataset class in tstk represents an aggregation of multiple TSPatches and facilitates the execution of key functions on the entire EOS dataset rather than on individual components. Originally, the class had four attributes: eopatches_folder, _eopatches, df_eopatches, and index_dic. These were renamed and enhanced for clarity and functionality as such: eopatches_folder became eops_path, storing the directory path of all EOPatches. _eopatches became TSPatches, holding a collection of TSPatches created from each EOPatch in the folder. df_eopatches, initially a pandas.DataFrame of EOPatch locations, was renamed to gth_df and reprogrammed to include all non-EOS data (identification, geometry, date, and ground-truth) from each TSPatch. index_dic, a dictionary mapping index names to calculation formulas, was replaced by a list of lambda functions and renamed indices. Additional attributes were added for optimization: lock, which allows parallel TSPatch creation, bands that aggregate band mappings from TSPatches, and groundtruth_f which provide quick access to soil properties in TSPatches. Five getter functions for new and existing attributes. add_indices(), updating the previous add_index(). remove_indices(), reflecting the change from index_dic to a list. show(), enhancing toolkit utility. add_index() previously incremented index_dic with a string formula, limited to simple band calculations. Its implementation lacked visible outcomes and error handling. Accepts a dictionary mapping index names to lambda formulas. Notifies users of mathematical errors and avoids invalid TSPatches. The new version of add_indices() prevents errors by adding a small random number and ignoring fully invalid data. The new show() method displays the geographical shapes and locations of all TSPatch geometries in the Dataset, similar to show_geometries() in the Downloader class.

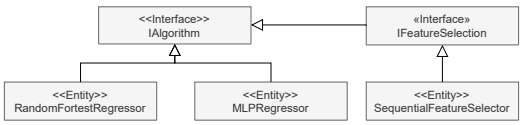

Fig. 4. Algorithms module before restructuring. (Shortened representation.

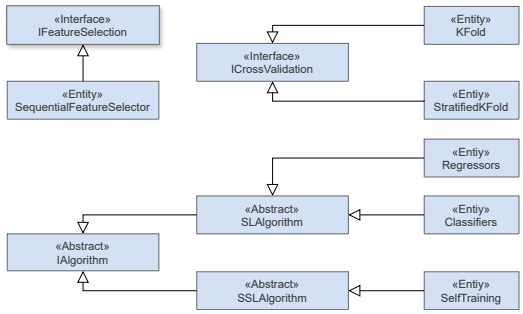

Fig. 5. Algorithms module after restructuring. (Classifiers and Regressors represent several implemented models of each type and are not actual classes in the toolkit).

4.5 Module: Algorithms

The Algorithms module underwent significant changes, particularly in its core structure and the introduction of semi-supervised learning (SSL) models.

The main structural change involved transforming IAlgorithm from an interface representing algorithms requiring training and testing into an abstract class that generalizes any predictive ML

[8]Sentinel-Hub, https://www.sentinel-hub.com

model. This abstraction simplified implementation of new algorithms and representation of existing ones. Consequently, IFeatureSelection was disconnected from IAlgorithm. Additionally, SLAlgorithm and SSLAlgorithm abstract classes were introduced to encompass all supervised and semi-supervised learning models. Lastly, class ICrossValidation was relocated from the Performance module for structural coherence. Before modifications, tstk lacked classifiers and SSL models. Given scikitlearn's reliability and USB and LAMDA limitations, scikitlearn was chosen as the source for SSL code. Six supervised classifiers, a SSL classification class (SelfTraining), and cross-validation (StratifiedKFold) were added.

4.6 Class: Results

The Result class isn't new to tstk; it was already present in the toolkit. Originally located in a utility directory alongside other unrelated classes/files, it aggregated all information regarding a model's result and evaluation. From our perspective, it made more sense for this class to reside within the Performance module. This is because it serves as the object from which all evaluations are conducted and contains the model's result.

In addition to its relocation, the class underwent minor modifications. Attributes crucial for measuring the model's performance, especially those required for classification measurements, were added. The private method __repr__() was altered to provide a better representation of the object's state. Furthermore, an auxiliary method conf() was included, capable of calculating the confusion matrix for Result objects.

4.7 Class: IMetrics

In our second transformation, the IMetrics interface was refactored into an abstract class because both regression and classification classes utilize its implemented methods. This conversion eliminates code duplication and prevents the creation of a measurement class without specifying the model's nature, while retaining the original purpose of the interface. Additionally, a new class, ClassifierMetrics, was introduced to specify the evaluation of classifiers. This class mirrors the structure and logic of the original RegressionMetrics class, which remained unaltered. It comprises methods, such as cmd_acc(), cmd_prec(), and cmd_ae(), each utilizing sklearn functions tailored to calculate specific measurements like accuracy, precision, and absolute error, respectively. These IMetrics subclasses are not limited to a single Result object; hence, one IMetric object can measure several Result instances.

4.8 Class: Experiment

The Experiment class enables users to conduct experiments by evaluating multiple tstk models on a specific dataset, defined by a Parser object. This class offers users a simple yet versatile way to test their ML models by customizing the Experiment object's parameters. However, the class initially contained several unnecessary attributes and methods, complicating its usage. Therefore, structural and logical modifications were necessary to streamline its functionality and improve its efficiency.

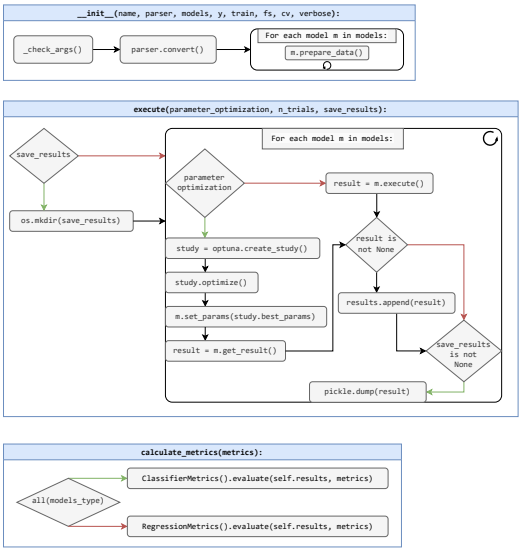

Fig. 6. Experiment flow graph after restructuring.

To reduce complexity, unnecessary attributes and methods were removed, resulting in a significant reduction in the number of attributes from 21 to seven. The remaining attributes, such as name, parser, fs, models, verbose, models_type, and results, were deemed essential for the class's functionality. Similarly, most methods were either removed or modified, resulting in one private and two public methods: _check_args(), execute(), and calculate_metrics().

The restructuring of the class's logical flow aimed to optimize the execution process. The previous approach lacked efficiency, prompting the introduction of a sequential execution process for each model. Models now manipulate their data upon construction, and users have the option to save Result objects in a pickle file. Additionally, each model only saves its best result, enhancing overall efficiency.

4.9 Summary

In conclusion, tstk underwent extensive modifications to expand its scope, functionality, and utility while reducing complexity. These alterations included attribute changes, optimization of logical flows, and the introduction of several new features. The resulting restructuring is depicted in the final diagram (see 7, illustrating the toolkit's enhanced structure and capabilities.

5 SEMI-SUPERVISED LEARNING: CASE STUDY SPECIFICATION

5.1 Experiments Description

Our goal is to have a general assessment of SSL models when predicting soil nutrient data, and compare them to their SL versions. This comparison was performed by evaluating SL models and their Self-training versions on both `dataset` and `dataset_wheat`. To achieve a broad performance measurement, we created 3 core experiments:

(1) Using all twelve bands plus twenty-eight SVI.

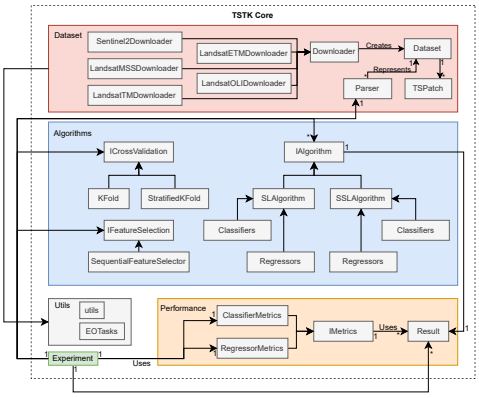

Fig. 7. tstk core after restructuring.

(2) Utilizing features proven to yield adequate results for predicting P values.
(3) Using features selected by feature selection algorithms.

The first experiment serves as a baseline, the second uses features proven by literature, and the third employs features selected by *Pearson correlation*, PCA, and Recursive Feature Elimination (RFE). For the second experiment, fifteen SVI and eleven bands were chosen based on literature [Richardson and Everitt 1992], [K. Kawamura et al. [n. d.]], [Lin et al. 2015], [Özyiğit and Bilgen 2013], and [Maleki et al. 2006]. The third experiment's features will be detailed in the following section.

To address dataset imbalance, each experiment is also executed with undersampled data, equalizing 'infertile' and 'fertile' records. This reduces 'infertile' records from 7,772 to 3,968, resulting in 7,936 data points with equal classification distribution. Additionally, six experiments with varying percentages of unlabeled data (5%, 25%, 50%, 75%, 100%) are conducted on the undersampled dataset to study the impact of unlabeled data on SSL models.

6 SEMI-SUPERVISED LEARNING: RESULTS & DISCUSSION

6.1 Data Pre-Processing

6.1.1 Data Distribution. Almost no bands contain invalid data. Among 16,033 data points, `SWIR9` had the most invalid data, with only 316 points, representing 1.97%. Since all SVI are derived from band values, they likely share this validity. Indeed, upon investigation, SVI also showed coherent values. Out of ten SVI, only three had erroneous data points, with `NDRE` having the highest at 13 `EOPatches`. While a small amount of invalid values might raise concerns, it is common for remote sensing sensors, such as S2, to read incoherent data due to reflections on bright areas like snow, which can have values higher than 1.0. Given that `dataset` has minimal inconsistencies, we can confidently assert its high quality as a source of EOS. This guarantees the reliability of our experiments and the robustness of our python toolkit.

6.1.2 Outliers. Band `Ultra-blue` has the highest percentage of outliers, reaching 8.39% (1.345 data points). Other bands have outlier percentages around 7%, 5%, 3%, and 2%. Interestingly, SVI features have fewer outliers, with `CCCI` being the highest at 4.61% and all others below 3%.

Outliers are data points far from the rest of the dataset's distribution and their definition is relative. In remote sensing, outliers can result from diverse land covers like snow regions or dense forests, which are correctly measured but appear as outliers due to their rarity in the dataset. Given `dataset`'s coverage of various European regions with differing land covers and sun inclinations, these outliers don't indicate unreliability.

Restricting the dataset to the Iberian peninsula, we find 4039 data points, with `Ultra-blue` still having the most outliers (234) but the percentage dropping to 5.79%. In `dataset_wheat` from the Iberian peninsula, outliers in `Ultra-blue` drop to 5.9% (16 data points).

6.1.3 Feature Selection. To analyze our data, we utilized a combination of feature selection models to identify the most accurate predictors for P, followed by the *Pearson* method to remove features sharing similar information. This process involved intercepting features selected by PCA and RFE, then eliminating features with a *Pearson* correlation above 0.9. The resulting set of optimal features was used for experiment 3.

First, PCA identified bands `Ultra-blue`, `SWIR9`, and `VNIR8`, and SVI `GRNDVI`, `RSI(1385, 1705)`, `CVI`, `FE2`, `CCCI`, `CL`, `EVI`, `CIG`, `Chlgreen`, `NDSI(R523, R583)`, `NDSI`, `MCARI`, `GNDVI`, `TSAVI`, `WDRVI`, `NDVI(780, 670)`, and `GSAVI` as the top twenty features. RFE discarded bands `BLUE` and `SWIR12`, and SVI `CVI`, `EVI`, `FE2`, `PVI`, `CIrededge`, `NDSI`, `GRI`, `CCCI`, `CL`, `RSI(1385, 1705)`, `Chlgreen`, `NDSI(R523, R583)`, and `CIG`. By intersecting these sets, we obtained bands `Ultra-blue`, `VNIR8`, `SWIR9`, and SVI `GNDVI`, `GRNDVI`, `GSAVI`, `MCARI`, `NDVI(780, 670)`, `TSAVI`, and `WDRVI` as crucial features for predicting P.

Next, we applied the *Pearson* method to measure correlations among these features to eliminate redundancy. The correlation matrices in 8 revealed high correlations among bands `Ultra-blue`, `Blue`, `Green`, `Red`, and `VNIR5`, as well as among `VNIR7`, `VNIR8`, `VNIR8a`, `VNIR6` with `SWIR9`, and `SWIR11` with `SWIR12`. However, none of the bands in our selected set (`Ultra-blue`, `VNIR8`, `SWIR9`) exceeded a 90% correlation threshold, so all were retained for experiment 3.

Applying the same methodology to SVI features, we found strong correlations between `GRNDVI` and `GNDVI`, `WDRVI`, and `NDVI(780, 670)`, as well as between `WDRVI` and `NDVI(780, 670)`. To avoid redundant computations, we discarded `GRNDVI` and `NDVI(780, 670)` from the interception mentioned before.

6.2 ML Experiments

6.2.1 Performance per sub experiment. Contrary to our expectations, the pre-selection of features did not result in significant performance gains. The highest average balanced accuracy achieved in our three experiments was 0.5318 in Experiment 3 only marginally better than the 0.5309 achieved when considering all twenty-eight features in Experiment 1. Similarly, the f1-score results showed that the highest value of 0.5476 was obtained in Experiment 2, while Experiment 1 was close with a peak f1-score of 0.5321, a difference of less than 2%.

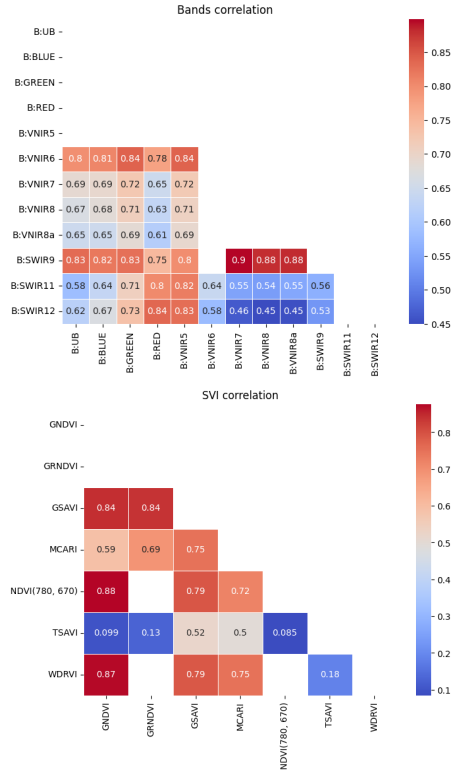

Fig. 8. Both matrices portray the correlation between features present on `dataset`. The left one represents all bands, and the other, for simplification, only shows the values shared between the SVI elected by PCA and RFE. Note: Squares omitted present a correlation higher than 0.9.

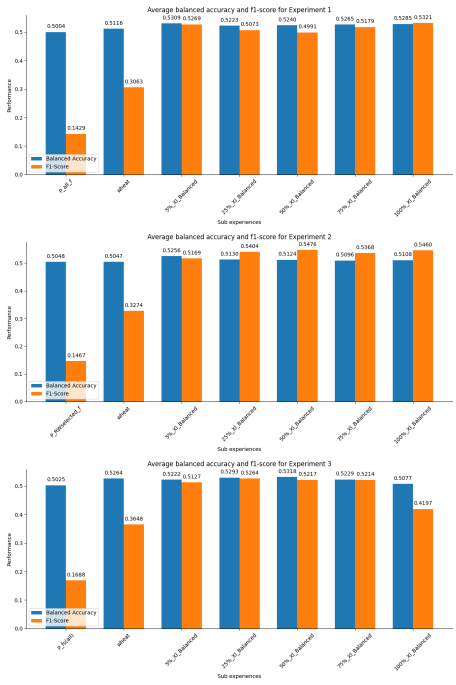

Fig. 9. Balanced accuracy and f1-score average per sub experiment conducted on the experiment 1, 2 and 3. Experiment 1: Highest balanced accuracy achieved with sub experiment 5%_X_l_*Balanced* (0.5309); 100%_X_l_*Balanced* reached the biggest f1-score (0.5321). Experiment 2: Highest balanced accuracy achieved with sub experiment 5%_X_l_*Balanced* (0.5256); 50%_X_l_*Balanced* reached the biggest f1-score (0.5476). Experiment 3: Highest balanced accuracy achieved with sub experiment 50%_X_l_*Balanced* (0.5318); 25%_X_l_*Balanced* reached the biggest f1-score (0.5264). Note: Values were calculated by averaging all models.

Additionally, the bar plots reveal the dependency of f1-scores on the balance state of the dataset. This conclusion is evident from the noticeable improvement in f1-scores when comparing sub experiments with unbalanced datasets ('All-features' and 'wheat') to others. The results also indicate an increase in f1-score when comparing `dataset` to `dataset_wheat`. This observation partially confirms that filtering data by plant species can enhance model performance. Specifically, our models exhibited higher f1-scores with the smaller, more homogeneous `dataset_wheat` (1.218 data points) compared to the larger, more diverse `dataset` (16.131 data points).

6.2.2 SL and SSL Comparison. Experiment 1 showed that three out of five models improved with SSL, with `AdaBoosting` remaining stable and `HGradientBoosting` decreasing by 0.042. In Experiment 2, `AdaBoosting` slightly increased while `KNN` dropped by 0.06. For Experiment 3, only `ComplementNB` and `RandomForest` performed worse with SSL. Out of fifteen comparisons, eight favored SSL and five favored SL.

Overall, SSL tends to yield slightly better outcomes than SL. Notably, `ComplementNB` in Experiments 1 and 2 had significant improvements of 0.259 and 0.1552 when using SSL, indicating that SSL performs better when 75% of data is unlabeled.

6.2.3 Performance by percentage of X_u. To conclude our discussion, we studied the impact that the percentage of unlabeled data has on our SSL models. This behaviour was investigated by drawing the average performance reached with each of the sub experiments that suffered from undersampling and have unlabeled data. They are sub experiments `Undersampled-5%Labeled`, `Undersampled-25%Labeled`, `Undersampled-50%Labeled` and lastly `Undersampled-75%Labeled`. Thus, we created one plot per major experiment, where their X-axis corresponds to the four unique sub experiments mentioned.

From them, we visualize that the three experiments reached the highest performance values when 75% of their data was unlabeled. One could say that SSL models meant to predict P values behave better when only 25% of the dataset is labeled. However, when taking a closer look into the actual values we achieved and observe that these measurements only increase around 2%, such claim cannot be taken for granted.

7 CONCLUSIONS

Concerning the dataset creation, we presented four potential datasets (AfSis, BEN, NCSS and LUCAS) that were discovered and explored to be merged as a final dataset. AfSis lacked reliance, BEN was dispersed plus had missing land cover labels, and NCSS had only a

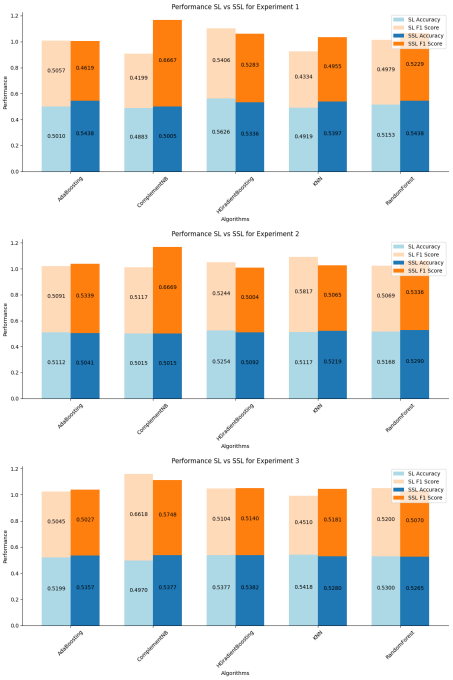

Fig. 10. Balanced accuracy and f1-score comparison between each model and their SSL adaptation. These values only take into account the sub experiment `Undersampled-25%Labeled`.

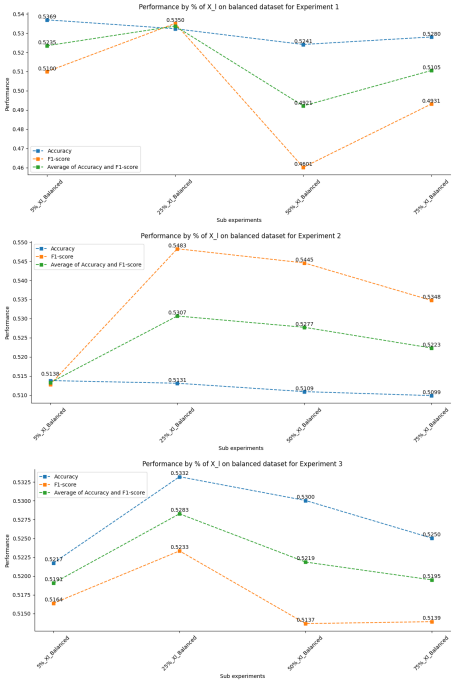

Fig. 11. Average accuracy, average f1-score, and the average of both averages of our models results portrayed on a scatter plot by the percentage of labeled data..

small amount of usable data. LUCAS, instead, was the only dataset that had enough potential and quality to be used as our source of data. It presented land cover and land usage, was built by a reliable organization (European Space Agency (ESA)), and had been previously applied in [Manuel Pereira [n. d.]]. After filtering and transformations, we obtained a classification dataset. Given the underrepresentation of N and K, we solely focused on P for the subsequent experiments, referred to as `dataset`.

From attribute renaming and reduction, restructuring of hierarchy, the addition of useful methods, the inclusion of SSL algorithms, and optimization of code, tstk suffered a complete restructure, which made it more clear, user-friendly, useful, and broad. These updates served as a contribution to the AI for EOS research community as a refined piece of software that allows for easy EOS data extraction and subsequent ML training and prediction.

Concerning our final objective, this study demonstrated a slight SSL superiority to classic SL models when estimating P values on datasets with high amounts of unlabeled data. Moreover, this investigation also confirmed what was theorized in the previous chapters, that performance levels increase when the training and testing data share a unique land cover (`dataset_wheat` gave better results than `dataset`), and also when balanced (all of the `Undersampled` experiments had higher performances than `dataset` core experiment). However, despite SSL demonstrating slightly superior performance, given our limited dataset and marginal performance disparities to their SL counterparts, further research is necessary to confirm our conclusions.

7.1 System Limitations, Challenges and Future Work

The first major limitation that bounded our investigation was the inconsistency found in the soil fertility literature. The papers read disagreed on the nutrient concentrations that make soil fertile. [Havlin 2020] indicated that soil N must be between 40 and 15.000 *ppm*, and P around 2.000 *ppm*, which is completely incoherent with literature [Pattison et al. 2010] [Geisseler and Horwath 2016] [Griffin et al. 1995] and [Thomas 2017], which state that N values should be between 10 to 50 mg/kg, and paper [Bai et al. 2013] which reported that *"optimal crop yield ranged from* 10, 9 mg/kg *to* 21, 4 mg/kg*"* of P.

Furthermore, we learned that soil fertility requires more features besides EOS data, such as Cation Exchange Capacity (CEC), humidity, bulk density and many other variables, which were all disregarded in this research. Soil fertility, which in our research directly translates to P concentration, unveiled as a naturally difficult to predict with ML. With the inclusion of a soil fertility expert, we would be more certain of the features to consider and obtain a precise method to classify soils as fertile or infertile. We failed to encounter any relevant data source with high quality information about all the other twelve elements that contribute to soil fertility, which would be relevant to undergo the same investigation. Finally, our future work would also lay on datasets focused on the same crop and geography, as soils from different regions naturally diverge on their properties, and so, fertility requirements, even with the same land-cover. Furthermore, we know that crop filtering is indeed one of the keys for better performances.

Moving to sources of data, we conclude this thesis with a confident notion that most publicly available data fails to meet our demand. The dataset that we achieved resulted of an intensive deepening of four different sources of data, but still lacked a good representation. Therefore, as future work, we intend to apply DL data augmentation models that generate fake but coherent data from our `dataset_wheat`. To achieve a high quality hybrid dataset, we would also need to add more data related to *Triticum spelta*.

Another limitation worth mentioning is the lack of validation tstk has. Like any other software, it may contain bugs that require fixing, and that could, unbeknownst to us, affect our results. Allied to the fact that the toolkit is complex, we believe to be crucial the addition of a module for unit testing, where we could evaluate the toolkit's performance and so deviate the blame of any suspicious results from tstk.

Due to the complexity of the problem, and the other tasks at hand (dataset creation/validation, plus tstk improvement) we directed our research to classic ML models. Surely, the following research inspired from our results should make use of predictive DL models, as they are able to detect non-linear relationships in the data, and are also prepared to train on more features. Moreover, the low amount of SSL algorithms available definitely constrained our ML investigation, as we were not able to discover any of the most updated models.

We come to an end by encouraging the application of tstk to other related issues. With its updated structure and easy access, it can be used to investigate other environmental properties through EOS data. Our contribution to this toolkit can be applied to a variety of use cases, such as land-cover or land-usage prediction, polar ice investigations, ocean research, volcanic activity, crop yield and many more. We constrained its usage to land-cover and wheat crops yield, and so did the first contributor [Manuel Pereira [n. d.]], but one should bear in mind that tstk was built to handle any type of problem where ML models require EOS data.

REFERENCES

Nikos Alexandratos and Jelle Bruinsma. 2012. World agriculture towards 2030/2050: the 2012 revision. (2012).

Zhaohai Bai, Haigang Li, Xueyun Yang, Baoku Zhou, Xiaojun Shi, Boren Wang, Dongchu Li, Jianbo Shen, Qing Chen, Wei Qin, et al. 2013. The critical soil P levels for crop yield, soil fertility and environmental safety in different soil types. *Plant and Soil* 372 (2013), 27–37.

Else K Bünemann, Giulia Bongiorno, Zhanguo Bai, Rachel E Creamer, Gerlinde De Deyn, Ron de Goede, Luuk Fleskens, Violette Geissen, Thom W Kuyper, Paul Mäder, et al. 2018. Soil quality–A critical review. *Soil Biology and Biochemistry* 120 (2018), 105–125.

Raphaël d'Andrimont, Astrid Verhegghen, Michele Meroni, Guido Lemoine, Peter Strobl, Beatrice Eiselt, Momchil Yordanov, Laura Martinez-Sanchez, and Marijn van der Velde. 2021. LUCAS Copernicus 2018: Earth-observation-relevant in situ data on land cover and use throughout the European Union. *Earth System Science Data* 13, 3 (2021), 1119–1133.

J Dewis, F Freitas, et al. 1970. Physical and chemical methods of soil and water analysis. *FAO soils Bulletin* 10 (1970).

Daniel Geisseler and William R Horwath. 2016. Sampling for soil nitrate determination.

Gary Griffin, William Jokela, Don Ross, Dawn Pettinelli, Thomas Morris, and A Wilf. 1995. Recommended soil nitrate-N tests.

Wei Han, Ruyi Feng, Lizhe Wang, and Yafan Cheng. 2018. A semi-supervised generative framework with deep learning features for high-resolution remote sensing image scene classification. *ISPRS Journal of Photogrammetry and Remote Sensing* 145 (2018), 23–43.

John L Havlin. 2020. Soil: Fertility and nutrient management. In *Landscape and land capacity*. CRC Press, 251–265.

Tomislav Hengl, Johan GB Leenaars, Keith D Shepherd, Markus G Walsh, Gerard Heuvelink, Tekalign Mamo, Helina Tilahun, Ezra Berkhout, Matthew Cooper, Eric Fegraus, et al. 2017. Soil nutrient maps of Sub-Saharan Africa: assessment of soil nutrient content at 250 m spatial resolution using machine learning. *Nutrient Cycling in Agroecosystems* 109, 1 (2017), 77–102.

Tomislav Hengl, Matthew AE Miller, Josip Križan, Keith D Shepherd, Andrew Sila, Milan Kilibarda, Ognjen Antonijević, Luka Glušica, Achim Dobermann, Stephan M Haefele, et al. 2021. African soil properties and nutrients mapped at 30 m spatial resolution using two-scale ensemble machine learning. *Scientific Reports* 11, 1 (2021), 1–18.

Lin-Han Jia, Lan-Zhe Guo, Zhi Zhou, and Yu-Feng Li. 2022. LAMDA-SSL: Semi-Supervised Learning in Python. *arXiv preprint arXiv:2208.04610* (2022).

K. Kawamura, A. D. Mackay, M. P. Tuohy, K. Betteridge, I. D. Sanches, and Y. Inoue. [n. d.]. Potential for spectral indices to remotely sense phosphorus and potassium content of legume-based pasture as a means of assessing soil phosphorus and potassium fertility status. *International Journal of Remote Sensing* 32, 1 ([n. d.]), 103–124.

Chen Lin, Ronghua Ma, Qing Zhu, and Jingtao Li. 2015. Using hyper-spectral indices to detect soil phosphorus concentration for various land use patterns. *Environmental monitoring and assessment* 187 (2015), 1–10.

M.R. Maleki, L. Van Holm, H. Ramon, R. Merckx, J. De Baerdemaeker, and A.M. Mouazen. 2006. Phosphorus Sensing for Fresh Soils using Visible and Near Infrared Spectroscopy. *Biosystems Engineering* 95, 3 (2006), 425–436. https://doi.org/10.1016/j.biosystemseng.2006.07.015

Manuel Pereira. [n. d.]. TerraSenseTK: A Toolkit for Remote Soil Nutrient Estimation.

Alberto Orgiazzi, Cristiano Ballabio, Panagiotis Panagos, Arwyn Jones, and Oihane Fernández-Ugalde. 2018. LUCAS Soil, the largest expandable soil dataset for Europe: a review. *European Journal of Soil Science* 69, 1 (2018), 140–153.

Yassine Ouali, Céline Hudelot, and Myriam Tami. 2020. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278* (2020).

Anthony B Pattison, Phil Moody, and John Bagshaw. 2010. Vegetable plant and Soil health.

Amanda Ramcharan, Tomislav Hengl, Dylan Beaudette, and Skye Wills. 2017. A soil bulk density pedotransfer function based on machine learning: A case study with the NCSS soil characterization database. *Soil Science Society of America Journal* 81, 6 (2017), 1279–1287.

T Reinsch, L West, et al. 2010. The US national cooperative soil characterization database. In *Proceeding of the 19th World Congress of Soil Science*. 1–6.

Arthur J. Richardson and James H. Everitt. 1992. Using spectral vegetation indices to estimate rangeland productivity. *Geocarto International* 7, 1 (1992), 63–69.

Francesco Staccone. 2020. Deep learning for sea-ice classification on synthetic aperture radar (SAR) images in earth observation. Classification using semi-supervised generative adversarial networks on partially labeled data. (2020).

Gencer Sumbul. 2021. BigEarthNet-MM: A Large Scale Multi-Modal Multi-Label Benchmark Archive for Remote Sensing Image Classification and Retrieval. *IEEE Geoscience and Remote Sensing Magazine* 9, 3 (2021), 174–180. https://doi.org/10.1109/MGRS.2021.3089174

Gencer Sumbul, Marcela Charfuelan, Begüm Demir, and Volker Markl. 2019. Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 5901–5904.

John David Ronald Thomas. 2017. *Ion-Selective Electrode Reviews: Volume 7*. Vol. 7. Elsevier.

Janez Trontelj ml and Olga Chambers. 2021. Machine Learning Strategy for Soil Nutrients Prediction Using Spectroscopic Method. *Sensors* 21, 12 (2021), 4208.

Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. *Machine learning* 109, 2 (2020), 373–440.

Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Wei Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki, Bernt Schiele, Jindong Wang, Xing Xie, and Yue Zhang. 2022. USB: A Unified Semi-supervised Learning Benchmark for Classification. (2022). https://doi.org/10.48550/ARXIV.2208.07204

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. 2022. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering* (2022).

Lei Zhang, Lin Yang, Tianwu Ma, Feixue Shen, Yanyan Cai, and Chenghu Zhou. 2021. A self-training semi-supervised machine learning method for predictive mapping of soil classes with limited sample data. *Geoderma* 384 (2021), 114809. https://doi.org/10.1016/j.geoderma.2020.114809

Y. Özyiğit and M. and Bilgen. 2013. Use of Spectral Reflectance Values for Determining Nitrogen, Phosphorus, and Potassium Contents of Rangeland Plants. *Journal of Agricultural Science and Technology* 15, 7 (2013). arXiv:http://jast.modares.ac.ir/article-23-5138-en.pdf http://jast.modares.ac.ir/article-23-5138-en.html

Received 31 May 2024