POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Machine Learning Techniques For The Estimation Of Soil Moisture From Satellite Data

Master's Thesis in Space Engineering

Marco Varalla, 991400

Abstract:
The focus of this study is to evaluate and compare different types of machine learning algorithms for accurately estimating surface soil moisture using satellite data. The work aims to identify the most effective architecture for this purpose, striving to create a globally applicable "tool" that can be used in areas where in-situ stations are not present.

The chosen study area is the TxSON network in Texas, USA, characterized by arid conditions, uniform features, and sparse vegetation. The research covers a four-year period from January 1, 2018, to December 31, 2021.

To conduct this analysis, images with dual-polarized radar back-scatter (VH and VV polarizations) have been extracted from Sentinel-1, while red and near-infrared bands from Sentinel-2 have been used to calculate the Normalized Difference Vegetation Index (NDVI). In total 115 satellite observations have been collected.

In addition to satellite data, the study also incorporates in-situ data from the ISMN database, to retrieve soil moisture hourly time series. These data are later used for aligning and refining the machine learning models. Then the collected data have been partitioned into training and inference sets to develop a comprehensive database for analysis.

The work evaluates various ML algorithms, including Linear, Random Forest (RF), Support Vector Machine (SVM), Gaussian Process Regression (GPR), Multi-Layer Perceptron (MLP) and others, with the aim to fine-tune the hyperparameters of these models to achieve the lowest possible Root Mean Square Error (RMSE), which serves as a measure of the accuracy of the models' predictions.

However, after conducting the entire process and analyzing the outcomes, the research acknowledges that the results didn't align with the intended objectives.
In fact, the most noteworthy finding is the 'discovery' that this workflow, specifically involving the training of algorithms for predicting soil moisture values, demonstrates its effectiveness when applied in a '*localized*' approach.

The task of training a ML model for a specific site and accurately predicting values in an different area, in order to achieve the initial goal of the research, the *global tool*, appears to be seemingly impossible.

Advisor:
Prof. Claudio Maria Prati
Politecnico di Milano

Co-advisors:
Dr. Alfonso Amendola
Eni S.p.A.

Dr. Simone Sala
Eni S.p.A.

Academic year:
2022-2023

Key-words: Soil moisture, Sentinel-1, Sentinel-2, VH & VV Polarization, NDVI, ML Algorithms

Contents

1 Introduction **3**

2 Study Area **4**

3 Datasets **7**
3.1 Satellite - Sentinel Program . . . 7
3.2 In-situ measurements . . . 9

4 Machine Learning Algorithms **10**

5 Methodology **15**
5.1 Satellite Data Pre-processing . . . 15
5.2 Machine Learning Phase (Training and Inference) . . . 17

6 Results **19**

7 Conclusions **21**

A Appendix A **25**
A.1 Soil Moisture Time Series . . . 25
A.2 Input vs Output Dependencies . . . 26
A.3 Linear Regression Model Results . . . 27
A.4 Support Vector Machine Regression Model Results . . . 29
A.5 Random Forest Regression Model Results . . . 31
A.6 Ensemble of Learners Regression Model Results . . . 33
A.7 Multi-Layer Perceptron Regression Model Results . . . 35
A.8 Gaussian Process Regression Model Results . . . 37
A.9 Gaussian Kernel Regression Model Results . . . 39

1. Introduction

Soil moisture plays a crucial role as it quantifies the amount of water present within the soil matrix. This measurement is a fundamental aspect of Earth's hydrological cycle and holds great importance across various environmental sectors. These include agricultural productivity, weather forecasting, climate modeling, etc. [37]. Given the ongoing changes in climate patterns and the increasing challenges related to water resources, gaining an understanding of soil moisture dynamics has become even more essential.

The pivotal role played by soil moisture in different ecological and atmospheric processes underscores its critical importance. Within the agricultural realm, optimal soil moisture levels directly influence crop health and yield, while in hydrology [17], they dictate the partitioning of precipitation into runoff and groundwater recharge. Furthermore, soil moisture strongly influences regional climate patterns, impacting precipitation rates, surface energy fluxes, and even the severity of droughts and heatwaves. The management of this precious resource emerges as a pressing global concern, with implications for food security, water availability, and ecosystem sustainability [39].

Historically, the challenge of obtaining precise soil moisture measurements across various terrains, especially in remote or inaccessible areas, has been a significant hurdle. At present, there are approximately 71 International Soil Moisture Networks (ISMN) [8, 31] worldwide, boasting over 2800 operational stations. These stations provide nearly real-time measurements of soil moisture at specific points. However, the current distribution of these soil moisture network stations globally is far from uniform. This non-uniform distribution results in data gaps, particularly in regions lacking these measurement stations or where they are sparsely scattered.

In response to these challenges, researchers have proposed utilizing satellite imagery to estimate soil moisture on regional and global scales, addressing these distribution limitations [10, 38]. Integrated microwave radiometers and radar systems [15, 45] on satellites now could enable to assess soil moisture without invasive procedures, surpassing obstacles like vegetation cover and geographical limitations. This technological leap has ushered in a new era of global-scale soil moisture observation, bolstering our capability to address both local and global water resource concerns.

The rapid advancement of machine learning techniques has triggered a paradigm shift in the way data have been analyzed and interpreted [9, 12, 19]. Machine learning, a subset of artificial intelligence, excel at identifying intricate patterns within complex datasets. Their capacity to capture complex relationships in data positions could serve as a powerful tool for unraveling the subtleties present in satellite-derived information.

In the field of soil moisture analysis, ML algorithms could offer a promising solution to overcome significant challenges. Their ability to handle non-linear and multi-dimensional data makes them well-equipped for modeling the intricate interactions among variables that influence soil moisture levels. Through training on historical datasets, taken by the ISMN hub [31], and high-resolution satellite observations, provided by Sentinel Program [6, 7], machine learning could develop a nuanced grasp of the intricate correlation between remote sensing data and actual soil moisture conditions. This newfound understanding empowers them to predict soil moisture levels in regions where ground-based measurements are sparse, effectively bridging data gaps and enhancing the spatial coverage of soil moisture estimates [21, 29, 40, 46, 50].

This study's primary goal is to explore and harness the potential of artificial intelligence in advancing soil moisture estimation using satellite-derived data. Through the fusion of cutting-edge machine learning techniques and satellite remote sensing, this research aims to construct a predictive model that offers precise and adaptable soil moisture estimates across diverse landscapes and varying climatic conditions. Additionally, this investigation will delve into the efficacy of machine learning in processing and integrating heterogeneous satellite data sources, with the ultimate aspiration of refining the comprehension of soil moisture dynamics [11, 14, 20, 47, 48].

This thesis is organized into several structured sections, below listed.

- **Section 2 : Study Area**
- **Section 3 : Dataset**
- **Section 4 : Machine Learning Algorithms**
- **Section 5 : Methodology**
- **Section 6 : Results**
- **Section 7 : Conclusions**

2. Study Area

In this section, the chosen study area will be introduced, shedding light on its geographical characteristics. The selection of the study area holds intrinsic significance, setting the spatial stage for the research and lending depth to its findings.

This study was conducted over the region covered by the Texas Soil Observation Network (**TxSON**) [36]. The network consists of 40 monitoring stations arranged in a grid of approximately 1500 km^2, located in an area near Fredericksburg, TX, between the Pedernales and Colorado rivers. These network stations monitor and measure various site characteristics, such as weather and soil conditions.

This region experiences a semi-arid climate characterized by average annual rainfall of around 30 inches. Summers tend to be hot, with temperatures frequently exceeding 32°C, while winters are relatively mild, averaging around 15°C. The landscape comprises rolling hills, rocky terrains, and intermittent river valleys, contributing to diverse ecosystems and vegetation patterns.

The selection of TxSON as the study site was based on an analysis of various networks within the **International Soil Moisture Network** [31]. Factors considered for discrimination included the number of satellite observations and the ratio of the number of stations to the covered area. After screening, the most suitable networks were found to be *TxSON*, along with *WegenerNet* (Austria) [34], *Tahmo* (Kenya) [33], *Grow* (Hungary) [30], and *OzNet* (Australia) [32].

The final choice, after numerous attempts made over the available data, has fallen upon TxSON due to its more homogeneous characterization of the territory. Indeed, Texas features sparsely vegetated and arid lands, which might yield better insights through the integration of in-situ and satellite data.

Figure 1 below shows a satellite view (taken from *Google Earth* [13]) of the region of interest with a pin for each of the 40 stations. As can be observed, it is a highly arid region with sparse vegetation and few cultivated fields. In this scenario, the satellite may encounter fewer difficulties in detecting soil moisture compared to a territory with denser vegetation, where the vegetation itself could obstruct or deflect the signal, like encountered with WegenerNet site.

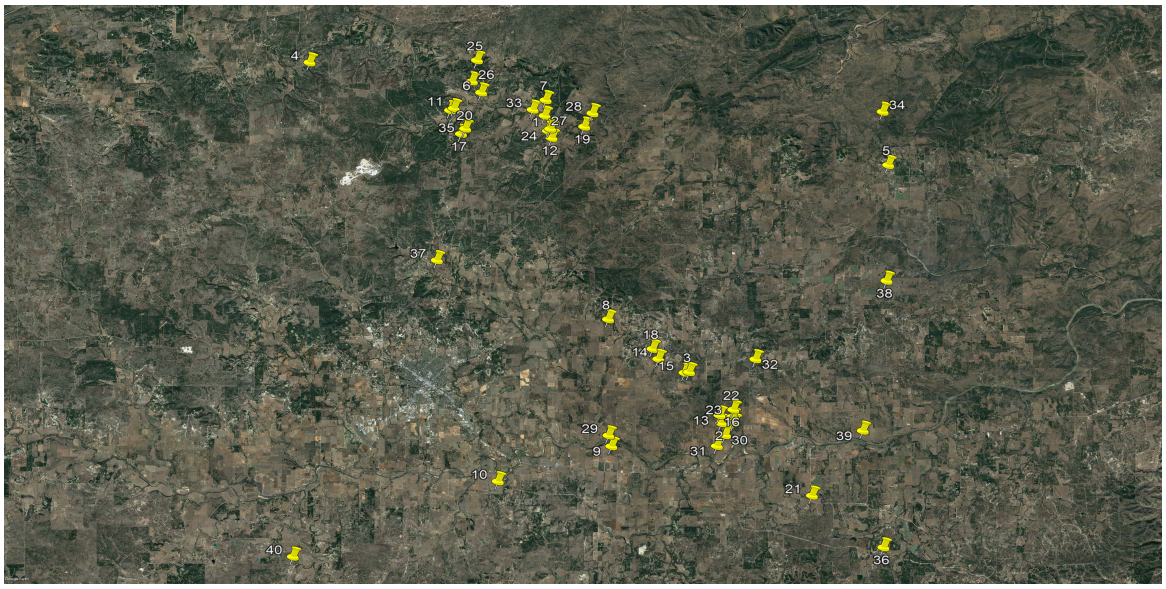


Figure 1: TxSON's Google Maps View

Matching the description given in the figure 1 above, it is apparent that certain stations are in closer proximity to each other compared to others. As a result, five distinct groups have been identified and are listed in the table 1 below. The stations that are relatively more isolated have not been taken into consideration.

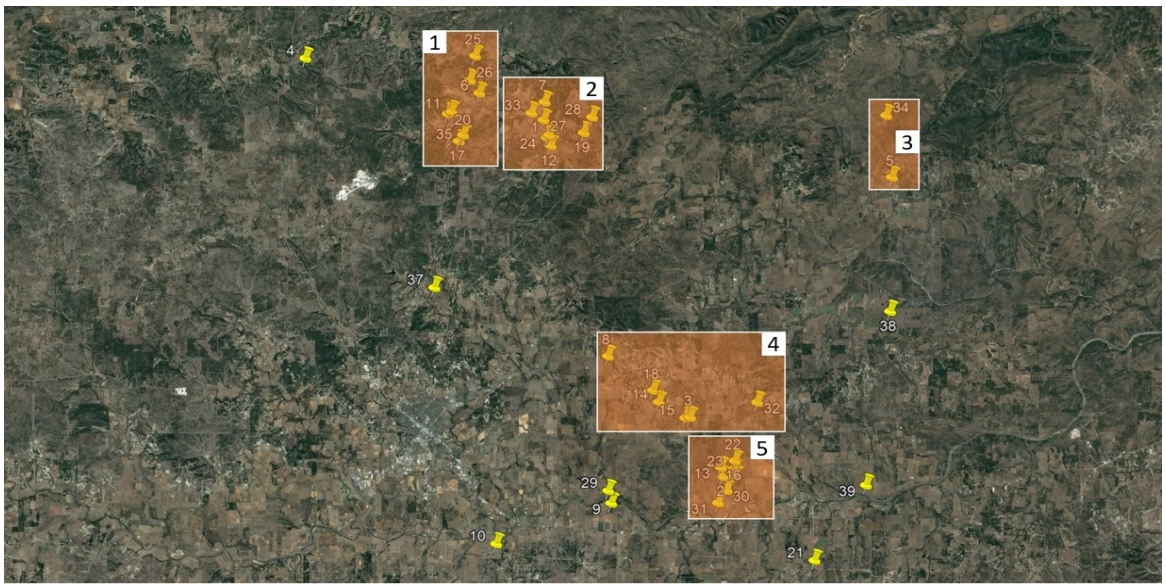


Figure 2: TxSON's Google Maps View with Groups

The size of the identified sub-regions is approximately 25 square kilometers. Within these scales, significant variations are not expected to occur in terms of soil moisture, as well as in the magnitudes of SAR (Synthetic Aperture Radar) images in VH (Vertical-Horizontal) and VV (Vertical-Vertical) polarizations, and NDVI (Normalized Difference Vegetation Index).

Figure 2 provides a visual representation of the extent of the five groups with their numeration and the accompanying table 1 displays the stations that constitute each respective group.

GROUP	STATION
1	6 - 11 - 17 - 20 - 25 - 26 - 35
2	1 - 7 - 12 - 19 - 24 - 27 - 28 - 33
3	5 - 34
4	2 - 13 - 16 - 22 - 23 - 30 - 31
5	3 - 8 - 14 - 15 - 18 - 32

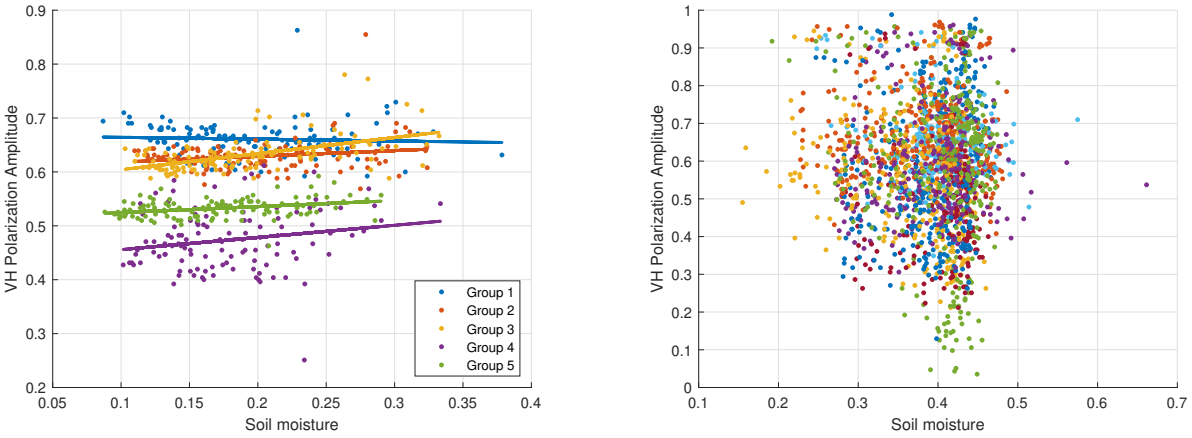
Table 1: Groups with relative stations

On the other hand, the stations 4, 10, 21, 29, 36, 37, 38, 39, 40, more isolated with respect to the others, have not been included in the analysis. Station number 9 was excluded beforehand due to evident errors in its measurements.

Attempts have also been made to include these more isolated stations among the others, but they do not yield improvements. This is likely due to the fact that being more isolated, they cannot be effectively correlated with satellite data due to their significantly different operational scale. Ground stations operate at a nearly punctual level, while satellite images cover much larger areas, where a single pixel corresponds to several hundred square meters.

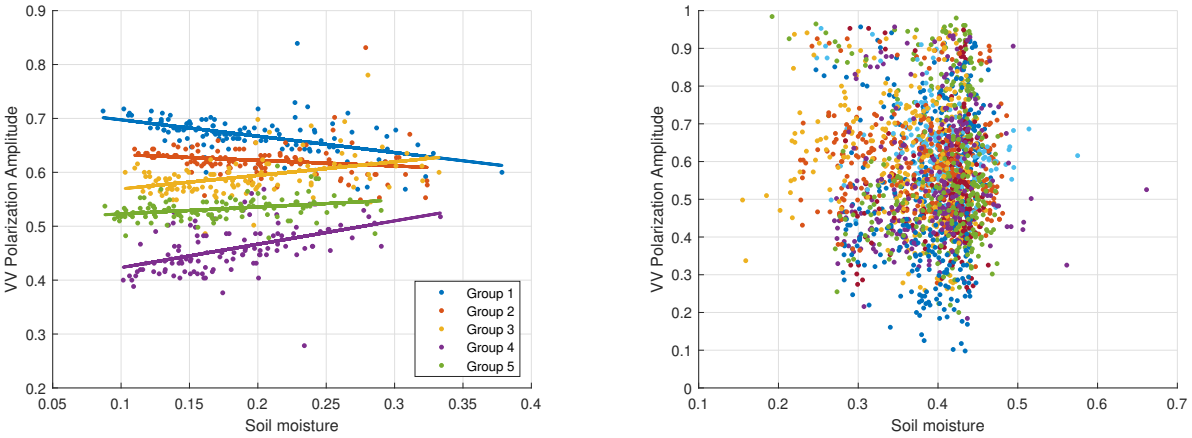
The decision was also influenced by VH-SM and VV-SM correlations in each network. Figures 3 and 4 highlight the comparisons between the networks of WegenerNet and TxSON. Notably, figures reveals a distinct correlations within the TxSON network, suggesting potential positive results, while the WegenerNet network presents a more complex situation due to likely differences in terrain characterization and higher vegetation levels compared to Texas.

Delving further into the specifics, positive outcomes are expected through the sequential process of machine learning models training followed by inference, considering each site individually. Conversely, the prospect of training the network on one area and predicting on another seems implausible and unpromising. Indeed, the correlations are discernible solely among each site, whereas on a global scale, the entire scenario appears more intricate, as exemplified by the WegenerNet case.



(a) TxSON Network (b) WegenerNet Network

Figure 3: VH Amplitude vs Soil Moisture Comparison



(a) TxSON Network (b) WegenerNet Network

Figure 4: VV Amplitude vs Soil Moisture Comparison

3. Datasets

3.1. Satellite - Sentinel Program

The study utilized openly accessible data from both Sentinel-1, utilizing **S**ynthetic **A**perture **R**adar (SAR), and Sentinel-2, employing optical imagery. These datasets were acquired by downloading the requisite information from the official European Space Agency (ESA) website [3], serving as a reliable source for data pertaining to both of these satellite missions.

ESA launched Sentinel-1A on April 3, 2014, with a revisit time of 12 days. On April 25, 2016, Sentinel-1B was also launched, with the same revisit time. Both satellites are part of the Copernicus Program [1] (formerly known as Global Monitoring for Environment and Security). When considering both satellites together, it is possible to combine the data and thus be able to obtain an observation of the same area, even if using two different satellites, every 6 days, also with product of the same type.

Operating at a frequency of 5 GHz, both Sentinel-1A and 1B [6], utilize microwave signals that possess the capability to effectively penetrate up to 5 cm beneath the surface of dry soil, enabling them to gather valuable data and images from beneath the Earth's topmost layer.

The Sentinel-1 satellites acquire images in strip map mode (with a spatial resolution of approximately 5 meters in both the azimuth and range directions), interferometric wide swath mode (5 meters in the range direction and 20 meters in the azimuth direction), extra-wide swath mode (40 meters in both directions), and wave mode (3 meters in both directions). Depending upon the acquisition mode, the SAR products are available at three levels:

- ◇ Level - 0 : Unfocused SAR raw data
- ◇ Level - 1 : Single Look Complex (SLC) and Ground Range Detected (GRD) data
- ◇ Level - 2 : Ocean geophysical product derived from level-1.

This study utilized VV and VH dual polarization GRD (level-1) images to capture comprehensive information about surface characteristics and interactions, as the VV and VH polarizations offer insights into different scattering behaviors of the target, enhancing the accuracy and depth of analysis.
The VH polarization configuration means that the radar waves are transmitted with a vertical orientation (V) and received with a horizontal orientation (H). Instead, VV polarization refers to the radar waves being transmitted with a vertical orientation (V) and received with a vertical orientation as well (V). These different polarizations have distinct interactions with various types of surfaces, which makes them useful in investigating alterations and patterns that become evident with variations in surface soil moisture.

In the following figure 5, it is possible to see an example of the naming convention for downloadable Sentinel-1 files.

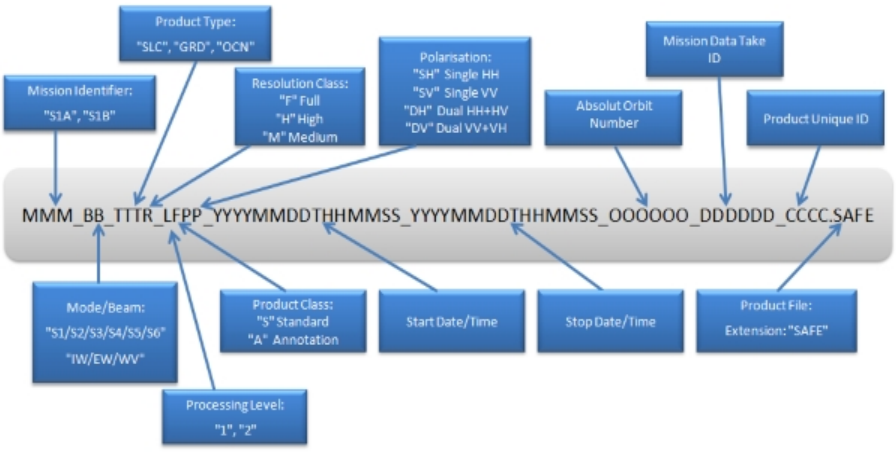


Figure 5: Sentinel-1 data official nomenclature

As can be seen, it's possible to extract multiple pieces of information from the nomenclature of each file. With a one-to-one correspondence between files and observations, the nomenclature provides details such as the mission, processing level, polarizations, start and end times of the observation, relative orbit, and other parameters that are subsequently useful during the data processing phase.

However, for the selected region, only data from Sentinel-1A were available, except for a small time window in June 2019. Therefore, the decision was made to utilize only the data from Sentinel-1A between **January 1, 2018**, and **December 31, 2021**. This approach resulted in the collection of 115 observations of the region, hence, approximately one every 12 days, in adherence to the presented revisit time, all related to relative orbit 107.

Subsequently, within the continuation of the Sentinel Program, two additional satellites were launched, namely Sentinel-2A (on 23^{rd} June 2015) and Sentinel-2B (on 7^{th} March 2017) [7], each with a revisit time of 10 days. By combining the passes of both satellites, it is possible to achieve an observation on average every 5 days. These satellites, positioned in polar orbits, significantly expanded the program's capabilities by providing high-resolution optical imagery, contributing crucial data for applications such as land cover mapping, environmental monitoring, and agricultural assessment.

They provide data in two levels :

- ◇ Level - 1C : Top-of-atmosphere corrected
- ◇ Level - 2A : Bottom-of-atmosphere corrected

In the context of this study, when both product levels were concurrently available for a given observation, a deliberate choice was made to exclusively utilize the '2A' level. This selection was driven by the recognition of the '2A' level's enhanced accuracy and calibration, which ultimately contributed to the study's overarching goal of achieving robust and dependable results.

Similarly to Sentinel-1, in the figure 6 it is possible to observe an example of the file naming convention for Sentinel-2. These filenames also encompass numerous pieces of information, such as the mission, relative orbit, start and end times of the observation, and so forth.

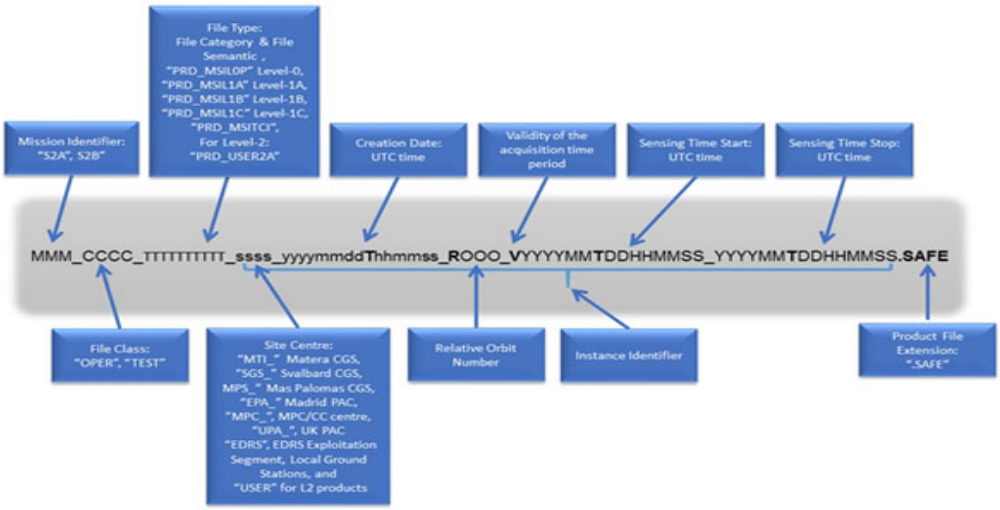


Figure 6: Sentinel-2 data official nomenclature

Sentinel-2 data has the limitation of "seeing" clouds since it operates with an optical view. To address this issue, a filtering process was implemented to eliminate data with a cloud cover percentage exceeding 9.4% . This filtering step aims to remove images that are significantly affected by cloud cover, as clouds can obscure the Earth's surface and affect the quality and reliability of the data for further analysis. By excluding such cloudy images, the data-set becomes more suitable for applications that require cloud-free observations to achieve more accurate and reliable results. However, some observations could be excluded due to their percentage of cloud cover, which leads to some gaps in the trend of observations.

They acquire images of the earth in 13 different spectral bands, with a spatial resolution depending on the band from 10 to 100 meters, from Visible Near Infra-Red (VNIR) to Short Wave Infra-Red (SWIR) of the electromagnetic spectrum. For this purposes, only the band-4 (Red; 665 nm) and band-8 (NIR; 865 nm) have been used. Indeed, by knowing these parameters, it is possible to calculate the **Normalized Difference Vegetation Index** (NDVI) using the following formula:

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{1}$$

As the name suggests, it is a simple normalized difference that serves as a graphical indicator, used to assess whether the observed area contains live vegetation, making it a valuable supplementary input alongside VH and VV polarizations, as suggested by the authors in their study [42]. Several attempts were made to check whether there was a genuine need for NDVI as an input or if it could be used solely as a kind of 'feasibility index,' providing information about the level of vegetation present in the region under examination. It was observed that using it as an input helps achieve lower RMSE indices, then presented in the section 6.

Normalized Difference Vegetation Index (NDVI) has a range that extends from -1 to 1. Negative values mostly represent bodies of water, such as rivers, lakes, seas, and oceans. Values close to zero but positive indicate arid areas with sparse vegetation and a prevalence of sand, rocks, or snow. Values closer to 1 on the positive side represent areas with predominant vegetation, approaching dense rain forests.

Concluding the discussion, Sentinel1 data were employed to retrieve amplitude values for VH and VV polarizations' observations, while Sentinel2 data were utilized for NDVI computation. For each of the five station groups, medians of values within their respective regions (table 1) were calculated. This approach aims to standardize the region's values, mitigating the impact of features like water bodies or populated areas that might otherwise result in nonsensical values.

3.2. In-situ measurements

In line with the previous subsection, the chosen time window of interest ranges from 01/01/2018 to 31/12/2021. Using the ISMN [31], time series data for each of the 40 stations within the TxSON network were obtained. For each station, information about the type of probe used, their respective coordinates, and soil moisture measurements at hourly intervals are available. These measurements were taken at a depth between 3 cm and 5 cm for each station with *CS655 probes* [41], visible in the figure 7 below.

The CS655 sensor [7] consists of two 12 cm long stainless steel rod electrodes connected to an electronics board. CS655 sensors with short electrodes are easier to install in hard soil. The CS655 sensors are designed for soils with high electrical conductivity. It measures propagation time, signal attenuation and temperature. From these values, the dielectric permittivity, volume water content and volume electrical conductivity are then derived. The measured signal attenuation is used to correct the effect of loss on reflection detection and thus to measure the propagation time.

Figure 7: CS655 Probe

The volumetric electrical conductivity of the soil is also calculated from the attenuation measurement. A thermistor in thermal contact with a probe near the epoxy surface measures the temperature. The horizontal installation of the sensor ensures accurate measurement of the soil temperature at the same depth as the water content. Temperature measurement in sensor orientations other than horizontal will sense the temperature near the inlet of the rods to the sensor body.

Data points that do not meet a specific reliability criterion, indicated by a negative exit-flag, are discarded. In cases where there are missing readings, they are filled with NaN (Not a Number) values to ensure complete time series vectors, with a value each hour.

For each of the hourly time series, a daily average was computed, aggregating over 24 data points, with the intention of "filling in" potential gaps and measurement absences. Subsequently, for each of the five groups, averages were calculated across all stations within each group. This process further contributes to standardizing the values across the designated region, thereby facilitating enhancements in the subsequent comparison with satellite observation values.

In Appendix A (figure 15) , it is possible to find graphs displaying the temporal variations of soil moisture for each of the five group of stations. These graphs illustrate how soil moisture levels change over time for the respective sets of stations. It can be observed from the graphs in the appendix A that the minimum value for the various identified regions is approximately 0.1 millimeters of water per cubic meter of soil (mm/m^3), while the maximum value is around 0.4 mm/m^3.

4. Machine Learning Algorithms

In the realm of data-driven decision-making, machine learning models stand as the vanguards of computational intelligence. These models, rooted in the principles of artificial intelligence, provide a systematic approach to extracting patterns, making predictions, and uncovering insights from vast and complex datasets. In this section, different landscape of machine learning models have been explored, focusing on their fundamental concepts and applications, selecting the most suitable model for these specific tasks [16].

Each model among those explored begins from the same starting point. From the dataset described in Section 3, the model's input and output are identified.

$$(\mathbf{x}^{(i)}, y^{(i)}) \quad i = 1, \ ..., \ n \tag{2}$$

Starting from each input-output pair 2, where the input consists of a three-component vector (VH, VV, and NDVI values) and the output represents the soil moisture value, the task of the model lies in identifying a function of some kind corresponding to:

$$y = f(\mathbf{x}, \mathbf{w}, \theta) \tag{3}$$

Where $\mathbf{w}$ is the vector of the weight and θ the set of the optimized hyperparameters.

The primary objective of the research is to identify the most suitable category and structure among various machine learning model families. The central focus of this optimization and minimization effort revolves around the root mean square error (RMSE), a critical metric calculated using the following formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y(i) - \hat{y}(i))^2}{n}} \tag{4}$$

Where:

- $y(i)$ represents the actual observed value for a particular data point
- $\hat{y}(i)$ represents the predicted value by the model for the same data point
- n is the total number of data points used for evaluation

Attaining a lower RMSE signifies the establishment of a highly dependable model, proficient in making predictions with remarkable precision and minimal error. This enhanced accuracy should render the model versatile and aptly suited for an array of diverse applications and tasks.

In the following compilation, a variety of distinct types is listed, with each type accompanied by its corresponding MATLAB function, used as tool for conducting hyperparameter optimization:

1. **Linear Regression model** using Matlab *fitrlinear* function [25]

   It is a model that establishes a linear relationship between a dependent variable and one or more independent variables. It aims to predict the dependent variable's value based on the weighted sum of the independent variables, using a line that minimizes the sum of squared differences between predicted and actual values.
   The objective is to find a function like:

   $$\hat{\mathbf{y}} = \mathbf{w} \cdot \mathbf{x} + \mathbf{w_0} \tag{5}$$

   Where $\hat{\mathbf{y}}$ is the predicted value, through a combination of weights ($\mathbf{w}$ and $\mathbf{w_0}$) and input $\mathbf{x}$.
   To achieve this goal and subsequently determine the optimal hyperparameters, equation 6 is minimized. This equation calculates the Mean Squared Error (MSE) in addition to a value that depends on the hyperparameters, which are grouped within the variable θ.

   $$\mathcal{L}(\mathbf{w}, \theta) = \frac{1}{n}\sum_{i=1}^{N}(\hat{y}^{(i)}(\mathbf{w}) - y^{(i)})^2 + \theta\|\mathbf{w}\|_1 \tag{6}$$

   Where the variables have the same meaning as those in equation 4.

This model is particularly suitable when the relationship between the variables can be approximated by a linear equation, which represents a straight line on a graph. Evidently, by also observing the figures 16 in the appendix A, it could serve as an initial step to assess the behavior of various parameters; however, undoubtedly, further enhancement of the situation will necessitate the exploration of alternative architectures.

The eligible hyperparameters, in the used built-in function, are:

- ◇ *Lambda* : regularization term strength
  The function searches among positive values, by default log-scaled in the range [1e-5/n, 1e5/n], where n is the number of observations.

- ◇ *Learner* : linear regression model type
  The function searches among 'svm' and 'leastsquares'.

- ◇ *Regularization* : complexity penalty type
  The software composes the objective function for minimization from the sum of the average loss function and the regularization term (last part of the equation 6).
  - ○ ridge : the function sets the Solver value to 'lbfgs' by default.
  - ○ lasso : the function sets the Solver value to 'sparsa' by default.

2. **Support vector machine (SVM) Regression model** using Matlab *fitrsvm* function [27]

It works by finding a hyperplane that best separates data points of different classes in a high-dimensional space. SVM aims to maximize the margin between classes, relying on support vectors, data points nearest to the separating hyperplane, to achieve robust and accurate predictions. It is shown in the figure 8 below.
It is typically used when there's a need to classify data points into different categories or groups. SVMs are versatile and can handle both linear and non-linear relationships between variables.
It creates a non-probabilistic binary linear classifier. An SVM model represents examples as points in space, mapped in a way that examples from the two distinct categories are clearly separated by the widest possible margin. New examples are then mapped into the same space, and the prediction of the category they belong to is made based on the side they fall on.

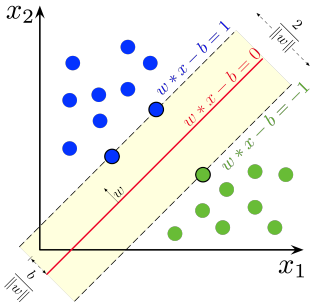


Figure 8: SVM Method

The eligible hyperparameters, in the used built-in function, are:

- ◇ *BoxConstraint* : constraint for the alpha coefficients (the absolute value of the Alpha coefficients cannot exceed the value of BoxConstraint).
  The function searches among positive values, by default log-scaled in the range [1e-3, 1e3].

- ◇ *KernelScale* : the software divides all elements of the predictor matrix X by the value of KernelScale. Then, the software applies the appropriate kernel norm to compute the Gram matrix.
  The function searches among positive values, by default log-scaled in the range [1e-3, 1e3].

- ◇ *Epsilon* : half the width of the epsilon-insensitive band, a non-negative scalar value.
  The function searches among positive values, by default log-scaled in the range [1e-3, 1e2] * iqr(Y)/1.349.

- ◇ *KernelFunction* : function used to compute the Gram matrix.
  The function searches among 'gaussian', 'linear', and 'polynomial'.

- ◇ *PolynomialOrder* : kernel function order, a positive integer.
  The function searches among integers in the range [2, 4].

- ◇ *Standardize* : flag to standardize the predictor data.
  The function searches among 'true'(1) and 'false'(0).

3. **Random Forest (RF) Regression model** using Matlab *fitrtree* function [28]

It constructs multiple decision trees during training and combines their predictions to improve accuracy and reduce over-fitting. Each tree is built on a subset of the data and features, making the final prediction by aggregating the outcomes of individual trees, resulting in a more robust and accurate model.
It is obtained through the aggregation, called 'bagging', of decision trees 9. Random Forests serve as a solution aimed at minimizing over-fitting of the training set compared to individual decision trees. 'Bagging' is an ensemble learning technique in which multiple models of the same type are trained on different datasets, each derived from an initial dataset through random sampling with replacement.

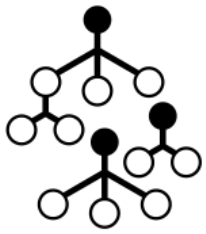


Figure 9: RF Decision Tree

The eligible hyperparameters, in the used built-in function, are:

- ◇ *MaxNumSplits* : maximal number of decision splits (or branch nodes).
  The function searches among integers, by default log-scaled in the range [1, max(2,n - 1)], where n is the number of observations.

- ◇ *MinLeafSize* : minimum number of leaf node observations, a positive integer value.
  The function searches among integers, by default log-scaled in the range [1, max(2, floor(n/2))], where n is the number of observations.

- ◇ *NumVariablesToSample* : number of predictors to select at random for each split.
  The function does not optimize over this hyperparameter.

4. **Ensemble of learners Regression model** using Matlab *fitrensemble* function [22]

It entails amalgamating the forecasts generated by multiple models (referred to as learners) to produce predictions that are not only more accurate but also more resilient compared to the predictions of individual models.
It trains regression ensemble model object that contains the results of boosting 100 regression decision trees, as shown in figure 10.

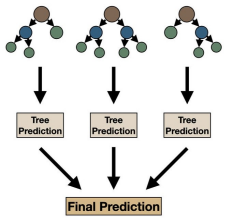


Figure 10: Ensemble of Decision Tree

The eligible hyperparameters, in the used built-in function, are:

- ◇ *Method* : ensemble aggregation method.
  The function searches among 'Bag' or 'LSBoost'.

- ◇ *NumLearningCycles* : number of ensemble learning cycles, a positive integer.
  The function searches among positive integers, by default log-scaled with range [10, 500].

- ◇ *LearnRate* : learning rate for shrinkage.
  The function searches among positive reals, by default log-scaled with range [1e-3, 1].

- ◇ *MinLeafSize* : minimum number of leaf node observations, a positive integer.
  The function searches among integers log-scaled in the range [1, max(2,floor(n/2))], where n is the number of observations.

- ◇ *MaxNumSplits* : maximal number of decision splits (or branch nodes).
  The function searches among integers, by default log-scaled in the range [1, max(2,n - 1)], where n is the number of observations.

- ◇ *NumVariablesToSample* : number of predictors to select at random for each split.
  The function does not optimize over this hyperparameter.

5. **Multi-layer Perceptron (MLP) Regression model** using Matlab *fitrnet* function [26]

It consists of multiple layers of interconnected nodes (neurons), including an input layer, one or more hidden layers, and an output layer, with each layer fully connected to the next. Each node is a neuron, with an activation function and its weighted inputs, allowing the network to learn complex relationships in data (figure 11).

MLP neural networks are highly flexible but may require careful hyperparameter tuning and a sufficiently large amount of data to prevent over-fitting, especially for deeper architectures. It is a model of an artificial neural network that maps sets of input data to appropriate sets of output data. The MLP employs a supervised learning technique called back-propagation for network training.

The basic computations performed by each neuron are used to predict the model performance. This is a two-step process. In the first step, individual inputs of the neuron ($\mathbf{x}$) and the corresponding weight values ($\mathbf{w}$) are combined together by a summation function. The output of a summation function is a dot product of weight vectors and input vectors. A bias (b) is added to the dot product forming the output (f) according to equation 7. In the second step, output (f) is fed into the argument of an activation function, which is then used to calculate a scalar value.

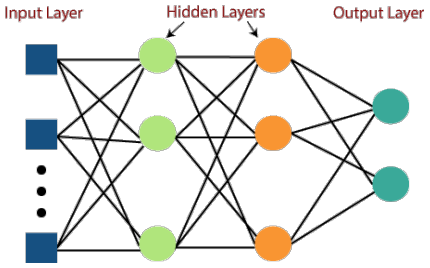


Figure 11: MLP Architecture

$$\mathbf{f} = \sum \mathbf{x} \cdot \mathbf{w} + \mathbf{b} \tag{7}$$

The neuron input in each hidden layer is constructed as a linear combination of its received input values that correspond to the output of the previous layers according to equation 8:

$$\mathbf{a^l} = \sum \mathbf{w^l} \cdot \mathbf{x^{l-1}} + \mathbf{b^l} \tag{8}$$

where, $\mathbf{a^l}$ is the input of a neuron present at l layer, $\mathbf{w^l}$ is the weight vector for the neurons present at l layer, $\mathbf{a^{l-1}}$ is the output of a neuron present at *l-1* layer, and $\mathbf{b^l}$ is the bias value at layer l which is followed by an activation function.

The eligible hyperparameters, in the used built-in function, are:

- ◇ *Activations* : activation functions for the fully connected layers of the neural network model.
  The function optimizes Activations over the set 'relu', 'tanh', 'sigmoid', 'none'.

- ◇ *Lambda* : regularization term strength, a non-negative scalar. The software composes the objective function for minimization from the mean squared error (MSE) loss function and the ridge (L2) penalty term.
  The function optimizes Lambda over continuous values in the range [1e-5, 1e5]/n, where n is the number of observations.

- ◇ *LayerBiasesInitializer* : type of initial fully connected layer biases.
  The function optimizes LayerBiasesInitializer over the two values 'zeros', 'ones'.
  If 'zeros', each fully connected layer has an initial bias of 0. If 'ones', each fully connected layer has an initial bias of 1.

- ◇ *LayerWeightsInitializer* : function to initialize the fully connected layer weights.
  The function optimizes LayerWeightsInitializer over the two values 'glorot', 'he'.

- ◇ *LayerSizes* : sizes of the fully connected layers in the neural network model, a positive integer vector.
  The function optimizes over the three values 1, 2, and 3 fully connected layers, excluding the final fully connected layer. fitrnet optimizes each fully connected layer separately over 1 through 300 sizes in the layer, sampled on a logarithmic scale.

6. **Gaussian Process Regression (GPR) model** using Matlab *fitrgp* function [23]

   It models the relationship between input data points and corresponding output values as a Gaussian process, which is a collection of random variables. GPR not only provides predictions but also estimates the uncertainty associated with those predictions. Gaussian Process regression models should be particularly useful when dealing with small datasets or when uncertainty estimation is important.

   The eligible hyperparameters, in the used built-in function, are:

   - ◇ *BasisFunction* : if n is the number of observations, the basis function adds the term $H * \beta$ to the model, where H is the basis matrix and β is a p-by-1 vector of basis coefficients.
     The function searches among 'constant', 'none', 'linear', and 'pureQuadratic'.

   - ◇ *KernelFunction* : form of the covariance function
     The function searches among 'ardexponential', 'ardmatern32', 'ardmatern52', 'ardrationalquadratic', 'ardsquaredexponential', 'exponential', 'matern32', 'matern52', 'rationalquadratic', and 'squaredexponential'.

   - ◇ *KernelScale* : the function uses the KernelParameters argument to specify the value of the kernel scale parameter, which is held constant during fitting. KernelScale cannot be optimized for any of the ARD kernels.

   - ◇ *Sigma* : initial value for the noise standard deviation of the Gaussian process model, a positive scalar value.
     The function searches among real value in the range [1e-4, max(1e-3,10*ResponseStd)], where ResponseStd = std(Y).

   - ◇ *Standardize* : flag to standardize the predictor data.
     The function searches among 'true'(1) and 'false'(0).

7. **Gaussian Kernel Regression model** using Matlab *fitrkernel* function [24]

   It is a non-parametric machine learning technique used for regression tasks. Unlike traditional regression models that involve fitting a predefined function to the data, kernel regression estimates the target value for a data point by considering the weighted average of its neighboring points in the feature space.

   The eligible hyperparameters, in the used built-in function, are:

   - ◇ *Epsilon* : half the width of the epsilon-insensitive band, a non-negative scalar value.
     The function searches among positive values, by default log-scaled in the range [1e-3, 1e2] * iqr(Y)/1.349.

   - ◇ *KernelScale* : kernel scale parameter, a positive scalar.
     The function searches among positive values, by default log-scaled in the range [1e-3, 1e3].

   - ◇ *Lambda* : regularization term strength, a non-negative scalar.
     The function searches among positive values, by default log-scaled in the range [1e-3, 1e3]/n, where n is the number of observations.

   - ◇ *Learner* : linear regression model type.
     The function searches among 'svm' and 'leastsquares'.

   - ◇ *NumExpansionDimensions* : number of dimensions of the expanded space.
     The function searches among positive integers, by default log-scaled in the range [100, 10000].

5. Methodology

In this section, the adopted methodology for conducting the analysis and obtaining the subsequent results will be presented in detail. The procedures, tools, and techniques employed to address the research questions and attain the predetermined objectives will be elucidated. [4, 42–44, 49].

5.1. Satellite Data Pre-processing

First and foremost, the satellite data from both Sentinel-1 and Sentinel-2 missions were processed. Each of them underwent a specific workflow. All the data has been processed using the *ESA SNAP (Sentinel Application Platform)* software [2].

It is a powerful remote sensing software designed to process and analyze data from the European Space Agency's Sentinel satellites. It offers a range of tools for image manipulation, data pre-processing, and information extraction. With its user-friendly interface, SNAP facilitates the exploration of Earth observation data, enabling users to derive valuable insights for various applications, including environmental monitoring, disaster management, and agricultural assessment.

Processing Sentinel-1 data, which employs Synthetic Aperture Radar (SAR), requires a more elaborate and time-consuming approach compared to Sentinel-2's optical data. The SAR data undergoes complex steps like calibration, speckle filtering, and terrain correction to transform raw signals into usefull images. In contrast, Sentinel-2's optical data processing involves tasks such as radiometric calibration and atmospheric correction, making it relatively less intricate and quicker.

In fact, for Sentinel-1 data, the following steps were followed [18]:

- *Apply Orbit File*

  It is used to apply precise orbit files to Sentinel satellite imagery. It corrects satellite positioning errors, enhancing the geometric accuracy of the data for further processing and analysis.

- *Calibration*

  It involves comparing the observed values with known reference values or standards and making necessary adjustments to the measurements to remove systematic errors and uncertainties.

- *Speckle Filtering*

  It aims to preserve image details and edges while suppressing the speckle noise to improve the visual quality and aid in subsequent analysis or interpretation of the images.

- *Terrain Flattening*

  It is used to remove the effects of terrain variations from satellite or aerial imagery. The terrain's relief can cause distortions in the image, making it challenging to accurately compare or analyze features.

- *Terrain Correction*

  It corrects geometric distortions caused by Earth's topography in satellite or aerial imagery using digital elevation models (DEMs). It produces geo-referenced and accurate representations of the Earth's surface for precise analysis and mapping.

- *Subset*

  It is a process used to 'crop' regions of interest from satellite data images that cover much larger areas.

- *Conversion*

  It is a conversion from the output generated by ESA SNAP to an image format (PNG). This conversion is done to make the data more easily analyzable during the post-processing phase.

Regarding Sentinel-2 data, the process involved taking Band 4 (red) and Band 8 (near-infrared) into account and applying the formula 1 to calculate the NDVI index. Afterward, a "Subset" operation was performed to focus on specific regions of interest within the images. Finally, the data was converted into PNG format for further analysis and visualization.

In line with what was previously mentioned in section 3, Sentinel-1A has a revisit time of 12 days, while the Sentinel-2 constellation combined (Sentinel-2A and Sentinel-2B) has a revisit time of 5 days, with the possibility of discarding some observations due to predominant cloud cover. This time offset between the two satellites is evident.

To address this challenge, a strategic approach was implemented, shown in figure 12, involving the correlation of each of the 115 Sentinel-1 images with the temporally closest available Sentinel-2 image. Through this method, the acquisitions from both satellite platforms could be effectively synchronized, minimizing temporal disparities between them.

Consequently, an extensive dataset emerged, encompassing 115 images for VH polarization, an additional 115 for VV polarization, and a corresponding 115 images for NDVI values. This ensured a pairing of data from both Sentinel-1 and Sentinel-2 satellites, rendering them suitable for subsequent analytical pursuits.

Once these images are ready, the analysis moves on to examine the different stations and their geographical positions. The approach selected involves using data from nearby stations, approximately enclosed within an area of 40 square kilometres. The average of the soil moisture temporal trends from each station within this area is calculated. This process results in the formation of five groups, visible in the figure 2. However, data originating from stations that are relatively isolated from others were not subjected to analysis. This discrepancy arises due to the contrasting operational scales of ground stations, which operate on an almost point-wise scale, and satellite observations, which encompass significantly larger geographical areas.

Within each of these delineated regions, the respective VH and VV polarization images, along with NDVI data, are extracted through a process of 'cropping' the primary images into smaller segments, leveraging the precise station coordinates. Following this, a median value calculation is executed for these three images, strategically employed to alleviate the potential impact of water bodies or urban locales, which might otherwise disrupt the SAR and optical signal measurements.

The ultimate objective is to acquire a set of quantitative values for each of the newly defined regions. This encompasses distinct numeric values for VH polarization, VV polarization, NDVI, and soil moisture. This iterative procedure is performed for all 115 images spanning the duration of the four-year investigation period.

Concluding the process, a database represented as a 115x4 matrix for each group is assembled. In this matrix, the first column corresponds to VH, the second to VV, the third to NDVI, and the fourth to soil moisture. This structured compilation results in a total of five different datasets. These will be treated as separate network, playing a pivotal role in the training and inference phases of the analysis. This methodology is qualitatively shown, as an example, in the following figure 12.

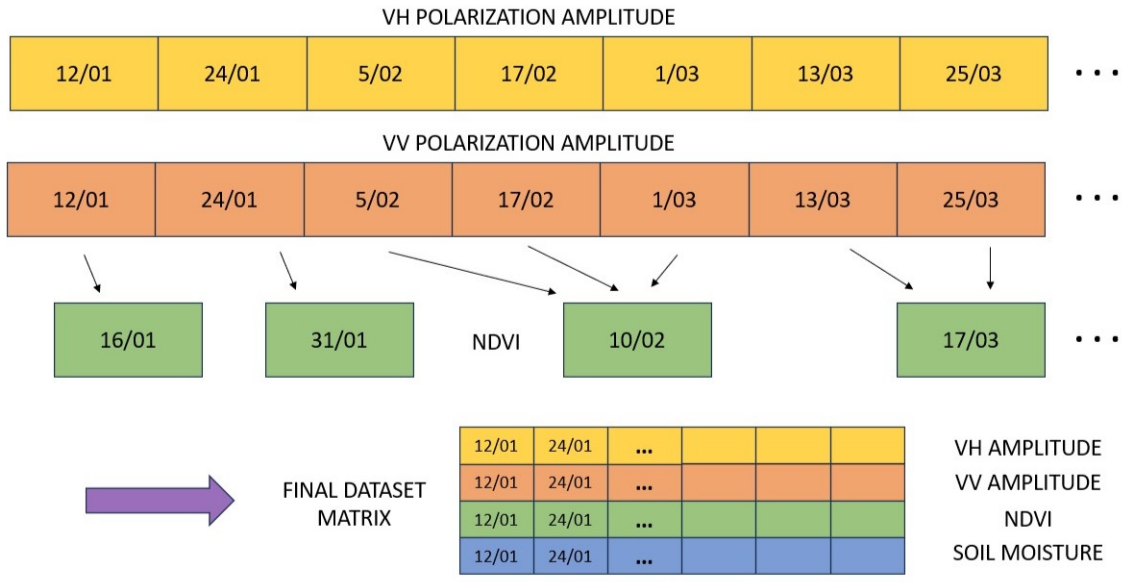


Figure 12: S1-S2 Data Match-up

5.2. Machine Learning Phase (Training and Inference)

Following the pre-processing phase and the creation of the 5-matrices database representing each group, the subsequent steps involve embarking on the training phase of machine learning model, followed by the inference one [5, 35].

Each of the five database were divided into distinct 'training' and 'inference' subsets. For this instance, a distribution of 75 % for the first phase and 25 % for the second one was chosen. Before this partition, the rows in the database underwent shuffling to introduce a higher degree of randomness and to ensure the avoidance of data solely linked to specific time periods being confined to a single phase.

Subsequent to the creation of the distinct databases, the following step entails the utilization of MATLAB functions, enumerated in Section 4. A uniform process is employed across all types, commencing with model training and its refinement through the automated selection of hyperparameters. For each of the architectures, the process begins with the utilization of three inputs (VH amplitude, VV amplitude and NDVI), culminating in the generation of a singular output represented by soil moisture [42–44]. An example of the used code is reported below.

```
mdl = fitr****(X_train, Y_train, 'OptimizeHyperparameters', 'all', ...
'HyperparameterOptimizationOptions', ...
struct('MaxObjectiveEvaluations', 200));
```

Each of the models (*mdl*) is trained with the corresponding function (*fitr*****), listed in Section 4, taking VH, VV, and NDVI values as input (X_{train}) and providing moisture values as output (Y_{train}), using 200 iterations for each training.

Upon the establishment of model hyperparameters, the training database is partitioned into four blocks. Employing a cyclic approach, one of these blocks is systematically excluded while the model undergoes retraining (with the unchanged hyperparameters) using the remaining three blocks. This trained model is then tested against the omitted block. By comparing the resultant predictions to the actual values, the calculation of Root Mean Square Error (RMSE) transpires, serving as a pivotal discriminating metric. Through this iterative "Leave One Out" process, a total of 4 indices are determined. By calculating their average, a precise and representative value for the training phase is derived.

This process is graphically exemplified in the figure 13 below.

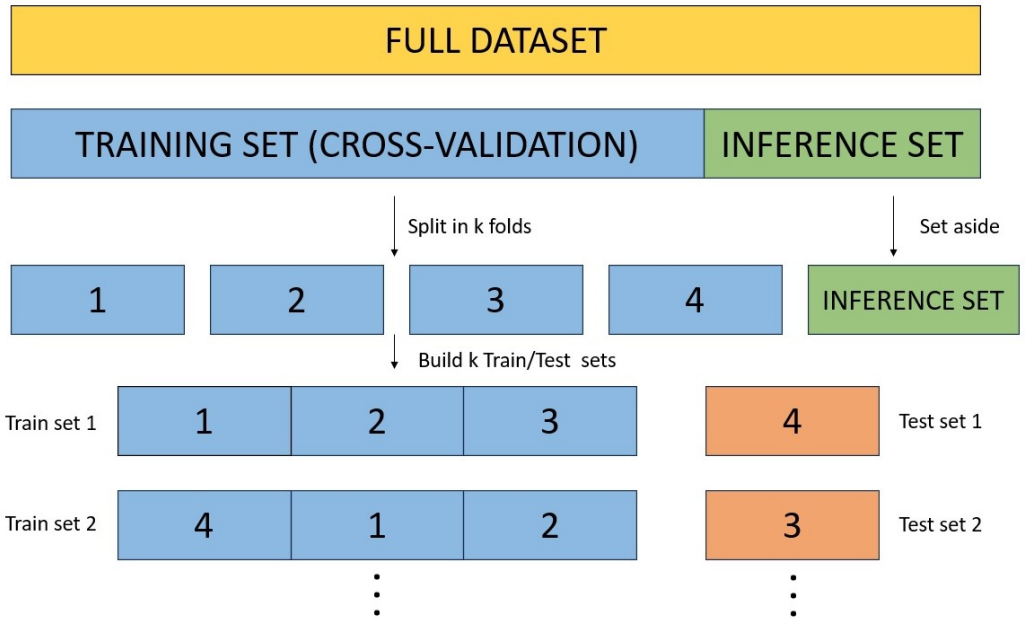


Figure 13: Cross-validation Process Diagram

Following that, the inference phase takes place, wherein the algorithm is tested on the portion of data that had not been encountered before, constituting the remaining 25 %. By contrasting predictions with actual data, the Root Mean Square Error value can be computed for each site and configurations, as shown below.

```
Y_pred = predict(mdl, X_inf);

rmse_inf = rmse(Y_inf, Y_pred);
```

Here, RMSE for the inference phase is computed using the true (Y_{inf}) and the predicted (Y_{pred}) values of soil moisture, evaluated using the optimized model (mdl) and the 'unseen' input data (X_{inf}).

Enhanced performance is denoted by decreased RMSE values in this phase. Hence, identifying the optimal architecture is synonymous with uncovering the one displaying the minimal disparity between predictions and actual values. Prudence should be exercised in refraining from employing RMSE from the training phase, as specific types of model could overly train the system, resulting in over-fitting issues: the model becomes exceptionally proficient with the training data, yet falters when confronted with new data due to the acquisition of noise and idiosyncrasies.

Ensuring an ample quantity of measurements is crucial, as an insufficient dataset could lead to outcomes disproportionately influenced by the limited sample size. Inadequacies in complexity may hinder the network's capacity to grasp essential relationships, ultimately yielding diminished accuracy.

Both the problem concerning the phenomenon of over-fitting and the limited amount of data will be addressed in the sections 6 and 7.

6. Results

This section is dedicated to presenting the outcomes emerged from the research conducted in this study's domain. Before delving into the details, it is essential to observe, through the graphs presented in the Appendix A (figure 16), the intricate interdependence existing among the three distinct inputs (VH, VV, and NDVI) and the resultant output.

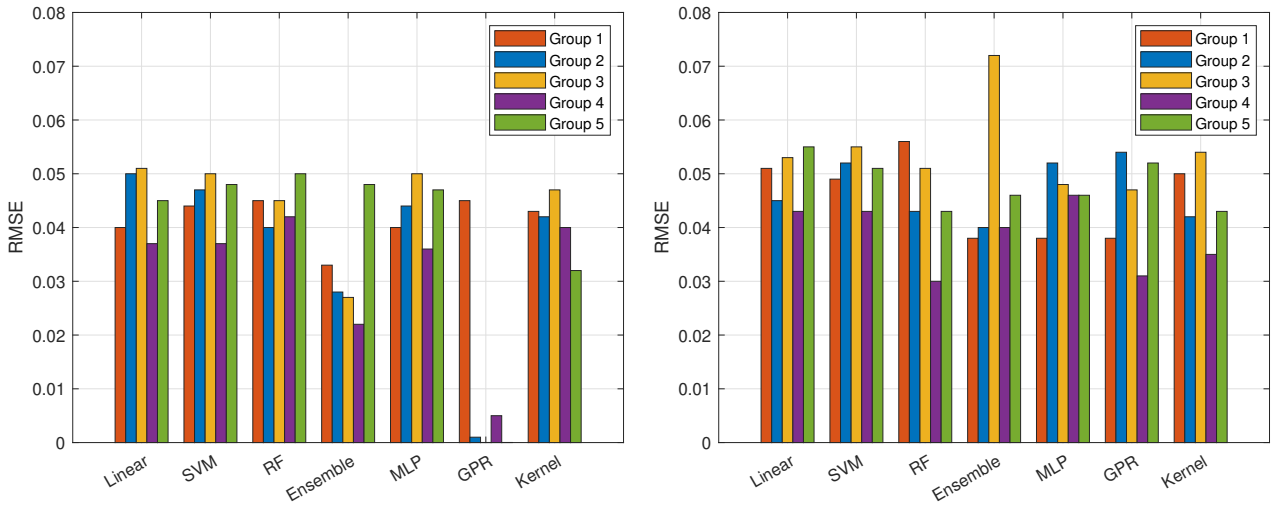
It becomes evident that some of the three inputs (mainly VV and VH polarization), in specific site, when considered individually rather than collectively, exhibit a promising correlation. Therefore, the use of machine learning models is expected to yield at least satisfactory results. Their application holds the promise of potentially uncovering intricate connections and patterns that might remain otherwise elusive.

For each of the individual categories, it is feasible to extrapolate the projected outcomes from both the training phase and the subsequent inference phase. These projections can then be graphically represented using parity plots, wherein actual values are positioned along the x-axis and predicted values along the y-axis. The presence of a diagonal line on the plot signifies perfect prediction, a scenario in which each predicted value impeccably aligns, in an ideal way, with its corresponding actual value.

Concluding the analysis, each of the specified categories presents an associated Root Mean Square Error (RMSE) value for both the training and inference phases, computed following the formula 4. These values serve as essential metrics, providing insights into the model's accuracy and performance during distinct phases of the analysis.

For every set of the five groups, treated individually as if they were completely separate networks, each architecture outlined in Section 4 is utilized, and the corresponding parity plots are reported in Appendix A, together with a table containing the RMSE indices relative to each phase.

Figure 14 depicts the outcomes arising from the conducted process. In these histograms, each bar corresponds to a group (represented by a distinct color), with differentiation for each of the selected models, both for training and inference phase. Additionally, it is feasible to observe the numerical values for each model and group, along with the chosen hyperparameters, in Appendix A.



(a) Training Phase

(b) Inference Phase

Figure 14: ML Models RMSEs

The superiority of the result obtained from the GPR architecture, particularly for site 4, is clearly evident, even when represented graphically. Notably, the Random Forest architecture's incongruity with the task (as evidenced in figure 19) further accentuates its unsuitability, so its value can be discarded.

Table 2 displays the architecture-wise results, highlighting the group with the most minimal RMSE, relative at the inference phase, for each configuration.

Machine Learning Algorithm	Group	RMSE Inference Phase
Linear	4	0.043
SVM	4	0.043
Random Forest	4	0.030
Ensemble	1	0.038
Multi-Layers	1	0.038
GPR	4	0.031
Kernel	4	0.035

Table 2: Minimal RMSEs for each architecture

Table 2 underscores the discernible advantage held by groups 1 and 4, offering databases that excel in supporting various models use cases, outshining their counterparts. Although the RMSE values range from 0.03 to 0.045, which doesn't quite reach the optimal range, they still achieve a satisfactory level of performance.

The GPR architecture emerges as the best one, boasting an unequivocal RMSE low of 0.031, relative at the group 4. This result can be deemed highly satisfactory.

In the appendix plots, particularly within the training phase of the Gaussian Process Regression model (figure 22), a distinct manifestation of over-fitting becomes evident, even visible in figure 14a, in group 2, 3 and 5. This arises as the network excessively tailors itself to the available data, driven by an abundance of parameters compared to the limited observations. Despite this side effect, the model manages to maintain its superiority in terms of performance during tests also in the inference phase, as evident in table 2.

Once the 'best' group was identified, looking at the minimum RMSE, a comparison was conducted with a simple linear regression, using the dataset associated with it (group 4). Table 3 displays the differences in RMSE, between the minimal one (GPR Model) and the linear regressions of single input (VH and VV) against the moisture. It is evident that a simple linear regression of the individual input alone is not satisfactory, justifying the use of the selected machine learning models.

GPR Model	L.R. VH-SM	L.R. VV-SM
0.031	0.046	0.042

Table 3: Optimized Models vs Linear Regression

Upon scrutinizing the presented plots, it becomes evident that even in cases like the ones observed in groups 3 and 5, the incorporation of supplementary inputs (VH and NDVI) to the single correlation VV-SM helps reasonably the moisture predictions.
However, the most favorable outcomes materialize predominantly within group 4, and to a somewhat lesser degree in group 1. In these instances, the correlations are visibly pronounced (positive for the former and negative for the latter), discernible even without sophisticated analysis, encompassing both VV and VH data.

Nevertheless, it remains crucial to emphasize that attaining these values relies on training and inference architectures with site-specific data. Employing an architecture trained on one area and tested on another fails to produce satisfactory outcomes. In order to predict the soil moisture of a given region, it is necessary to have access to historical data for that region. However, it cannot be assumed that every region exhibits a correlation, as is evident in the cases of groups 3 and 5 of this study. Therefore, the creation of software that can be used universally in every part of the globe is not possible.

7. Conclusions

In conclusion, as previously expounded upon in the introductory section (see section [1]), the fundamental objective of this research was to analyze the intricate dataset procured from the Sentinel Program's satellites and to create a globally applicable, universally adaptable tool. The goal was to delve into the dataset's depths and unearth potential correlations that may exist between the observations gleaned from these satellites and the levels of soil moisture content found within the diverse regions under investigation.

Upon examination of the minimum RMSE values, outlined in table 2, it becomes evident that these values, while moderately satisfactory, exhibit inherent variability only across individual sites. This variability is a crucial consideration due to the site-specific nature of RMSE values during the training and inference stages of each model. This underscores the infeasibility of training a model in one location to predict soil moisture in another, different one.

This observation aligns with the insights gleaned from figure 16 in appendix A.2, where the strong site-dependent correlation between input parameters (VH and VV polarizations) and output (soil moisture) is visually evident across distinct color-coded groups. This correlation, particularly pronounced in VV polarization, highly responsive to moisture changes, showcases negative correlation in the first group, neutrality in groups 3 and 5, and positive correlation in groups 2 and 4. These trends are further substantiated in the parity plots expounded in section A.3 of the appendix, outlining the results of linear regression.

Significantly, the introduction of supplementary inputs (VH and NDVI) presents a scenario where discrete outcomes seemed improbable (groups 3 and 5). However, tangible results are still achievable, as exemplified in figures in section A.6 pertaining to the ensemble for decision tree model.

It is important to highlight the notable inadequacy of the Random Forest model in both the training and predictive aspects of its model architecture. The graphs elucidated in appendix A.5 provide unmistakable evidence of the model's inefficacy, as it manages to generate predictions for just some values, regardless of the specific values it endeavors to approximate.

This performance can potentially be attributed to several contributing factors. One could be the relatively limited size of the dataset employed for training, as the Random Forest model typically excels with more extensive datasets. As a consequence, while the reported value in table 2 might appear to be the best in absolute terms for this model type, it is essentially a result of a 'flaw' in the code, rendering it incapable of visually detecting instances when the network is evidently malfunctioning.
Consequently, the 'Random Forest' architecture proves to be incapable of providing meaningful assistance in this context, highlighting its inherent limitations in this scenario.

Another noteworthy observation pertains to the MLP architecture. The graphs in appendix A.7 distinctly portray how the network, within each group, is proficient at predicting values only surpassing a certain threshold. This recurring pattern might indeed originate from the relatively constrained dataset size, underscoring the contrast with the expansive datasets typically leveraged for such architectures.

In contrast, the GPR architecture emerges as prominently advantageous across nearly all examined sites. Its capability to discern correlations and patterns with fewer data points, compared to what conventional models necessitate for accurate modeling, contributes to its superiority.

The examination of table 2 brings to light that the sites with the most favorable RMSE values are the first and the fourth. This outcome aligns with the visual analysis of Figure 16, where both group 1 and group 4 demonstrate correlations manifested through steeper regression lines. This correlation is particularly pronounced in the context of VV polarization, but also extends to VH polarization.

In summarizing the findings from this undertaken study, it is evident that the utilization of machine learning algorithms can yield moderately satisfactory results in the realms of soil moisture data training and prediction. However, these achievements are notably constrained **only** to a '*local*' context, a factor not documented in any literary source. Presently, the ambition of training a single model applicable across diverse global locations remains unattainable, diverging from the initial overarching goal of this endeavor.

Nevertheless, it's important to recognize that this conclusion bears a positive undertone, as it establishes the practicability of implementing these methodologies at a local scale. For instance, when armed with existing moisture data from a specific region and armed with a substantial pool of samples, the process outlined in this study can be replicated. This replication could effectively facilitate the estimation of moisture levels for that designated area, obviating the necessity for intrusive sensors and exhaustive measurements.

Indeed, there lie ample prospects for augmenting the undertaken work. A notably promising avenue involves diverging from the reliance on ISMN data for in-situ measurements. Instead, one could opt for the collection of self-curated data from delineated and precisely tailored areas. This approach, akin to that pursued by the authors of the article [42–44], has the potential to yield richer and more contextually relevant datasets.

Moreover, the expansion of the operating dataset could encompass the assimilation of data from Sentinel-1 and Sentinel-2 satellites, encompassing diverse relative orbits. This augmentation should also consider the angle of incidence, an aspect remained unexplored in this study. This because all satellite data were stemmed from a uniform orbit, making the introduction of this variable somewhat redundant, as it would have remained constant for all observations.

Lastly, advancing the methodology could involve the integration of more sophisticated machine learning models, emphasizing each hyperparameter's role. This entails discerning which hyperparameters to optimize and which to withhold from optimization, fostering a more refined and accurate modeling process.

Certainly, with these available data, namely Sentinel1's SAR and Sentinel2's optical data to retrieve VH and VV polarizations along with NDVI, coupled with data extracted from the ISMN hub, it appears unfeasible to create a general model capable of being trained on a region for which historical data is available and subsequently applied to any other site.

The only viable operation, nonetheless of utmost significance, that could lead to significant advancements in this field, remains the training of models on a specific network and subsequently predicting values within the **same** network.

Undoubtedly, the pursuit of harmonizing two seemingly divergent domains, agriculture and space science, via the infusion of AI, a force present in all realms, stands as a commendable accomplishment. This study harbors the power to inaugurate a sequence of research endeavors and partnerships among universities and scholars who share a mutual aspiration: to augment and propel advancements across a spectrum of disciplines. This synergy bears the potential to drive innovation and catalyze progress in unanticipated ways.

References

[1] European Space Agency. Copernicus open hub access. https://scihub.copernicus.eu/dhus/#/home.

[2] European Space Agency. European space agency sentinel application platform. https://earth.esa.int/eogateway/tools/snap.

[3] European Space Agency. European space agency website. https://www.esa.int/.

[4] Balenzano, Mattia, Satalino, Lovergine, Palmisano, and Davidson. Dataset of sentinel-1 surface soil moisture time series at 1 km resolution over southern italy. *Elsevier Inc.*, 2021.

[5] Bengio and Courville. *A Review of Machine Learning and Deep Learning Applications*. IEEE, 2018.

[6] Bourbigot, Hajduch, Johnsen, and Piantanida. Sentinel-1 product specification. *ESA Unclassified – For Official Use*, 2022.

[7] Courrech, De Gaujac, Naud, and Provenzano. Sentinel-2 products specification document. *ESA Unclassified – For Official Use*, 2021.

[8] Dorigo, Wagner, Hohensinn, Hahn, Paulik, Xaver, Gruber, Draper, Mecklenburg, and van Oevelen. The international soil moisture network: a data hosting facility for global in situ soil moisture measurements. *Hydrology and Earth System Sciences*, 2011.

[9] Du, Zhang, Li, and Zhang. Deep learning for classification of hyperspectral data: A comparative review. In *IEEE Transactions on Geoscience and Remote Sensing*, 2018.

[10] Entekhabi, Njoku, O'Neill, Kellogg, Crow, Edelstein, Entin, Goodman, Jackson, Johnson, et al. The soil moisture active passive (smap) mission. *Proceedings of the IEEE*, 2010.

[11] Gallegos, Rivera, and Estevez. Neural networks for remote sensing of soil moisture. In *Proceedings of the 5th International Conference on Advances in Image Processing*, 2020.

[12] Goodfellow, Bengio, and Courville. *Deep Learning*. MIT Press, 2016.

[13] Google. Google earth. https://www.google.it/intl/it/earth/.

[14] Hong, Wu, Zhang, Hong, and Pan. Multi-scale recurrent convolutional neural network for soil moisture retrieval from smap data. *Remote Sensing*, 2019.

[15] Jackson and Cosh. Microwave remote sensing of soil moisture: Recent advances, challenges, and applications. *Vadose Zone Journal*, 2019.

[16] James, Witten, Hastie, and Tibshirani. *An introduction to statistical learning*. Springer Science+Business Media, 2021.

[17] Kerr, Waldteufel, Wigneron, Delwart, Cabot, Boutin, Escorihuela, Font, Reul, Gruhier, et al. The smos mission: New tool for monitoring key elements of the global water cycle. *Proceedings of the IEEE*, 2010.

[18] ktmagar's YT [Youtube Channel]. Geocoding and rtc processing of sentinel-1 data in snap + asf hyp3. https://www.youtube.com/watch?v=0n99UlhQFZg.

[19] LeCun, Jackel, Bottou, Brunot, Cortes, Denker, Drucker, Guyon, Muller, and Säckinger. Backpropagation applied to handwritten zip code recognition. In *Neural computation*. MIT Press, 1989.

[20] Malhotra, Panigrahi, Sharma, and Garg. A novel deep learning framework for improved soil moisture retrieval using sentinel-1 sar data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2021.

[21] Malhotra, Sharma, Panigrahi, and Kumar. Soil moisture estimation using sentinel 1 sar imagery and deep learning. *Remote Sensing*, 2019.

[22] MathWorks. Fitrensemble documentation. https://www.mathworks.com/help/stats/fitrensemble.html.

[23] MathWorks. Fitrgp documentation. https://www.mathworks.com/help/stats/fitrgp.html.

[24] MathWorks. Fitrkernel documentation. https://www.mathworks.com/help/stats/fitrkernel.html.

[25] MathWorks. Fitrlinear documentation. https://www.mathworks.com/help/stats/fitrlinear.html.

[26] MathWorks. Fitrnet documentation. https://www.mathworks.com/help/stats/fitrnet.html.

[27] MathWorks. Fitrsvm documentation. https://www.mathworks.com/help/stats/fitrsvm.html.

[28] MathWorks. Fitrtree documentation. https://www.mathworks.com/help/stats/fitrtree.html.

[29] Merzouki, Prigent, Cabot, and Zribi. Deep learning for soil moisture estimation from passive microwave satellite observations. *International Geoscience and Remote Sensing Symposium*, 2019.

[30] Grow Network. Grow data portal. https://growobservatory.org/index.html.

[31] International Soil Moisture Network. Ismn data access point. https://ismn.earth/en/.

[32] OzNet Network. Oznet data portal. https://www.oznet.org.au/.

[33] Tahmo Network. Tahmo data portal. https://tahmo.org/.

[34] WegenerNet Network. Wegenernet data portal. https://wegenernet.org/portal/.

[35] Nielsen. *Neural Networks and Deep Learning: A Textbook*. Determination Press, 2015.

[36] Bureau of Economic Geology. Texas soil observation network website. https://www.beg.utexas.edu/research/programs/txson.

[37] Rasmusson. *Climate monitoring and diagnostics laboratory. World climate programme*. World Meteorological Organization, 1994.

[38] Reichle, Koster, and De Lannoy. A review of global soil moisture observations: Synthesis and recommendations for earth system modeling. *Hydrology and Earth System Sciences*, 2019.

[39] Rodell, Houser, Jambor, Gottschalck, Mitchell, Meng, Arsenault, Cosgrove, Radakovich, Bosilovich, et al. The global land data assimilation system. *Bulletin of the American Meteorological Society*, 2004.

[40] Samadzadegan, Rafiei-Sarmazdeh, and Homayouni. Soil moisture estimation using sentinel-1 sar data and deep learning techniques. In *International Geoscience and Remote Sensing Symposium*, 2020.

[41] Campbell Scientific. Campbell scientific cs655 soil moisture sensor manual. https://psl.noaa.gov/data/obs/instruments/CampellSciSoilMositureCS650.pdf.

[42] Singh and Gaurav. Deep learning and data fusion to estimate surface soil moisture from multi-sensor satellite images. *Scientific Reports*, 2023.

[43] Singh, Gaurav, and Naik. Drainage congestion due to road network on the kosi alluvial fan, himalayan foreland. *International Journal of Applied Earth Observations and Geoinformation*, 2022.

[44] Singh, Gaurav, Naik, and Meena. Estimation of soil moisture applying modified dubois model to sentinel-1; a regional study from central india. *Remote Sensing*, 2020.

[45] Wagner, Blöschl, Pampaloni, Calvet, Bizzarri, and Wigneron. Operational readiness of microwave remote sensing of soil moisture for hydrologic applications. *Nordic Hydrology*, 2007.

[46] Wang, Zhang, Du, and Wu. Soil moisture retrieval using sentinel-1 sar data and a deep learning approach. *Remote Sensing*, 2020.

[47] Wang, Zhang, Du, and Wu. Soil moisture retrieval using dual-polarization sentinel-1 data and a deep learning approach. *Remote Sensing*, 2021.

[48] Yuan, Yuan, Tang, and Zhu. Soil moisture retrieval based on sentinel-1 sar data and convolutional neural network. In *International Conference on Measuring Technology and Mechatronics Automation*, 2020.

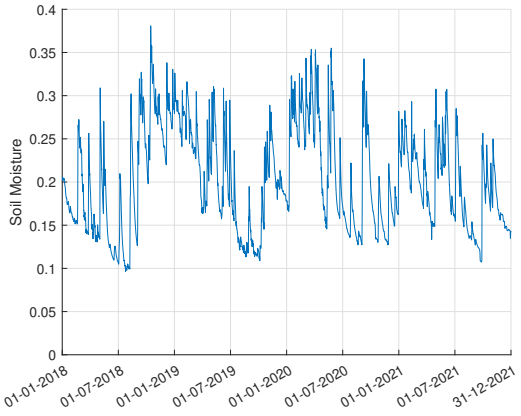
[49] Zappa, Schlaffer, Bauer-Marschallinger, Nendel, Zimmerman, and Dorigo. Detection and quantification of irrigationwater amounts at 500 m using sentinel-1 surface soil moisture. *Remote Sensing*, 2021.

[50] Zhang, Zhang, Du, and Wu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017.

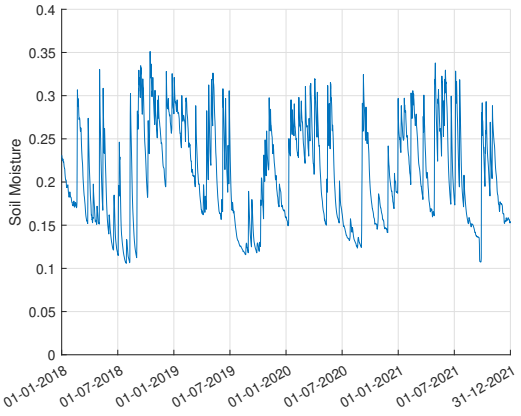
A. Appendix A

A.1. Soil Moisture Time Series

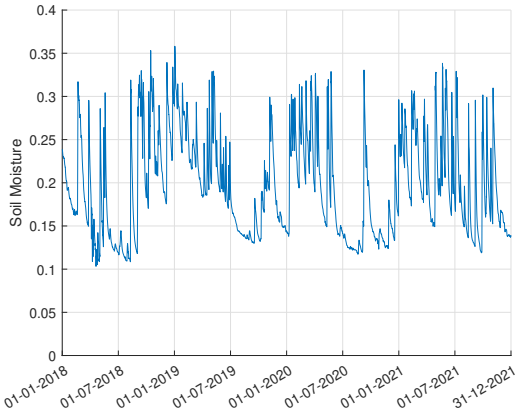
In the following plots, it is possible to observe the time series data spanning from January 1, 2018, to December 31, 2021, for each of the five identified regions. These time series consist of daily values, representing the average values from each station within the respective group. A discernible pattern emerges, characterized by peaks followed by declining segments. These peaks correspond to instances of atmospheric precipitation, succeeded by periods of clear weather during which the terrain undergoes drying.



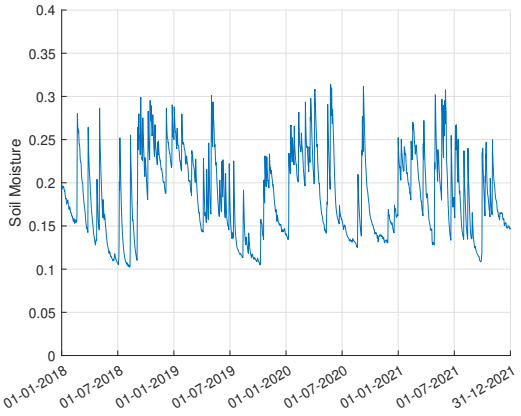
(a) 1^{st} Group



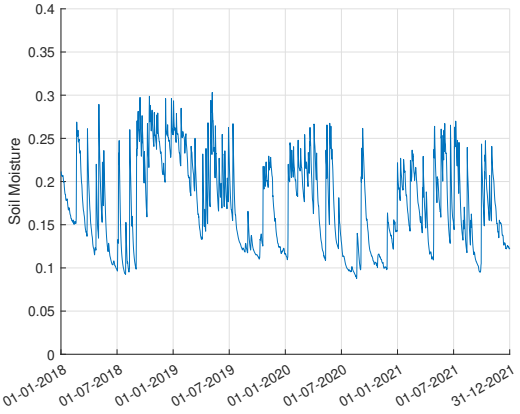
(b) 2^{nd} Group



(c) 3^{rd} Group



(d) 4^{th} Group



(e) 5^{th} Group

Figure 15: Groups Time Series

A.2. Input vs Output Dependencies

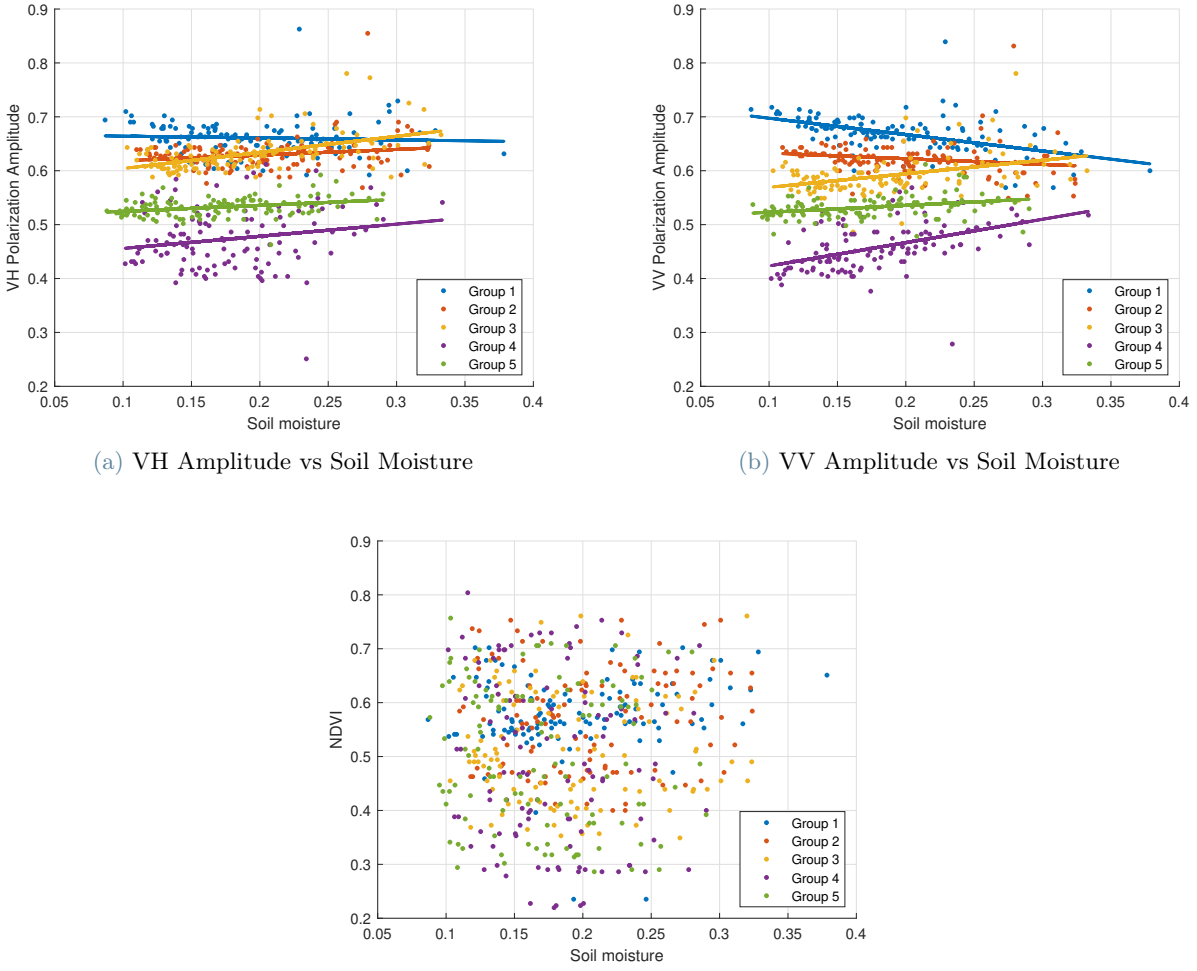
Within this subsection, the graphical depictions at hand allow for the observation of correlations among the three fundamental inputs of the machine learning algorithms, namely, VH and VV polarization amplitudes, along with NDVI values, contrasted against the resulting output denoting soil moisture.

It is clear that discerning patterns within specific groups reveals certain dependencies. Yet, when examining the entire database indiscriminately, the complexity significantly deepens. Consequently, adopting a strategic approach by addressing individual station groups consecutively is the most prudent path forward during the ML phase.

In the initial two graphs representing VH and VV amplitudes, additional regression lines have been integrated for each of the five groups. These lines effectively illustrate the qualitative nature of the correlation between the input (VH or VV) and the output (soil moisture), whether it is positive, neutral, or negative.

Upon examination, the first VH plot reveals a generally minor positive reliance across all groups, barring the first group which exhibits a mostly neutral to slightly negative correlation. Conversely, in the subsequent VV plot, the first group distinctly showcases a negative dependency, the second and fourth groups indicate a positive correlation, whereas the third and fifth groups seem to exhibit nearly neutral correlations.

As for the NDVI aspect, the omission of regression lines is deliberate due to the notably intricate nature of the overall pattern. Unlike VH and VV, which display distinct correlations within certain groups, the NDVI data lacks specific discernible patterns across the examined groups, rendering the incorporation of regression lines less meaningful in this context. However, this value is subsequently helpful as input in training the model to achieve better results.



(a) VH Amplitude vs Soil Moisture

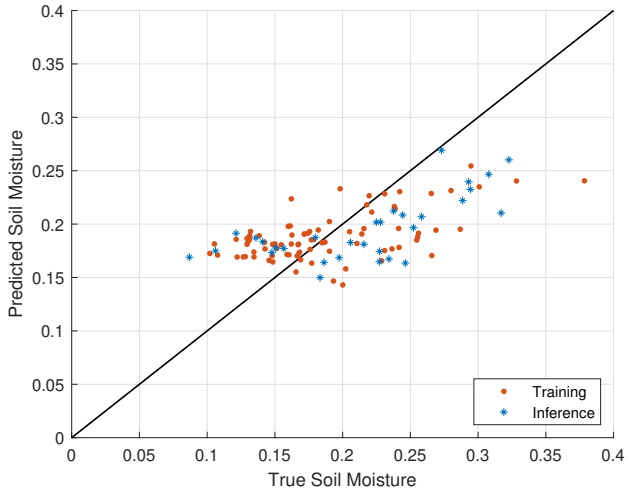
(b) VV Amplitude vs Soil Moisture

(c) NDVI vs Soil Moisture

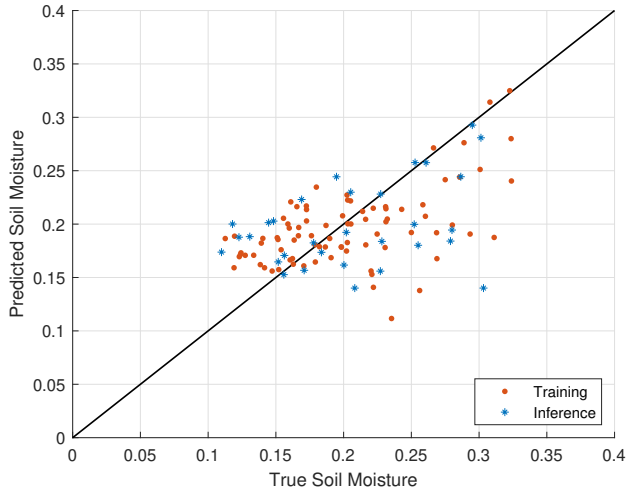
Figure 16: Input vs Output Dependencies

A.3. Linear Regression Model Results

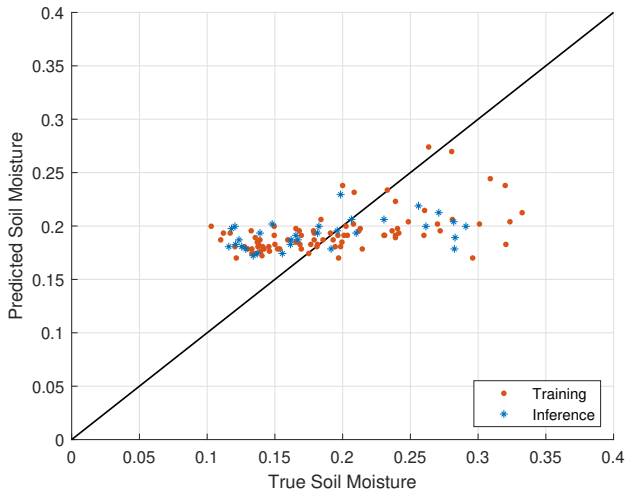
Within this section, parity plots pertaining to the linear regression model can be observed. These plots illustrate the predicted values against the true ones between training and inference phases across the five established groups.



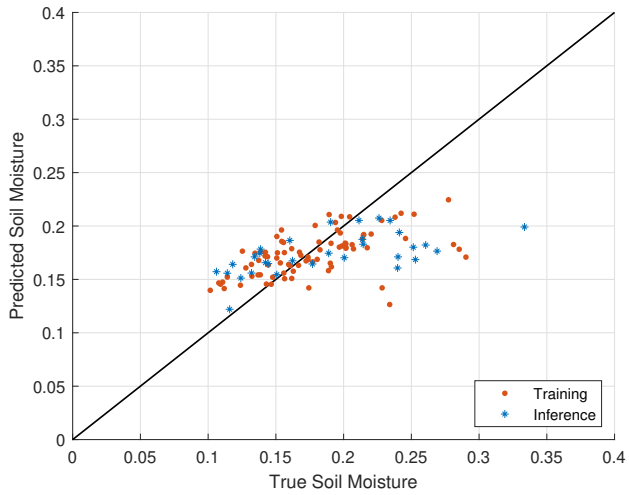
(a) 1^{st} Group



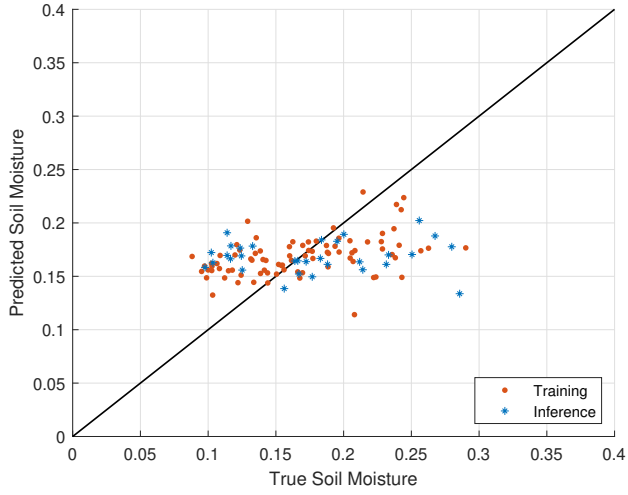
(b) 2^{nd} Group



(c) 3^{rd} Group



(d) 4^{th} Group



(e) 5^{th} Group

Figure 17: Linear Regression Model Parity Plots

By employing the 'fitrlinear' function, for each of the five groups, the hyperparameters were established and listed in the table 4.

Group	Lambda	Learner	Regularization
1	1.8499e-07	'leastsquares'	'ridge'
2	0.00037705	'svm'	'ridge'
3	0.00058512	'leastsquares'	'ridge'
4	0.0047878	'svm'	'ridge'
5	2.7948e-05	'leastsquares'	'lasso'

Table 4: Optimized Hyperparameters for Linear Regression Model

The following table 5 presents the RMSE corresponding to the various stages of network training, namely the training and inference phase.

Group	RMSE Training Phase	RMSE Inference Phase
1	0.040	0.051
2	0.050	0.045
3	0.051	0.053
4	0.037	0.043
5	0.045	0.055

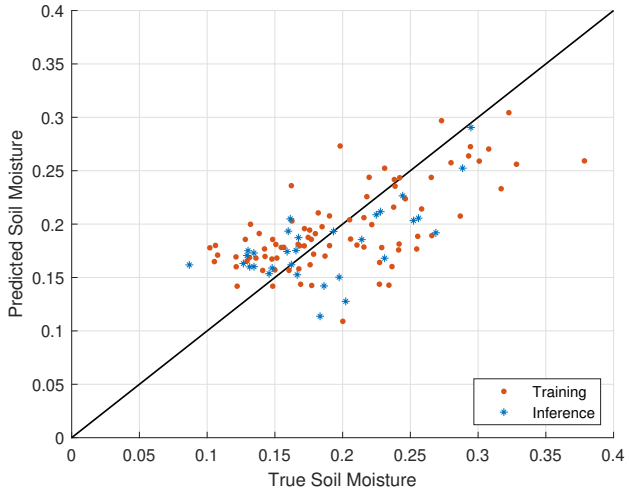
Table 5: RMSEs for Linear Regression Model

In the graphs 17, some promising correlations are beginning to emerge; however, it is undoubtedly necessary to employ more sophisticated and robust models in order to achieve improved results.

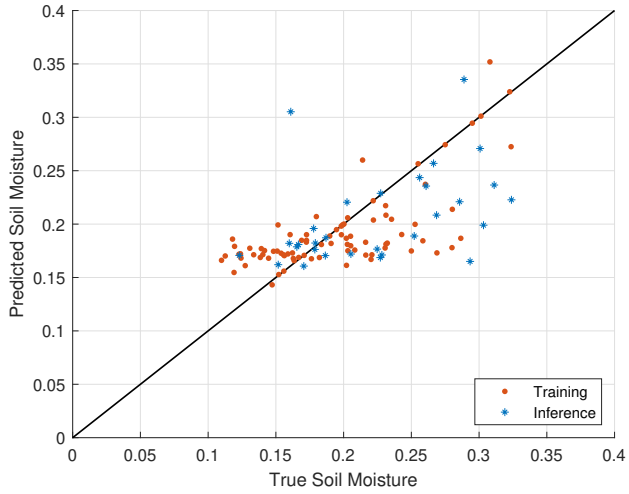
In Table 4 it is shown how the regularization function varies from site to site, emphasizing the previously mentioned 'site-specific' nature of the correlations. Finding general hyperparameters is not feasible in this context.

A.4. Support Vector Machine Regression Model Results

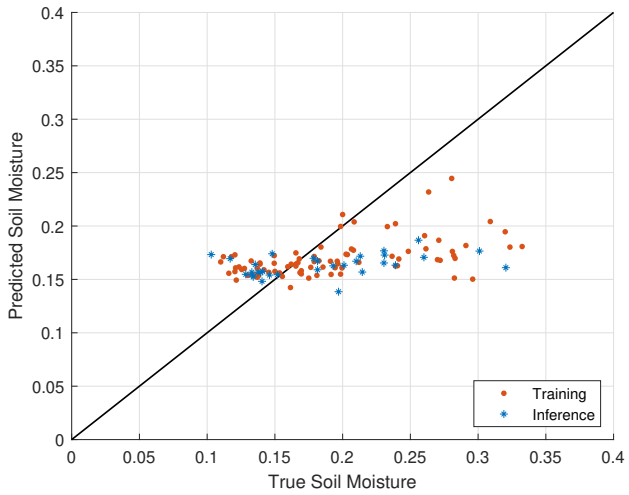
Within this section, parity plots pertaining to the SVM regression model can be observed. These plots illustrate the predicted values against the true ones between training and inference phases across the five established groups.



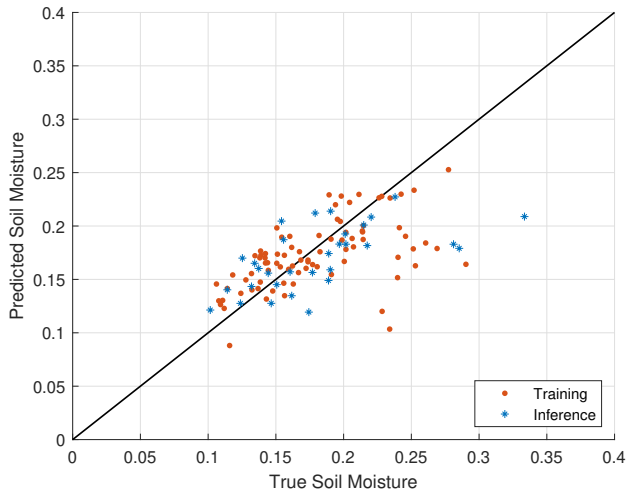
(a) 1^{st} Group



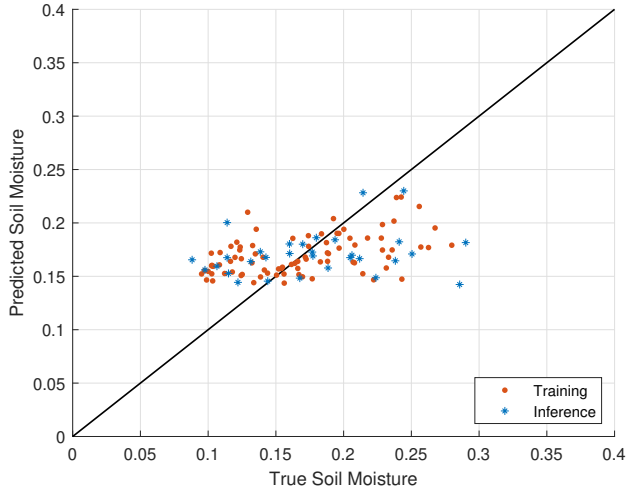
(b) 2^{nd} Group



(c) 3^{rd} Group



(d) 4^{th} Group



(e) 5^{th} Group

Figure 18: SVM Regression Model Parity Plots

By employing the 'fitrsvm' function, for each of the five groups, the hyperparameters were established and listed in the table 6.

Group	BoxContraint	KernelScale	Epsilon	KernelFunction	PolynomialOrder	Standarize
1	0.58497	NaN	6.7827e-05	'linear'	NaN	'true'
2	0.69526	NaN	0.021047	'polynomial'	3	'false'
3	0.57301	NaN	0.0052056	'linear'	NaN	'true'
4	27.291	NaN	0.014181	'linear'	NaN	'false'
5	10.513	NaN	0.00023875	'linear'	NaN	'false'

Table 6: Optimized Hyperparameters for Support Vector Machine Regression Model

The following table 7 presents the RMSE corresponding to the various stages of network training, namely the training and inference phase.

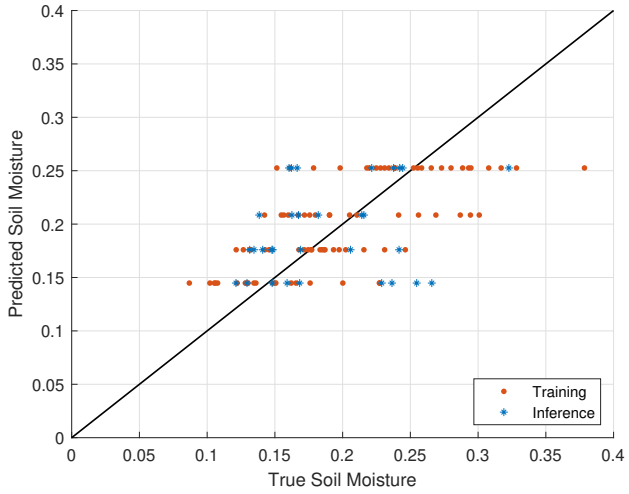
Group	RMSE Training Phase	RMSE Inference Phase
1	0.044	0.049
2	0.047	0.052
3	0.050	0.055
4	0.037	0.043
5	0.048	0.051

Table 7: RMSEs for Support Vector Machine Regression Model

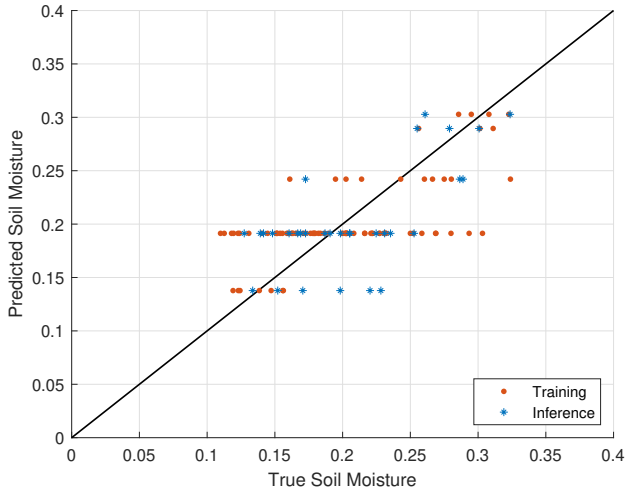
Similarly to the previous case, here too discrete results are provided, but yet far from the desired objective. Furthermore, it is evident that hyperparameters vary from site to site in this context as well.

A.5. Random Forest Regression Model Results

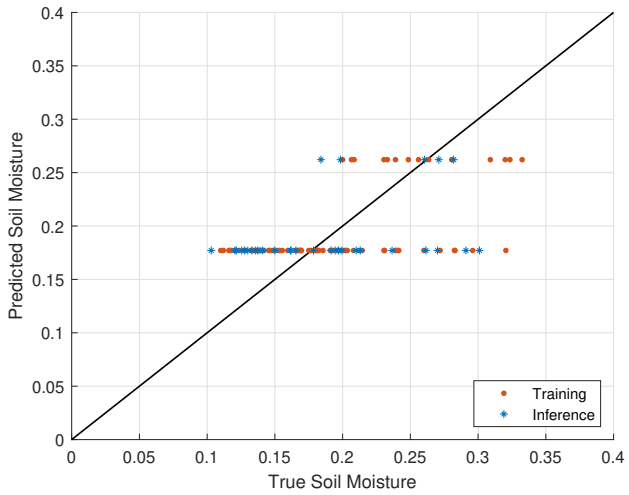
Within this section, parity plots pertaining to the Random Forest regression model can be observed. These plots illustrate the predicted values against the true ones between training and inference phases across the five established groups.



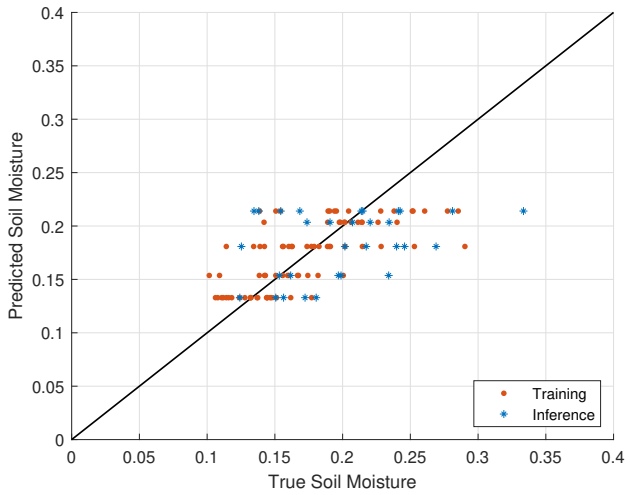
(a) 1^{st} Group



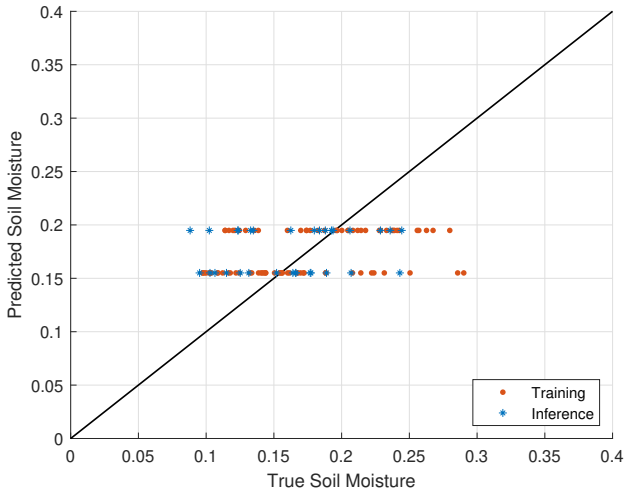
(b) 2^{nd} Group



(c) 3^{rd} Group



(d) 4^{th} Group



(e) 5^{th} Group

Figure 19: Random Forest Regression Model Parity Plots

By employing the 'fitrtree' function, for each of the five groups, the hyperparameters were established and listed in the table 8.

Group	MaxNumSplits	MinLeafSize	NumVar
1	17	6	3
2	5	5	3
3	2	2	3
4	31	8	3
5	32	18	3

Table 8: Optimized Hyperparameters for Random Forest Regression Model

The following table 9 presents the RMSE corresponding to the various stages of network training, namely the training and inference phase.

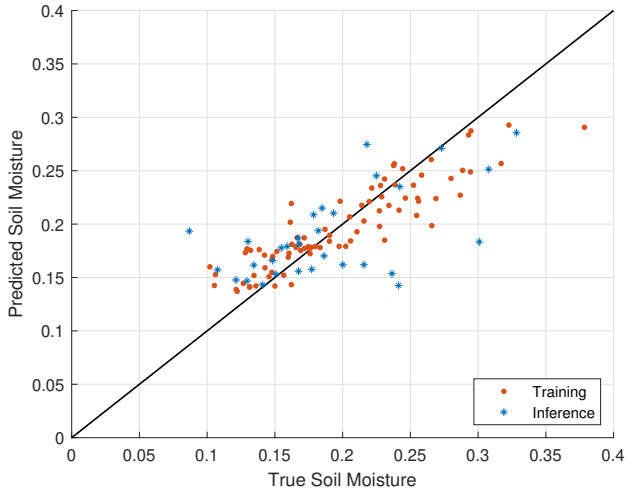
Group	RMSE Training Phase	RMSE Inference Phase
1	0.045	0.056
2	0.040	0.043
3	0.045	0.051
4	0.042	0.030
5	0.050	0.043

Table 9: RMSEs for Random Forest Regression Model

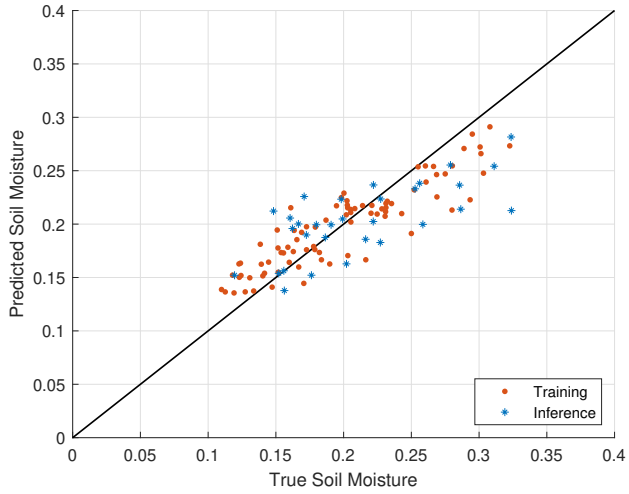
Observing figure 19, it is evident that this model is not functioning as expected. It can only predict just some values, regardless of the input data. This is likely due to the relatively small amount of data, as this model is accustomed to and more efficient with large datasets. Therefore, despite the lowest RMSE value in overall (0.03 relative to group 4) being present here, it can be safely discarded.

A.6. Ensemble of Learners Regression Model Results

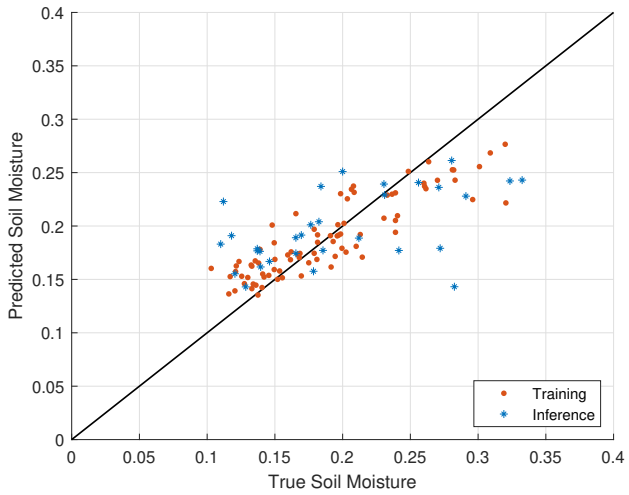
Within this section, parity plots pertaining to the ensemble of learners regression model can be observed. These plots illustrate the predicted values against the true ones between training and inference phases across the five established groups.



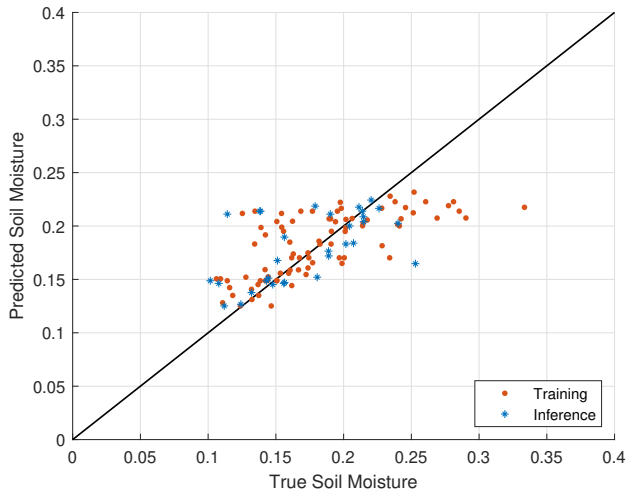
(a) 1^{st} Group



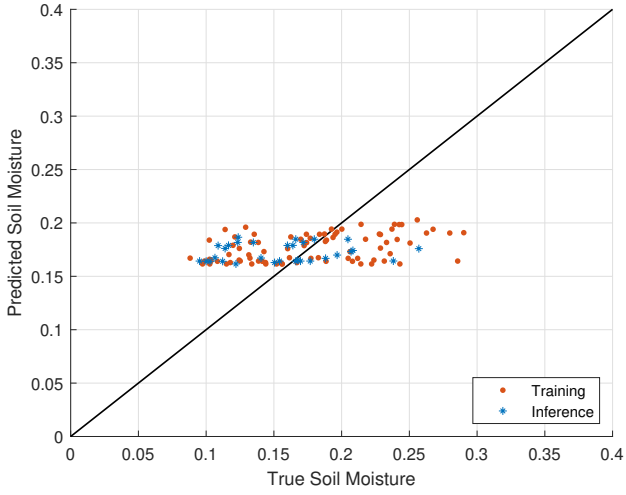
(b) 2^{nd} Group



(c) 3^{rd} Group



(d) 4^{th} Group



(e) 5^{th} Group

Figure 20: Ensemble of Learners Regression Model Parity Plots

By employing the 'fitrensemble' function, for each of the five groups, the hyperparameters were established and listed in the table 10.

Group	Method	NumLearnCycles	LearnRate	MinLeafSize	MaxNumSplits	NumVar
1	'LSBoost'	138	0.16062	1	1	2
2	'Bag'	69	NaN	1	80	2
3	'Bag'	11	NaN	2	81	3
4	'LSBoost'	38	0.086582	1	5	3
5	'Bag'	10	NaN	21	1	2

Table 10: Optimized Hyperparameters for Ensemble of Learners Regression Model

The following table 11 presents the RMSE corresponding to the various stages of network training, namely the training and inference phase.

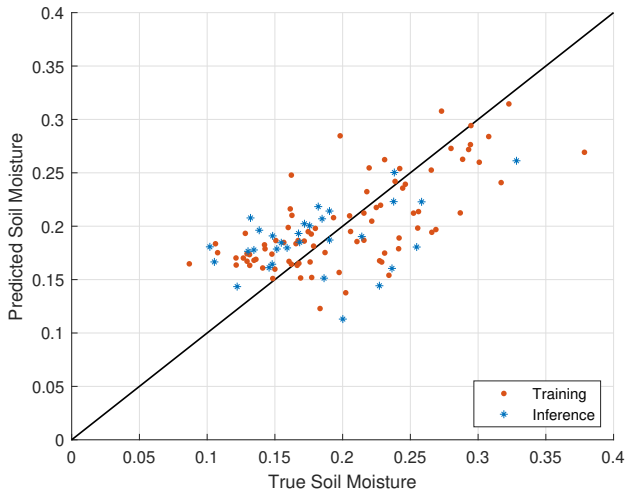
Group	RMSE Training Phase	RMSE Inference Phase
1	0.033	0.038
2	0.028	0.040
3	0.027	0.072
4	0.022	0.040
5	0.048	0.046

Table 11: RMSEs for Ensemble of Learners Regression Model

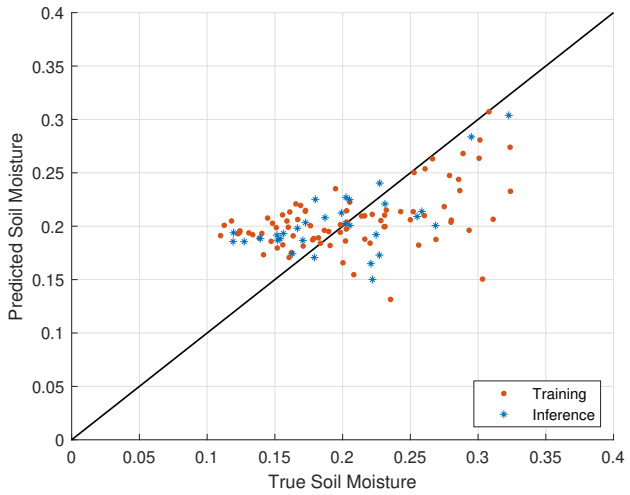
This model demonstrates promising results, particularly in groups 1, 2, and 4, as evident in Table 11 and Figure 20. These outcomes align with expectations, given that groups 3 and 5 did not exhibit visible correlations even at a cursory examination.

A.7. Multi-Layer Perceptron Regression Model Results

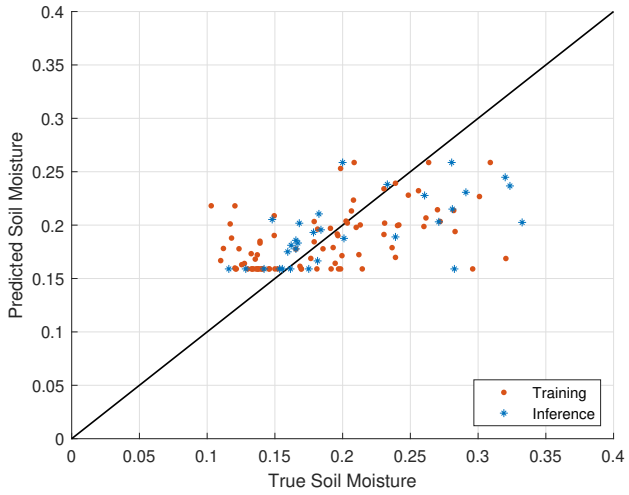
Within this section, parity plots pertaining to the MLP model can be observed. These plots illustrate the predicted values against the true ones between training and inference phases across the five established groups.



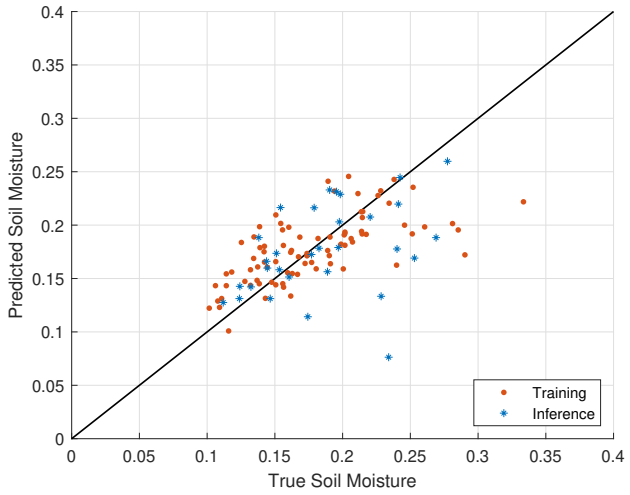
(a) 1^{st} Group



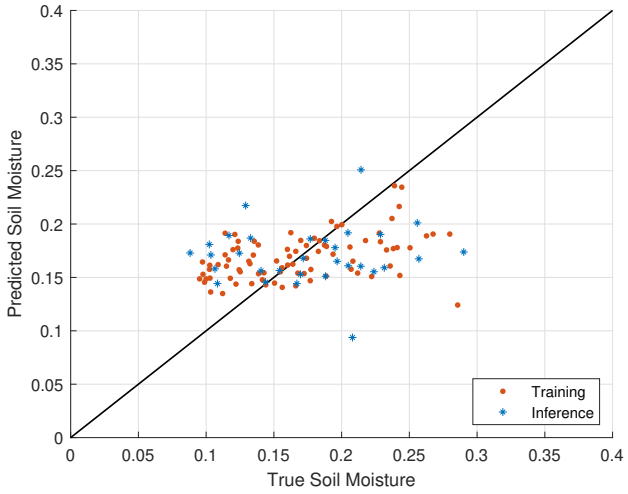
(b) 2^{nd} Group



(c) 3^{rd} Group



(d) 4^{th} Group



(e) 5^{th} Group

Figure 21: MLP Model Parity Plots

By employing the 'fitrnet' function, for each of the five groups, the hyperparameters were established and listed in the table 12.

Group	Activation	Standardize	Lambda	LayerWeights	LayerBiases	LayerSizes
1	'sigmoid'	'true'	6.2816e-06	'glorot'	'zeros'	[1 3]
2	'relu'	'true'	0.016303	'glorot'	'ones'	[2]
3	'none'	'false'	6.2626e-05	'glorot'	'zeros'	[169]
4	'sigmoid'	'true'	0.00020994	'he'	'zeros'	[1]
5	'relu'	'true'	0.0031493	'glorot'	'zeros'	[227 8]

Table 12: Optimized Hyperparameters for Multi-Layer Perceptron Regression Model

The following table 13 presents the RMSE corresponding to the various stages of network training, namely the training and inference phase.

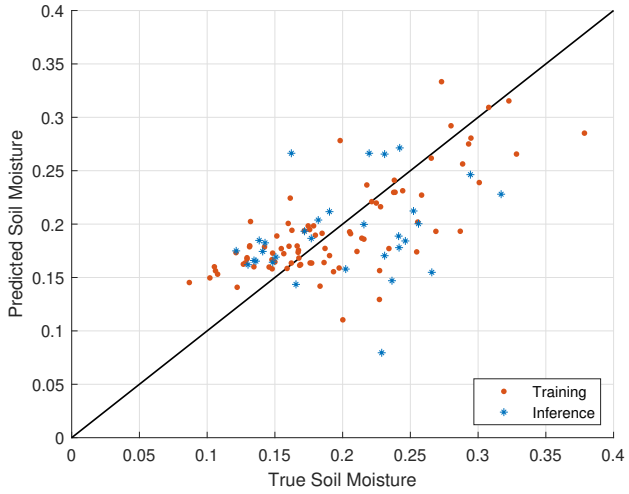
Group	RMSE Training Phase	RMSE Inference Phase
1	0.040	0.038
2	0.044	0.052
3	0.050	0.048
4	0.036	0.046
5	0.047	0.046

Table 13: RMSEs for Multi-Layer Perceptron Regression Model

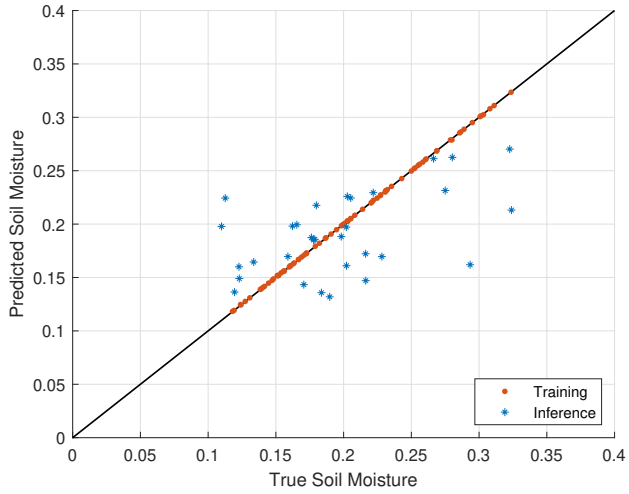
The various networks identified through the optimization of this model all share a common issue: they are unable to predict values below a certain threshold, as if there were a virtual barrier (figure 21). This phenomenon could be attributed to the relatively small amount of data, as this family of models is better suited to handling large volumes of data, similar to what was observed in the Random Forest architecture (see A.5).

A.8. Gaussian Process Regression Model Results

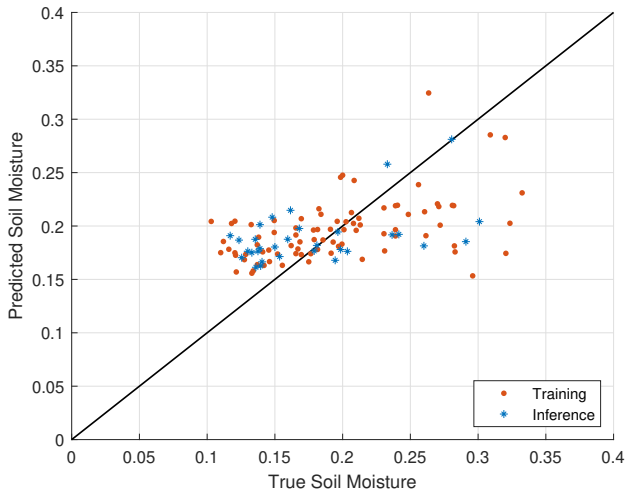
Within this section, parity plots pertaining to the Gaussian Process Regression model can be observed. These plots illustrate the predicted values against the true ones between training and inference phases across the five established groups.



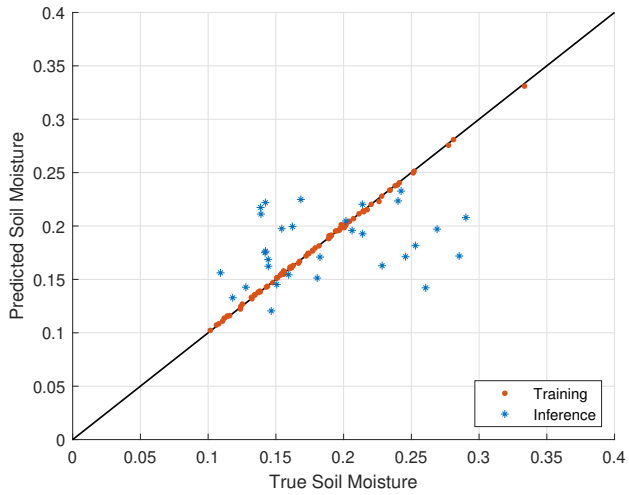
(a) 1^{st} Group



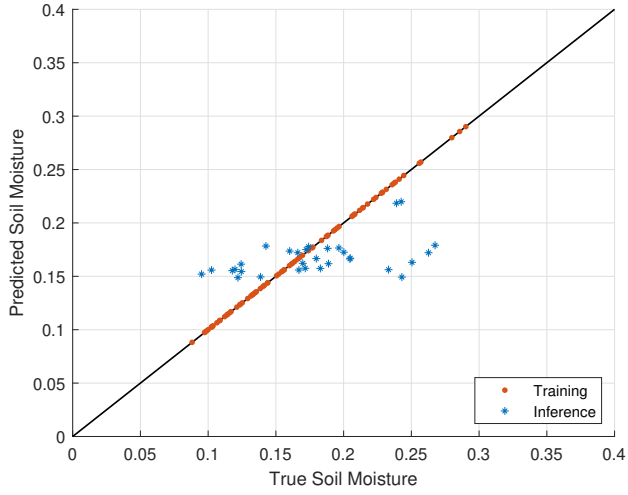
(b) 2^{nd} Group



(c) 3^{rd} Group



(d) 4^{th} Group



(e) 5^{th} Group

Figure 22: Gaussian Process Regression Model Parity Plots

By employing the 'fitrgp' function, for each of the five groups, the hyperparameters were established and listed in the table 14.

Group	Sigma	BasisFunction	KernelFunction	KernelScale	Standardize
1	0.09241	'linear'	'ardmatern32'	NaN	'false'
2	0.0076331	'linear'	'exponential'	0.00092669	'false'
3	0.00010204	'none'	'ardrationalquadratic'	NaN	'false'
4	0.015045	'linear'	'matern32'	0.024048	'true'
5	0.00010414	'none'	'rationalquadratic'	0.0011248	'true'

Table 14: Optimized Hyperparameters for Gaussian Process Regression Model

The following table 15 presents the RMSE corresponding to the various stages of network training, namely the training and inference phase.

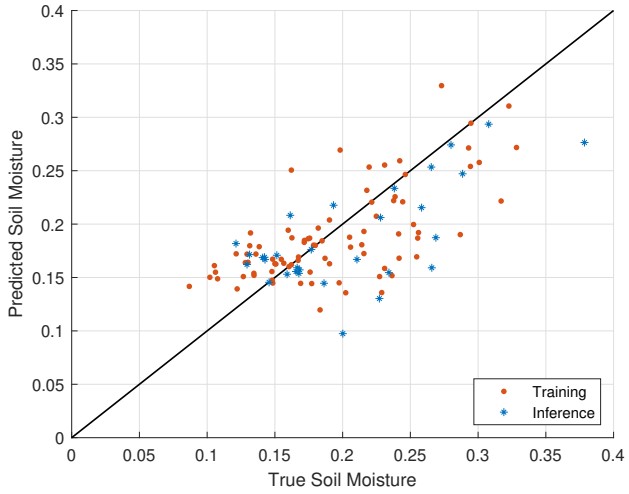
Group	RMSE Training Phase	RMSE Inference Phase
1	0.045	0.038
2	0.001	0.054
3	0.00002	0.047
4	0.005	0.031
5	0.00002	0.052

Table 15: RMSEs for Gaussian Process Regression Model

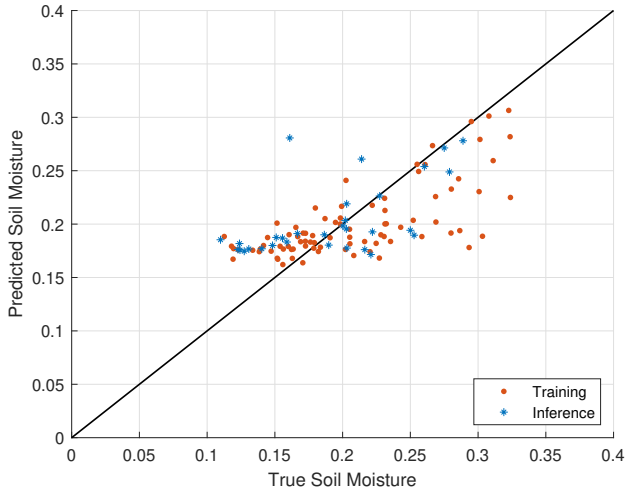
Figure 22 exhibits clear signs of overfitting across virtually all groups, as visually evident in Figure 14a. Nevertheless, the optimized model for group 4 stands out as the overall best performer with a value of 0.031. The success of this architecture can be attributed to its ability to identify patterns and relationships even with relatively small quantities of data.

A.9. Gaussian Kernel Regression Model Results

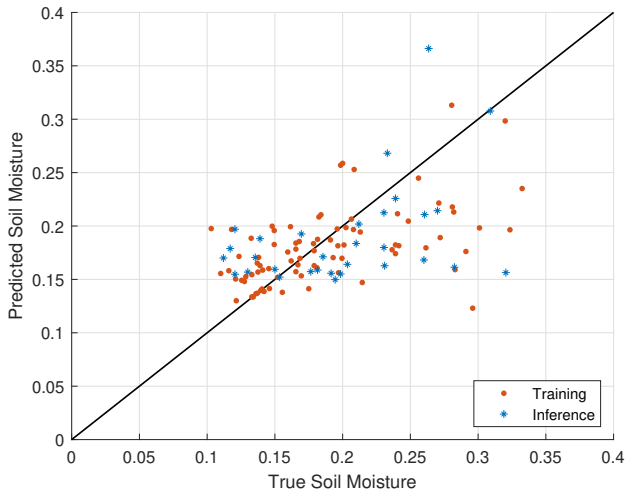
Within this section, parity plots pertaining to the Gaussian Kernel Regression model can be observed. These plots illustrate the predicted values against the true ones between training and inference phases across the five established groups.



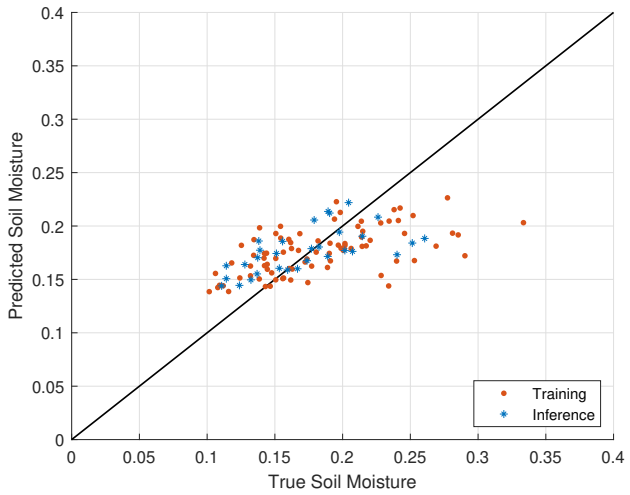
(a) 1^{st} Group



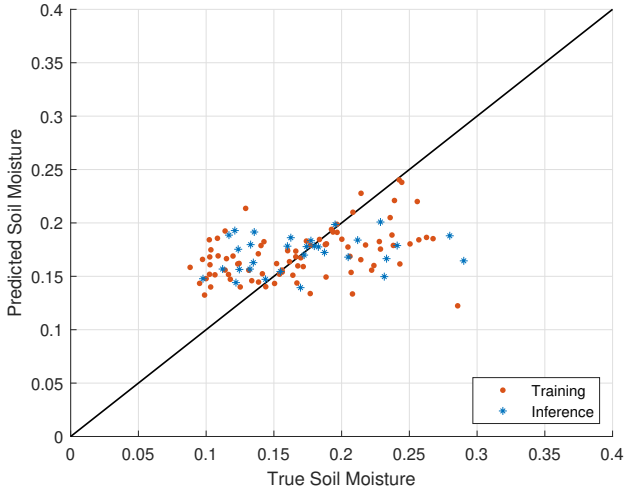
(b) 2^{nd} Group



(c) 3^{rd} Group



(d) 4^{th} Group



(e) 5^{th} Group

Figure 23: Gaussian Kernel Regression Model Parity Plots

By employing the 'fitrkernel' function, for each of the five groups, the hyperparameters were established and listed in the table 16.

Group	Learner	Lambda	NumExpansionDimensions	KernelScale	Epsilon
1	'leastsquares'	1.3521e-05	105	2.9955	NaN
2	'svm'	0.10768	174	0.096751	0.0010527
3	'leastsquares'	1.2422e-05	137	2.8916	NaN
4	'svm'	1.2065e-05	291	6.7998	0.017079
5	'svm'	0.15656	101	0.019773	0.0045992

Table 16: Optimized Hyperparameters for Gaussian Kernel Regression Model

The following table 17 presents the RMSE corresponding to the various stages of network training, namely the training and inference phase.

Group	RMSE Training Phase	RMSE Inference Phase
1	0.043	0.050
2	0.042	0.042
3	0.047	0.054
4	0.040	0.035
5	0.032	0.043

Table 17: RMSEs for Gaussian Kernel Regression Model

Following a similar path as the previous model (see A.8), here too, the architecture optimized for group 4 exhibits a very satisfactory RMSE value, albeit slightly higher than the absolute best result.

List of Figures

1 TxSON's Google Maps View . . . 4
2 TxSON's Google Maps View with Groups . . . 5
3 VH Amplitude vs Soil Moisture Comparison . . . 6
4 VV Amplitude vs Soil Moisture Comparison . . . 6
5 Sentinel-1 data official nomenclature . . . 7
6 Sentinel-2 data official nomenclature . . . 8
7 CS655 Probe . . . 9
8 SVM Method . . . 11
9 RF Decision Tree . . . 12
10 Ensemble of Decision Tree . . . 12
11 MLP Architecture . . . 13
12 S1-S2 Data Match-up . . . 16
13 Cross-validation Process Diagram . . . 17
14 ML Models RMSEs . . . 19
15 Groups Time Series . . . 25
16 Input vs Output Dependencies . . . 26
17 Linear Regression Model Parity Plots . . . 27
18 SVM Regression Model Parity Plots . . . 29
19 Random Forest Regression Model Parity Plots . . . 31
20 Ensemble of Learners Regression Model Parity Plots . . . 33
21 MLP Model Parity Plots . . . 35
22 Gaussian Process Regression Model Parity Plots . . . 37
23 Gaussian Kernel Regression Model Parity Plots . . . 39

List of Tables

1 Groups with relative stations . . . 5
2 Minimal RMSEs for each architecture . . . 20
3 Optimized Models vs Linear Regression . . . 20
4 Optimized Hyperparameters for Linear Regression Model . . . 28
5 RMSEs for Linear Regression Model . . . 28
6 Optimized Hyperparameters for Support Vector Machine Regression Model . . . 30
7 RMSEs for Support Vector Machine Regression Model . . . 30
8 Optimized Hyperparameters for Random Forest Regression Model . . . 32
9 RMSEs for Random Forest Regression Model . . . 32
10 Optimized Hyperparameters for Ensemble of Learners Regression Model . . . 34
11 RMSEs for Ensemble of Learners Regression Model . . . 34
12 Optimized Hyperparameters for Multi-Layer Perceptron Regression Model . . . 36
13 RMSEs for Multi-Layer Perceptron Regression Model . . . 36
14 Optimized Hyperparameters for Gaussian Process Regression Model . . . 38
15 RMSEs for Gaussian Process Regression Model . . . 38
16 Optimized Hyperparameters for Gaussian Kernel Regression Model . . . 40
17 RMSEs for Gaussian Kernel Regression Model . . . 40

Abstract in lingua italiana

L'obiettivo di questo studio è valutare e confrontare diversi tipi di algoritmi di machine learning per stimare con precisione l'umidità superficiale del suolo utilizzando dati satellitari. Il lavoro mira a identificare l'architettura più efficace per questo scopo, cercando di creare uno "strumento" globalmente applicabile che possa essere utilizzato in aree dove non sono presenti stazioni in-situ.

L'area di studio scelta è il network di TxSON in Texas, USA, caratterizzato da condizioni aride, caratteristiche uniformi e vegetazione scarsa. La ricerca copre un periodo di quattro anni, dal 1 gennaio 2018 al 31 dicembre 2021.

Per condurre questa analisi, sono state estratte immagini con retrodiffusione radar a doppia polarizzazione (polarizzazioni VH e VV) da Sentinel-1, mentre le bande del rosso e del vicino-infrarosso da Sentinel-2 sono state utilizzate per calcolare l'Indice di Vegetazione della Differenza Normalizzata (NDVI). In totale sono state raccolte 115 osservazioni satellitari.

Oltre ai dati satellitari, lo studio incorpora anche le serie temporali orarie di umidità del suolo presenti sul database ISMN. Questi dati vengono successivamente utilizzati per addestrare i modelli di machine learning.

Successivamente, i dati raccolti sono stati suddivisi in subset di addestramento e di inferenza. Il lavoro valuta diversi algoritmi di machine learning, tra cui Lineare, Random Forest (RF), Support Vector Machine (SVM), Gaussian Process Regression (GPR), MultiLayer Perceptron (MLP) e altri, con l'obiettivo di ottimizzare i loro iperparametri per ottenere il più basso Errore Quadratico Medio (RMSE) possibile, che funge da misura dell'accuratezza delle previsioni dei modelli.

Tuttavia, dopo aver condotto l'intero processo e analizzato i risultati, si è notato che i risultati non sono in linea con gli obiettivi prefissati. In realtà, la scoperta più significativa è che questo flusso di lavoro dimostra la sua efficacia quando viene applicato in un approccio "localizzato". L'addestrare un modello di machine learning per un sito specifico e utilizzarlo per prevedere con precisione i valori in un'area diversa, al fine di raggiungere l'obiettivo iniziale di creare uno strumento globale, sembra essere apparentemente impossibile.

Parole chiave: Umidità del suolo, Sentinel-1, Sentinel-2, Polarizzazioni VH & VV, Algoritmi di ML

Acknowledgements

I wish to convey my deep gratitude to all the individuals who have supported and encouraged me throughout the journey of completing this thesis. Without their assistance, support, and contributions, this work would not have been possible.

First and foremost, I would like to express my gratitude to my supervisor, Professor Claudio Maria Prati, for his expert guidance, unwavering support, and invaluable critiques that have shaped this thesis. His perspectives and knowledge have truly enriched my work.

A heartfelt thank you also goes to Dr. Alfonso Amendola and Dr. Simone Sala of ENI s.p.a., who generously shared their time and expertise, providing constructive feedback and valuable insights that have elevated the quality of this research.

I also wish to thank my parents, Domenica Rosa and Michele, my sister Miryea, my uncle Giuseppe and my family for their constant support, encouragement, and the trust they have placed in me. Your words of encouragement have been a source of motivation in every moment.

My warmest gratitude goes to my grandparents, Tonino, Mario, Pina, and Melina. Your life stories and unconditional love have motivated me to pursue academic success with determination. You are my source of strength.

A heartfelt thank you extends to my friends and colleagues who have shared ideas, resources, and precious leisure moments with me. Your presence has made this journey much more enjoyable.

I would like to thank Politecnico di Milano for providing me with the resources and learning environment necessary to develop this research. Your dedication to academic excellence has been a key factor in my journey.

Lastly, a special thank you also goes to myself. This journey has been a path of personal growth and self-discovery. Thank you for maintaining determination, patience, and enthusiasm as I worked on this thesis. This achievement is also the result of my constant commitment.

To all the mentioned individuals and others who have contributed in various ways, I sincerely thank you. This thesis is the collective outcome of your support and affection.

Thank you. Marco

Ringraziamenti

Desidero esprimere la mia profonda gratitudine a tutte le persone che mi hanno sostenuto e incoraggiato lungo il percorso di realizzazione di questa tesi. Senza il loro aiuto, supporto e contributi, questo lavoro non sarebbe stato possibile.

In primo luogo, vorrei ringraziare il mio supervisore, il Prof. Claudio Maria Prati, per la sua guida esperta, il suo supporto costante e le preziose critiche che hanno contribuito a plasmare questa tesi. Le sue prospettive e conoscenze hanno davvero arricchito il mio lavoro.

Un sentito ringraziamento va anche ai Dott. Alfonso Amendola e Dott. Simone Sala di ENI s.p.a., che hanno generosamente condiviso il loro tempo e le loro competenze, fornendo feedback costruttivi e spunti preziosi che hanno contribuito a elevare la qualità di questa ricerca.

Desidero anche ringraziare i miei genitori, Domenica Rosa e Michele, mia sorella Miryea, mio zio Giuseppe e la mia famiglia per il loro sostegno costante, il loro incoraggiamento e la fiducia che hanno riposto in me. Le vostre parole di supporto sono state una fonte di motivazione in tutti i momenti.

Ai miei nonni, Tonino, Mario, Pina e Melina, va il mio affettuoso ringraziamento. Le vostre storie di vita e il vostro amore incondizionato mi hanno motivato a perseguire il successo accademico con determinazione. Siete la mia fonte di forza.

Un sentito ringraziamento va ai miei amici e colleghi, che hanno condiviso con me idee, risorse e momenti di svago preziosi. La vostra presenza ha reso questo percorso molto più piacevole.

Vorrei ringraziare il Politecnico di Milano per avermi fornito le risorse e l'ambiente di apprendimento necessari per sviluppare questa ricerca. La vostra dedizione all'eccellenza accademica è stata un fattore chiave nel mio percorso.

Infine, un ringraziamento speciale va anche a me stesso. Questo percorso è stato un viaggio di crescita personale e di auto-scoperta. Grazie per aver mantenuto la determinazione, la pazienza e l'entusiasmo mentre lavoravo su questa tesi. Questo risultato è anche il frutto del mio impegno costante.

A tutte le persone menzionate e a tutte le altre che hanno contribuito in modi diversi, vi ringrazio di cuore. Questa tesi è il risultato collettivo del vostro supporto e affetto.

Grazie. Marco