




Article

A Machine Learning-Based High-Resolution Soil Moisture Mapping and Spatial–Temporal Analysis: The mlhrsm Package

Yuliang Peng ¹, Zhengwei Yang ² , Zhou Zhang ^{3,*}  and Jingyi Huang ^{4,*} ¹ Department of Statistics, University of Wisconsin-Madison, Madison, WI 53706, USA; peng68@wisc.edu² National Agricultural Statistics Service, U.S. Department of Agriculture, Washington, DC 20250, USA; zhengwei.yang@usda.gov³ Department of Biological Systems Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA⁴ Department of Soil Science, University of Wisconsin-Madison, Madison, WI 53706, USA

* Correspondence: z Zhang347@wisc.edu (Z.Z.); j Huang426@wisc.edu (J.H.)

Abstract: Soil moisture is a key environmental variable. There is a lack of software to facilitate non-specialists in estimating and analyzing soil moisture at the field scale. This study presents a new open-sourced R package **mlhrsm**, which can be used to generate Machine Learning-based high-resolution (30 to 500 m, daily to monthly) soil moisture maps and uncertainty estimates at selected sites across the contiguous USA at 0–5 cm and 0–1 m. The model is based on the quantile random forest algorithm, integrating in situ soil sensors, satellite-derived land surface parameters (vegetation, terrain, and soil), and satellite-based models of surface and rootzone soil moisture. It also provides functions for spatial and temporal analysis of the produced soil moisture maps. A case study is provided to demonstrate the functionality to generate 30 m daily to weekly soil moisture maps across a 70-ha crop field, followed by a spatial–temporal analysis.

Keywords: remote sensing; quantile random forest; visualization; leaflet; spatial–temporal analysis; water resources management



Citation: Peng, Y.; Yang, Z.; Zhang, Z.; Huang, J. A Machine Learning-Based High-Resolution Soil Moisture Mapping and Spatial–Temporal Analysis: The mlhrsm Package. *Agronomy* **2024**, *14*, 421. <https://doi.org/10.3390/agronomy14030421>

Academic Editors: Sara Álvarez and Sergio Vélez

Received: 21 January 2024

Revised: 12 February 2024

Accepted: 14 February 2024

Published: 22 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Soil moisture is a key variable for a variety of applications, including agricultural management [1], ecological modeling [2], weather forecasting [3], and environmental monitoring [4]. It varies from field to global scales and from seconds to decades as controlled by meteorological forcing (e.g., precipitation and air temperature), plant water uptake (e.g., evapotranspiration), soil texture, topography, and water resources management such as irrigation and drainage [5]. Existing soil moisture retrieval from remote sensing data relies on optical, visible near-infrared, thermal infrared, microwave, and radio wave sensors [6–8]. Due to the heterogeneity of soil moisture in space and time, combinations of observations from remote sensing platforms and process-based mechanistic models have been used to model and map soil moisture information across large areas and at short time intervals [9–12]. Current soil moisture products with a global coverage include surface soil moisture retrieved directly from the NASA Soil Moisture Active Passive (SMAP) mission [13] and ESA Soil Moisture and Ocean Salinity (SMOS) mission [14], GNSS-R-based NASA’s CYGNSS missions [15–17], as well as model-based surface and rootzone soil moisture estimated by assimilating SMAP product or CYGNSS products with water balance or land surface models [12,18–20]. Coupled with land information systems [21,22], interactive maps of land surface parameters can become transparently available to users for scientific research and natural resources management.

With the recent advances in machine learning (ML) algorithms, researchers have also developed data-driven models for mapping soil moisture, combining ground-based (in situ) sensors with remote sensing observations or models. Assisted with the cloud-based storage and computation infrastructure (e.g., Google Earth Engine—GEE), these data-

driven models have the potential to be applied to map soil moisture at regional to global scales. Recently, [23] used a quantile random forest model to map surface soil moisture (~0–5 cm) globally on a 12-day basis at 100-m. The model integrated global in situ soil moisture sensors from the International Soil Moisture Network (ISMN) with satellite observations (e.g., SMAP and Sentinel-1) and remotely sensed land surface parameters (terrain and soil properties). Similarly, [24] developed a Python package to generate global surface soil moisture maps at a 50 m resolution using ML models by combining in situ soil moisture observations from the ISMN with satellite data (Sentinel-1, Copernicus Global Land Cover Layer, Global Land Data Assimilation System soil temperature and snow water equivalent, Landsat 8 Shortwave Reflectance and Thermal Radiance, MODIS Enhanced Vegetation Index, OpenLandMap soil information). Despite the success, compared to process-based models [12,20], current data-driven ML models neither provide soil moisture dynamics at a high temporal resolution (currently 12-day interval limited by the revisiting time of Sentinel-1) nor did they provide estimates of soil moisture at subsurface/rootzone given most satellite platforms (e.g., optical and microwave sensors) only measure top few centimeters of soil. Obtaining field-level (<500 m spatial resolution) soil moisture information at a short time interval (e.g., subweekly) and at depths (e.g., within the rootzone) is particularly important and necessary for agricultural condition monitoring, hydrological modeling, and water resources management. Furthermore, existing open-source packages (e.g., [23–25]) only provide basic functions to generate soil moisture estimates across a geographic region or at single locations, and there are no companion functions/tools to process the soil moisture information to understand its spatial and temporal variability and analyze the model uncertainty.

To the best of our knowledge, there is a lack of open-sourced software to facilitate non-specialists such as researchers, educators, and students from domain sciences (e.g., agricultural science, environmental science, hydrology, and ecology) in estimating soil moisture variations at the field scale (<500 m) across large spatial extents and over a long period using various remote sensing derived geospatial datasets and at the same time facilitates the processing of soil moisture maps for scientific studies and water resources management. To address the problem, this article presents a new R package, **mlhrsm**, which allows the user to generate machine learning-based high-resolution (30 to 500 m) soil moisture (volumetric water content—VWC) maps and uncertainty estimates across the contiguous United States (and elsewhere wherever in situ soil moisture sensor measurements are available for refining the models) for both soil surface (0–5 cm) and rootzone (0–1 m) on a daily basis using newly developed ML models. In addition to producing application-ready high spatial and temporal resolution surface and rootzone soil moisture data, the **mlhrsm** package will also provide functions for spatial and temporal analysis of the retrieved soil moisture maps for scientific research and water resources management across scales. A case study will be provided to demonstrate the functionality of the **mlhrsm** package to obtain 30 m soil moisture maps across a 70-ha cropland field and at selected sites in Wisconsin (WI), USA, followed by spatial–temporal analysis to understand the variations in soil moisture for field-level water resources management.

2. Materials and Methods

The ML models were established based on the quantile random forest algorithm [26] combining nationwide in situ soil moisture measurements averaged at two depths (0–5 cm and 0–1 m) with spatiotemporal resampled satellite imagery (e.g., Sentinel-1 Synthetic Aperture Radar backscatter, Landsat-8 visible, near-infrared, and thermal bands and indices), satellite-based models of the soil surface and subsurface moisture (NASA-USDA Enhanced SMAP soil moisture), land surface temperature (MODIS), and ancillary land surface parameters (NLCD land cover, USGS digital elevation model and terrain attributes, Polaris soil properties at 0–5 cm and 0–1 m) extracted to the in situ sites at a 30 m resolution. The selected covariates for modeling surface and rootzone soil moisture are described in Supplementary Material S1.

The quantile random forest model was selected because (1) it had a relatively better overall performance in handling covariates that had non-linear relationships with model responses (e.g., soil moisture) and were inter-correlated with each other [23,27]; (2) it directly provided model uncertainty estimates (e.g., standard deviation and percentiles), which were useful for hydrological modeling and risk assessment in decision-making. Although many other advanced deep learning algorithms were available for soil moisture mapping, we did not choose them as the default model because they either did not generate models that were statistically more accurate than our models or required much more computation power and expert knowledge in parameter tuning and extrapolation to large regions (e.g., [28–31]). As such, the quantile random forest algorithm was selected as the default method to balance the model performance and software usability (refer to Section 3.1). Note that users can always modify the default model based on their study site and preference (refer to Supplementary Material S3.2).

3. Results

3.1. Model Performance

Data were collected from 220 stations across the contiguous USA (CONUS), spanning 1 March 2016 to 30 September 2019. For each year, daily data from 1 March to 30 September are included, leading to a total of 172,887 data points. The detailed list of stations used as the training dataset can be found in the Supplementary Material. As shown in Table 1, the models trained from the randomly selected stations (70%, Figure 1) were applied to the randomly hold-out testing stations and yielded overall r^2 (Pearson's correlation coefficient squared) of 0.649 and 0.535, bias of 0.010, $0.006 \text{ m}^3 \text{ m}^{-3}$, and RMSE of 0.075 and $0.095 \text{ m}^3 \text{ m}^{-3}$, for modeling the surface (0–5 cm) and rootzone (0–1 m) soil VWC within the CONUS, respectively. Note that the random split of in situ soil moisture stations was only for evaluating the ML models, and the final models used both training and testing stations for fitting the models to make predictions of soil VWC across the CONUS. As shown in the Supplementary Material, we used five-fold cross-validation to train the models with 40 trees and a minimum node size of 10. These parameters were empirically determined where model performance did not increase statistically with the increasing complexity of the tree structure.

Table 1. Summary statistics of model performance at the testing stations as shown in Figure 1.

Surface Soil Moisture	Performance
Bias ($\text{m}^3 \text{ m}^{-3}$)	0.010 [−0.092, −0.015, 0.125]
RMSE ($\text{m}^3 \text{ m}^{-3}$)	0.075 [0.024, 0.072, 0.133]
Correlation coefficient (squared, r^2)	0.649 [0.006, 0.406, 0.835]
Kling–Gupta Efficiency (KGE)	0.624 [−0.899, 0.382, 0.796]
Nash–Sutcliffe efficiency (NSE)	0.376 [−5.195, 0.092, 0.604]
Rootzone Soil Moisture	
Bias ($\text{m}^3 \text{ m}^{-3}$)	0.006 [−0.216, −0.015, 0.203]
RMSE ($\text{m}^3 \text{ m}^{-3}$)	0.095 [0.020, 0.065, 0.218]
Correlation coefficient (squared, r^2)	0.535 [0.000, 0.318, 0.896]
Kling–Gupta Efficiency (KGE)	0.492 [−2.470, 0.180, 0.754]
Nash–Sutcliffe efficiency (NSE)	0.042 [−177.149, −0.237, 0.521]

The values outside square brackets are metrics calculated with all the stations from either training or testing datasets, while the values inside the square brackets are the minimum, median, and maximum values of the metrics calculated among individual stations, respectively. For example, in terms of predicting surface soil moisture on the testing dataset, when considering all the stations, the model has an overall RMSE of $0.075 \text{ m}^3 \text{ m}^{-3}$; when considering individual stations, the model can achieve an RMSE as low as $0.024 \text{ m}^3 \text{ m}^{-3}$ at one station, as high as $0.133 \text{ m}^3 \text{ m}^{-3}$ at another station, and the median RMSE among all the stations is $0.072 \text{ m}^3 \text{ m}^{-3}$.

To assess the variability of model performance across sites, the model performance is also calculated at individual in situ soil moisture stations and presented with the minimum, median, and maximum performance metrics among the training and validation

sites (values in square brackets in Table 1). Despite the varying performance of the model across sites for both surface soil moisture and rootzone soil moisture, the ML models perform reasonably well for most sites, as indicated by the median values of the performance metrics. Compared to coarse-resolution soil moisture products (e.g., several kilometers for SMAP L3 and L4) that report an overall RMSE of less than $0.04 \text{ m}^3 \text{ m}^{-3}$ [20], the model performance of the ML-based high-resolution soil moisture models is moderate, which is mainly caused by three issues: (1) the predictability of the environmental covariates (e.g., remote sensing bands or derived rootzone soil moisture models and land surface properties), which do not have a perfect ($r = 1$) correlation with soil moisture due to the limitations of their physical mechanisms and varying spatial–temporal resolutions [13,20]; (2) the strong spatial–temporal variability of soil moisture at the field level [23–27], which reduces the performance of the field-level soil moisture models compared to coarse-resolution soil moisture products; and (3) the transferability of the ML models to regions with sparse in situ soil moisture observations, which will be overcome in the future version of the **mlhrsm** package by adopting transfer learning algorithms that are less sensitive to shift in data distributions between training and testing domains [28].

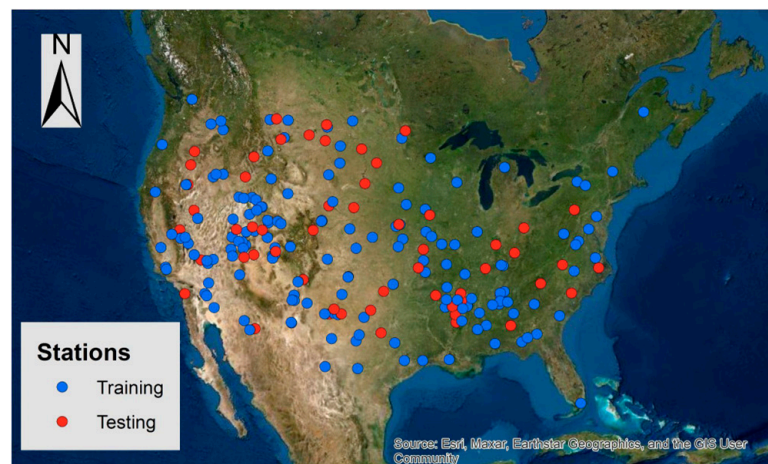


Figure 1. Locations of training (70%) and testing (30%) stations randomly selected across the contiguous USA (CONUS) for evaluating the ML model performance.

To better assist users in using the proposed R package in their research and field applications, the model performance calculated across different land cover types (at the validation sites) is presented in Table 2. Overall, when predicting surface soil moisture, the ML models perform well for pasture and developed (urban) soils, followed by grassland, barren, and cropland. This is not unexpected because these sites often have relatively light vegetation canopy cover, and the responses of satellite data (e.g., Sentinel-1 backscatter and MODIS/Landsat 8 shortwave infrared and/or thermal bands) to changes in soil moisture are relatively strong [13,32,33]. Also note that our ML models have reasonably good performance for shrubs, forests, and wetlands. Traditional studies agree that satellite soil moisture products are only able to capture surface soil moisture variations under light vegetation canopy. Theoretically, when coupled with subsurface soil properties, rootzone hydrological models could capture the rootzone soil moisture variability. This is also confirmed by other researchers when evaluating the potential use of SMAP soil moisture for monitoring rootzone soil moisture dynamics, given that there is a strong correlation between surface and rootzone soil moisture [34].

However, the ML models established here could barely capture the rootzone soil moisture variability under cropland, pasture, shrub, forest, and developed land (r^2 : 0.25–0.64). This means that ML models alone are not enough to capture the rootzone soil moisture variability. To improve the performance of the ML models, mechanistic models can be incorporated in the future using a data assimilation framework [9,12,13].

Table 2. Summary statistics of model performance at the testing (30%) stations for each land cover type. Note: N.A. means the number of sites is not statistically sufficient at that depth.

Surface Soil Moisture	Cropland	Pasture	Grassland	Shrub	Forest	Barren	Wetland	Developed
Bias ($\text{m}^3 \text{m}^{-3}$)	−0.027	−0.019	0.001	0.004	−0.032	0.036	−0.033	0.002
RMSE ($\text{m}^3 \text{m}^{-3}$)	0.081	0.063	0.083	0.054	0.089	0.048	0.095	0.060
r^2	0.543	0.682	0.348	0.436	0.419	0.835	0.531	0.748
KGE	0.609	0.703	0.553	0.466	0.481	0.493	0.418	0.758
NSE	0.484	0.648	0.293	0.428	0.330	0.594	0.407	0.744
Rootzone Soil Moisture								
Bias ($\text{m}^3 \text{m}^{-3}$)	0.019	−0.063	0.007	0.012	−0.070	N.A.	−0.074	−0.010
RMSE ($\text{m}^3 \text{m}^{-3}$)	0.113	0.090	0.096	0.064	0.089	N.A.	0.135	0.072
r^2	0.296	0.427	0.203	0.304	0.252	N.A.	0.086	0.640
KGE	0.232	0.574	0.302	0.482	0.433	N.A.	0.004	0.616
NSE	0.249	−0.153	0.185	0.233	−2.281	N.A.	−0.312	0.617

We also assessed the temporal stability of the ML models using observations from all the stations from 2016 to 2018 as training and data in 2019 as validation. We did not notice significant drifts in model performance over time.

3.2. Overview of the Functionality of *mlhrsm*

The R package has been developed and tested on the Windows operation system with R software (version 4.2.1). To use the *mlhrsm* package (version 1.0), the user should first create a Google Earth Engine account (<https://earthengine.google.com/signup/>) and then install the gcloud CLI software (<https://cloud.google.com/sdk/docs/install#windows>). A number of dependency R packages are needed for the *mlhrsm* package, including *raster*, *rgee*, *sf*, *tidyverse*, *viridis*, *FedData*, *RColorBrewer*, *caret*, *chillR*, *leaflet*, *hydroGOF*, *quantregForest*, *randomForest*, *reshape2*, *rgdal*, *sp*, *lubridate*, *geojsonio*, and *stars*. After loading the *mlhrsm* package, the packages *sf*, *raster*, *tidyverse*, and *rgee* are automatically loaded as well. For detailed instructions on package installation, the user can refer to Supplementary Material S2.

The main functionality of *mlhrsm* is described in Figure 2 and can be grouped into four categories. (1) Preprocessing: splitting a large region of interest (ROI) into subregions (**split_region**), downloading covariates and making soil moisture maps in small subregions (**download_map**); (2) mapping: downloading covariates and making soil moisture maps in a small region (**VWC_map**) or at selected sites (**VWC_point**), spatially extracting soil moisture from generated maps to points of interest (**point_extraction**); (3) analysis: calculating daily VWC and statistical summaries (e.g., maximum) for the entire region of interest and their time series plots over time (**area_sum**), generating maps of temporal statistical summaries pixel-wise across the ROI (**pixelwise_sum**), generating pixel-wise temporally aggregated maps of VWC with a specified temporal resolution (**aggregate_interval**) and for a specified period (**aggregate_sum**), generating a scatter plot for predicted VWC vs. measured VWC provided by the user (**point_performance**); and (4) plotting: plotting the maps of VWC on a specified date (**plot_map**) or temporally aggregated VWC (**plot_aggregated_VWC**), plotting the locations and VWC estimates at the observation sites provided by the user (**site_variation**) and time series plot of VWC on these observation sites (**plot_CI**). A detailed explanation of these functions is provided below.

3.2.1. Main Functions

The main functions provided by the package are used to download satellite data and apply the ML models to the selected areas or sites. To use these functions, the user is expected to provide an input file of the region of interest (ROI, in the shapefile format) or

GPS coordinate locations (in the CSV format) for soil moisture retrieval. It is recommended that the input files be stored in the root directory of the **mlhrsm** package.

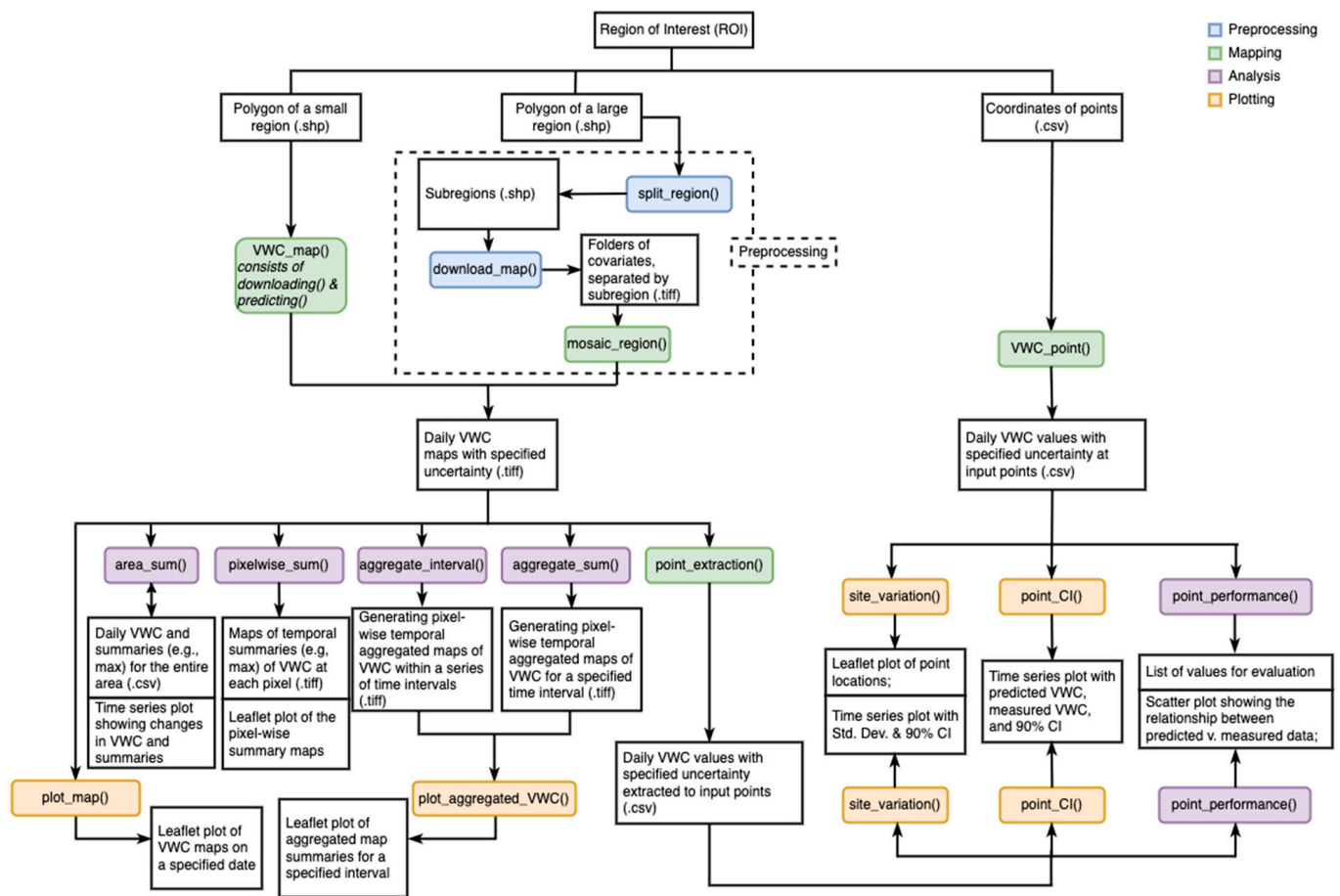


Figure 2. Flowchart of the functions of the **mlhrsm** package.

Note that the established (default) ML models and input in situ soil moisture (Figure 1) used for training and evaluating the models can also be accessed from the R package. Previous studies have suggested that the inclusion of local training datasets in a regional or global soil moisture model can significantly improve the model performance in regions with sparse training datasets (e.g., [23]). The user can choose to modify the default ML models by incorporating a local training dataset (in situ soil moisture measurements) collected from their study fields. Details about building the models for the **mlhrsm** package are provided in Supplementary Material S3.

VWC_Map

The **mlhrsm** package provides three ways to apply the established ML models to generate VWC maps. The basic function **VWC_map** allows the user to download input covariates and map VWC across a small ROI in one step. To use the **VWC_map** function, the user is expected to provide the name of the shapefile of the ROI (preferably in the WGS84 system, “EPSG: 4326”), start and end dates specifying the period of daily soil moisture maps to be retrieved (in the format of “yyyy-mm-dd”), and the spatial resolution of the output VWC maps (in the unit of meter). If indicated by the user (percentile = TRUE), the model will also calculate the upper and lower bounds of the 90% Confidence Interval (CI) of the VWCs predicted from the quantile random forest models. The outputs of this function are raster files of covariates and predicted VWCs (daily mean and standard deviation (sd) of VWC at 0–5 cm and 0–100 cm depths, with or without CI of VWC) in GeoTIFF format. The output maps will be saved in the NAD83 system

and USA Contiguous Albers Equal Area Conic, USGS (“EPSG: 5070”) for water resources management in the USA. A project name is needed in the function so that all the downloaded covariates and VWC maps will be saved in a folder under the working directory named after the project name. This project name will also be used later for spatial-temporal analysis and visualization with other functions.

With the limitation of the availability of several input satellite covariates (e.g., Sentinel-1 backscatter data), the earliest date of the available VWC map is 1 January 2016 (no sufficient Sentinel-1 data prior to that date for soil moisture estimation), and the end date lasts to the present (ongoing until the end of the mission of Sentinel-1, SMAP, MODIS, or Landsat 8). The finest spatial resolution is 30 m, and the user can specify other resolutions up to 500 m. Any mapping attempt beyond the ranges of the input arguments will result in errors.

The **VWC_map** function involves three main steps, as briefly described below (refer to Supplementary Material S1 for details). First, the function will download various input covariates data from the GEE within the ROI, including the following:

- (a) Constant land surface parameters: 30 m National Land Cover Dataset (NLCD) land cover maps in 2016, 10 m elevation data from the USGS 10 m digital elevation model and derived slope, aspect, and hillshade, 30 m Polaris soil clay and sand content and bulk density maps at 0–5 cm and 0–1 m, and
- (b) Dynamic variables spanning the input period of VWC maps (with a buffer period of 6–64 days for temporal interpolation depending on the available satellite data): 30 m 12-day Sentinel-1 backscatter data measured at VV and VH polarizations and incidence angle (masked for outliers and despeckling following [23]), 1 km daily SMAP land surface temperature, Landsat-8 bands 5, 6, 7, and 10, and NDVI and NDWI indices, and NASA-USDA Enhanced SMAP 10 km surface and subsurface soil moisture storage maps.

Second, the constant covariates will be resampled across the ROI to the spatial resolution defined by the user using bilinear interpolation, and the dynamic covariates will be resampled spatially to the input resolution and temporally to a daily basis using bilinear interpolation.

Lastly, the pre-established ML models will be applied to the processed satellite covariates within the ROI to generate VWC maps at the surface and rootzone at the specified spatial resolution on a daily basis. Note that the current version of the **mlhsrcm** package also allows the users to refit the ML models by either including more in situ soil moisture observations or using other ML algorithms and subsequently using the updated ML models to map soil moisture at the target sites. If the users want to adapt the ML models to map soil moisture outside the CONUS and globally, they can also modify the code accordingly. Detailed instructions on these modifications are provided in Supplementary Material S3.

VWC_Point

In addition to generating soil moisture maps across a certain area of interest using **VWC_map**, the **VWC_point** function is created for predicting VWC at individual points for the user interested in exploring the VWC dynamics at specific locations, especially with points located far from each other (e.g., in different US states). The **VWC_point** is similar to **VWC_map**, except that the former generates a CSV file containing the input points’ IDs and coordinates along with the VWC values extracted at 30 m (with the points located at the center of the 30 m pixels) while the latter produces VWC maps across the input ROI in the GeoTIFF format.

Split_Region, Download_Map, and Mosaic_Region for Large ROIs

For large-size satellite imagery, GEE often tends to split the file into multiple GeoTIFF files, which can lead to memory errors when these files are post-processed locally in the R environment (e.g., mosaicing, masking, resampling). In this case, **VWC_map** will return a message directing the user to split the ROI and mapping task into subregions

using **split_region**, **download_map**, and **mosaic_region**. Although the user can still map soil moisture across large regions with **VWC_map**, it is suggested that these alternate functions be used to save computation time when the input ROI is large ($\geq 1000 \text{ km}^2$) or a high input image resolution ($\leq 50 \text{ m}$) is specified within an ROI covering a catchment or larger ($\geq 500 \text{ km}^2$).

The **split_region** requires the input of an ROI polygon (and the project name if the user wants to save it in a new folder rather than the working directory). It will split the large polygon stored in the root directory into several small subregions with a cell size of 0.25 arc-degrees (approximately 500 km^2) and save the split subregions as a new shapefile with a default name of *sub_regions.shp*. If there are multiple subregions with an area smaller than 250 m^2 , the **split_region** function will first merge them into one subregion to decrease the total number of subregions to reduce the data processing and downloading time in the GEE. The total number of subregions will be printed after splitting is finished. An example of the splitting process is provided below with the subregions shown in Figure 3.

```
> library("mlhrsm")
> split_region("Grant.shp", "Grant2021")
> sub_grant <- read_sf("Grant2021/sub_regions.shp")$geometry
> plot(sub_grant)
```

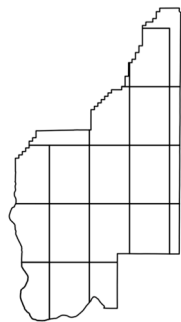


Figure 3. The plot of the subregions after splitting an approximately 7000 km^2 ROI of Grant County, WA, USA.

After splitting the large ROI into smaller subregions, the user can choose to download the input covariates using the **download_map** function and produce VWC maps in all subregions sequentially on one computer. Alternatively, the user could use multiple computers to download covariates and map VWC in different subregions in parallel to reduce the processing time. Note that the latter requires multiple GEE accounts to be activated on different computers to prevent all submitted tasks from being assigned to the same GEE account and held in a long waiting line. The **download_map** function requires similar user inputs as the **VWC_point**, except for an additional **sub_area** argument that allows the user to specify the subregion IDs in which the function will download the satellite covariates. This argument can either be assigned to a single number (subregion ID), indicating a specific subregion, or a vector of IDs for multiple subregions.

After the covariate maps are downloaded for each subregion, the **mosaic_region** function needs to be run to generate soil VWC maps in each subregion and merge the VWC maps into one master map once the mapping process is completed. If one or several subregions do not have any Sentinel-1 (due to its flight path) or Landsat (due to cloud contamination) imagery on specified dates, the mean values of the corresponding Sentinel-1 or Landsat imagery across the entire ROI (covering all subregions) on those dates are used to fill these NA values on a daily basis. This simple gap-filling is used to avoid sharp artifacts at the boundaries of the neighboring subregions and save computation time. It is chosen so that the temporal variability of Sentinel-1 or Landsat imagery is preserved in favor of soil moisture retrieval since the changes in soil moisture are mainly caused by precipitation and evapotranspiration, both of which display relatively larger temporal

variations on a daily to monthly scale than spatial variations at a catchment level (and users care more about their temporal change) [35–41]. In the future, it is worth investigating other gap-filling algorithms [42–44] and including them in the **mlhrsm** package. Afterward, the ML models for surface (0–5 cm) and rootzone (0–1 m) VWC will be applied to the subregions to generate maps of VWC and CIs (if selected). Lastly, VWC maps from all subregions will be automatically mosaiced into master maps for the original large ROI on a daily basis, similar to those VWC maps produced from the **VWC_map** function.

3.2.2. Functions for Spatial and Temporal Analysis

Apart from the main mapping functions, **mlhrsm** also provides spatial–temporal analysis and plotting functions to assist with the interpretation and visualization of the output maps. The plotting functions extract and plot the VWC maps and their spatial statistics within different time intervals for a specific ROI or several locations. The spatial–temporal analysis functions calculate the VWC summary statistics, including mean, sd, min, and max in space or time, and round the summary statistics to the third decimal place.

Area-Based/Zonal Functions

plot_VWC plots the statistical values (mean, sd, and CI if percentile = TRUE) of VWC maps for one specified date (“yyyy-mm-dd”) and depth (0–5 cm or 0–1 m) with the *leaflet* package. It returns an interactive leaflet map showing the soil VWC values of the study area at the defined spatial resolution on a selected date. The user can choose to display the mean, sd, or CI values of the VWC maps estimated from the quantile random forest models. The user can also plot VWC maps on multiple days and compare the change in soil moisture (see case study below).

area_sum summarizes the area-based/zonal statistical values (mean, min, max, and sd) of the daily VWC at a specified depth (0–5 cm or 0–1 m) across the entire ROI. The output of this function includes the calculated statistical values on a daily basis as a CSV file (*VWC_depth_ts_data.csv*) in the specific project folder for further analysis and a time series plot showing the changes in these summary statistics of VWC over the entire study period on a daily basis during the study period. If the ROI represents a specific field, farm, or catchment, the time series plot will indicate the field-, farm-, or catchment-averaged VWC statistics during the study period.

Unlike **area_sum**, **pixelwise_sum** returns the summary statistics of VWC at a specified depth calculated pixel-wise across the ROI area over the study period displayed as area maps of these statistics in *leaflet*. The maps are also saved as GeoTIFF raster files in a folder called *temporal_VWC* under the project folder. These maps delineate subregions within the ROI area with wetter/drier soils over a longer period than neighboring places.

To facilitate the interpretation of VWC dynamics at different temporal scales (e.g., daily vs. weekly vs. seasonally), **aggregate_sum** and **aggregate_interval** provide methods to aggregate the daily VWC maps according to specified time intervals. **aggregate_sum** has similar functionality as **pixelwise_sum**, except that the user can specify a different start and end date (needs to be within the initial study period provided in the **VWC_map** function). **aggregate_sum** then summarizes the VWC maps at a specified depth pixel-wise over the newly defined period and outputs the summary statistics (pixel-wise mean, median, min, max, sd across the ROI) as GeoTIFF maps in the *VWC_aggregation* folder. It also saves the summary statistics as separate CSV files along with each grid pixel’s coordinates. Different from **aggregate_sum**, **aggregate_interval** allows the user to define the time intervals/temporal scale for the aggregation. Besides a start and end date, the user will provide a scale/frequency argument (number of days) representing the aggregation interval (e.g., 7 means converting daily VWC maps to weekly averaged VWC maps). The **aggregate_interval** function will return a list of dates indicating the starting dates of each interval and save the pixel-wise temporally aggregated maps named after the list of the dates in the *VWC_aggregation* folder, along with the CSV files of the grid coordinates and aggregated VWC values on the new temporal scale/intervals. If the total number of days

(between the newly defined start and end dates) is not a multiple of the frequency, several days (less than the length of frequency) will be left behind, and the summaries of VWC maps over these days will still be calculated and saved as output data (unless there is only one day left).

Once the temporal aggregation is performed, **plot_aggregated_VWC** serves as a plotting function specifically for the aggregated maps. It needs inputs of a start and end date and returns a similar output plot in *leaflet* as the **pixelwise_sum**. If the user wants to visualize the map saved by **aggregate_interval** (e.g., a weekly VWC map), they need to provide a specific date (starting date of one of the day intervals) and the frequency for the function to locate and plot the processed files.

Point-Based Functions

For users who want to investigate VWC dynamics at specific locations within the produced VWC maps, **point_extraction** can be used. The user needs to specify the site locations by assigning the name of the shapefile (saved in the same project folder) as the function argument. **point_extraction** will then extract the predicted VWC values (mean, sd, and lower and upper bounds of 90% CI if such files exist in the project's VWC folder) to the points and save the resulting site coordinates and daily VWCs as a CSV file with a default name of *VWC_point_data.csv*. The user should note that if no specific filename is identified, the saved CSV file will overwrite the older version in the project folder, which may result in loss of data. If a specific filename is given by the user, the VWC values will be saved according to the specified filename. The user is then able to run the plotting and evaluation functions for point VWCs at a certain depth from the CSV file produced.

site_variation visualizes the change in VWC values at specified point locations over the study period defined when running the **VWC_map** and **point_extraction** or **VWC_point** function. The output of this function includes a *leaflet* plot showing the locations of the selected points (specified in the **site_variation** function) and time series plots of the predicted VWC at different point locations over time, with mean \pm sd highlighted in shaded regions and dashed lines for the upper and lower bounds of the 90% CI (when `percentile = TRUE`). If no site location is specified, the function will automatically plot VWC values at all the points. When the total site number exceeds 12, a random selection of 12 points will be included in the output plot. The function inputs data from the *VWC_point_data.csv* file by default but can be applied to another extracted point dataset by specifying it in the function.

point_CI also returns a time series plot. It provides an evaluation of the model performance at the selected point by plotting the 90% CI of the predicted VWC values along with the measured VWC values from an in situ soil moisture sensor at a specified depth (the measured data need to be provided in the input CSV file). The plot follows a similar algorithm as **site_variation**. The user can choose the sites to visualize, and if no site is specified, a total of 12 randomly chosen points will be presented. In the visualization plot, the shaded region represents the 90% CI of daily VWC values, with solid lines representing predicted mean VWC values from the machine learning models and dashed lines as measured values from in situ sensors.

To fully evaluate the model performance at selected sites with in situ VWC measurements, **point_performance** can be used to calculate several performance metrics between the sensor-measured VWC and model-derived VWC values, including coefficient of determination (R^2), root-mean-square-error (RMSE), Nash–Sutcliffe efficiency (NSE) [45], Kling–Gupta efficiency (KGE) [46], Bias, the ratio of performance to deviation (RPD), and the ratio of performance to inter-quartile (RPIQ) [47]. The evaluation metrics will be returned in a list together with a scatter plot showing predicted VWC against measured VWC at the selected sites. A diagonal (1:1) line is also included in the plot. If indicated (`stats = TRUE`), the validation values of R^2 , RMSE, Bias, and KGE will also be presented on the scatter plot.

To use the **point_CI** and **point_performance** functions, the user should provide a CSV file of the measured VWC (from in situ soil sensors), which includes site *ID*, coordinates (*Longitude*, *Latitude*), *Date* (in “yyyy-mm-dd” format), and depth interval (*VWC_5* or *VWC_100* for averaged VWC at 0–5 cm or 0–1 m, respectively). The evaluation CSV file can be stored outside the working directory and passed to the function with the full directory when needed.

3.3. Case Study

3.3.1. Study Area

A 70 ha cropland field in Rock County, WI, USA, is used here as an example to demonstrate the functionalities of the **mlhrsm** package. In total, 12 in situ soil moisture sensors (TEROS 12, Meter Group, Inc., Pullman, WA, USA) were installed across the field, and soil moisture data were collected at 5 cm depth from June to September 2020. This study area was used for demonstration because closely spaced in situ soil moisture data were available across this field for evaluating the high-resolution models and because most of the existing in situ soil moisture sensor networks do not have sensors that are spaced closely enough to each other (spacing often larger than 1 km) for model evaluation.

3.3.2. Generating Soil Moisture Maps at Different Spatial Resolutions

Since the ROI has a small area of 0.7 km², after the **mlhrsm** package is installed and loaded into the R environment and a GEE account is activated (refer to Supplementary Material S2), the user can use the **VWC_map** to generate soil VWC maps directly for the study field and plot the results using **plot_map**.

```
> VWC_map("WI.shp", "2020-06-15", "2020-09-15", 30, TRUE, "WI_region")
> plot_map("2020-08-01", 5, TRUE, project = "WI_region")
```

The estimated soil VWC values at the soil surface (0–5 cm) across the study field on 1 August 2020 are displayed as a *leaflet* map object using the **plot_map** function (Figure 4). The default layer is set at the mean. The user can select different layers of standard deviation (*SD*) and upper (*0.95*) and lower (*0.05*) bounds of the 90% confidence interval for VWC values on that date produced by the quantile random forest model if previously computed by **VWC_map**. The map has a spatial resolution of 30 m, pre-determined in the **VWC_map** function, and can no longer be changed during plotting (unless the user resamples the GeoTIFFs using other R functions). As shown in Figure 4, pixels highlighted in red indicate dry regions of the field (VWC of 0.15–0.16 m³ m⁻³), while pixels in blue indicate wet regions of the field (VWC of 0.21–0.23 m³ m⁻³). In summary, the field was dry on the selected day during the crop-growing season.

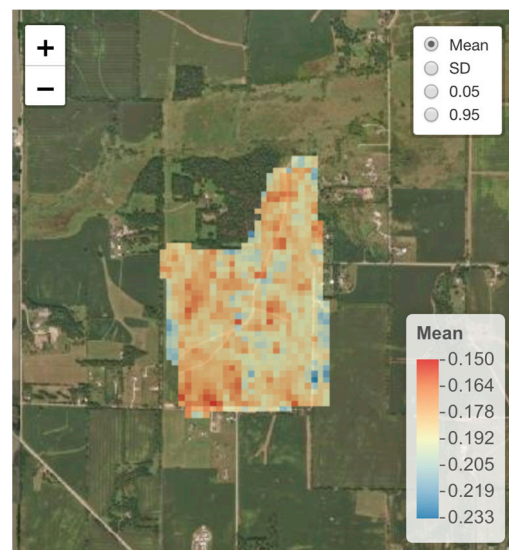


Figure 4. A *leaflet* plot of the mapped VWC of a specific date (1 August 2020) and depth (0–5 cm) is returned upon calling `plot_map`. The mean, SD, 0.05, and 0.95 (confidence intervals) are the mean and uncertainty on that date predicted from the quantile random forest model.

3.3.3. Spatial and Temporal Analysis

Compared to other existing packages that generate soil moisture maps (e.g., [24]), one major contribution of the `mlhrsm` package is its spatial–temporal analysis tools. For example, the user can use the function `area_sum` to calculate and save the statistical summaries (mean, sd, min, and max) at a certain depth of the daily average VWC across the entire ROI, as shown below:

```
> area_sum(5, project = "WI_region")
> head(read.csv("WI_region/VWC_5_ts_data.csv"))
```

The function `area_sum` saves zonal statistics of soil VWC values across the entire ROI and returns a time series plot of its temporal variations. As shown in Table 3, the *Date* variable indicates the dates for which the summaries are calculated, and *Summary* defines the type of summary for each *value*. The changes in the VWC summary characteristics are presented in a different color for different statistical values and visualized on a daily basis, as in Figure 5. It is noted that the predicted VWC for the study area (ROI) increases to a peak around 22 June 2020 (due to a major rainfall event), and then decreases gradually until the end of August (due to water uptake by soil and plant evapotranspiration), where the VWC values increased again to another peak in mid-September (due to another major rainfall event).

Table 3. Subset of `VWC_5_ts_data.csv`, produced by `area_sum`.

	Date	Summary	Value
1	2020-06-15	Mean	0.264
2	2020-06-15	Median	0.265
3	2020-06-15	Min	0.190
4	2020-06-15	Max	0.311
5	2020-06-16	Mean	0.228
6	2020-06-16	Median	0.229

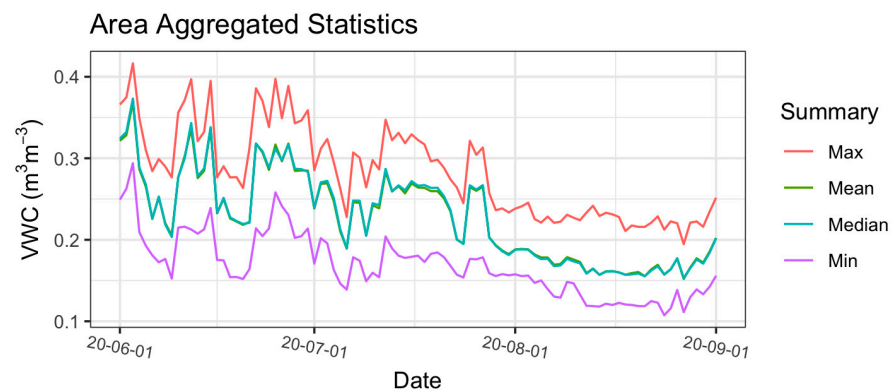


Figure 5. Time series plot output of `area_sum`.

The user can also calculate and save the maps of the summary statistics of each pixel at a certain depth by calling `pixelwise_sum`, as shown below:

```
> pixelwise_sum(100, project = "WI_region")
```

After running the `pixelwise_sum`, six summary maps of pixel-wise *Mean*, minimum (*Min*), *Median*, maximum (*Max*), standard deviation (*SD*), and *Range* of the VWC values at the rootzone (0–1 m, “100” means 100 cm) across the entire ROI area from 15 June 2020 to 15 September 2020, produced by the `VWC_map`, will be automatically saved and visualized. As shown in Figure 6, the plots consist of the six map layers with a similar color scale as the `plot_VWC` function output. The pixel-wise mean VWC values across the ROI from 15 June 2020 to 15 September 2020 ranged between 0.23 and 0.36 $\text{m}^3 \text{m}^{-3}$. The users should not confuse these statistical summary plots with those shown in Figure 4, as the maps in Figure 6 are calculated pixel-wise using multiple dates of VWC maps over a specified period, while the maps in Figure 4 are generated from the default ML models on a specified date based on the quantile random forest algorithm. The uneven distribution of soil VWC is mostly caused by the spatial variations in soil texture (e.g., clay and sand content), given the field has a relatively uniform elevation (refer to [48] for details about this field). Note that if the user sets a large ROI and/or with higher spatial resolution, the calculation time for the `pixelwise_sum` will increase accordingly.

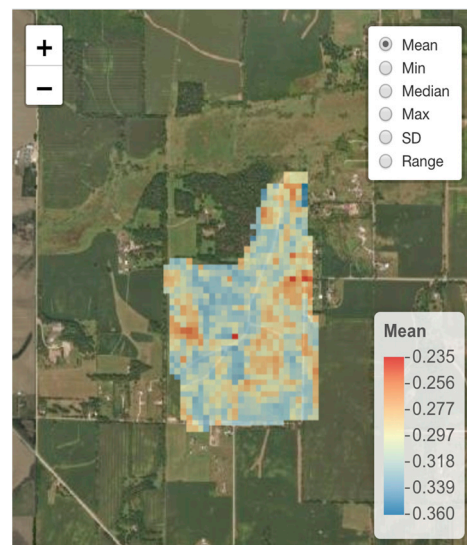


Figure 6. *leaflet* plot output of `pixelwise_sum`. The statistical summaries are calculated pixel-wise from previously generated VWC maps over a specified period (15 June 2020 to 15 September 2020).

The **mlhrsm** package also allows the calculation of the statistical summaries for the pre-mapped VWCs across multiple intervals of time. For this, the user can apply **aggregate_interval** on a subset of the entire range of time defined by **VWC_map** as below:

```
> aggregate_interval(5, "2020-06-15", "2020-08-15", frequency = 7, project = "WI_region")
> head(read.csv("WI_region/VWC_aggregation/2020-06-15_2020-08-15/7days_5cm/7_day_mean.csv"))
```

The pixel-wise summary statistics for the 7-day intervals are saved in designated directories as CSV files and GeoTIFF maps. Table 4 shows the first six rows of the saved CSV file on average pixel-wise VWC values. *x* and *y* represent the map pixels' coordinates (in NAD83 projection system, "EPSG: 5070"); *layer* stores the pixels' mean values, and *date* shows the starting date of each time interval. The user can use **plot_aggregated_VWC** to visualize the maps saved by **aggregate_interval** with the same inputs and a *date* parameter specifying the starting date of the time intervals to plot. **aggregate_sum** can be run following a similar logic, except that no starting date needs to be defined when plotting.

Table 4. A subset of 7_day_mean.csv, saved by **aggregate_interval**.

	X	Y	Layer	Date
1	561512.8	2196288	0.228	2020-06-15
2	561542.8	2196288	0.248	2020-06-15
3	561572.8	2196288	0.221	2020-06-15
4	561602.8	2196288	0.236	2020-06-15
5	561512.8	2196258	0.227	2020-06-15
6	561542.8	2196258	0.246	2020-06-15

```
> plot_aggregated_VWC(5, "2020-06-15", "2020-08-15", date = "2020-06-29", frequency = 7, project = "WI_region")
```

plot_aggregated_VWC returns a similar *leaflet* plot as **pixelwise_sum** with four layers of *Mean*, *Min*, *Median*, *Max*, and *SD* of the pixels' VWC values during the 7-day interval starting from 29 June 2020 (Figure 7). From the plot, it was noted that the study area had an average surface (0–5 cm) VWC ranging from 0.19 to 0.32 m³ m⁻³ during the week of 29 June 2020, and some of the highest mean VWC values were located at the southwestern corner of the ROI.

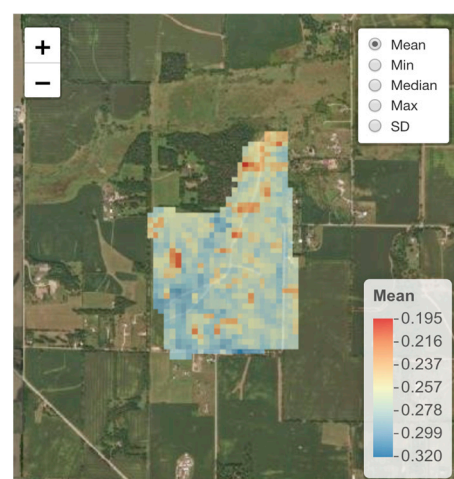


Figure 7. *leaflet* plot of the 7-day VWC average starting from 29 June 2020, as calculated by **aggregate_interval**.

The user can also compare the VWCs on a daily vs. weekly basis using the results saved by **VWC_map** and **aggregate_interval**, as demonstrated below:

```
> dates <- c("2020-07-06", "2020-07-13", "2020-07-20", "2020-07-27") # define the dates to plot
```

```

> pattern <- paste(paste0("VWC_5_mean_", dates), sep = "", collapse = "|") # define the files
(with specific VWC statistical characteristics) to plot
> maps <- stack(list.files("WI_region/VWC", pattern, full.names = T))
> plot(maps, zlim = c(0.15, 0.3), legend.args = list(text = "VWC (m3m-3", side = 3, font=2,
line = 0.5, cex = 0.75)) # set at same scale and define legend title

```

The daily VWC maps (Figure 8) show that the predicted surface soil moisture of the study area was similarly wet on the four days except for 20 July 2020, which was drier than the other three days. The trend of VWC maps was consistent with the time series plot (Figure 5), produced from the **area_sum**, where the average VWC values for 6, 13, and 27 July were approximately at the same level, while the mean VWC on 20 July was lower than the other three. On the other hand, when plotted using the weekly VWC maps (Figure 9), the week of 27 July 2020 experienced the lowest surface VWC value, while the week of 6 July 2020 had the wettest, and the weeks of 13 and 20 July had similar intermediate VWC values:

```

> WD <- "WI_region/VWC_aggregation/2020-06-15_2020-08-15/7days_5cm"
> pattern <- paste(paste0("mean_", dates), sep = "", collapse = "|")
> maps <- stack(list.files(paste0(WD, "/aggregated_VWC", pattern, full.names = T))
> plot(maps, zlim = c(0.15, 0.3), legend.args = list(text = TeX("VWC $m^3m^{-3}$"), side = 3,
font = 2, line = 0.5, cex = 0.75), axes=F) # set at same scale and define legend title

```

Both daily and weekly maps were consistent with the time series plot shown in Figure 4. It should also be noted that although the daily VWC on 20 July 2020 was the driest among the four, seeing from a larger temporal scale (weekly), the week of 27 July 2020 became drier compared with the other three weeks. This suggests that the **Aggregate_interval** function is useful for inspecting the intermediate- to long-term trend in VWC while daily VWC dynamics produced by the **VWC_map** capture short-term variations in soil moisture. A combination of these two functions will help the user understand the time-dependent variations/memories in VWC for a specific area [49].

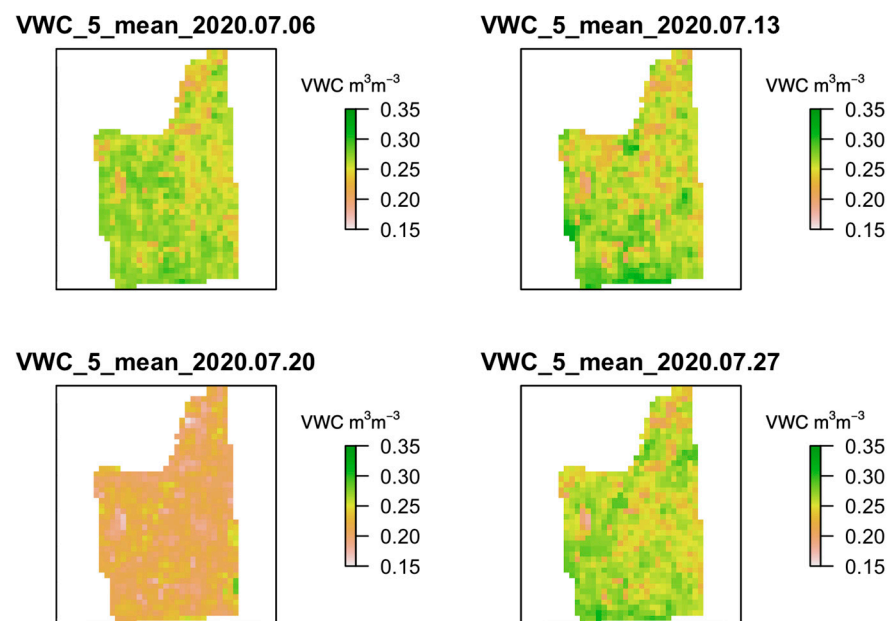


Figure 8. Four daily VWC maps, extracted from outputs saved by **VWC_map**.

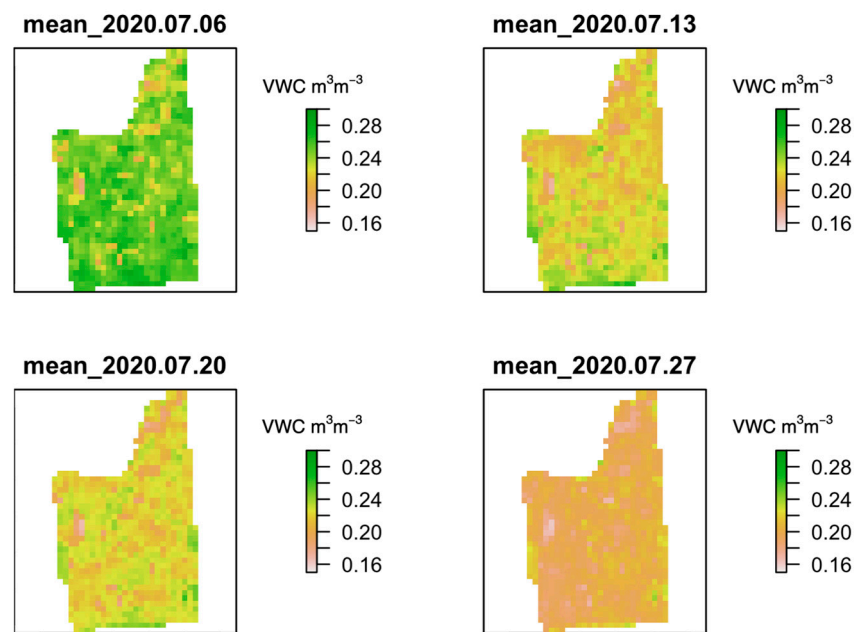


Figure 9. Four weekly VWC maps, extracted from outputs saved by `aggregate_interval`.

3.4. Extract Soil Moisture Data at Individual Sites

To extract soil VWC values from the generated maps to the points where measurements are available (saved in the `WI_SM.shp` file), the user can apply `point_extraction`:

```
> point_extraction("WI_SM.shp", TRUE, project = "WI_region")
> head(read.csv("WI_region/VWC_point_data.csv"))
```

Table 5 shows the VWC values extracted using `point_extraction`. The `ID` represents site identification, and `Longitude` and `Latitude` are the coordinates of the sites. `Date` corresponds to the date on which the VWC values are extracted, which should have the same range as defined in `VWC_map` (start date to end date). Columns `VWC_5_mean_pts` through `VWC_100_upper_pts` are the extracted VWC values (mean, sd, lower and upper bounds of 90% CI) at the specific depth (5 or 100). The user should be aware that `point_extraction` does not calculate VWC CIs for the points. Therefore, if the user did not save CI values in the mapping process (via `VWC_map` or `mosaic_region`), the 90% CI option in `point_extraction` (`percentile = TRUE`) will not be applicable when extracting values for the points.

Table 5. A subset of the extracted VWC values stored in `VWC_point_data.csv`.

ID	Longitude	Latitude	Date	VWC_5_Mean_pts	VWC_5_sd_pts	VWC_100_Mean_pts	VWC_100_sd_pts	VWC_5_Lower_pts	VWC_5_Upper_pts	VWC_100_Lower_pts	VWC_100_Upper_pts	
1	S2	-89.11825	42.57247	2020-06-15	0.268	0.082	0.356	0.097	0.153	0.391	0.224	0.471
2	S2	-89.11825	42.57247	2020-06-16	0.212	0.074	0.363	0.101	0.106	0.331	0.199	0.476
3	S2	-89.11825	42.57247	2020-06-17	0.249	0.087	0.358	0.098	0.092	0.367	0.206	0.476
4	S2	-89.11825	42.57247	2020-06-18	0.255	0.073	0.358	0.097	0.148	0.374	0.205	0.476
5	S2	-89.11825	42.57247	2020-06-19	0.212	0.078	0.348	0.103	0.112	0.358	0.188	0.468
6	S2	-89.11825	42.57247	2020-06-20	0.210	0.081	0.348	0.101	0.103	0.354	0.188	0.466

After obtaining the point VWC values, the user can plot the dynamics of VWC at a chosen depth for different sites using `site_variation`. This function can also be used on data generated from `VWC_point`, whose output has the same format as that of `point_extraction`.

```
> plots <- site_variation(5, TRUE, project = "WI_point")
> plots[[1]] # returns leaflet plot showing site locations
> plots[[2]] # returns time series plot showing point variation over time
```

The first plot returned from `site_variation` is a *leaflet* plot indicating the locations of the study sites, as shown in Figure 10a. The sites are highlighted using fixed circle makers and ID labels for easier identification. Since no sites are specified in the code, the second plot returned by `site_variation`, as shown in Figure 10b, is a time series plot visualizing the changes in VWC of 12 sites, faceted by site IDs, and the dates on the x-axis are labeled in the format of “yy-mm-dd.” In the time series plot, solid lines represent the mean, and gray bands represent the standard deviation; 90% CI will be added in the form of dashed lines (if available). The time series plot showed that all 12 sites experienced similar trends in the change in surface-level soil moisture, as controlled by precipitation and crop evapotranspiration, with low VWC values during August 2020 and an increase in the VWC values at the beginning of September. The magnitude of the VWC changes differed, with several sites having a steeper increase than others (S2, S13, etc.), as controlled by environmental factors such as soil texture and topography (both used as covariates in modeling soil moisture variations at the field level).

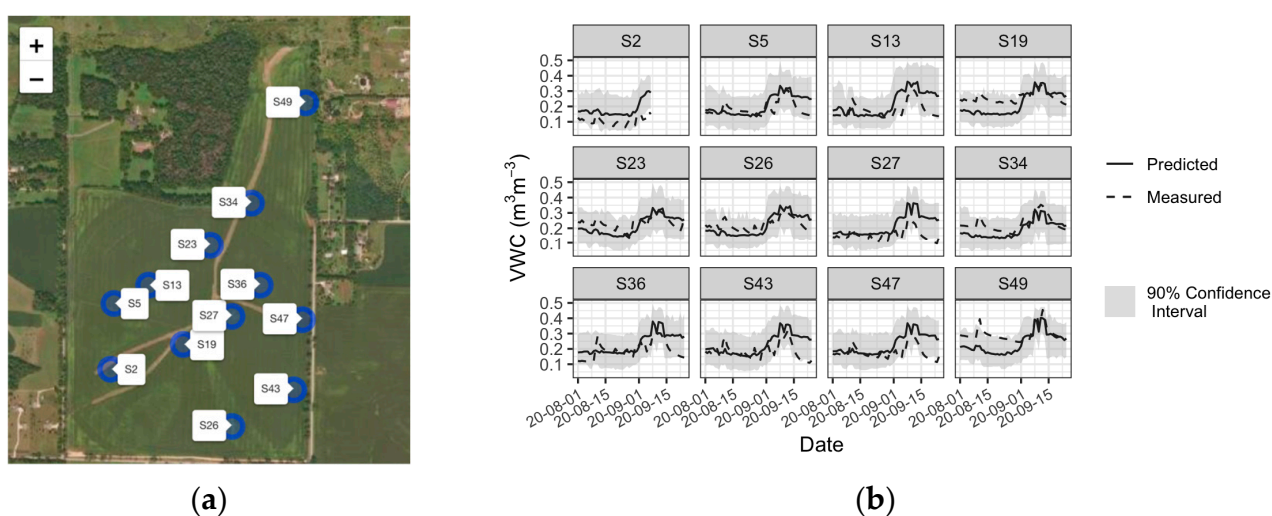


Figure 10. (a) leaflet plot marking the point locations; (b) time series plot showing the trends in change in VWC at 12 sites.

3.5. Evaluation of Model Estimation with Ground Truth Data

After VWC information is extracted to the points by `point_extraction`, `point_CI`, and `point_performance` can be used to evaluate the performance of the models by comparing the predicted values to the measured data (saved in “Huges_VWC.csv”). The two evaluation functions can also be applied to the point-based VWC datasets obtained from `VWC_point` with the same input parameters.

```
> point_CI("Huges_VWC.csv", 5, project = "WI_region")
```

`plot_CI` returns a similar time series plot as `site_variation`. As shown in Figure 11, most of the extracted VWC values on the points have a similar trend as the measured VWC values, with most of the measured values within the 90% confidence interval of the predicted values. To further evaluate the model performance at the selected fields, the user can calculate the performance statistics (R^2 , RMSE, NSE, KGE, etc.) by running `point_performance`:

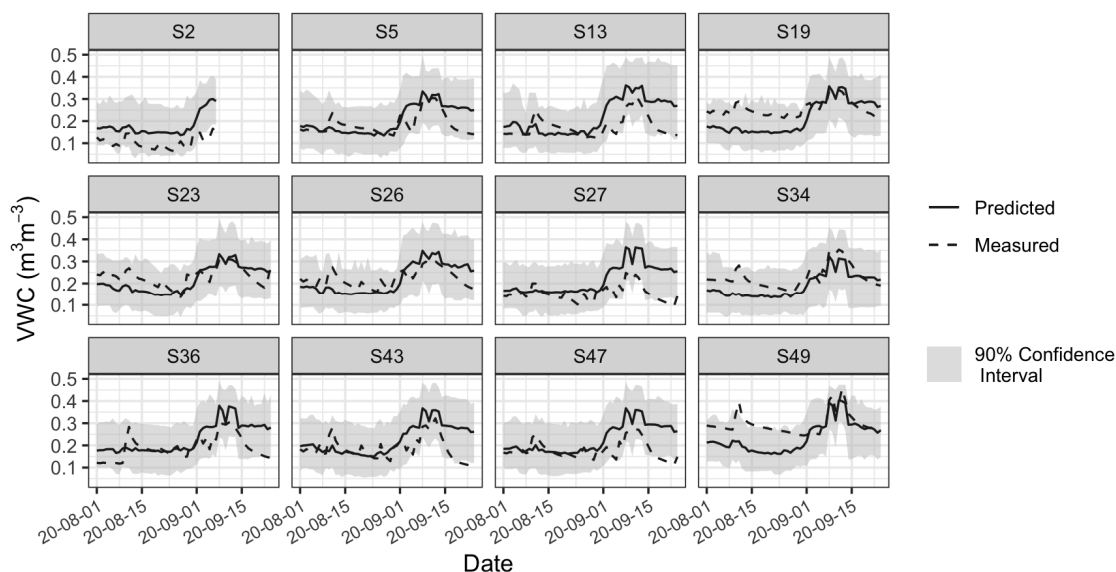


Figure 11. Time series plot as a result of calling `point_CI`.

```
> point_performance("Huges_VWC.csv", 5, TRUE, project = "WI_region")
```

Figure 12 illustrates the scatter plot of the predicted VWC against the field-measured VWC with the metrics R^2 , RMSE, Bias, and KGE. The predictions were smoothed as compared to the actual values (due to the use of quantile random forest models), which yielded a relatively low R^2 of 0.329 (Pearson’s $r = 0.57$). However, the small RMSE and bias (0.06 and $-0.00 \text{ m}^3 \text{ m}^{-3}$) and an intermediate KGE of 0.565 indicate the modeling result is reasonable at such a high spatial resolution given the soil texture is extremely heterogeneous across the study field (soils formed from glacial outwash, see data from [48]). The mediocre model performance at this site is also consistent with other studies that attempted to delineate soil moisture at the subfield level (e.g., 30 m resolution). For instance, when assimilating a land surface model with SMAP satellite data, due to the field-scale soil heterogeneity, the modeled and measured soil VWC at the soil surface had a median temporal correlation coefficient of 0.73 ± 0.13 and a median KGE of 0.52 ± 0.20 across the CONUS [12].

Validation

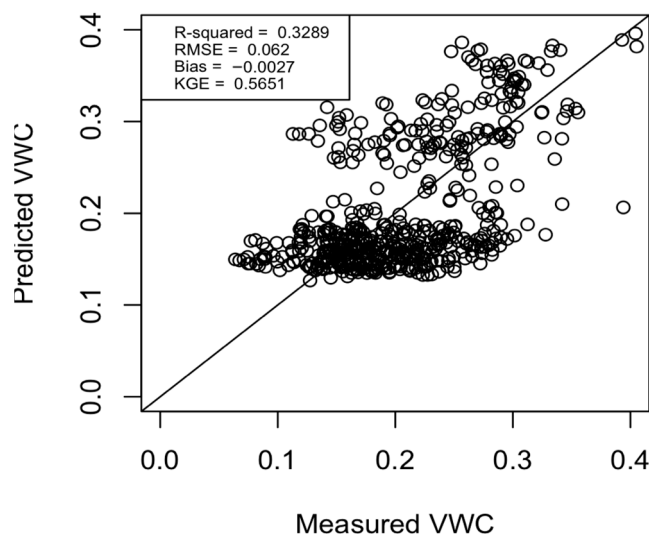


Figure 12. Evaluation plot returned by `point_performance`.

4. Discussion

4.1. Usability Evaluation

Careful selection of input arguments of the package's various functions is needed for mapping soil moisture of any study area or at multiple locations, given the tradeoff between spatial resolution/area size/number of study sites and computation time. Here, we provide some basic computational performance statistics for different sizes of mapping areas in Table 6 for the user, which can help the user select suitable parameters for different tasks. The computer used for testing the parameters has a 64-bit Windows 10 operation system, with a 12th Gen Intel(R) Core (TM) i5-12500 at 3.00 GHz and 128 GB of RAM.

Table 6. Computation time of the **VWC_map** and **VWC_point** functions for different tasks.

Function	No. of Dates	Size of Study Area/No. of Points	Resolution	Calculation of CIs	Computation Time	Size of Output Folders/Files
VWC_map						
Small region	45	0.703 km ²	100	T	33 min	1.93 MB
Large region	30	7229.816 km ²	500	F	110 min	82.1 MB
VWC_point						
	30	10 points	\	T	260 min	1.4 MB
	60	20 points	\	T	792 min	5.1 MB
	30	30 points	\	T	764 min	4.1 MB

Based on the results shown in Table 6 and Section 3.5, when the soil in the study area is strongly heterogeneous, the user should set the map resolution to a lower value (e.g., 100 m) to achieve a balance between the accuracy of the models as well as computation time.

4.2. Sustainability Plan

The current product (**mlhrsm 1.0**) relies on several satellite data as input covariates and produces surface and subsurface soil moisture estimates from 1 January 2016 to the present (tested until 31 July 2022). In case of failure, decommissioning, or adding certain satellite products in the future, an updated version of the product and the **mlhrsm** package will be updated using replacement satellite products, including VIIRS/NPP VNP21A1D for MODIS land surface temperature, Landsat 9 and Sentinel-2 for Landsat 8 bands and indices, Global Land Data Assimilation System (GLDAS) or North American Land Data Assimilation System (NLDAS) Noah Land Surface Model for precipitation and ET and SMAP-derived surface and subsurface soil moisture.

4.3. Computational Details

The results in this paper were obtained using R 4.2.1 with the **mlhrsm 1.0**, raster 3.5-29, sf 1.0-8, tidyverse 1.3.2, viridis 0.6.2, FedData 2.5.7, RColorBrewer 1.1-3, caret 6.0-93, chillR 0.72.8, leaflet 2.1.1, hydroGOF 0.4-0, quantregForest 1.3-7, randomForest 4.7-1.1, reshape2 1.4.4, rgdal 1.5-32, sp 1.5-0, lubridate 1.8.0, geojsonio 0.9.4, stars 0.5-6, devtools 2.4.4, and R.rsp 0.45.0 packages. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/>.

4.4. Application and Limitations

People can utilize the **mlhrsm** package for water resource management and agricultural planning. The package provides an easy way to predict and visualize soil moisture retrospectively, which can be especially useful for evaluating water use efficiency, drought and flood impacts, and better scheduling planting and harvesting. It also provides statistical gap-filling of soil moisture in space and time. Limitations of the package also exist due to the quality and availability of open-access remote sensing data and in situ soil moisture observations. Particularly, the use of an active microwave sensor (Sentinel-1) to estimate soil moisture is strongly affected by terrain roughness conditions, and it is

advised that the user should be cautious about the soil moisture maps produced in such regions. Moreover, the data we used to train the model were from stations in the US, and it is strongly recommended the users follow our instructions (Supplementary Materials) to add local training datasets to improve the performance of the model in other regions of the world.

5. Conclusions

To facilitate domain scientists (researchers, students, and educators) to better use earth observing satellite data for soil moisture mapping with machine learning algorithms and spatial and temporal analysis of soil moisture data, this paper describes a novel open-sourced R package, **mlhrsm**, developed for generating machine learning-based high-resolution soil moisture maps for soil surface (0–5 cm) and rootzone (0–1 m) at 30 to 500 m from daily to seasonally or time-series data across the continental United States. The user can choose the spatial and temporal resolutions of the soil moisture maps based on the knowledge of soil variability of the study site and management needs. It has many built-in functions for spatial and temporal analysis of the produced soil moisture maps or time series data. The users can also rebuild the machine learning models or adjust the code to map soil moisture outside the US. It is envisioned that a combination of the easy-to-use mapping functions and spatial–temporal analysis tools in this R package will promote the use of machine learning and artificial intelligence among non-specialists and help advance water-related scientific studies across scales and inform land managers with field-level soil moisture information for water resources management. To improve the performance of the machine learning models, data assimilation with a process-based model will be incorporated in the future version of the package.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/agronomy14030421/s1>, Supplementary Materials S1–S3.

Author Contributions: Conceptualization, Y.P., J.H., and Z.Z.; methodology, Y.P.; formal analysis, Y.P. and J.H.; data curation, J.H. and Z.Y.; writing—original draft preparation, Y.P., Z.Y., J.H., and Z.Z.; writing—review and editing, Y.P., Z.Y., J.H., and Z.Z.; visualization, Y.P.; funding acquisition, J.H. and Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Support for this research was provided by the USDA Agriculture and Food Research Initiative Foundational Program (Grant No. 2023-67021-40007), the USDA Agriculture and Food Research Initiative Foundational Program Accession # 1028199, the USDA Hatch project # 7002632 and the University of Wisconsin–Madison Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation. The findings and conclusions in this publication are those of the authors and should not be construed to represent any official USDA or U.S. Government determination or policy.

Data Availability Statement: The source code of the **mlhrsm** package is available on GitHub (https://github.com/soilsensingmonitoring/mlhrsm_1.0). The datasets for the case study are also available for downloading (https://github.com/soilsensingmonitoring/mlhrsm_1.0/tree/main/data). The documentation and “Help Pages” for the **mlhrsm** package and all the functions can be accessed by searching the package name “**mlhrsm**” in “Packages” within the RStudio after the package is installed. Descriptions of the input and output of all the functions from the **mlhrsm** package can be found in these “Help Pages”. We have also created a Google Group email list (mlhrsm@googlegroups.com) for continuous user support on applications of the soil moisture maps for scientific research and water resources management and will collect feedback from users for future improvement in the product.

Acknowledgments: The authors would like to thank Randy Hughes and Wilden Hughes (Hughes Farms, WI, USA) for allowing us to access their field to collect in situ soil moisture measurements used in the demonstration example and for Sumanta Chatterjee and Jie Hu for assistance in the field data collection.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Mladenova, I.E.; Bolten, J.D.; Crow, W.T.; Anderson, M.C.; Hain, C.R.; Johnson, D.M.; Mueller, R. Intercomparison of Soil Moisture, Evaporative Stress, and Vegetation Indices for Estimating Corn and Soybean Yields Over the U.S. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 1328–1343. [[CrossRef](#)]
- Robinson, D.A.; Campbell, C.S.; Hopmans, J.W.; Hornbuckle, B.K.; Jones, S.B.; Knight, R.; Ogden, F.; Selker, J.; Wendroth, O. Soil Moisture Measurement for Ecological and Hydrological Watershed-Scale Observatories: A Review. *Vadose Zone J.* **2008**, *7*, 358–389. [[CrossRef](#)]
- Dirmeyer, P.A.; Halder, S. Sensitivity of Numerical Weather Forecasts to Initial Soil Moisture Variations in CFSv2. *Weather Forecast.* **2016**, *31*, 1973–1983. [[CrossRef](#)]
- Korošak, Ž.; Suhadolnik, N.; Pleteršek, A. The Implementation of a Low Power Environmental Monitoring and Soil Moisture Measurement System Based on UHF RFID. *Sensors* **2019**, *19*, 5527. [[CrossRef](#)]
- Vereecken, H.; Huisman, J.A.; Bogaen, H.; Vanderborght, J.; Vrugt, J.A.; Hopmans, J.W. On the Value of Soil Moisture Measurements in Vadose Zone Hydrology: A Review. *Water Resour. Res.* **2008**, *44*, 2008WR006829. [[CrossRef](#)]
- Wang, L.; Qu, J.J. Satellite remote sensing applications for surface soil moisture monitoring: A review. *Front. Earth Sci. China* **2009**, *3*, 237–247. [[CrossRef](#)]
- Babaeian, E.; Sadeghi, M.; Jones, S.B.; Montzka, C.; Vereecken, H.; Tuller, M. Ground, proximal, and satellite remote sensing of soil moisture. *Rev. Geophys.* **2019**, *57*, 530–616. [[CrossRef](#)]
- Li, Z.L.; Leng, P.; Zhou, C.; Chen, K.S.; Zhou, F.C.; Shang, G.F. Soil moisture retrieval from remote sensing measurements: Current knowledge and directions for the future. *Earth-Sci. Rev.* **2021**, *218*, 103673. [[CrossRef](#)]
- Ochsner, T.E.; Cosh, M.H.; Cuenca, R.H.; Dorigo, W.A.; Draper, C.S.; Hagimoto, Y.; Kerr, Y.H.; Larson, K.M.; Njoku, E.G.; Small, E.E.; et al. State of the Art in Large-Scale Soil Moisture Monitoring. *Soil Sci. Soc. Am. J.* **2013**, *77*, 1888–1919. [[CrossRef](#)]
- Petropoulos, G.P.; Ireland, G.; Barrett, B. Surface Soil Moisture Retrievals from Remote Sensing: Current Status, Products & Future Trends. *Phys. Chem. Earth Parts A/B/C* **2015**, *83–84*, 36–56. [[CrossRef](#)]
- Sadeghi, M.; Babaeian, E.; Tuller, M.; Jones, S.B. The optical trapezoid model: A novel approach to remote sensing of soil moisture applied to Sentinel-2 and Landsat-8 observations. *Remote Sens. Environ.* **2017**, *198*, 52–68. [[CrossRef](#)]
- Vergopolan, N.; Chaney, N.W.; Pan, M.; Sheffield, J.; Beck, H.E.; Ferguson, C.R.; Torres-Rojas, L.; Sadri, S.; Wood, E.F. SMAP-HydroBlocks, a 30-m Satellite-Based Soil Moisture Dataset for the Conterminous US. *Sci. Data* **2021**, *8*, 264. [[CrossRef](#)]
- Entekhabi, D.; Njoku, E.G.; O'Neill, P.E.; Kellogg, K.H.; Crow, W.T.; Edelstein, W.N.; Entin, J.K.; Goodman, S.D.; Jackson, T.J.; Johnson, J.; et al. The Soil Moisture Active Passive (SMAP) Mission. *Proc. IEEE* **2010**, *98*, 704–716. [[CrossRef](#)]
- Kerr, Y.H.; Waldteufel, P.; Richaume, P.; Wigneron, J.P.; Ferrazzoli, P.; Mahmoodi, A.; Al Bitar, A.; Cabot, F.; Gruhier, C.; Juglea, S.E.; et al. The SMOS Soil Moisture Retrieval Algorithm. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 1384–1403. [[CrossRef](#)]
- Chew, C.C.; Small, E.E. Soil moisture sensing using spaceborne GNSS reflections: Comparison of CYGNSS reflectivity to SMAP soil moisture. *Geophys. Res. Lett.* **2018**, *45*, 4049–4057. [[CrossRef](#)]
- Kim, H.; Lakshmi, V. Use of cyclone global navigation satellite system (CyGNSS) observations for estimation of soil moisture. *Geophys. Res. Lett.* **2018**, *45*, 8272–8282. [[CrossRef](#)]
- Clarizia, M.P.; Pierdicca, N.; Costantini, F.; Floury, N. Analysis of CYGNSS data for soil moisture retrieval. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 2227–2235. [[CrossRef](#)]
- Senyurek, V.; Lei, F.; Boyd, D.; Kurum, M.; Gurbuz, A.C.; Moorhead, R. Machine learning-based CYGNSS soil moisture estimates over ISMN sites in CONUS. *Remote Sens.* **2020**, *12*, 1168. [[CrossRef](#)]
- Mladenova, I.E.; Bolten, J.D.; Crow, W.; Sazib, N.; Reynolds, C. Agricultural Drought Monitoring via the Assimilation of SMAP Soil Moisture Retrievals Into a Global Soil Water Balance Model. *Front. Big Data* **2020**, *3*, 10. [[CrossRef](#)]
- Reichle, R.; Lannoy, G.D.; Koster, R.; Crow, W.; Kimball, J.; Liu, Q. SMAP L4 Global 3-Hourly 9 km EASE-Grid Surface and Root Zone Soil Moisture Geophysical Data, Version 4 (SPL4SMGP). Distributed by NASA National Snow and Ice Data Center Distributed Active Archive Center. 2018. Available online: <https://nsidc.org/data/spl4smgp/versions/4> (accessed on 20 January 2024). [[CrossRef](#)]
- Kumar, S.; Petersliard, C.; Tian, Y.; Houser, P.; Geiger, J.; Olden, S.; Lighty, L.; Eastman, J.; Doty, B.; Dirmeyer, P. Land Information System: An Interoperable Framework for High Resolution Land Surface Modeling. *Environ. Model. Softw.* **2006**, *21*, 1402–1415. [[CrossRef](#)]
- Zhang, C.; Yang, Z.; Zhao, H.; Sun, Z.; Di, L.; Bindlish, R.; Liu, P.-W.; Colliander, A.; Mueller, R.; Crow, W.; et al. Crop-CASMA: A Web Geoprocessing and Map Service Based Architecture and Implementation for Serving Soil Moisture and Crop Vegetation Condition Data over U.S. Cropland. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102902. [[CrossRef](#)]
- Huang, J.; Desai, A.R.; Zhu, J.; Hartemink, A.E.; Stoy, P.C.; Loheide, S.P.; Bogaen, H.R.; Zhang, Y.; Zhang, Z.; Arriaga, F. Retrieving Heterogeneous Surface Soil Moisture at 100 m Across the Globe via Fusion of Remote Sensing and Land Surface Parameters. *Front. Water* **2020**, *2*, 578367. [[CrossRef](#)]
- Greifeneder, F.; Notarnicola, C.; Wagner, W. A Machine Learning-Based Approach for Surface Soil Moisture Estimations with Google Earth Engine. *Remote Sens.* **2021**, *13*, 2099. [[CrossRef](#)]

25. Tischler, M.; Garcia, M.; Peterslidard, C.; Moran, M.; Miller, S.; Thoma, D.; Kumar, S.; Geiger, J. A GIS Framework for Surface-Layer Soil Moisture Estimation Combining Satellite Radar Measurements and Land Surface Modeling with Soil Physical Property Estimation. *Environ. Model. Softw.* **2007**, *22*, 891–898. [[CrossRef](#)]
26. Meinshausen, N. Quantile Regression Forests. *J. Mach. Learn. Res.* **2006**, *7*, 983–999.
27. Chatterjee, S.; Huang, J.; Hartemink, A.E. Establishing an Empirical Model for Surface Soil Moisture Retrieval at the U.S. Climate Reference Network Using Sentinel-1 Backscatter and Ancillary Data. *Remote Sens.* **2020**, *12*, 1242. [[CrossRef](#)]
28. Batchu, V.; Nearing, G.; Gulshan, V. A Deep Learning Data Fusion Model Using Sentinel-1/2, SoilGrids, SMAP, and GLDAS for Soil Moisture Retrieval. *J. Hydrometeorol.* **2023**, *24*, 1789–1823. [[CrossRef](#)]
29. Mirvakhabova, L.; Pukalchik, M.; Matveev, S.; Tregubova, P.; Oseledets, I. Field heterogeneity detection based on the modified FastICA RGB-image processing. *J. Phys. Conf. Ser.* **2018**, *1117*, 012009. [[CrossRef](#)]
30. Vasilyeva, N.A.; Vladimirov, A.; Vasiliev, T. Image Recognition for Large Soil Maps Archive Overview: Metadata Extraction and Georeferencing Tool Development. In Proceedings of the International Conference on Data Analytics and Management in Data Intensive Domains, Moscow, Russia, 26–29 October 2021; Springer International Publishing: Cham, Switzerland, 2021; pp. 193–204.
31. Mirpulatov, I.; Illarionova, S.; Shadrin, D.; Burnaev, E. Pseudo-Labeling Approach for Land Cover Classification through Remote Sensing Observations with Noisy Labels. *IEEE Access* **2023**, *11*, 82570–82583. [[CrossRef](#)]
32. Gao, Y.; Walker, J.P.; Ye, N.; Panciera, R.; Monerris, A.; Ryu, D.; Rudiger, C.; Jackson, T.J. Evaluation of the Tau–Omega Model for Passive Microwave Soil Moisture Retrieval Using SMAPEX Datasets. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 888–895. [[CrossRef](#)]
33. Entekhabi, D.; Rodriguez-Iturbe, I. Analytical Framework for the Characterization of the Space-Time Variability of Soil Moisture. *Adv. Water Resour.* **1994**, *17*, 35–45. [[CrossRef](#)]
34. Merz, B.; Plate, E.J. An Analysis of the Effects of Spatial Variability of Soil and Soil Moisture on Runoff. *Water Resour. Res.* **1997**, *33*, 2909–2922. [[CrossRef](#)]
35. Yoo, C.; Kim, S. EOF Analysis of Surface Soil Moisture Field Variability. *Adv. Water Resour.* **2004**, *27*, 831–842. [[CrossRef](#)]
36. Famiglietti, J.S.; Ryu, D.; Berg, A.A.; Rodell, M.; Jackson, T.J. Field Observations of Soil Moisture Variability across Scales. *Water Resour. Res.* **2008**, *44*, 2006WR005804. [[CrossRef](#)]
37. Brocca, L.; Melone, F.; Moramarco, T.; Morbidelli, R. Spatial-temporal Variability of Soil Moisture and Its Estimation across Scales. *Water Resour. Res.* **2010**, *46*, 2009WR008016. [[CrossRef](#)]
38. Ma, Y.; Zhang, Z.; Yang, H.L.; Yang, Z. An Adaptive Adversarial Domain Adaptation Approach for Corn Yield Prediction. *Comput. Electron. Agric.* **2021**, *187*, 106314. [[CrossRef](#)]
39. Ojha, N.; Merlin, O.; Molero, B.; Suere, C.; Olivera-Guerra, L.; Ait Hssaine, B.; Amazirh, A.; Al Bitar, A.; Escorihuela, M.; Er-Raki, S. Stepwise Disaggregation of SMAP Soil Moisture at 100 m Resolution Using Landsat-7/8 Data and a Varying Intermediate Resolution. *Remote Sens.* **2019**, *11*, 1863. [[CrossRef](#)]
40. Liu, P.-W.; Bindlish, R.; O’Neill, P.; Fang, B.; Lakshmi, V.; Yang, Z.; Cosh, M.H.; Bongiovanni, T.; Collins, C.H.; Starks, P.J.; et al. Thermal Hydraulic Disaggregation of SMAP Soil Moisture Over the Continental United States. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 4072–4092. [[CrossRef](#)]
41. Feldman, A.F.; Short Gianotti, D.J.; Dong, J.; Akbar, R.; Crow, W.T.; McColl, K.A.; Konings, A.G.; Nippert, J.B.; Tumber-Dávila, S.J.; Holbrook, N.M.; et al. Remotely Sensed Soil Moisture Can Capture Dynamics Relevant to Plant Water Uptake. *Water Resour. Res.* **2023**, *59*, e2022WR033814. [[CrossRef](#)]
42. Roy, D.P.; Ju, J.; Lewis, P.; Schaaf, C.; Gao, F.; Hansen, M.; Lindquist, E. Multi-Temporal MODIS–Landsat Data Fusion for Relative Radiometric Normalization, Gap Filling, and Prediction of Landsat Data. *Remote Sens. Environ.* **2008**, *112*, 3112–3130. [[CrossRef](#)]
43. Chen, J.; Zhu, X.; Vogelmann, J.E.; Gao, F.; Jin, S. A Simple and Effective Method for Filling Gaps in Landsat ETM+ SLC-off Images. *Remote Sens. Environ.* **2011**, *115*, 1053–1064. [[CrossRef](#)]
44. Carrasco, L.; O’Neil, A.; Morton, R.; Rowland, C. Evaluating Combinations of Temporally Aggregated Sentinel-1, Sentinel-2 and Landsat 8 for Land Cover Mapping with Google Earth Engine. *Remote Sens.* **2019**, *11*, 288. [[CrossRef](#)]
45. McCuen, R.H.; Knight, Z.; Cutter, A.G. Evaluation of the Nash–Sutcliffe Efficiency Index. *J. Hydrol. Eng.* **2006**, *11*, 597–602. [[CrossRef](#)]
46. Gupta, H.V.; Kling, H.; Yilmaz, K.K.; Martinez, G.F. Decomposition of the Mean Squared Error and NSE Performance Criteria: Implications for Improving Hydrological Modelling. *J. Hydrol.* **2009**, *377*, 80–91. [[CrossRef](#)]
47. Wijewardane, N.K.; Ge, Y.; Wills, S.; Loecke, T. Prediction of Soil Carbon in the Conterminous United States: Visible and Near Infrared Reflectance Spectroscopy Analysis of the Rapid Carbon Assessment Project. *Soil Sci. Soc. Am. J.* **2016**, *80*, 973–982. [[CrossRef](#)]
48. Chatterjee, S.; Hartemink, A.E.; Triantafyllis, J.; Desai, A.R.; Soldat, D.; Zhu, J.; Townsend, P.A.; Zhang, Y.; Huang, J. Characterization of Field-Scale Soil Variation Using a Stepwise Multi-Sensor Fusion Approach and a Cost-Benefit Analysis. *CATENA* **2021**, *201*, 105190. [[CrossRef](#)]
49. Koster, R.D.; Suarez, M.J. Soil Moisture Memory in Climate Models. *J. Hydrometeorol.* **2001**, *2*, 558–570. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.