Estimating Daily Soil Moisture at High Spatial Resolution for Drought Monitoring by Fusing Multi-Source Data Based on Random Forest

--Data Management Plan

XINRONG LI

July, 2023

SUPERVISORS:

Dr. Y. Zeng
Prof.dr. Z. Su

ADVISORS:

MSc. Q Han

Ir. A.M. van Lieshout

Contents

1. INTRODUCTION

In the context of global warming, the acceleration of the water cycle increases the odds of extreme weather events. The Netherlands experienced one of the most severe droughts on record in 2018, which posed threats to agriculture, ecology, etc., and caused huge economic losses to the entire country. Furthermore, many studies indicate there is a trend of worsening drought in the Netherlands under climate change (Cook et al., 2018; Philip et al., 2020; Zhao et al., 2020).

Soil moisture (SM) is an indicator for drought monitoring because of its essential role in the water and energy cycle (André et al., 1986; Idso et al., 1975). Drought monitoring aims to track the severity and location of drought. Detailed drought descriptions help decision-makers develop measures to reduce drought-related losses. However, local/regional drought monitoring is facing difficulties with the lack of high spatiotemporal resolution SM data. This is mainly due to the fact that in situ measurements provide reliable point-scale high temporal resolution SM observation but lack the spatial coverage. While remotely sensed SM data provide local/regional spatial monitoring they are at very coarse spatial resolution (e.g., 25km, 9km, 1km). Therefore, we need a high-resolution SM product for better field water condition monitoring in the Netherlands to provide more accurate drought monitoring across the country.

In situ measurement is regarded as the most reliable and robust method to obtain SM (Yang et al., 2021). The in situ measurement networks in Raam region and Twente in the Netherlands provide references for remote sensing SM product validation (Benninga et al., 2018; Dente et al., 2012). Because of the high investment and maintenance cost of sensors, in situ measurements can only detect SM in a limited area. As an alternative, remote sensing technique has been introduced to SM monitoring for years. Passive and active remotely sensed SM is obtained by inversing the measured emitted or reflected microwave radiation from the land surface. Passive microwave radiometers provide frequent but coarse resolution SM observations while active microwave sensors retrieve SM at higher spatial resolution with low frequency. Although microwave technology has been greatly improved to better retrieve SM in recent years, it is still facing the difficulty to remove vegetation effects on the microwave emission of soils. Optical remotely sensed SM is derived from the relationships between land surface parameters, and thermal approaches rely on variations in soil surface temperature caused by different SM. Compared with passive and active remotely sensed SM, optical and thermal methods have better spatiotemporal resolutions, but they are sensitive to both vegetation and the weather conditions (Zhuang et al., 2023).

Coarse remotely sensed SM limits the application of SM at finer spatial resolution. Fortunately, data fusion approach can help address partially this issue. Data fusion is the process of assembling multiple data to represent a more useful real-world object (Zeng et al., 2016; Zhuang et al., 2020). Zhan et al. (2006) merged 36 km radiometer brightness temperature and 3 km radar backscatter with ancillary data including surface roughness, vegetation water content and surface temperature to generate the SM data at 9 km. Das et al. (2011) generated the gridded 9 km SM by integrating 36 km near-surface soil moisture

L-band radiometer retrievals and 3km L-band radar observations. Same as Zhan et al.'s study, Das et al. took advantage of the passive and active SM observations. Estimating SM by fusing optical image with microwave data was proposed in SM retrieval studies which combined the strengths of microwaves and optical sensing techniques (He et al., 2014; Mattar et al., 2012; Notarnicola et al., 2006). The arrival of Sentinel 1 and Sentinel 2 give us the chance to map SM with higher details. The twin Sentinel 1 satellites (Sentinel 1a and Sentinel 1b) provide high spatial resolution SAR images every 6 days while Sentinel 2a and Sentinel 2b provide high spatial resolution Visible (VNIR) and Near Infra-Red (NIR) to the Short Wave Infra-Red (SWIR) images with a 5-day revisit frequency (Drusch et al., 2012; Potin, 2019). The studies synthesizing sentinel 1 SAR images with sentinel 2 can generate SM with better resolution (Hajj et al., 2017; Ma et al., 2020), so do other SM fusion studies using multiple sensor. However, the high computational cost of the fusion algorithms restricts the generation of near real-time SM.

Lately, the machine learning method was applied to downscale SM (Su et al., 2020). The computationally-efficient machine learning (ML) provides a possibility to better establish nonlinear relationships between environmental factors and SM. Abbaszadeh et al. (2019) proposed a random forest (RF) based model with predictors including normalized difference vegetation index (NDVI), land surface temperature (LST), precipitation, elevation and soil texture to generate 1 km resolution SM. Zhang et al. (2021) applied in-situ observation-constrained RF for the SM estimates at a global scale with the same predictors, and the spatiotemporal patterns of the SM were compared against the in-situ observations, showing reasonable agreements. Lv et al. (2021) used environmental variables such as LST, the NDVI, the normalized shortwave-infrared difference bare soil moisture indices (NSDSI), the digital elevation model (DEM), and calculated slope data (SLOPE) to estimate SM at 1 km resolution with artificial neural networks (ANN). In the study of estimating SM with gradient boosting decision tree regression (GBDT) over the Tibetan plateau, 26 indices frequently used in SM downscaling were analyzed by filter method for their relative importance and high relative importance indices Enhanced Vegetation Index (EVI), Distance Drought Index (DDI), Modified Perpendicular Drought Index (MPDI), Modified Shortwave Infrared Perpendicular Water Stress (MSPSI), NDVI, Modified Perpendicular Drought Index (PDI), Shortwave Single Slope Index (SASI) (Wei et al., 2019). Although the above studies lead to acceptable predictions, feature selection is still a challenge in ML-based SM downscaling (Sachindra & Kanae, 2019). Appropriate variables not only reduce the complexity of the model but also result in a better estimation. Apart from predictors, the choice of algorithm also affects the SM estimate. There are many studies that compared the performance of different algorithms in generating SM products. Im et al. (2016b) compared the performance of random forest (RF), boosted regression trees, and cubist in generating 1 km SM in Korea and Australia. The result indicates the superior performance of RF with high accuracy and robustness. In the comparison study of six different algorithms in estimating SM, the effectiveness of RF was confirmed again (Liu et al., 2020). The same result was seen in the Yan and Bai (2020)'s comparison study. In summary, RF is the best-performing ML algorithm in the existing SM studies.

In this study, we use sentinel 1 Ground Range Detected (GRD) dataset and vegetation indices (EVI and LAI) from sentinel 2 with environmental variables (precipitation, soil texture, Surface Reflectance and ground water level (GWL)) and 1km SM to generate a high-resolution (10m) SM product for the Netherland by RF. The 1km dataset consist of aggerated 1 km predictors and 1km SM will be used to train, validate and test the model. Then we use the 10m predictors to generate the 10m SM product for the Netherlands. The generated high resolution SM product is evaluated by the in situ measurement and the importance of the predictors are analyzed. We expect the generated 10 m SM product can better reveal the field water conditions to achieve more accurate drought monitoring in the Netherlands under global warming.

Due to the computational limitations, the 10m soil moisture data we generated is only for the Twente region in this study, the workflow of which can be easily scaled up to the whole Netherlands.

2. DATA DESCRIPTION

In this research, auxiliary data including precipitation, LST, soil texture, surface reflectance, GWL will be selected as predictors with sentinel 1 level 1 GRD and vegetation indices (EVI and LAI) from sentinel 2 and 1 km SM simulated from in situ measurement to generate daily SM at 10 m. Other details of the dataset are shown in the following table.

Table 3.1 Required Dataset

Variable	Description	Source	Data Duration	Resolution	Unit
Precipitation	The KNMI precipitation dataset provides gridded 24 hour precipitation accumulations from climatological gauge-adjusted radar dataset for The Netherlands	KNMI Data Platform	01/01/2008-01/01/2023	1 km/Hourly	mm
LST	LST_day and LST_night are obtained through interpolation from MODIS 1km LST_day and LST_night data.	(Han et al., 2023)	01/01/2000-01/01/2020	1 km/ Daily	°C
Soil Texture	Soil texture is the proportion of sand, silt and clay content. sized particles that make up the mineral fraction of the soil.	OpenGeoHub	/	30m	clay % sand % bulk 10kg/m³
LAI	The Leaf Area Index is an important indicator reflecting the growth status of the plant population. It is the one-sided green leaf area over a unit of land.	Sentinel-2	28/03/2017-present	10m/ 5d	/
		Modis	18/02/2000-present	500m/8d	/

EVI	Enhanced vegetation index is an 'optimized' vegetation index. It is calculated by NIR, red, C1, C2, blue and L bands observed by Sentinel-2.	Sentinel-2	28/03/2017-present	10m/ 5d	/
		Modis	18/02/2000-present	250m/16d	/
					K
GWL	The 1km GWL is generated from The DINO groundwater database.	DINO	Depend on station	pointed	m
GRD	The sentienl-1 L1 Ground Range Detected dataset provided by GEE is a calibrated, ortho-corrected product.	Sentienl-1	01/10/2014-present	10m/6d	dB
SM	The in situ measurement datasets are from the Raam region and Twente.	In situ measurements(Dente et al., 2012)	01/01/2016-31/12-2019	15 minutes	cm^3/cm^3
SM	The 1km SM was generated by global in situ dataset.	(Han et al., 2023)	01/01/2000-01/01/2020	1km/daily	cm^3/cm^3

3. DATA PREPROCESSING AND HARMONIZATION

Since our model trains at 1km daily resolution and predict at 10m daily resolution, we need to resample all the predictor variables into 1km daily and 10m daily resolution.

Before the aggregation, we remove the clouds of Sentienl2 data by the characteristics of the QA band and then calculate EVI and LAI follow the equations below (R. Zhang et al., 2022) (Boegh et al., 2002).Then we resample and interpolate the EVI and LAI into 1km daily and 10m daily resolution. The code(in Google Earth Engine) can be accessed via the link:
https://code.earthengine.google.com/2dd8e31b7a4430270b9996b4536f9d34

$$EVI = G\frac{N-R}{N+C_1R-C_2B+L} \quad \text{eq.1}$$

$$LAI = 3.618 * EVI - 0.118 \quad \text{eq.2}$$

For GWL, we convert it from pointed to raster by kriging. The code is shown as follows.

```python
import pandas as pd
from pykrige.ok import OrdinaryKriging
from matplotlib import pyplot as plt         # plotting
import numpy as np                        # dense matrices
from osgeo import gdal,osr    # sparse matrices

data = pd.read_csv(r"D:\gw_kriging\20170101.csv")
outTif=r"D:\gw_kriging\20170101.tiff"
df= pd.DataFrame(data)
lat = (np.array((data.loc[:, 'Y_WGS']).tolist())).reshape(-1, 1)
lon = (np.array((data.loc[:, 'X_WGS']).tolist())).reshape(-1, 1)
Val = (np.array((data.loc[:, 'Value']).tolist())).reshape(-1, 1)
time=(np.array((data.loc[:, 'Time']).tolist())).reshape(-1, 1)

grid_lon = np.linspace(50,54,num=41)
grid_lat = np.arange(3, 7.6, 0.1)

ok3d = OrdinaryKriging(lon,lat,Val, variogram_model="linear", verbose=False,enable_plotting=False,)
k3d1, ss3d = ok3d.execute("grid", grid_lon, grid_lat)
xgrid, ygrid = np.meshgrid(grid_lon, grid_lat)
df_grid = pd.DataFrame(dict(long=xgrid.flatten(),lat=ygrid.flatten()))
df_grid["Krig_linear"] = k3d1.flatten()
fig, (ax1) = plt.subplots(1)
ax1.imshow(k3d1, origin="lower")
ax1.set_title("ordinary kriging")
plt.tight_layout()
plt.show()

LonMin, LatMax, LonMax, LatMin = [df_grid.long.min(), df_grid.lat.max(), df_grid.long.max(), df_grid.lat.min()]
N_Lat = len(grid_lat)
N_Lon = len(grid_lon)
Lon_Res = (LonMax - LonMin) / (float(N_Lon))
Lat_Res = (LatMax - LatMin) / (float(N_Lat))
Tif_ds = gdal.GetDriverByName('GTiff').Create(outTif, N_Lon, N_Lat, 1, gdal.GDT_Float32)
geotransform = (LonMin, Lon_Res, 0, LatMax, 0, -Lat_Res)
Tif_ds.SetGeoTransform(geotransform)
srs = osr.SpatialReference()
srs.ImportFromEPSG(4326)
Tif_ds.SetProjection(srs.ExportToWkt())
Tif_ds.GetRasterBand(1).SetNoDataValue(-9999)
Tif_ds.GetRasterBand(1).WriteArray(k3d1)
Tif_ds.FlushCache()
TMP_ds = None
```

For GRD, we use the bilinear and temporal interpolation for the resampling. The code(in Google Earth Engine) can be accessed via the link: https://code.earthengine.google.com/?scriptPath=users%2Fxli-14%2Ftry%3AGRD_1km_10m_all

For other predictor variables, the resampling process followed the function defined in the code of GRD and EVI/LAI.

4.MODEL BUILDING

4.1 Data splitting

After complementation the data processing and harmonization, we extract the pixel values of images at 1km resolution by GDAL and save them as csv. Based on the seasonal variation of SM, we spilt the 1km dataset by time that validation set consists of data from January, April, July, and October of 2018 (first month of each season), while the test set includes data from the same months in 2019. The dataset can be accessed via the following links:

https://crib.utwente.nl/geospatialhub/hub/user-redirect/lab/tree/model_1km/training_set_0510.csv

https://crib.utwente.nl/geospatialhub/hub/user-redirect/lab/tree/model_1km/training_set_0510.csv

https://crib.utwente.nl/geospatialhub/hub/user-redirect/lab/tree/model_1km/test_set_0510.csv

4.2 Model Tuning

We tune the hyperparameters n_estimators, min_samples_leaf in the range of 10 to 200, and 10 to 50 respectively by bayesian optimization. Bayesian optimization is the method used in machine learning for hyperparameter tuning and model optimization which combines Bayesian inference and optimization algorithms (Snoek et al., 2012). The target for an optimization problem depends on the specific case. Our study, prediction of 10m SM, is a regression problem, therefore we select RMSE as the target. Futhermore, we add a minus to RMSE for application since bayesian optimization aims to solve maximization problems (in the context of model performance, a lower RMSE indicates better performance). The code is shown as follows.

```
import pandas as pd
import numpy as np
from bayes_opt import BayesianOptimization
from sklearn.metrics import mean_squared_error
import sklearn
from sklearn.ensemble import RandomForestRegressor
from scipy.stats import pearsonr
from sklearn.metrics import r2_score

```

```python
dataset = pd.read_csv('validation_set_0510.csv')
dataset.head()
feature_names =['GRD','GW','preci','bulk','clay'
,'sand','LST_day','LST_night','EVI_modis','LAI_modis']
label = "SM"
X = dataset[feature_names].values
y = dataset[label].values
def gbdt_cv(n_estimators,min_samples_leaf):

    rf=RandomForestRegressor(n_estimators=int(n_estimators),
min_samples_leaf=int(min_samples_leaf),n_jobs=-1).fit(X, y)
    y_pre=rf.predict(X)

    df=pd.DataFrame({'SM_pre':y_pre,'SM':y})

    RMSE=np.sqrt(mean_squared_error(df['SM'], df['SM_pre']))
    corr, p_value = pearsonr(df['SM'], df['SM_pre'])

    return -RMSE

gbdt_op = BayesianOptimization(
        gbdt_cv,
        {'n_estimators': (10,200)
      ,'min_samples_leaf': (10, 50)

        })

gbdt_op.maximize(iter=10,init_points=0)
print(gbdt_op.max)
print(gbdt_op.max)
```

4.3 Model Training

Our random forest model is trained with the training set by sklearn. The code is shown as follows. The model can be accessed via the link:
https://crib.utwente.nl/geospatialhub/hub/user-redirect/lab/tree/model_1km/rf_r_0.92.pkl

```python
import pandas as pd
from sklearn import ensemble
import pickle
df = pd.read_csv('training_set_051022.csv')
feature_names =['GRD','GW','preci','bulk','clay','sand','LST_day','LST_night','EVI_modis','LAI_modis']
label = "SM"
X = df[feature_names]
y = df[label]
rf = ensemble.RandomForestRegressor(n_estimators=126,min_samples_leaf=10,n_jobs=-1).fit(X,y)
with open('rf_0.92.pkl', 'wb') as f:
    pickle.dump(rf, f)
```

4.4 Model Testing

Our model is tested by the test set. Our evaluation matrix include Root Mean Square Error (RMSE), unbiased Root Mean Square Error (ubRMSE), Pearson Correlation Coefficient (r) and Mean Difference (MD). The equations for the metrics can be seen below) (Zhang et al., 2021).

$$RMSE = \sqrt{\frac{\sum_i^N (y_{pred,i} - y_{ref,i})^2}{N}} \qquad \text{eq.3}$$

$$r = \frac{\sum (y_{pred} - \overline{y_{pred}})(y_{ref} - \overline{y_{ref}})}{\sqrt{\sum (y_{pred} - \overline{y_{pred}})^2} \sqrt{\sum (y_{ref} - \overline{y_{ref}})^2}} \qquad \text{eq.4}$$

$$MD = \frac{\sum_i^N (y_{pred,i} - y_{ref,i})}{N} \qquad \text{eq.5}$$

$$ubRMSE = \sqrt{RMSE^2 - MD^2} \qquad \text{eq.6}$$

The code is shown as follows:

```
import pickle
import pandas as pd
from scipy.stats import pearsonr
from sklearn.metrics import mean_squared_error
import numpy as np
res= open('rf_r_0.92.pkl','rb')
rf=pickle.load(res)
test=pd.read_csv('test_set_0510.csv')
feature_names =['GRD','GW','preci','bulk','clay','sand','LST_day','LST_night','EVI_modis','LAI_modis']
label = "SM"
X_test = test[feature_names]
y_test = test[label]
y_pre=rf.predict(X_test)
df=pd.DataFrame({'SM':y_test,'SM_pre':y_pre})

corr, p_value = pearsonr(df['SM'], df['SM_pre'])
print("Pearson correlation coefficient: ", corr)
print(f"RMSE": {np.sqrt(mean_squared_error(df['SM'], df['SM_pre']))}")
MD=np.mean(df['SM']-df['SM_pre'])
print(MD)
def ubRMSE(y_true, y_pred):
    n = len(y_true)
    mse = np.sum((y_pred - y_true) ** 2) / (n - 1)
    ubrmse = np.sqrt(mse)
    return ubrmse
ubRMSE=ubRMSE(df['SM'], df['SM_pre'])
print(ubRMSE)
```

4.5 10m Product Generation

After building the RF-based model, we apply the 10m resolution predictor variable to predict the SM for the Twente region in the Netherlands. Due to the large size of the dataset, we are unable to

upload 10m predictors variable dataset. Please download the 10m predictor variables from Google Earth Engine and follow the step of filling gap in previous sections . The code is shown as follows. The 10m SM product can be accessed via link: https://drive.google.com/file/d/1Hfi5jrF0f-pJ_-VRfjXezvAv_ujtpFBg/view?usp=drive_link
https://drive.google.com/file/d/1XuOudey7yDbUDYQpBr0AUkj7zcpSAteH/view?usp=drive_link

```
import pickle
from osgeo import gdal
import numpy as np
import pandas as pd
import datetime
from osgeo import gdal
import os
import time
res= open(r"D:\model\rf_r_0.92.pkl",'rb')
rf=pickle.load(res)
path="D:/predictors_2018/predictors_2018_236.tif"
#path="C:/Users/28756/OneDrive - University of Twente/Desktop/1km_dataset/predictors_10m/"+'predictors'+date+'.tif'
dataset = gdal.Open(path)
im_geotrans = dataset.GetGeoTransform()
im_width = dataset.RasterXSize
im_height = dataset.RasterYSize
band = dataset.RasterCount
im_lon = [im_geotrans[0] + i * im_geotrans[1] for i in range(im_width)]
im_lat = [im_geotrans[3] + i * im_geotrans[5] for i in range(im_height)]
xgrid, ygrid = np.meshgrid(im_lon, im_lat)
ravelx=xgrid.ravel()
ravely=ygrid.ravel()
date = datetime.date(2018, 1, 1)
projection = dataset.GetProjection()
geotransform = dataset.GetGeoTransform()
bands=[1,4,10]
def read_band(num):
    im_data = dataset.GetRasterBand(num).ReadAsArray(xoff=0, yoff=0, win_xsize=im_width, win_ysize=im_height)
    #name = dataset.GetRasterBand(num).GetDescription()
    #target_date = date + datetime.timedelta(days=num)
    #df=pd.DataFrame({str(name):im_data})
    return im_data
def replace_nan(variable_wn):
    variable=np.nan_to_num(variable_wn, nan=0).flatten()
    #name = dataset.GetRasterBand(num).GetDescription()
    #target_date = date + datetime.timedelta(days=num)
    #df=pd.DataFrame({str(name):im_data})
    return variable
# soildgrid with nan
bulk_wn=read_band(5)
clay_wn=read_band(6)
```

```
sand_wn=read_band(7)
# soildgrid nans replaced by 0
bulk=replace_nan(bulk_wn)
clay=replace_nan(clay_wn)
sand=replace_nan(sand_wn)
for i in range(30):
    #start_time = time.time()
    #GRD_GW_preci
    predictor_path='D:/predictors_2018/predictors_2018_'+str(i+1)+'.tif'
    dataset=gdal.Open(predictor_path)
    GRD_wn=read_band(1)
    GW_wn=read_band(4)
    preci_wn=read_band(10)
    GRD=replace_nan(GRD_wn)
    GW=replace_nan(GW_wn)
    preci=replace_nan(preci_wn)
    #evi
    evi_path='D:/EVI_interpolated/'+str(i+1)+'.tif'
    dataset = gdal.Open(evi_path)
    evi_wn=read_band(1)
    evi=replace_nan(evi_wn)
    #lai
    lai_path='D:/LAI_interpolated/'+str(i+1)+'.tif'
    dataset = gdal.Open(lai_path)
    lai_wn=read_band(1)
    lai=replace_nan(lai_wn)
    #lst_day
    LST_day_path="D:/LST_day_2018/"+str(i)+'.tif'
    dataset = gdal.Open(LST_day_path)
    LST_day_wn=read_band(1)*0.01
    LST_day=replace_nan(LST_day_wn)
    #LST_night
    LST_night_path="D:/LST_day_2018/"+str(i)+'.tif'
    dataset = gdal.Open(LST_night_path)
    LST_night_wn=read_band(1)*0.01
    LST_night=replace_nan(LST_night_wn)
    #predictors
    predictors_10m=np.vstack([GRD,GW,preci,bulk,clay,sand,LST_day,LST_night,evi,lai]).transpose()
    #predict
    yy=rf.predict(predictors_10m).reshape(4204, 8206)
    #mask
    mask_lai = np.isnan(lai_wn)
    mask_sand=np.isnan(sand_wn)
    yy[mask_lai] = np.nan
    yy[mask_sand] = np.nan
    filename = 'SM_10m_'+str(i+1)+'.tif'
    #wrtie in geotiff
    driver = gdal.GetDriverByName("GTiff")
    a = driver.Create(filename,8206 ,4204, 1, gdal.GDT_Float32)

    a.SetGeoTransform(geotransform)
    a.SetProjection(projection)

    band = a.GetRasterBand(1)
    band.WriteArray(yy)

    band.FlushCache()
    a.FlushCache()
    a = None
    print(filename)
```

REFERENCE

Abbaszadeh, P., Moradkhani, H., & Zhan, X. (2019). Downscaling SMAP Radiometer Soil Moisture Over the CONUS Using an Ensemble Learning Method. *Water Resources Research*, *55*(1), 324–344. https://doi.org/10.1029/2018WR023354

André, J.-C., Goutorbe, J.-P., & Perrier, A. (1986). HAPEX—MOBILHY: A Hydrologic Atmospheric Experiment the Study of Water Budget and Evaporation Flux at the Climatic Scale. *Bulletin of the American Meteorological Society*, *67*(2), 138–144. https://doi.org/10.1175/1520-0477-67.2.138

Benninga, H. J. F., Carranza, C. D. U., Pezij, M., Van Santen, P., Van Der Ploeg, M. J., Augustijn, D. C. M., & Van Der Velde, R. (2018). The Raam regional soil moisture monitoring network in the Netherlands. *Earth System Science Data*, *10*(1), 61–79. https://doi.org/10.5194/ESSD-10-61-2018

Cook, B. I., Mankin, J. S., & Anchukaitis, K. J. (2018). Climate Change and Drought: From Past to Future. *Current Climate Change Reports*, *4*(2), 164–179. https://doi.org/10.1007/s40641-018-0093-2

Das, N. N., Entekhabi, D., & Njoku, E. G. (2011). An algorithm for merging SMAP radiometer and radar data for high-resolution soil-moisture retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, *49*(5), 1504–1512. https://doi.org/10.1109/TGRS.2010.2089526

Dente, L., Su, Z., & Wen, J. (2012). Validation of SMOS Soil Moisture Products over the Maqu and Twente Regions. *Sensors 2012, Vol. 12, Pages 9965-9986*, *12*(8), 9965–9986. https://doi.org/10.3390/S120809965

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., Meygret, A., Spoto, F., Sy, O., Marchese, F., & Bargellini, P. (2012). Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sensing of Environment*, *120*, 25–36. https://doi.org/10.1016/J.RSE.2011.11.026

Hajj, M. El, Baghdadi, N., Zribi, M., & Bazzi, H. (2017). Synergic use of Sentinel-1 and Sentinel-2 images for operational soil moisture mapping at high spatial resolution over agricultural areas. *Remote Sensing*, *9*(12), 1–28. https://doi.org/10.3390/rs9121292

Han, Q., Zeng, Y., Zhang, L., Wang, C., Prikaziuk, E., Niu, Z., & Su, B. (2023). Global long term daily 1 km surface soil moisture dataset with physics informed machine learning. *Scientific Data 2023 10:1*, *10*(1), 1–12. https://doi.org/10.1038/s41597-023-02011-7

He, B., Xing, M., & Bai, X. (2014). A synergistic methodology for soil moisture estimation in an alpine prairie using radar and optical satellite data. *Remote Sensing*, *6*(11), 10966–10985. https://doi.org/10.3390/rs61110966

Idso, S., Jackson, R., Reginato, R., Kimball, B., & Nakayama, F. (1975). The Dependence of Bare Soil Albedo on Soil Water Content. *J Appl Meteorol*, *14*, 109–113.

Im, J., Park, S., Rhee, J., Baik, J., & Choi, M. (2016). Downscaling of AMSR-E soil moisture with MODIS products using machine learning approaches. *Environmental Earth Sciences*, *75*(15). https://doi.org/10.1007/S12665-016-5917-6

Liu, Y., Jing, W., Wang, Q., & Xia, X. (2020). Generating high-resolution daily soil moisture by using spatial downscaling techniques: a comparison of six machine learning algorithms. *Advances in Water Resources*, *141*, 103601. https://doi.org/10.1016/J.ADVWATRES.2020.103601

Lv, A., Zhang, Z., & Zhu, H. (2021). A neural-network based spatial resolution downscaling method for soil moisture: Case study of qinghai province. *Remote Sensing*, *13*(8). https://doi.org/10.3390/RS13081583

Ma, C., Li, X., & McCabe, M. F. (2020). Retrieval of high-resolution soil moisture through combination of Sentinel-1 and Sentinel-2 data. *Remote Sensing*, *12*(14), 1–28. https://doi.org/10.3390/rs12142303

Mattar, C., Wigneron, J. P., Sobrino, J. A., Novello, N., Calvet, J. C., Albergel, C., Richaume, P., Mialon, A., Guyon, D., Jimenez-Munoz, J. C., & Kerr, Y. (2012). A combined optical-microwave method to retrieve soil moisture over vegetated areas. *IEEE Transactions on Geoscience and Remote Sensing*, *50*(5 PART 1), 1404–1413. https://doi.org/10.1109/TGRS.2011.2179051

Notarnicola, C., Angiulli, M., & Posa, F. (2006). Use of Radar and Optical Remotely Sensed Data for Soil Moisture Retrieval Over Vegetated Areas. *Ieee Transactions on Geoscience and Remote Sensing*, *44*(4), 925–935.

Philip, S. Y., Kew, S. F., Van Der Wiel, K., Wanders, N., Jan Van Oldenborgh, G., & Philip, S. Y. (2020). Regional differentiation in climate change induced drought trends in the Netherlands. *Environmental Research Letters*, *15*(9). https://doi.org/10.1088/1748-9326/ab97ca

Potin, P. (2019). *Copernicus Sentinel-1 Constellation Mission Operations Status; Copernicus Sentinel-1 Constellation Mission Operations Status*. *1*.

Sachindra, D. A., & Kanae, S. (2019). Machine learning for downscaling: the use of parallel multiple populations in genetic programming. *Stochastic Environmental Research and Risk Assessment*, *33*(8–9), 1497–1533. https://doi.org/10.1007/S00477-019-01721-Y/FIGURES/13

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems*, *4*, 2951–2959.

Su, Z., Zeng, Y., Romano, N., Manfreda, S., Francés, F., Ben Dor, E., Szabó, B., Vico, G., Nasta, P., Zhuang, R., Francos, N., Mészáros, J., Dal Sasso, S. F., Bassiouni, M., Zhang, L., Rwasoka, D. T., Retsios, B., Yu, L., Blatchford, M. L., & Mannaerts, C. (2020). An Integrative Information Aqueduct to Close the Gaps between Satellite Observation of Water Cycle and Local Sustainable Management of Water Resources. *Water 2020, Vol. 12, Page 1495*, *12*(5), 1495. https://doi.org/10.3390/W12051495

Wei, Z., Meng, Y., Zhang, W., Peng, J., & Meng, L. (2019). Downscaling SMAP soil moisture estimation with gradient boosting decision tree regression over the Tibetan Plateau. *Remote Sensing of Environment*, *225*, 30–44. https://doi.org/10.1016/J.RSE.2019.02.022

Yan, R., & Bai, J. (2020). A New Approach for Soil Moisture Downscaling in the Presence of Seasonal Difference. *Remote Sensing 2020, Vol. 12, Page 2818*, *12*(17), 2818. https://doi.org/10.3390/RS12172818

Yang, Y., Zhang, J., Bao, Z., Ao, T., Wang, G., Wu, H., & Wang, J. (2021). Evaluation of multi-source soil moisture datasets over central and eastern agricultural area of China using in situ monitoring network. *Remote Sensing*, *13*(6), 1–26. https://doi.org/10.3390/rs13061175

Zeng, Y., Su, Z., Van Der Velde, R., Wang, L., Xu, K., Wang, X., & Wen, J. (2016). Blending satellite observed, model simulated, and in situ measured soil moisture over Tibetan Plateau. *Remote Sensing*, *8*(3), 1–22. https://doi.org/10.3390/rs8030268

Zhan, X., Houser, P. R., Walker, J. P., & Crow, W. T. (2006). A method for retrieving high-resolution surface soil moisture from hydros L-band radiometer and radar observations. *IEEE Transactions on Geoscience and Remote Sensing*, *44*(6), 1534–1544. https://doi.org/10.1109/TGRS.2005.863319

Zhang, L., Zeng, Y., Zhuang, R., Szabó, B., Manfreda, S., Han, Q., & Su, Z. (2021). In situ observation-constrained global surface soil moisture using random forest model. *Remote Sensing*, *13*(23). https://doi.org/10.3390/rs13234893

Zhao, Y., Weng, Z., Chen, H., & Yang, J. (2020). Analysis of the evolution of drought, flood, and drought-flood abrupt alternation events under climate change using the daily SWAP index. *Water (Switzerland)*, *12*(7). https://doi.org/10.3390/w12071969

Zhuang, R., Manfreda, S., Zeng, Y., Su, Z., Ben Dor, E., & Petropoulos, G. P. (2023). Soil moisture monitoring using unmanned aerial system. *Unmanned Aerial Systems for Monitoring Soil, Vegetation, and Riverine Environments*, 179–200. https://doi.org/10.1016/B978-0-323-85283-8.00003-5

Zhuang, R., Zeng, Y., Manfreda, S., & Su, Z. (2020). Quantifying long-term land surface and root zone soil moisture over Tibetan plateau. *Remote Sensing*, *12*(3), 1–20. https://doi.org/10.3390/rs12030509