



GEWEX 4th Workshop on Central Asia

Hands-On with TPDC: Platform Workflows and Water Body Change Monitoring Tutorial

By Xiaoduo Pan

National Tibetan Plateau / Third Pole
Environment Data Center Institute of
Tibetan Plateau Research
Chinese Academy of Sciences

**May 12 - 14, 2026 | OSH,
KYRGYZ REPUBLIC**

National Tibetan Plateau / Third Pole Environment Data Center System Training

National Tibetan Plateau / Third Pole Environment Data Center:

<https://data.tpdc.ac.cn/home>



Figure 1: System Login Page

1. Registration / Login

1) Click Register to enter the user registration interface (Figure 2). All fields marked with an asterisk (*) are required. Click Register to submit. After submission, open your registration email inbox — you will receive an activation email. Click the activation link to activate your account.

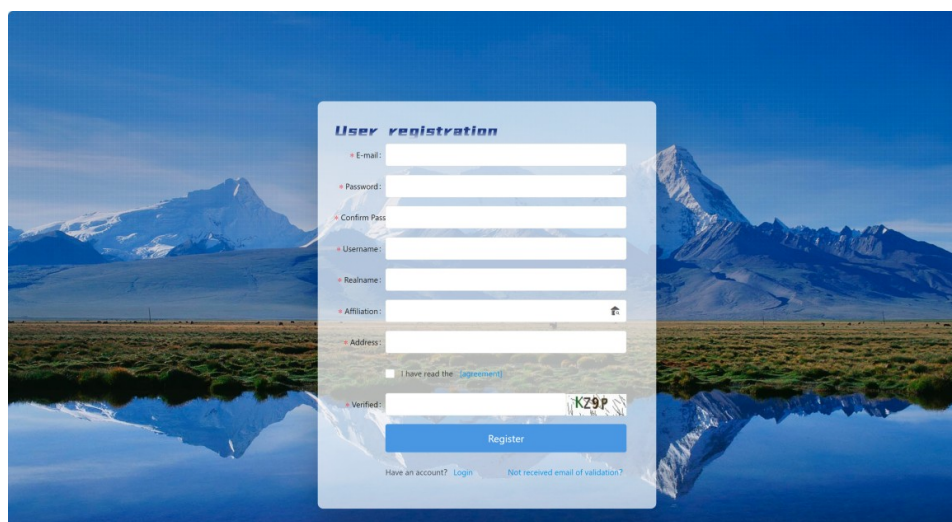


Figure 2: System Registration Page

2) Once registration is complete, click the Login button on the homepage, enter your username and password, and enter the data center.

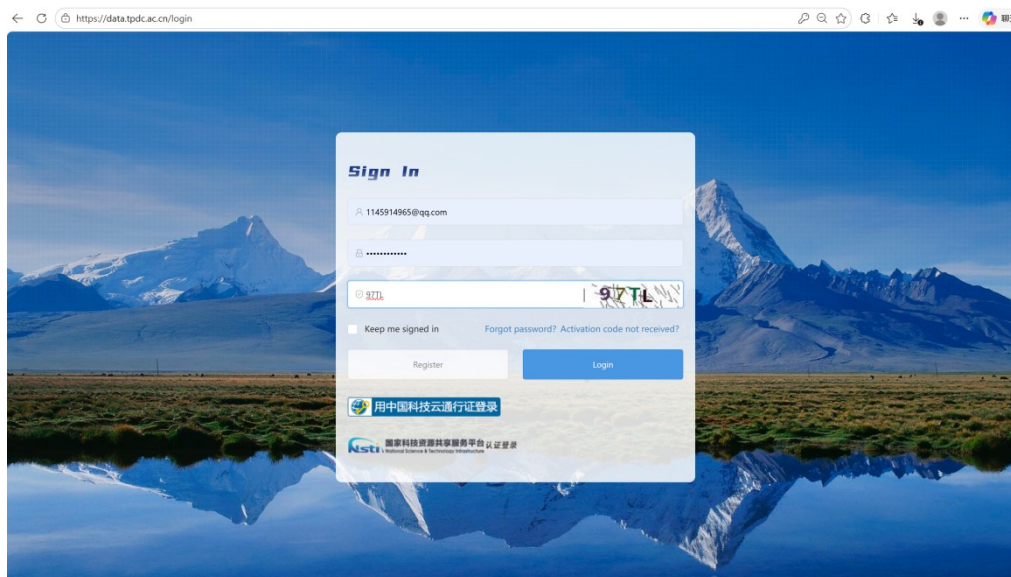


Figure 3: System Login Interface

2. Data Search

1) A persistent search bar is located at the top of every page (Figure 4). Users can use this search bar at any time to filter data. The website homepage also features a prominently placed search tool that supports fuzzy matching across titles, keywords, authors, and description content.

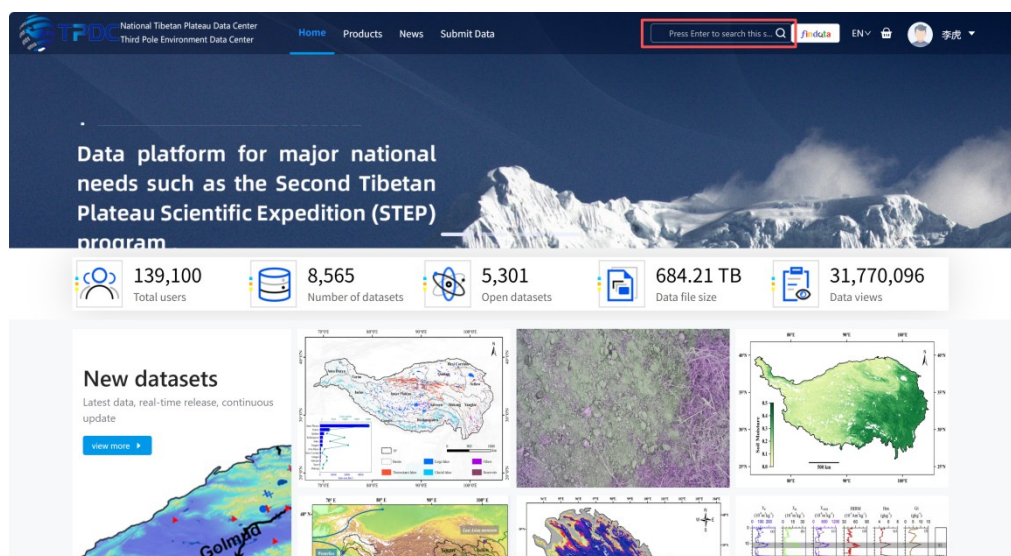


Figure 4: Persistent Search Bar at the Top of the Page

2) In the Data Products section, multiple data browsing and classification methods are provided based on different logical categorization approaches, including:

Themes, Authors, Spatial Search, Key Datasets, and Latest Datasets (Figure 5).

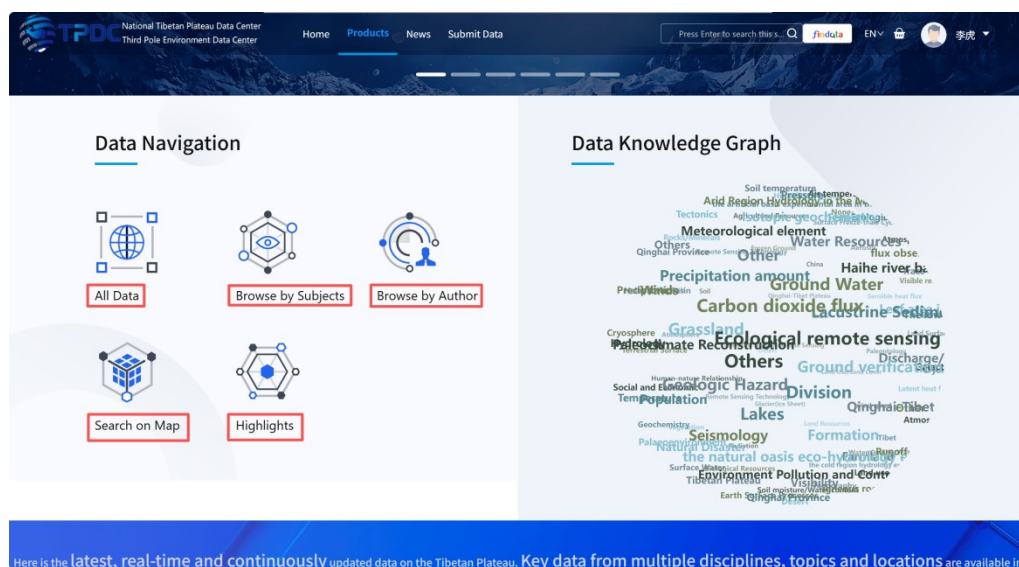


Figure 5: Multiple Logical Data Classification Browsing Methods

3) On the search results page, users can also perform a secondary filtered search on data resources using discipline keywords, subject keywords, and location keywords (Figure 6).

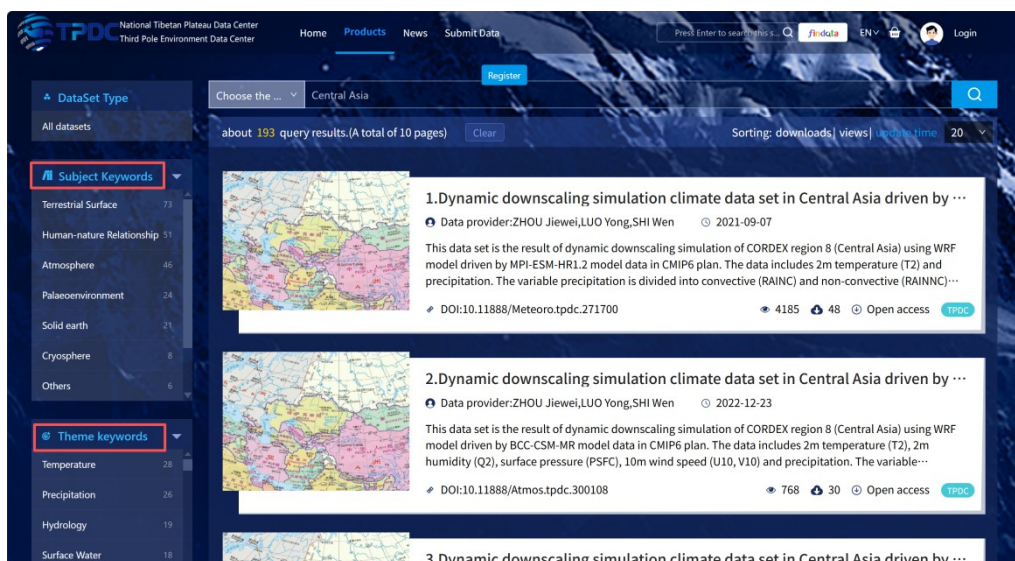


Figure 6: Secondary Filter Search

3. Data Access / Download

1) It is recommended to install FlashFXP

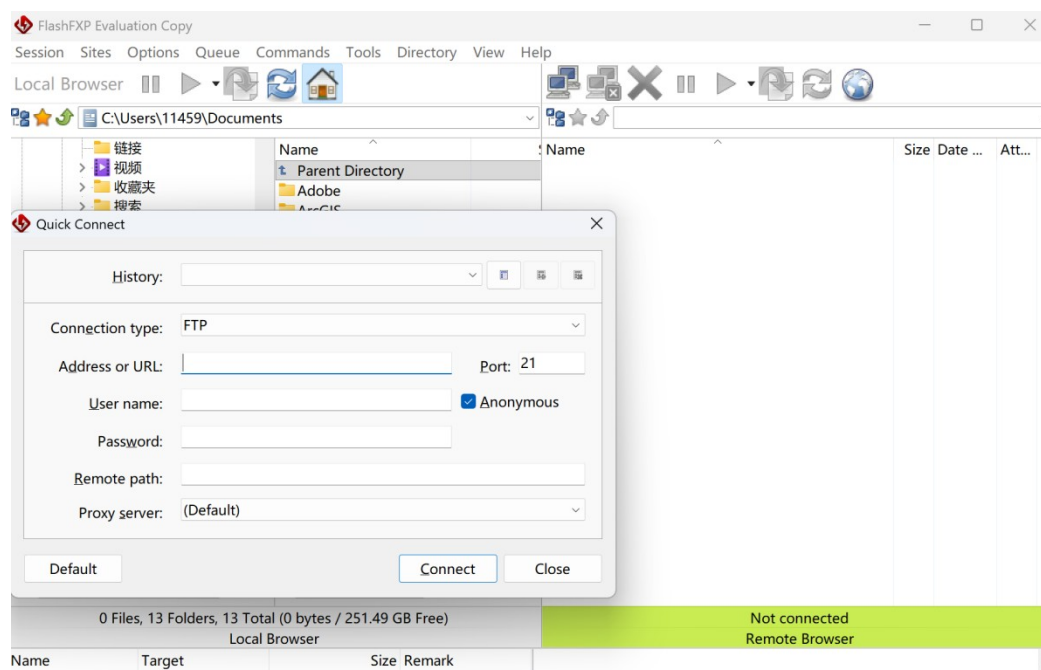


Figure 7: FlashFXP Program

2) Open Data Access: After locating the desired data through search, click on the metadata title to enter the metadata details page. Here you can view a full introduction to the dataset (including data abstract, data citation, article citation, keywords, etc.). Click the Download button — an FTP account will pop up.

If no login is required for download: An FTP account will pop up directly. Use the provided server address, username, and password to log in with an FTP client tool (such as FlashFXP) and download the dataset.

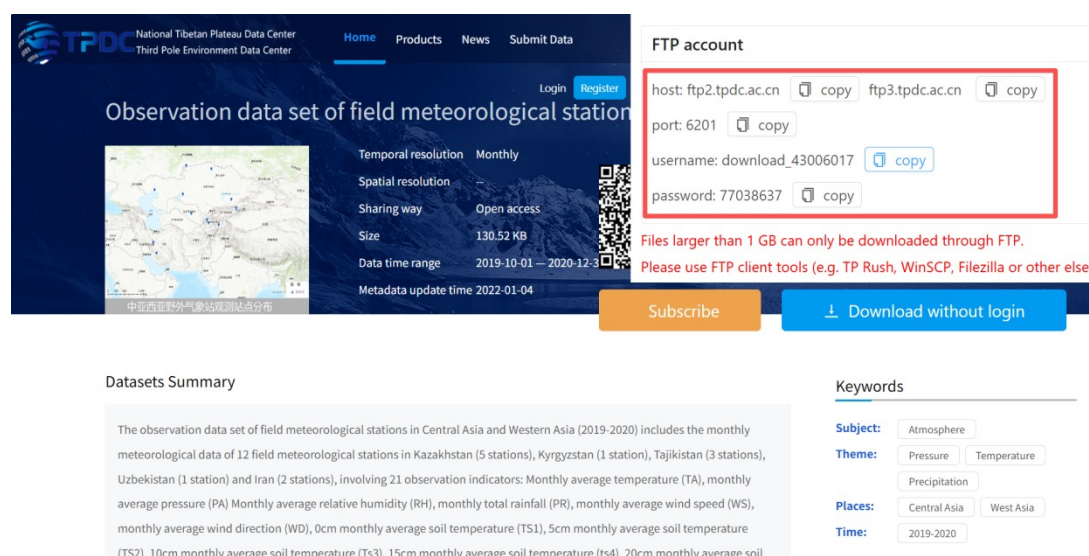


Figure 8: FTP Account Information Pop-up After Clicking Download for Open-Access Data

Enter the FTP account information into FlashFXP to log in to the server. Select

the desired file, right-click, and choose Download.

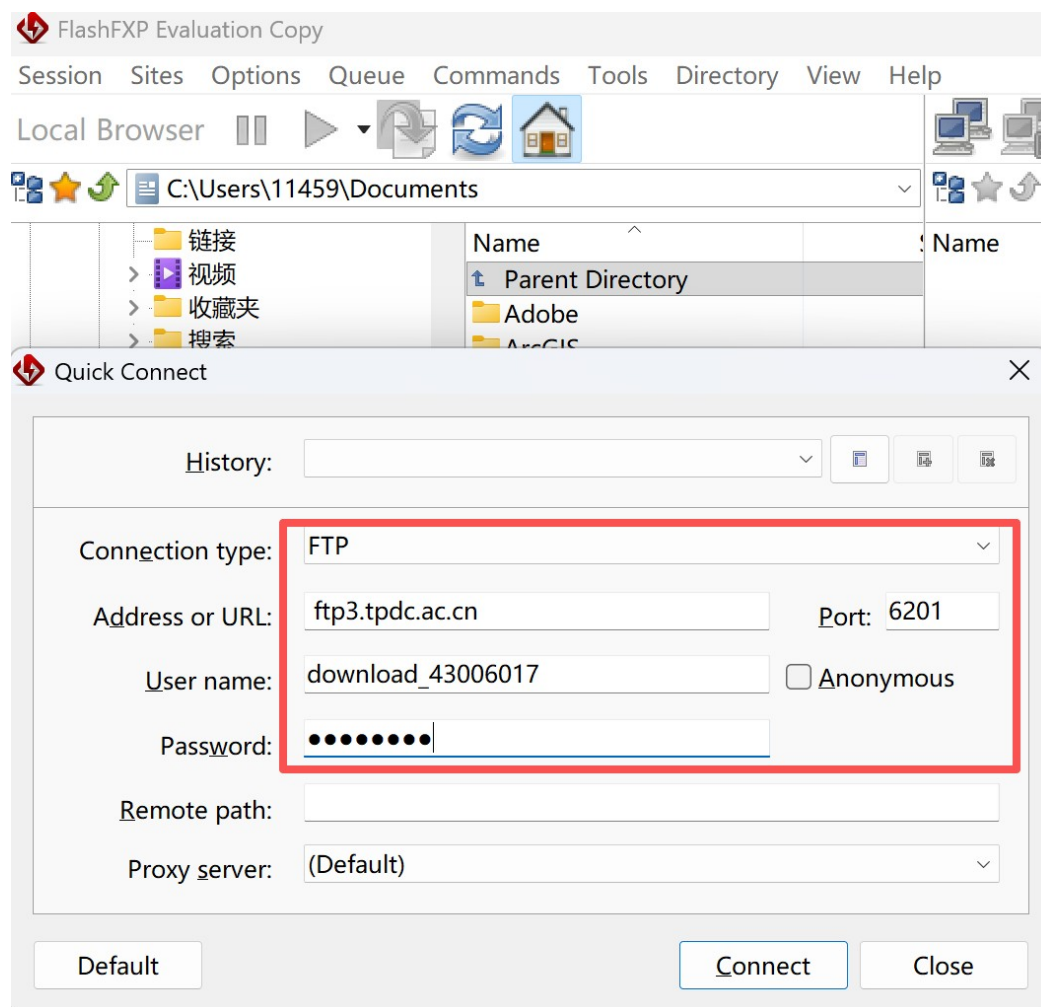


Figure 9: Entering the FTP Account Information into FlashFXP

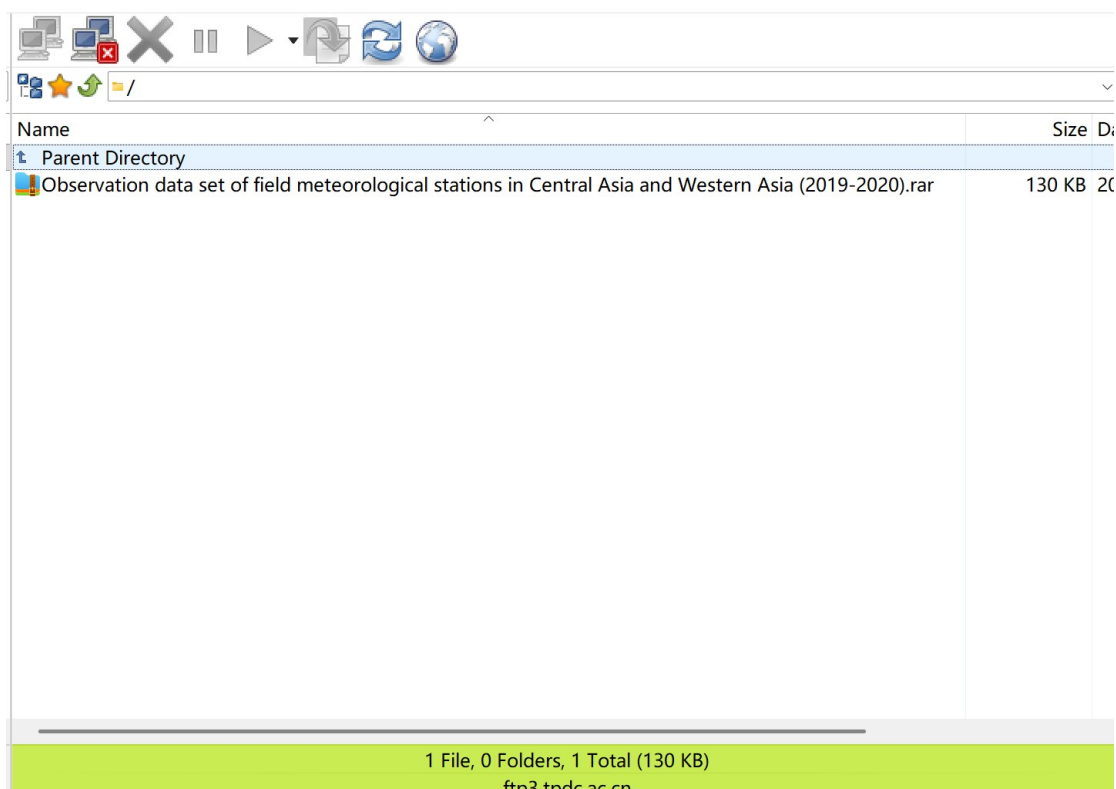


Figure 10: Successfully Logged into the Server for Data Download in FlashFXP

3) Apply for access: Click the Download button — a login interface will pop up. After logging in, locate the desired dataset again and click the Download button. An Apply for Download button will appear on the page. Click Apply for Download, enter your reason for the application in the text input box that appears, and click Submit.

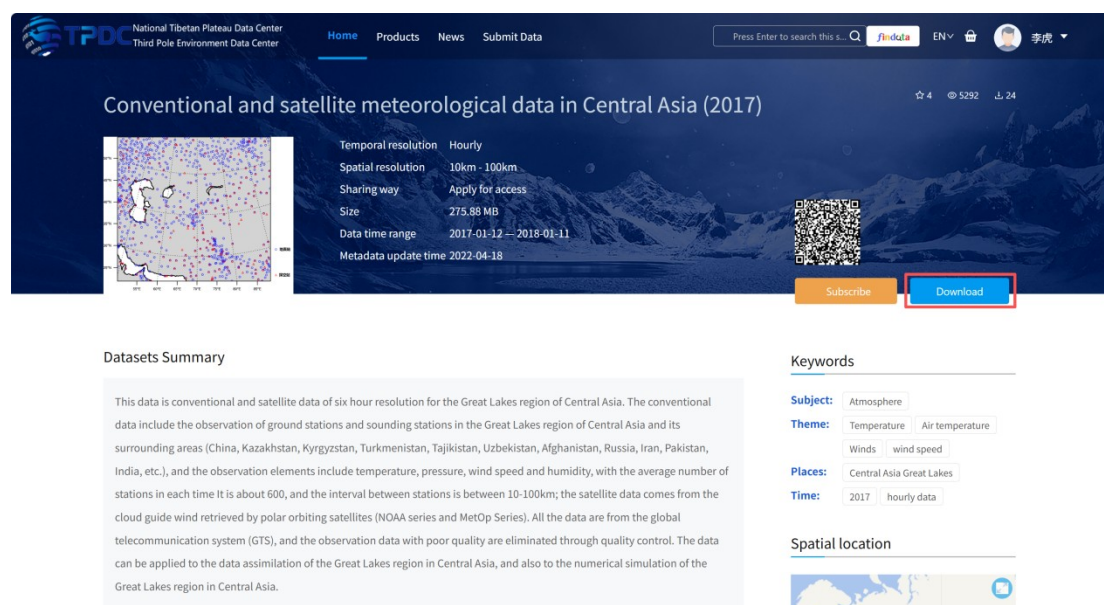




Figure 11: The "Download" Button Changes to "Apply for Download" for Application-Required Data

Data added to the data basket, you can only fill in the data purpose once, batch submit download application, fill in the detailed and accurate data purpose can obtain better data services

* Reason

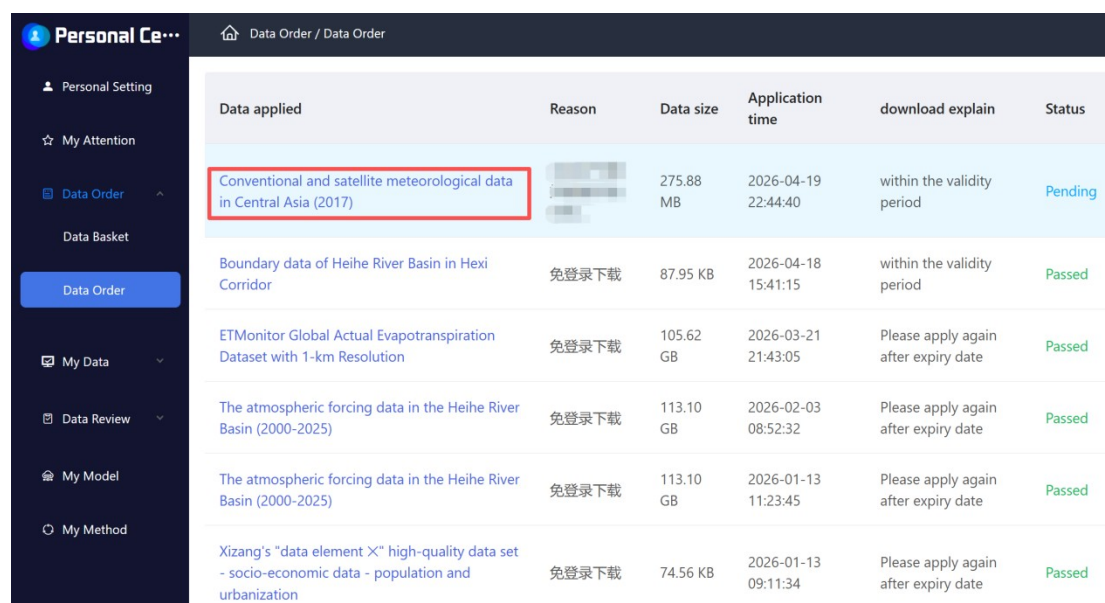
Please enter your data usage

Note:
 Please fill in your real name, company email and the purpose detail of use.
 If there is project support, fill in the details, such as: National Science and Technology major project [2016YFA0600303-3].
 For master and doctoral papers, please indicate the name of the paper and the name of the supervisor in the purpose.

ⓘ Please read it before using the data [Data reference help](#)

Add to data basket Submit

Figure 12: Reason Input Box for Application-Based Data Download



Data applied	Reason	Data size	Application time	download explain	Status
Conventional and satellite meteorological data in Central Asia (2017)		275.88 MB	2026-04-19 22:44:40	within the validity period	Pending
Boundary data of Heihe River Basin in Hexi Corridor	免登录下载	87.95 KB	2026-04-18 15:41:15	within the validity period	Passed
ETMonitor Global Actual Evapotranspiration Dataset with 1-km Resolution	免登录下载	105.62 GB	2026-03-21 21:43:05	Please apply again after expiry date	Passed
The atmospheric forcing data in the Heihe River Basin (2000-2025)	免登录下载	113.10 GB	2026-02-03 08:52:32	Please apply again after expiry date	Passed
The atmospheric forcing data in the Heihe River Basin (2000-2025)	免登录下载	113.10 GB	2026-01-13 11:23:45	Please apply again after expiry date	Passed
Xizang's "data element X" high-quality data set - socio-economic data - population and urbanization	免登录下载	74.56 KB	2026-01-13 09:11:34	Please apply again after expiry date	Passed

Figure 13: Order Status Can Be Viewed in the Personal Center After Submission

4. Data Submission

The data submission module supports various types of data submissions, including:

- Routine data submission
- Data repository for manuscripts of unpublished papers
- Project submission portal

Click the corresponding submission entry point — for example, Routine Data Submission — to enter the Create New Metadata interface.

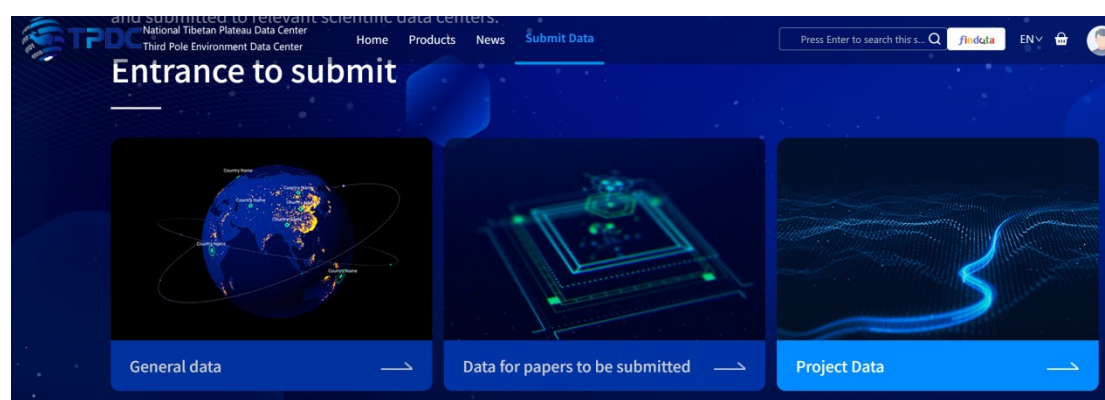


Figure 14: Submission Entry Interface

The new metadata creation interface includes the following fields: Basic Information, Resource Provider, Data Citation Literature, Keywords, Thumbnail, Data Files, Documentation / Description

Fields marked with a red asterisk (*) are required. Metadata cannot be submitted if any required fields are left incomplete.

Figure 15: New Metadata Creation Interface

In the Data Files section, an FTP account for uploading data will be provided. Following the instructions on the right side of the page, download the FTP client tool, log in using the provided FTP address, username, and password, and upload your data files. After the upload is complete, you must click Confirm Upload Complete.

Figure 16: Data Upload Interface

Once all required fields (marked in red) have been filled in, click the Submit button in the upper-right corner to submit the metadata.

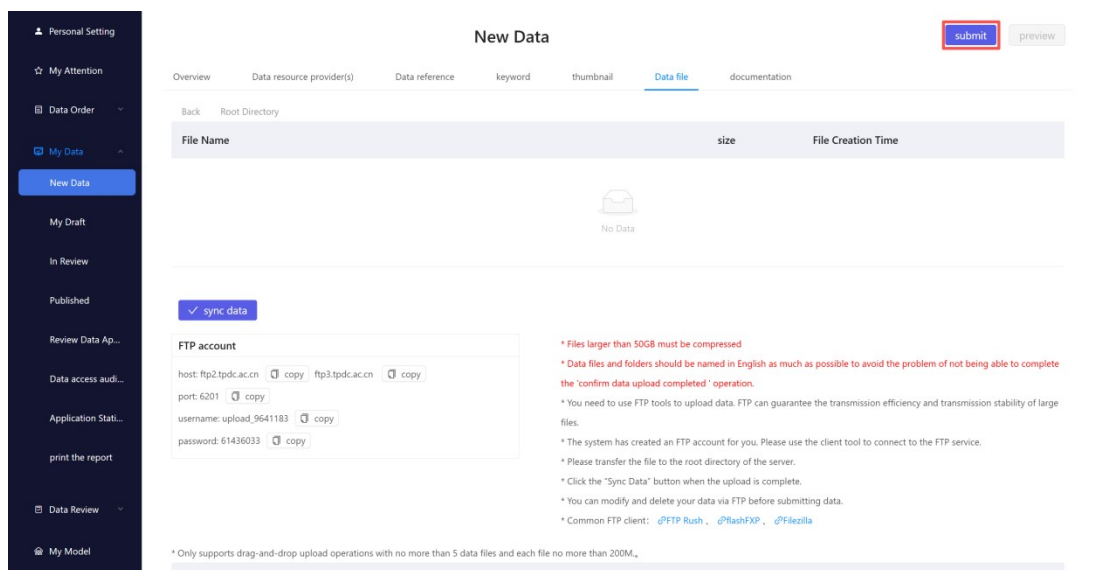


Figure 17: Submitting Metadata

If any required fields have not been completed at the time of submission, a prompt window will pop up to notify the user.

Checking		
Check item	Requirement	Requirement
Title of Submitted Paper	Required	✓
Name of journal to be submitted	Required	✓
Title	Required	✗
English title	Required	✓
Abstract	At least 200 words	✗
English abstract	Required	✓
Data file naming and usage	Required	✗
Data file naming and usage (English)	Required	✓
Time limit	Required	✗
Platform	Required	✓
Sharing policy	Required	✗
DOI type	Required	✗
DOI	If the DOI type is "This data follows the existing DOI of the data", DOI is required	✓
Spatial extent	Required	✗
Time resolution	Required	✗
Spatial resolution	Required	✗
Data resource provider	Required	✗
Data resource provider (translation)	At least one	✗
Keywords	All types of keywords are required	✗
Thumbnail	Required	✗

Figure 18: Metadata System Validation

After the metadata is submitted, go to My Drafts, and click Submit for Publication — the dataset submission is then complete.

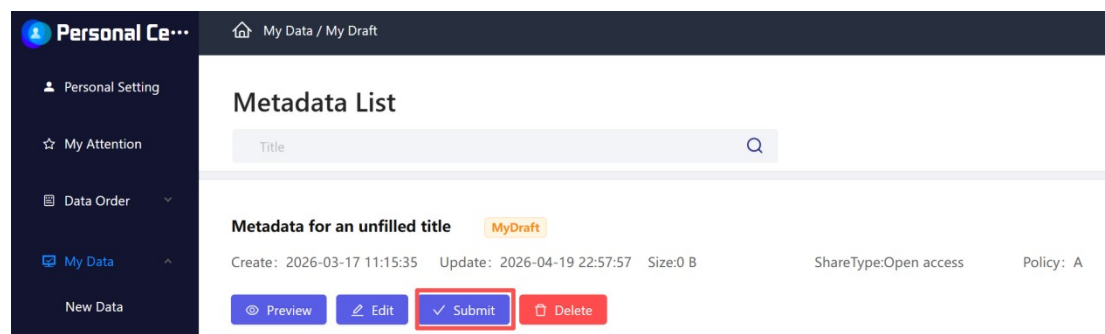


Figure 19: Submitting a Dataset for Publication in the Personal Center

Once the dataset is successfully submitted, it enters the review process. You can view the submitted metadata under Personal Center - Under Review. Notification emails will also be sent to the user's registered email address for updates on data submission, review, and publication status.

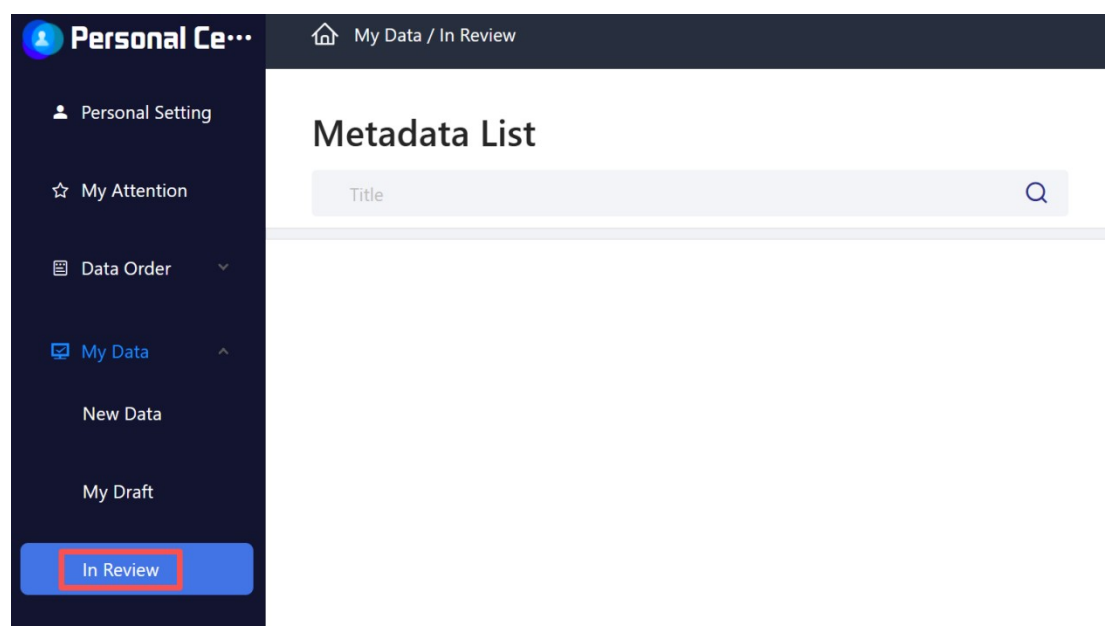


Figure 20: Viewing Submitted Data Status in the Personal Center

The Personal Center - My Data module includes the following functions: Create New Data, My Drafts, Under Review, Published, Data Access Statistics, Data-Related Application Statistics.

Users can check the status and sharing situation of their datasets at any time.

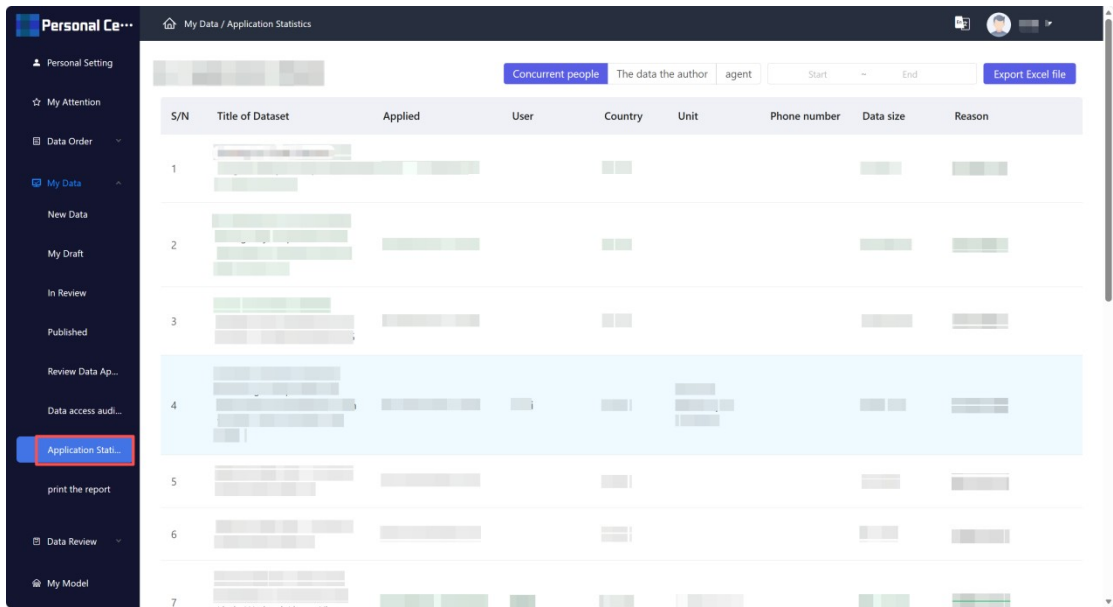


Figure 21: Viewing Submitted Data Sharing Status in the Personal Center

Water Body Change Monitoring Experiment Based on Machine Learning

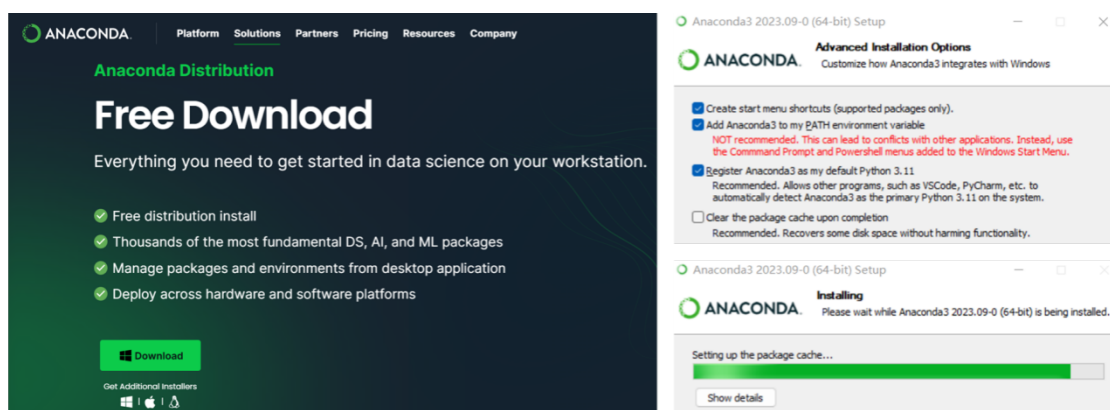
1. Software Installation and Data Acquisition

1.1 Anaconda Installation

Download and install Anaconda (version 2023.09-0, 64-bit) from the official website: <https://www.anaconda.com/download>

During installation, it is recommended to:

- Create Start Menu shortcuts (for supported packages only)
- Register Anaconda3 as the default Python 3.11
- Clear the package cache upon completion



<https://www.anaconda.com/download>

Figure 1: Anaconda Installation Setup Interface

1.2 Environment Configuration

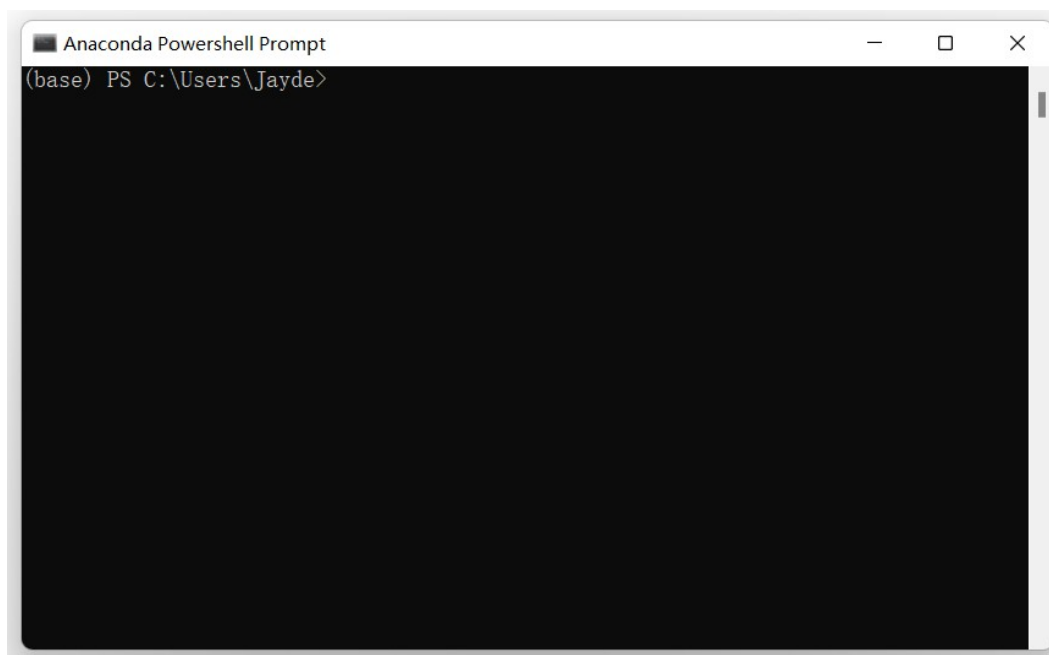


Figure 2: Environment Configuration Terminal Interface

- Create a virtual environment:

```
conda create -n test python=3.11
```

```
conda activate test
```

- Install required third-party libraries:

```
cd C:\Users\Jayde\Desktop\Experiment_3\software
```

```
pip install pyproj
```

```
pip install Shapely
```

```
pip install GDAL-3.4.3-cp311-cp311-win_amd64.whl
```

```
pip install Fiona-1.8.21-cp311-cp311-win_amd64.whl
```

```
pip install geopandas
```

```
pip install rasterio-1.2.10-cp311-cp311-win_amd64.whl
```

```
pip install scikit-learn
```

```
pip install scipy
```

```
pip install tqdm
```

```
pip install matplotlib
```

- Add the environment to Jupyter Notebook:

```
conda install ipykernel
```

```
python -m ipykernel install --user --name test --display-name test
```

1.3 QGIS Installation

Download QGIS 4 .0(or QGIS 3.44LTR for the most stable version) from:

- <https://www.qgis.org/en/site/forusers/download.html>

It is recommended to use the OSGeo4W Network Installer, which allows multiple QGIS versions to coexist and supports individual component updates without downloading the full package.

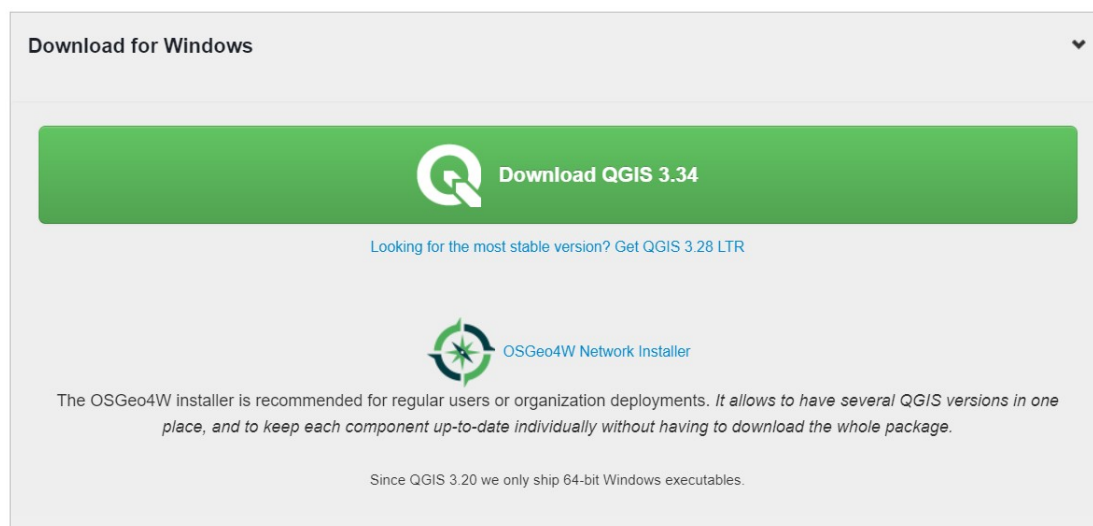


Figure 3: QGIS Download Page

(Note: QGIS can be downloaded and installed independently at a later time.)

1.4 Image Download

Download Landsat imagery via Google Earth Engine using the following links:

Landsat 7:

- <https://code.earthengine.google.com/b23caacca6baaled43a42f87ff576dba>

Landsat 8:

- <https://code.earthengine.google.com/78ba511f7561648d460a659288f0599a>



Figure 4: Satellite Image Map of the Study Area

2. Water Body Change Monitoring Model Construction

2.1 Feature Selection

The following spectral indices are selected as model input features:

- NDWI — Normalized Difference Water Index
- NDSI — Normalized Difference Snow Index
- NDVI — Normalized Difference Vegetation Index

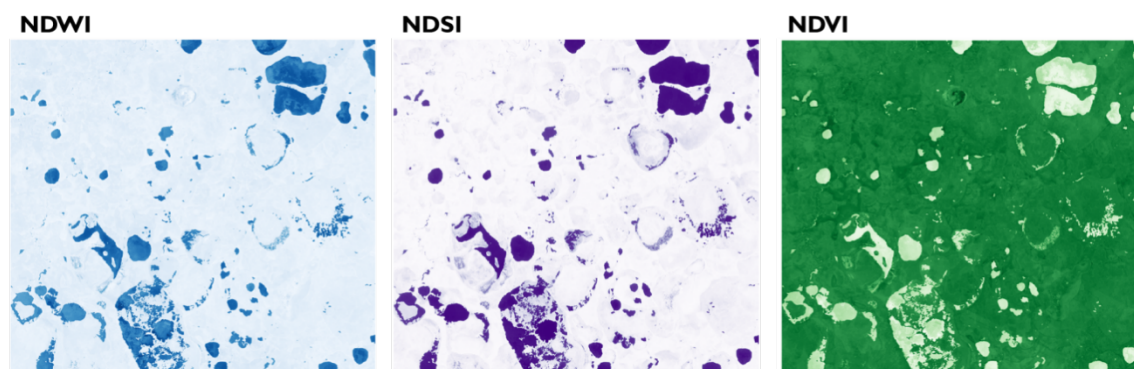


Figure 5: Feature Selection — NDWI, NDSI, NDVI

2.2 Creating Sample Points (Part I)

Step 1: Create a Point Layer in QGIS

Navigate to: Layer → Create Layer → New Shapefile Layer

Step 2: Parameter Settings

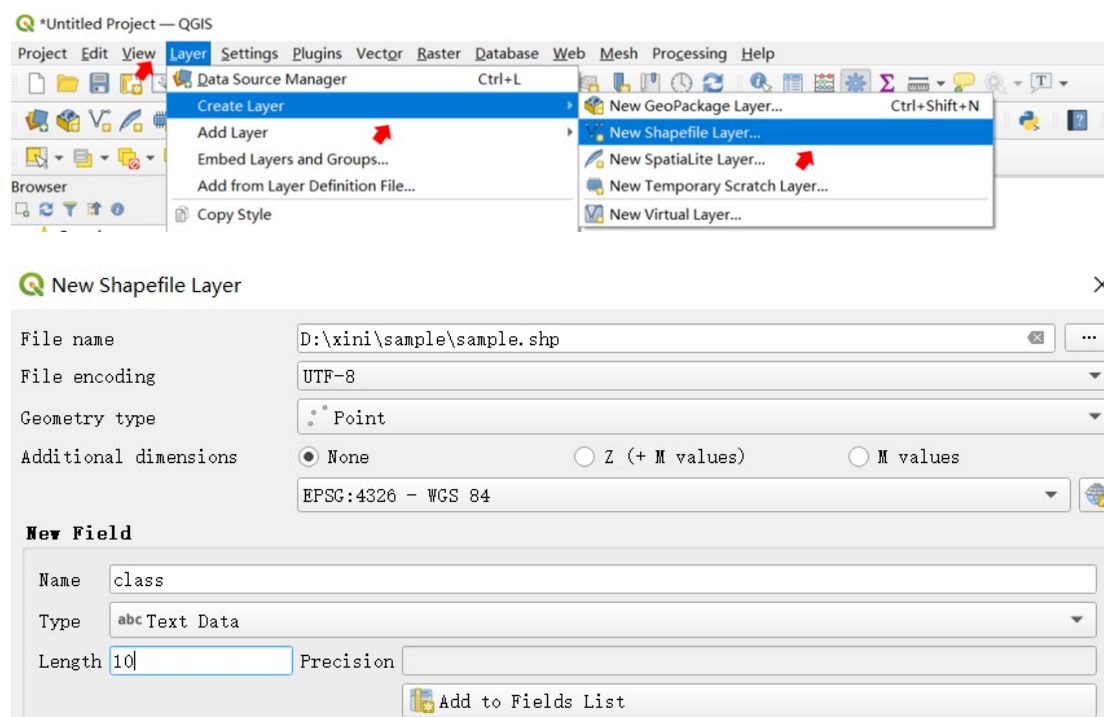
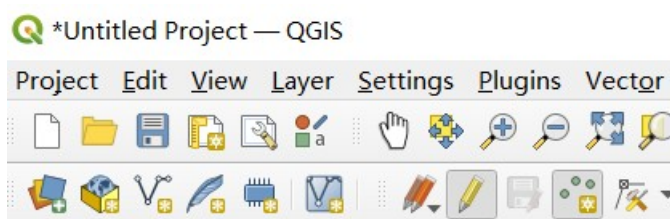


Figure 6: New Shapefile Layer Parameter Settings in QGIS

2.3 Creating Sample Points (Part II)

Step 3: Select Sample Points



Manually select point locations on the map in QGIS.

Step 4: Add Attribute Values

Assign class labels to each sample point:

1 = Water body

2 = Non-water body

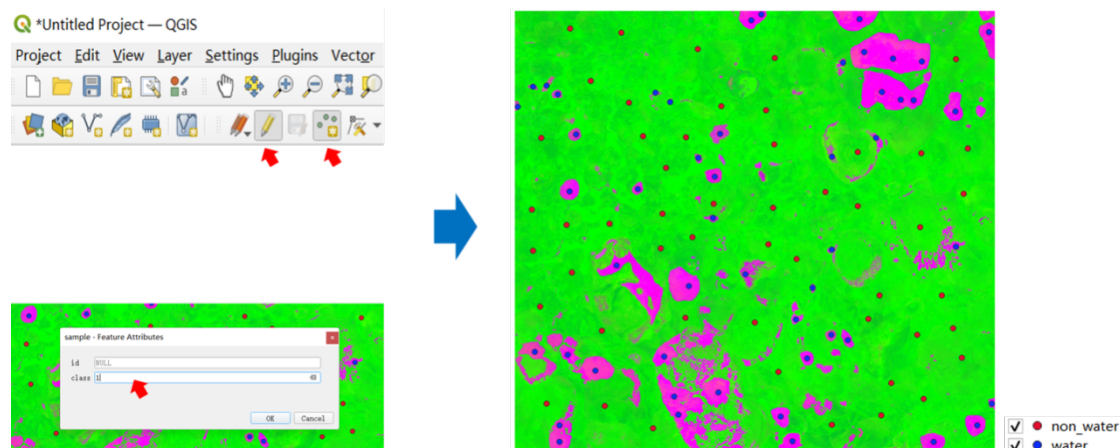


Figure 7: Sample Point Feature Attribute Assignment Interface (water / non_water)

2.4 Jupyter Notebook

Launch Jupyter Notebook using one of the following methods:

Navigate to the file directory, type cmd in the address bar, then enter jupyter notebook in the terminal.

Common Jupyter Notebook Keyboard Shortcuts:

- Shift + Enter Run current cell and move to the next
- Ctrl + Enter Run current cell and stay
- dd Delete current cell
- b Add a new empty cell below
- a Add a new empty cell above

- m Convert cell to Markdown
- y Convert Markdown cell to code cell



Figure 8: Jupyter Notebook File Browser Interface

2.5 Creating the Training Dataset

The `getfeat()` function extracts feature values (NDSI, NDVI, NDWI) at each sample point location from the raster image and saves the results as a CSV file.

```
def getfeat(points, feature, pathout):
    import rasterio
    from rasterio.features import geometry_mask
    import fiona
    from shapely.geometry import shape, Point
    import pandas as pd

    src = rasterio.open(feature)
    narray = src.read()

    feature_list = []
    with fiona.open(points, 'r') as vector_file:
        for feature in vector_file:
            geom = shape(feature['geometry'])
            point = Point(geom.x, geom.y)

            row, col = src.index(point.x, point.y)
            value = narray[:, row, col]

            newRow = {'class':feature['properties']['class'], 'lon':geom.x, 'lat':geom.y, 'NDSI':value[0], 'NDVI':value[1], 'NDWI':value[2]}
            feature_list.append(newRow)

    df = pd.DataFrame(feature_list)
    df.to_csv(pathout + '/' + 'feature_L8_2022.csv', index=False)

    #print(df)

    return
```

	class	lon	lat	NDSI	NDVI	NDWI
0	1	1.755235e+07	1.112071e+07	0.885396	-0.260029	0.309333
1	1	1.755235e+07	1.112378e+07	0.872578	-0.339355	0.396217
2	1	1.757395e+07	1.112923e+07	0.900967	-0.312559	0.396958
3	1	1.757474e+07	1.112580e+07	0.878754	-0.343003	0.438723
4	1	1.758238e+07	1.113010e+07	0.975571	-0.380899	0.473052
..
97	0	1.754869e+07	1.114629e+07	-0.541363	0.475888	-0.572185
98	0	1.757608e+07	1.114418e+07	-0.465070	0.459599	-0.566323
99	0	1.757740e+07	1.115945e+07	-0.579100	0.524475	-0.591679
100	0	1.760259e+07	1.116797e+07	-0.499173	0.480894	-0.545999
101	0	1.757239e+07	1.117253e+07	-0.548221	0.424919	-0.518841

Figure 9: Training Dataset Creation — Code and Output Table

2.6 Model Training

The `rf_classify()` function trains a Random Forest classifier using the extracted feature dataset, evaluates model accuracy, and saves the trained model as a `.pkl` file.

```
def rf_classify(features, pathout):
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import accuracy_score
    import rasterio
    import pandas as pd
    import joblib

    dataset = pd.read_csv(features)
    dataset = dataset.dropna()

    features = dataset.iloc[:, 3:]
    labels = dataset.iloc[:, 0]

    x_train, x_test, y_train, y_test = train_test_split(features, labels, test_size=0.3, random_state=42)

    rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

    rf_classifier.fit(x_train.values, y_train.values)

    predictions = rf_classifier.predict(x_test.values)

    accuracy = accuracy_score(y_test.values, predictions)
    print("Accuracy:", accuracy)

    joblib.dump(rf_classifier, pathout + '/' + 'L8_2022.pkl')

    return
```

Figure 10: Random Forest Model Training Code and Output Model File (L8_2022.pkl)

2.7 Image Classification

The `read_fea_image()` function reads the input raster image pixel by pixel and extracts feature values. The `predict()` function then applies the trained model to classify each pixel as water or non-water.

```
def read_fea_image(img_path):
    import rasterio
    import numpy as np
    import tqdm
    import os
    import joblib
    from sklearn.preprocessing import MinMaxScaler
    from osgeo import gdal

    img_data = rasterio.open(img_path).read()
    band_num, height_num, width_num = np.shape(img_data)

    img_data_list, row_col = [], []
    for i in tqdm.trange(height_num):
        for j in range(width_num):
            temp = img_data[:, i, j]
            if np.any(np.isnan(temp)):
                pass
            else:
                img_data_list.append(temp.tolist())
                row_col.append((i, j))

    fea_dataset = np.array(img_data_list)
    rows_cols = np.array(row_col)

    return fea_dataset, rows_cols

def predict(feature, pkl):
    import joblib

    moedl = joblib.load(pkl)
    pre_value = moedl.predict(feature)

    return pre_value
```

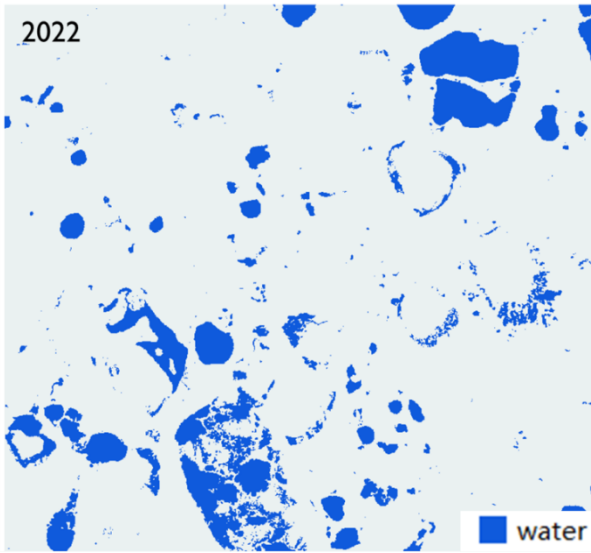


Figure 11: Image Classification Code and 2022 Water Body Classification Result Map

3. Water Body Area Change Analysis

3.1 Water Body Area Statistics

The `stat()` function counts the number of pixels classified as water (value = 1) in each classified image and converts the pixel count to area (km²) based on a 30 m × 30 m pixel resolution.

<pre>def stat(data): from osgeo import gdal import numpy as np ds = gdal.Open(data) band = ds.GetRasterBand(1) array = band.ReadAsArray() del ds count = np.ma.masked_where(array!=1, array).count() area = count*30*30/1000000 return area def main(): import pandas as pd pathin = r'results\predict\all' pathout = r'results\stat' datas = listdatas(pathin) feature_list = [] for data in datas: year = int(data.split('/')[0].split('.')[0].split('_')[-1]) area = stat(data) newRow = {'year':year, 'area':area} feature_list.append(newRow) df = pd.DataFrame(feature_list) df = df.sort_values(by=['year']) df.to_csv(pathout + '/' + 'area_stat.csv', index=False)</pre>	<table border="1"> <thead> <tr> <th></th> <th>year</th> <th>area</th> </tr> </thead> <tbody> <tr><td>0</td><td>2000</td><td>751.9707</td></tr> <tr><td>1</td><td>2001</td><td>835.4691</td></tr> <tr><td>2</td><td>2002</td><td>744.4233</td></tr> <tr><td>3</td><td>2003</td><td>778.1913</td></tr> <tr><td>4</td><td>2004</td><td>863.5896</td></tr> <tr><td>5</td><td>2005</td><td>824.9652</td></tr> <tr><td>6</td><td>2006</td><td>851.4369</td></tr> <tr><td>7</td><td>2007</td><td>768.9402</td></tr> <tr><td>8</td><td>2008</td><td>782.5617</td></tr> <tr><td>9</td><td>2009</td><td>822.5928</td></tr> <tr><td>10</td><td>2010</td><td>808.0740</td></tr> <tr><td>11</td><td>2011</td><td>972.1836</td></tr> <tr><td>12</td><td>2012</td><td>848.2608</td></tr> <tr><td>13</td><td>2013</td><td>769.9806</td></tr> <tr><td>14</td><td>2014</td><td>814.0887</td></tr> <tr><td>15</td><td>2015</td><td>898.2675</td></tr> <tr><td>16</td><td>2016</td><td>848.2356</td></tr> <tr><td>17</td><td>2017</td><td>930.5487</td></tr> <tr><td>18</td><td>2018</td><td>884.9322</td></tr> <tr><td>19</td><td>2019</td><td>969.8400</td></tr> <tr><td>20</td><td>2020</td><td>887.1039</td></tr> <tr><td>21</td><td>2021</td><td>869.3370</td></tr> <tr><td>22</td><td>2022</td><td>866.0025</td></tr> </tbody> </table>		year	area	0	2000	751.9707	1	2001	835.4691	2	2002	744.4233	3	2003	778.1913	4	2004	863.5896	5	2005	824.9652	6	2006	851.4369	7	2007	768.9402	8	2008	782.5617	9	2009	822.5928	10	2010	808.0740	11	2011	972.1836	12	2012	848.2608	13	2013	769.9806	14	2014	814.0887	15	2015	898.2675	16	2016	848.2356	17	2017	930.5487	18	2018	884.9322	19	2019	969.8400	20	2020	887.1039	21	2021	869.3370	22	2022	866.0025
	year	area																																																																							
0	2000	751.9707																																																																							
1	2001	835.4691																																																																							
2	2002	744.4233																																																																							
3	2003	778.1913																																																																							
4	2004	863.5896																																																																							
5	2005	824.9652																																																																							
6	2006	851.4369																																																																							
7	2007	768.9402																																																																							
8	2008	782.5617																																																																							
9	2009	822.5928																																																																							
10	2010	808.0740																																																																							
11	2011	972.1836																																																																							
12	2012	848.2608																																																																							
13	2013	769.9806																																																																							
14	2014	814.0887																																																																							
15	2015	898.2675																																																																							
16	2016	848.2356																																																																							
17	2017	930.5487																																																																							
18	2018	884.9322																																																																							
19	2019	969.8400																																																																							
20	2020	887.1039																																																																							
21	2021	869.3370																																																																							
22	2022	866.0025																																																																							

Figure 12: Water Body Area Statistics Code and Output Table

3.2 Water Body Area Change Trend Plot

The `draw_1()` function plots the inter-annual variation of water body area along with a linear trend line, and annotates the slope, p-value, and R² value.

```

def draw_1(data, pathout):
    import pandas as pd
    from scipy.interpolate import make_interp_spline
    import scipy.stats as stats
    from scipy.optimize import curve_fit

    df = pd.read_csv(data, header=0)
    #print(df)

    fig, ax = plt.subplots()

    ax.plot(df['year'], df['area'], color='blue', marker='o', label="Area")
    ax.set_xlabel('Year')
    ax.set_ylabel('Area (km2)', color='k')

    # slope, intercept, r, p_value, std_err = stats.linregress(df['year'], df['area'])
    # ax.plot(df['year'], func(df['year'], slope, intercept), color='r', linestyle='-', linewidth=1, label="Liner trend")

    # plt.text(2001, 920, 'Slope = {}, p = {}'.format(round(slope, 2), round(p_value, 3)))
    # plt.text(2001, 905, '$R^2$ = {}'.format(round(r*r, 2)))

    ax.tick_params(axis='y', labelcolor='k')
    ax.legend(ncol=1, frameon=False, loc=2, borderaxespad=0)
    plt.show()

    fig.savefig(pathout + '/' + 'fig_area.png', dpi=700, bbox_inches='tight', transparent=True)
    return

```

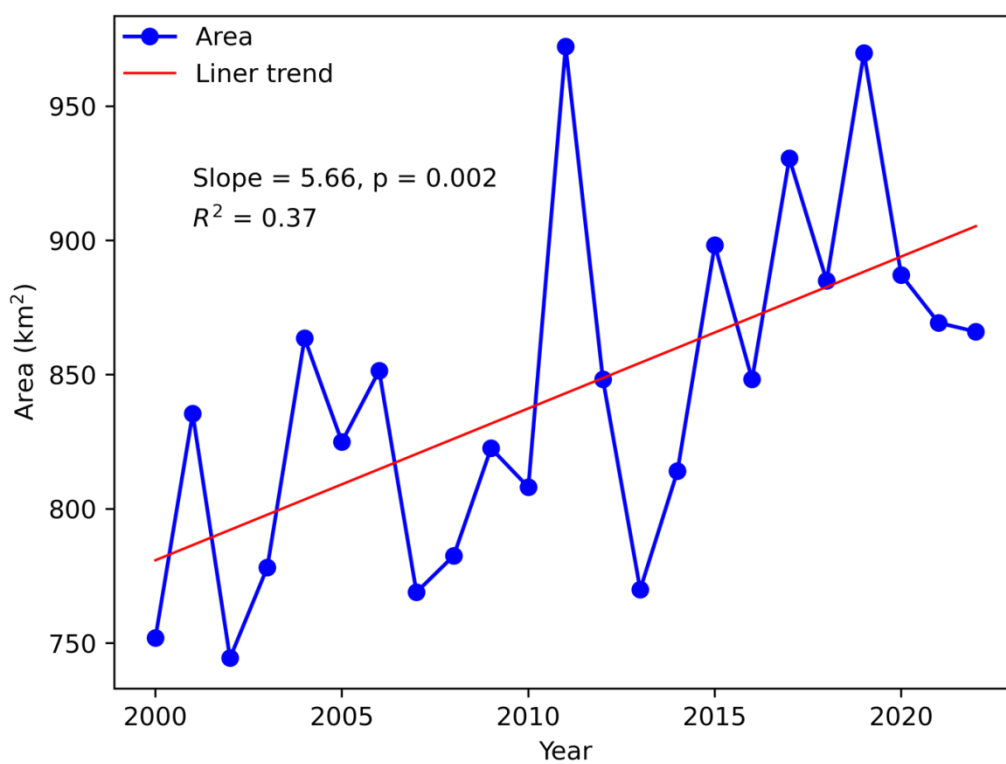
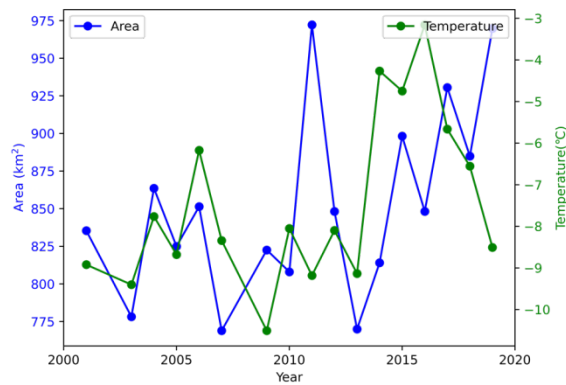


Figure 13: Inter-Annual Water Body Area Change and Linear Trend (2000–2022)

3.3 Relationship Between Water Body Area and Temperature



```
def draw_3(data_water, data_climate, pathout):
    import pandas as pd
    from scipy.interpolate import make_interp_spline
    import scipy.stats as stats
    from scipy.optimize import curve_fit
    from matplotlib import pyplot as plt
    from matplotlib.pyplot import MultipleLocator
    import matplotlib as mpl
    import matplotlib.gridspec as gridspec

    df_water = pd.read_csv(data_water, header=0)
    df_climate = pd.read_csv(data_climate, header=0)
    df_climate = df_climate.dropna()
    #print(df_climate)

    df_merged = pd.merge(df_water, df_climate, on='year', how='inner')
    #print(df_merged)

    fig, ax1 = plt.subplots()

    ax1.plot(df_merged['year'], df_merged['area'], color='blue', marker='o')
    ax1.set_xlabel('Year')
    ax1.set_ylabel('Area (km$^{2}$)', color='blue')
    ax1.tick_params(axis='y', labelcolor='blue')

    ax1.set_xlim(2000,2020)
    ax1.set_xticks([2000,2005,2010,2015,2020])
    ax1.set_xticklabels([2000,2005,2010,2015,2020], rotation=0)
    ax1.legend(['Area'], loc='upper left')

    # ax2 = ax1.twinx()
    # ax2.plot(df_merged['year'], df_merged['Temperature'], color='green', marker='o')
    # ax2.set_ylabel('Temperature (°C)', color='green')
    # ax2.tick_params(axis='y', labelcolor='green')
    # ax2.legend(['Temperature'], loc='upper right')

    plt.show()
```

Figure 14: Co-variation of Water Body Area and Annual Mean Temperature (2000–2020)

3.4 Relationship Between Water Body Area and Precipitation

The draw_3() function creates a dual-axis plot showing the co-variation of water body area and annual precipitation.

```
def draw_3(data_water, data_climate, pathout):
    import pandas as pd
    from scipy.interpolate import make_interp_spline
    import scipy.stats as stats
    from scipy.optimize import curve_fit
    from matplotlib import pyplot as plt
    from matplotlib.pyplot import MultipleLocator
    import matplotlib as mpl
    import matplotlib.gridspec as gridspec

    df_water = pd.read_csv(data_water, header=0)
    df_climate = pd.read_csv(data_climate, header=0)
    df_climate = df_climate.dropna()
    #print(df_climate)

    df_merged = pd.merge(df_water, df_climate, on='year', how='inner')
    #print(df_merged)

    fig, ax1 = plt.subplots()

    ax1.plot(df_merged['year'], df_merged['area'], color='blue', marker='o')
    ax1.set_xlabel('Year')
    ax1.set_ylabel('Area (km$^{2}$)', color='blue')
    ax1.tick_params(axis='y', labelcolor='blue')

    ax1.set_xlim(2000,2020)
    ax1.set_xticks([2000,2005,2010,2015,2020])
    ax1.set_xticklabels([2000,2005,2010,2015,2020], rotation=0)
    ax1.legend(['Area'], loc='upper left')

    # ax2 = ax1.twinx()
    # ax2.plot(df_merged['year'], df_merged['Precipitation'], color='green', marker='o')
    # ax2.set_ylabel('Precipitation (mm)', color='green')
    # ax2.tick_params(axis='y', labelcolor='green')
    # ax2.legend(['Precipitation'], loc='upper right')
```

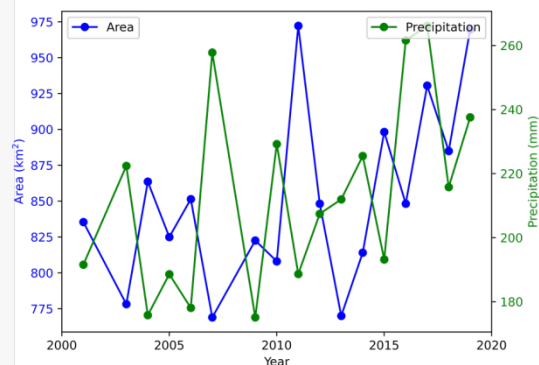


Figure 15: Co-variation of Water Body Area and Annual Precipitation (2000–2020)