Gridded Data Validator

User Manual

Saurav Bhattarai

March 2025

Contents

1 Introduction

The Climate Data Fetcher is a specialized tool designed for researchers and climate analysts to evaluate the accuracy of various gridded precipitation datasets by comparing them with ground station observations. This tool addresses a critical need in climate research: determining which gridded dataset provides the most reliable precipitation estimates for different regions.

1.1 Purpose and Capabilities

This manual will guide you through the installation, configuration, and usage of the Climate Data Fetcher. The tool enables users to:

- Download ground station precipitation data from the NOAA network via Meteostat
- Access gridded precipitation datasets including ERA5, DAYMET, and PRISM
- Calculate statistical metrics to quantify the agreement between gridded datasets and ground observations
- Generate visualizations that highlight spatial patterns in dataset accuracy
- Analyze performance for regular and extreme precipitation events
- Evaluate seasonal, monthly, and yearly performance differences

The primary goal is to help researchers determine which gridded dataset best represents actual precipitation patterns in their region of interest, improving the reliability of climate studies and hydrological modeling.

1.2 System Requirements

Before installing the Climate Data Fetcher, ensure your system meets the following requirements:

- Python 3.10 or higher
- Jupyter Notebook or JupyterLab
- Internet connection for data downloading
- Sufficient disk space for storing climate data
- Google Earth Engine account (for accessing gridded datasets)

2 Installation

2.1 Setting Up the Environment

Follow these steps to install the Climate Data Fetcher:

1. Clone the repository:

```
git clone https://github.com/Saurav-JSU/GeeData-GroundData-validator.git
cd GeeData-GroundData-validator

```

2. Create and activate a virtual environment:

```
# Using venv
python -m venv venv
source venv/bin/activate  # On Windows: venv\Scripts\activate

# Or using conda
conda create -n climate_fetcher python=3.10
conda activate climate_fetcher

```

3. Install dependencies:

```
pip install -r requirements.txt

```


2.2 Earth Engine Authentication

For accessing gridded datasets like ERA5, DAYMET, and PRISM, you need to authenticate with Google Earth Engine:

1. Sign up for a Google Earth Engine account at `https://earthengine.google.com/signup/` if you don't already have one
2. Install the Earth Engine Python API:

```
pip install earthengine-api

```

3. Authenticate your account:

```
earthengine authenticate

```

4. Follow the authentication steps in your browser

3 Project Structure

The Climate Data Fetcher is organized into the following directory structure:

```
climate_data_fetcher/
            config.py                # Configuration classes
            main.ipynb               # Main notebook interface
            requirements.txt         # Required Python packages
            src/
                  base_fetcher.py      # Abstract base classes
                  cli.py               # Command-line interface
                  data/                # Data fetching modules
                        ground_fetcher.py
                        gridded_fetcher.py
                  ui/                  # UI components
                        climate_ui.py
                  analysis/            # Statistical analysis
                        statistical_analyzer.py
                  visualization/       # Plotting and visualization
                      plot_results.py
            utils/                   # Utility functions
                plotting_utils.py
                seasonal_utils.py
                statistical_utils.py
                utils.py
```

Additionally, the tool creates the following directories for storing data and outputs:

```
climate_data_fetcher/
            Data/                    # Raw data files
            Results/                 # Statistical analysis results
            Plots/                   # Generated visualizations
```

4 Using the Jupyter Notebook Interface

The main interface for Climate Data Fetcher is a Jupyter Notebook (`main.ipynb`), which guides you through the entire workflow.

4.1 Running the Notebook

To start using the tool:

1. Activate your Python environment:

```
# Using venv
source venv/bin/activate  # On Windows: venv\Scripts\activate

# Or using conda
conda activate climate_fetcher

```

2. Launch Jupyter Notebook or JupyterLab:

```
jupyter notebook
# or
jupyter lab

```

3. Open `main.ipynb` from the file browser

4.2 Notebook Structure

The notebook is divided into four main sections:

1. Setup and Initialization
2. Data Selection and Download
3. Statistical Analysis
4. Visualization

It's important to run the cells in sequential order and complete each section before moving to the next.

5 Data Selection and Download

The first step in the workflow is selecting and downloading the climate data you need.

5.1 Interactive UI Overview

The notebook provides an interactive user interface for data selection with the following components:

- Data Type Selection: Choose between ground data, gridded data, or both
- Year Range Selection: Select the start and end years for data collection
- State Selection: Choose between all US states or specific states
- Gridded Dataset Selection: Select which gridded datasets to download
- Download Button: Initiates the data download process
- Output Area: Displays progress and results

5.2 Selecting Data Types

You can choose between three options:

- **Ground data only**: Downloads data from NOAA ground weather stations via Meteostat
- **Gridded data only**: Downloads data from selected gridded products
- **Both**: Downloads both ground and gridded data

5.3 Selecting Time Period

Use the year sliders to select the time period:

- **Start Year**: Between 1980 and 2024
- **End Year**: Between 1980 and 2024

Note that the end year must be equal to or later than the start year.

5.4 Selecting States

You can select states in two ways:

- **All US States**: Retrieves data for the entire United States
- **Select specific states**: Opens a multi-select widget where you can choose one or more specific states

5.5 Selecting Gridded Datasets

If you've chosen to download gridded data, you can select from the following datasets:

- **ERA5**: ECMWF Reanalysis v5 data (precipitation variable: total_precipitation_sum)
- **DAYMET**: Daily Surface Weather Data on a 1-km Grid for North America (precipitation variable: prcp)
- **PRISM**: Parameter-elevation Regressions on Independent Slopes Model (precipitation variable: ppt)

5.6 Downloading Data

Once you've made your selections:

1. Click the **Download Data** button
2. Monitor the progress in the output area
3. Wait for the "Processing complete" message before proceeding

6 Statistical Analysis

The core functionality of the Climate Data Fetcher is the statistical analysis that compares gridded precipitation datasets with ground observations.

6.1 Confirmation Step

Before running the analysis, you need to confirm that the data download is complete:

1. Run the confirmation cell
2. Click the **"I've completed data download, proceed with analysis"** button
3. Proceed to the next cell once confirmed

6.2 Running the Analysis

Run the analysis cell to start the statistical processing:

```
from src.analysis.statistical_analyzer import GriddedDataAnalyzer

analyzer = GriddedDataAnalyzer(data_dir='Data', results_dir='Results')
analyzer.run_analysis()
```


6.3 Analysis Processes

The analysis performs the following calculations for each gridded dataset, comparing it with ground station data:

- **Data Validation**: Checks if there's sufficient data for each station
- **Daily Statistics**: Calculates metrics for daily precipitation values
- **Extreme Value Statistics**: Analyzes performance for low (10th percentile) and high (90th percentile) precipitation events
- **Monthly Statistics**: Aggregates and analyzes monthly precipitation totals
- **Yearly Statistics**: Aggregates and analyzes yearly precipitation totals
- **Seasonal Statistics**: Groups data by season and calculates performance metrics

6.4 Statistical Metrics

The analysis calculates the following metrics to quantify how well each gridded dataset matches ground observations:

Metric	Description	Optimal Value
R^2	Coefficient of determination	1.0 (perfect fit)
RMSE	Root Mean Square Error	0.0 (no error)
Bias	Mean Bias	0.0 (unbiased)
MAE	Mean Absolute Error	0.0 (no error)
NSE	Nash-Sutcliffe Efficiency	1.0 (perfect match)
PBIAS	Percent Bias	0.0 (unbiased)
Correlation	Pearson correlation coefficient	1.0 (perfect correlation)

These metrics help identify which gridded dataset most accurately represents ground observations across different spatial and temporal scales.

6.5 Analysis Output

The analysis results are saved in the `Results/` directory, with a subdirectory for each dataset (ERA5, DAYMET, PRISM). Each subdirectory contains:

- `analysis_summary.csv`: Overview of the analysis
- `data_validation.csv`: Data quality validation results
- `daily_stats.csv`: Statistics for daily precipitation
- `monthly_stats.csv`: Statistics for monthly aggregated precipitation
- `yearly_stats.csv`: Statistics for yearly aggregated precipitation
- `low_extreme_stats.csv`: Statistics for low precipitation events (10th percentile)
- `high_extreme_stats.csv`: Statistics for high precipitation events (90th percentile)
- `seasonal_stats.csv`: Combined statistics for all seasons
- Individual season files: `winter_stats.csv`, `spring_stats.csv`, etc.

These files contain station-by-station statistical metrics that can be used to evaluate dataset performance at different spatial locations.

7 Visualization

After the statistical analysis, the Climate Data Fetcher generates visualizations to help interpret spatial patterns in the performance of different gridded datasets.

7.1 Running the Visualization

Run the visualization cell to generate all plots:

```
from src.visualization.plot_results import ResultPlotter

plotter = ResultPlotter()
plotter.run()
```

7.2 Types of Visualizations

The tool generates several types of visualizations, with emphasis on spatial patterns:

- **Spatial Distribution Plots**: Show performance metrics on maps, allowing you to see where each gridded dataset performs well or poorly
- **Box Plots**: Display the distribution of statistics across stations
- **Seasonal Comparison Plots**: Compare dataset performance across different seasons

The spatial distribution plots are particularly important as they help identify regional patterns in dataset accuracy. These maps allow researchers to determine which gridded dataset is most reliable for their specific study area.

7.3 Visualization Output

All plots are saved in the `Plots/` directory, with a subdirectory for each dataset (ERA5, DAYMET, PRISM). Each subdirectory contains:

- `daily_spatial.png`: Spatial distribution of daily statistics
- `daily_boxplot.png`: Box plots of daily statistics
- `monthly_spatial.png`: Spatial distribution of monthly statistics
- `monthly_boxplot.png`: Box plots of monthly statistics
- `yearly_spatial.png`: Spatial distribution of yearly statistics
- `yearly_boxplot.png`: Box plots of yearly statistics
- `seasonal_comparison.png`: Comparison of performance across seasons
- Seasonal spatial plots: `seasonal_winter_spatial.png`, etc.

7.4 Interpreting the Visualizations

When interpreting the spatial distribution plots:

- Look for patterns related to elevation, distance from coast, or urban areas
- Identify regions where all gridded datasets perform poorly (potential gaps in observation networks)
- Compare performance for normal precipitation versus extreme events (low and high)
- Note seasonal differences that may impact which dataset is most appropriate for different times of year

These visualizations help answer the fundamental question: "Which gridded precipitation dataset should I use for my study area?"

8 Understanding the Data Sources

8.1 Ground Station Data (Meteostat)

Ground station data serves as the reference dataset against which gridded products are compared:

- **Data Source**: National Oceanic and Atmospheric Administration (NOAA) ground weather stations
- **Access Method**: Via Meteostat Python package
- **Variable**: Daily precipitation (prcp)
- **Coverage**: Weather stations across the United States
- **Time Range**: 1980-2024 (availability varies by station)
- **Spatial Resolution**: Point-based measurements at station locations

While ground stations provide direct measurements, they have limitations including sparse spatial coverage, missing data, and potential measurement errors.

8.2 ERA5 Reanalysis

ERA5 is a global reanalysis dataset produced by the European Centre for Medium-Range Weather Forecasts:

- **Full Name**: ECMWF Reanalysis v5
- **Collection ID**: "ECMWF/ERA5_LAND/DAILY_AGGR"
- **Variable**: total_precipitation_sum
- **Units**: kg/m^2/s (converted to mm/day in the tool)
- **Spatial Resolution**: $\sim$9 km globally
- **Temporal Coverage**: 1980-present

ERA5 combines model data with observations using data assimilation. It provides complete spatial and temporal coverage but may not capture local precipitation features accurately.

8.3 DAYMET

DAYMET provides gridded estimates of daily weather parameters for North America:

- **Full Name**: Daily Surface Weather Data on a 1-km Grid for North America
- **Developer**: Oak Ridge National Laboratory
- **Collection ID**: "NASA/ORNL/DAYMET_V4"
- **Variable**: prcp
- **Units**: mm/day
- **Spatial Resolution**: 1 km
- **Temporal Coverage**: 1980-present

DAYMET's higher spatial resolution may better capture topographic effects on precipitation, but its accuracy varies regionally.

8.4 PRISM

PRISM uses point measurements and digital elevation models to produce continuous grid estimates:

- **Full Name**: Parameter-elevation Regressions on Independent Slopes Model
- **Developer**: Oregon State University
- **Collection ID**: "OREGONSTATE/PRISM/AN81d"
- **Variable**: ppt
- **Units**: mm
- **Spatial Resolution**: 4 km
- **Temporal Coverage**: 1981-present

PRISM is specifically designed to account for topographic effects on precipitation and is widely used in the United States, particularly in mountainous regions.

8.5 Google Earth Engine Platform

Google Earth Engine is used to access and process the gridded datasets:

- **Purpose**: Cloud computing platform for geospatial data analysis
- **Access Method**: Via the Earth Engine Python API
- **Functions Used**:
  - Filtering data by date and region
  - Extracting data at station locations for comparison
  - Unit conversions
  - Export to local files for analysis

The tool uses Earth Engine to efficiently extract gridded data at ground station locations, enabling direct station-to-grid comparisons.

9 Best Practices

9.1 Data Selection

For optimal evaluation of gridded precipitation datasets:

- Use at least 10 years of data for statistically robust comparisons
- Include multiple regions with varying terrain to identify spatial patterns in dataset performance
- Select all three gridded datasets (ERA5, DAYMET, PRISM) for comprehensive comparison
- For regional studies, focus on specific states rather than the entire US to get more detailed results
- Ensure the time period has sufficient ground station coverage for reliable validation

9.2 Computational Resources

To optimize performance:

- For large areas or long time periods, consider processing in smaller batches
- Close other applications when downloading data or running analysis
- If using limited hardware, avoid selecting the entire US for analysis
- Pre-download station metadata to speed up subsequent runs

9.3 Results Interpretation

When interpreting the comparison results:

- Remember that no gridded dataset will perfectly match ground observations
- Consider the native resolution of each gridded dataset when interpreting results
- Look for consistent patterns across multiple statistical metrics
- Pay attention to extreme event performance if your application involves floods or droughts
- Consider topographic effects when interpreting spatial patterns of accuracy
- Examine seasonal differences to determine if dataset performance varies throughout the year

9.4 Common Findings in Dataset Comparisons

While your specific results may vary, some common findings in gridded dataset evaluations include:

- Most gridded products tend to underestimate extreme precipitation events
- Performance often decreases in regions with complex topography
- Coastal areas frequently show greater discrepancies due to land-sea contrasts
- Urban areas may show impacts from the urban heat island effect
- Datasets with higher spatial resolution (like PRISM) often perform better in regions with heterogeneous terrain
- Performance can vary by season, with some datasets performing better in certain seasons

This knowledge can help guide your interpretation of the results produced by the Climate Data Fetcher.

10 Troubleshooting

10.1 Earth Engine Authentication Issues

If you encounter authentication issues with Earth Engine:

1. Ensure you're logged into your Google Earth Engine account
2. Run `earthengine authenticate` in the terminal
3. Follow the browser instructions to complete authentication
4. Restart the Jupyter notebook

10.2 Data Download Errors

For errors during data download:

- **No stations found**: Verify your state selection contains stations
- **Earth Engine initialization failed**: Check your authentication status
- **Network timeout**: Check your internet connection and retry
- **Empty results**: Try expanding your time range or geographic area

10.3 Common Analysis Errors

For errors during the analysis phase:

- **No common stations found**: Ensure both ground and gridded data were downloaded for the same region
- **Insufficient data**: Try a longer time period or different regions with better station coverage
- **Missing input files**: Verify that the data download step completed successfully before proceeding
- **Statistics calculation error**: Check for extreme outliers in your data that might disrupt calculations

10.4 Visualization Errors

For errors during the visualization phase:

- **Missing metadata**: Check that the station metadata file exists and was downloaded correctly
- **Missing statistics files**: Ensure the analysis step completed successfully
- **Empty plots**: Verify that your analysis produced valid results with sufficient data points
- **Missing required packages**: Install any additional Python packages like contextily for map backgrounds

11 Future Enhancements

While the current version of the Climate Data Fetcher focuses on comparing precipitation datasets, several potential enhancements could be implemented:

11.1 Additional Variables

The tool could be extended to compare other climate variables:

- Temperature (maximum, minimum, and mean)
- Humidity and vapor pressure
- Wind speed and direction
- Solar radiation

11.2 Additional Datasets

Support for more gridded datasets could be added:

- CHIRPS (Climate Hazards Group InfraRed Precipitation with Station data)
- GPM IMERG (Global Precipitation Measurement mission)
- NLDAS (North American Land Data Assimilation System)
- Satellite-derived precipitation estimates

11.3 Enhanced Analysis

Additional analysis capabilities could include:

- Calculation of standard climate indices (PRCPTOT, RX1day, RX5day, etc.)
- Trend analysis to identify changes in dataset accuracy over time
- Ensemble approaches that combine multiple gridded datasets
- Machine learning methods to improve gridded dataset accuracy

11.4 Extended Visualization

Visualization enhancements could include:

- Interactive web-based maps and plots
- Time series animations showing changes in accuracy over time
- Customizable color schemes and plot formats
- Direct export to common publication formats

12 References

1. Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., ... & Thépaut, J. N. (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999-2049.

2. Thornton, P. E., Thornton, M. M., Mayer, B. W., Wei, Y., Devarakonda, R., Vose, R. S., & Cook, R. B. (2018). Daymet: Daily Surface Weather Data on a 1-km Grid for North America, Version 3. ORNL DAAC, Oak Ridge, Tennessee, USA.

3. Daly, C., Halbleib, M., Smith, J. I., Gibson, W. P., Doggett, M. K., Taylor, G. H., Curtis, J., & Pasteris, P. A. (2008). Physiographically-sensitive mapping of temperature and precipitation across the conterminous United States. *International Journal of Climatology*, *28*, 2031-2064.

4. Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, *202*, 18-27.

5. Meteostat. (2023). Meteostat Python Package. `https://github.com/meteostat/meteostat-python`

13 Citation

If you use this tool in your research, please cite:

Bhattarai, S., & Talchabhadel, R. (2024). Comparative Analysis of Satellite-Based Precipitation Data across the CONUS and Hawaii: Identifying Optimal Satellite Performance. *Remote Sensing*, *16*(16), 3058. `https://doi.org/10.3390/rs16163058`