

Supplementary Information

Verification of Data Provenance

To confirm the provenance of the climatic data used in this study, we validated one representative location (Kuwait City; 29.37° N, 47.98° E) by downloading ERA5 data directly from the Copernicus Climate Data Store (CDS). Values matched the Open-Meteo ERA5 output within rounding precision, confirming that the Open-Meteo Historical API accesses ERA5 atmospheric fields without blending with ERA5-Land or other reanalyses. For transparency and reproducibility, the exact API query, dataset metadata, and the Python source code used for data retrieval are provided below.

API Query Example

```
https://archive-api.open-meteo.com/v1/archive?
latitude=29.37&longitude=47.98&
start_date=2000-01-01&end_date=2022-12-31&
hourly=temperature_2m,relative_humidity_2m,
precipitation,shortwave_radiation&timezone=GMT
```

Python Source Code: Download_Weather_Data.py

```
# Python Download_Weather_Data.py
"""
- This script downloads historical weather data for various locations
  using the Open-Meteo API, takes location coordinates from an Excel file,
  fetches hourly weather data, and saves the results.
"""

import pandas as pd
import openmeteo_requests
import requests_cache
import time, xlsxwriter
from retry_requests import retry
from tqdm import tqdm

# Setup the Open-Meteo API client with cache and retry on error
cache_session = requests_cache.CachedSession('.cache', expire_after=-1)
retry_session = retry(cache_session, retries=5, backoff_factor=0.2)
openmeteo = openmeteo_requests.Client(session=retry_session)

# Load existing location data from Excel
location_data_df = pd.read_excel(r'./Python_Outputs/locations_coordinates.xlsx')

# Iterate through the data frame rows
for index, location_row in tqdm(location_data_df.iterrows(),
                                total=location_data_df.shape[0],
```

```

                                desc="Downloading the Weather Data"):
location_name = location_row['City']
location_country = location_row['Country']
latitude = location_row['Latitude']
longitude = location_row['Longitude']
print('\n' + location_name + '_' + location_country)

# Define API parameters
url = "https://archive-api.open-meteo.com/v1/archive"
params = {
    "latitude": latitude,
    "longitude": longitude,
    "start_date": "2000-01-01",
    "end_date": "2022-12-31",
    "hourly": ["temperature_2m", "relative_humidity_2m",
               "precipitation", "shortwave_radiation"],
    "timezone": "GMT"
}
time.sleep(1)

# Get hourly weather data
responses = openmeteo.weather_api(url, params=params)
response = responses[0]
hourly = response.Hourly()

# Process location details
location_details = {
    "Location": location_name + '_' + location_country,
    "Latitude": latitude,
    "Longitude": longitude,
    "timezone": "GMT"
}
location_df = pd.DataFrame([location_details])

# Process hourly data
hourly_data = {
    "time": pd.date_range(
        start=pd.to_datetime(hourly.Time(), unit="s"),
        end=pd.to_datetime(hourly.TimeEnd(), unit="s"),
        freq=pd.Timedelta(seconds=hourly.Interval()),
        inclusive="left"
    ),
    "temperature_2m (°C)": hourly.Variables(0).ValuesAsNumpy(),
    "relative_humidity_2m (%)": hourly.Variables(1).ValuesAsNumpy(),
    "precipitation (mm)": hourly.Variables(2).ValuesAsNumpy(),

```

```

        "shortwave_radiation (W/m2)": hourly.Variables(3).ValuesAsNumpy(),
    }

weather_df = pd.DataFrame(data=hourly_data)
weather_df['time'] = pd.to_datetime(weather_df['time'])
weather_df['Year'] = weather_df['time'].dt.year
weather_df['Month'] = weather_df['time'].dt.month
weather_df['Date'] = weather_df['time'].dt.day
weather_df['Time(hour)'] = weather_df['time'].apply(lambda x: x.hour)
weather_df = weather_df.drop('time', axis=1)
weather_df = weather_df[['Year', 'Month', 'Date', 'Time(hour)']
                        + [c for c in weather_df.columns
                           if c not in ['Year', 'Month', 'Date', 'Time(hour)']]]
print('\tWeather data is downloaded')

file_name = f'{location_name}_{location_country}.xlsx'
excel_file_path = (r'./Python_Outputs/Downloaded_Weather_Data_files/'
                  + file_name)
with pd.ExcelWriter(excel_file_path, engine='xlsxwriter') as writer:
    location_df.to_excel(writer, sheet_name='location_df', index=False)
    weather_df.to_excel(writer, sheet_name='weather_df', index=False)

```

This code ensures reproducible, automated retrieval of ERA5-based hourly weather data for all analyzed sites between 2000 and 2022.