03_report

November 6, 2025

1 Project Report

1.1 Part 3 - Data Sources

1.1.1 Fabian Heflo

1.1.2 GitHub Repository:

https://github.com/fheflo1/IND320

1.1.3 Streamlit App:

https://ind320-fheflo1.streamlit.app/

1.2 AI Usage and Reflections

AI is consitantly used for this project. Setting up the resources correctly and guiding me through my issues. I had around 70% code completions from Copilot on this assignment. If we compare it to the previous assignment, I had only around 30% code completions. Since I have used a structure (explained in **01_report** under Compulsory Work) for the Elhub data which can be repeated for the open-meteo data, copilot knew how to complete a lot of the code I was producing. I also used ChatGPT to set up the streamlit pages.

1.3 Compulsory Work

Since I used a structure with **Bronze**, **Silver** and **Gold** for the Elhub data in the previous assignment, I kept the system for this assignment. The structure became so that inside the **notebooks/**, I add a folder for each API I want to fetch and process data. Trying to do this modularly with functions stored in **src/**. Goal is to be fully modular. For each of the different APIs there will be a folder, for now Elhub and meteo, consisting of the **Bronze**, **Silver** and **Gold** structure. Since the Open-meteo data didn't need any processing, it only goes to silver, with the analysis there.

The Elhub data was uploaded to a MongoDB server after being stored and processed in Cassandra. This was not necessary for the meteo-data. The Streamlit page was supposed to just fetch the data via API, so the process for working with the meteo-data in the notebooks was quite short. The focus was mostly on the analytical aspect of the assignment, trying to get good plots and readable results.

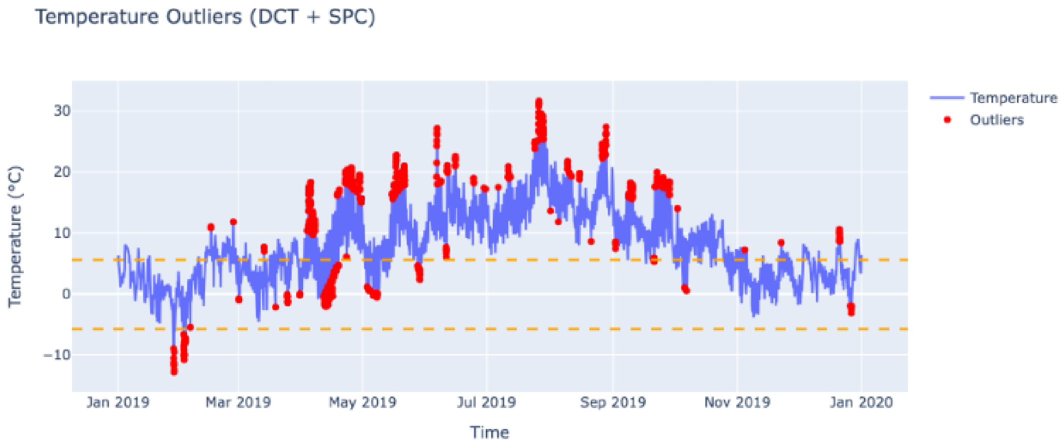
The restructuring of the pages for the Streamlit app was one of the hardest parts about this assignment. Fixing old pages, renaming and making sure the new pages, A and B, were correct. After reviewing feedback from others, I have also fixed a lot when it comes to the aesthetics, e.g. consistently using one color for a specific feature on the **Energy Production** page. The buttons on the front page are still weird, but don't bug and overlap when the app is not in full screen.

1.3.1 Plots from 02_meteo_silver.ipynb and 04_elhub_gold.ipynb

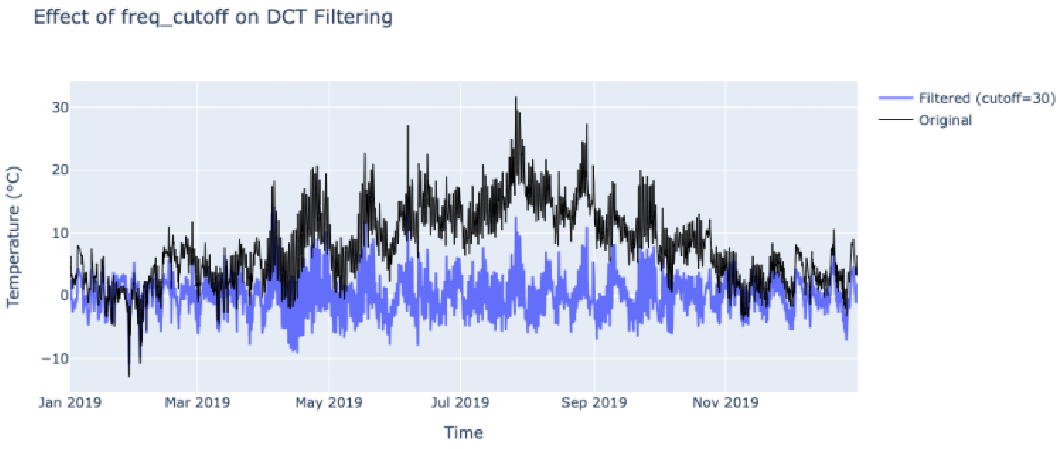
[26]:

```python
from pathlib import Path
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

root = Path.cwd().parents[0]
plot_dir = root / "data/plots"

# List of plot filenames
plot_files = ["temp_DCT_SPC.png", "DCT.png", "perc_LOF.png", "STL_hydro.png", "spectrogram_hydro.png"]

for file in plot_files:
    img_path = plot_dir / file
    img = mpimg.imread(img_path)

    plt.figure(figsize=(10, 6))
    plt.imshow(img)
    plt.axis("off")
    plt.title(file.replace(".png", "").replace("_", " "), fontsize=14, pad=20)
    plt.show()
```

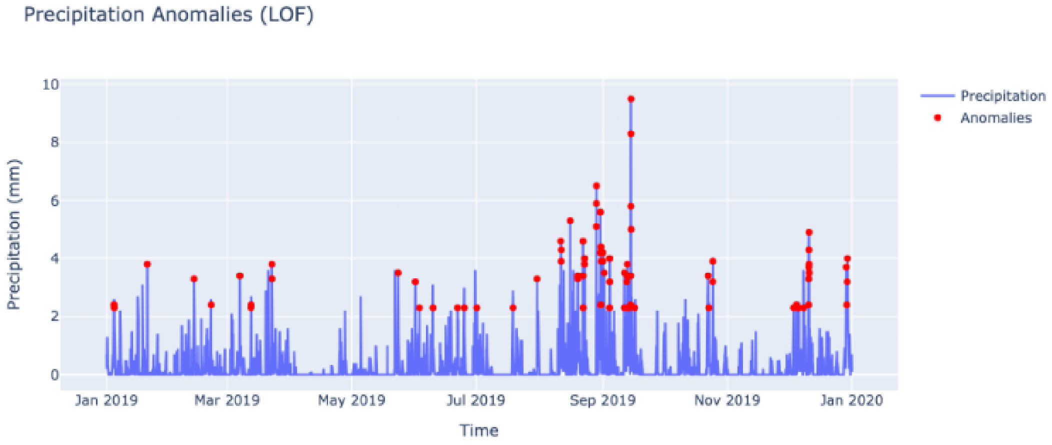
temp DCT SPC



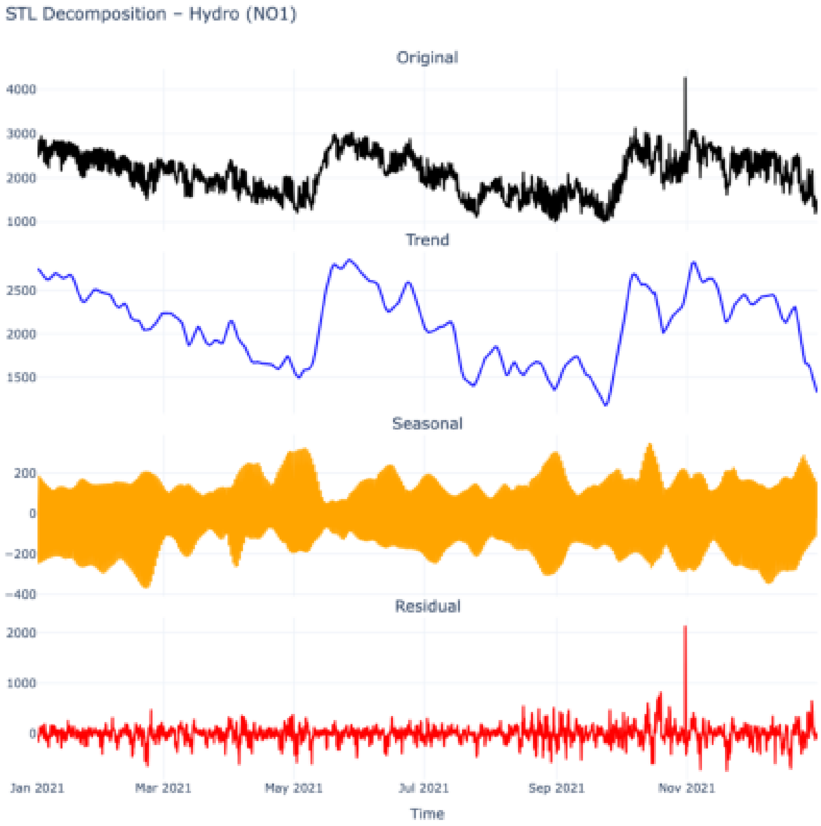
DCT


Effect of freq_cutoff on DCT Filtering
Filtered (cutoff=30)
Original
Temperature (°C)
30
20
10
0
−10
Jan 2019
Mar 2019
May 2019
Jul 2019
Sep 2019
Nov 2019
Time

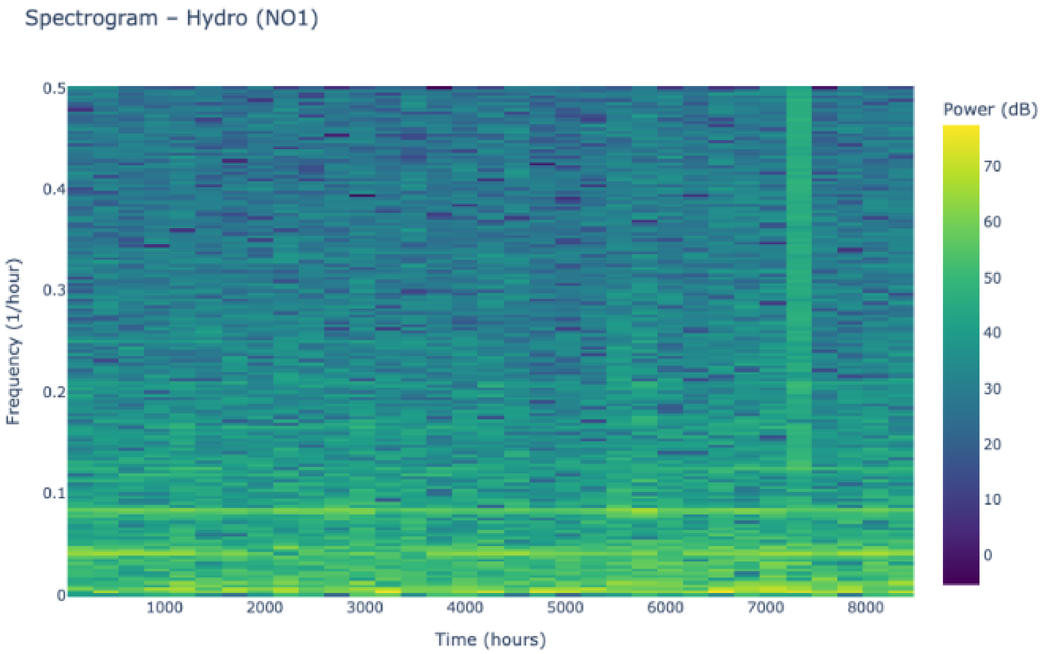
perc LOF


Precipitation Anomalies (LOF)
Precipitation
Anomalies
Precipitation (mm)
10
8
6
4
2
0
Jan 2019
Mar 2019
May 2019
Jul 2019
Sep 2019
Nov 2019
Jan 2020
Time

STL hydro


STL Decomposition – Hydro (NO1)
Original
4000
3000
2000
1000
Trend
2500
2000
1500
Seasonal
200
0
−200
−400
Residual
2000
1000
0
Jan 2021
Mar 2021
May 2021
Jul 2021
Sep 2021
Nov 2021
Time

spectrogram hydro


Spectrogram – Hydro (NO1)
Frequency (1/hour)
Time (hours)
Power (dB)