TU Clausthal
Clausthal University of Technology

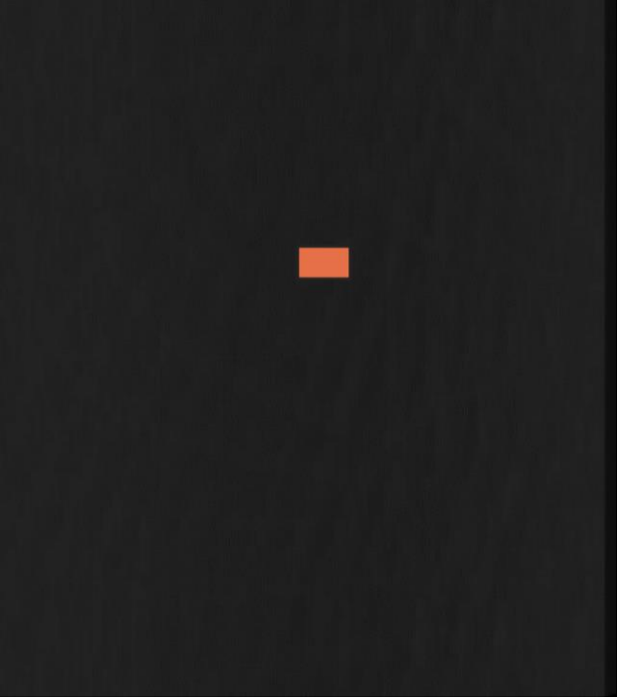
PREDICTION OF EXTREME EVENTS USING IMAGES FROM PRECIPITATION RADAR

SUPERVISORS:

- Prof. Dr. Andreas Rausch
- Dr. Stefan Wittek
- MS.c Dhananjeyan Jeyaraj
- **PRODUCED BY:**
- Arman Abouali

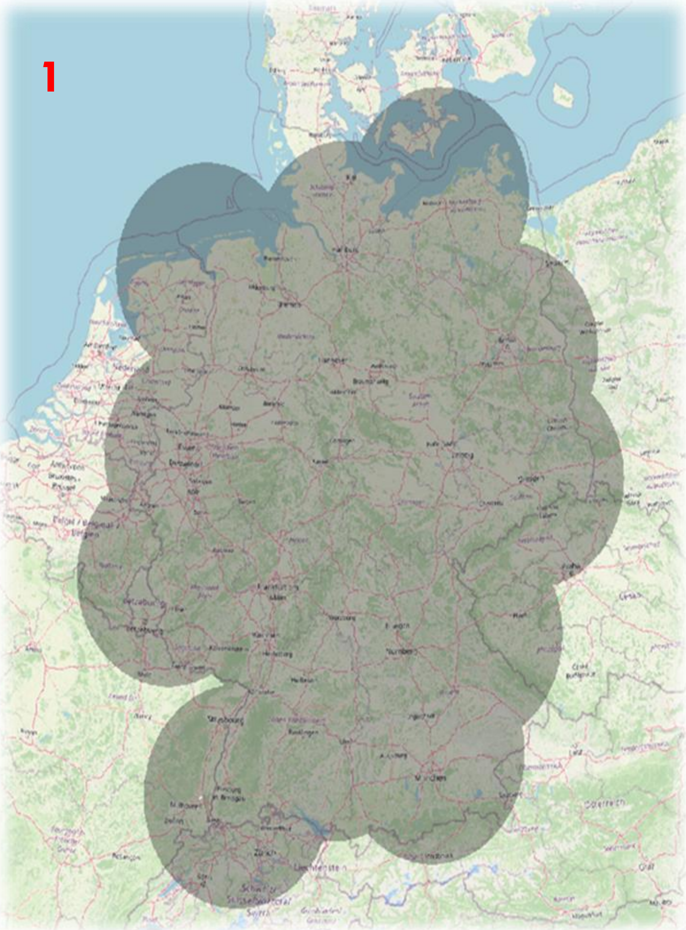
PROJECT METHODOLOGY

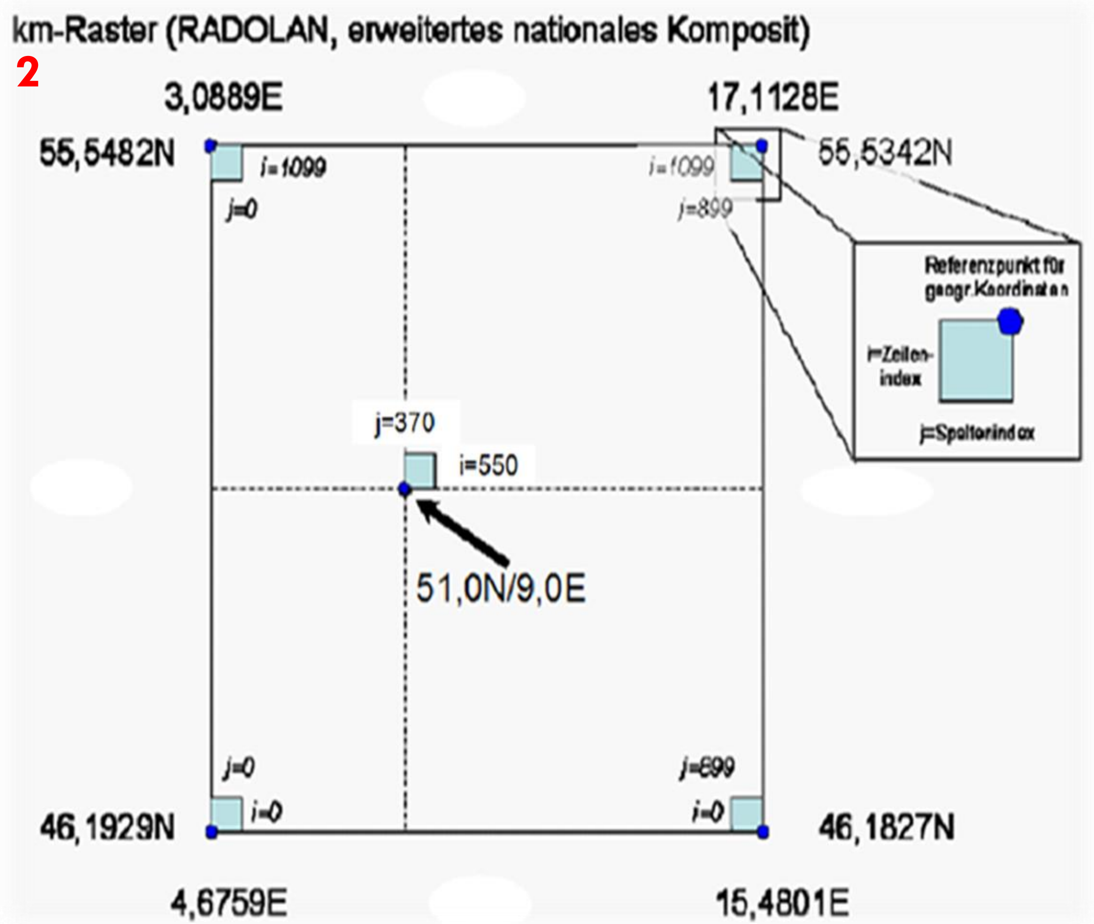
1. **Preprocessing**
   - Finding coordinates of the target location using QGIS software.
   - DWD_Bot: A pipeline for downloading and extracting radar imagery data from the DWD website and storing it for future processing.
2. **Data Analysis**
   - Evaluation of the calculated location on an image file.
   - Comparison of the measurement of the precipitation by the radar image data with the measurements of the sensor in the same location.
3. **Data Modeling**
   - Implementation of deep learning models to the image radar data to assess its effectiveness in forecasting a rare event. (Use case: Flood on 26th July 2017, Goslar, Niedersachsen, Germany)

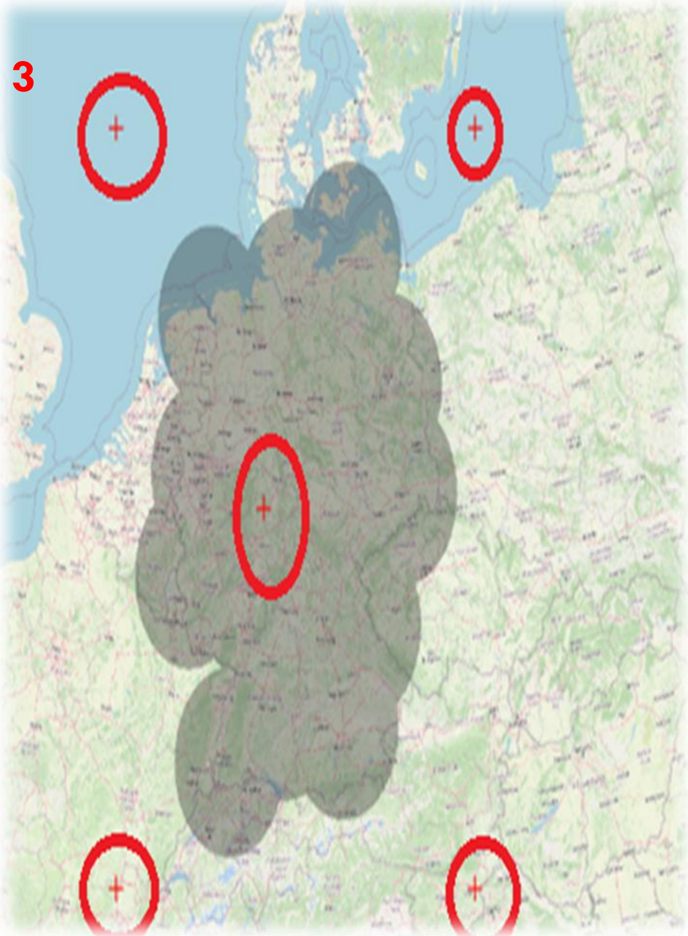


PREPROCESSING
INITIAL APPROACH

- Investigated how to configure and set base layers to obtain coordinates for the area of interest in QGIS software.
- Implemented using Bokeh tool from an extracted image file.
  - **Failures:**
  - Zooming in causes the image to become blurry.
  - low accuracy of the extracted coordinate values.
  - The sized sliced window was 40 by 70 pixels.

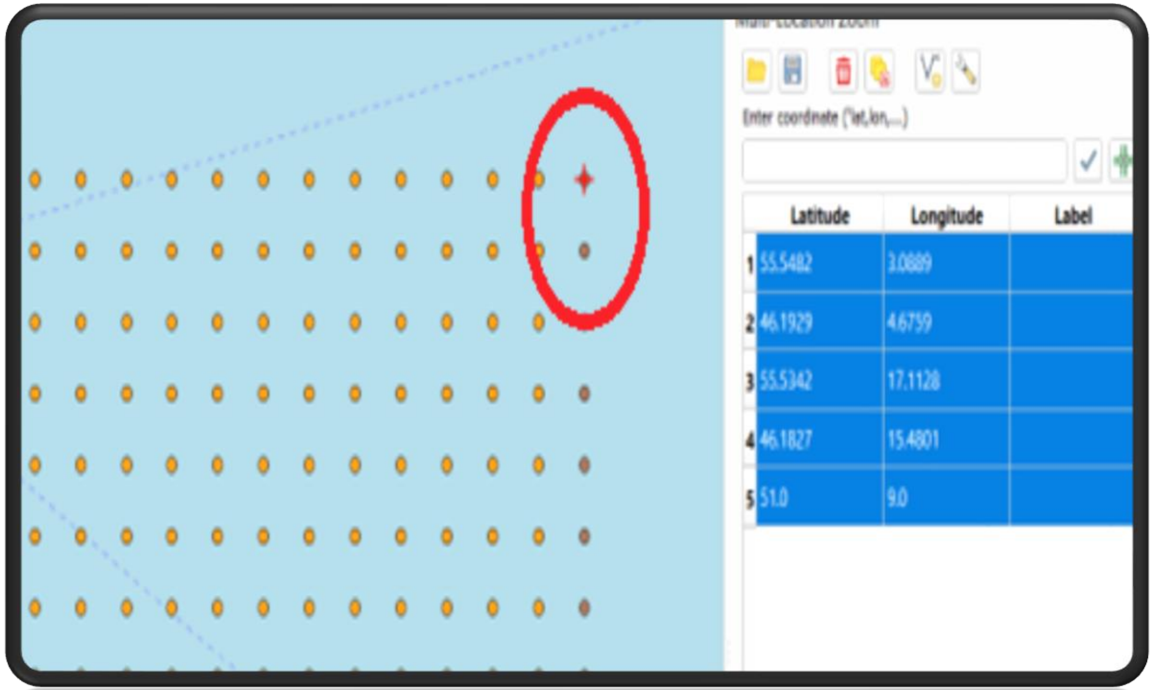


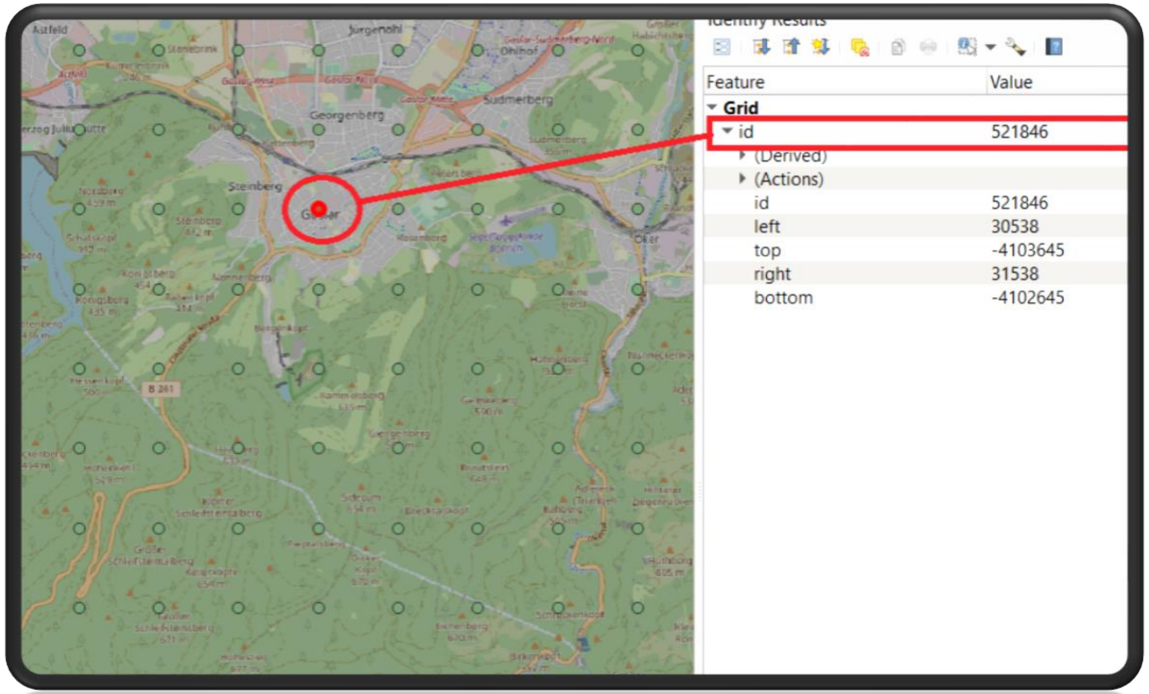




PREPROCESSING

SECTION ONE: FINDING TARGET COORDINATES AND ADJUST THE GRID LOCATION





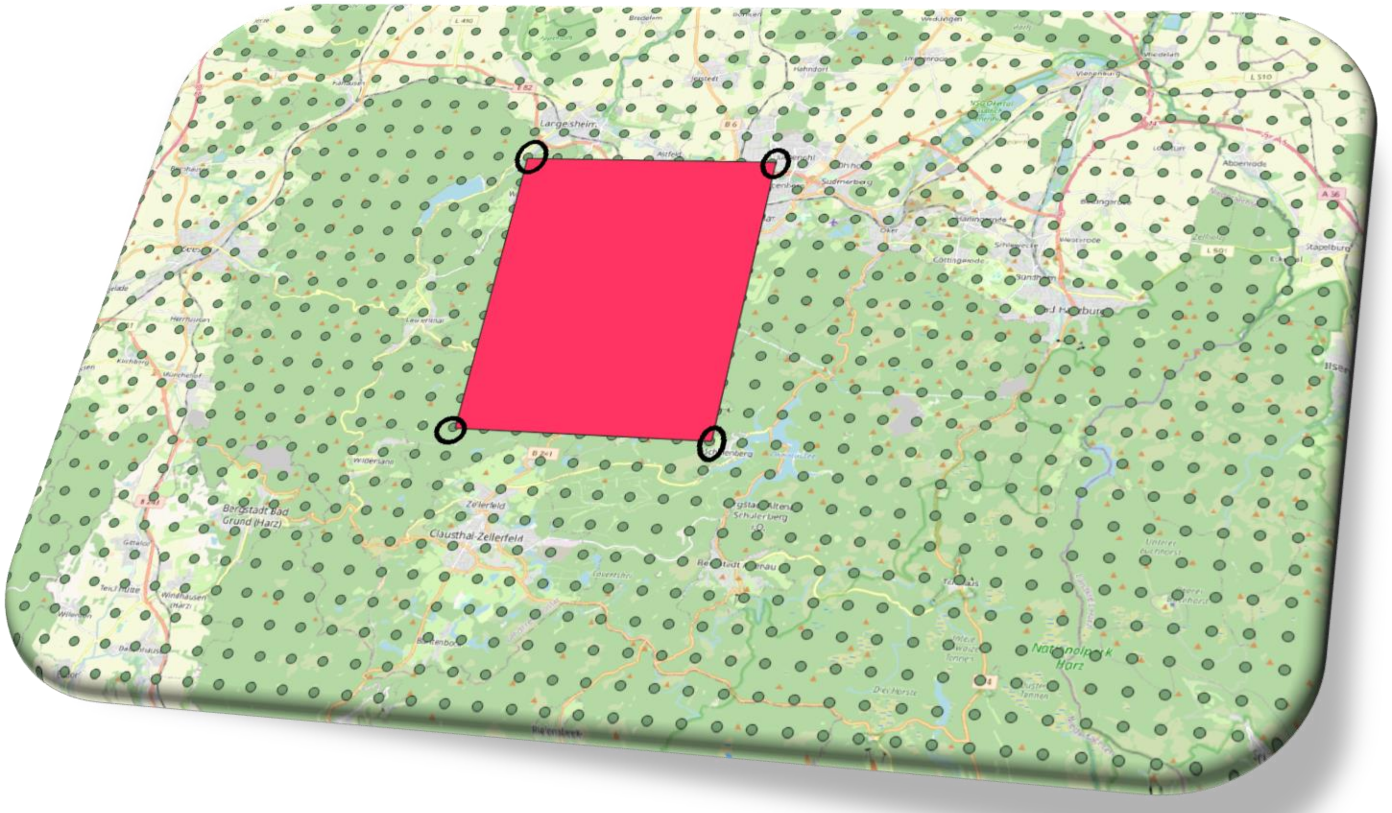
PREPROCESSING

SECTION ONE: FINDING TARGET COORDINATES AND ADJUST THE GRID LOCATION

PREPROCESSING

SECTION ONE: FINDING TARGET COORDINATES AND ADJUST THE GRID LOCATION

- Write down ID values of these edge points.
- Reshape them into 901 rows and 1101 columns to be matched with the image coordinates.
- Reduced size of the sliced window to an 8 by 8 window.
- Last but not least, use these coordinate values to slice images into a smaller window.



PREPROCESSING

SECTION TWO: FUNCTIONALITIES OF THE DWD_BOT

i. Downloading data from the DWD website and storing them in a specified address by the user.

ii. Extract the downloaded files in order to get the data for a month.

iii. Unzip the extracted files to get files for the days of the month.

iv. Extract data to obtain ASCII files for each day.

v. Aggregate every 3 images into 1, to make it comparable with sensor data.

vi. Compress original files in a .zip format.

vii. Slice a new window based on the calculated coordinate of x and y.

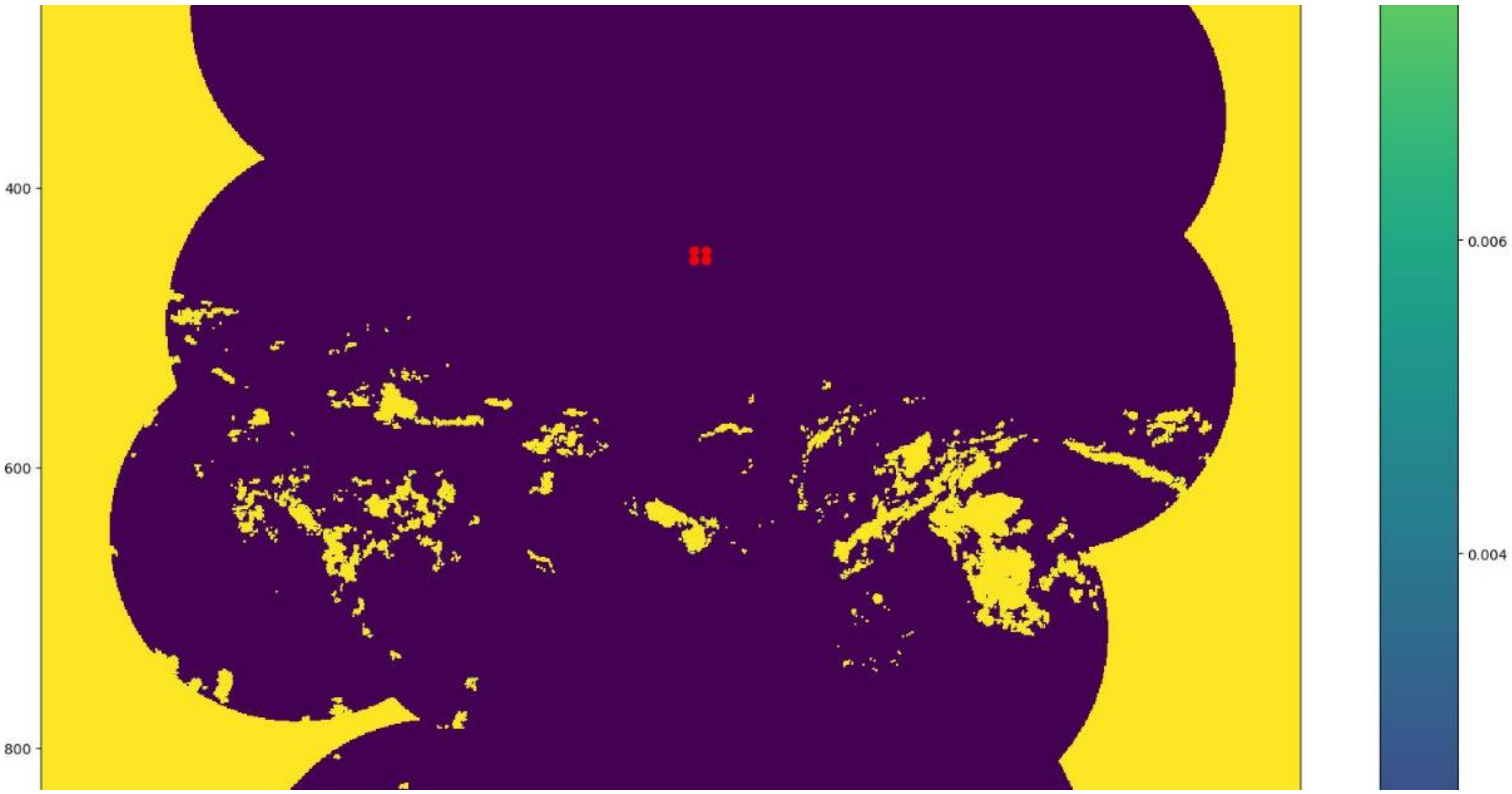
viii. A pipeline to automate the procedure.

ix. A functionality is available to access the zip files of step 6, unzip them, and slice them according to a new coordinate.

PREPROCESSING

SECTION TWO: FUNCTIONALITIES OF THE DWD_BOT

- **Outcome of the Bot:**
- Preserve images in their initial dimension (1100 * 900) for reprocessing purposes.
- Possibility of processing a single data.
- All of the image-sliced window files will be stored and maintained in a parquet file.
- **Why Parquet file:**
  - Compressed 35 GB of data into 5 MB while preserving data quality and structure.
  - Easily convertible and processable using a Pandas data frame.



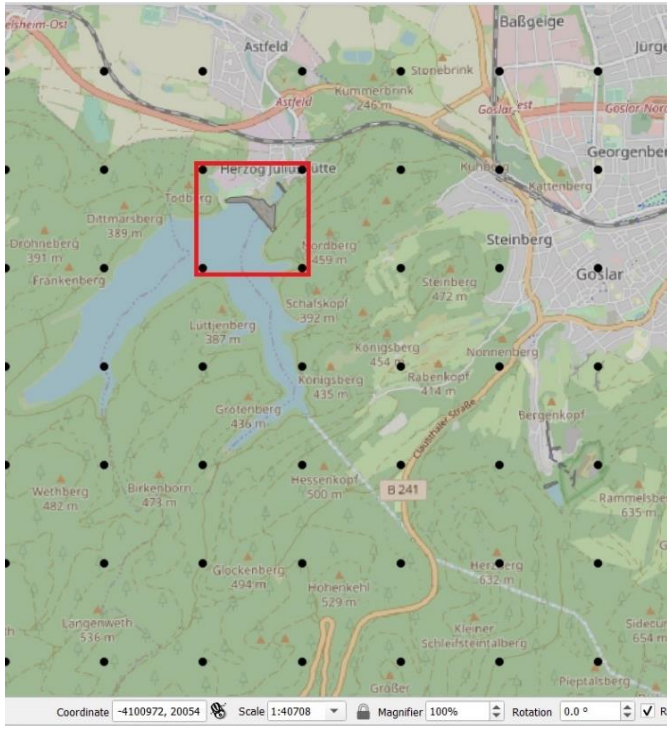
DATA ANALYSIS

SECTION ONE: EVALUATION OF THE CALCULATED LOCATION ON AN IMAGE FILE

DATA ANALYSIS

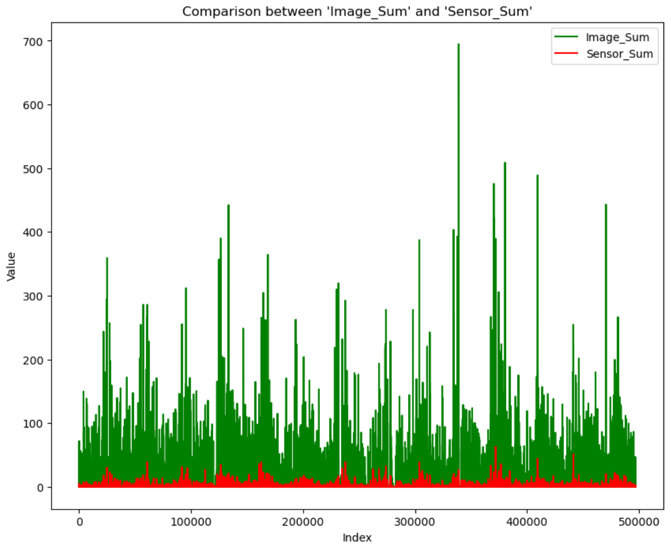
SECTION ONE: EVALUATION OF THE CALCULATED LOCATION ON AN IMAGE FILE

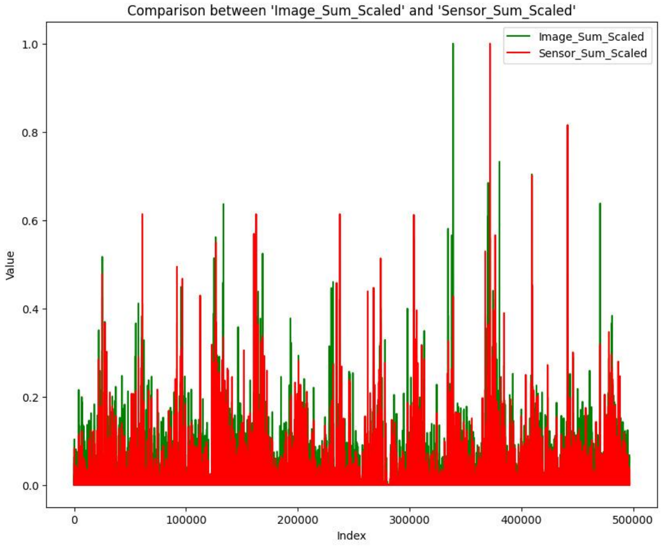
- Calculate the coordinates of a 1*1 window from image data from a particular place where the "Granetalsperre" sensor is situated.
- Run the procedure in pipeline to download and extract data from Nov 2003 to Dec 2017.
- Comparing precipitation measured by the sensor and image in this particular location.
- Correlation result = 0.23



DATA ANALYSIS
SECTION TWO: COMPARISON

- **Goal:**
  - To conduct a comparative analysis of the precipitation data obtained from the sensors and the image data.
  - **Procedure:**
    - The sum of each sensor's precipitation measurement column is stored in the 'Sensor_Sum' column.
    - Compute the total precipitation quantity across all cells in each image and store the result in the ‘Image_Sum’ column.
    - 8% of the cases ‘Image_Sum’ < ‘Sensor_Sum’





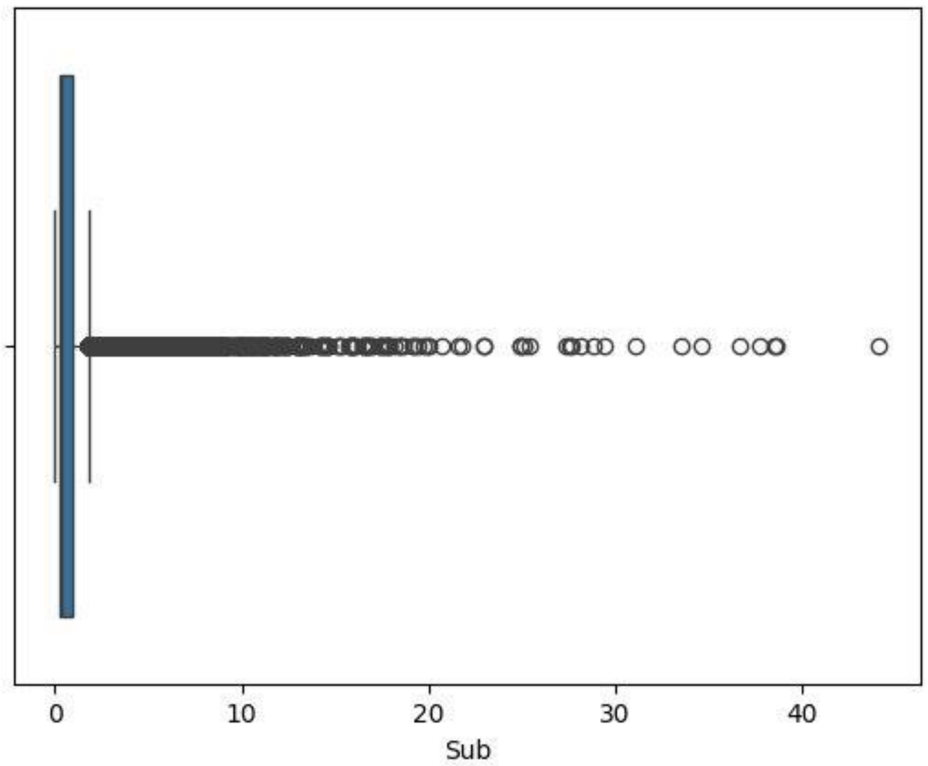
DATA ANALYSIS

SECTION TWO: COMPARISON

- Statistical analysis of their calculated subtraction.

Mean	Max	Min	Median	Mode
0.85	44.1	5.55×10^{-17}	0.4	0.3

- Mean (0.85) Greater than Median (0.4) and Mode (0.3).
- Maximum Value (44.1) Significantly Higher.
- Therefore, "common" events are low in magnitude and "rare" events are very high in magnitude.

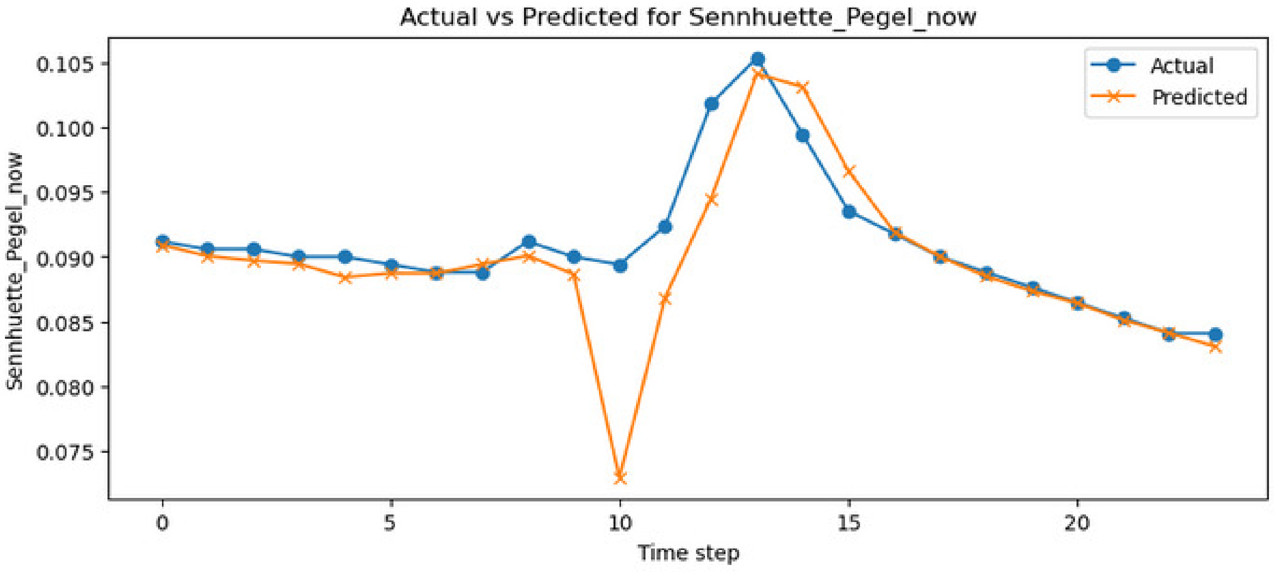


DATA MODELING

CNN-LSTM MODEL

- Input data, [Aug 2016 – Aug 2017].
- Both sensor and image data are involved.
- A low error rate in statistical findings allows us to evaluate the model as 'GOOD'.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv (TimeDistributed)      (None, 4, 69, 39, 32)     320

 maxpool (TimeDistributed)   (None, 4, 34, 19, 32)     0

 conv2 (TimeDistributed)     (None, 4, 32, 17, 64)     18496

 maxpool2 (TimeDistributed)  (None, 4, 16, 8, 64)      0

 flatten (TimeDistributed)   (None, 4, 8192)           0

 lstm (LSTM)                 (None, 4, 64)             2113792

 lstm2 (LSTM)                (None, 32)                12416

 output (Dense)              (None, 2)                 66

=================================================================
Total params: 2145090 (8.18 MB)
Trainable params: 2145090 (8.18 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

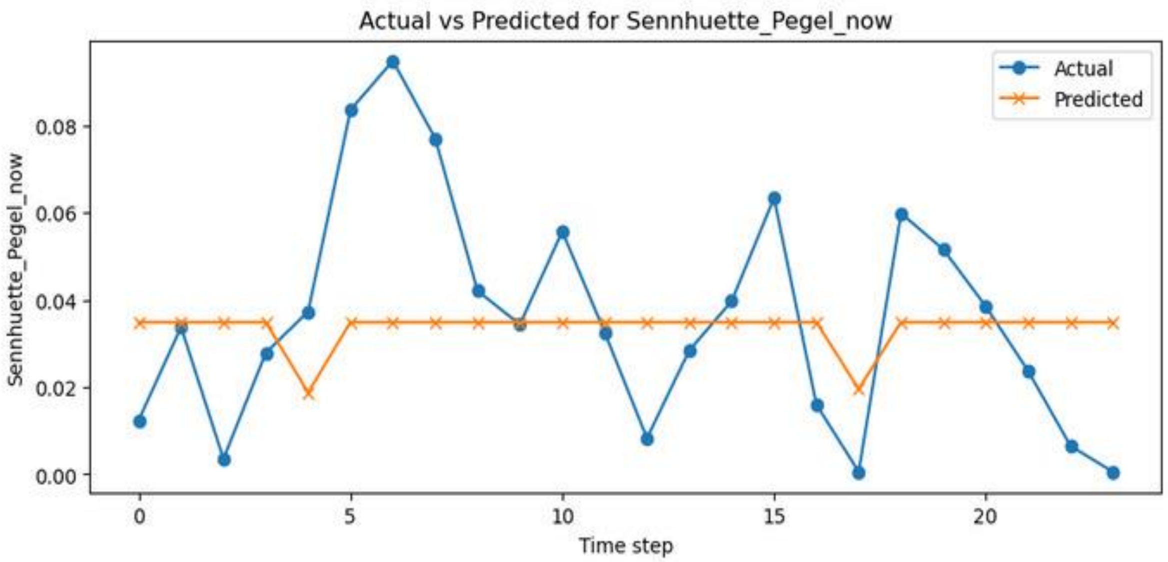


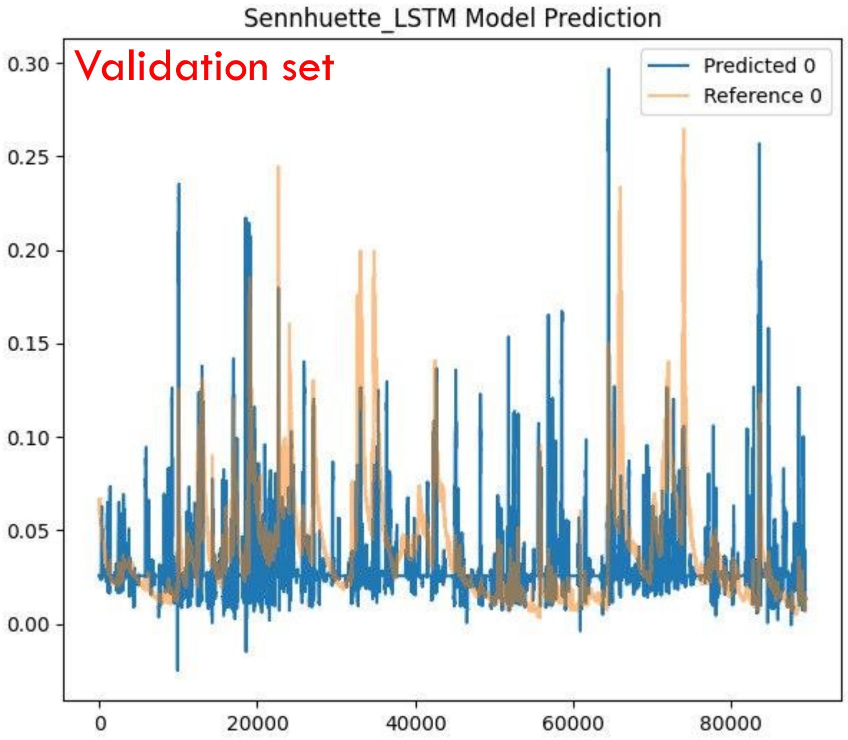
DATA MODELING

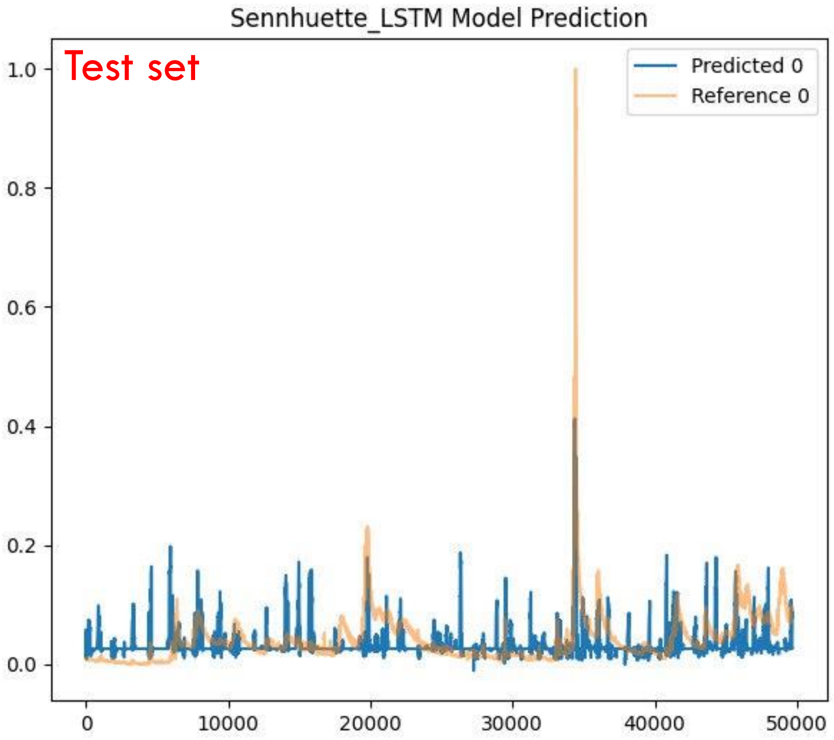
CNN MODEL

- Only image data is involved.
- Input data, [Aug 2016 – Aug 2017].
- Statistical findings as well as the prediction plot allows us to evaluate the model as ‘BAD’.
- Size of the sliced window is 40 * 70

```
Model: "sequential"

 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 8, 8, 64)          640

 max_pooling2d (MaxPooling2  (None, 4, 4, 64)          0
 D)

 conv2d_1 (Conv2D)           (None, 4, 4, 32)          18464

 max_pooling2d_1 (MaxPoolin  (None, 2, 2, 32)          0
 g2D)

 flatten (Flatten)           (None, 128)               0

 dense (Dense)               (None, 128)               16512

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 43937 (171.63 KB)
Trainable params: 43937 (171.63 KB)
Non-trainable params: 0 (0.00 Byte)
```







DATA MODELLING

SENNHUETTE LSTM MODEL PREDICTION

CONCLUSION AND FUTURE WORKS

- Parquet files were used to solve the image file storage issue.
- Automate Bot, to prepare required data.
- In Phase of analysis,
  - Investigated on evaluation of the calculated coordinates on the target location.
  - Investigated on finding a comparison between image and sensor data.
- In phase of modeling,
  - Experienced with several models' architecture.
  - Models have ample opportunity for improvement at all times.

Thank You for Listening

Any Questions?