Hands-on session “users”

Exercises

Exercise 1: Install and configure pysteps in Colab

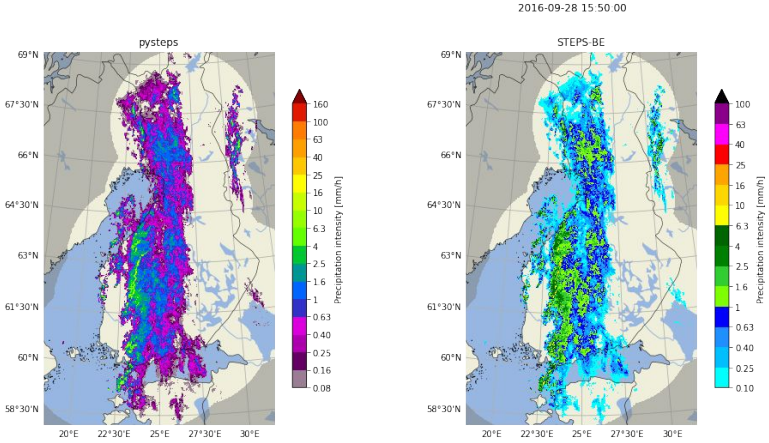
1. Install pysteps dependencies by using pip or conda-colab & mamba, see environment.yml
2. Install pysteps by using pip & git
3. Download example data to Google drive
4. Create `pystepsrc` and configure pysteps

Exercise 2: Read, visualize and process input data

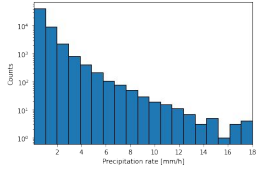
1. Install pysteps with helper_setup.ipynb notebook
2. Read the example data downloaded above, pick either fmi or mch dataset
3. Plot precipitation fields on top of a basemap
4. Try different preprocessing methods:
    a. upsampling
    b. clipping
    c. unit conversions
    d. Logarithmic transformation
    e. normal quantile transformation

Exercise 2: Read, visualize and process input data

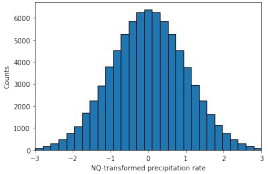
Basemap + precipitation with different colormaps



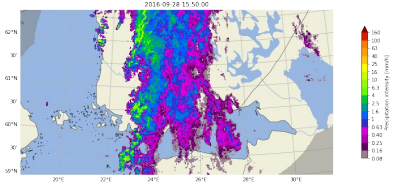
Distribution of precipitation rates



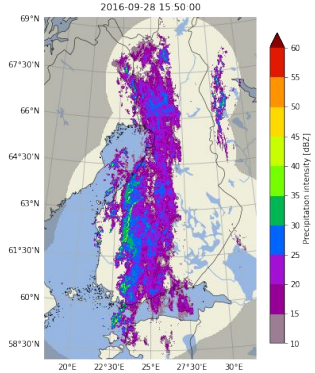
After NQ transformation



Clipping



Conversion:
rain rate <-> reflectivity



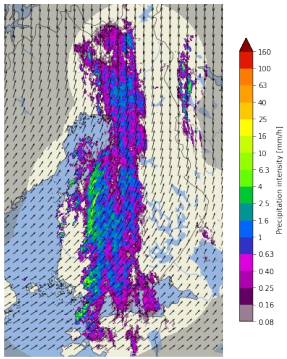
Exercise 3: Optical flow and extrapolation

1. Load example data with helper_input_data.ipynb notebook
2. Try different optical flow methods: Lucas-Kanade, VET and DARTS. Questions:
    a. What the advection fields look like? Which one would you trust?
    b. What is the effect of data transformations (e.g. log-transform) before applying the optical flow?
3. Apply semi-Lagrangian extrapolation and plot the extrapolated precipitation fields for lead times of 15, 30, 45 and 60 minutes
4. Apply extrapolation with time steps that are different from the input data

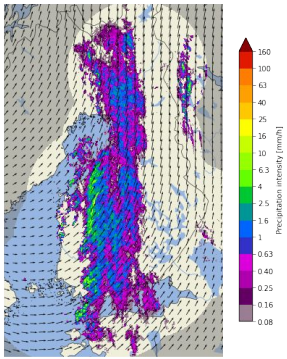
Exercise 3: Optical flow and extrapolation

Different optical flow methods

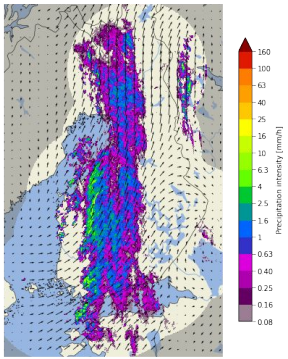
Lucas-Kanade



VET

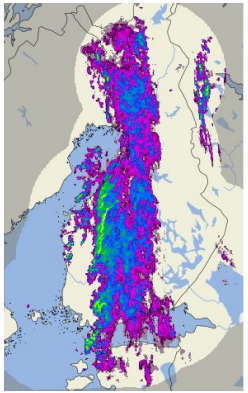


DARTS

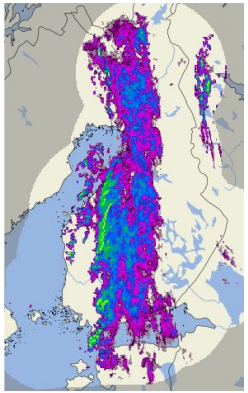


Extrapolation

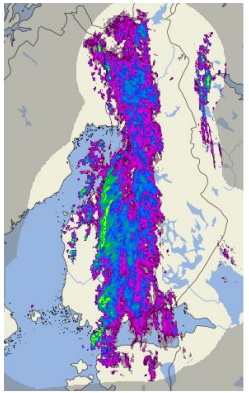
15 minutes



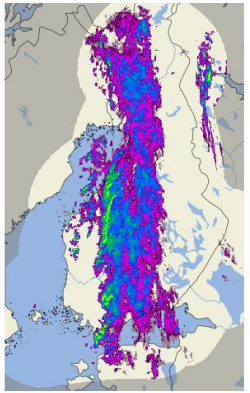
30 minutes



45 minutes



60 minutes



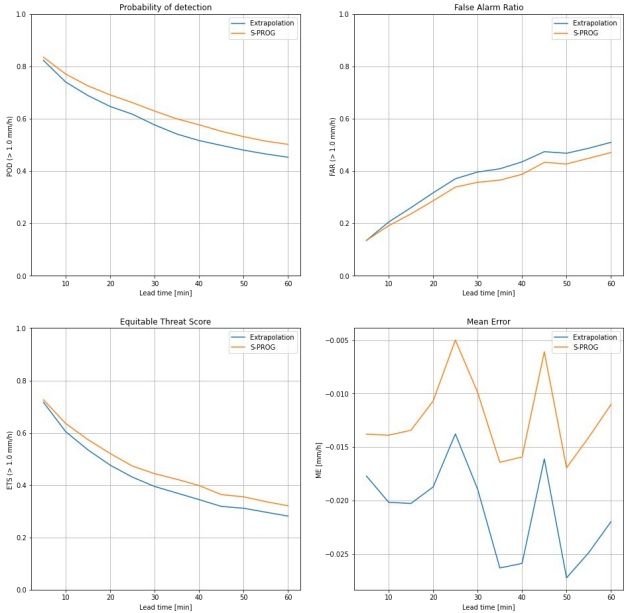
Exercise 4: Deterministic nowcasting

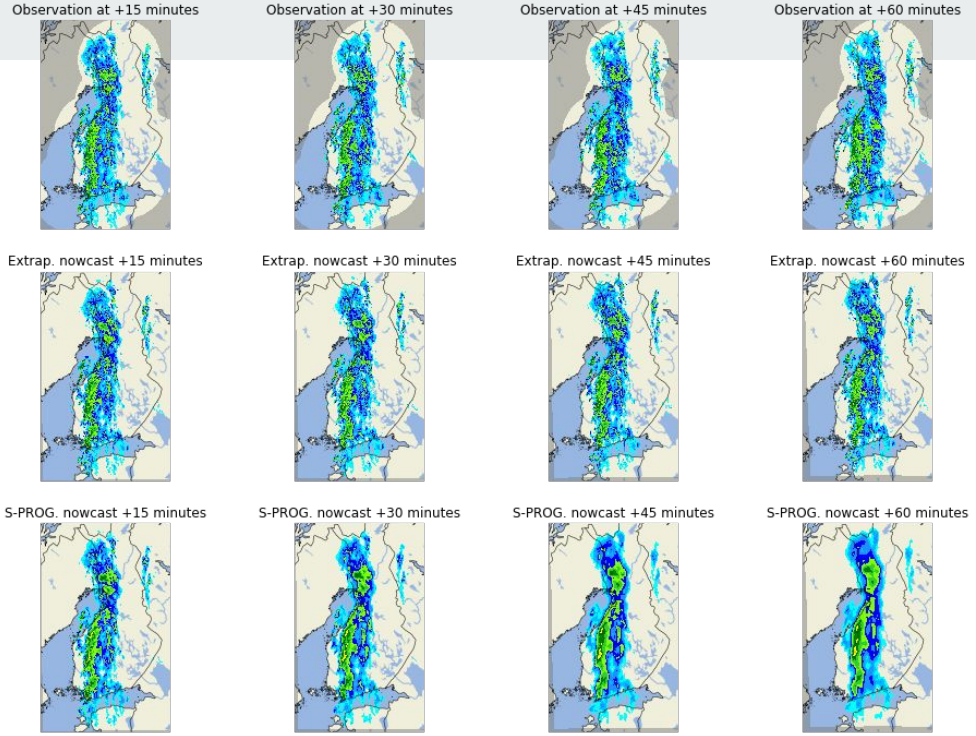
1. Load example data with `helper_nowcasting_methods.ipynb` notebook
2. Create & visualize nowcasts with at least extrapolation method
3. Calculate verification metrics between observations and nowcasts
4. Export the created nowcast to NetCDF file

Deterministic nowcasting methods in pysteps:

- Extrapolation
- ANVIL
- LINDA
- S-PROG

Exercise 4: Deterministic nowcasting





Remember to check parameter values for each nowcasting method
(default values not necessarily good for all datasets)!

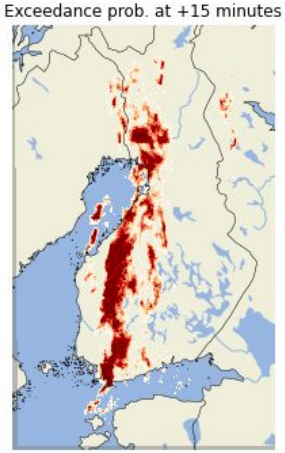
Exercise 4: Probabilistic nowcasting

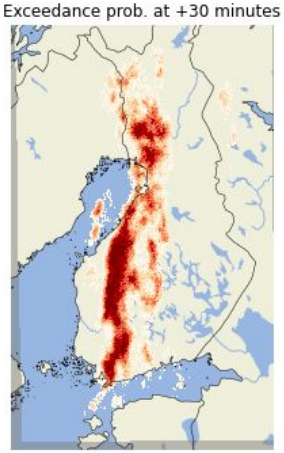
1. Load example data with `helper_nowcasting_methods.ipynb` notebook
2. Create an ensemble STEPS nowcasts with 20 ensemble members and 12 lead times (+1 h)
3. Visualize output, ensemble members and probabilities
4. Calculate probabilistic verification metrics between observations and nowcasts
5. If time allows: Try out a LINDA-P nowcast

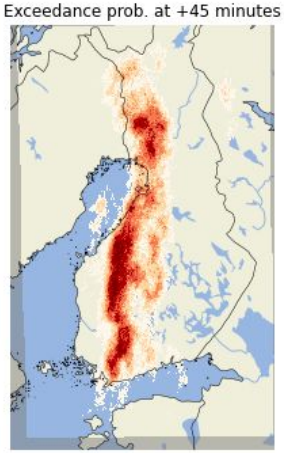
Probabilistic nowcasting methods in pysteps:

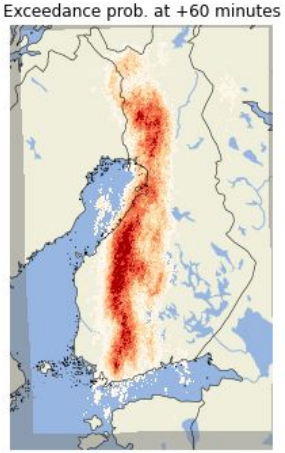
- Lagrangian probability
- STEPS
- LINDA-P

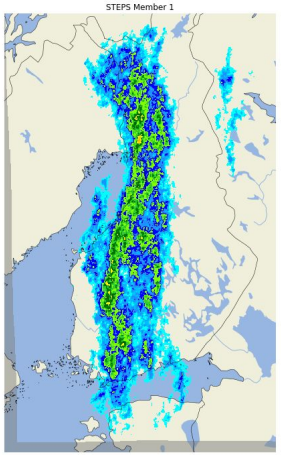
Exercise 4: Probabilistic nowcasting

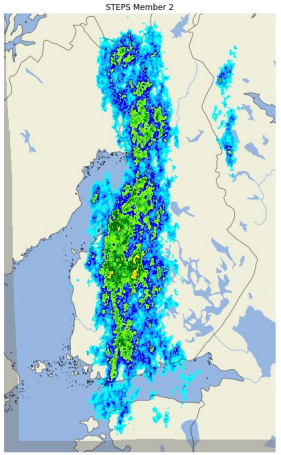


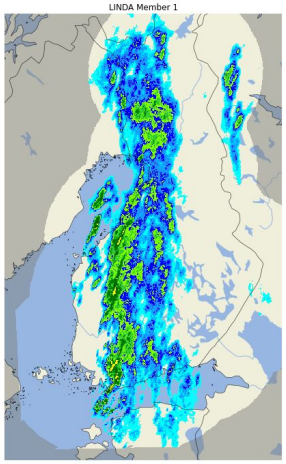


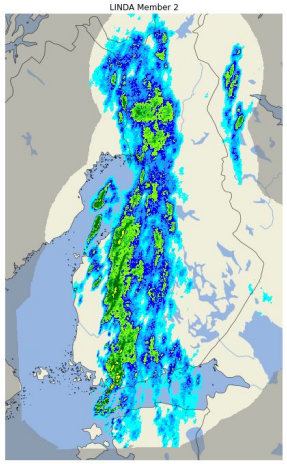








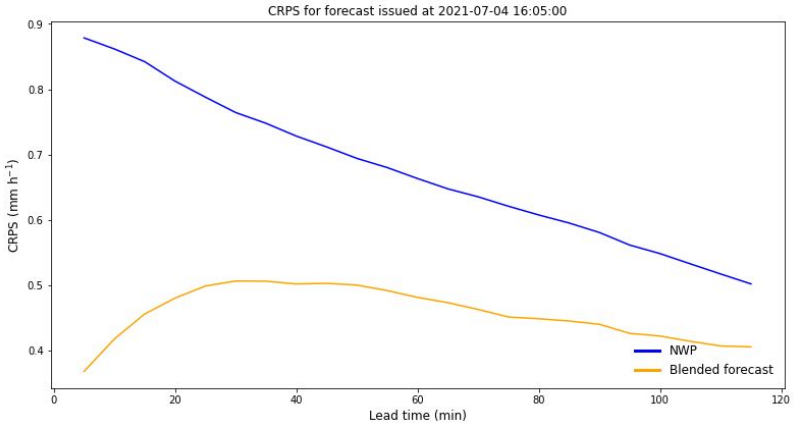




Exercise 5: Blending with NWP

Make sure to use *helper_blending.ipynb* as starting point for this exercise.

Try out a linearly, saliency-based and STEPS blended nowcast. Feel free to play around with it!



Linear blending - adjustable settings:

- Start and end time blending
- Nowcasting method
- Probabilistic forecast

Saliency-based blending: better preservation of high-intensity rainfall cells during blending

STEPS blending - other adjustable settings:

- Weights determination method
- Skill window size
- Option to store decomposed NWP forecast
- Blend ensemble nowcast with ensemble NWP