OpenSense Training School
OS-based nowcasting with pysteps

Introduction to the nowcasting training session

Training instructors:
Jenna Ritvanen (Finnish Meteorological Institute) and Ruben Imhoff (Deltares)

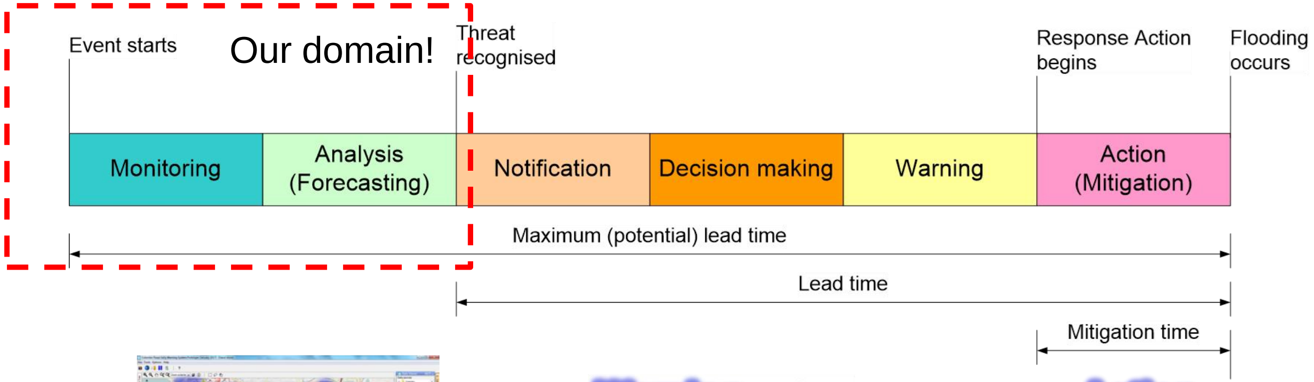
Outline

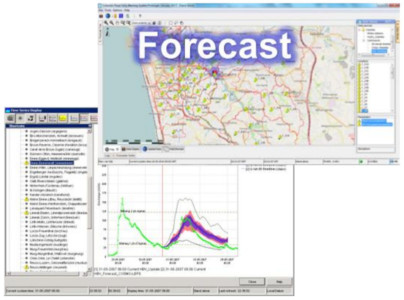
- When and why to use nowcasting
- Introduction to nowcasting with pysteps
- Nowcasting with OS data
- Setup of this training session

Goal of this training session: To give a “hands-on” experience of using pysteps with conventional radar and OS data

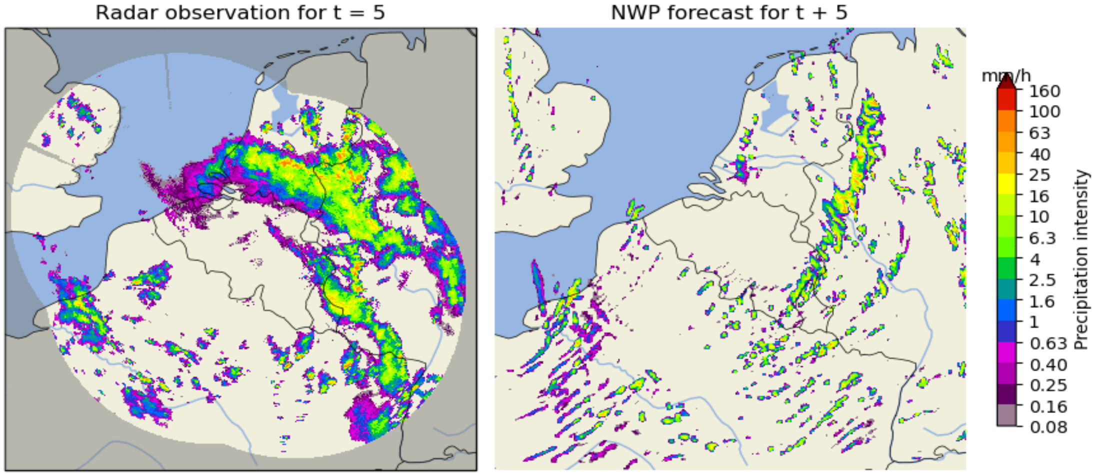
When and why to use nowcasting

Forecasts, warning and action

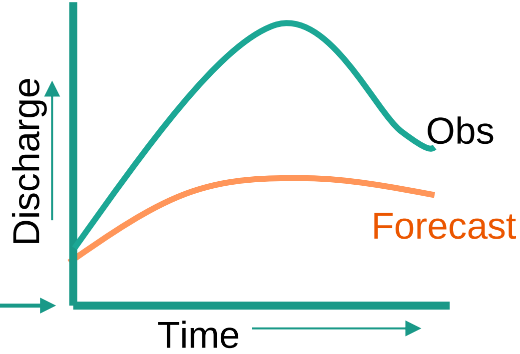




For short lead times Numerical Weather Prediction models are generally not sufficient



Effect on discharge forecast

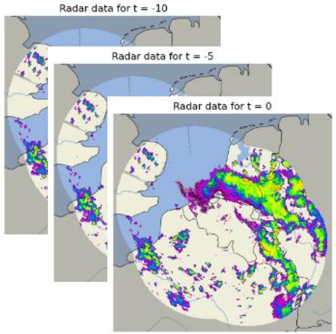


Nowcasting as a fast alternative for the short lead

Statistical extrapolation

Fast!

But no physics involved..

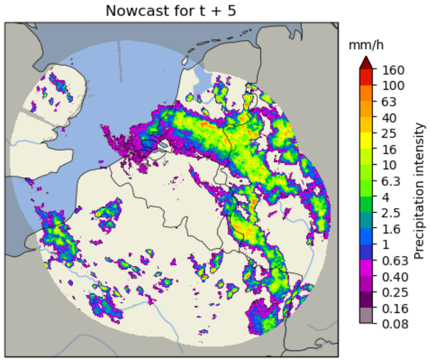


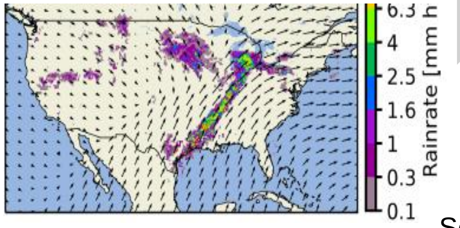
Determine movement of rainfall fields

Rainfall field evolution

Add stochastic noise

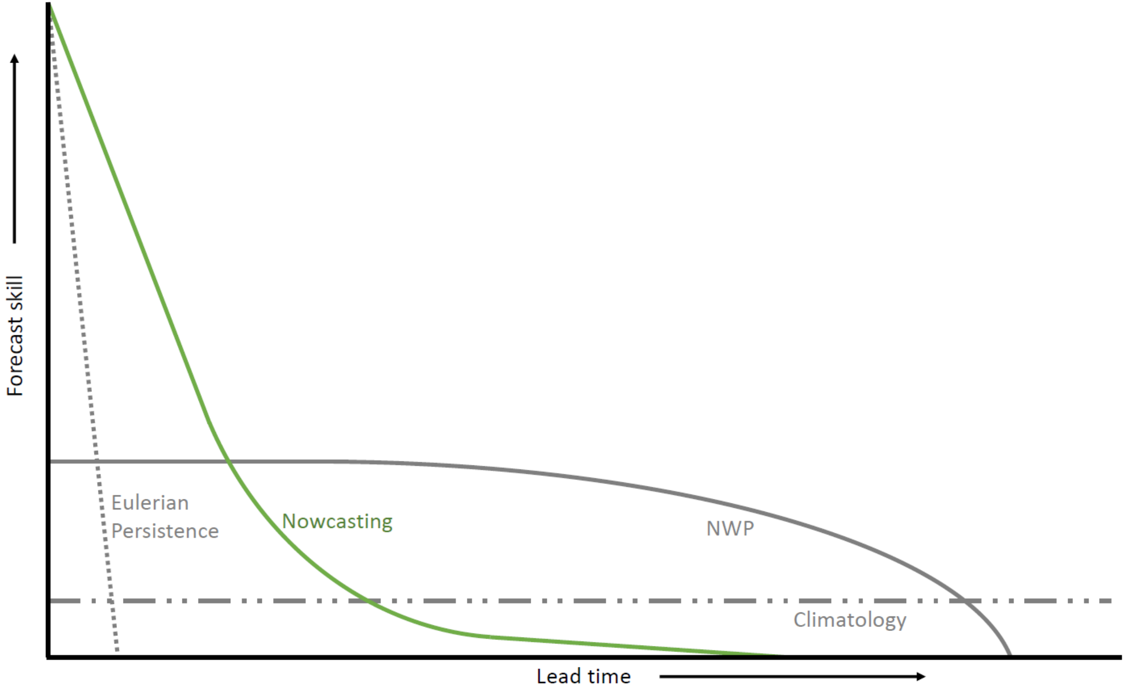
Extrapolate to future and <u>blend</u>



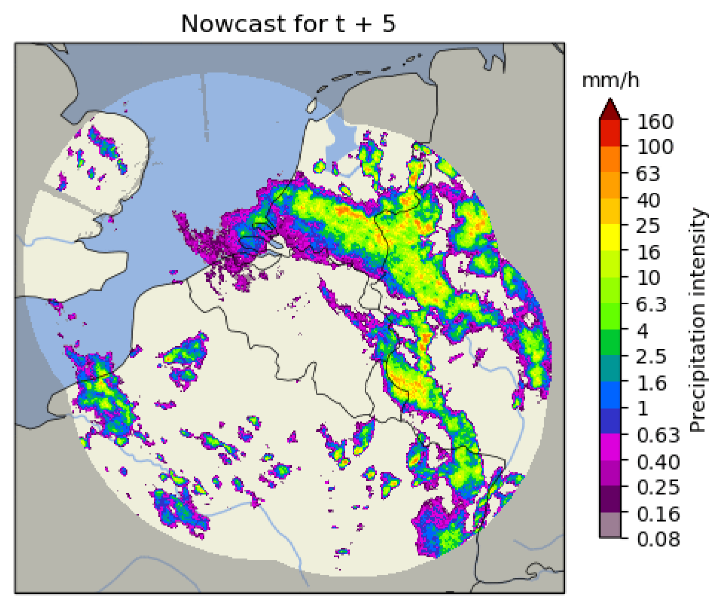


Source: Pulkkinen et al., GMD, 2019

The skill of nowcasting



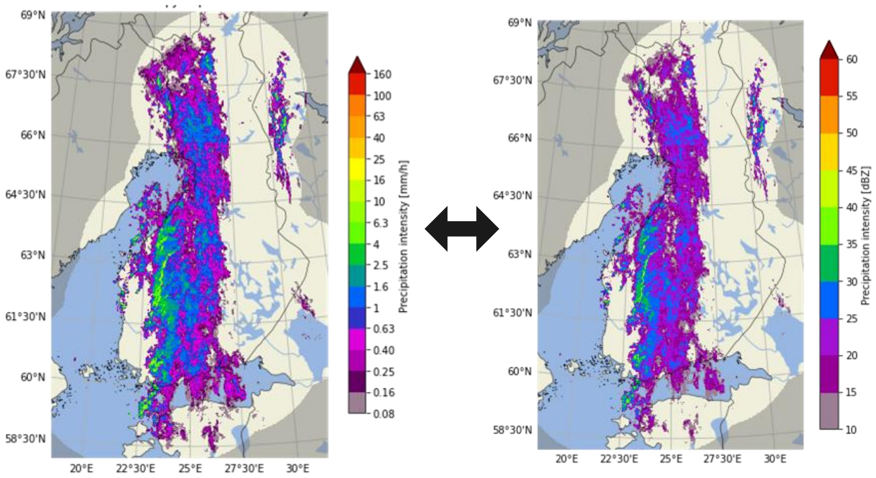
Introduction to nowcasting with pysteps



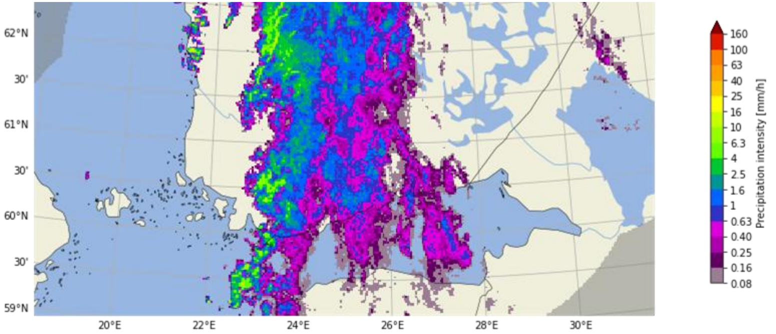
Input data

- pysteps uses gridded data as the main input data source
  - Often this consists of radar rainfall data.
- Reading input data from various sources has been implemented in the **io.importers** module
- To implement your own importers, you can use cookiecutter: https://pysteps.readthedocs.io/en/stable/developer_guide/importer_plugins.html or simply wrap it in an xarray reader.
- The **utils** module contains different methods for converting, transforming, clipping and upsampling the input data
  - Helps you, but also the motion fields and noise generation in pysteps!

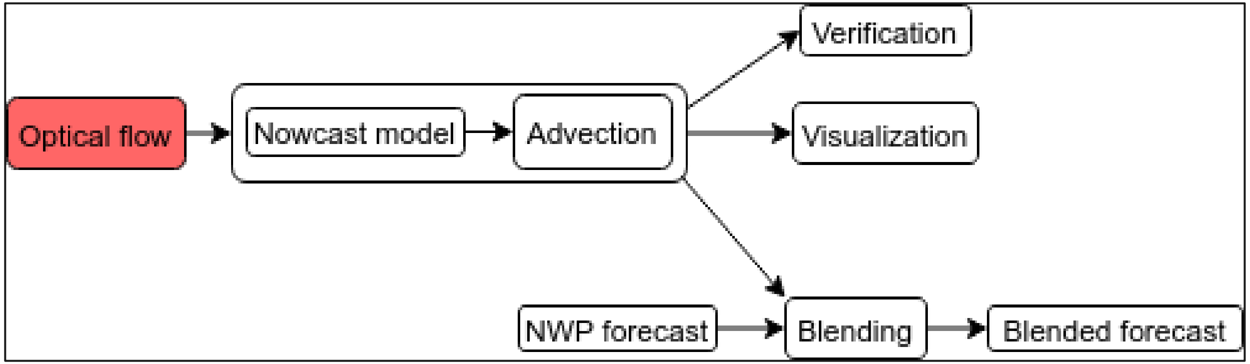
Transformations and conversions between units

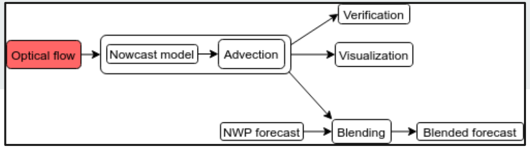


Clipping



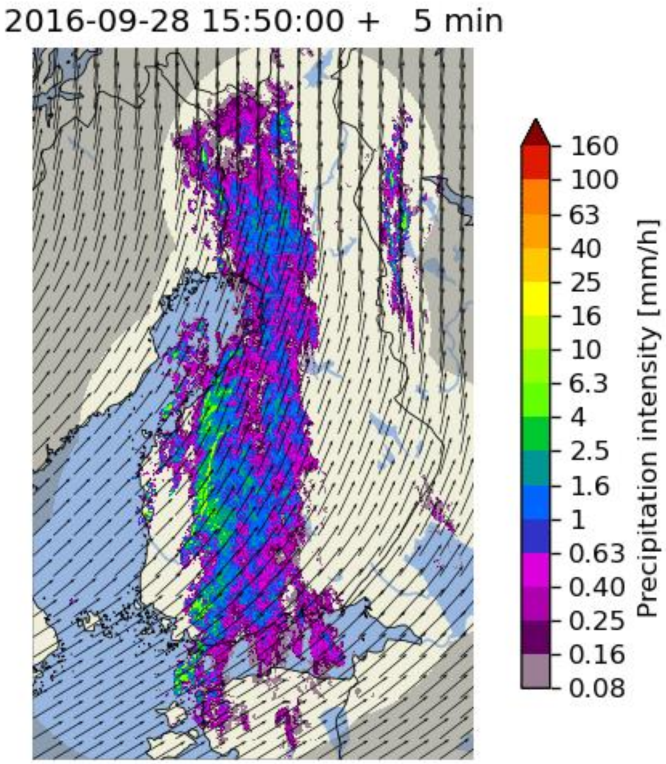
A typical workflow when using pysteps

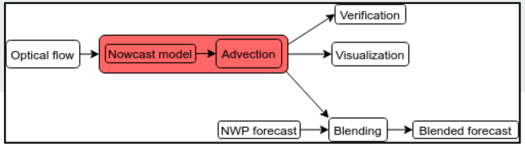




Optical Flow and Extrapolation

- Advection (optical flow & extrapolation) is the key component of all pysteps nowcasting methods
- All methods are based on the "Lagrangian persistence" nowcast shown on the right
- Three different types of optical flow methods have been implemented in the **motion** module:
  - feature tracking: Lucas-Kanade
  - variational: VET and Proesmans
  - spectral: DARTS
- For advection, pysteps implements the backward semi-Lagrangian scheme in the **extrapolation** module

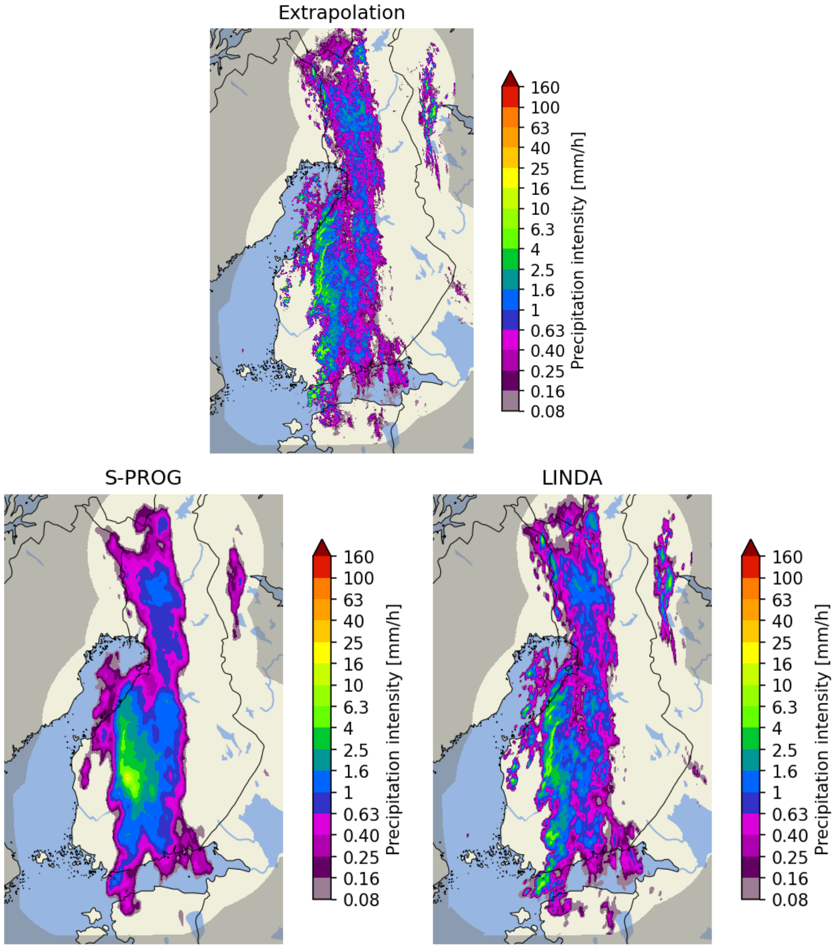




Deterministic Nowcasts

The main methods implemented in the **nowcasts** module:

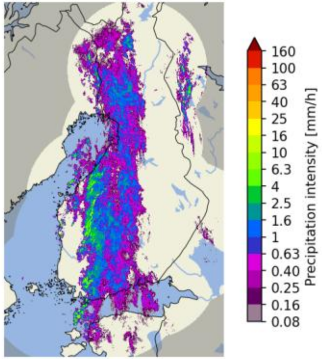
Method	Pros	Cons	Typical computation time
Extrapolation	very fast	no prediction of growth or decay of precipitation	< 10 seconds
S-PROG	• for low-intensity precipitation (< 1-2 mm/h) has generally the best skill • choose for stratiform events	inability to preserve the spatial structure of rainfall fields, and particularly convective cells	< 20 seconds
LINDA-D	• the most accurate method for intense precipitation (> 1-2 mm/h) • choose for convective events	slow to compute	might take several minutes



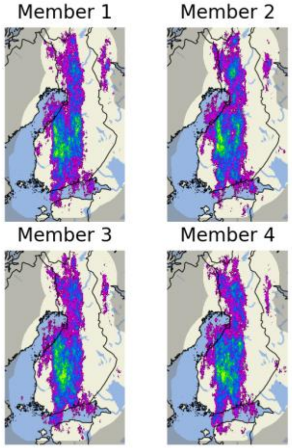
Ensemble Nowcasts

- The main ensemble methods implemented in the **nowcasts** module are STEPS and LINDA-P
- They model two sources of uncertainty: advection field estimation and Lagrangian growth and decay
- The basic rule for choosing the method:
  - stratiform events: STEPS
  - convective events: LINDA-P
- LINDA-P generally produces more realistic ensemble members
- Computation times for the 4-member ensembles shown on the right:
  - STEPS: ~20 seconds
  - LINDA-P: ~5 minutes

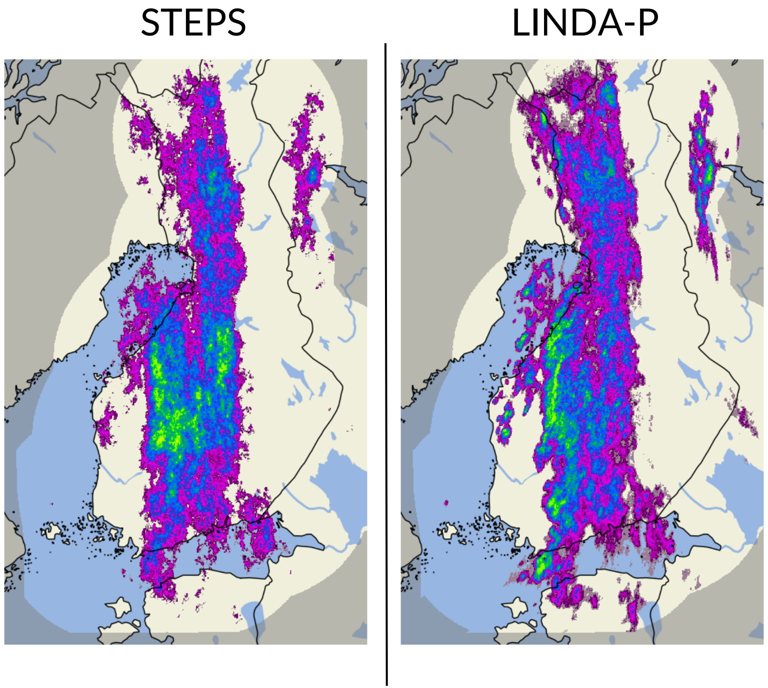
Observations at 2016-09-28 15:50 UTC

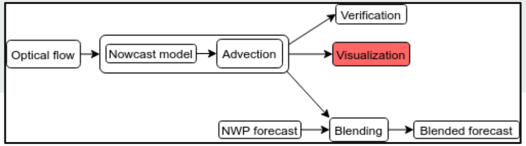


Nowcast ensemble



First ensemble members



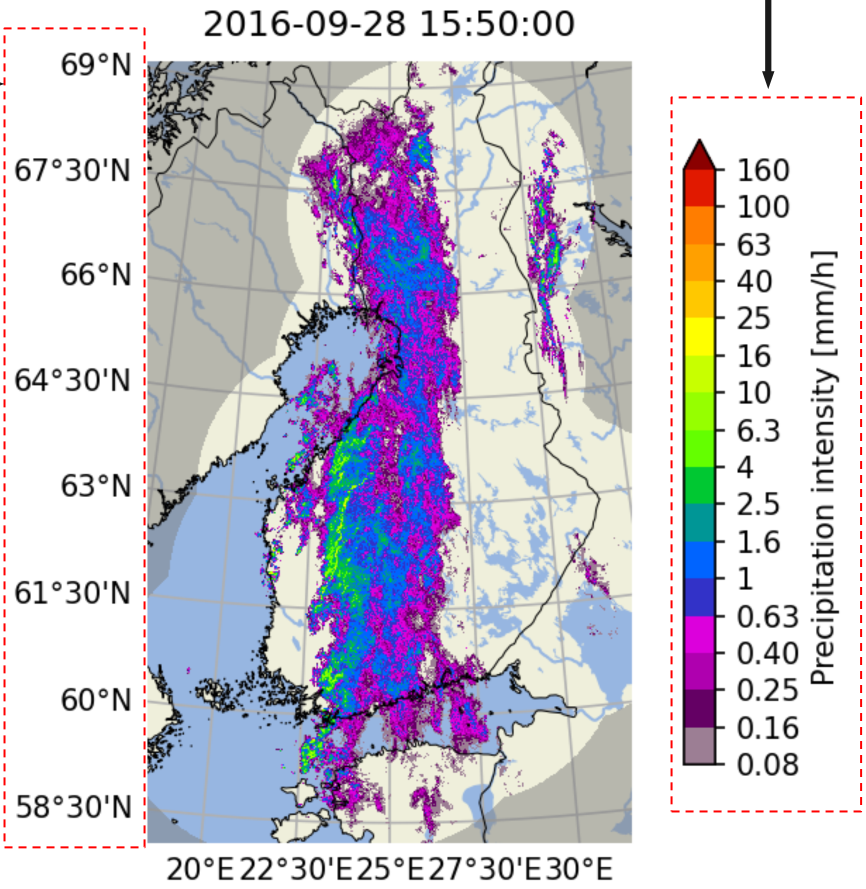


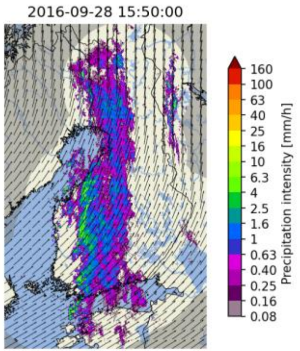
Colorbars with several pre-configured scales and for different data units

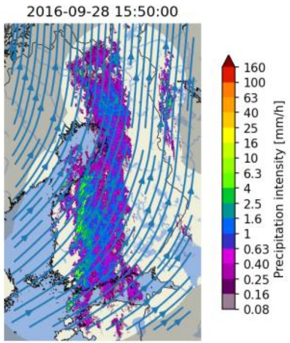
Longitude-latitude lines with labels

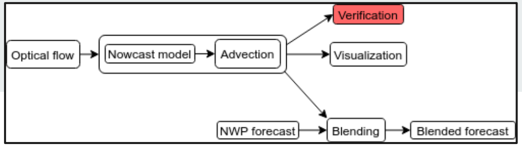
Visualization tools

- Extensive set of visualization tools has been implemented in the **visualization** module
- Support for multiple layers: basemap, precipitation and motion field:
    - plotting of basemaps by using cartopy
    - quivers and streamlines for advection fields









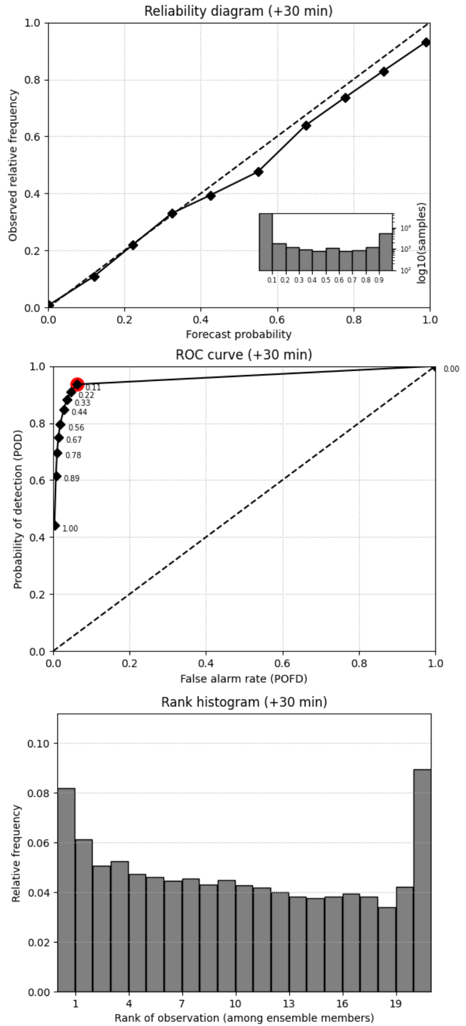
Verification tools & metrics

- A large number of verification utilities and metrics have been implemented in the **verification** module
- Functionality
  - creation of verification objects and aggregation from multiple nowcasts
  - plotting of verification results

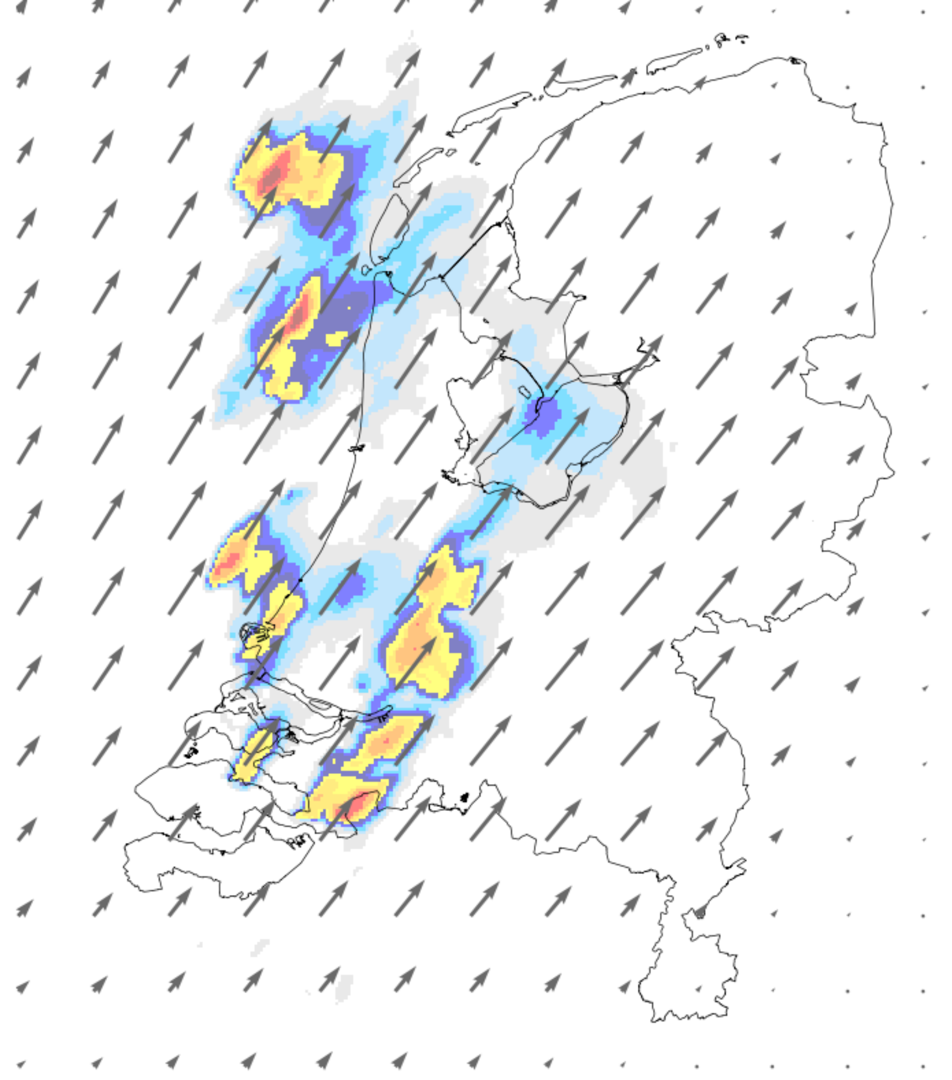
Metrics

- Deterministic
  - categorical: CSI, ETS, POD, FAR
  - continuous: MAE, ME
  - scale/intensity-based metrics: FSS, intensity-scale
  - radially averaged power spectral density (RAPSD)
- Probabilistic
  - CRPS
  - reliability diagram
- Ensemble
  - spread
  - rank histogram

Examples of verification plots for 30-minute STEPS nowcasts

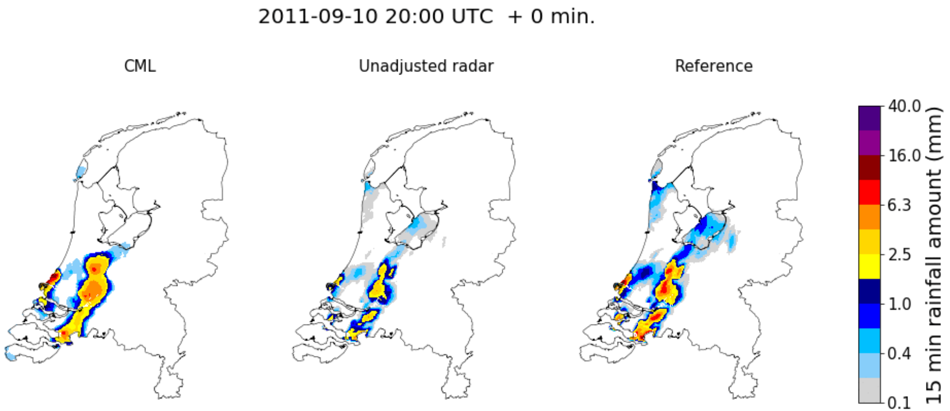


Nowcasting with OS data

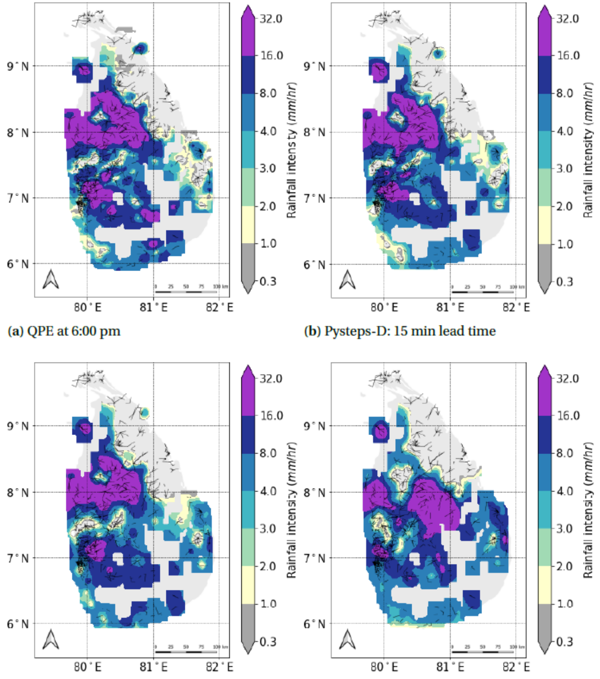


Rainfall nowcasting with alternative sensors

Schoenmaker, 2022



Imhoff et al., GRL, 2020



(a) QPE at 6:00 pm

(b) Pysteps-D: 15 min lead time

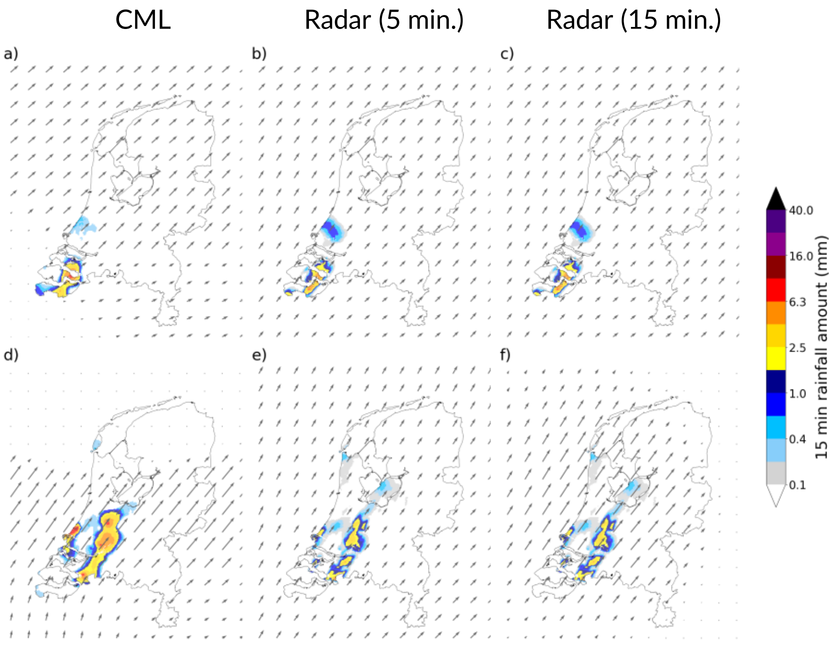
(c) Pysteps-D: 30 min lead time

(d) Pysteps-D: 60 min lead time

Schoenmaker, 2022 and current work of Bas Walraven (TU Delft)

Challenges when nowcasting with OS data

- Advection field derivation
- Data consistency
- Can you think of other possible issues when using OS data for nowcasting?
- Request to you: have a close look at the differences in the optical flow fields, processing steps and resulting nowcasts between the various OS products that you are testing today.



Imhoff et al., GRL, 2020

Setup of this training session

Hands-on experience through four exercises

Exercises:

1. Check pysteps installation and download example data
2. Read, visualize and process input data with pysteps
3. Optical flow and extrapolation
   a) Optional: Advection correction of the gridded rainfall fields for accumulation.
4. Creating your first nowcasts with the OS datasets
   a) Optional: Creating a probabilistic (ensemble) nowcast with the same datasets

Setup of the training material

Nowcasting-session folder

- [exercises]
- [notebooks]
- [presentations]
- environment
- README

Setup of the training material

Nowcasting-session folder

- [exercises]
- [notebooks]
- [presentations]
- environment
- README

This folder contains the description of each exercise

exercise_01_local_conda_setup	md
exercise_02_input_data	md
exercise_03_optical_flow_and_extrapolation	md
exercise_03a_advection_interpolation	md
exercise_04_deterministic_nowcasting	md
exercise_04a_probabilistic_nowcasting	md

Setup of the training material

Nowcasting-session folder

- [exercises]
- [notebooks]
- [presentations]
- environment
- README

These folders contain the notebooks in which you can work on the exercises (exercise_notebooks, which include some hints to get there) and the notebooks with the solutions.

- [exercise_notebooks]
- [solutions]

block_01_local_conda_setup	ipynb
block_02_input_data	ipynb
block_03_optical_flow_and_extrapolation	ipynb
block_03a_advection_interpolation	ipynb
block_04_deterministic_nowcasts	ipynb
block_04a_probabilistic_nowcasts	ipynb

Setup of the training material

Nowcasting-session folder

[exercises]
[notebooks]
[presentations]
environment
README

Finally, all presentations of this session can be found in this folder

Time to try it out yourself!

Schedule:

~~- Introduction to the session (25 minutes)~~

- Hands on: Getting started with pysteps, read, visualize and process the radar and opportunistic sensing data. (20 minutes)
- Hands on: Optical flow and extrapolation, the base of advection-based nowcasting. (30 minutes)
- Wrap up of the first hands-on sessions. (10 minutes)
- Hands on: Your first nowcasts, using and verifying radar and opportunistic sensing data for nowcasting. Additional, if time allows, creation of ensemble nowcasts. (60 minutes)
- Wrap up and closure. (15 minutes)

No to forget, update opensense_training python package:

```
git add *

git commit -m "day2 changes part 1“

git fetch

git merge origin/main
```