Integrating harp verification into Python based visualization tools

Gatis Priedītis

ACCORD
A Consortium for COnvection-scale modelling Research and Development

5th ACCORD All Staff Workshop

01.04.2025

Summary

- Motivation
  - Why choose a different programming language?
  - How to increase interactivity?
  - How to make it more user-friendly?
- Current visualization workflow
  - Existing workflow in R
  - Proposed workflow in Python
- Suggested improvements and further work

General HARP workflow

harpIO -> harpCore -> harpPoint -> harpVis

Input/output package for reading in/writing out various model/observation data

Main harp package for main harp_df and helper functions

Verification package for point data
Results saved as .rds

Visualization package

Shiny app

Proposed HARP workflow

harpIO -> harpCore -> harpPoint -> harpVis

Shiny app

Visualization package

Input/output package for reading in/writing out various model/observation data

Main harp package for main harp_df and helper functions

Verification package for point data
~~Results saved as .rds~~
Results saved as .feather

Python package for reading .feather files and creating plotly based visualizations

Dash visualization tool

Pros and cons

Advantages

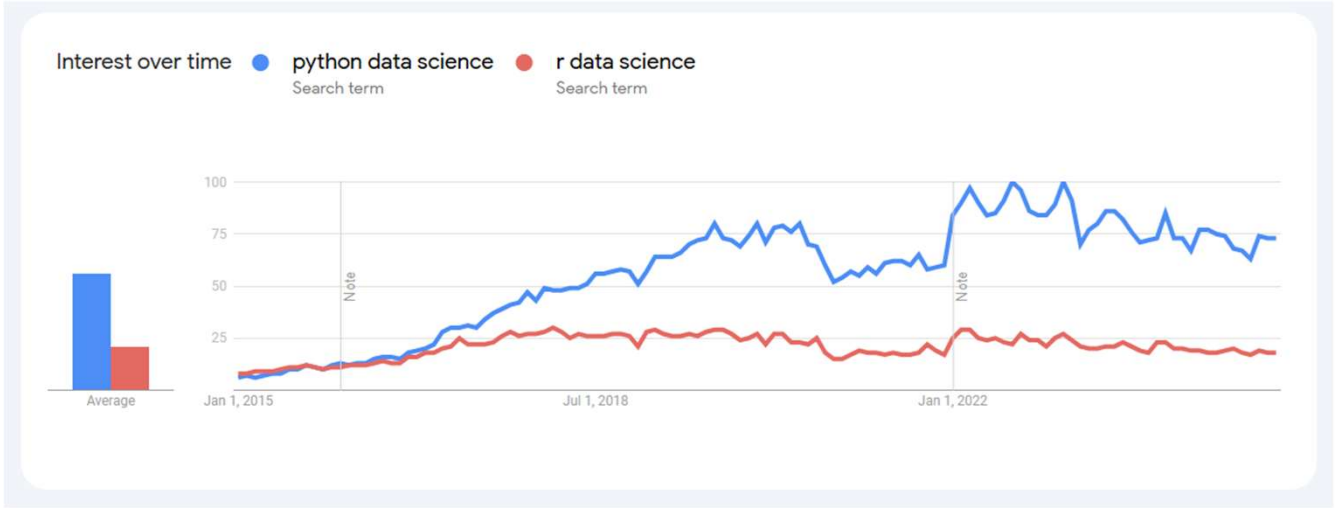
- Interactive graphs
- Highly customizable output of data
- Higher literacy in python over R amongst developers
- R development can be focused on core packages

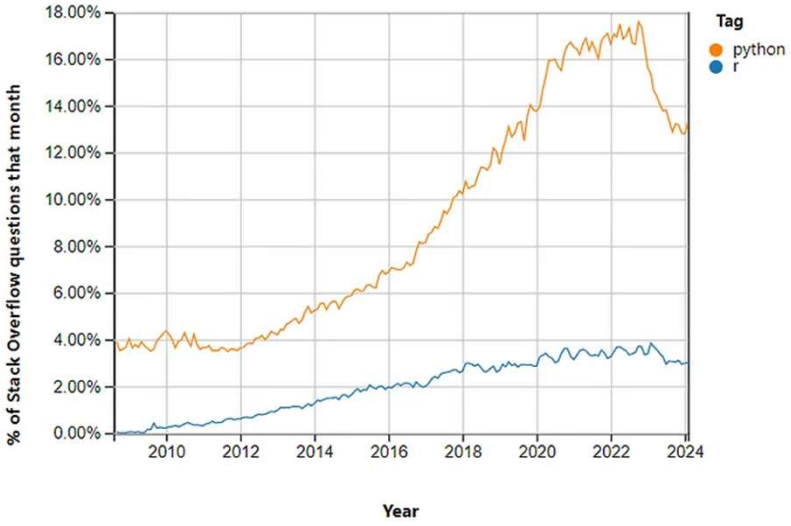
Concerns

- Ability to integrate with current setup
- Learning new technologies
- Duplication of work

Motivation for change

python™





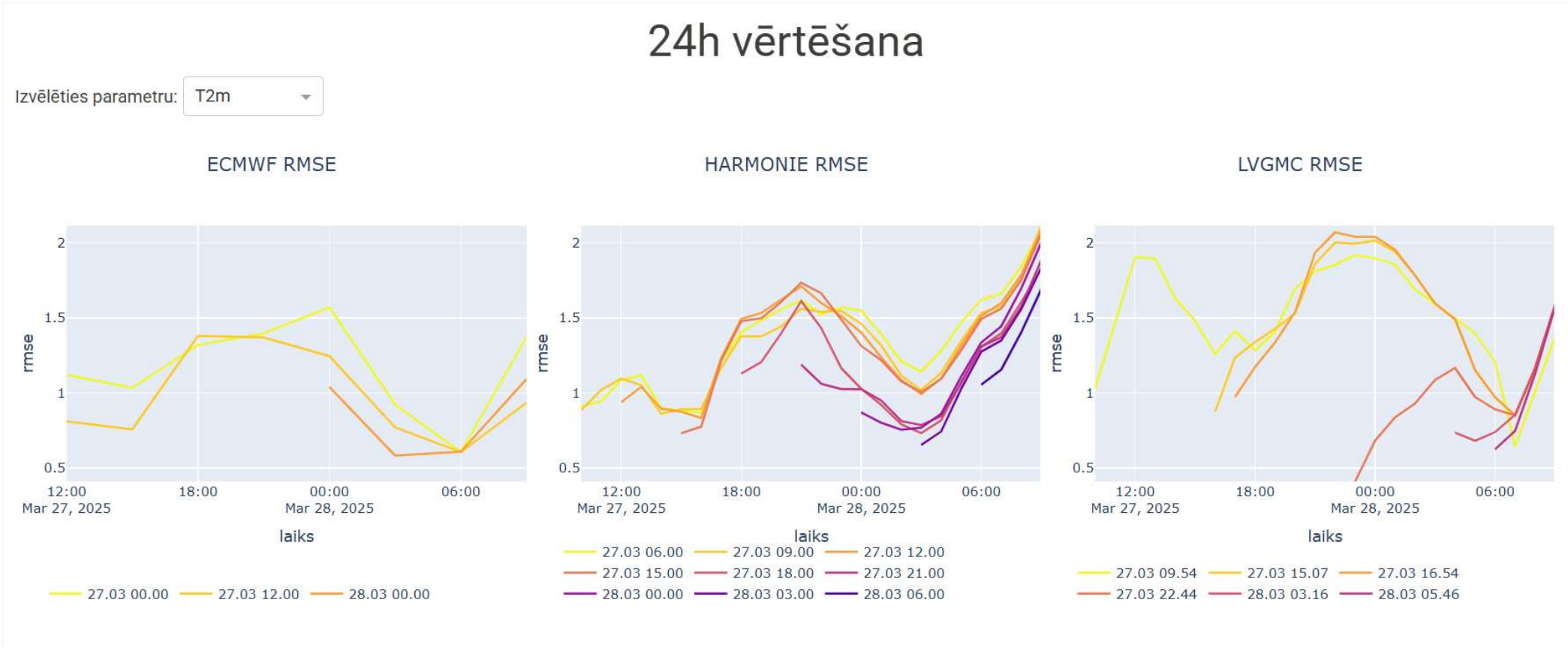
Flask/Dash options in Python

- Python packages
  - Flask - web framework that allows developers to build lightweight web applications quickly and easily
  - Plotly - open-source, and browser-based graphing library. It offers Python-based charting, powered by plotly.js
  - Dash - Web interface library for plotly output, based on flask webserver

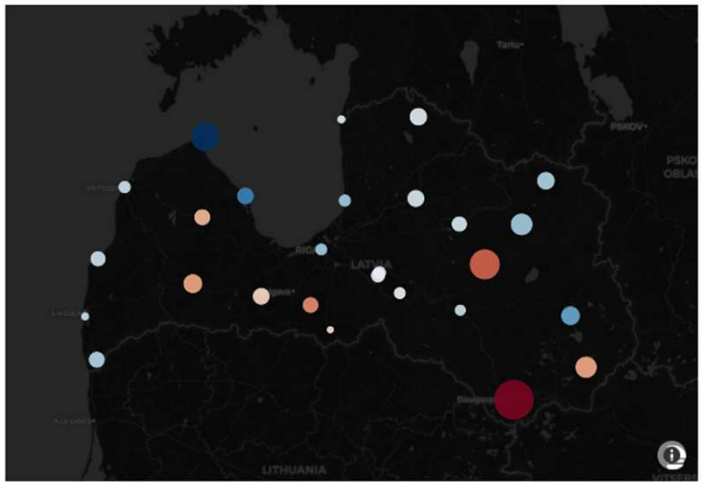
plotly | Dash

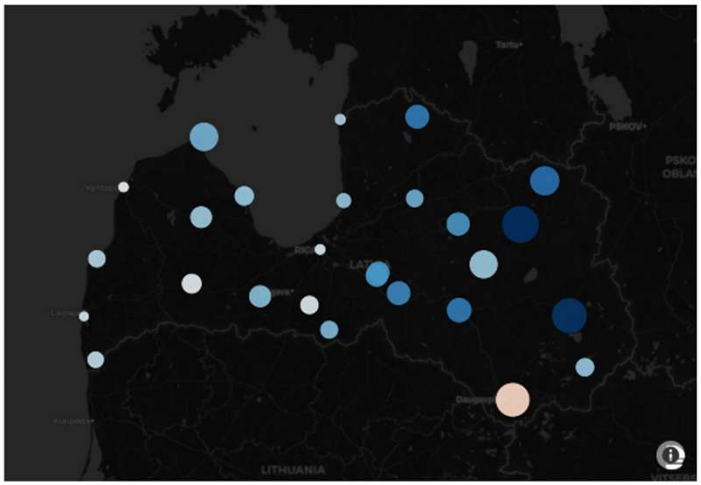
harp shiny app

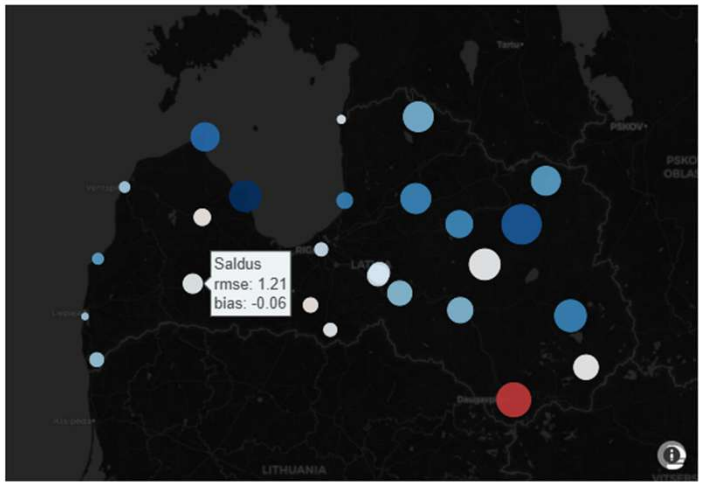
LEGMC 24-hour verification system in Dash



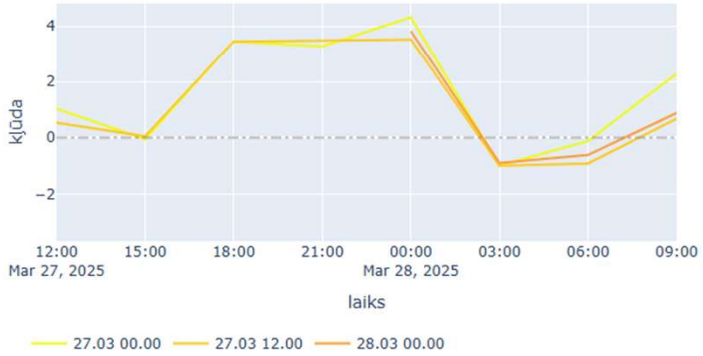
LEGMC point verification by station



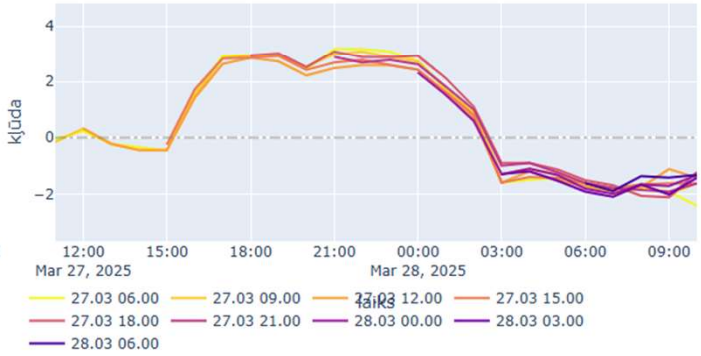




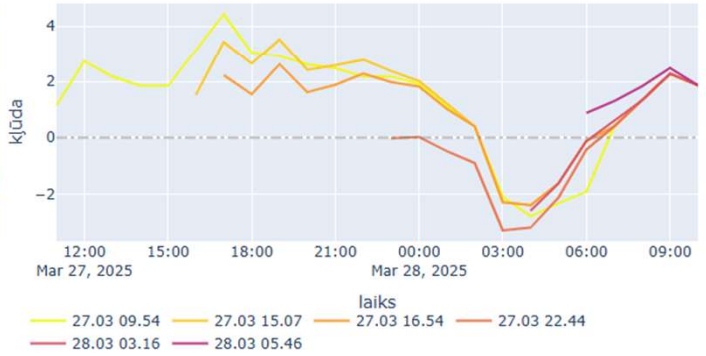
Daugavpils, ecmwf



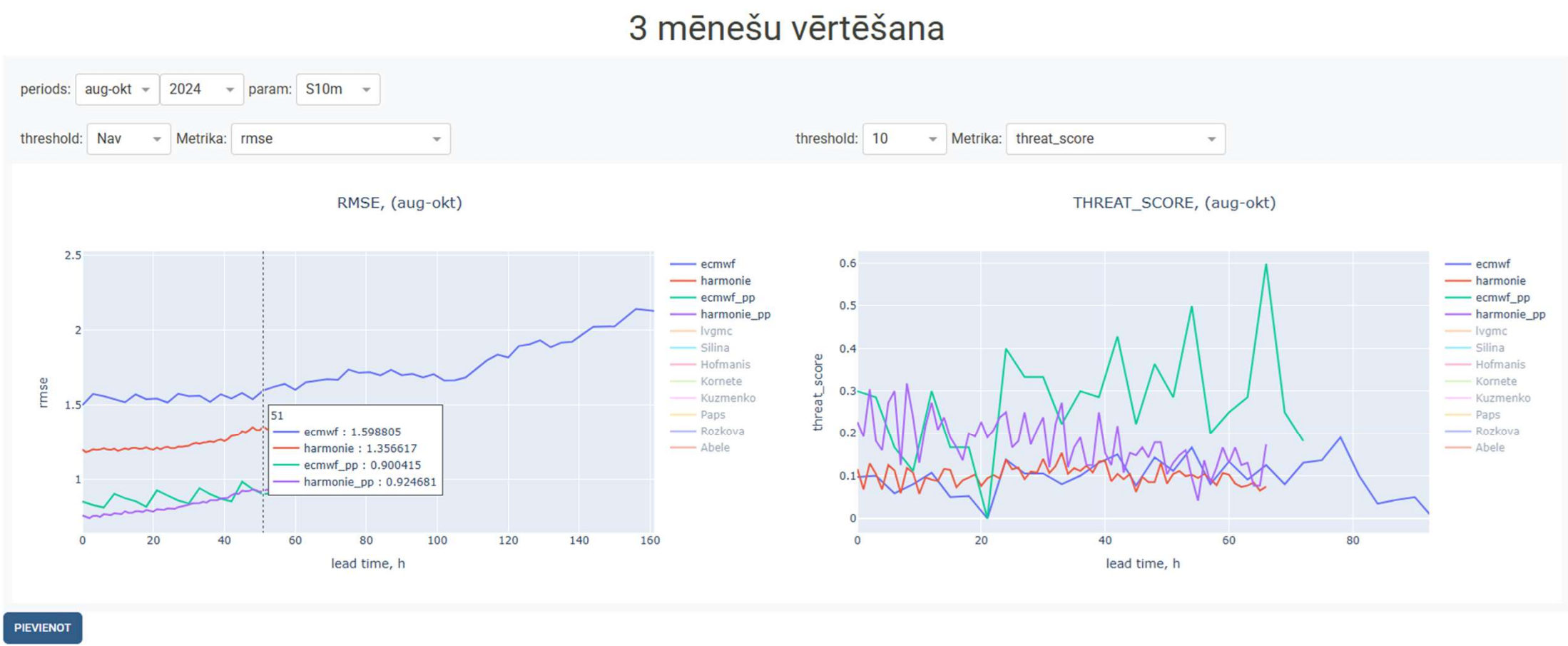
Daugavpils, harmonie



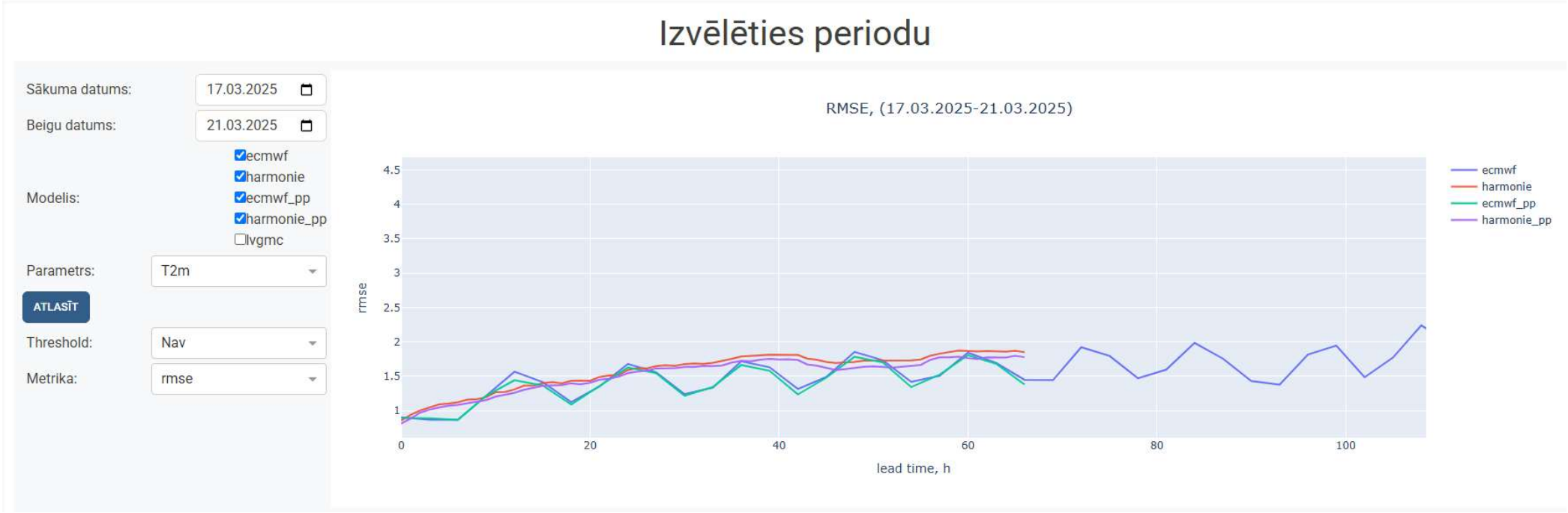
Daugavpils, lvgmc



LEGMC three-month verification system



LEGMC manual selection verification



Progress within ACCORD

- Work has been undertaken within MQA to create a general approach for point verification visualization:

  - Tool must be able to work with any set of *.rds files from verification output
  - Over time, the end user must be able to easily install and set-up the tool
  - Options for probabilistic and spatial verification must be added over time
  - Initial work has used two *.rds sample files from Met Éireann colleagues for surface and upper-air point verification

Overall view of the tool

Date: 2024 12 1 Experiment: surface Parameter: Temperature (T)

Grouping: ◉fcst_model ○station_group ○fcst_cycle ○threshold

X-axis: ◉lead_time ○valid_hour ○valid_dttm ○threshold

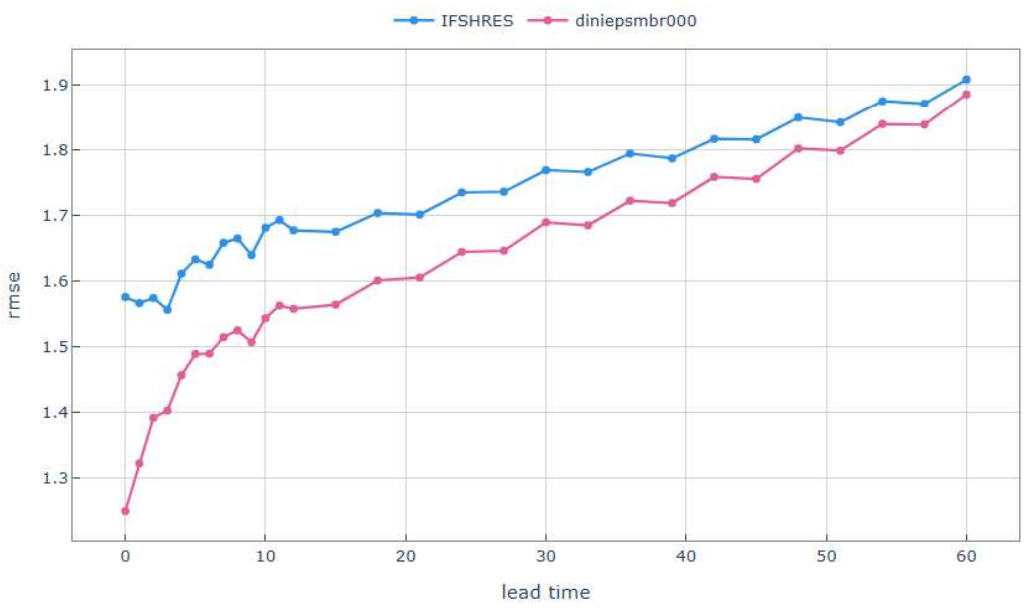
station_group: All

fcst_cycle: All

threshold: None

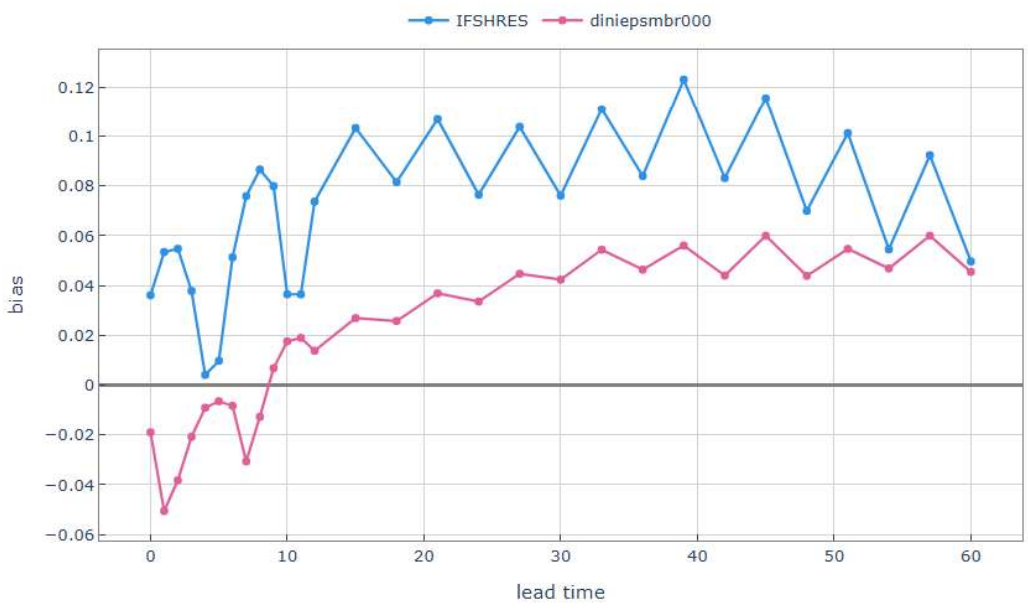
RMSE for Temperature (T)

date: 01.12.2024 **station count:** 1767 **station group:** All **fcst cycle:** All **threshold:** None



BIAS for Temperature (T)

date: 01.12.2024 **station count:** 1767 **station group:** All **fcst cycle:** All **threshold:** None



Various options – valid time

Date: 2024 12 1 Experiment: surface Parameter: Temperature (T)

Grouping: ◉fcst_model ○station_group ○fcst_cycle ○threshold

X-axis: ○lead_time ○valid_hour ◉valid_dttm ○threshold

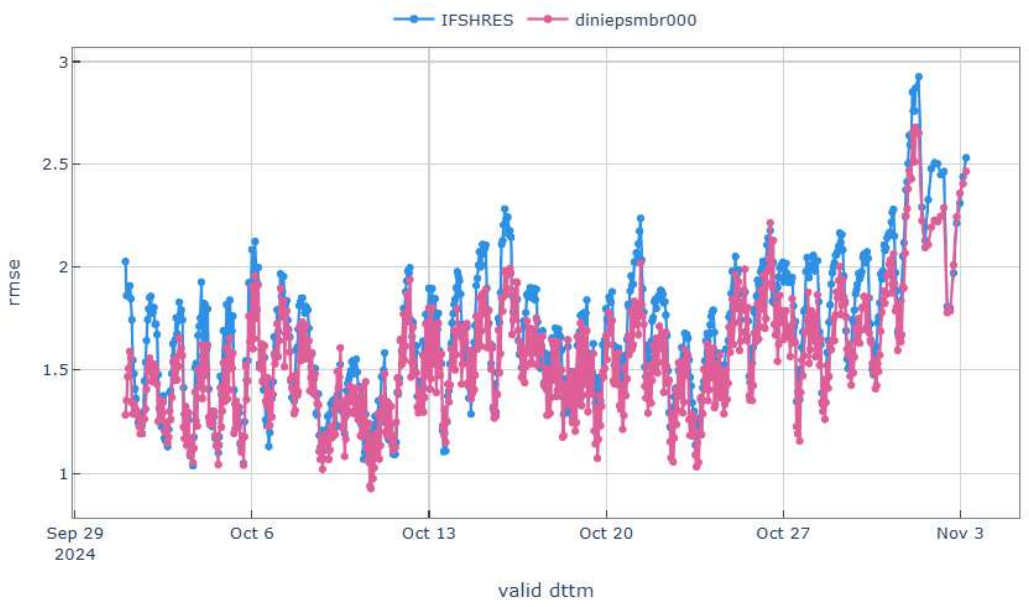
station_group: All

fcst_cycle: All

threshold: None

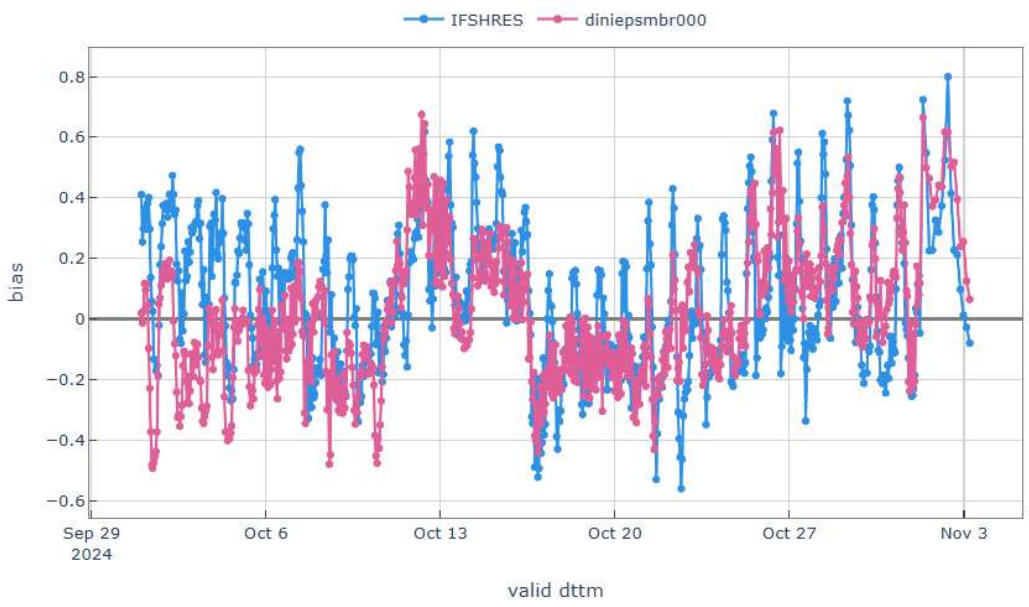
RMSE for Temperature (T)

date: 01.12.2024 **station count:** 1767 **station group:** All **fcst cycle:** All **threshold:** None



BIAS for Temperature (T)

date: 01.12.2024 **station count:** 1767 **station group:** All **fcst cycle:** All **threshold:** None



Various options – vertical levels

Date: 2024 12 1 Experiment: upper Parameter: Temperature (T)

Grouping: fcst_model, station_group, fcst_cycle, p

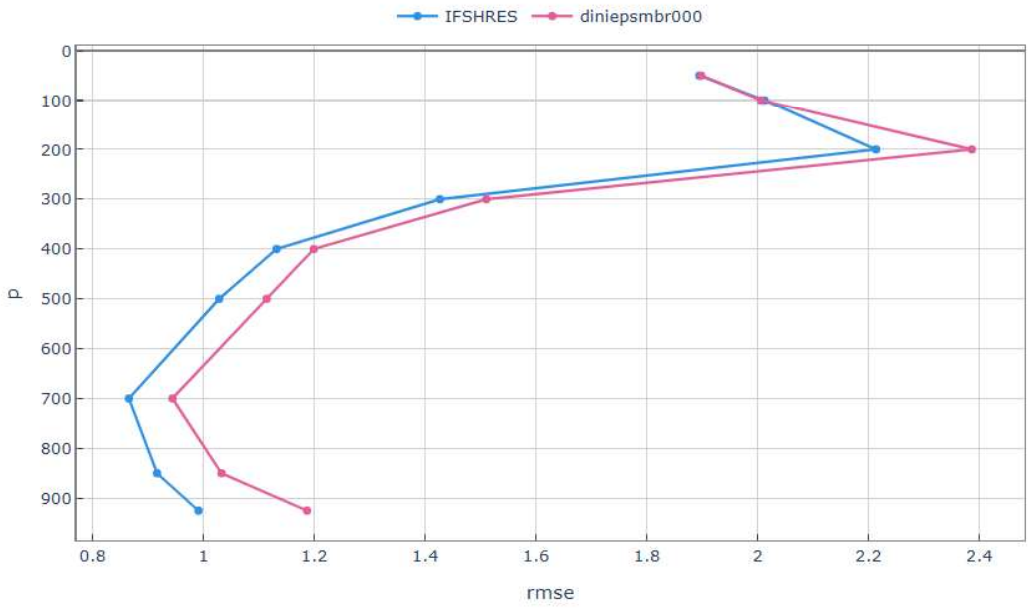
X-axis: lead_time, valid_hour, p

station_group: All

fcst_cycle: All

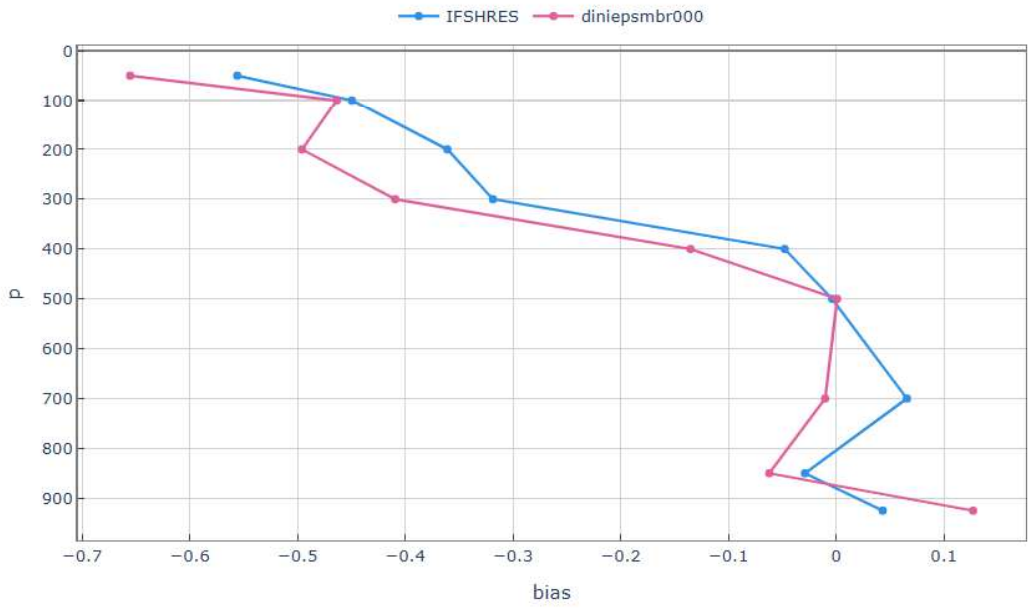
RMSE for Temperature (T)

date: 01.12.2024 **station count:** 9 **station group:** All **fcst cycle:** All



BIAS for Temperature (T)

date: 01.12.2024 **station count:** 9 **station group:** All **fcst cycle:** All



Different grouping and x-axis combinations

Date: 2024 12 1 Experiment: surface Parameter: Temperature (T)

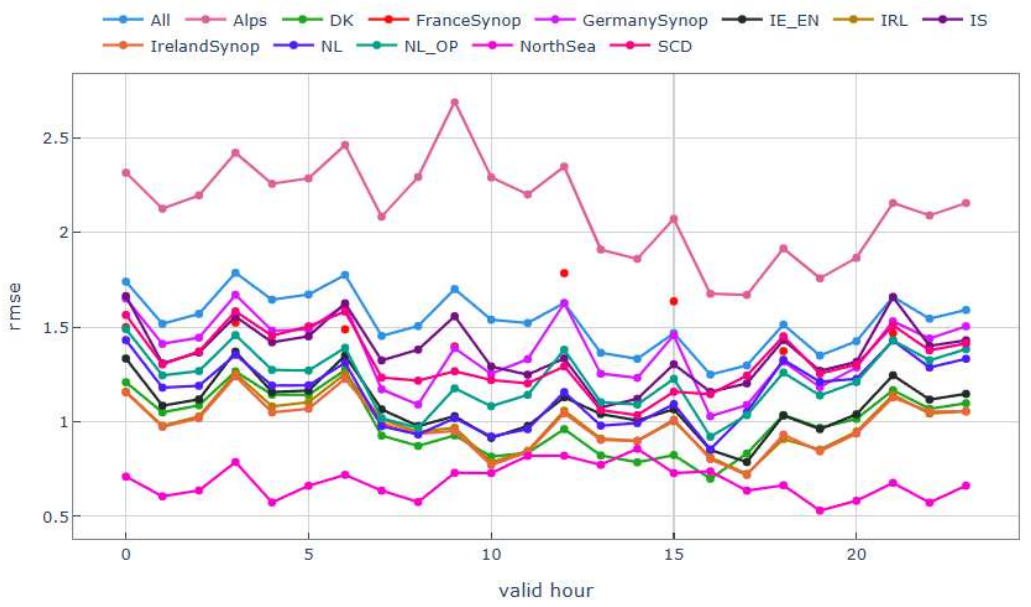
Grouping: ○fcst_model ◉station_group ○fcst_cycle ○threshold

X-axis: ○lead_time ◉valid_hour ○valid_dttm ○threshold

fcst_model: diniepsmbr0... fcst_cycle: All threshold: None

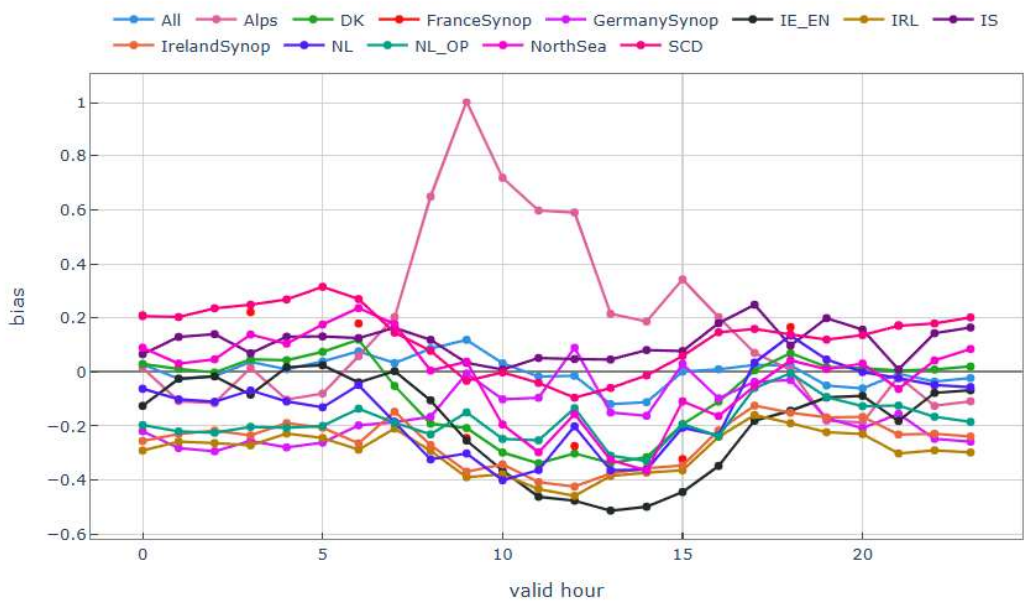
RMSE for Temperature (T)

date: 01.12.2024 **station count:** 83 **fcst model:** diniepsmbr000 **fcst cycle:** All **threshold:** None



BIAS for Temperature (T)

date: 01.12.2024 **station count:** 83 **fcst model:** diniepsmbr000 **fcst cycle:** All **threshold:** None



Inclusion of thresholds

Date: 2024 12 1 Experiment: surface Parameter: Temperature (T)

Grouping: ○fcst_model ◉station_group ○fcst_cycle ○threshold

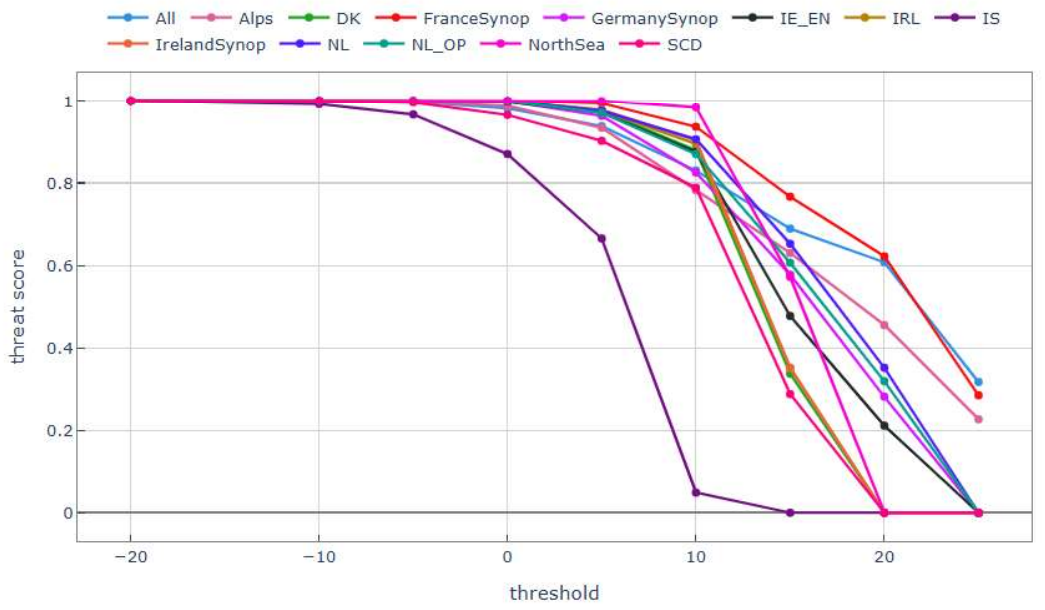
X-axis: ○lead_time ○valid_hour ○valid_dttm ◉threshold

fcst_model: diniepsmbr0...

fcst_cycle: All

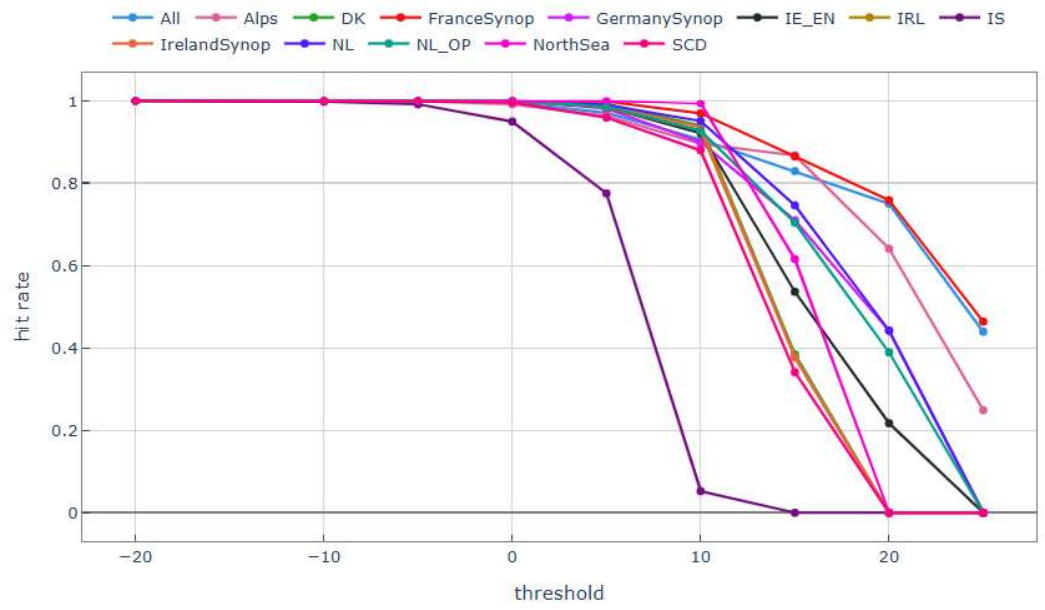
THREAT SCORE for Temperature (T)

date: 01.12.2024 **station count:** 83 **fcst model:** diniepsmbr000 **fcst cycle:** All



HIT RATE for Temperature (T)

date: 01.12.2024 **station count:** 83 **fcst model:** diniepsmbr000 **fcst cycle:** All



Interactivity – zoom and data points

Date: 2024 12 1 Experiment: surface Parameter: Temperature (T)

Grouping: fcst_model, station_group, fcst_cycle, threshold

X-axis: lead_time, valid_hour, valid_dttm, threshold

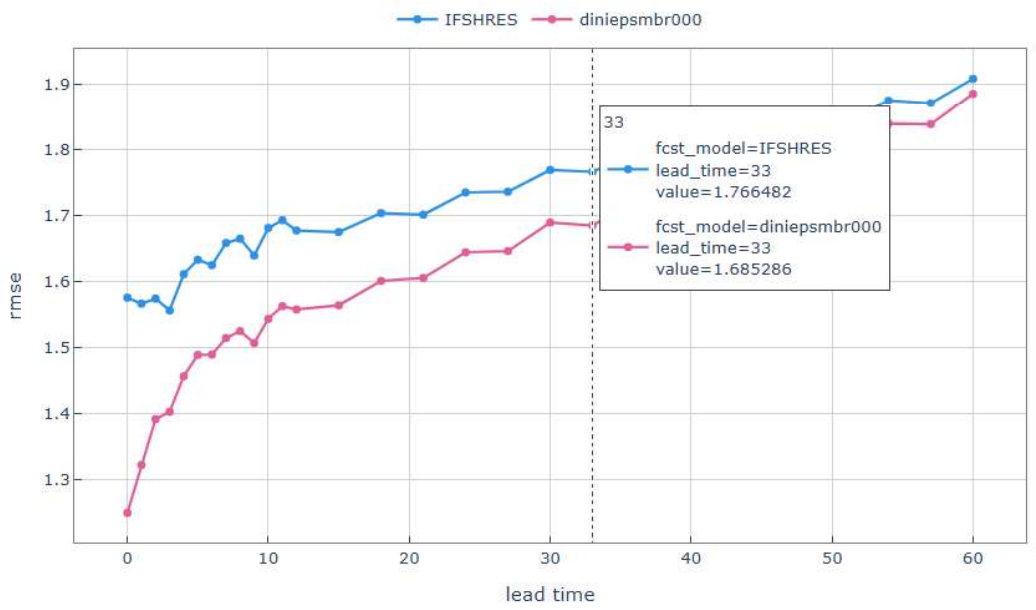
station_group: All

fcst_cycle: All

threshold: None

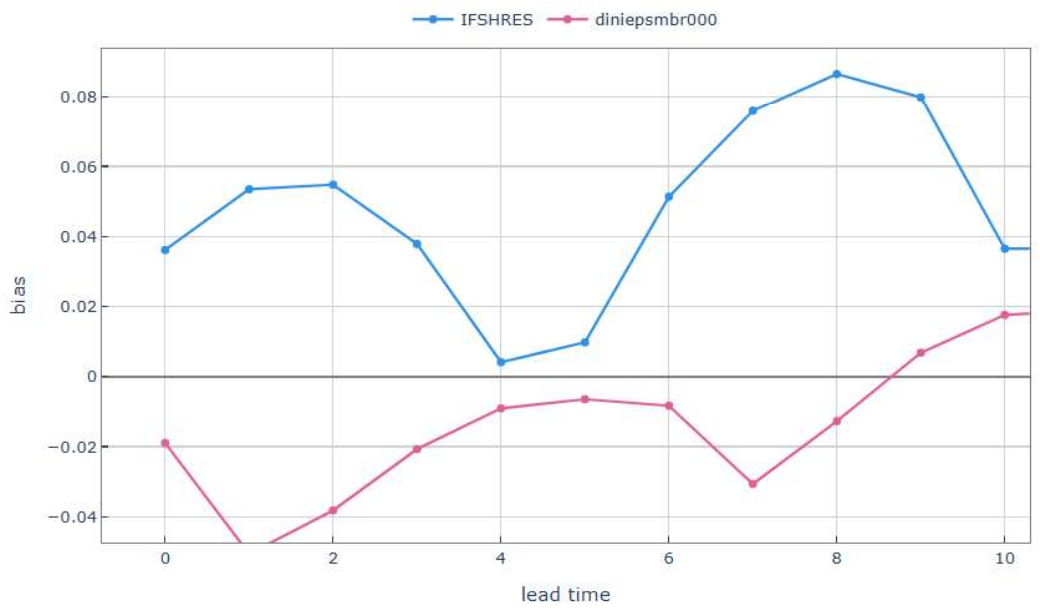
RMSE for Temperature (T)

date: 01.12.2024 **station count:** 1767 **station group:** All **fcst cycle:** All **threshold:** None



BIAS for Temperature (T)

date: 01.12.2024 **station count:** 1767 **station group:** All **fcst cycle:** All **threshold:** None



Future plans and improvements

1. Python module for portability of harpVis functionality
2. Additional options in harp verification output format
3. User options for additional panels, map view and other functions
4. Default dash server configuration and set-up, replacing need for shiny server