A reproducible flood forecasting case study using different machine learning techniques

ESoWC 2019 - Machine learning for predicting extreme weather hazards

Lukas Kugler & Sebastian Lehner

ESoWC-Mentors: Claudia Vitolo, Julia Wagemann, Stephan Siemen

16.10.2019

Motivation & goal

Modeling river discharge => high-dimensional problem: many meteorological/ hydrological parameters important with complex interactions => coupled dynamical models.

- Can statistical connections based on ML techniques with comparable skill be established for extreme flooding events?

➔ Explorative comparison study: investigate flood forecasting capability of ML techniques (with data from ECMWF/Copernicus)

➔ Code and documentation for interested peers (open source and reproducible; https://github.com/esowc/ml_flood)

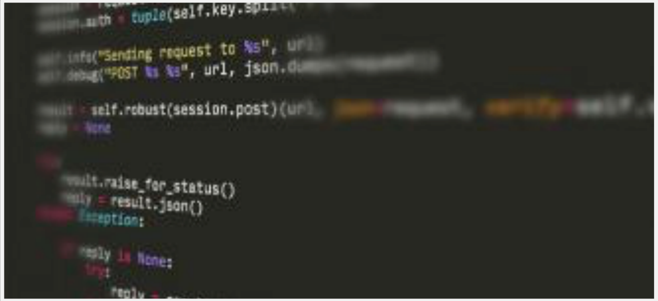
Data

- Predictor data from ERA5
  via the Climate Data Store API for python

- Predictand data from GloFAS v2.0
  - Reanalysis
  - 4 Forecast reruns for a flooding event

  directly from the GloFAS team via FTP

- Daily resolution (preprocessing CDO);
  spatial domain of interest

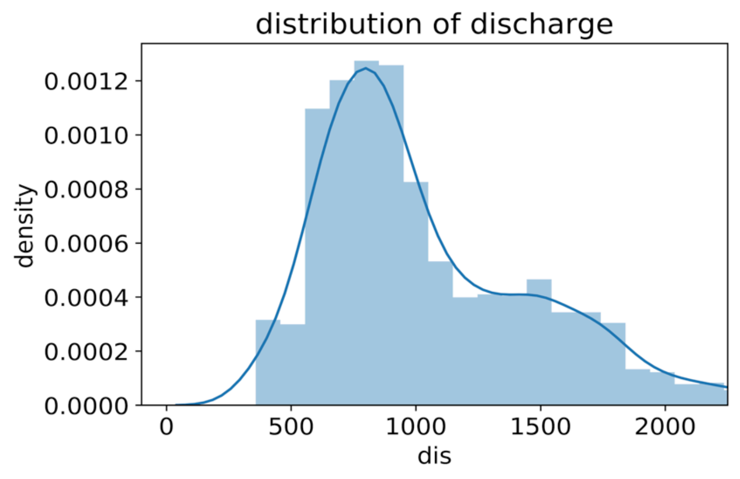


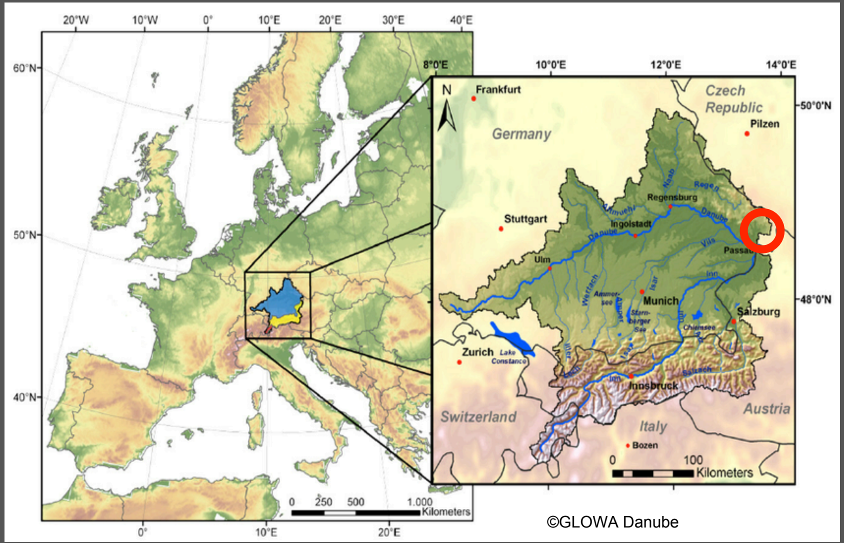
Climate Data Store **API**

Notebook: 1.03_data_overview.ipynb

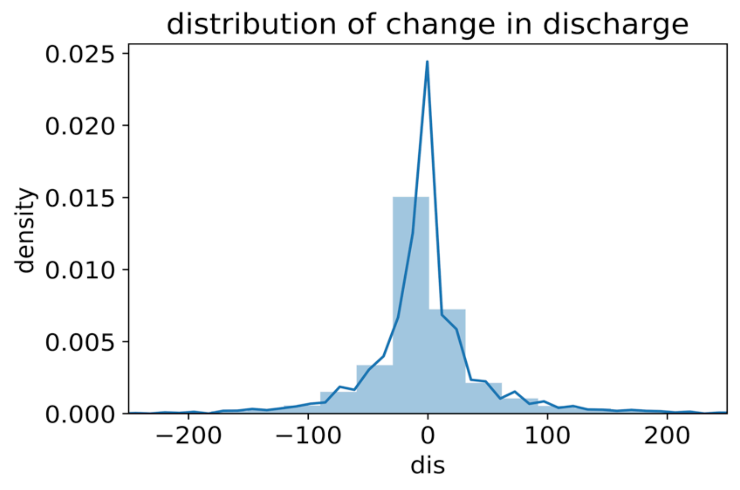
Data characteristics

- River discharge non-normal distribution => Predict change in discharge
- Predictors => normalization





©GLOWA Danube

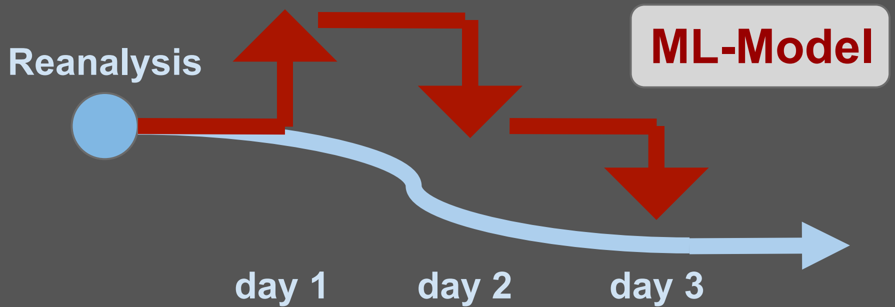


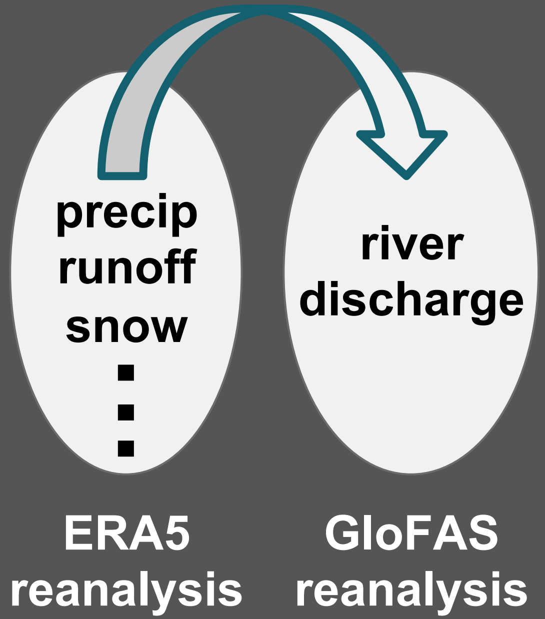
ECMWF

Workflow

- Determine point of interest (for discharge forecasts)
- Find upstream area (catchment)
- Spatially average features in the corresponding area
- Predictand: discharge at specified outflow point

=> <u>one-day forecast for change in discharge!</u>





ML model setup // Overview

Models:

- LinearRegressionModel via scikit learn
- SupportVectorRegressor via scikit learn
- GradientBoostingRegressor via scikit learn
- Time-Delay Neural Net via Keras

Training
1981 - 2005

Validation
2006 - 2011

Test
2012 - 2016

ML model settings // Validation

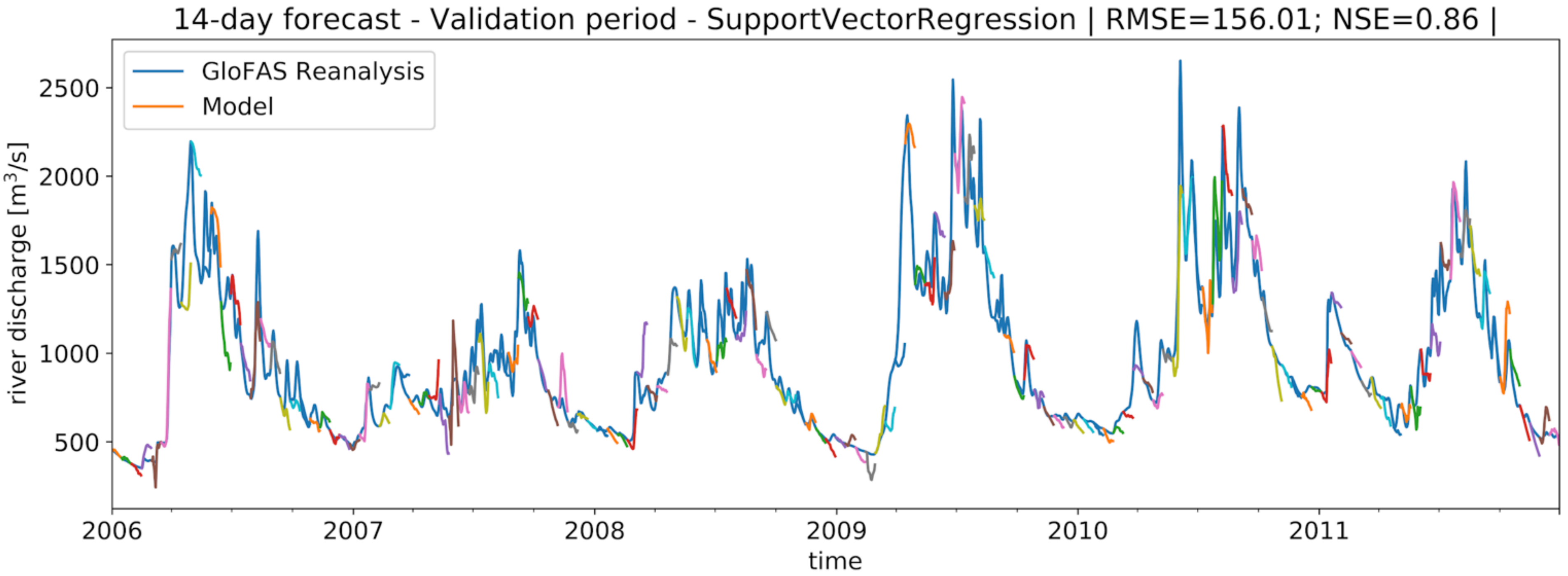
Hyperparameter optimization (via GridSearch):

- Standard OLS Linear Regression does not have any hyperparameters ✓
- SupportVectorRegressor: kernel=poly, C=100, epsilon=0.01, degree=3
- GradientBoostingRegressor: n_estimators=200, learning_rate=0.1, max_depth=5
- Time-Delay Neural Net: one hidden layer, 64 nodes (1281 dof), batch size 90 d, tanh activation

Validation	LR	SVR	GBR	TDNN
RMSE [m^3/s]	169	156	142	126
NSE	0.84	0.86	0.89	0.93

Notebooks: 3.02_LinearRegression.ipynb 3.03_SupportVectorRegression.ipynb
3.04_GradientBoost.ipynb 3.05_Time-delay_neural-net.ipynb

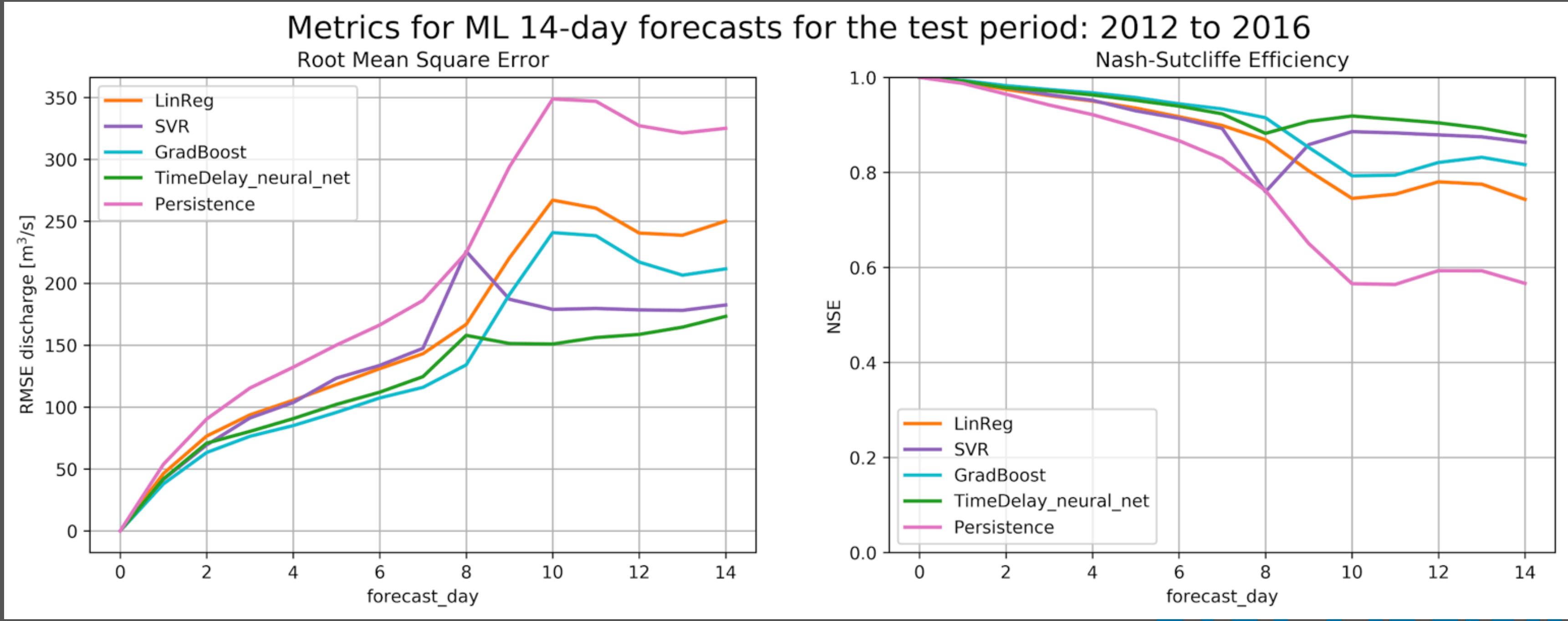
ML model setup // SupportVectorRegressor example



Notebook: [3.03_SupportVectorRegression.ipynb]()

Results // Model comparison in the full test period

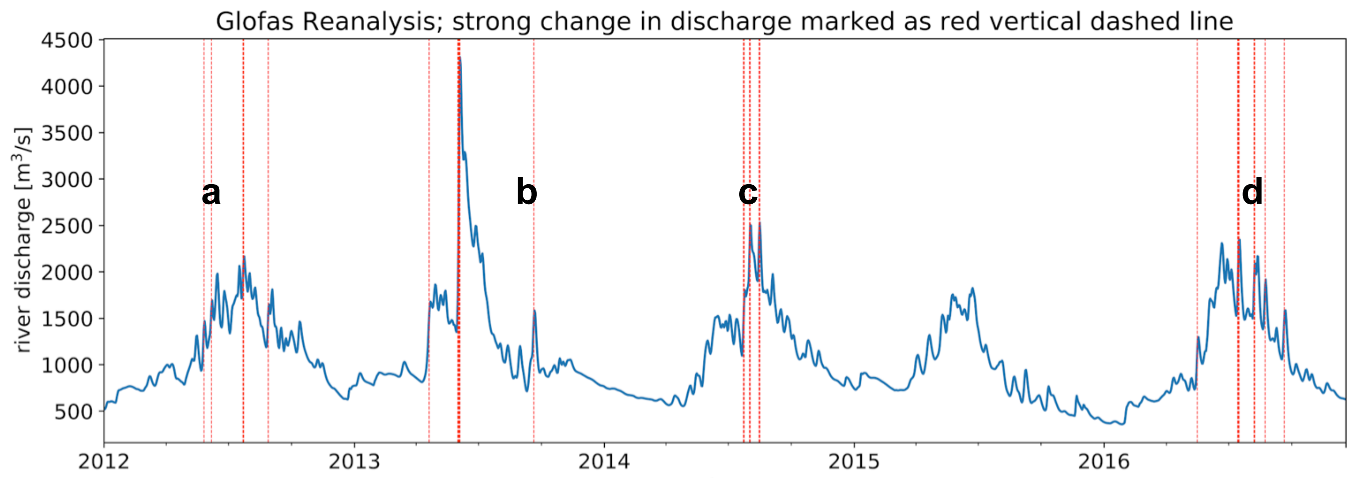
14-day forecasts over the full test period: Best model => Time-delay neural net



Notebook: 3.06_model_evaluation.ipynb

Results // Samples: rapid changes in discharge

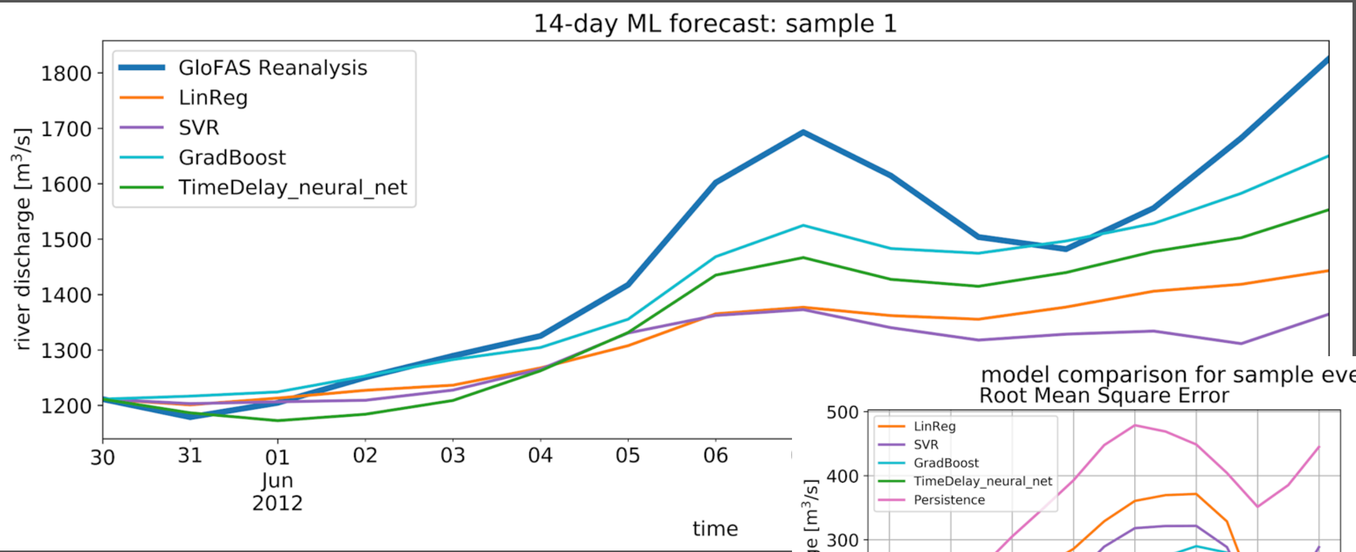
- 14-day forecast samples
- Init times:
    a. 30.06.2012
    b. 22.10.2013
    c. 03.08.2014
    d. 23.08.2016



- Flooding event 2013 => more detailed case study

Notebook: [3.06_model_evaluation.ipynb]

Results // Samples: rapid changes in discharge

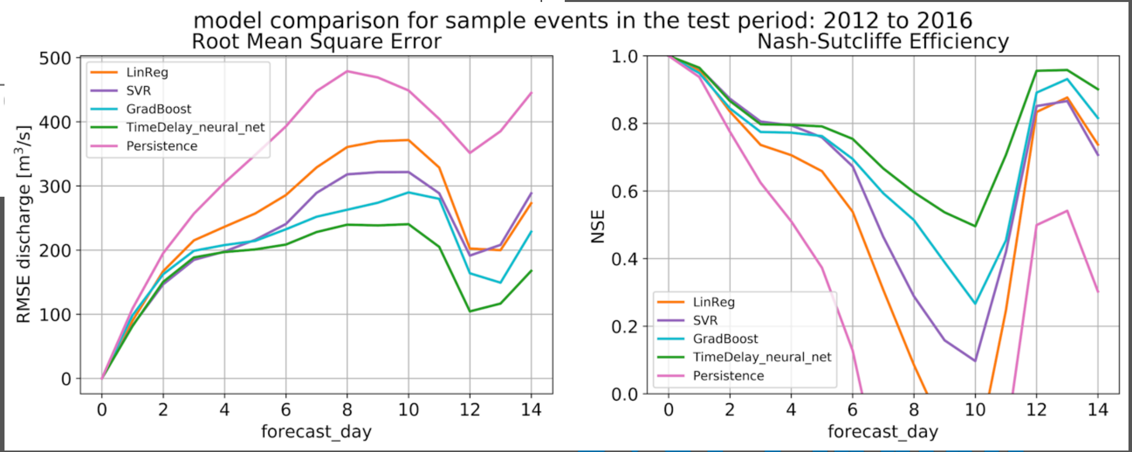


<= metrics

for sample period a)

metrics for all 4

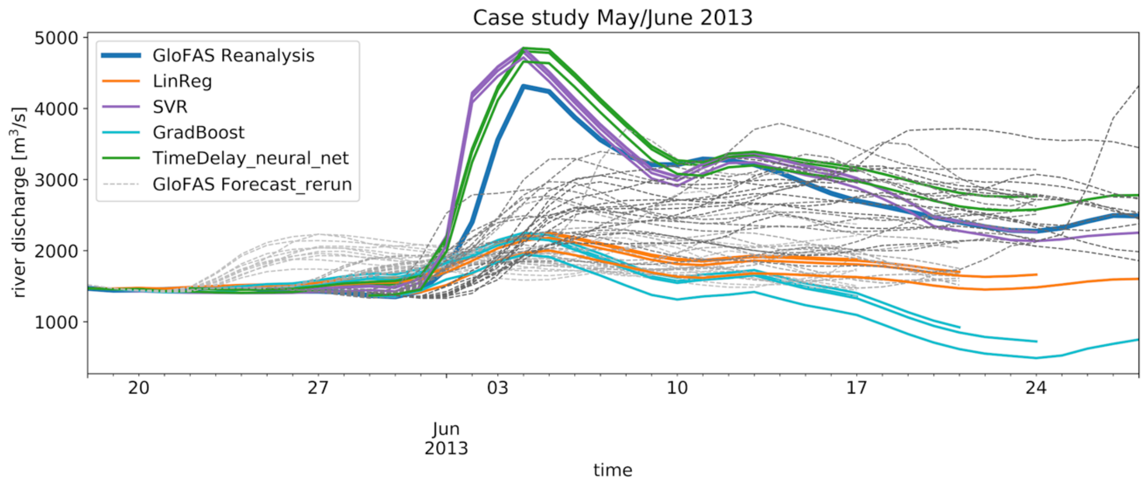
sample periods together =>



Notebook: [3.06_model_evaluation.ipynb]

Results // Case study: 2013 European Flood

- Reruns are forecast-driven
- ML is Reanalysis-driven



➔ Time-delay neural net & Support Vector Reg. perform best

➔ Grad. Boost. & Lin. Reg. Model show no significant increase in discharge (shape is quite fine, but amplitude not)

Notebook: [3.06_model_evaluation.ipynb]

Conclusion

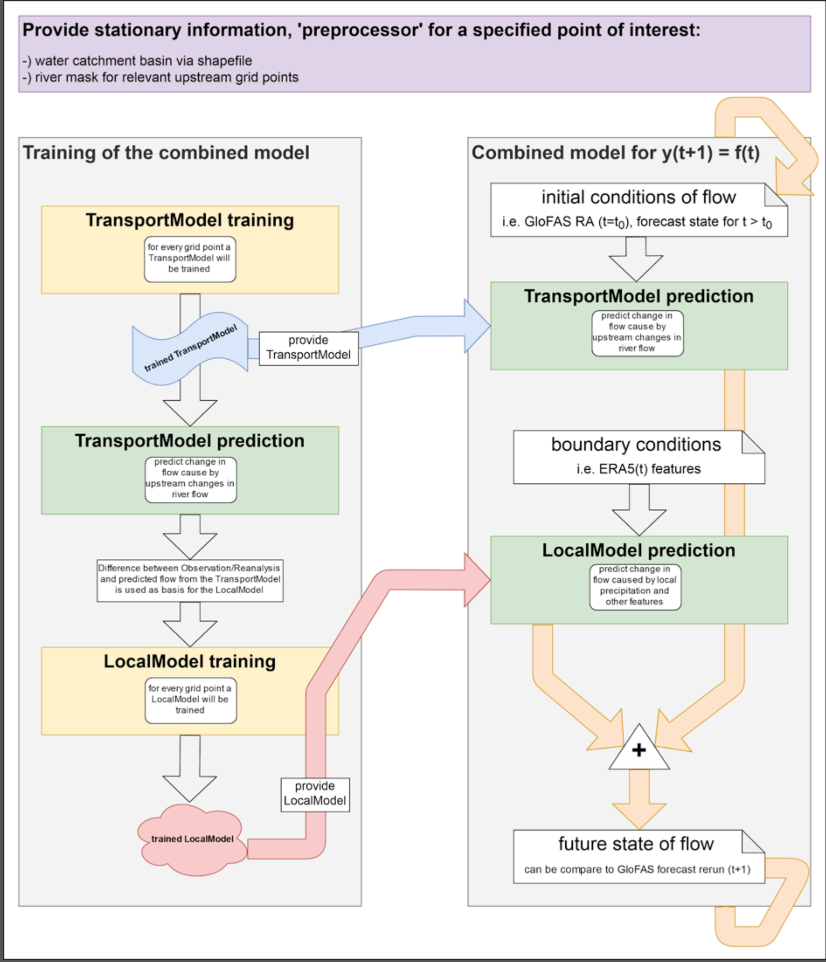
- The TD-NN and SVR models:
  - are able to predict an extreme flooding event, despite their relatively simple setting,
  - may provide cheap additional information to assist forecasts.
- LR and GBR were <u>not</u> able to predict an adequate increase in discharge (although in general GBR performed about as good as TD-NN and SVR)

Additional tuning of models → neural nets exhibit the most theoretical potential due to a lot of possible improvements in the base architecture (LSTM)

- Open source/reproducible (e.g. via Github) approach helps in getting quick feedback

Outlook

- Incorporate stationary information
  - e.g. catchment specific variables in a CNN or an LSTM as Embedding (see e.g. Kratzert et al 2019*)
- Explore different kinds of catchments
- Compare more extreme events with forecast reruns and analyse them
- Coupled model: Concept idea



Github: github.com/esowc/ml_flood Nbviewer: link

*Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., & Nearing, G. (2019). Benchmarking a Catchment-Aware Long Short-Term Memory Network (LSTM) for Large-Scale Hydrological Modeling. arXiv preprint arXiv:

ECMWF Copernicus Europe's eyes on Earth Climate Change Service Atmosphere Monitoring Service

Acknowledgements

Big thanks go to   

- Our mentors: Claudia Vitolo, Julia Wagemann and Stephan Siemen,
  for always providing helpful comments
- Esperanza Cuartero,
  for communicating the results
- The University of Vienna, Dept. of Meteorology and Geophysics,
  for office and server access
- ECMWF-Copernicus,
  for setting up ESoWC and providing such an opportunity!

universität wien

imgw Institut für Meteorologie und Geophysik