



ESoWC 2019

MATEHIW // MAchine learning TEchniques for High-Impact Weather

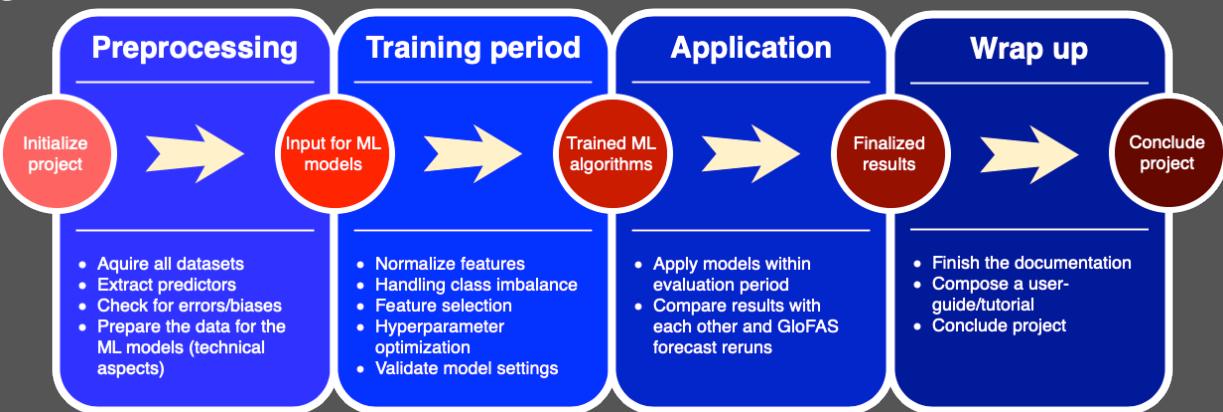
Lukas Kugler & Sebastian Lehner

Mentors: Claudia Vitolo, Julia Wagemann, Stephan Siemen

20.09.2019

Table of Contents

- Motivation & goal
- Principal model design
- Data
- Feature selection
- ML model setup
- Results
- Outlook



Motivation & goal

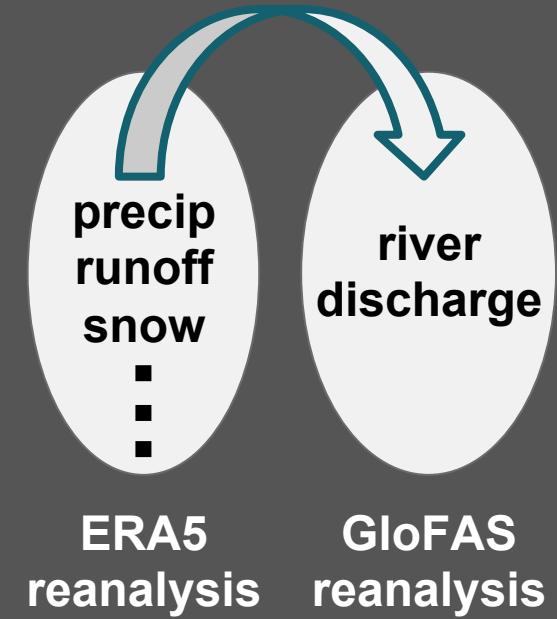
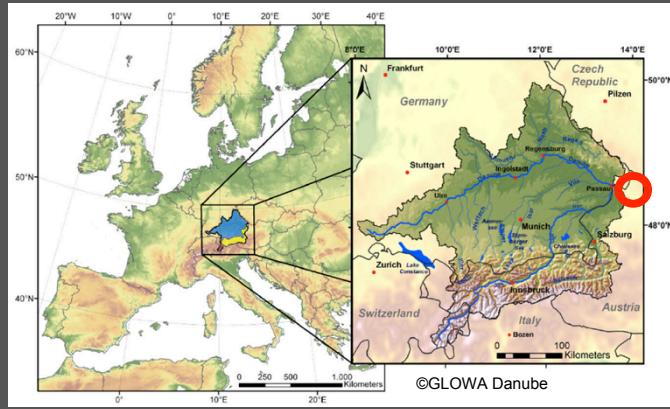
Modeling river discharge => high-dimensional problem: many meteorological/hydrological parameters important with complex interactions => coupled dynamical models.

- Can statistical connections based on ML techniques with comparable skill be established?
- 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)



Principal model design

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



Data // Download

- Daily resolution; spatial domain of interest
- Predictor data from ERA5 via the Climate Data Store API for python
- Predictand data from GloFAS
 - Reanalysis
 - 4 Forecast reruns for a flooding eventdirectly from the GloFAS team via FTP

A screenshot of a Python code editor showing a snippet of code interacting with the Climate Data Store API. The code includes imports like `logging`, `os`, and `requests`. It shows the creation of a session, sending a POST request to a URL, and handling the response. A callout box at the bottom right of the screenshot contains the text "Climate Data Store API".



Data // Preprocessing

CDO (Climate data operator) used for processing raw (hourly) data

Generate:

- daily averages
- daily accumulated precipitation
- extract data from bounding box (spatial region)
- merge into one .nc file

Setup and doc (mpimet): [link](#)

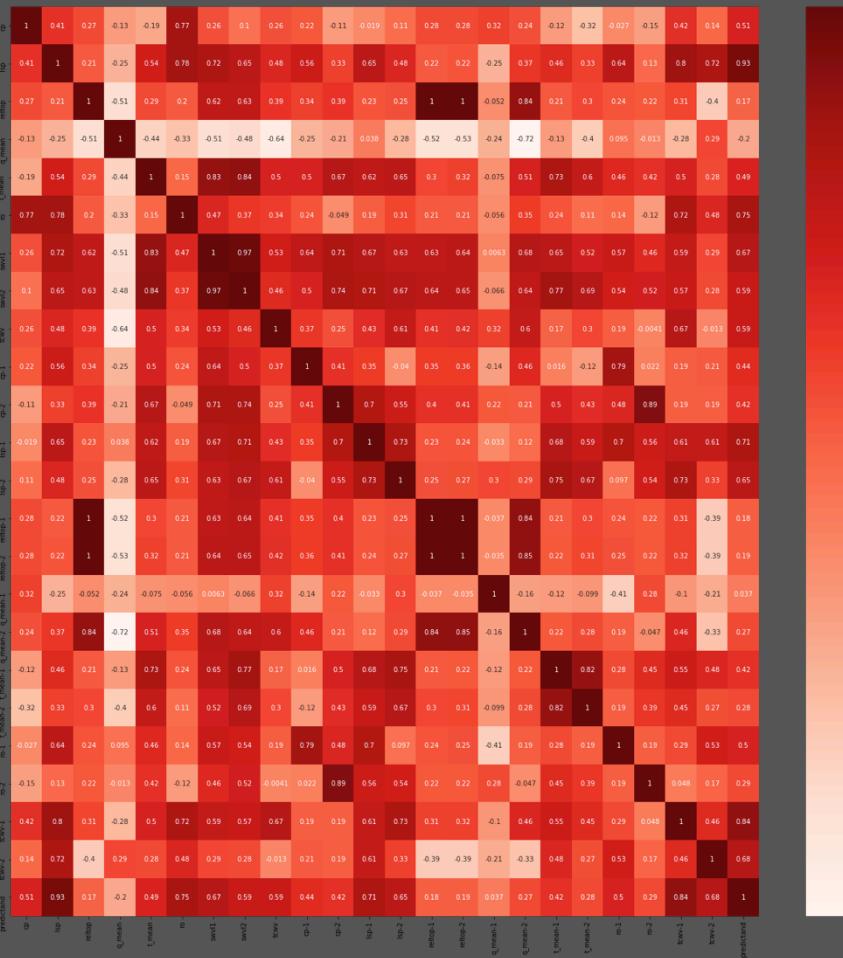
Python implementation:
<https://pypi.org/project/cdo/>

Feature selection // Heatmap

- Filter method: Correlation analysis
ERA5 variables:

runoff *large-scale precipitation* *convective precipitation*
total-column water vapor *soil water content layer 1* Δ *snowdepth*
relative topography *snowdepth* *soil water content layer 2*

=> same time as well as time shifted

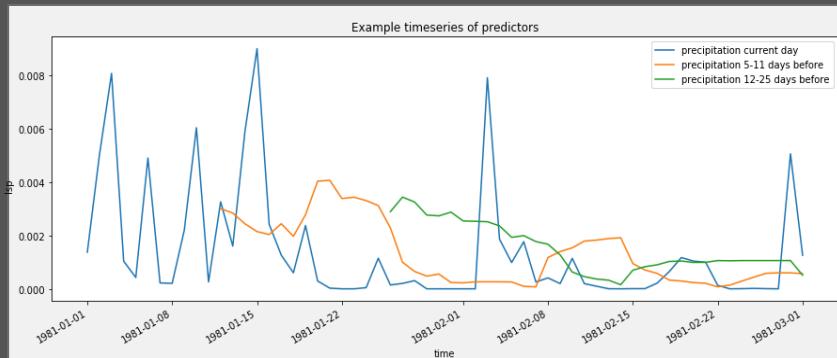
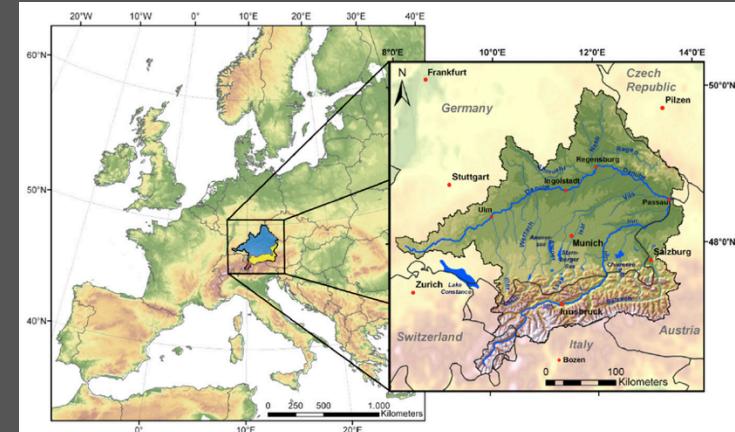


Feature selection // Relevant parameters

- Filter method: Correlation analysis (Heatmap)
 - precipitation (convective as well as large-scale)
 - time-lagged precipitation
 - runoff
 - soil water content layer 1
- ★ Total column water vapor high correlation, BUT collinearity with precipitation!
- ★ Time-lagged runoff high corr, BUT collinearity to itself (non time-lagged)!

Feature selection // Engineering

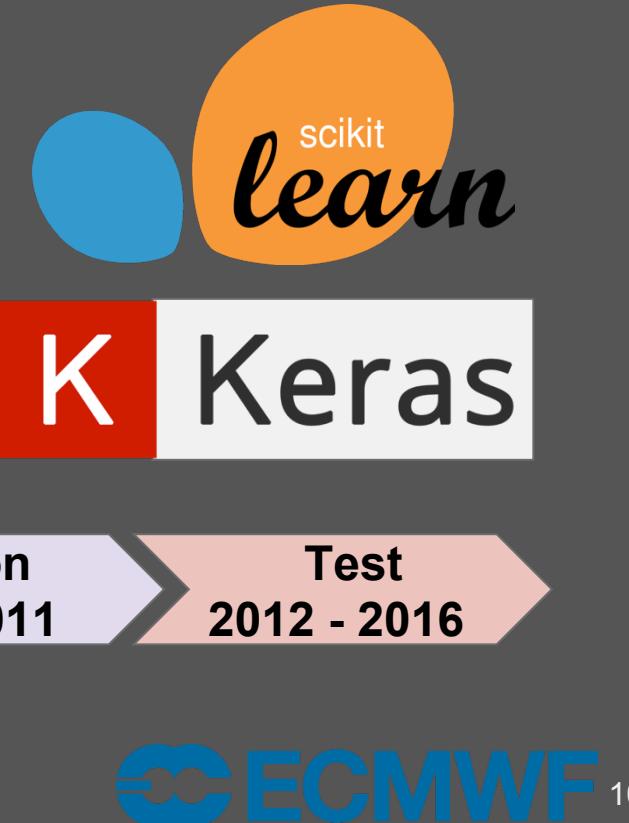
- Only consider data in the given catchment
- Average features
- Shift time (time-lagged features)
- Specially treated: Precipitation (large-scale)
 - Aggregation over time large time periods
 - captures long memory effects
 - up to 180 days into the past



ML model setup // Overview

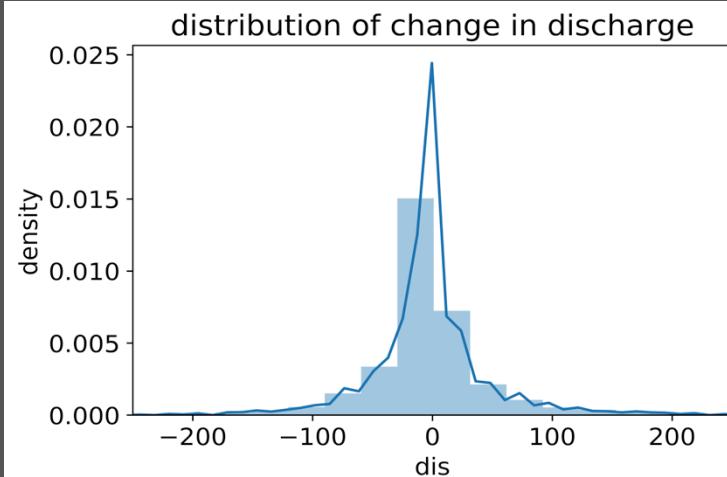
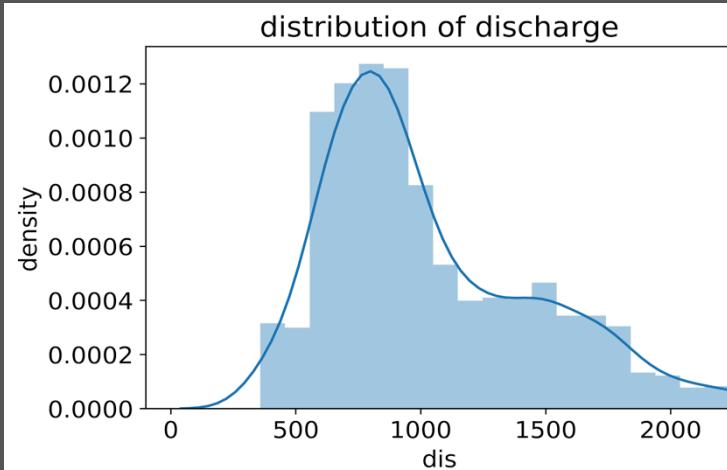
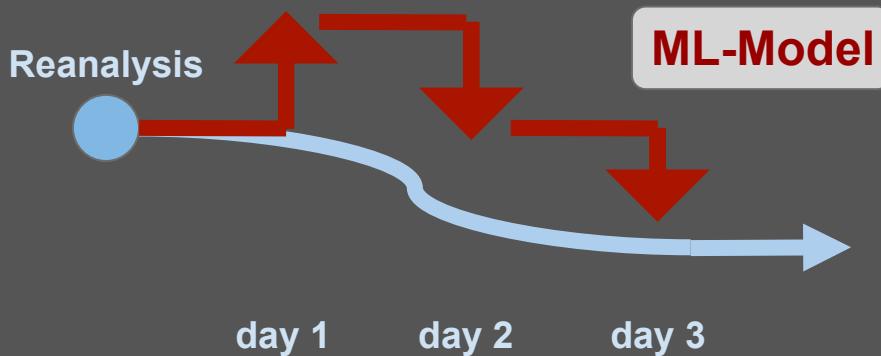
Models:

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



ML model setup // Overview

- River discharge non-normal distribution => Predict change in discharge
- 14-day forecast: Reanalysis baseline + cumulative sum of predicted change

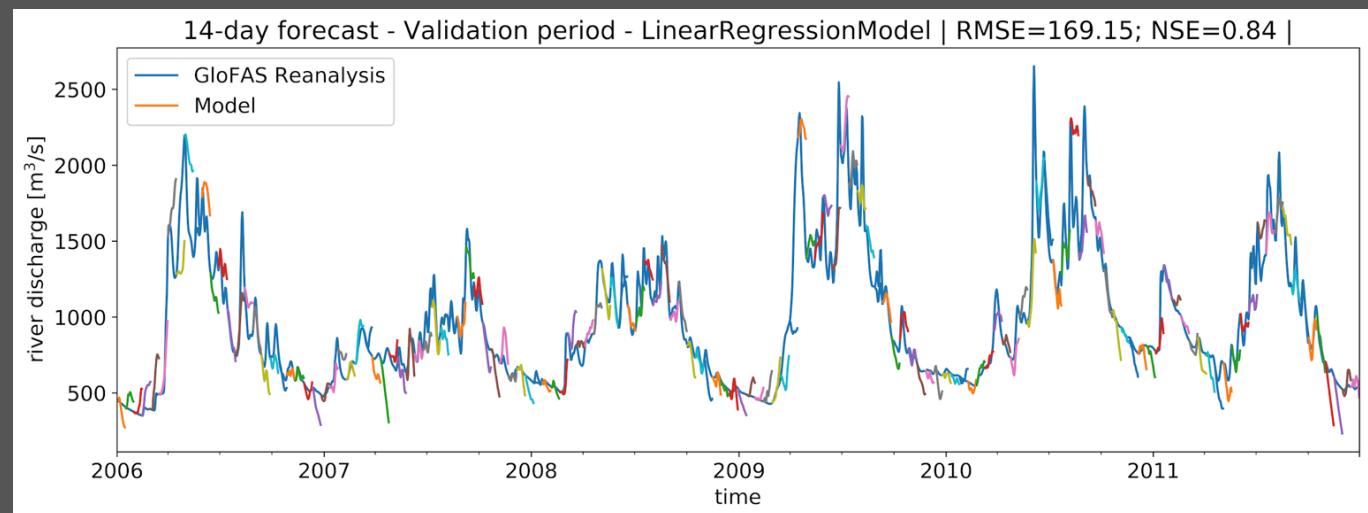


ML model setup // LinearRegressionModel

Hyperparameter optimization:

- Standard OLS Linear Regression does not have any hyperparameters ✓

RMSE	NSE
169 m ³ /s	0.84



ML model setup // SupportVectorRegressor

Hyperparameter optimization: via GridSearch

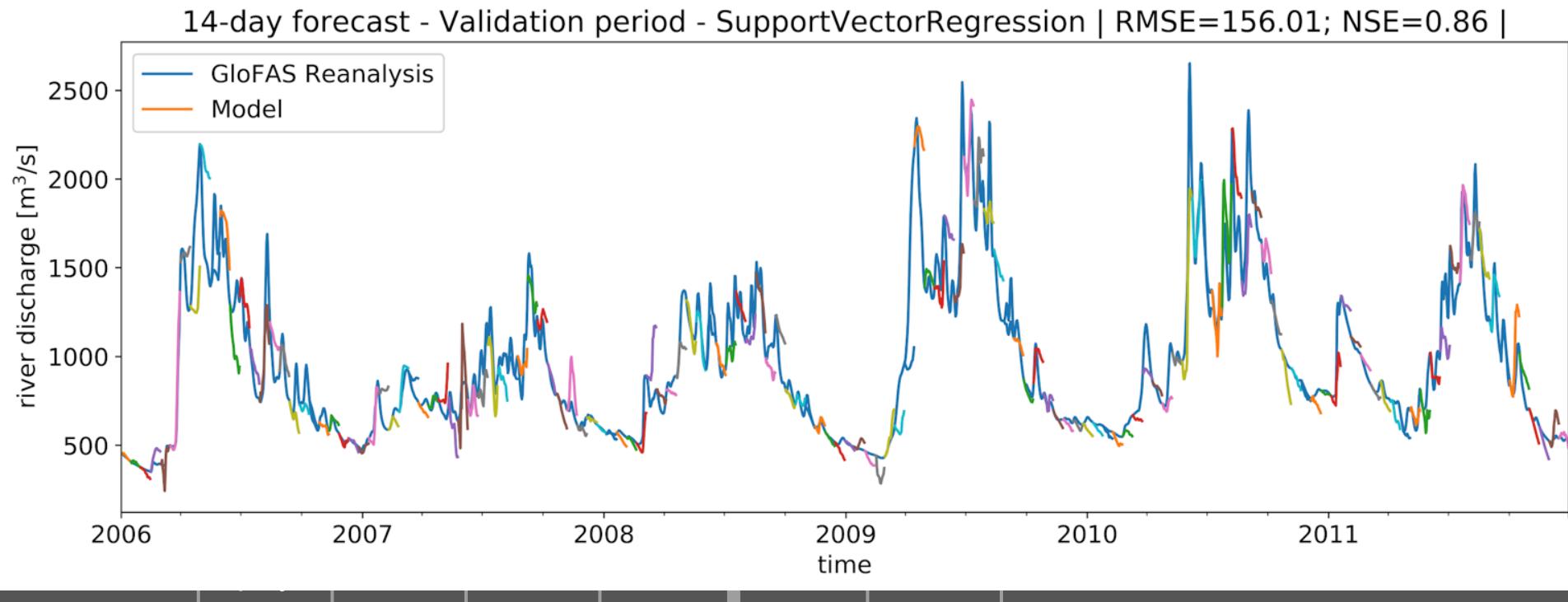
- kernel: rbf, poly
- C: [1, 10, 100]
- epsilon: [0.01, 0.1, 1]
- degree (if poly): [1, 2, 3, 4, 5]



Best setting:

kernel	C	epsilon	degree	RMSE	NSE
poly	100	0.01	3	156 m³/s	0.86

ML model setup // SupportVectorRegressor



ML model setup // GradientBoostingRegressor

Hyperparameter optimization: via GridSearch

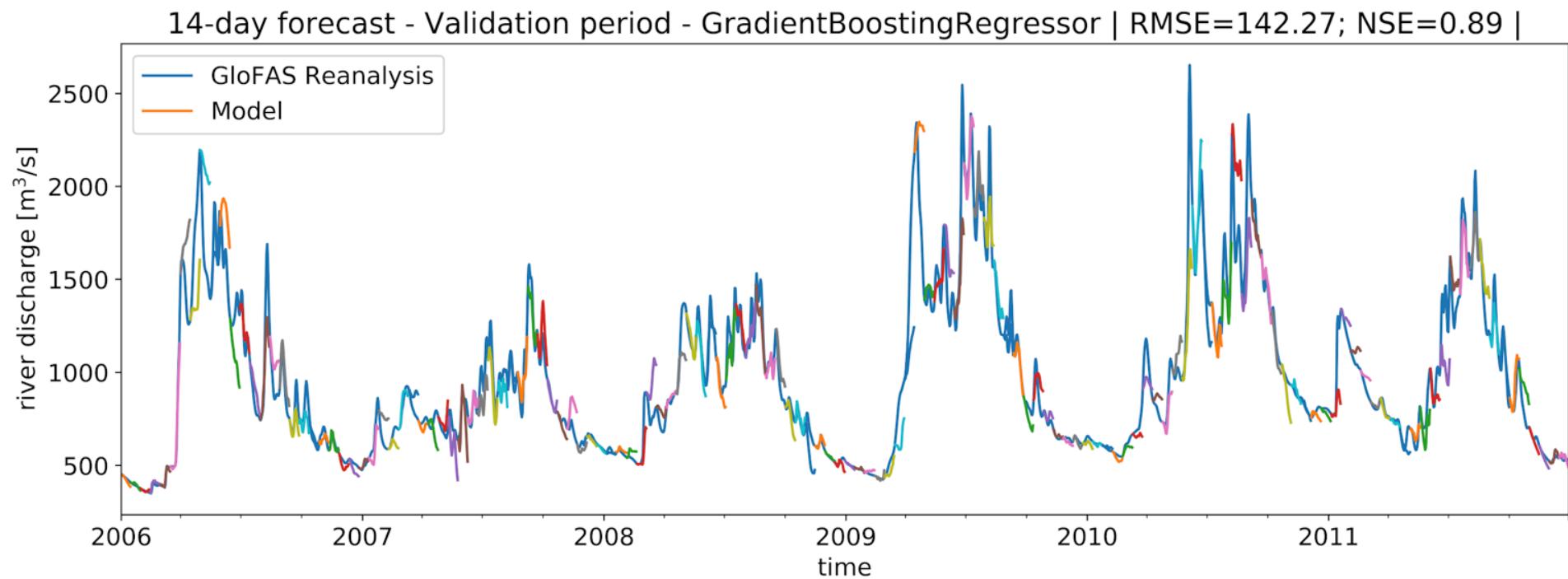
- n_estimators: [10, 50, 100, 200, 300]
- learning_rate: [0.05, 0.1, 0.2]
- max_depth: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15]



Best setting:

n_estimators	learning_rate	max_depth	RMSE	NSE
200	0.1	5	142 m ³ /s	0.89

ML model setup // GradientBoostingRegressor



ML model setup // Time-Delay Neural Net

Hyperparameter optimization:

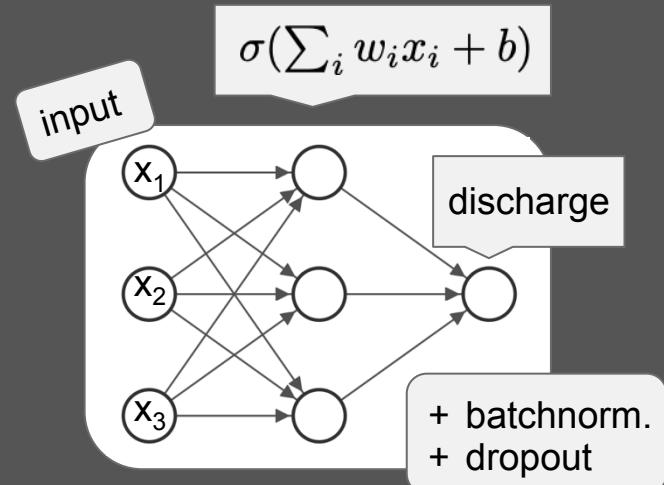
- Number of hidden layers: 1 - 4
- Number of nodes per layer: [4, 8, 16, 32]
- Dropout: 10% - 50%



Best setting:

one hidden layer, 64 nodes (1281 free parameters)

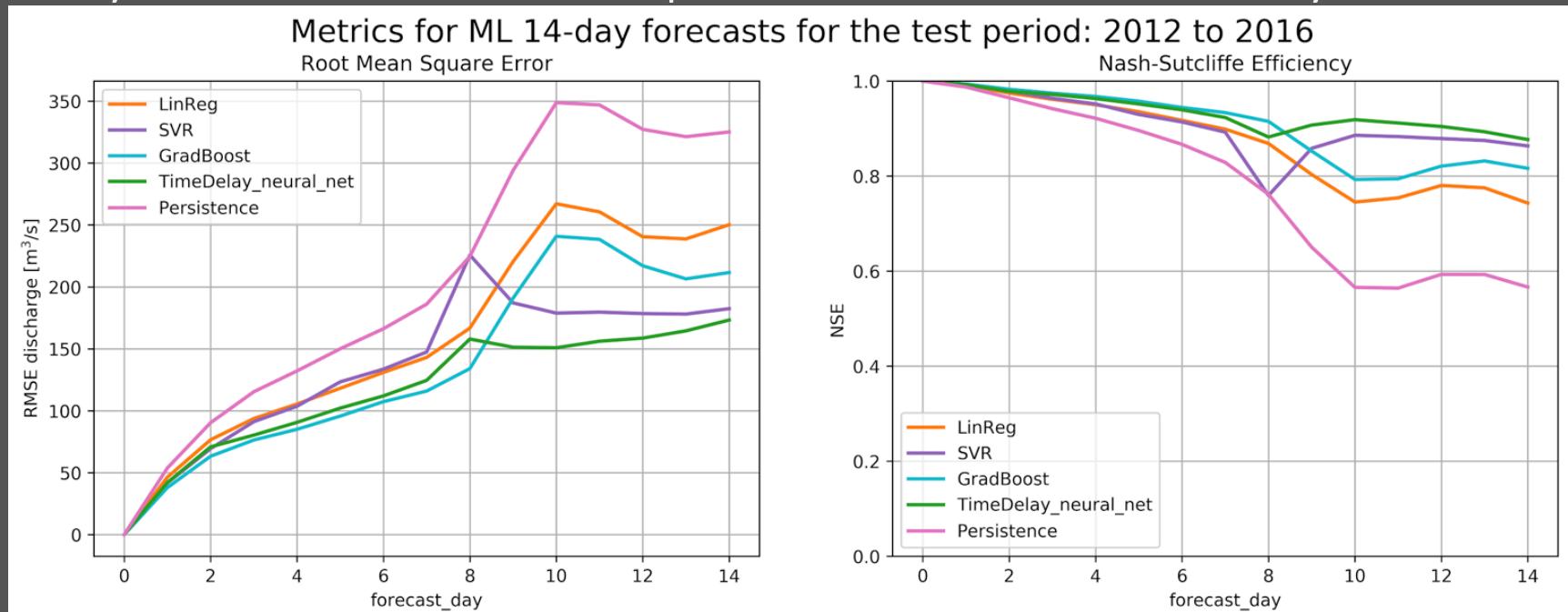
with batch size 90 days, tanh activation, 2 min training on 4 cores



RMSE	NSE
126 m ³ /s	0.93

Results // Model comparison in the full test period

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

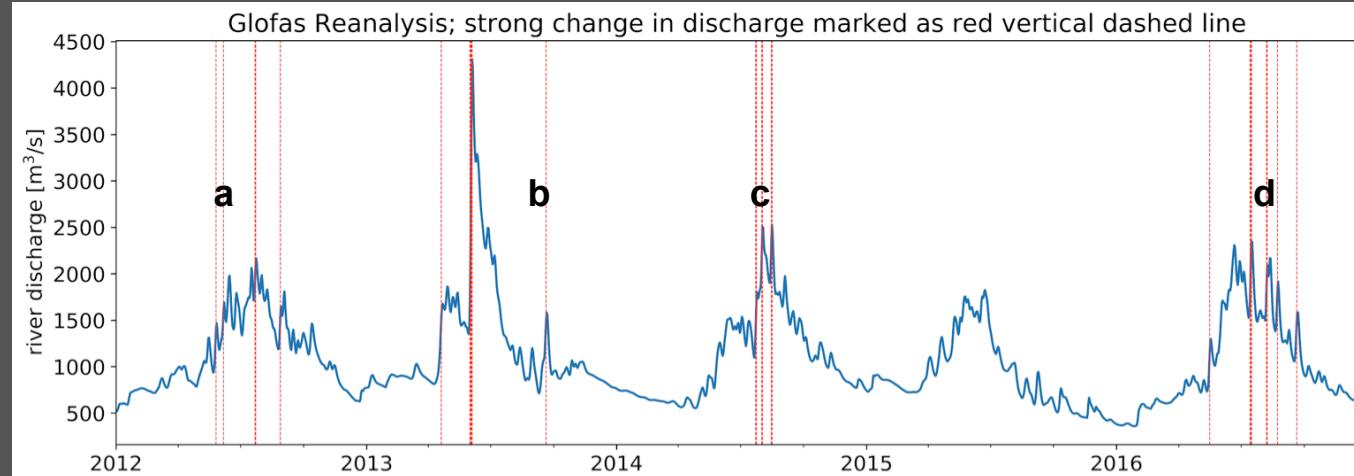


Results // Samples: rapid changes in discharge

- 14-day forecast samples

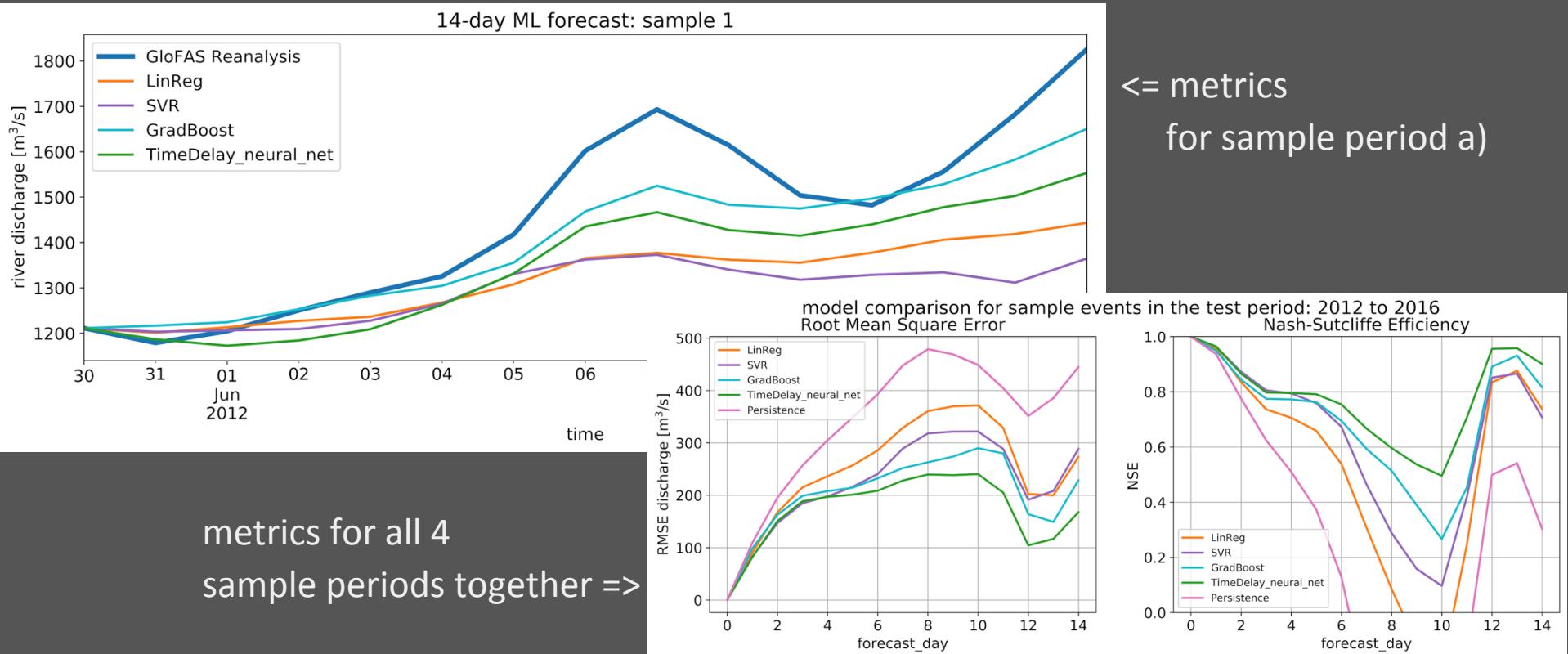
- Init times:

- a. 30.06.2012
- b. 22.10.2013
- c. 03.08.2014
- d. 23.08.2016



- Flooding event 2013 => more detailed case study

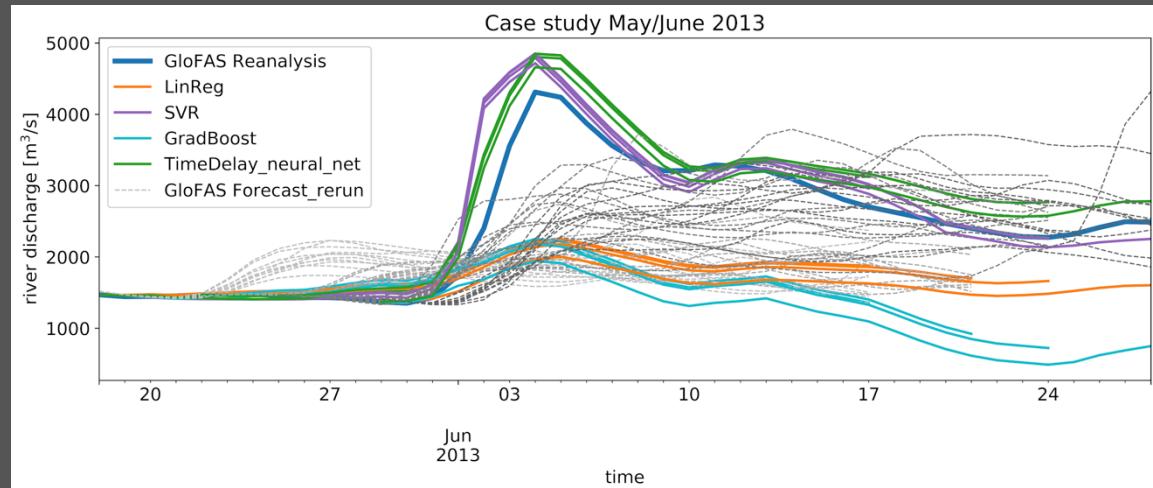
Results // Samples: rapid changes in discharge



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)



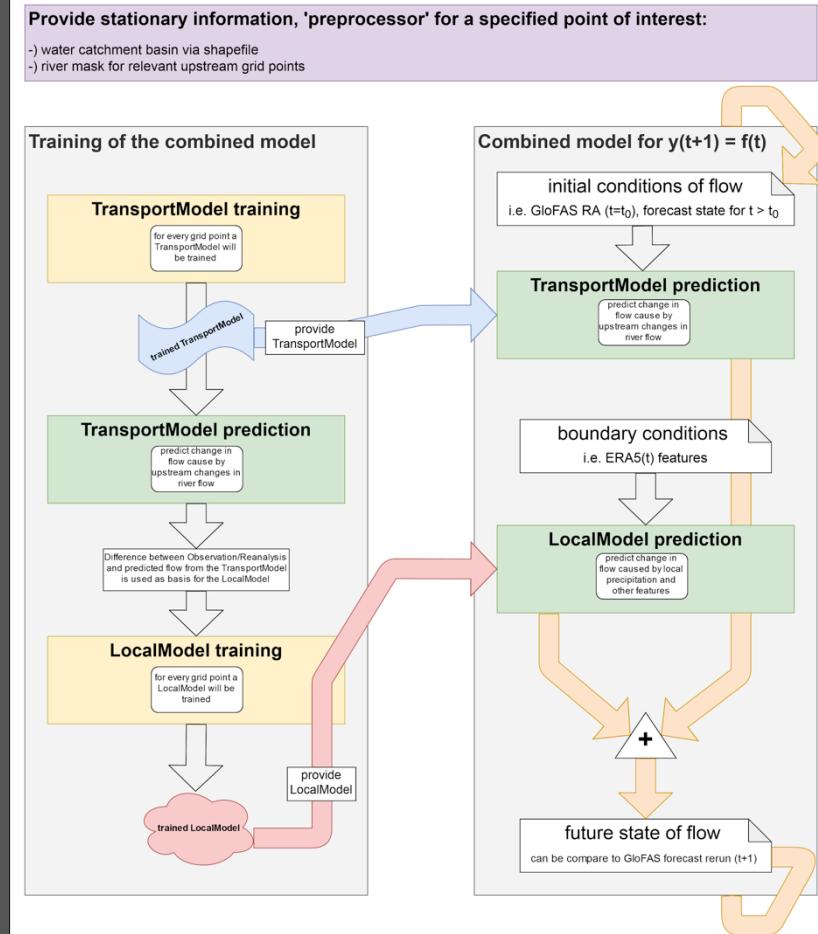
Conclusion

- 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 not 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

Outlook

- Incorporate stationary information
 - e.g. catchment specific variables in a CNN or an LSTM as Embedding
- More comprehensive case study in different kinds of catchments (surface characteristics)
- Coupled model: Concept idea



Github: github.com/esowc/ml_flood

Nbviewer: [link](#)

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!