# Decision Tree Ensembles for Postprocessing of Ensemble Weather Forecast

Ensemble weather forecasts often exhibit systematic errors and dispersion that must be corrected before the results can be used. Traditionally, these corrections are done by employing distributional regression models relaying on parametric forecast distributions. Although this linear regression framework works well, the use of additional predictors in the model can lead to its overfitting. We aim to propose a flexible alternative based on machine learning methods that are able to include arbitrary predictors and learn nonlinear dependencies. The decision tree ensemble methods such as quantile random forest, quantile extra-trees and natural gradient boosting are deployed for the postprocessing of ensemble weather forecasts.

We aim to correct the 24/240-hour ECMWF ensemble forecast of the 2m temperature (t2m) and the 24-hour sum of precipitation (prec24) at 39 surface stations in the Czech Republic. For the forecast correction, we employ five postprocessing algorithms that are trained/tested during the period from 2015 to 2019. The pipeline of the postprocessing method consists of four main parts:

1. Getting data
2. Data preprocessing
3. Build model
4. Model evaluation

## Data

#### Forecast

The forecasts are taken from the Interactive Grand Global Ensemble (TIGGE) dataset (Bougeault et al., 2010) via TIGGE archive. In particular, we use the global European Centre for Medium-Range Weather Forecasts (ECMWF) 50-member ensemble forecasts from 2016 to 2019, initialized at 00 UTC every day with a forecast lead time of +24h, +36h and +240h. The forecast domain is limited over the central Europe (2E, 30E; 42N, 55N) with 0.5° horizontal resolution. The following variables are use in the postprocessing MLPP system:

* 15 surface variables
* 5 variables from 500hPa pressure level
* 5 variables from 850hPa pressure level

Downloading the forecast data is done in:

forecasts/run\_forec\_down.sh

#### Observation

The forecasts of t2m and prec24 is evaluated at 36 meteorological stations in the Czech Republic. We use hourly measurements from a Weather Information Service provided by OGIMET. It uses freely available data from the net, mainly from NOAA, and it uses Open Software to process it. The download of observation data is based on the *climate* R package (Czernecki et al., 2020).

Downloading the observation data is done in:

observations/run\_obs\_down.sh

## Data preprocessing

For comparison the forecast with the station observations, the gridded data are interpolated bilinearly to the observation points. The 50-member ensemble forecasts are reduced to its mean and standard deviation. In addition to the target variables, we add extra metadata such as station altitude, latitude, longitude and the model orography. The final dataset is divided into train and test sets. The train set is used for the model estimation (model building) and we use two periods 2015-2018 and 2018 only to assess the importance of the training sample size. The model evaluation is performed during the period 2019.

Preprocessing the data is done in:

data\_preproc/run\_data\_interp.sh

data\_preproc/run\_metadata\_merge.sh

## Build model

#### Baseline models

It is used the distributional regression framework Ensemble Model Output Statistics (EMOS; Gneiting et al., 2005) where the conditional distribution of target variables (t2m/prec24) is modelled given the set of predictors. The gaussian distribution is used as a predictive distribution of target variables where the distribution parameters are connected to summary statistics of ensemble predictions. For precipitation the square root transformation is applied to achieve the bell shape of data. The EMOS coefficients are estimated by minimizing the mean Continuous Ranked Probability Score (CRPS) for Gaussian distribution.

Two types of EMOS models are used regarding the parameter estimation approach:

* EMOS-glb: one set of parameters for all stations
* EMOS-loc: different set of parameters for each station

The EMOS models are implemented in R by the use of *crch* package (Messner et al., 2016). Executing the EMOS models is done in:

methods/benchmark/run\_emos.sh

#### Decision Tree based models

Decision tree (DT) is a base learner in decision-making machine learning models that can be used to find nonlinear relationships between target variables and other relevant predictors. The predictors are used to identify meaningful weather regimes that may represent different types of synoptic conditions or geographical locations. A combination of hundreds or thousands DT either in ensemble (Ensemble learning models) or iteratively (Gradient Boosting machine) can improve the prediction accuracy while being robust to overfitting. In the context of postprocessing of ensemble weather prediction, the following three methods are employed:

* Quantile Random Forest Regressor (QRF; Meinshausen, 2006)
* Quantile Extreme Tree Regressor (XTR, Geurts et al., 2006)
* Natural Gradient Boosting Regressor (NGBoost; Duan et al., 2020)

The first two methods QRF and XTR are non-parametric distributional regression ensemble learning approaches which approximate the conditional distribution by a set of quantiles. The NGBoost method is gradient boosting algorithm that is generalized to probabilistic regression by treating the parameters of the conditional distribution as targets for a boosting algorithm.

The parameters of the all tree-based methods are tuned by minimizing the mean CRPS from a finite sample of ensemble forecast distribution.

The QRF and XTR models are implemented in Python by the use of *scikit-garden* library (scikit-garden, 2017). The NGBoost model is implemented in Python by the use of *ngboost* library (Duan et al., 2020). Training the tree-based models and running their predictions is done in:

methods/tree\_based/train\_tree\_models.sh

methods/tree\_based/pred\_tree\_models.sh

## Evaluation

The EMOS and tree-based model predictions are evaluated for target variables with respect to the station observations. The reference prediction is the raw forecast represented by the mean of the ECMWF ensemble forecast without any postprocessing methods (RAW-fcst). The quality of all the postprocessing models is evaluated during 2019. We evaluate the forecast lengths of 24 and 240 hours with respect to two training samples sizes 2015-2018 and 2018. The latter training period is evaluated only for the forecast length of 24 hours. The evaluation is based on the mean CRPS, which is a generalization of mean absolute error calculated from a finite number of samples of a probability distribution (Matheson and Winkler, 1976). The relative CRPS decrease represents improvement of CRPS by the use of postprocessing with respect to the raw forecast (RAW-fcst).

The model evaluation is implemented in R by the use of *scoringRules* package (Jordan et al., 2017). The evaluation is done in:

evaluation/run\_eval.sh

Tab. 1 The CRPS evaluation of 2m temperature and 24h precipitation forecasts corrected by the models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *2m temperature* | | | | |
| **Forecast length [hour]** | **24h** | | **240h** | |
| **Model/Training period** | **2015-2018** | **2018** | **2015-2018** | **2018** |
| RAW-fcst | 1.19 | 1.19 | 1.96 | - |
| EMOS-glb | 1.03 | 1.03 | 1.91 | - |
| EMOS-loc | 0.86 | 0.86 | **1.82** | - |
| QRF | **0.76** | 0.83 | 1.94 | - |
| XTR | **0.76** | **0.82** | 1.91 | - |
| NGBoost | 0.77 | 0.88 | 1.87 | - |
| *24h precipitation* | | | | |
| **Forecast length [hour]** | **24h** | | **240h** | |
| **Model/Training period** | **2015-2018** | **2018** | **2015-2018** | **2018** |
| RAW-fcst | 0.83 | 0.83 | 1.69 | - |
| EMOS-glb | 0.63 | 0.65 | 1.65 | - |
| EMOS-loc | 0.63 | 0.65 | 1.62 | - |
| QRF | **0.42** | 0.57 | **1.33** | - |
| XTR | **0.42** | 0.55 | **1.33** | - |
| NGBoost | 0.46 | **0.48** | 1.53 | - |

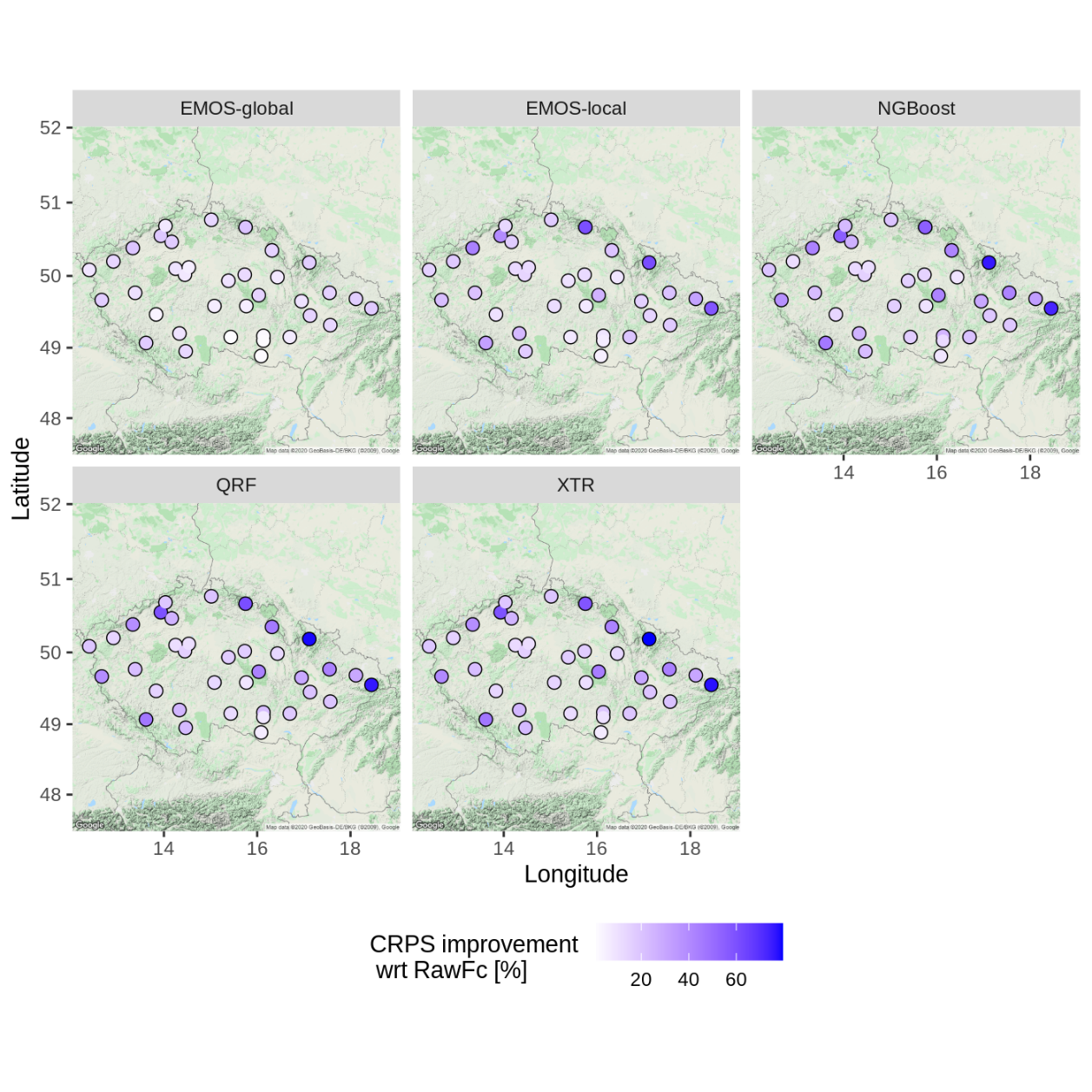


Fig. 1 The relative CRPS decrease of t2m at observation locations valid for the 24h forecast. The models are trained on the period 2015-2018.

## Conclusion

1. Nonlinear tree-based methods do better jobs on postprocessing of temperature and precipitation ensemble forecasts at lead time of 1 day. For the medium-range 10-day forecast, the linear EMOS methods are comparative especially for the temperature forecast.
2. The relative impact of postprocessing methods decreases with the forecast length. For tree-based methods:
   1. mean relative CRPS decrease of temperature forecast is 35% and 3% for 1-day and 10-day lead times, respectively
   2. mean relative CRPS decrease of precipitation forecast is 48% and 17% for 1-day and 10-day lead times, respectively
3. The positive impact of tree-based and EMOS-loc postprocessing methods is mostly located:
   1. at mountain stations for temperature forecasts
   2. evenly distributed at all stations
4. Regarding the calibration of corrected precipitation forecasts, the raw (uncorrected) forecast overestimates weak precipitation and underestimates medium and heavy precipitation, which is in line with expectations. The use of postprocessing methods show a poor calibration. While little positive impact is detected on the medium/heavy precipitation, all the methods tend to underestimate weak precipitations. The best calibration of precipitations is provided by the NGBoost method.

## Technical details

* Source code: R (76.2%), Python 3.8.1 (21.5%), Shell (2.3%)
* The workspace environment created by the software Anaconda
* Scikit-garden issues:
  + Parallelization of XTR and QRF by *n\_jobs* model parameter does not work. This problem was already opened [here](https://github.com/scikit-garden/scikit-garden/issues/59). Therefore, we use the *multiprocessing* package to fully leverage multiple processors and speed-up the model training.
  + The default QRF and XTR models are very slow at making predictions. We make the model much faster by selecting a random target value per leaf instead of their weighting average. Provided a large ensemble of trees (100-500), this approach has very little effect on the prediction confidence interval.

## Acknowledgement

This study was progressed rapidly thanks to S. Lerch and S. Rasp who provided the public with the source code of their project (Rasp and Lerch, 2018).

## References

BOUGEAULT, Philippe, et al. The THORPEX interactive grand global ensemble. *Bulletin of the American Meteorological Society*, 2010, 91.8: 1059-1072.

CZERNECKI, Bartosz; GŁOGOWSKI, Arkadiusz; NOWOSAD, Jakub. Climate: An R package to access free in-situ meteorological and hydrological datasets for environmental assessment. *Sustainability*, 2020, 12.1: 394.

DUAN, Tony, et al. Ngboost: Natural gradient boosting for probabilistic prediction. In: *International Conference on Machine Learning*. PMLR, 2020. p. 2690-2700.

GEURTS, Pierre; ERNST, Damien; WEHENKEL, Louis. Extremely randomized trees. *Machine learning*, 2006, 63.1: 3-42.

GNEITING, Tilmann, et al. Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review*, 2005, 133.5: 1098-1118.

JORDAN, Alexander; KRÜGER, Fabian; LERCH, Sebastian. Evaluating probabilistic forecasts with scoringRules. *arXiv preprint arXiv:1709.04743*, 2017.

MATHESON, James E.; WINKLER, Robert L. Scoring rules for continuous probability distributions. *Management science*, 1976, 22.10: 1087-1096.

MEINSHAUSEN, Nicolai. Quantile regression forests. Journal of Machine Learning Research, 2006, 7. Jun: 983-999.

MESSNER, Jakob W.; MAYR, Georg J.; ZEILEIS, Achim. Heteroscedastic Censored and Truncated Regression with crch. *R J.*, 2016, 8.1: 173.

RASP, Stephan; LERCH, Sebastian. Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, 2018, 146.11: 3885-3900. The results available at: github.com/slerch/ppnn

SCIKIT-GARDEN: API reference, 2017: <https://scikit-garden.github.io/api/>