Neural Computing and Applications
https://doi.org/10.1007/s00521-020-05139-4

ORIGINAL ARTICLE

Machine learning for total cloud cover prediction

Ágnes Baran[1] · Sebastian Lerch[2] · Mehrez El Ayari[1,3] · Sándor Baran[1]

Received: 19 March 2020 / Accepted: 16 June 2020

Abstract
Accurate and reliable forecasting of total cloud cover (TCC) is vital for many areas such as astronomy, energy demand and production, or agriculture. Most meteorological centres issue ensemble forecasts of TCC; however, these forecasts are often uncalibrated and exhibit worse forecast skill than ensemble forecasts of other weather variables. Hence, some form of post-processing is strongly required to improve predictive performance. As TCC observations are usually reported on a discrete scale taking just nine different values called oktas, statistical calibration of TCC ensemble forecasts can be considered a classification problem with outputs given by the probabilities of the oktas. This is a classical area where machine learning methods are applied. We investigate the performance of post-processing using multilayer perceptron (MLP) neural networks, gradient boosting machines (GBM) and random forest (RF) methods. Based on the European Centre for Medium-Range Weather Forecasts global TCC ensemble forecasts for 2002–2014, we compare these approaches with the proportional odds logistic regression (POLR) and multiclass logistic regression (MLR) models, as well as the raw TCC ensemble forecasts. We further assess whether improvements in forecast skill can be obtained by incorporating ensemble forecasts of precipitation as additional predictor. Compared to the raw ensemble, all calibration methods result in a significant improvement in forecast skill. RF models provide the smallest increase in predictive performance, while MLP, POLR and GBM approaches perform best. The use of precipitation forecast data leads to further improvements in forecast skill, and except for very short lead times the extended MLP model shows the best overall performance.

Keywords Ensemble calibration · Logistic regression · Multilayer perceptron · Total cloud cover

Abbreviations

CDF	Cumulative distribution function
CRPS	Continuous ranked probability score
CRPSS	Continuous ranked probability skill score
CTRL	(ECMWF) Control (forecast)
DM	Diebold–Mariano (test)
ECMWF	European Centre for Medium-Range Weather Forecasts
ENS	(50-member ECMWF) ensemble
EPS	Ensemble prediction system
GBM	Gradient boosting machine
HRES	(ECMWF) High-resolution (forecast)
LogS	Logarithmic score
LogSS	Logarithmic skill score
MLP	Multilayer perceptron
MLR	Multiclass logistic regression
NWP	Numerical weather prediction
PIT	Probability integral transform
PMF	Probability mass function
POLR	Proportional odds logistic regression
QRF	Quantile regression forest
RF	Random forest
SYNOP	Synoptic observation
TCC	Total cloud cover
UTC	Coordinated universal time

✉ Sándor Baran
baran.sandor@inf.unideb.hu

[1] Faculty of Informatics, University of Debrecen, Kassai út 26, Debrecen 4028, Hungary

[2] Institute for Stochastics, Karlsruhe Institute of Technology, Englerstraße 2, 76128 Karlsruhe, Germany

[3] Doctoral School of Informatics, University of Debrecen, Kassai út 26, Debrecen 4028, Hungary

Published online: 06 July 2020

1 Introduction

Reliable and accurate prediction of total cloud cover (TCC) has a principal importance in observational astronomy [1] and in the prediction of photovoltaic energy production, as it is the main cause of variation in solar-radiation energy supply [2, 3], but it is also of great relevance in agriculture, tourism and in some other fields of economy. According to the definition of the World Meteorological Organization, "total cloud cover is the fraction of the sky covered by all the visible clouds" [4]. Even though this definition indicates a continuous quantity in the [0, 1] interval, TCC observations are usually reported in eighths of sky cover called oktas taking just nine different values. In this way, TCC forecasting can be considered as a nine-group classification problem and thus requires markedly different methods than those used for other weather variables such as temperature, wind speed or precipitation accumulation, which are treated as continuous quantities.

TCC forecasts are generated using numerical weather prediction (NWP) models (for a comparison of the performance of the state-of-the-art techniques see [5]), and recently all major meteorological centres issue ensemble forecasts of TCC using their operational ensemble prediction systems (EPSs). Examples include the Global Ensemble Forecast System of National Centers for Environmental Prediction [6] or the EPS of the independent intergovernmental European Centre for Medium-Range Weather Forecasts (ECMWF) [7–9]. With the help of a forecast ensemble, one can estimate the probability distribution of future weather variables, which opens the door for probabilistic weather forecasting [10], where besides the future atmospheric states the related uncertainty information (variance, probabilities of various events, etc.) are also predicted. However, ensemble forecasts often tend to be underdispersive, that is the spread of the ensemble is too small to accurately capture the full uncertainty, and can be subject to systematic bias. This phenomenon can be observed with several operational EPSs (see e.g. [11, 12]) calling for some form of statistical post-processing [13]. TCC ensemble forecasts are even more problematic, as in terms of forecast skill they highly underperform ensemble forecasts of other weather variables such as temperature, wind speed, pressure or precipitation (see e.g. [14, 15]).

Over the past decade, various statistical post-processing methods have been proposed in the meteorological and statistical literature, for an overview see e.g. [16] or [17]. These include parametric approaches like Bayesian model averaging [18] or non-homogeneous regression [19] providing estimates of the probability distributions of the weather quantities of interest, nonparametric techniques like quantile regression (see e.g. [20, 21]) or mixed methods such as quantile mapping (see e.g. [22, 23]). Recently, machine learning methods have become more and more popular in ensemble post-processing. For example, Taillardat et al. [24] used quantile regression forests (QRF) for calibration of ensemble forecasts of temperature and wind speed, and Taillardat et al. [25] recently extended the technique to precipitation forecasts. Rasp and Lerch [26] applied neural networks for post-processing of ECMWF near-surface temperature ensemble forecasts using QRF as a benchmark model, whereas Bremnes [27] employed neural networks in quantile function regression for calibrating ensemble forecasts of wind speed. Bakker et al. [28] compare several machine learning approaches for post-processing NWP predictions of solar radiation based on quantile regression, including random forests, gradient boosting and neural networks.

Probabilistic forecasting approaches estimating predictive distributions can be considered as the most advanced prediction methods not only in atmospheric sciences, but in other fields of science and economy, e.g. in economical risk management, seismic hazard prediction or financial forecasting. For a detailed overview of the main concepts and properties of probabilistic forecasts and the areas of application see [29].

The discrete nature of TCC means that the predictive distribution should take the form of a discrete probability distribution and post-processing can be considered as a classification problem resulting in the probabilities of the oktas. For calibrating TCC ensemble forecasts, Hemri et al. [30] propose two discrete parametric post-processing approaches, namely multiclass logistic regression (MLR) [31] and proportional odds (or ordered) logistic regression (POLR) [32]. Different versions of logistic regression had already been successfully applied in statistical post-processing (see e.g. [33, 34]) and ordered logistic regression also showed good performance for forecasts of discrete categories [35].

Since probabilistic multi-category classification is one of the main areas of application of machine learning, the main goal of our work here is to investigate the use of machine learning methods for total cloud cover prediction in the framework of statistical post-processing of TCC ensemble forecasts. With the help of ECMWF global ensemble forecasts for the period 2002–2014, we test the performance of multilayer perceptron neural networks (MLP) [36], gradient boosting machine (GBM) [37] and random forest algorithms (RF) [38], and compare their forecast skill with the raw TCC ensemble and the MLR and POLR approaches of [30]. We further investigate the effect of using precipitation ensemble forecasts as additional predictors in TCC post-processing. More accurate TCC forecasts can be expected to result in improved predictions of produced photovoltaic energy; however, this topic is beyond the scope of the current work and is a subject of further research.

The paper is organized as follows. Section 2 contains the description of the TCC and precipitation ensemble forecasts and observations. Section 3 reviews the various calibration methods and tools used for forecast evaluation. A case study on post-processing of TCC ensemble forecasts is provided in Sect. 4, and the article concludes with a discussion in Sect. 5.

2 Data

We consider 52-member ECMWF global ensemble forecasts (high-resolution forecast (HRES), control forecast (CTRL) and 50 members (ENS) generated using random perturbations) of TCC and 24 h precipitation accumulation initialized at 1200 UTC for 10 different lead times ranging from 1 day to 10 days for the period between January 1, 2002, and March 20, 2014, together with the corresponding observations. The TCC data set is identical to the one investigated in [30] containing data for 3330 synoptic observation (SYNOP) stations left after an initial quality control. TCC SYNOP observations are reported in values $\mathcal{Y} = \{0, 0.1, 0.25, 0.4, 0.5, 0.6, 0.75, 0.9, 1\}$ corresponding to the different oktas, whereas the raw ensemble forecasts are continuous values in the [0, 1] interval. The matching of forecasts and observations is performed with quantization of forecast values using intervals

$$[0, 0.01[, \quad [0.01, 0.1875[, \quad [0.1875, 0.3125[,$$
$$[0.3125, 0.4375[, \quad [0.4375, 0.5625[,$$
$$[0.5625, 0.6875[, \quad [0.6875, 0.8125[,$$
$$[0.8125, 0.99[, \quad [0.99, 1],$$

that is raw or post-processed forecasts falling, e.g. into the interval [0.1875, 0.3125[correspond to observation value 0.25 (see [30, Table A1]).

Our additional precipitation data set, which has been investigated in [39], contains forecast-observation pairs for 2917 SYNOP stations after quality control. At 2239 of these station, both TCC and precipitation data are available.

3 Calibration methods and forecast evaluation

In what follows, let $Y \in \mathcal{Y} = \{y_1, y_2, \ldots, y_9\}$ be TCC at a given location and time expressed in oktas and denote by $\boldsymbol{f} = (f_1, f_2, \ldots, f_{52})$ the corresponding 52-member ECMWF TCC ensemble forecast with a given lead time, where $f_1 = f_{\text{HRES}}$ and $f_2 = f_{\text{CTRL}}$ are the high-resolution and control members, respectively, whereas $f_3, f_4, \ldots, f_{52}$ correspond to the 50 statistically indistinguishable (and thus exchangeable) ensemble members $f_{\text{ENS},1}, f_{\text{ENS},2}, \ldots, f_{\text{ENS},50}$ generated using random perturbations. In this discrete setting, the estimation of the predictive distribution of Y reduces to the estimation of conditional probabilities

$$\mathsf{P}(Y = y_k \,|\, \boldsymbol{f}), \qquad k = 1, 2, \ldots, 9. \tag{1}$$

Obviously, in (1) the raw ensemble forecast $\boldsymbol{f}$ can be replaced by any feature vector $\boldsymbol{x}$ derived from the ensemble and/or other covariates. In order to ensure comparability with the reference MLR and POLR approaches for classification using TCC data only (see Sect. 4.2), we consider the same feature set as in [30]. The investigated covariates are the HRES forecast f_{HRES}, the control forecast f_{CTRL}, the mean of the 50 exchangeable ensemble members $\overline{f}_{\text{ENS}}$, the ensemble variance

$$s^2 := \frac{1}{51}\sum_{i=1}^{52}(f_i - \overline{f})^2, \quad \text{where} \quad \overline{f} := \frac{1}{52}\sum_{i=1}^{52} f_i,$$

the proportions of forecasts predicting zero and maximal cloud cover

$$p_0 := \frac{1}{52}\sum_{i=1}^{52} \mathbb{I}_{\{f_i = 0\}} \quad \text{and} \quad p_1 := \frac{1}{52}\sum_{i=1}^{52} \mathbb{I}_{\{f_i = 1\}},$$

respectively, where $\mathbb{I}_H$ denotes the indicator function of a set H, and an interaction term

$$I := s^2 \text{sign}(d) d^2 \quad \text{with} \quad d := \big((f_{\text{HRES}} - 0.5) + (f_{\text{CTRL}} - 0.5) + (\overline{f}_{\text{ENS}} - 0.5)\big)/3$$

connecting the ensemble variance and the mean deviation of the first three features from 0.5.

As additional feature we also consider the mean $\overline{f}_{\text{PREC}}$ of the ECMWF 51-member precipitation ensemble forecast for some of the models (see Sect. 4.3). The use of the HRES precipitation forecast or of the mean of the 52-member precipitation ensemble (including HRES) instead of $\overline{f}_{\text{PREC}}$ was also tested; however, these models did not result in a significant improvement in the forecast skill.

In the following, we introduce the different post-processing models for TCC. Implementation details for all models, including details on tuning parameters and parameter estimation, are provided in Sect. 4.1.

3.1 Multiclass and proportional odds logistic regression

In multiclass logistic regression, after choosing an arbitrary reference class, the log-odds of a remaining class with respect to the reference class is expressed as an affine function of the features. This means that after setting, e.g. the last okta y_9 as reference class, the conditional distribution of TCC with respect to an M-dimensional feature vector $\boldsymbol{x}$ equals

$$\mathrm{P}(Y = y_k|\boldsymbol{x}) = \begin{cases} \dfrac{\mathrm{e}^{L_k(\boldsymbol{x})}}{1 + \sum_{\ell=1}^{8} \mathrm{e}^{L_\ell(\boldsymbol{x})}}, & k = 1, 2, \ldots, 8; \\ \dfrac{1}{1 + \sum_{\ell=1}^{8} \mathrm{e}^{L_\ell(\boldsymbol{x})}}, & k = 9, \end{cases} \quad (2)$$
$$\text{with} \quad L_k(\boldsymbol{x}) := \beta_{0k} + \boldsymbol{x}^\top \beta_k,$$

where $\beta_{0k} \in \mathbb{R}, \boldsymbol{\beta}_k \in \mathbb{R}^M$, resulting in $8(M+1)$ free parameters to be estimated on the basis of the training data.

The POLR model is designed to fit ordered data such as the TCC observations at hand. Given a feature vector $\boldsymbol{x}$, the conditional cumulative probabilities of Y are expressed as

$$\mathrm{P}\big(Y \le y_k \mid \boldsymbol{x}\big) = \frac{\mathrm{e}^{\mathcal{L}_k(\boldsymbol{x})}}{1 + \mathrm{e}^{\mathcal{L}_k(\boldsymbol{x})}}, \quad (3)$$
$$\text{with} \quad \mathcal{L}_k(\boldsymbol{x}) := \gamma_{0k} + \boldsymbol{x}^\top \boldsymbol{\gamma}, \quad k = 1, 2, \ldots, 9,$$

where we assume that $\gamma_{01} < \gamma_{02} < \cdots < \gamma_{09}$. In this way, POLR model (3) is more parsimonious than MLR model (2), as it has just $9 + M$ unknown parameters.

3.2 Multilayer perceptron neural network

A multilayer perceptron (MLP) is a classical feedforward neural network, consisting of an input layer, an output layer and some intermediate layers (so-called hidden layers) comprised of several neurons each. The value in each of the neurons is a transformed value (via an activation function) of a weighted sum of all neuron values from the previous layer plus a bias term. The number of neurons in the input and output layers are uniquely determined by the number of features and number of classes, respectively, whereas the number of the hidden layers and the number of the neurons in a particular hidden layer are free (or tuning) parameters of the network. For a comprehensive introduction to neural networks, see e.g. [36].

The network is trained using a set of labelled data (training set): The weights of the neurons are determined in order to minimize a given loss function on the training set. To avoid overfitting, it is recommended to use early stopping rules based on a validation set. Typically, it is a randomly chosen subset of the labelled data set available for the training. The minimization process terminates if the value of the loss function computed on the validation set does not improve during a given number of subsequent iterations. Similar techniques are applied for the other machine learning methods, see Sect. 4.1 for details.

Another tool to prevent overfitting is the extension of the loss function with a regularization term. Here we use an L_2 regularization where the sum of squares of the weights of the network is multiplied by a factor (which is an additional tuning parameter of the network). The trained network provides for each feature vector a probability distribution corresponding to the oktas.

3.3 Random forest models and gradient boosting machines

Random forests (RF) and gradient boosting machines (GBM) are machine learning models which are both based on ensembles of decision trees. Decision trees are flowchart-like structures that have been used in meteorological forecasting since the 1950s [40]. Decision tree models are obtained through iteratively splitting training data into groups according to a threshold in one of the features $\boldsymbol{x}$ which is chosen to maximize the homogeneity of the target variable within the resulting subsets. This process is iterated until a stopping criterion is reached. Out-of-sample forecasts can be obtained by proceeding through the decision tree according to the predictor input and estimating class probabilities by the empirical frequencies of observed classes in the corresponding subset of the recursively partitioned feature space. While there exist several algorithms for decision tree learning, we will here focus on classification and regression trees first introduced by Breiman et al. [41].

3.3.1 Random forest models

To improve robustness and address overfitting issues of decision trees, random forest models [38] repeatedly resample the training set to obtain multiple decision trees. This bootstrap aggregation (or bagging) approach is used in conjunction with only considering a random subset of the predictors at each splitting node. Class probability predictions for out-of-sample cases are obtained by averaging over the decision trees in the RF ensemble.

Several tuning parameters have to be chosen when implementing RF models. Most importantly, the number of trees in the forest has to specified, and the depth (the number of levels of recursive partitioning) as well as the number of predictor variables randomly selected at each splitting node have to be selected for the individual trees. Generally, RF models are often relatively robust to these tuning parameters and tend to not be prone to overfitting for a wide range of parameter choices.

3.3.2 Gradient boosting machines

In contrast to randomly resampling the training data, gradient boosting machines consist of ensembles of decision trees which are grown sequentially, using information from previously grown trees. Thereby, each decision tree is fit on a modified version of the original training set focusing on

regions where previous model iterations provide poor predictions.

The umbrella term boosting refers to machine learning algorithms that fit models by combining several simpler models, decision trees in our case. Following [37], various notions of gradient boosting have been developed, and it was demonstrated that boosting can be interpreted as gradient descent algorithm in function space where a loss function is iteratively optimized by choosing a function that points in the negative gradient direction. Gradient boosting principles are applicable for wide range of loss functions, and corresponding algorithms have been developed for a wide range of machine learning tasks. For a general introduction to gradient boosting see e.g. [42].

We here employ a specific variant of tree-based gradient boosting called extreme gradient boosting [43], which relies on second-order approximations of the objective function. GBM model predictions are obtained via

$$\hat{z}^c = \sum_{m=1}^{M} h_m^c(\boldsymbol{x}), \tag{4}$$

where h_m^c denotes a regression tree for category $c \in \{1, \ldots, 9\}$ containing a continuous value in all terminal leaves, and M is the number of boosting iterations. For probabilistic classification tasks, separate sets of regression trees are fitted simultaneously for all categories, and the obtained latent values $\hat{z}^c$ are transformed according to a softmax function. A regularized version of the LogS (see Sect. 3.4) is used to learn the set of functions used in the model (4). For details, see [43].

Compared to RF models, GBM often provide better predictions in a variety of applications, but are more prone to overfitting and more difficult to tune. In particular, the number of boosting iterations, M, is of crucial importance. Further, the complexity of the individual trees h_m must often be restricted, see Sect. 4.1 for details.

3.4 Verification scores

As discussed in [44], the main goal of probabilistic forecasting is to maximize the sharpness of the predictive distribution subject to calibration. Sharpness refers to the concentration of the predictive distribution, whereas calibration means a statistical consistency between forecasts and observations. These two goals can be simultaneously addressed with the help of proper scoring rules, which are loss functions $\mathcal{S}(F, x)$ assigning numerical values to pairs (F, x) of forecasts and observations. As mentioned in the Introduction, in the case of TCC by forecast F we refer to a discrete probability distribution on $\mathcal{Y}$ characterized by a probability mass function (PMF) $p_F(y)$.

In the atmospheric sciences, probably the most popular proper scoring rules are the logarithmic score (LogS) [45] and the continuous ranked probability score (CRPS) [46, 47]. The former is the negative logarithm of the PMF evaluated at the observation, that is

$$\mathrm{LogS}\left(F, x\right) := -\log\left(p_F(x)\right),$$

whereas for TCC probabilistic forecasts at hand the latter can be given as

$$\mathrm{CRPS}\left(F, x\right) = \sum_{k=1}^{9} p_F(y_k)\left|y_k - x\right| - \sum_{k=2}^{9}\sum_{\ell=1}^{k-1} p_F(y_k)p_F(y_\ell)\left|y_k - y_\ell\right|,$$

which is the discrete version of the representation

$$\mathrm{CRPS}\left(F, x\right) = \mathsf{E}|X - x| - \frac{1}{2}\mathsf{E}|X - X'|,$$

where X and X' are independent random variables with distribution F and finite first moment. Both LogS and CRPS are negatively oriented, that is smaller score values indicate better predictive performance.

For a given lead time, the goodness of fit of competing TCC forecasts in terms of probability distributions are compared with the help of the mean CRPS and mean LogS values $\overline{\mathrm{CRPS}}$ and $\overline{\mathrm{LogS}}$, respectively, over all forecast cases in the verification data. Further, the improvement in CRPS and LogS with respect to a reference model can be quantified using the continuous ranked probability skill score (CRPSS) and logarithmic skill score (LogSS), respectively, defined as

$$\mathrm{CRPSS} := 1 - \frac{\overline{\mathrm{CRPS}}}{\overline{\mathrm{CRPS}}_{\mathrm{ref}}} \quad \text{and} \quad \mathrm{LogSS} := 1 - \frac{\overline{\mathrm{LogS}}}{\overline{\mathrm{LogS}}_{\mathrm{ref}}},$$

where $\overline{\mathrm{CRPS}}_{\mathrm{ref}}$ and $\overline{\mathrm{LogS}}_{\mathrm{ref}}$ denote the mean CRPS and LogS of the reference approach (see e.g. [46, 48]). Note that both CRPSS and LogSS are positively oriented, that is larger skill scores mean better predictive performance.

Further, following the suggestions of Gneiting and Ranjan [49], statistical significance of the differences between the verification scores is examined using the Diebold-Mariano (DM) [50] test, which allows accounting for the temporal dependencies in the forecast errors. In simultaneous testing for the different stations, we also address spatial dependencies by applying a Benjamini-Hochberg algorithm [51] to control the false discovery rate at a 5% level of significance (see e.g. [52]). We further provide confidence intervals for mean score values and skill scores, which are obtained with the help of 2 000 block bootstrap samples using the stationary bootstrap

scheme with mean block length determined according to [53].

Finally, a simple tool of visual perception of calibration is the probability integral transform (PIT) histogram, where the PIT is defined as the value of the predictive cumulative distribution (CDF) at the validating observation, with a possible randomization at points of discontinuity [54]. In the case of proper calibration, PIT should follow a uniform distribution on the [0, 1] interval; moreover, if uniformity fails to be achieved, the shape of the PIT histogram provides information about the possible reason of the problem.

4 Results

All calibration approaches presented in Sect. 3 require training data which should be large enough to provide numerical stability and reasonable predictive performance. Following [39], we here focus on local calibration, i.e. post-processing of forecasts for a given station is performed using only training data of that particular station. Therefore, relatively long training periods are required to achieve a suitably large training set. In order to ensure comparability with the reference approaches, we consider 5-year training periods and both non-seasonal and seasonal training schemes as in [30]. In the non-seasonal training, forecasts and observations of 5 calendar years (e.g. 1 January 2003–31 December 2007) are used to train the model for calibration of TCC ensemble forecasts for the whole next calendar year (1 January–31 December 2008), then the training period is rolled ahead by one year (1 January 2004–31 December 2008). In the seasonal approach, two different seasons are considered covering April–September and October–March, and TCC ensemble forecast for a given day is calibrated using training data from the same season only. The use of 5-year training periods means that predictive PMFs are available for the time interval between January 1, 2007, and March 20, 2014 (2636 calendar days), where one can test the forecast skill of the post-processing methods presented in Sect. 3.

Further, as suggested by Hemri et al. [30], numerical problems with LogS calculation are avoided by replacing unrealistically low values $p_F(y_j)$ of the predictive PMF corresponding to okta y_j with a probability $p_{\min}$ ensuring that with a 1% chance one observes okta y_j at least once during the training period. Translated to formulae, this means that instead of $p_F(y_j)$ we consider $\max\{p_{\min}, p_F(y_j)\}$, where $p_{\min}$ solves $0.01 = 1 - (1 - p_{\min})^T$ with T being the length of the training period in days and adjust the probabilities to get a PMF again (for more details see [30]). Note that this is only a minor technical adjustment, and compared with the original predictive PMFs it results in negligible differences in CRPS or PIT values.

4.1 Implementation details

Here, we discuss implementation details for the different statistical and machine learning methods for TCC post-processing.

4.1.1 Multiclass and proportional odds logistic regression

Both models have several implementations. Here, coefficients of the various MLR and POLR models are estimated with the help of `R` packages `nnet` and `MASS` [55], respectively. Note that the implementation based on the `nnet` package utilizes neural networks for estimating the parametric MLR model (2) which is a fundamentally different use of neural networks compared to our MLP models introduced in Sect. 3.2.

4.1.2 Multilayer perceptron neural networks

In our computations, we apply the `patternnet` function of `Matlab` with two hidden layers, consisting of 10 and 15 neurons. Both hidden layers use the hyperbolic tangent as activation function. We consider the LogS as loss function (sometimes termed cross-entropy in the machine learning literature) with a 0.1 regularization parameter and scaled conjugate gradient as minimization algorithm. In each 5-year training period (both for the seasonal and non-seasonal approaches), the corresponding data set is split into a training and validation set, the latter is a randomly selected subset consisting of 15% of the data. As an alternative to the 5-year rolling training period, training with a growing data set using all available forecast cases from the previous years and simultaneously increasing the weight of the regularization term was also tested. However, this approach did not result in an improved forecast skill.

4.1.3 Random forests

Our implementation of RF models is based on the `R` package `XGBoost` [56]. The tuning parameters (depth of trees, number of predictors sub-sampled at each splitting node) for a specific observation station and forecast horizon are determined as follows. The first of the rolling 5-year training periods consisting of the years 2002–2006 is split into an initial training set (years 2002–2005) and a validation set (year 2006). For all combinations of tree depths between 2 and 4, and numbers of predictors between 1 and 3, RF models consisting of 300 trees are estimated based on the initial training set and evaluated on the validation set using the LogS. The combination of tuning

parameters resulting in the lowest LogS on the validation set is then used to fit a RF model consisting of 1000 trees for the full training set (years 2002–2005) and to produce forecasts for the first out-of-sample test set (year 2007). To limit computational costs, this optimal combination of tuning parameters is also used for all subsequent 5-year training periods for that specific station and lead time.

For rolling 5-year training periods, tree depths of 2, 3 and 4 are selected in around 43%, 36% and 21% of the cases, respectively. The chosen number of predictors for subsampling is slightly more evenly distributed, and the most frequently selected tuning parameter combination consists of trees of depth 3 with 3 predictors sub-sampled at each split (around 17% of all cases). Note that since initial tests did not indicate improvements in predictive performance and RF models often tend to be relatively robust to the choice of tuning parameters, we did not consider a more extensive set of possible parameter values in order to limit computational costs.

4.1.4 Gradient boosting machines

We implement GBM models based on the `R` package `XGBoost` [56]. Throughout, we use shrinkage with a learning rate of $\lambda = 0.1$ which reduces the influence of each individual tree h_m^c by adding a scaled version of that tree only. To further prevent overfitting, we determine the number of boosting iterations M for a fixed tree depth by using an early stopping criterion. To that end, each 5-year training set is split into an initial training set (first 4 years) and a validation set (last year). GBM models of the form (4) are then estimated iteratively for $m = 1, 2, \ldots$ based on the initial training set until the LogS on the validation set has not improved during the last 25 iterations. This process is repeated for all tree depth values between 1 and 4, and the combination of tree depth and corresponding optimal number of boosting iterations that results in the best LogS on the training set is selected as set of tuning parameters. The final out-of-sample forecasts for the test set are produced based on a GBM model fitted on the full training set using these tuning parameters. A separate set of tuning parameters is determined according to the procedure described above for any combination of station and lead time and any of the rolling 5-year training periods.

For models with a rolling 5-year training period, an optimal tree depth of 1 is selected for around 86.5% of all GBM models, a depth of 2 in around 11.5% of the cases and a depth of 3 or 4 in less than 2%. The average number of boosting iterations is 78.3, but generally depends on the corresponding tree depth.

The procedures to determine optimal tuning parameters of RF and GBM models described above are applied separately to the two seasons when fitting seasonal RF and GBM models. Therefore, the sets of optimal tuning parameters differ not only by station, lead time and year (only for GBM), but also by season for those variants.

4.2 Post-processing of TCC ensemble forecasts

As a first step, we investigate the post-processing of TCC ensemble forecasts using the MLP, RF and GBM approaches. As references, we consider the raw TCC ensemble forecast and the MLR and POLR models. All calibrated forecasts are based on the 7-dimensional feature vector $(\overline{f}_{\text{ENS}}, f_{\text{CTRL}}, f_{\text{HRES}}, s^2, p_0, p_1, I)^\top$ except the MLR, where following [30] the number of parameters is reduced by omitting the interaction term I. Note that the MLP model was also tested with the 52-member TCC forecast ensemble as feature vector; however, this approach did not result in an improved predictive performance. Further, following again the suggestions of [30], in the POLR model the coefficients of $\overline{f}_{\text{ENS}}, f_{\text{CTRL}}$ and f_{HRES} are forced to be non-negative by iterative exclusion of covariates with negative weights. Finally, for all five calibration methods we test both non-seasonal and seasonal training, forecasts obtained using the latter are referred as MLPS, RFS, GBMS, MLRS and POLRS, respectively.

Figure 1 shows the mean CRPS and LogS of the raw ensemble and post-processed forecasts together with 95% confidence intervals as functions of the lead time. All calibrated TCC forecasts outperform the raw ensemble by a wide margin and one can observe a clear grouping of the various approaches. The first group, resulting in the lowest mean CRPS and LogS values, consists of the MLP, GBM, POLR and MLR methods and their seasonally estimated versions showing very small differences in forecast skill. The second group contains the non-seasonally and seasonally estimated RF forecasts, where the latter results in slightly lower score values than the former.

One can compare the performance of the forecasts in the first group more easily by examining Fig. 2, where the CRPSS and LogSS with respect to the POLRS forecasts are plotted, which showed the best forecast skill among the methods studied in [30]. According to Fig. 2a, in terms of the mean CRPS, POLRS outperforms its competitors up to day 7, whereas for longer lead times MLPS has the best predictive performance. In general, forecasts based on seasonal training result in lower mean CRPS than their non-seasonal counterparts; however, the differences decrease with the increase of the lead time. Results in terms of the LogS shown in Fig. 2b indicate a different behaviour and ranking of the models in that the mean LogS of the MLPS approach reaches that of the POLRS model only at

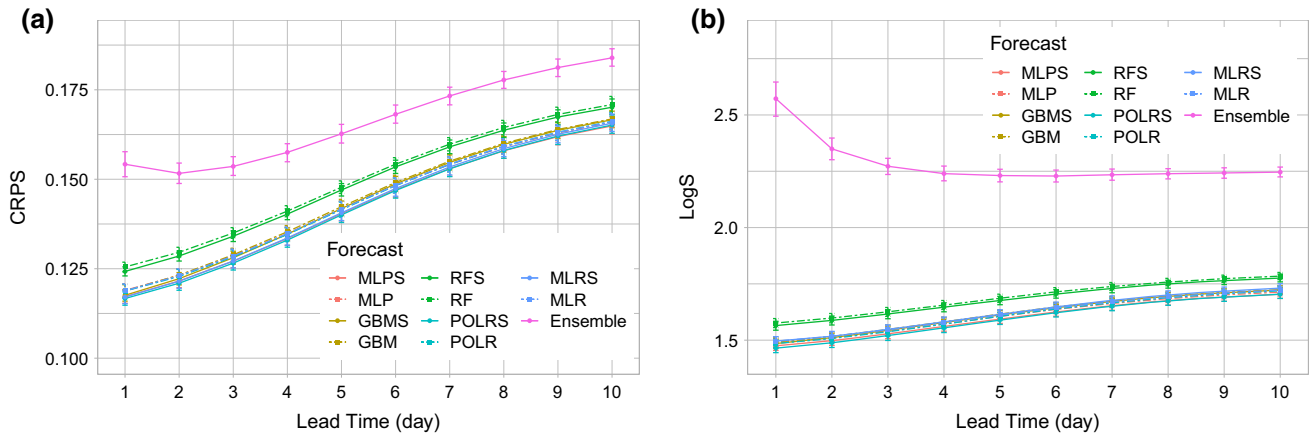

Fig. 1 Mean CRPS (**a**) and LogS (**b**) of the raw ensemble and post-processed forecasts together with 95% confidence intervals

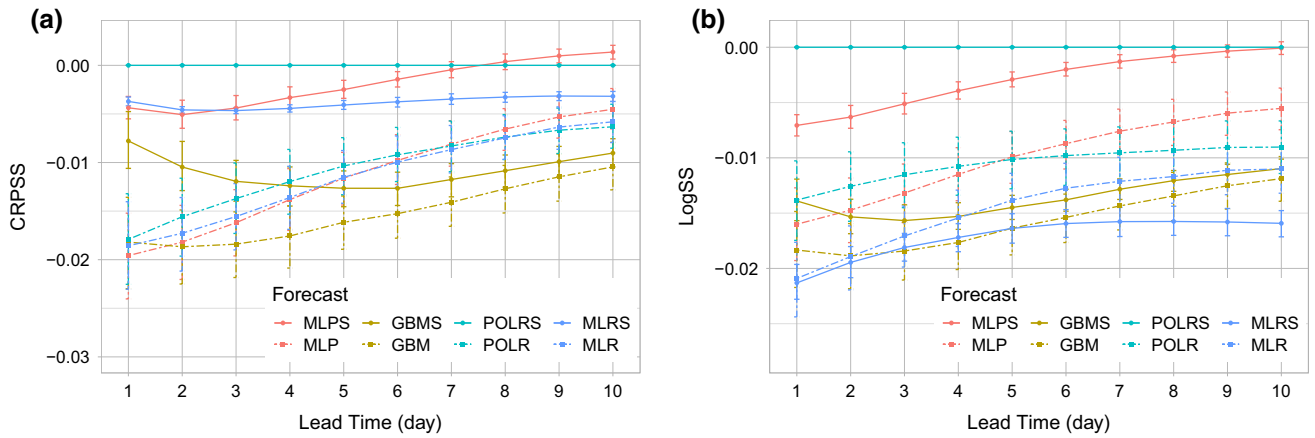

Fig. 2 CRPSS (**a**) and LogSS (**b**) with respect to the POLRS model of MLPS, MLP, GBMS, GBM, MLRS, MLR and POLR forecasts together with 95% confidence intervals

day 10 and MLRS underperforms all other methods for all lead times.

These observations are further supported by Fig. 3 showing the proportion of stations where DM test indicates significant difference in mean CRPS and LogS for lead times 1, 4, 7 and 10 days. To simplify the presentation here, we compare just the raw ensemble and the seasonally trained versions of the calibration approaches, as in general seasonal models outperform their non-seasonal counterparts. Raw ensemble and RFS forecasts are clearly separated from the other four approaches for all lead times, as most entries of the corresponding cells are close to 100%. For longer lead times GBMS also differs significantly from its competitors in almost all stations both in terms of CRPS and LogS. On the contrary, the increase of the lead time reduces the proportion of stations where the mean LogS of MLPS and POLRS forecast differ, whereas in terms of the mean CRPS after decrease one can observe a slight increase. This behaviour is in line with the MLPS skill scores of Fig. 2a and b, respectively. Overall, we note that even though the absolute differences in terms of CRPS and LogS between the different methods are relatively small, they thus are often statistically significant for a large proportion of the stations.

The positive effect of post-processing can also be observed in the PIT histograms in Fig. 4, where again, only the results for better performing seasonally trained models are shown. The U-shaped histograms of the raw ensemble at days 1 and 4 clearly indicate underdispersion, whereas at days 7 and 10 a small hump starts to appear. RFS forecasts are overdispersive for short lead times and develop some bias as the forecast horizon increases. GBMS forecasts exhibit the same behaviour, however, to a much smaller extent. The PIT histograms of POLRS and MLPS are almost perfectly flat, indicating a better calibration compared to the other methods.

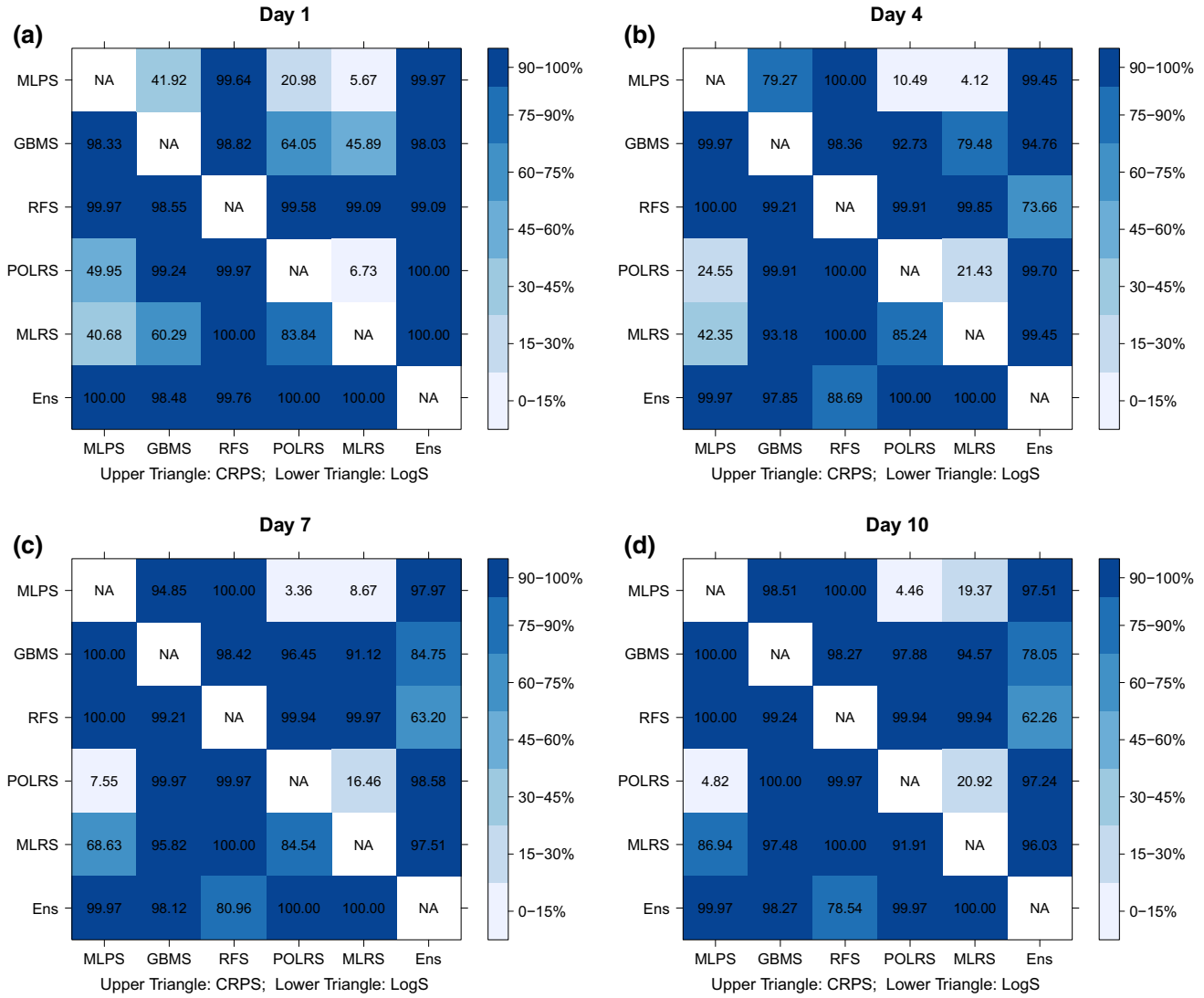

Fig. 3 Proportion of stations with significantly different mean CRPS (upper triangle) and LogS (lower triangle) at a 5% level of significance for lead times 1 (**a**), 4 (**b**), 7 (**c**) and 10 (**d**) days

4.3 Post-processing using an extended feature set

The added value of incorporating additional features based on geographical data of SYNOP stations and/or forecasts of other weather variables has been demonstrated in various recent articles on post-processing (e.g. [24, 26, 28]). Due to the direct connection to clouds [57], functionals of precipitation ensemble forecasts represent a natural choice for additional predictors. We here use the mean $\overline{f}_{\mathrm{PREC}}$ of the ECMWF 51-member precipitation forecast as additional covariate and investigate the performance of MLP, GBM and POLR approaches, showing the best forecast skill in Sect. 4.2, with extended feature vector

$$\left(\overline{f}_{\mathrm{ENS}}, f_{\mathrm{CTRL}}, f_{\mathrm{HRES}}, s^2, p_0, p_1, I, \overline{f}_{\mathrm{PREC}}\right)^{\top}.$$

Again, we consider both non-seasonal and seasonal training, the corresponding models are referred as MLP-P, GBM-P, POLR-P and MLPS-P, GBMS-P, POLRS-P, respectively.

According to Fig. 5a, b, where the mean CRPS and LogS values of different MLP, GBM and POLR forecasts are plotted as functions of the lead time, and Fig. 5c, d showing the corresponding skill scores with respect to the POLRS model, the additional covariate results in different effects for the MLP, and the GBM and POLR models. After day 2 MLP models using also precipitation forecasts significantly outperform MLP models based on TCC forecasts only in terms of both CRPS and LogS regardless of the training scheme (MLP is not shown), moreover, for longer lead times MLPS-P and MLP-P show the best

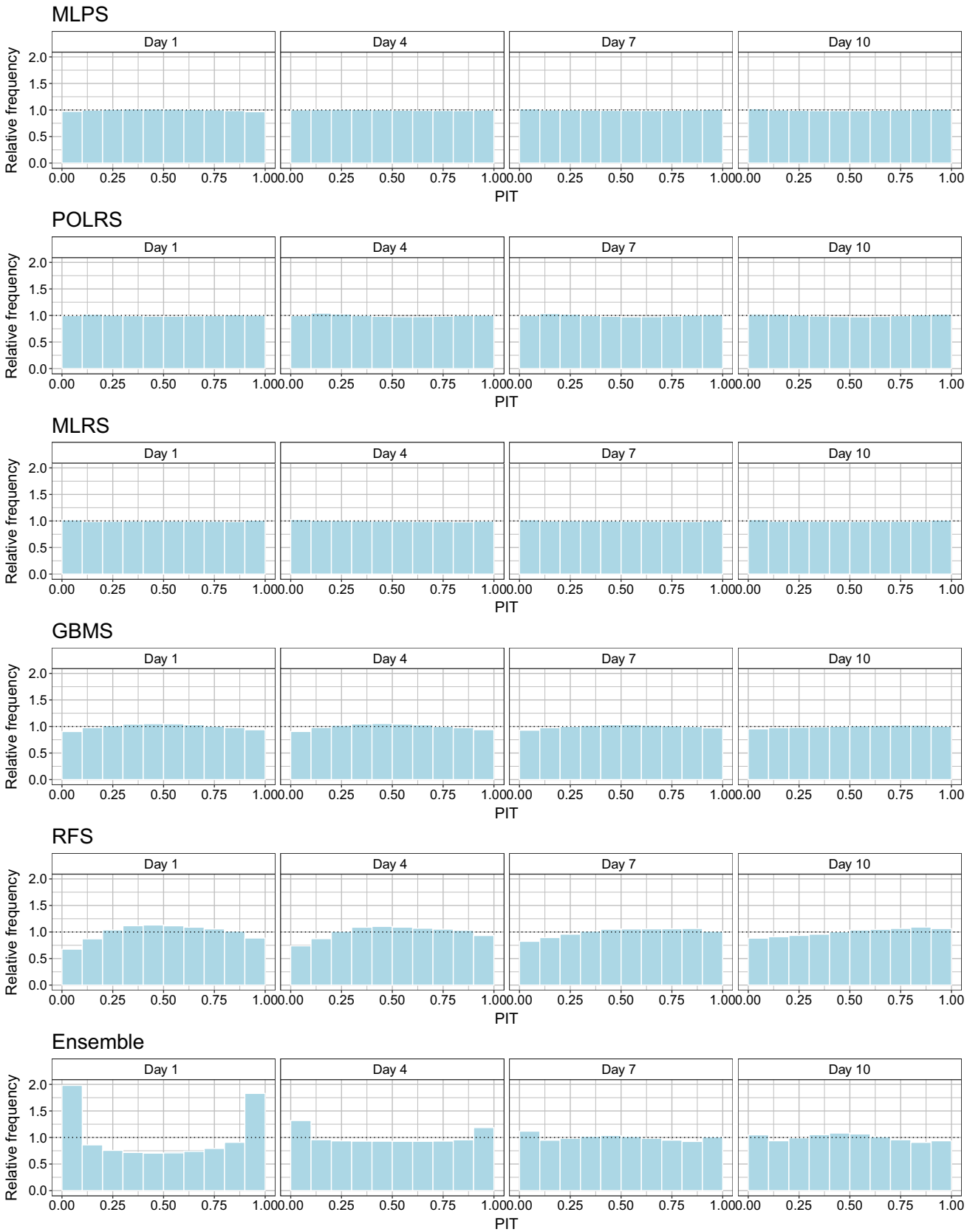

Fig. 4 PIT histograms over all stations and dates (3300 stations, 2636 days) of the seasonally trained calibration approaches and the raw ensemble at days 1, 4, 7 and 10

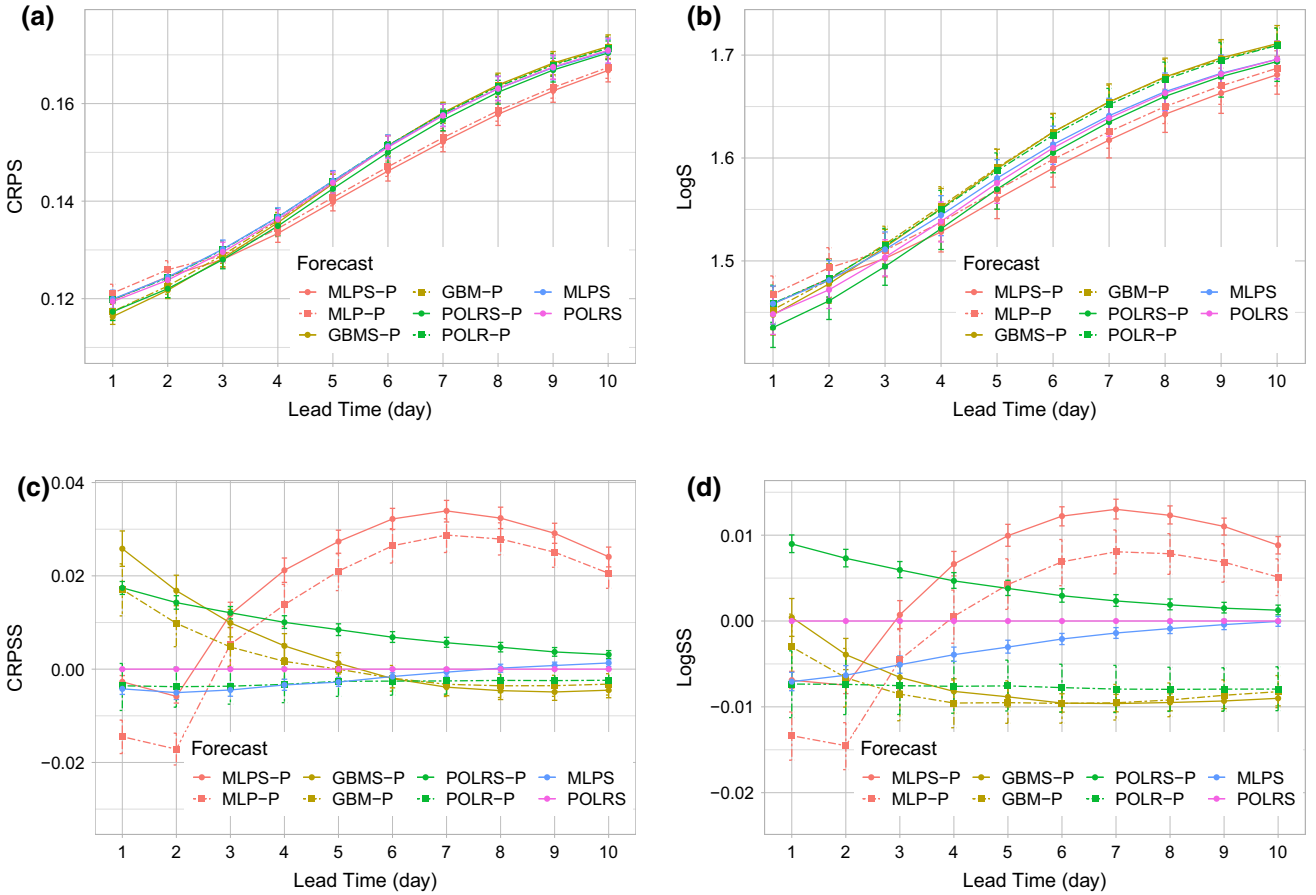

Fig. 5 CRPS (**a**) and LogS (**b**) of different MLP, GBM and POLR forecasts and the corresponding skill scores with respect to the POLRS model (**c**, **d**) together with 95% confidence intervals

predictive performance. In contrast, the use of precipitation has the highest effect on POLR models at day 1, and the differences between POLRS-P and POLRS and POLR-P and POLR models (POLR is not shown) are decreasing with the increase of the lead time. The same phenomenon can be observed with GBMS and GBM models (not shown). The use of precipitation forecast substantially improves the predictive performance; however, the difference decreases with the increase of the lead time. Up to day 5, GBMS-P and GBM-P approaches result in lower mean CRPS than the POLRS model, whereas for days 1 and 2 GBMS-P outperforms POLRS-P and MLPS-P.

These results are in line with proportions of stations with significantly different mean CRPS and LogS values provided in Fig. 6, where we consider only the models with the extended feature set in the interest of visual clarity. For instance, the proportion of stations where the mean CRPS of GBMS-P and GBM-P models differ shows a monotone decreasing sequence of 38.59%, 31.80%, 28.18%, 20.99%, mimicking the decreasing distance of the corresponding curves in Fig. 5c, while the bow of the CRPSS of MLPS-P with respect to POLRS and the decrease of the CRPSS of POLRS-P matches the change of the corresponding entries (68.65%, 26.98%, 78.70%, 75.72%) in Fig. 6.

Addressing calibration, Fig. 7 shows the PIT histograms of the calibration approaches using precipitation forecasts at days 1, 4, 7 and 10. In general, all six methods result in rather well calibrated predictive PMFs for all lead times. The histograms of GBMS-P and GBM-P approaches are overdispersive for all lead times, whereas MLPS-P, MLP-P and POLR-P are slightly overconfident only at day 1, which transforms to a small underdispersion at longer lead times. Note that in contrast to Fig. 4, which is based on verification data of 3330 locations; here, we consider PIT values for just 2239 SYNOP stations where precipitation ensemble forecasts are also available. However, this reduction does not change the general shape of the PIT histograms of the raw ensemble and the MLPS, GBMS and POLRS forecasts, so they are not shown in this case. Finally, the general behaviour of the MLPS, MLP, GBMS, GBM, POLRS and POLR forecasts in terms of PIT values is almost completely inherited to the corresponding MLPS-

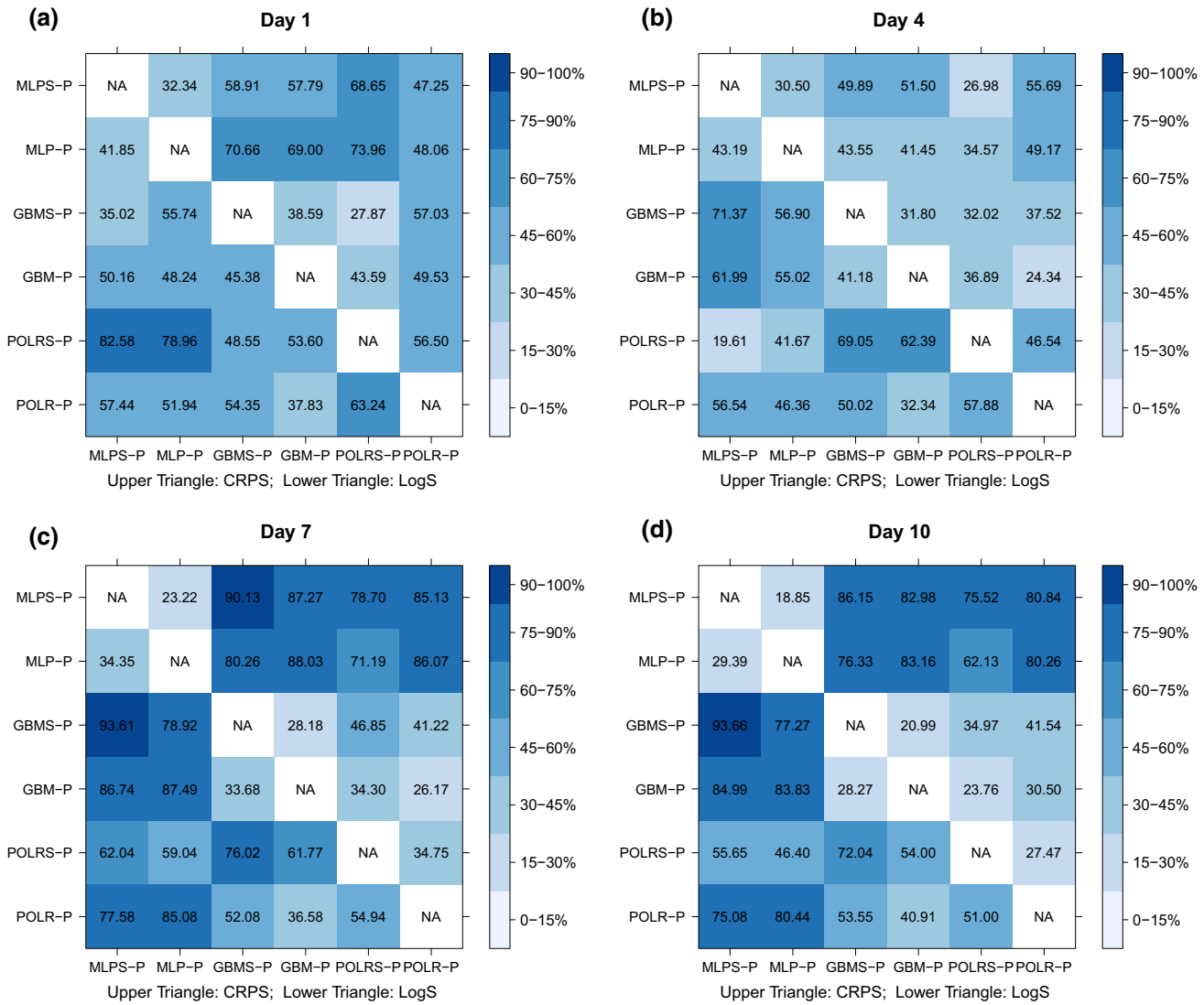

Fig. 6 Proportion of stations with significantly different mean CRPS (upper triangle) and LogS (lower triangle) at a 5% level of significance for lead times 1 (**a**), 4 (**b**), 7 (**c**) and 10 (**d**) days

P, MLP-P, GBMS-P, GBM-P, POLRS-P and POLR-P approaches.

5 Discussion

We investigate various machine learning classifiers for statistical post-processing of total cloud cover ensemble forecasts. In particular, we consider multilayer perceptron neural networks, random forest methods and gradient boosting machines, which are tested on ECMWF global TCC ensemble forecasts with lead times from 1 to 10 days and the corresponding discrete SYNOP observations. Raw TCC ensemble forecasts, multiclass and proportional odds logistic regression are used as reference models, and we consider both seasonal and non-seasonal training (following [30]).

First we investigate the settings of [30], where the classification is based on predictors calculated from the TCC ensemble forecasts only. In general, all post-processing methods significantly outperform the raw ensemble for all lead times both in term of the mean CRPS and the mean LogS over the verification data, and the corresponding PIT histograms are closer to the uniform distribution than those of the raw forecasts. Seasonally trained models further result in slightly better predictive performance than their non-seasonal counterparts. RF models underperform their competitors, whereas the difference between MLP, GBM, POLR and MLR approaches are generally small. For short and medium forecast horizons, the POLR model with seasonal training occurs to be the

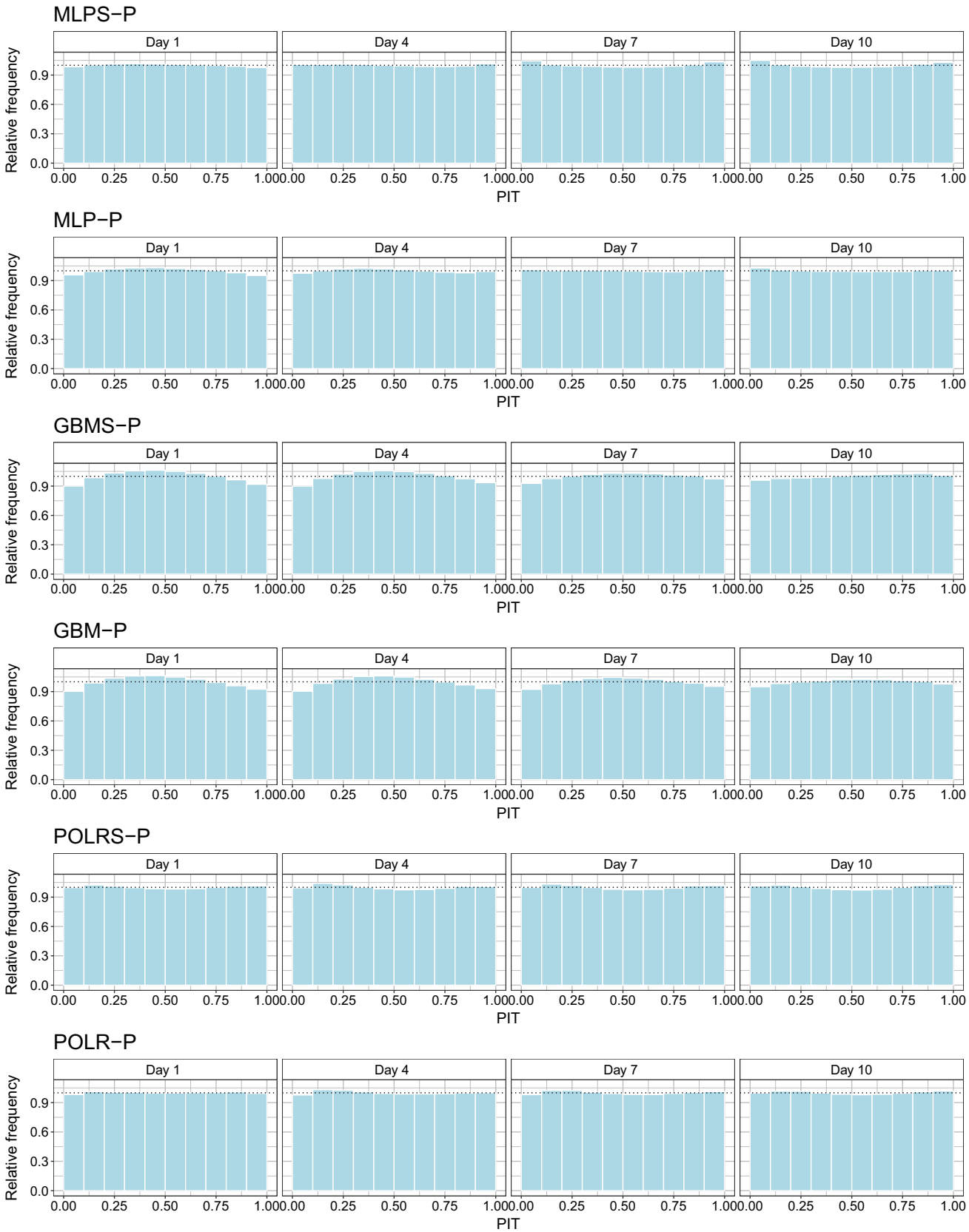

Fig. 7 PIT histograms over all stations and dates (2239 stations, 2636 days) of the calibration approaches using precipitation forecasts at days 1, 4, 7 and 10

most skillful, closely followed by the seasonally trained MLP model which performs best for long lead times. Several of the probabilistic classification methods exhibit complementary systematic errors in calibration. Therefore, forecast combination techniques along the lines of [58] could potentially improve predictive performance. The related topic of calibrating and combining probabilistic classifiers has recently received some interest in the machine learning literature, see e.g. [59].

Due to the flexibility of neural network model architectures, particularly the MLP model provides several promising starting points for future extensions. For example, long short-term memory neural networks [60] are widely used for time series modelling and may allow to incorporate temporal dependencies of forecast errors of the raw ensemble predictions. Further, techniques along the lines of station embeddings proposed in [26] could potentially help construct a single MLP model jointly for all stations which still is locally adaptive.

The use of the mean precipitation accumulation as additional covariate further improves the predictive performance and changes the ranking of the different methods. With this extended feature set, the seasonal POLR model exhibits the best overall performance only for short lead times; after days 3–4 it is significantly outperformed both by the seasonally and non-seasonally trained MLP. However, in general, the advantage of the extended set of covariates fades with the increase of the lead time.

The improved performance when information on precipitation is added further indicates advantages of modern machine learning methods such as GBM and MLP for total cloud cover prediction. By contrast to the classical MLR and POLR approaches, these methods allow to add additional predictors in a straightforward manner and provide tools for avoiding overfitting. The inclusion of further predictor variables such as, for example, indices of atmospheric stability, pressure, humidity and temperature information at upper levels of the atmosphere, or seasonal information may further improve predictive performance. Further, more complex machine learning models incorporating many predictors may not only improve TCC predictions, but may also allow to better understand the shortcomings of the raw ensemble predictions utilizing techniques such as measures of feature importance [26, 38].

6 Conclusions

This paper provides a new approach to statistical post-processing of TCC ensemble forecasts using various machine learning based classification methods. According to the best knowledge of the authors, this is the first work to compare the state-of-the-art machine learning approaches with the parametric classification techniques. Via an extended case study based on the ECMWF global TCC ensemble forecasts for 2002–2014, the superiority of neural network classification over the best parametric models is shown [30]. The possibility of involving additional covariates into statistical post-processing of TCC is also studied. The results indicate that when the mean precipitation accumulation is used as additional covariate, for long lead times multilayer perceptron neural network classification exhibits far the best predictive performance. The flexibility of neural network models and the wide range of reasonable covariates opens a gate for further investigations. These studies might lead to direct economic benefit as more accurate prediction of TCC plays a fundamental role, e.g. in energy production, agriculture or tourism.

Acknowledgements Open access funding provided by University of Debrecen. Essential part of this work was made during the visit of Sándor Baran at the Heidelberg Institute of Theoretical Studies. Sándor Baran further received support from the National Research, Development and Innovation Office under Grant No. NN125679. Ágnes Baran and Sándor Baran were supported by the EFOP-3.6.2-16-2017-00015 project. The project was co-financed by the Hungarian Government and the European Social Fund. Ágnes Baran and Mehrez El Ayari were supported by the EFOP-3.6.3-VEKOP-16-2017-00002 project. The project was co-financed by the Hungarian Government and the European Social Fund. Sebastian Lerch was further supported by the Deutsche Forschungsgemeinschaft through SFB/TRR 165 "Waves to Weather." The authors are grateful to Stephan Hemri for providing the data and for useful suggestions and comments. Last but not least, the authors are indebted to the reviewers and the editor for their valuable comments.

Availability of data and materials The data used in this study are proprietary, and the authors are not allowed to share it. However, it may be obtained from the European Centre for Medium-Range Weather Forecasts directly for research purposes.

Compliance with ethical standards

Conflict of interest No conflict of interest to be declared.

References

1. Ye QZ, Chen SS (2013) The ultimate meteorological question from observational astronomers: how good is the cloud cover forecast? Mon Not R Astron Soc 428:3288–3294
2. Matuszko D (2012) Influence of the extent and genera of cloud cover on solar radiation intensity. Int J Climatol 32:2403–2414
3. McEvoy A, Markvart T, Castañer L (eds) (2012) Practical handbook of photovoltaics. Fundamentals and applications, 2nd edn. Academic Press, Waltham
4. World Meteorological Organization (2017) International Cloud Atlas. Manual on the Observation of Clouds and Other Meteors. WMO-No. 407. https://cloudatlas.wmo.int/home.html. Accessed 13 March 2020
5. Køltzov M, Casati B, Bazile E, Haiden T, Valkonen T (2019) An NWP model intercomparison of surface weather parameters in the European Arctic during the Year of Polar Prediction Special Observing Period Northern Hemisphere 1. Wea Forecasting 34:959–983
6. Zhou X, Zhu Y, Hou D, Luo Y, Peng J, Wobus R (2017) Performance of the new NCEP Global Ensemble Forecast System in a parallel experiment. Wea Forecasting 32:1989–2004
7. Molteni F, Buizza R, Palmer TN, Petroliagis T (1996) The ECMWF ensemble prediction system: methodology and validation. Q J R Meteorol Soc 122:73–119
8. Leutbecher M, Palmer TN (2008) Ensemble forecasting. J Comput Phys 227:3515–3539
9. ECMWF Directorate (2012) Describing ECMWF's forecasts and forecasting system. ECMWF Newsl 133:11–13
10. Gneiting T, Raftery AE (2005) Weather forecasting with ensemble methods. Science 310:248–249
11. Buizza R, Houtekamer PL, Toth Z, Pellerin G, Wei M, Zhu Y (2005) A comparison of the ECMWF, MSC, and NCEP global ensemble prediction systems. Mon Weather Rev 133:1076–1097
12. Park Y-Y, Buizza R, Leutbecher M (2008) TIGGE: preliminary results on comparing and combining ensembles. Q J R Meteorol Soc 134:2029–2050
13. Buizza R (2018) Ensemble forecasting and the need for calibration. In: Vannitsem S, Wilks DS, Messner JW (eds) Statistical postprocessing of ensemble forecasts. Elsevier, Amsterdam, pp 15–48
14. Haiden T, Forbes R, Ahlgrimm M, Bozzo A (2015) The skill of ECMWF cloudiness forecasts. ECMWF Newsl 143:14–19
15. Haiden T, Janousek M, Bidlot J, Buizza R, Ferranti L, Prates F, Vitart F (2018) Evaluation of ECMWF forecasts, including the 2018 upgrade. ECMWF Technical Memorandum No 831. https://www.ecmwf.int/sites/default/files/elibrary/2018/18746-evaluation-ecmwf-forecasts-including-2018-upgrade.pdf
16. Williams RM, Ferro CAT, Kwasniok F (2014) A comparison of ensemble post-processing methods for extreme events. Q J R Meteorol Soc 140:1112–1120
17. Vannitsem S, Wilks DS, Messner JW (eds) (2018) Statistical postprocessing of ensemble forecasts. Elsevier, Amsterdam
18. Raftery AE, Gneiting T, Balabdaoui F, Polakowski M (2005) Using Bayesian model averaging to calibrate forecast ensembles. Mon Weather Rev 133:1155–1174
19. Gneiting T, Raftery AE, Westveld AH, Goldman T (2005) Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. Mon Weather Rev 133:1098–1118
20. Friederichs P, Hense A (2007) Statistical downscaling of extreme precipitation events using censored quantile regression. Mon Weather Rev 135:2365–2378
21. Bremnes JB (2019) Constrained quantile regression splines for ensemble postprocessing. Mon Weather Rev 147:1769–1780
22. Hamill TM, Scheuerer M (2018) Probabilistic precipitation forecast postprocessing using quantile mapping and rank-weighted best-member dressing. Mon Weather Rev 146:4079–4098
23. Gascón E, Lavers D, Hamill TM, Richardson DS, Ben Bouallègue Z, Leutbecher M, Pappenberger F (2019) Statistical post-processing of dual-resolution ensemble precipitation forecasts across Europe. Q J R Meteorol Soc 145:3218–3235
24. Taillardat M, Mestre O, Zamo M, Naveau P (2016) Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. Mon Weather Rev 144:2375–2393
25. Taillardat M, Fougères A-L, Naveau P, Mestre O (2019) Forest-based and semi-parametric methods for the postprocessing of rainfall ensemble forecasting. Wea Forecasting 34:617–634
26. Rasp S, Lerch S (2018) Neural networks for postprocessing ensemble weather forecasts. Mon Weather Rev 146:3885–3900
27. Bremnes JB (2020) Ensemble postprocessing using quantile function regression based on neural networks and Bernstein polynomials. Mon Weather Rev 148:403–414
28. Bakker K, Whan K, Knap W, Schmeits M (2019) Comparison of statistical post-processing methods for probabilistic NWP forecasts of solar radiation. Sol Energy 191:138–150
29. Gneiting T, Katzfuss M (2014) Probabilistic forecasting. Annu Rev Stat Appl 2014.1:125–151
30. Hemri S, Haiden T, Pappenberger F (2016) Discrete postprocessing of total cloud cover ensemble forecasts. Mon Weather Rev 144:2565–2577
31. Izenman AJ (2008) Modern multivariate statistical techniques. Regression, classification and manifold learning. Springer, New York
32. McCullagh P (1980) Regression model for ordinal data (with discussion). J R Stat Soc Ser B Stat Methodol 42:243–268
33. Wilks DS (2009) Extending logistic regression to provide full-probability-distribution MOS forecasts. Meteorol Appl 16:361–368
34. Schmeits MJ, Kok KJ (2010) A comparison between raw ensemble output, (modified) Bayesian model averaging, and extended logistic regression using ECMWF ensemble precipitation reforecasts. Mon Weather Rev 138:4199–4211
35. Messner JW, Mayr GJ, Wilks DS, Zeileis A (2014) Extending extended logistic regression: Extended versus separate versus ordered versus censored. Mon Weather Rev 142:3003–3014
36. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
37. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29:1189–1232
38. Breiman L (2001) Random forests. Mach Learn 45:5–32
39. Hemri S, Scheuerer M, Pappenberger F, Bogner K, Haiden T (2014) Trends in the predictive performance of raw ensemble weather forecasts. Geophys Res Lett 41:9197–9205
40. McGovern A, Elmore KL, Gagne DJ, Haupt EE, Karstens CD, Lagerquist R, Smith T, Williams JK (2017) Using artificial intelligence to improve real-time decision-making for high-impact weather. Bull Am Meteorol Soc 98:2073–2090
41. Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth International Group, Belmont
42. Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning. Data mining, inference, and prediction, 2nd edn. Springer, New York
43. Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. Working paper. https://arxiv.org/abs/1603.02754. Accessed 13 March 2020
44. Gneiting T, Balabdaoui F, Raftery AE (2007) Probabilistic forecasts, calibration and sharpness. J R Stat Soc Ser B Stat Methodol 69:243–268

45. Good IJ (1952) Rational decisions. J R Stat Soc Ser B Stat Methodol 14:107–114
46. Gneiting T, Raftery AE (2007) Strictly proper scoring rules, prediction and estimation. J Am Stat Assoc 102:359–378
47. Wilks DS (2011) Statistical methods in the atmospheric sciences, 3rd edn. Elsevier, Amsterdam
48. Murphy AH (1973) Hedging and skill scores for probability forecasts. J Appl Meteorol 12:215–223
49. Gneiting T, Ranjan R (2011) Comparing density forecasts using threshold- and quantile-weighted scoring rules. J Bus Econ Stat 29:411–422
50. Diebold FX, Mariano RS (1995) Comparing predictive accuracy. J Bus Econ Stat 13:253–263
51. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B Stat Methodol 57:289–300
52. Wilks DS (2016) "The stippling shows statistically significant grid points": How research results are routinely overstated and overinterpreted, and what to do about it. Bull Am Meteor Soc 97:2263–2273
53. Politis DN, Romano JP (1994) The stationary bootstrap. J Am Stat Assoc 89:1303–1313
54. Gneiting T, Ranjan R (2013) Combining predictive distributions. Electron J Stat 7:1747–1782
55. Venables WN, Ripley BD (2002) Modern applied statistics with S, 4th edn. Springer, New York
56. Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, Chen K, Mitchell R, Cano I, Zhou T, Li M, Xie J, Lin M, Geng Y, Li Y (2019) xgboost: Extreme gradient boosting. R package version 0.90.0.1. https://CRAN.R-project.org/package=xgboost. Accessed 13 March 2020
57. Mishra AK (2019) Investigating changes in cloud cover using the long-term record of precipitation extremes. Meteorol Appl 26:108–116
58. Baran S, Lerch S (2018) Combining predictive distributions for the statistical post-processing of ensemble forecasts. Int J Forecast 34:477–496
59. Kull M, Nieto MP, Kängsepp M, Silva FT, Song H, Flach P (2019) Beyond temperature scaling: obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R (eds) Advances in neural information processing systems 32. Curran Associates Inc, Red Hook, pp 12295–12305
60. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780