



Forecasting Low-Visibility Procedure States with Tree-Based Statistical Methods

SEBASTIAN J. DIETZ,¹  PHILIPP KNERINGER,¹ GEORG J. MAYR,¹ and ACHIM ZEILEIS²

Abstract—Low-visibility conditions at airports can lead to capacity reductions and, therefore, to delays or cancelations of arriving and departing flights. Accurate visibility forecasts are required to keep the airport capacity as high as possible. We generate probabilistic nowcasts of low-visibility procedure (lvp) states, which determine the reduction of the airport capacity due to low visibility. The nowcasts are generated with tree-based statistical models based on highly-resolved meteorological observations at the airport. Short computation times of these models ensure the instantaneous generation of new predictions when new observations arrive. The tree-based ensemble method boosting provides the highest benefit in forecast performance. For lvp forecasts with lead times shorter than 1 h variables with information of the current lvp state, ceiling, and horizontal visibility are most important. With longer lead times, visibility information of the airport's vicinity and standard meteorological variables such as humidity also become relevant.

Key words: Aviation meteorology, visibility, nowcast, decision tree, bagging, random forest, boosting.

1. Introduction

Low-visibility conditions reduce the operational capacity of airports. At peak hours, capacity reductions may lead to flight delays or even cancelations. Consequently, costs for airports and airlines as well as the environmental impact increase. These effects grow considerably when low visibility is predicted incorrectly. Pessimistic visibility predictions may cause an overly strong reduction of the airport capacity. Hence, scheduled short-distance flights may be kept on ground at their airport of departure to

ensure the flight safety. This action results in costs for airlines, as well as decreased revenues from landing fees for airports. On the other hand, if visibility is predicted too optimistically, the number of arriving aircraft might exceed the remaining capacity of an airport. Consequently, many en route flights have to circle into stacks until landing is possible. Results are increased costs for airlines through crew scheduling, fuel consumption, and emissions. The costs for delays depend on the type of aircraft and the type of delay (tactical or strategic delay; delays at taxi, en-route or gate) and vary, therefore, over a wide range (University of Westminster 2015). Accurate predictions of low-visibility states, however, can help to reduce such supplemental costs. Precise forecasts can be used by air traffic controllers to plan the number of arriving and departing aircraft in a safe and economical way.

The reduction of aircraft movements due to low visibility depends on different safety operations, which decelerate the air traffic and increase the distance between successive arrivals and departures. The execution of these operations is defined by low-visibility procedure (lvp) states. Upcoming available capacity is, therefore, directly connected to lvp forecasts at airports. Typically, lvp states are determined by particular thresholds of horizontal and vertical visibility. The exact thresholds and resulting capacity reductions, however, vary for each airport. In this study, lvp forecasts are investigated for Vienna International Airport (VIE). Nowcasts with lead times up to 2 h are of main interest. Currently, human forecasters generate the lvp predictions at VIE using observations at the airport and information from numerical weather prediction (NWP) models. Providing probabilistic predictions of lvp states, however, would support the forecasters producing their predictions.

¹ Department of Atmospheric and Cryospheric Science, University of Innsbruck, Innsbruck, Austria. E-mail: sebastian.j.dietz@gmail.com

² Department of Statistics, University of Innsbruck, Innsbruck, Austria.

Two different types of approaches are typically employed for automatic predictions of meteorological variables. The first one is based on numerical modeling and uses all relevant physical equations to compute forecasts. Many physically-based models for visibility and fog forecasts were developed in the past, e.g. HRRR (Benjamin et al. 2016), The London Model (Boutle et al. 2016), and PAFOG (Bott and Trautmann 2002). Generally, these models are computationally expensive and special end-user-related variables such as lvp have to be derived afterwards from their output.

The second approach uses historical data to train a statistical model and produces probabilistic forecasts of variables such as lvp directly. This approach is generally computationally cheaper so that new predictions are available instantaneously with new input information. In the past, various statistical methods were tested to generate visibility forecasts. Vislocky and Fritsch (1997), for example, used multiple linear regression to generate visibility forecasts from observations. Their forecasting system was later improved by Leyton and Fritsch (2003, 2004) through increasing the density and frequency of the surface observations. An operational visibility forecasting system for several lead times and locations was developed by Ghirardelli and Glahn (2010), using again multiple linear regression. Glahn et al. (2017) combined this system with the physically-based forecasts of Benjamin et al. (2016) to improve the performance. Other statistical techniques to forecast visibility are, for example, neural networks (e.g., Pasini et al. 2001; Marzban et al. 2007), Bayesian model averaging (e.g., Roquelaure et al. 2009), or decision trees (e.g., Bartoková et al. 2015; Dutta and Chaudhuri 2015). Herman and Schumacher (2016) compared various statistical methods for visibility predictions at airports and found that no specific model performs best overall.

The first investigation of lvp, which is the relevant variable for airport operations, was conducted in a companion paper by Kneringer et al. (2018). They used ordered logistic regression (OLR) models to produce lvp nowcasts at VIE for the cold season (September–March) with lead times up to 2 h and 30-min resolution. To provide decision makers with more detailed information, we increase the temporal

resolution of the predictions to 10 min and extend the predictor variable setup of Kneringer et al. (2018) with temporally lagged and spatially averaged visibility information. During the cold season, the peak of lvp is in the early morning hours, coinciding with the airport's rush hours (Kneringer et al. 2018). Since low visibility has most impact during these times, we generate lvp predictions in this paper only from September to March during 6 and 9 UTC. Tree-based statistical methods are used as a flexible nonparametric alternative to the parametric OLR for generating the forecasts. The predictions of these models are compared amongst each other and to the forecasts of the OLR models of Kneringer et al. (2018). Furthermore, the impact of the predictors on the forecasts is analyzed to provide information of the most important inputs for statistically based lvp nowcasts.

2. Data

In this section, the predictand and the predictors used for lvp nowcasts are described. The first part is about the determination of lvp, which has to be derived from horizontal and vertical visibility measurements. In the second part, the predictor variables are described. These variables contain observations at VIE and its vicinity from September 2012 to March 2017 between 6 and 9 UTC.

2.1. Low-Visibility Procedure (lvp) States

The lvp state is an ordered categorical variable that specifies the application of safety procedures at airports due to low visibility. Generally, lvp states are determined by nonlinear, threshold-bounded combinations of horizontal and vertical visibility. Naturally, the exact thresholds and the associated capacity reductions vary for each airport. At VIE, three lvp states are specified (Table 1). The horizontal visibility used for lvp determination is runway visual range (rvr), which is defined as the distance over which the pilot of an aircraft on the centerline of the runway can see the runway surface markings or the lights delineating the runway or identifying its center line (International Civil Aviation Organization 2005).

Table 1

Definition of lvp states with their thresholds in runway visual range (rvr) and ceiling (cei), their occurrence probabilities for the forecast period (cold seasons and early morning hours from 2012 to 2017), and the resulting capacities at VIE

lvp state	rvr (m)		cei (m)	Occurrence (%)	Capacity (%)
0				90.4	100
1	< 1200	or	< 90	2.5	75
2	< 600	or	< 60	5.8	60
3	< 350			1.3	40

The vertical visibility threshold is determined by ceiling (cei), the altitude of the cloud base with at least five octa coverage.

Each touchdown point at VIE is equipped to quantify rvr and cei. While rvr is measured directly by transmissometers, cei has to be derived from ceilometer measurements. Typically, this procedure is conducted by human forecasters, who determine an airport-averaged cei every 30 min by comparing the backscatter profiles of several ceilometers at the airport. To obtain an lvp state at a 10-min resolution, however, cei is required also at 10-min resolution. Hence, we developed a fully automatic algorithm, which uses measurements of the altitude of lowest cloud height and vertical visibility from the ceilometer during the previous 10 min to compute the total cloud coverage at each touchdown point. The most recent measurements are weighted more strongly. Ceiling is computed only if the coverage exceeds five octa. The output of this algorithm closely follows the altitudes identified by human forecasters in the meteorological reports of Vienna Airport. The predictand lvp is computed afterwards with the 10-min values of cei and rvr at each touchdown point. In this study, we focus on forecasts at the touchdown point with the highest climatological lvp state occurrence.

2.2. Predictor Variables

All meteorological variables used as predictors are available in a 10-min output frequency. Moreover, most of them are measured at multiple locations within the vicinity of VIE. For example, wind speed is observed at nine different locations within an area of about 8 km². These measurements are highly correlated and thus the forecast performance of the statistical models would not increase when using all

wind speed measurements. To this end, we conduct a manual preselection of the observations at VIE to generate a highly informative predictor setup. This predictor setup consists of single-point measurements, derived observations, and climatological information. Some variables occur several times through spatial averaging and temporal lagging (Table 2, large setup).

The point measurements included in this predictor setup are rvr, horizontal visibility (vis), air temperature (tl), relative humidity (rh), wind direction (dir), and wind speed (ff). Postprocessed information from the observations are lvp, cei, dew point depression (dpd), temperature difference between 5 cm above the surface and 2 m (dts), wind speed difference between the height of 100 m and 2 m (ffd), and pressure difference from the forecast initial time to 3 h before (p_{diff}). All locations of the point measurements are close to the touchdown point with exception of vis, which is observed at a distance of about 7 km north-east of the touchdown point. This location is well-known to the forecasters since radiation fog often forms there first.

Originally, rvr is censored at 2000 m because visibility above this range is not relevant for the landing approach. For possible fog advection, however, visibility information from ranges above 2000 m is required. Hence, we replace all censored rvr values with visibility information measured next to the transmissometers.

Since lvp is a combination of rvr and cei, these variables may have the strongest influence on lvp nowcasts. We, therefore, generate spatially averaged and temporally lagged predictors from these variables. Spatially averaged observations are averages at the runway and from the complete airport. Runway averages only contain information directly from the

Table 2

Predictor variables used in the statistical models.

Variable	Unit	Description	Variable	Unit	Description
lvp	(0, 1, 2, 3)	Low-visibility procedure	dir	(°)	Wind direction
rvr	(m)	Runway visual range	tl	(°C)	Air temperature
cei	(m)	Ceiling	ff	(ms ⁻¹)	Wind speed
vis	(m)	Visibility	ffd	(ms ⁻¹)	Wind speed difference _{100m-2m}
dts	(°C)	Temperature difference _{2m-5cm}	p_{diff}	(hPa)	Pressure difference _{initial-3 h}
dpd	(°C)	Dew point depression	cc	(0-1)	Conditional climatology
rh	(0-100)	Relative humidity			
dir _N	(no,yes)	Wind direction from north _{binary}			
dir _{SE}	(no,yes)	Wind direction from south east _{binary}			
rr	(no,yes)	Rain in the last 12 h _{binary}			
sza	(°)	Solar zenith angle			

The variables on the left are used in the “standard setup” while the “large setup” additionally considers the variables on the right. Variables available in the standard setup contain information at forecast initialization and at the touchdown point. In the large setup lvp, cei, and rvr information is available also 10-min lags to forecast initialization and averaged for runway and airport, respectively. The variables dir_N, dir_{SE}, rr, and sza used in the standard setup are not included in the large setup (for more details to the standard predictor setup, see Kneringer et al. 2018)

runway. In case of rvr, three sensors are used for averaging (located at both touchdown points and the midpoint); cei averages at the runway contain information of two ceilometers (located at both ends of the runway). Airport averages, however, contain all observations of one variable available in the airport area and, therefore, cover an area of about 8 km². Averaged lvp information of the runway and airport are computed using runway and airport averaged values of rvr and cei, respectively. The temporally lagged information used is lvp, rvr, and cei observations at the touchdown point from 10 min before forecast initialization.

Additionally, we include a conditional 31-day lvp0 state climatology (cc) for each day of the year, containing the lvp0 proportion of the actual day plus/minus 15 days around this day from September 2012 to March 2017 between 6 and 9 UTC.

3. Methods

For the generation of statistical lvp state forecasts, ordered response models are required, which can be easily accommodated in decision trees. Such decision trees have a flowchart-type structure and are, therefore, easy to understand and interpret. Typically, the forecast performance of decision trees can be

improved by aggregating an ensemble of trees using methods such as bagging, random forest, or boosting (James et al. 2014). While such an ensemble usually improves predictive performance, the interpretability is not straightforward anymore due to the more complex structure. However, computing variable importance measures still allows to determine the predictors with the highest benefit for the forecast. In the following, an overview is provided of the tree-based forecasting methods along with details for validation and interpretation.

3.1. Decision Tree

Decision trees are composed of a recursive partitioning algorithm, which splits the training sample into different cells, depending on the association between the forecast variable and its predictors. After the splitting procedure, a constant model is usually computed for each terminal cell.

Classic decision trees, such as CART (Breiman et al. 1984) and C4.5 (Quinlan 1993) select their split variables by maximizing an information measure of node impurity for all possible splits. Such variable selection criteria, however, prefer split variables with many possible splits or missing values and tend to overfit the data (Hothorn et al. 2006b). The decision trees developed in this study are based on the unified

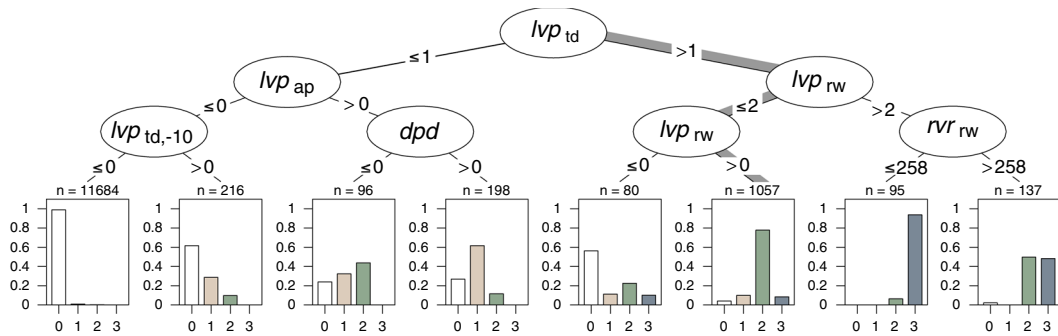


Figure 1

Illustration of the algorithm-selected decision tree for half-hourly lvp state forecasts at Vienna International Airport. The subscripts refer to: at touchdown point (td), averaged over runway (rw) or airport (ap), 10 min prior (-10); dpd is dew point depression, n is the number of observations in the terminal cell encountered in training that tree. Highlighted gray path will lead to the lvp prediction discussed in the text

conditional inference framework of Hothorn et al. (2006b). These trees separate the variable selection and splitting procedure into two steps and do not suffer from a systematic tendency towards split variables with many possible splits or many missing values (Hothorn et al. 2006b).

In the first step of the tree growing process, the association between the response and each of the covariates is computed by deriving the p value for each association. In case of ordinal variables such as lvp, the p values are derived by an χ^2 -test (Hothorn et al. 2006a). If the minimum p value is below a prespecified nominal level α , the covariate with lowest p value is selected as split variable. Otherwise no split is computed. Hence, α can be seen as statistical stopping criterion to avoid overfitting.

For the chosen split variable, the optimal split point is computed by applying a second test statistic over all subsets which can result from possible splits. The split point is set where the discrepancy between two subsets is maximized. In case of lvp forecasts, the χ^2 -test is used again as the test statistic (Hothorn et al. 2006a). Both steps, the variable selection and the split point determination, are repeated recursively until a stopping criterion is reached (e.g. all p values exceed α or a maximum growth depth is reached). The conditional distribution of the response in the particular terminal cells represents the probabilistic forecast of the tree.

An illustration how the described algorithm works for a 30-min lvp forecast at VIE is shown in Fig. 1. Moving through the split points until a terminal cell is

reached, a probabilistic forecast can be generated (histograms in Fig. 1). Note that the optimal selection of split variables and their respective thresholds is done by the algorithm.

In the example marked by the gray path in Fig. 1, the observed lvp states are lvp2 at touchdown point (first split) and lvp2 at the runway (second split). The resulting forecast is 4% lvp0, 10% lvp1, 78% lvp2 and 8% lvp3. This prediction is the most frequent one for prevailing lvp. In the tree building process 1057 observations drop into this terminal cell.

3.2. Tree-Based Ensemble Approaches

A well-known weakness of single decision trees is their high variance, i.e., the tree structure may change considerably when learning it on randomly perturbed (sub)samples from the same data set. Growing an ensemble of decision trees and aggregating them into one model typically reduces the variance and improves the forecast accuracy (James et al. 2014). Hence, the ensemble methods bagging, random forest, and boosting are employed subsequently.

Bagging

In bagging, multiple trees are grown and their predictions are merged to reduce the variance over single decision trees (Strobl et al. 2009; Bühlmann and Yu 2002). In the first part of bagging, multiple training samples of the size of the original training sample are generated by drawing random observations from the original training sample with replacement (bootstrapping). Afterwards, an

individual tree is fitted to each new sample and the predictions of the individual trees are merged. We use the aggregation pattern developed by Hothorn et al. (2004), which collects all observations contained in the obtained terminal cells of the particular trees. The forecast of bagging is then just the distribution of the collected observations.

Random Forest

While bagging considers all predictor variables for splitting at any stage of any tree, random forest only considers a different randomly drawn subset of predictors at each stage. Thus, bagging is a special case of random forests (Strobl et al. 2009) but the latter leads to trees that are less correlated with each other.

The principle of including additional randomness into the single trees can often improve the prediction accuracy of an ensemble. Commonly, one predictor has potentially stronger power than the other covariates. Most of the bagged trees will, therefore, select this predictor as first split variable so that the predictions of the individual trees will have a strong correlation, and consequently only a small variance reduction over a single tree when aggregating these trees (James et al. 2014). Trees grown in a random forest, on the contrary, are quite diverse amongst each other. Through the random preselection of possible split variables in random forests, weaker predictors have the chance to be selected first and may reveal interaction effects with other variables that would have been missed otherwise (Strobl et al. 2009).

Boosting

The third method that is used to combine an ensemble of decision trees is boosting. With boosting, new decision trees always grow on forecast information of previously grown trees, since the new tree is fitted to residuals of the previous ones. In this study, the residual information is computed with the component-wise gradient boosting algorithm of Bühlmann and Hothorn (2007), using the negative gradient vector of the loss-function from the current model. For ordinal response variables such as lvp, the loss-function is defined by the log-likelihood of the proportional odds model of Agresti (2003; Schmid et al. 2011). After the fitting of the new tree, its paths are aggregated to the paths of the previously grown

ones, albeit with a shrinkage parameter to grow the model slowly, which improves the fit relative to single trees (James et al. 2014). Computing new residuals, fitting a tree to them, and combining the new tree with the previous ones continues until a stopping criterion is reached (e.g., maximum number of iterations; see Sect. 3.3).

The trees used in the boosting algorithm are the conditional inference trees of Hothorn et al. (2006b, see Sect.3.1). Since boosting is able to project additive data structures, and decision trees can model nonlinear data features, boosting trees can project both, additive and nonlinear data features. The branch depth of the individual trees can be used to control whether boosting rather captures additive or nonlinear structures.

3.3. Computational Implementation

Decision Tree

The conditional inference trees used are implemented in the R package **party** (Hothorn et al. 2017b). Each tree developed in this investigation can grow until the p value between the response and its most associated predictor exceed the default α value of 0.05.

Bagging and Random Forest

Bagging and random forest models used in this investigation are also implemented in the R package **party** (Hothorn et al. 2017b). Both models contain 500 single decision trees, which turned out to be a reasonable number to ensure short computation times with good forecast performance. The settings of the individual trees in bagging and random forest are default. In random forests, the number of randomly preselected split variables is set to five, which is approximately the square root of the number of predictors (typically recommended in the literature; see James et al. 2014)

Boosting

The boosting method used in this study is implemented in the R package **mboost** (Hothorn et al. 2017a). Each boosting model consists of 1500 trees, which is a reasonable number to guarantee high forecast performance with low computational cost. The shrinkage parameter for tree aggregation is set to the default value 0.1. Each tree in the boosting can

grow to a maximum branch depth of three because we rather combine several small trees instead of few large ones. Using this adjustment, we are able to model both, additive and nonlinear data structures.

3.4. Reference Model

To analyze the benefit of the different tree-based models, we compare their forecast performance to an ordered logistic regression model (OLR), which was shown to outperform climatology, persistence, and predictions of human forecasters at VIE (Kneringer et al. 2018).

For consistency in model comparison, we change the half-hourly averages of *cei* and *lvp* used by Kneringer et al. (2018) to the 10-min resolved values described in Sect. 2.1. Furthermore, the size of the training sample is reduced to five cold seasons instead of the original nine cold seasons.

3.5. Forecast Verification

The forecast performance of the models is analyzed using the ranked probability score (RPS; Epstein 1969; Murphy 1971) and the Brier score (BS; Brier 1950). Both scores are well-known validation metrics for probabilistic forecasts. The RPS is used to assess the overall model performance of ordered response variables such as *lvp* states (Wilks 2011). It is computed by the squared errors of the cumulative distribution function of the forecast probabilities with respect to the observations. For a single forecast-observation pair i the RPS is defined as

$$\text{RPS}_i = \frac{1}{J-1} \sum_{s=1}^J \left[\sum_{j=1}^s y_{ij} - o_{ij} \right]^2,$$

with the forecast probabilities y_{ij} and observations o_{ij} for each category $j = 1, \dots, J$. A perfect forecast results in an RPS of 0; the worst possible forecast has an RPS of 1.

The BS, on the other hand, is used for binary responses. In case of *lvp* forecasts, this score can be used to assess the model performance for one particular *lvp* state. For a single forecast-observation pair i , it is simply defined as

$$\text{BS}_i = (y_i - o_i)^2.$$

For model comparison, we average the scores of all forecast-observation pairs produced from one model. Moreover, the forecast performance of one model relative to a reference model is compared with the respective skill score:

Ranked probability skill score (RPSS):

$$\text{RPSS} = 1 - \frac{\text{RPS}}{\text{RPS}_{\text{reference}}},$$

Brier skill score (BSS):

$$\text{BSS} = 1 - \frac{\text{BS}}{\text{BS}_{\text{reference}}}.$$

All scores are computed out-of-sample by applying a season-wise fivefold cross-validation for each model, where the dataset is subdivided into five samples, each of which contains one cold season. Afterwards, the models are trained on four samples and validated on the remaining one. This training and validating procedure is repeated five times, always with another sample for validation.

For the computation of the model uncertainty, we additionally bootstrap the scores calculated in the cross-validation algorithm above (bootstrapping is described in Sect. 3.2—bagging). Mean scores are computed for 2000 bootstrapped samples. The distribution of the mean scores describes the model uncertainty.

3.6. Variable Importance Measurement

To identify the variables with the highest impact on the forecasts, we apply variable permutation tests in which the true information of a particular predictor is replaced by randomly drawn information from the predictor's true distribution. The information of a predictor is thus intermingled randomly. Stronger decrease in forecast performance shows a higher impact of the permuted variable.

To conduct the variable permutation test we again use fivefold cross-validation. Therefore, we always fit the models to four cold seasons and test them on the remaining one. After generating predictions on the original test sample, we randomly permute one predictor variable and produce with the same model forecasts on the modified sample. This procedure is

repeated for each predictor variable in the test sample. The performance of the predictions from each modified sample is computed by the RPS (Sect. 3.5). We have to mention that permuting predictors with a strong association amongst others can lead to discrepancies in the model and, therefore, to strong effects in the forecast performance (for example, lv_{id} to cei_{id} and rvr_{id}). Nevertheless, the variables with the highest impact on the forecast performance can be effectively identified using this analysis.

4. Results

4.1. Model Comparison

The comparison of tree-based models and the OLR model (olr) of Kneringer et al. (2018) for lvp nowcasts is shown in Fig. 2. All models established for this comparison are based on the standard predictor setup described in Table 2 and tested for lead time + 30, + 60, + 90, and + 120 min.

OLR outperforms decision trees (tree) at each lead time. The benefit varies between 2.5% and 11% and is largest at +120 min. Tree-based ensemble methods (bagging, forest, boosting), however, perform similarly to OLR at most lead times. Boosting is overall best but its advantage over OLR varies on average only between 1% and 4% percent (not statistically significant).

These results show that aggregating multiple trees substantially improves the forecast performance of a

single decision tree. The most efficient aggregation method concerning lvp forecasts is boosting. Random forests perform best at + 120 min lead time when the forecast depends on multiple predictors instead of only few with very high importance. If, however, only few of the available predictors control the forecast like at short lead times, where the forecast relates strongly on persistence, the performance of random forests is worse. In such cases bagging leads to better results.

4.2. Impact of More Predictors

To improve the forecast performance of the models established in Sect. 4.1, we provide them with a large number of predictors (specified in Sect. 2.2). Figure 3 shows the performance of the tree-based models with the large predictor setup relative to the boosting model from Sect. 4.1, which is the best-performing model with the standard predictor setup (reference). In the model validation with the ranked probability score (RPSS, Fig. 3a), single decision trees with the large predictor setup underperform the reference at each lead time. The difference in forecast performance becomes even bigger with longer lead times. On the contrary, the benefit of the tree-based ensemble models random forest, bagging, and boosting compared to the reference increases slightly with longer lead times. Random forest has advantages only for the longest lead times. Bagging outperforms the reference after + 20 min, and boosting after a lead time of + 10 min.

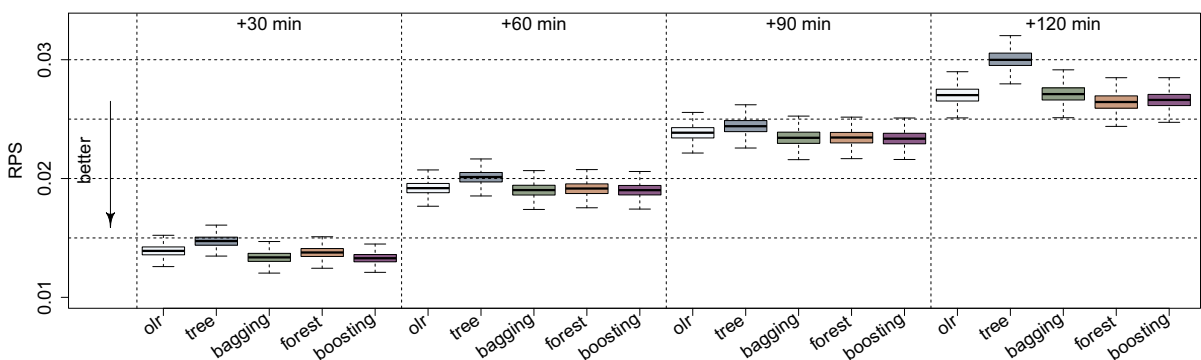


Figure 2

Ranked probability scores (RPS) of the OLR model from Kneringer et al (2018, olr) and the tree-based models decision tree (tree), bagging (bagging), random forest (forest), and boosting (boosting) for lead time + 30, + 60, + 90 and + 120 min. For each model, the standard predictor variable setup described by Kneringer et al. (2018) and in Table 2 is used

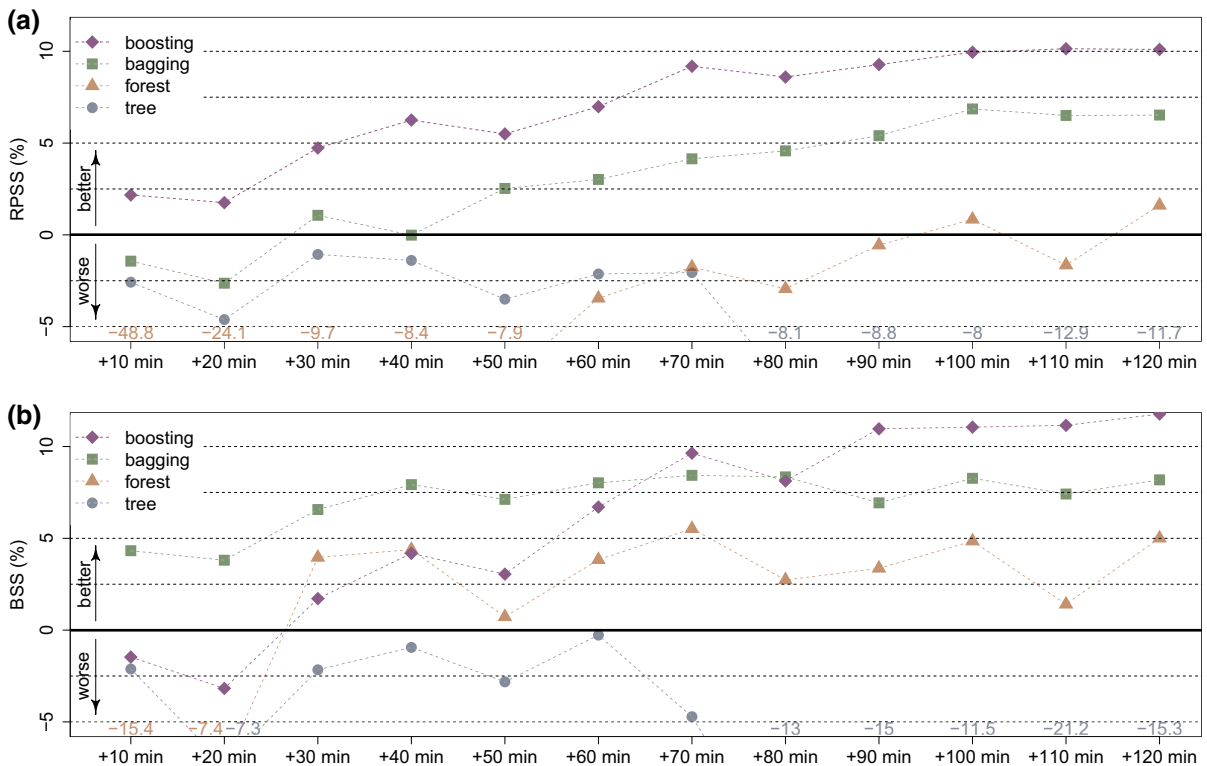


Figure 3

Evaluation of the tree-based models with the large predictor setup defined in Sect. 2.2 relative to the boosting models with the standard predictor set (of Fig. 2). The icons show the median values of the skill scores for the lead times from + 10 min to + 120 min in 10-min resolution. Values outside of the axis limits are written at the bottom of the plot. **a** Ranked probability skill score (RPSS) validation and **b** Brier skill score (BSS) comparison for forecasts of the combined lvp0/1 and lvp2/3 states

At lead time + 120 min, the benefit over the reference is approximately 5% for bagging and 7% for boosting. Again boosting performs best.

At the shortest lead time, boosting with the large predictor setup performs almost equally to the reference. The other models again perform somewhat worse, especially random forest which shows a decrease in forecast performance of about 50% compared to the reference. The reason, therefore, is the reduced set of randomly drawn split variables in the tree building process. Since the models mainly try to reproduce the persistence at the shortest lead times, the single trees strongly require information of current lvp. If this predictor is not included in the randomly selected split variable setup, the single trees perform badly. Aggregating many poorly performing trees results again in worse forecast performance.

To assess model performance for two classes only, we combine forecasts into a no/light capacity

reduction category (lvp0/1) and a severe one (lvp2/3) and verify them using the Brier skill score (BSS). Figure 3b shows the BSS performance of the models with the large predictor setup relative to the reference.

Random forests perform better when verifying on the combined categories. The reduced set of available split variables in the forests leads to a reduced forecast sharpness due to the frequently occurring lvp0 states. The combination lvp0 with lvp1 increases the sharpness of the random forest predictions and, therefore, also the performance due to the reference. For lead times shorter than 1 h, bagging performs much better as measured with the RPSS, and boosting worsens. The performance of decision trees fitted on different training samples varies considerably due to lvp1 since this category is a transition state with a low climatological occurrence. The averaged predictions from multiple similar decision trees, therefore, perform well on combined lvp0/1, but not for separate

lvp1 which is taken into account in the RPS. Boosting with the large predictor setup, on the other hand, predicts lvp1 well, especially at short lead times. This benefit disappears when validating combined lvp0/1 states.

4.3. Case Study

For a better understanding and interpretation of the forecasts and their scores, a case study of a typical fog event is analyzed and shown in Fig. 4. The statistical models used for forecast generation are the boosting trees with the large predictor setup.

The lvp event starts at 04:50 in the early morning of 2013-11-16 and lasts through 10:10 UTC with lvp categories varying between lvp1 and lvp3. Shortly before its start, the air temperature decreases to dew point. Before the event and during its first half, the wind has northwesterly direction with velocities between 4 and 5 ms^{-1} . Afterwards, the wind speed weakens and the direction turns to south and east. With the end of the event, the velocity rises slightly and the direction turns to southwest.

The lvp event starts with a fast decrease of the runway visual range from 2000 m to 650 m within 10 min. It then decreases even further and oscillates between 200 m and 600 m. The event ends with a slow increase of runway visual range and ceiling.

First signs of a possible lvp1 state are already visible in the + 60 min forecast for 03:40 with an increase in probability for 04:50 when lvp actually occurred. Even the + 120 min forecast has an increased probability for lvp1 and lvp2 starting half an hour before the actual onset. Changes between lvp categories are sometimes but not always captured by forecasts of various lead times. For example, the first return from lvp2 to lvp1 at 05:20 is not captured, but the change from lvp3 to lvp2 at 07:00 is.

All forecasts except for the + 120 min show a steady decrease of lvp probabilities before the event ends but still lag after the observed end at 10:10 UTC due to the importance of persistence for the forecasts.

4.4. Impact of Predictor Variables

The analysis of the variables with highest impact on the forecasts is conducted by variable permutation

tests (Sect. 3.6). Therefore, the forecast performance from a test sample with randomly permuted information of a particular predictor is computed and compared to the forecast performance of the original test sample. Figure 5 shows the fractional changes in RPS for predictions on permuted test samples relative to the original sample for bagging (5a) and boosting (5b). The permuted variables with strongest decrease in forecast performance are plotted for lead time + 10, + 60, and + 120 min.

The highest impact on lvp nowcasts with a lead time of + 10 min has the current lvp state at the touchdown point (lvp_{td}). Permuting this variable decreases the forecast performance of bagging on average more than eight times, and of boosting more than three times. Additional variables with impact on the forecasts are the averaged lvp state of the airport (lvp_{ap}) and runway (lvp_{rw}) for bagging, and cei information at touchdown (cei_{td}), as well as the lvp state at touchdown 10 min before forecast initialization (lvp_{td-10}) for boosting. With growing lead times, the strong dependence on lvp information at the touchdown point decreases. For the lead time + 60 min, lvp_{td} still has the highest impact on the forecasts. However, lvp_{rw} also has a strong influence in bagging at + 60 min lead time. Additional important variables are the visibility outside the airport area (vis_{ext}) and the airport-averaged lvp_{ap} and rvr (rvr_{ap}). Their influence is stronger in boosting models than in bagging. At the lead time + 120 min, lvp_{ap} has the strongest influence in bagging. Standard meteorological information of humidity (rh) and temperature (tl), as well as lvp_{td} , lvp_{rw} , and vis_{ext} have further influence. For boosting lvp_{ap} , rvr_{ap} , and vis_{ext} have the strongest impact. Additional impact has lvp information at touchdown and runway.

These results confirm the predictor selection analysis for the OLR model of Kneringer et al. (2018), which also show strongest contribution from current lvp information at short lead times, but from standard meteorological variables at longer lead times. Altogether, the variables with the highest influence are similar for both models for each lead time. At short lead times point measurements at touchdown have the highest influence, whereas averaged airport information becomes more important with increasing lead time. In bagging, the

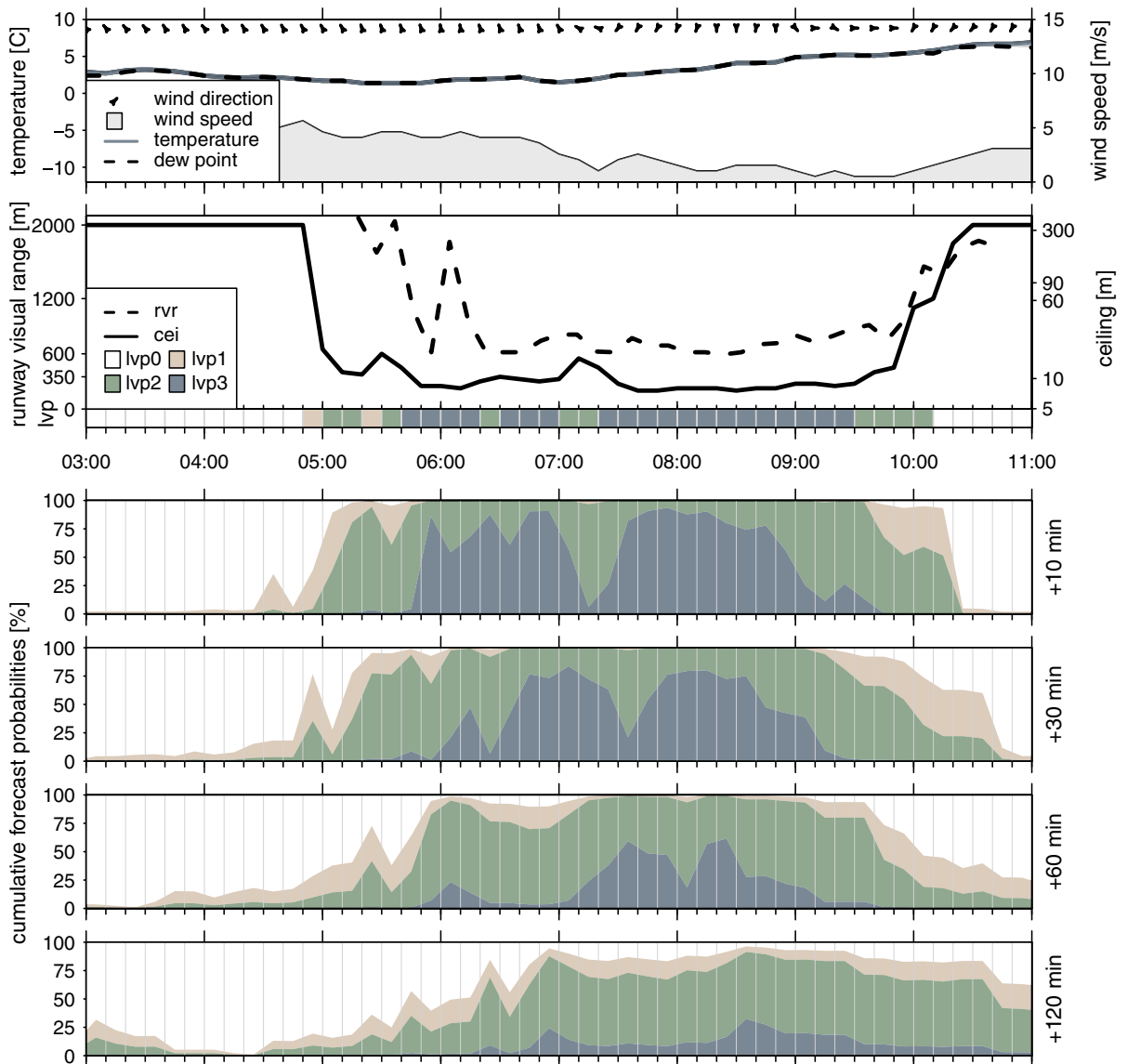


Figure 4

Case study of an lvp event at Vienna International Airport on 2013-11-16. The first panel shows the general meteorological conditions (wind, temperature, dew point) before, during, and after the lvp event. The second panel shows observed runway visual range, ceiling, and lvp (bottom). The panels 3–6 show the cumulative lvp probabilities for forecasts issued 10, 30, 60, and 120 min previously

dependency on lvp information is stronger, instead horizontal visibility information has higher impact in boosting.

5. Discussion and Conclusion

Nowcasts of lvp (low-visibility procedure) states at Vienna International Airport are produced in a high

temporal resolution using different tree-based models. The various models are compared amongst each other and to the OLR (ordered logistic regression) model of Kneringer et al. (2018). Tree-based boosting leads to the best forecasts, outperforming bagging, random forests, and OLR— all three of which perform similarly—while single decision trees have the lowest predictive performance for all lead times.

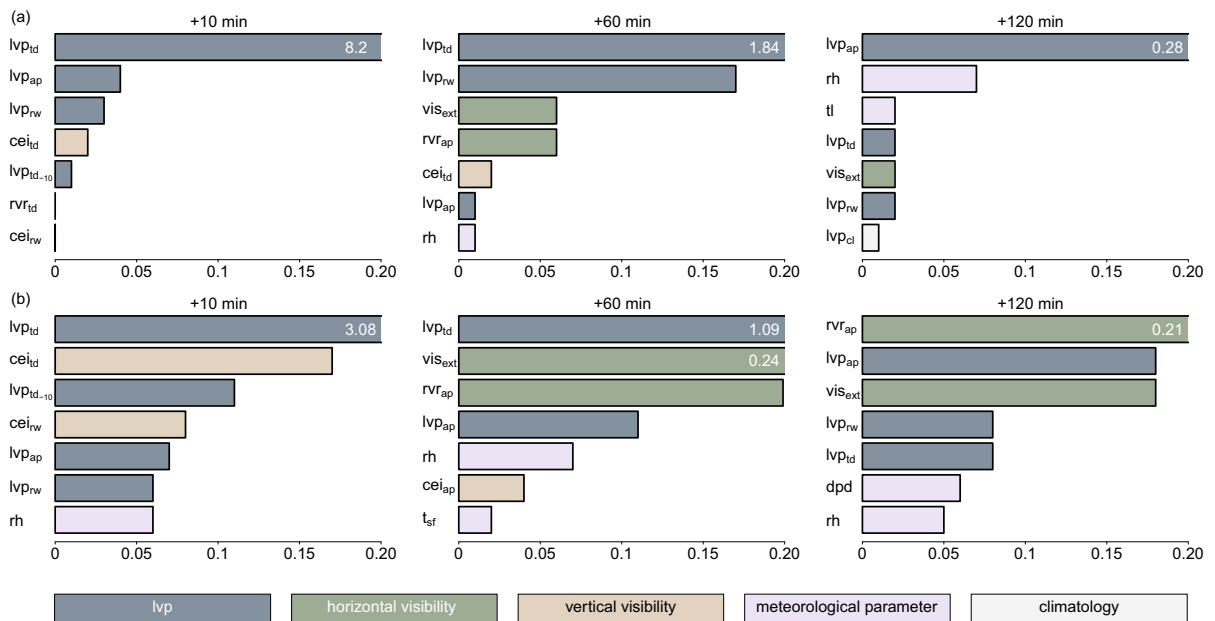


Figure 5

Variable importance analysis for the models bagging (a) and boosting (b). Each bar in the plot shows the fractional decrease in forecast performance for a test sample with a randomly permuted predictor compared to the original test sample. Note that the bars are cut at a maximum fractional decrease of 0.2. Higher values in fractional decrease are written in the bars. For each model, the most important variables are plotted for the lead times + 10, + 60 and + 120 min

To improve the forecast performance of lvp nowcasts, we enlarge the set of predictor variables and increase the output resolution from 30 to 10 min compared to the investigation of Kneringer et al. (2018). Increasing the number of potential predictors improves forecasting performance most for lead times longer than 30 min. Boosting trees are best overall followed by bagging. However, bagging has an advantage over boosting for forecasts of the combined most critical lvp_{2/3} states until 1 h lead time. Random forests only do well at longer lead times, while single decision trees are outperformed by all other models.

The reason, therefore, is based on the different working processes of the models. At very short lead times, the predictions strongly depend on information of the current lvp state at the touchdown point. Since this predictor often occurs as split variable in the trees all aggregated decision trees look similar. Hence, the performance of bagging and decision tree is similar at the shortest lead time. If most information for the forecasts is contained in only one predictor variable,

random forest performs worse due to their random preselection of potential split variables. With longer lead times, the strong dependency to only one predictor variable decreases and, therefore, the benefit of the ensemble-merging models increases strongly compared to single decision trees.

The impact of low visibility on flight activities at airports will increase in future. Although the annual number of low-visibility events over central Europe has declined during the last 40 years (e.g. Egli et al. 2017; Vautard et al. 2009), the number of delays has increased. The reason is the increased number of flights (Federal Aviation Administration 2017), which exhaust the capacity utilization of airports. Since weather contributes most to flight delays (Federal Aviation Administration 2017), the importance of accurate low-visibility forecasts will increase in the future.

In summary, this paper has shown that tree-based methods are suitable tools for lvp state nowcasts. Their computational costs are comparable with linear regression methods and, therefore, they are able to

produce instantaneous forecasts when new observations arrive. Especially the tree-based ensemble methods bagging and boosting perform well for lvp predictions. The predictors with the highest impact to lvp nowcasts contain information of the current lvp state and horizontal visibility at forecast initialization.

Acknowledgements

Open access funding provided by University of Innsbruck and Medical University of Innsbruck. This study is supported by the Austrian Research Promotion agency (FFG), 843457. We thank Markus Kerschbaum, Andreas Lanzinger, and the staff at Vienna International Airport for fruitful information and discussions regarding these investigations. The authors also thank the Austro Control GmbH for providing access to the observation data.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

REFERENCES

- Agresti, A. (2003). *Categorical Data Analysis*. Hoboken: Wiley. <https://doi.org/10.1002/0471249688>.
- Bartoková, I., Bott, A., Bartok, J., & Gera, M. (2015). Fog prediction for road traffic safety in a coastal desert region: Improvement of nowcasting skills by the machine-learning approach. *Boundary-Layer Meteorology*, 157(3), 501–516. <https://doi.org/10.1007/s10546-015-0069-x>.
- Benjamin, S. G., Weygandt, S. S., Brown, J. M., Hu, M., Alexander, C. R., Smirnova, T. G., et al. (2016). A North American hourly assimilation and model forecast cycle: The rapid refresh. *Monthly Weather Review*, 144(4), 1669–1694. <https://doi.org/10.1175/MWR-D-15-0242.1>.
- Bott, A., & Trautmann, T. (2002). PAFOG—a new efficient forecast model of radiation fog and low-level stratiform clouds. *Atmospheric Research*, 64(1–4), 191–203. [https://doi.org/10.1016/S0169-8095\(02\)00091-1](https://doi.org/10.1016/S0169-8095(02)00091-1).
- Boutle, I. A., Finnenkoetter, A., Lock, A. P., & Wells, H. (2016). The London model: Forecasting fog at 333 m resolution. *Quarterly Journal of the Royal Meteorological Society*, 142(694), 360–371. <https://doi.org/10.1002/qj.2656>.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. Wadsworth Statistics Series. Boca Raton: CRC Press.
- Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review* 78(1):1–3. [https://doi.org/10.1175/1520-0493\(1950\)078<0001:VOFEIT>2.0.CO;2](https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2)
- Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, 22(4), 477–505. <https://doi.org/10.1214/07-STS242>.
- Bühlmann, P., & Yu, B. (2002). Analyzing bagging. *The Annals of Statistics*, 30(4), 927–961. <https://doi.org/10.1214/aos/1031689014>.
- Dutta, D., & Chaudhuri, S. (2015). Nowcasting visibility during wintertime fog over the airport of a metropolis of India: Decision tree algorithm and artificial neural network approach. *Natural Hazards*, 75(2), 1349–1368. <https://doi.org/10.1007/s11069-014-1388-9>.
- Egli, S., Thies, B., Dröner, J., Cermak, J., & Bendix, J. (2017). A 10 year fog and low stratus climatology for Europe based on meteosat second generation data. *Quarterly Journal of the Royal Meteorological Society*, 143(702), 530–541. <https://doi.org/10.1002/qj.2941>.
- Epstein, E. S. (1969). A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology*, 8, 985–987. [https://doi.org/10.1175/1520-0450\(1969\)008<0985:ASSFPF>2.0.CO;2](https://doi.org/10.1175/1520-0450(1969)008<0985:ASSFPF>2.0.CO;2).
- Federal Aviation Administration (2017). Air Traffic by the Numbers. Tech. rep., Economic Analysis Group, URL: https://www.faa.gov/air_traffic/by_the_numbers/media/Air_Traffic_by_the_Numbers_2017_Final.pdf. Accessed 18 Apr 2018
- Ghirardelli, J. E., & Glahn, B. (2010). The meteorological development laboratorys aviation weather prediction system. *Weather and Forecasting*, 25(4), 1027–1051. <https://doi.org/10.1175/2010WAF2222312.1>.
- Glahn, B., Schnapp, A. D., Ghirardelli, J. E., & Im, J. S. (2017). A LAMP-HRRR MELD for improved aviation guidance. *Weather and Forecasting*, 32(2), 391–405. <https://doi.org/10.1175/WAF-D-16-0127.1>.
- Herman, G. R., & Schumacher, R. S. (2016). Using reforecasts to improve forecasting of fog and visibility for aviation. *Weather and Forecasting*, 31(2), 467–482. <https://doi.org/10.1175/WAF-D-15-0108.1>.
- Hothorn, T., Lausen, B., Benner, A., & Radespiel-Tröger, M. (2004). Bagging survival trees. *Statistics in Medicine*, 23(1), 77–91. <https://doi.org/10.1002/sim.1593>.
- Hothorn, T., Hornik, K., van de Wiel, M. A., & Zeileis, A. (2006a). A Lego system for conditional inference. *The American Statistician*, 60(3), 257–263. <https://doi.org/10.1198/000313006X118430>.
- Hothorn, T., Hornik, K., & Zeileis, A. (2006b). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3), 651–674. <https://doi.org/10.1198/106186006X133933>.
- Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., Hofner, B. (2017a). mboost: Model-Based Boosting. URL: <https://CRAN.R-project.org/package=mboost>, R package version 2.8-0. Accessed 10 May 2017
- Hothorn, T., Hornik, K., Strobl, C., Zeileis, A. (2017b). party – A Laboratory for Recursive Partytioning. URL: <http://CRAN.R-project.org/package=party>, R package version 1.2-3
- International Civil Aviation Organization (2005). Manual of Runway Visual Range Observing and Reporting Practices. Tech. Rep. Doc 9365 AN/908

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An Introduction to Statistical Learning: With Applications in R*. New York, NY, USA: Springer Texts in Statistics.
- Kneringer, P., Dietz, S., Mayr, G. J., & Zeileis, A. (2018). Probabilistic Nowcasting of Low-Visibility Procedure States at Vienna International Airport During Cold Season. *Pure and Applied Geophysics*, <https://doi.org/10.1007/s00024-018-1863-4>.
- Leyton, S. M., & Fritsch, J. M. (2004). The impact of high-frequency surface weather observations on short-term probabilistic forecasts of ceiling and visibility. *Journal of Applied Meteorology*, *43*, 145–156. [https://doi.org/10.1175/1520-0450\(2004\)043<0145:TIOHSW>2.0.CO;2](https://doi.org/10.1175/1520-0450(2004)043<0145:TIOHSW>2.0.CO;2).
- Leyton, S. M., & Fritsch, M. (2003). Short-term probabilistic forecasts of ceiling and visibility utilizing high-density surface weather observations. *Weather and Forecasting*, *18*, 891–902. [https://doi.org/10.1175/1520-0434\(2003\)018<0891:SPFOCA>2.0.CO;2](https://doi.org/10.1175/1520-0434(2003)018<0891:SPFOCA>2.0.CO;2).
- Marzban, C., Leyton, S., & Colman, B. (2007). Ceiling and visibility forecasts via neural networks. *Weather and Forecasting*, *22*(3), 466–479. <https://doi.org/10.1175/WAF994.1>.
- Murphy, A. H. (1971). A note on the ranked probability score. *Journal of Applied Meteorology*, *10*, 155–156. [https://doi.org/10.1175/1520-0450\(1971\)010<0155:ANOTRP>2.0.CO;2](https://doi.org/10.1175/1520-0450(1971)010<0155:ANOTRP>2.0.CO;2).
- Pasini, A., Pelino, V., & Potesta, S. (2001). A neural network model for visibility nowcasting from surface observations: Results and sensitivity to physical input variables. *Journal of Geophysical Research: Atmospheres*, *106*(D14), 14,951–14,959. <https://doi.org/10.1029/2001JD900134>.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Francisco: Morgan Kaufmann Publishers Incorporated.
- Roquelaure, S., Tardif, R., Remy, S., & Bergot, T. (2009). Skill of a ceiling and visibility local ensemble prediction system (LEPS) according to fog-type prediction at Paris-Charles de Gaulle Airport. *Weather and Forecasting*, *24*(6), 1511–1523. <https://doi.org/10.1175/2009WAF2222213.1>.
- Schmid, M., Hothorn, T., Maloney, K. O., Weller, D. E., & Potapov, S. (2011). Geoadditive regression modeling of stream biological condition. *Environmental and Ecological Statistics*, *18*(4), 709–733. <https://doi.org/10.1007/s10651-010-0158-4>.
- Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods*, *14*(4), 323–348. <https://doi.org/10.1037/a0016973>.
- University of Westminster (2015) European Airline Delay Cost Reference Values. Tech. rep., EUROCONTROL, URL: <http://www.eurocontrol.int/sites/default/files/publication/files/european-airline-delay-cost-reference-values-final-report-4-1.pdf>. Accessed 18 Apr 2018
- Vautard, R., Yiou, P., & van Oldenborgh, G. J. (2009). Decline of fog, mist and haze in Europe over the past 30 years. *Nature Geoscience*, *2*, 115–119. <https://doi.org/10.1038/ngeo414>.
- Vislocky, R. L., & Fritsch, M. J. (1997). An automated, observations-based system for short-term prediction of ceiling and visibility. *Weather and Forecasting*, *12*, 31–43. [https://doi.org/10.1175/1520-0434\(1997\)012<0031:AAOBSF>2.0.CO;2](https://doi.org/10.1175/1520-0434(1997)012<0031:AAOBSF>2.0.CO;2).
- Wilks, D. (2011). *Statistical Methods in the Atmospheric Sciences*. Cambridge: Academic Press.

(Received September 26, 2017, revised May 22, 2018, accepted May 31, 2018, Published online June 15, 2018)