

Vertical differences in liquid water content between warm and cold liquid-phase clouds over the North China Plain and their controlling factors: insights from in situ aircraft observations and ensemble learning models

5 Yulin Wang¹, Yichen Lu¹, Honglei Wang^{1,2*}, Chunsong Lu¹, Sihan Liu³, Yang Yang³, Zirui Liu², Yan Yin¹, Deyu Liu¹, Yue Chen¹, Tianliang Zhao¹

¹Collaborative Innovation Centre on Forecast and Evaluation of Meteorological Disasters (CIC-FEMD), China Meteorological Administration Aerosol-Cloud and Precipitation Key Laboratory, Nanjing University of Information Science and Technology, Nanjing 210044, China

10 ²State Key Laboratory of Atmospheric Environment and Extreme Meteorology, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing, China

³Hebei Provincial Weather Modification Office, Shijiazhuang 050000, China

Correspondence to: Honglei Wang (hongleiwang@nuist.edu.cn)

Text S1: Machine-learning models

15 This study selected 11 machine learning algorithms as base models: Decision Tree (DT), K-nearest neighbors (KNN), Support Vector Machine (SVM), Multilayer Perceptron (MLP), Bagging-DT, Random Forest (RF), ExtraTrees, Categorical Boosting (CatBoost), eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM) and Adaptive Boosting (AdaBoost). These models encompass three categories of methods—Single, Bagging and Boosting—enabling the characterization of the nonlinear relationship between input features and the target variable from various perspectives.

20 DT is a fundamental machine learning method commonly used for classification and regression tasks. It recursively partitions the sample space based on feature thresholds to construct an inverted tree structure comprising a root node, internal nodes, and leaf nodes, ultimately yielding predictions at the leaf nodes (Song and Lu, 2015).

KNN is a non-parametric learning method based on sample similarity. For a given sample, KNN identifies the K closest samples in the training set according to a distance metric and estimates the target value from these neighboring samples (Cover and Hart, 1967).

25 SVM is a supervised learning method based on statistical learning theory. In regression tasks, SVM constructs an optimal regression function that keeps the predicted values as close as possible to the observations within a specified error tolerance. By using kernel functions, SVM can map the original input space into a higher-dimensional feature space, enabling it to represent nonlinear relationships (Hearst et al., 1998).

30 MLP is a typical feed-forward artificial neural network consisting of an input layer, one or more hidden layers, and an output layer. Through nonlinear activation functions in the hidden layers, MLP learns complex mappings between input variables and the target variable, while network weights are updated using the back-propagation algorithm (Rumelhart et al., 1986).

Bagging-DT is an ensemble learning model that combines Bootstrap Aggregating with decision trees. It first constructs multiple sub-training sets from the original training set using sampling with replacement, then trains multiple decision trees individually, and finally averages the prediction results of these trees to obtain the final regression result (Breiman, 1996).

RF is a tree-based ensemble method developed from the Bagging framework. In addition to constructing multiple training subsets through bootstrap sampling, RF randomly selects a subset of features for splitting at each tree node, thereby increasing the diversity among individual trees. The final prediction is obtained by averaging the outputs of all trees (Breiman, 2001; Liaw and Wiener, 2002).

ExtraTrees is an ensemble learning method based on multiple decision trees. Similar to RF, ExtraTrees performs regression by constructing multiple randomized trees and aggregating their predictions. However, ExtraTrees differs by introducing random split thresholds during the node-splitting process, thereby enhancing the diversity of the tree structures within the model (Geurts et al., 2006).

CatBoost is a machine learning algorithm based on gradient boosting decision trees. It uses ordered boosting and ordered target statistics to reduce prediction shift during training and adopts symmetric tree structures to improve model stability and computational efficiency (Dorogush et al., 2018; Prokhorenkova et al., 2018).

XGBoost is an efficient gradient boosting tree model that extends traditional gradient boosting decision trees by incorporating regularization terms, second-order gradient optimization, row sampling, and column sampling. It sequentially builds new regression trees to fit the residuals of previous models, while considering both prediction error and model complexity in the objective function (Chen and Guestrin, 2016; Friedman, 2001).

LightGBM is an efficient gradient boosting decision tree framework. It uses a histogram-based feature discretization method to reduce computational cost and improve training efficiency. In addition, LightGBM adopts a leaf-wise growth strategy, splitting the leaf with the largest gain first, which allows the model to achieve strong fitting ability under comparable computational conditions (Ke et al., 2017).

AdaBoost is a classic ensemble learning method based on boosting. It trains multiple weak learners sequentially; in each iteration, it increases the weights of samples that were poorly predicted in the previous round, thereby directing the subsequent learners to focus more on hard-to-predict samples. The final prediction is obtained by a weighted combination of these weak learners (Drucker, 1997; Freund and Schapire, 1997).

Text S2: Model evaluation metrics

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (1)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

where n is the number of samples, y_i and $\hat{y}_i$ are the observed and predicted values of the i -th sample, respectively, and $\bar{y}$ is the mean of the observations.

65 Text S3: SHAP methodology

SHAP quantifies the contribution of each feature to the model prediction on the basis of the Shapley value from cooperative game theory. The underlying idea is to add a specific feature to a subset of features one by one, observe the resulting change in the model output, and take a weighted average of the marginal contributions of that feature over all possible subsets, which yields its SHAP value φ_i :

$$\varphi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|! (M - |S| - 1)!}{M!} [v(S \cup \{i\}) - v(S)] \quad (4)$$

where F is the set of all features, M is the total number of features, S is any subset that does not contain feature i , and $v(S)$ is the model output obtained using the subset S alone.

The prediction $f(x)$ for a given sample can be expressed as the linear sum of a base value φ_0 and the SHAP values φ_i of the individual features:

$$f(x) = \varphi_0 + \sum_{i=1}^M \varphi_i \quad (5)$$

The interaction between features i and j is described by the SHAP interaction value. A positive value indicates that their combined contribution exceeds the sum of their individual contributions, meaning that the features are mutually reinforcing, whereas a negative value indicates mutual suppression. The method for calculating the interaction value is as follows:

$$Interaction_{ij} = v(S \cup \{i, j\}) - v(S \cup \{i\}) - v(S \cup \{j\}) + v(S) \quad (6)$$

80 Table S1: Optimal hyperparameters of each base model

Category	Model	Optimal hyperparameters
Single	DT	max_depth = 12; min_samples_split = 21; min_samples_leaf = 10; max_features = None
	KNN	n_neighbors = 10; weights = uniform; p = 1
	SVM	C = 1.439; gamma = 0.09738; epsilon = 0.002615
	MLP	n_layers = 4; n_units_l0 = 116; n_units_l1 = 63; n_units_l2 = 107; n_units_l3 = 128; activation = relu; alpha = 0.001296; learning_rate_init = 0.0008455
Bagging	Bagging-DT	base_depth = 14; n_estimators = 42; max_samples = 0.7998; max_features = 0.7609
	RF	n_estimators = 361; max_depth = 14; min_samples_split = 4; min_samples_leaf = 2; max_features = 0.7

Boosting	ExtraTrees	n_estimators = 155; max_depth = 18; min_samples_split = 6; min_samples_leaf = 3; max_features = 0.5
	CatBoost	iterations = 331; depth = 6; learning_rate = 0.09651; l2_leaf_reg = 1.435; subsample = 0.6686
	XGBoost	n_estimators = 190; max_depth = 5; learning_rate = 0.1143; subsample = 0.8526; colsample_bytree = 0.9021; reg_alpha = 0.01899; reg_lambda = 0.01016
	LightGBM	n_estimators = 256; max_depth = 11; num_leaves = 54; learning_rate = 0.15; min_child_samples = 10; subsample = 0.8064; colsample_bytree = 0.8942; reg_alpha = 0.01134; reg_lambda = 0.01063
	AdaBoost	base_depth = 8; n_estimators = 148; learning_rate = 0.854

Note: Floating-point values retain 4 significant digits; max_features = None indicates that there is no limit to the number of features.

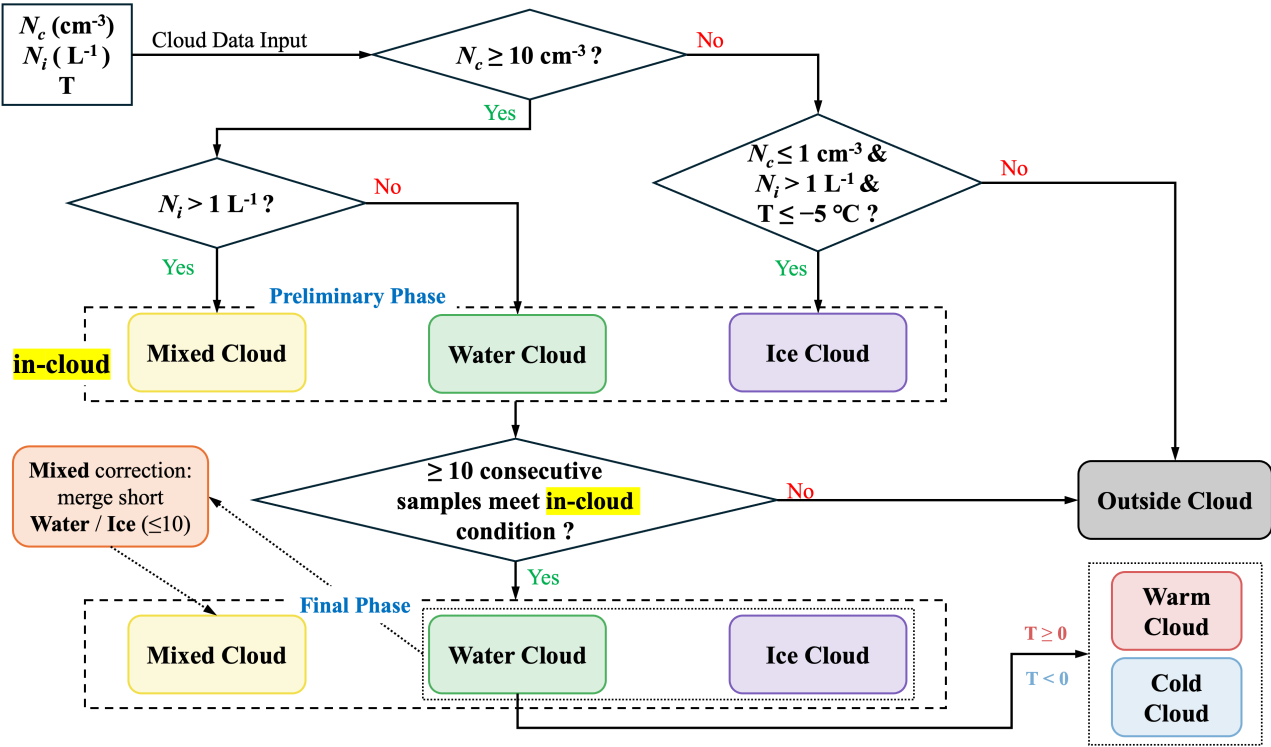


Figure S1: Cloud phase classification scheme.

85

90

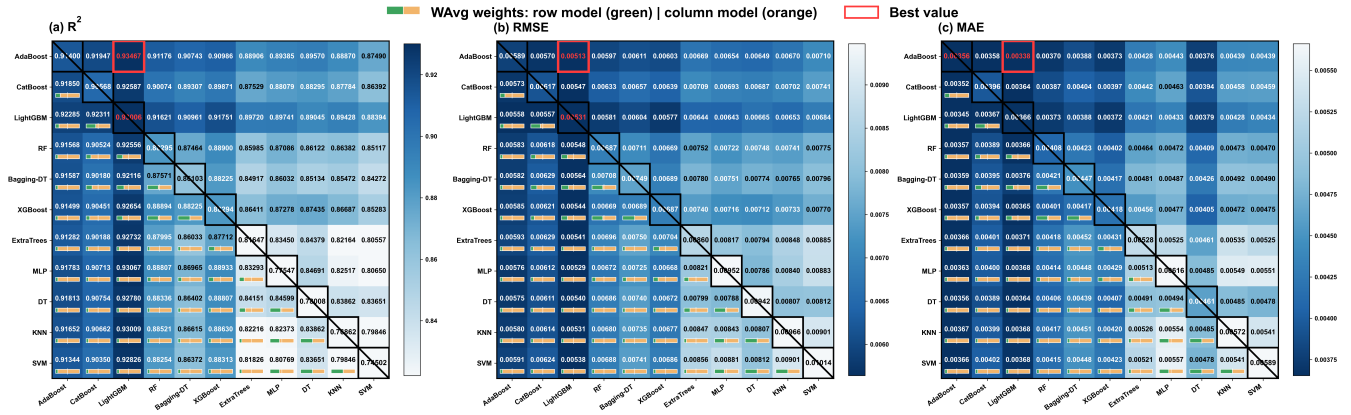


Figure S2: Pairwise ensemble performance matrices of the eleven base models for LWC prediction. Heatmaps of the (a) R^2 , (b) RMSE and (c) MAE of all two-model ensembles, evaluated on the test set. In each matrix, the off-diagonal cells give the metric of the ensemble combining the row and column models; cells above the diagonal use simple averaging (Avg) and cells below use weighted averaging (WAvg), while the diagonal gives the single-model performance. The paired green/orange bars in each WAvg cell indicate the optimal weights assigned to the row (green) and column (orange) model. The red box denotes the best-performing two-model ensemble, and the red value on the diagonal the best-performing single model.

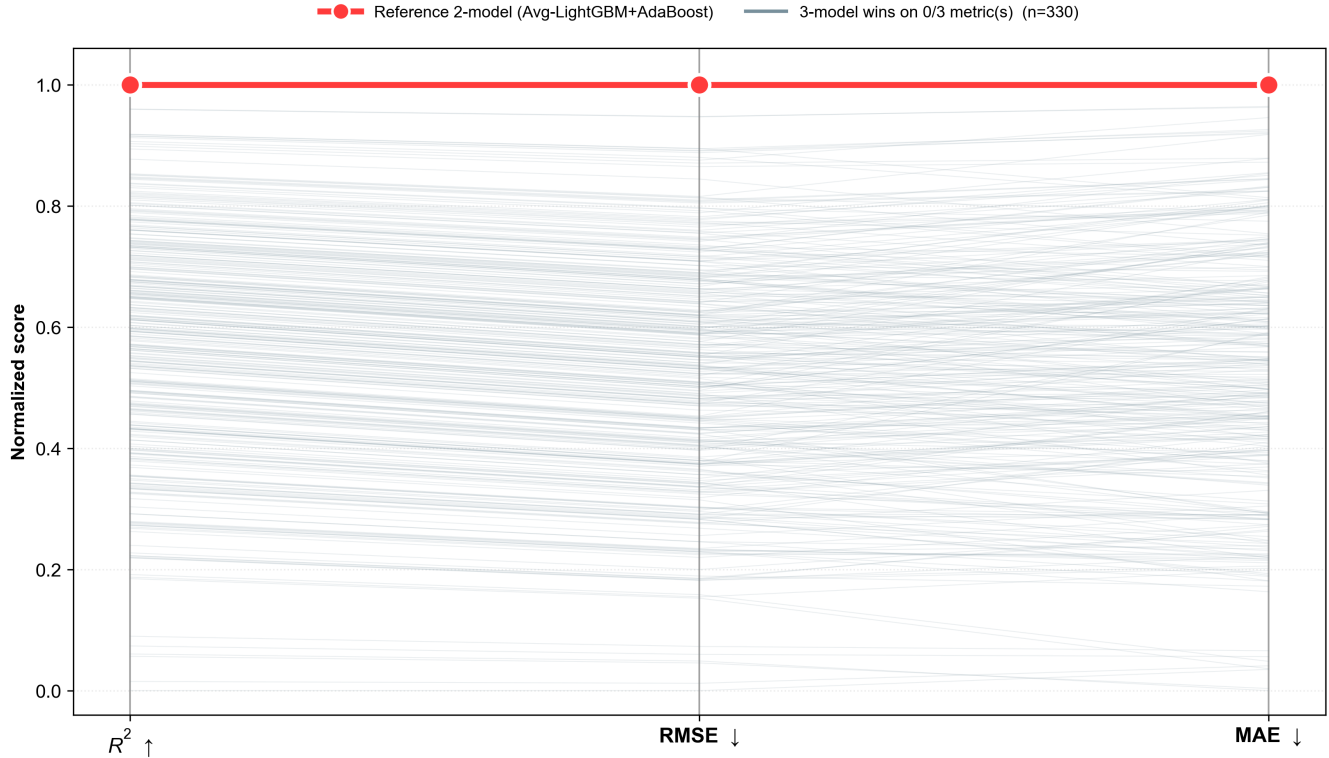


Figure S3: Metrics comparison between the selected two-model ensemble and all candidate three-model ensembles. R^2 , RMSE and MAE (arrows indicate the optimization direction) of the reference two-model ensemble (Avg-AdaLGBM, red) and all 330 candidate three-model ensembles (gray; 165 model combinations $\times$ two averaging schemes). Each metric is normalized so that the reference ensemble equals 1.0.

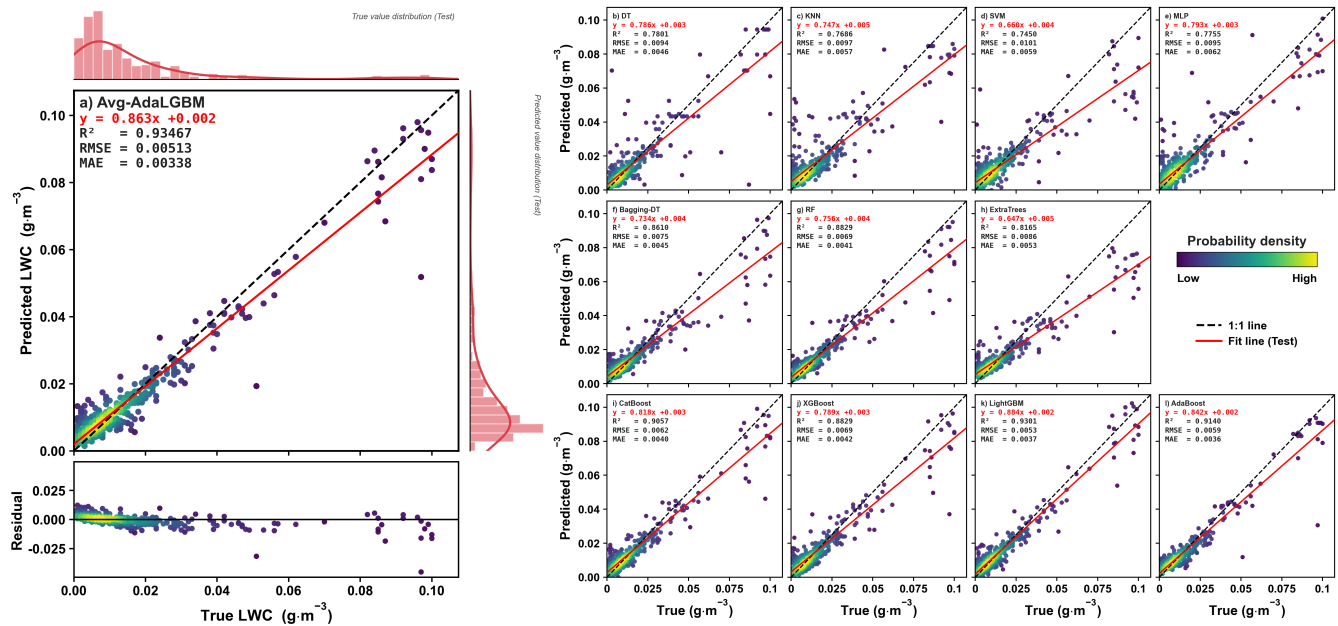


Figure S4: Scatter density plots of predicted and observed LWC for the 11 base models and the selected ensemble model. (a) Avg-AdaLGBM, with marginal histograms showing the true (top) and predicted (right) LWC distributions and the residuals shown below; (b) DT, (c) KNN, (d) SVM, (e) MLP, (f) Bagging-DT, (g) RF, (h) ExtraTrees, (i) CatBoost, (j) XGBoost, (k) LightGBM and (l) AdaBoost. The red line is the fitted line and the black dashed line is the 1:1 line.

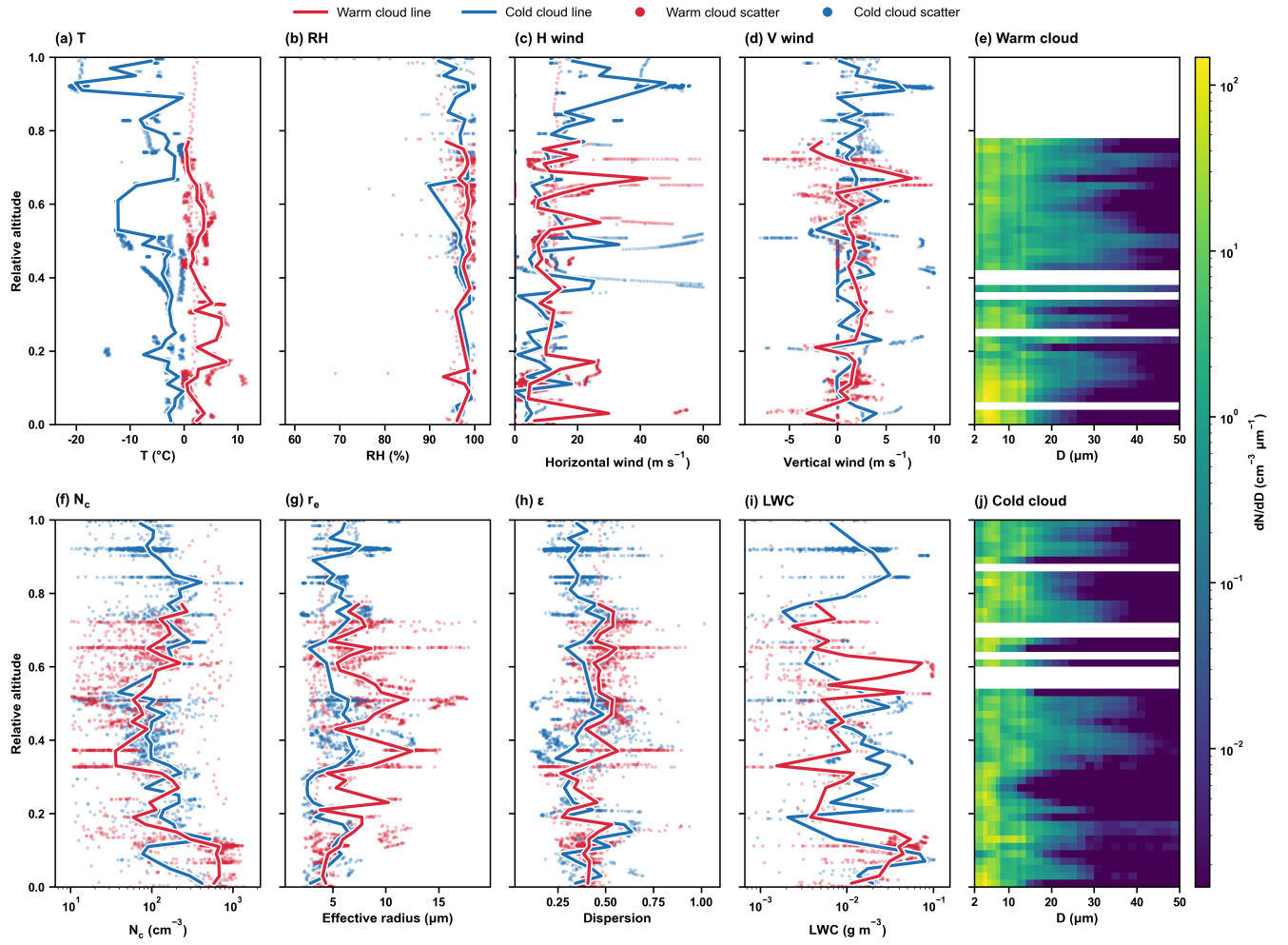


Figure S5: Vertical profiles of different environmental and cloud particle characteristics at relative altitude for warm and cold liquid-phase clouds. (a) T , (b) RH , (c) horizontal wind, (d) vertical wind, (e) warm-cloud droplet size distribution, (f) N_c , (g) r_e , (h) ϵ , (i) LWC and (j) cold-cloud droplet size distribution. Dots are individual samples and solid lines are the binned mean profiles for warm (red) and cold (blue) clouds. Colors in (e) and (j) represent dN/dD .

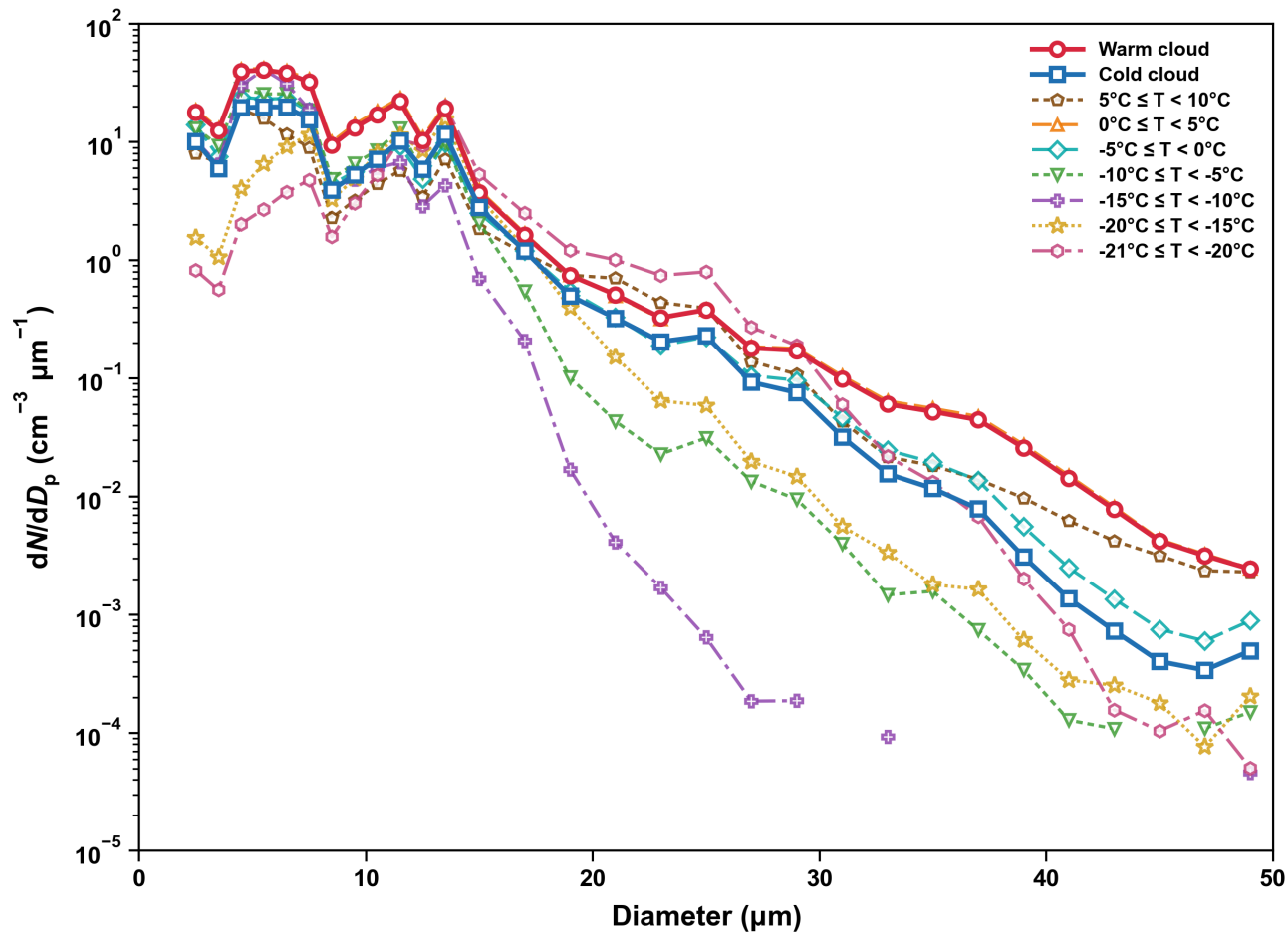
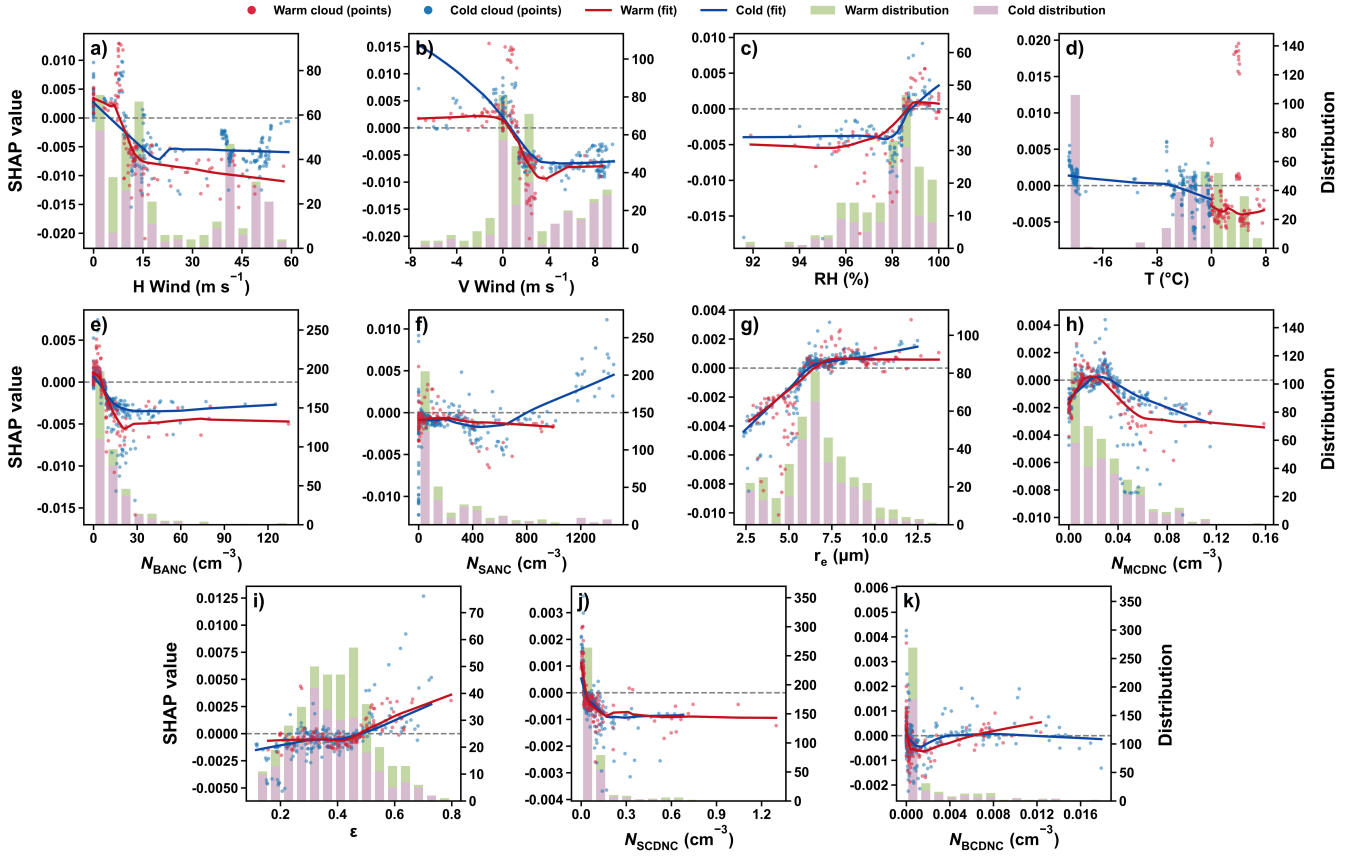
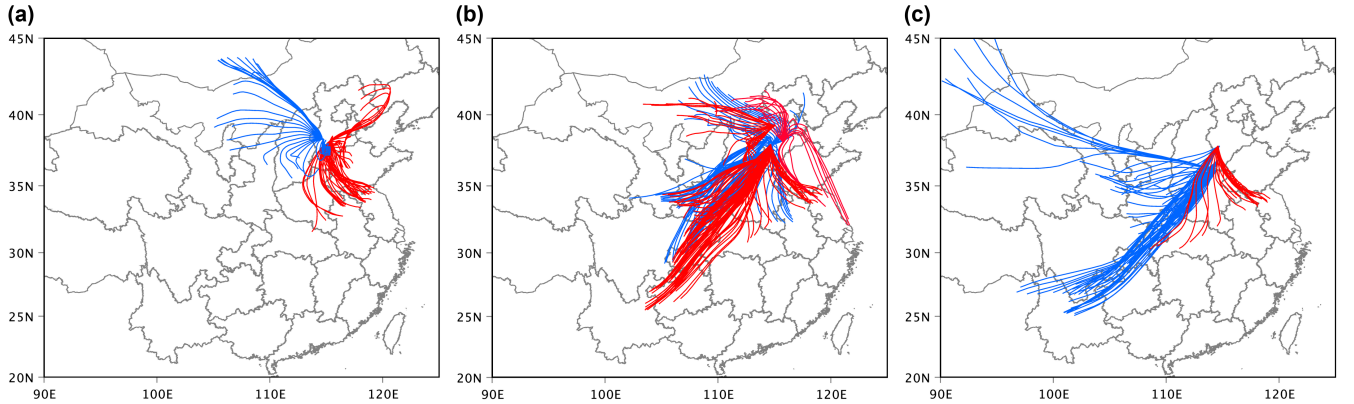


Figure S6: Cloud particle size distributions for warm and cold liquid-phase clouds. Mean number size distributions of cloud particles for all warm-cloud samples (red circles) and all cold-cloud samples (blue squares), together with those for individual temperature intervals: $5^\circ\text{C} \leq T < 10^\circ\text{C}$ (brown pentagons) and $0^\circ\text{C} \leq T < 5^\circ\text{C}$ (orange triangles) for warm clouds; $-5^\circ\text{C} \leq T < 0^\circ\text{C}$ (cyan diamonds), $-10^\circ\text{C} \leq T < -5^\circ\text{C}$ (green inverted triangles), $-15^\circ\text{C} \leq T < -10^\circ\text{C}$ (purple crosses), $-20^\circ\text{C} \leq T < -15^\circ\text{C}$ (gold stars) and $-21^\circ\text{C} \leq T < -20^\circ\text{C}$ (pink hexagons) for cold clouds. All intervals use a uniform 5°C bin width except the coldest one, which is a 1°C residual bin bounded by the lowest observed temperature. Samples with $T \geq 10^\circ\text{C}$ were also observed but are too few to yield a representative mean spectrum and are therefore not shown as a separate interval.



115

Figure S7: SHAP dependence plots in warm and cold liquid-phase clouds. (a) Horizontal wind, (b) vertical wind, (c) RH, (d) T , (e) N_{BANC} , (f) N_{SANC} , (g) r_e , (h) N_{MCDNC} , (i) ε , (j) N_{SCDNC} and (k) N_{BCDNC} . The dots are individual SHAP values and the solid lines are the corresponding fits (left y-axis; red = warm, blue = cold), and the bars give the feature distribution (right y-axis; green = warm, violet = cold).



120

Figure S8: 24h air-mass back trajectories for warm (red) and cold (blue) liquid-phase clouds. Back trajectories initialized at the (a) cloud base, (b) cloud middle, (c) and cloud top.

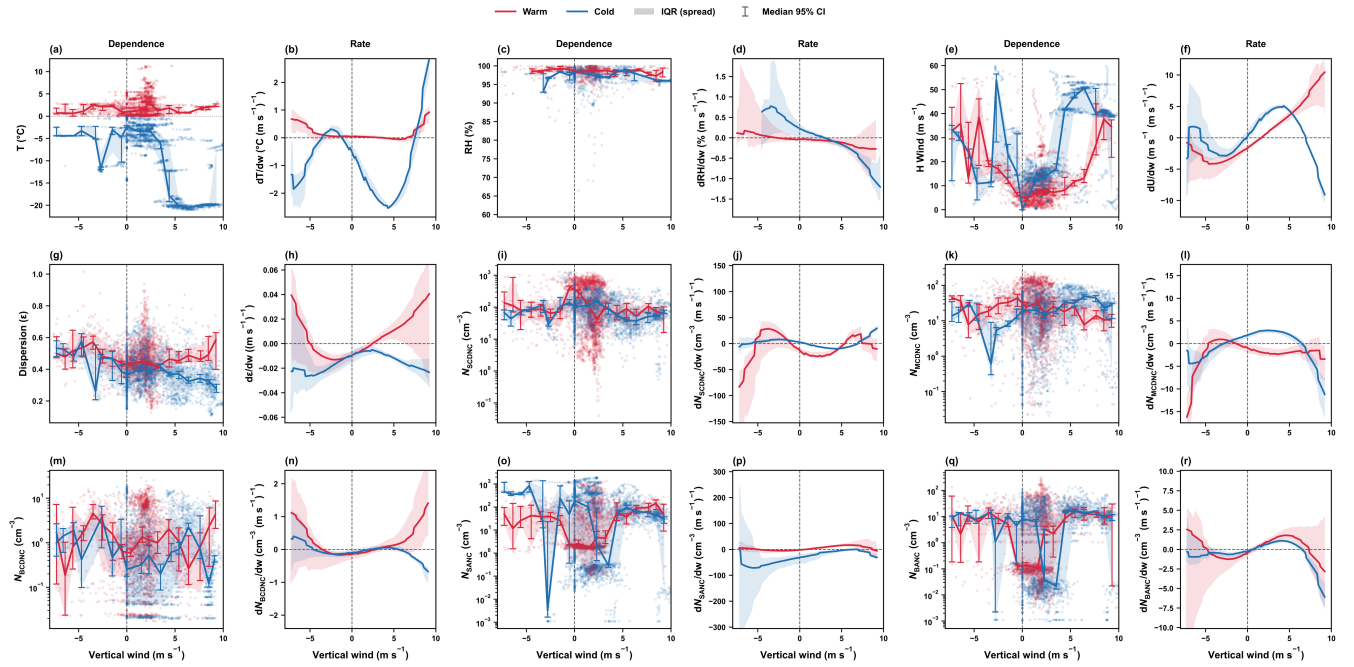


Figure S9: Vertical wind dependence of environmental and cloud microphysical properties in warm and cold liquid-phase clouds. Dependence on vertical wind and the corresponding rates of change, dY/dw , are shown for (a,b) T , (c,d) RH , (e,f) H Wind, (g,h) ε , (i,j) N_{SCDNC} , (k,l) N_{MCDNC} , (m,n) N_{BCDNC} , (o,p) N_{SANC} , and (q,r) N_{BANC} . Negative and positive w denote updrafts and downdrafts, respectively, and the vertical dashed line marks $w = 0$. In the dependence panels, dots show individual samples, lines show binned medians, shading shows interquartile ranges, and error bars show bootstrap 95% confidence intervals. Rate panels show local derivatives from sample-size-weighted local quadratic fits to the binned medians, with shading indicating bootstrap 95% confidence intervals. Rates are computed from untransformed values, so logarithmic concentration axes affect the display only.

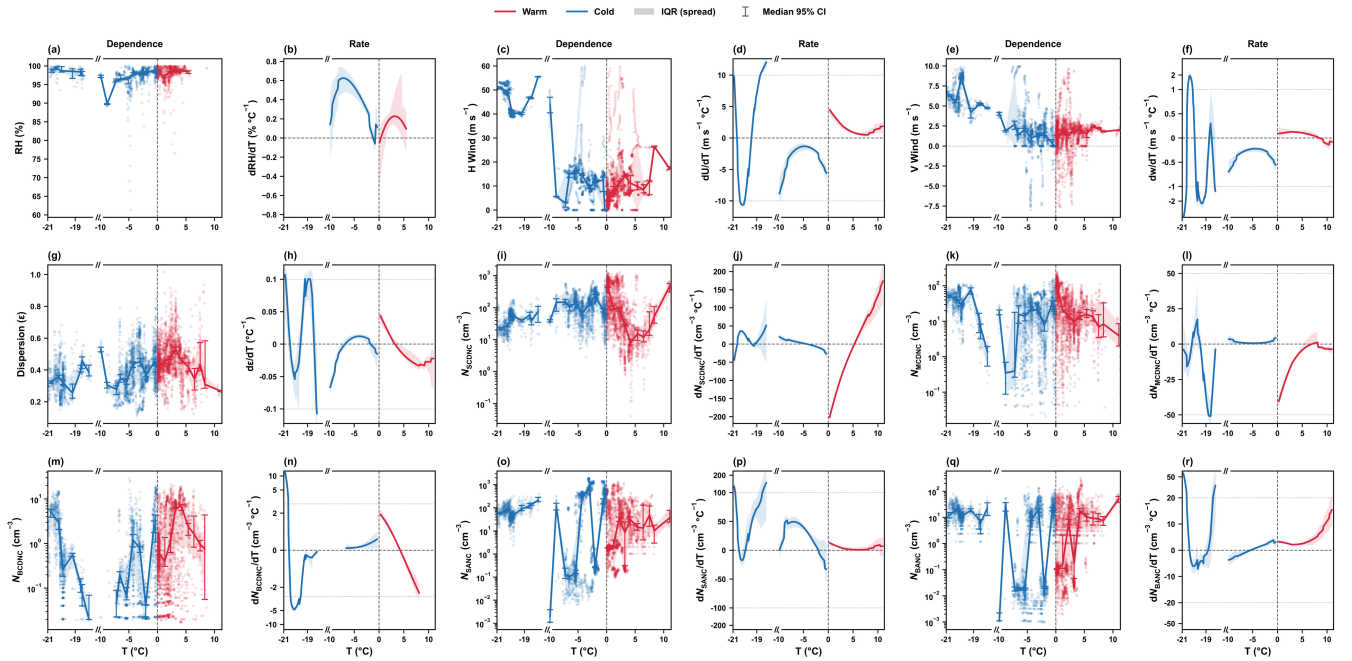


Figure S10: T dependence of environmental and cloud microphysical properties in warm and cold liquid-phase clouds. Dependence on T and the corresponding rates of change, dY/dT , are shown for (a,b) RH, (c,d) H Wind, (e,f) V Wind, (g,h) ε , (i,j) N_{SCDNC} , (k,l) N_{MCDNC} , (m,n) N_{BCDNC} , (o,p) N_{SANC} , and (q,r) N_{BANC} . The vertical dashed line marks the 0°C boundary between warm and cold liquid-phase clouds. The temperature axis is broken (//) between -17.5 and -10.0°C because this interval contained too few samples for reliable binning and was excluded. In the dependence panels, dots show individual samples, lines show binned medians, shading shows interquartile ranges, and error bars show bootstrap 95% confidence intervals. Rate panels show local derivatives from sample-size-weighted local quadratic fits to the binned medians, with shading indicating bootstrap 95% confidence intervals. Rates are computed from untransformed values, so logarithmic concentration axes affect the display only.

References

- Breiman, L.: Bagging predictors, *Mach. Learn.*, 24, 123–140, <https://doi.org/10.1007/BF00058655>, 1996.
- Breiman, L.: Random forests, *Mach. Learn.*, 45, 5–32, <https://doi.org/10.1023/A:1010933404324>, 2001.
- Chen, T. and Guestrin, C.: XGBoost: A scalable tree boosting system, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, 13–17 August 2016, 785–794, <https://doi.org/10.1145/2939672.2939785>, 2016.
- Cover, T. and Hart, P.: Nearest neighbor pattern classification, *IEEE Trans. Inf. Theory*, 13, 21–27, <https://doi.org/10.1109/TIT.1967.1053964>, 1967.
- Dorogush, A. V., Ershov, V., and Gulin, A.: CatBoost: gradient boosting with categorical features support, *arXiv [preprint]*, [arXiv:1810.11363](https://arxiv.org/abs/1810.11363), 2018.
- Drucker, H.: Improving regressors using boosting techniques, in: *Proceedings of the Fourteenth International Conference on Machine Learning*, Nashville, Tennessee, USA, 8–12 July 1997, 107–115, 1997.

- Freund, Y. and Schapire, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting, *J. Comput. Syst. Sci.*, 55, 119–139, <https://doi.org/10.1006/jcss.1997.1504>, 1997.
- Friedman, J. H.: Greedy function approximation: A gradient boosting machine, *Ann. Stat.*, 29, 1189–1232, <https://doi.org/10.1214/aos/1013203451>, 2001.
- Geurts, P., Ernst, D., and Wehenkel, L.: Extremely randomized trees, *Mach. Learn.*, 63, 3–42, <https://doi.org/10.1007/s10994-006-6226-1>, 2006.
- Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Schölkopf, B.: Support vector machines, *IEEE Intell. Syst. Appl.*, 13, 18–28, <https://doi.org/10.1109/5254.708428>, 1998.
- 160 Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y.: LightGBM: A highly efficient gradient boosting decision tree, *Adv. Neural Inf. Process. Syst.*, 30, 3146–3154, 2017.
- Liaw, A. and Wiener, M.: Classification and regression by randomForest, *R News*, 2, 18–22, 2002.
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A.: CatBoost: unbiased boosting with categorical features, *Adv. Neural Inf. Process. Syst.*, 31, 6638–6648, 2018.
- 165 Rumelhart, D. E., Hinton, G. E., and Williams, R. J.: Learning representations by back-propagating errors, *Nature*, 323, 533–536, <https://doi.org/10.1038/323533a0>, 1986.
- Song, Y. Y. and Lu, Y.: Decision tree methods: applications for classification and prediction, *Shanghai Arch. Psychiatry*, 27, 130–135, <https://doi.org/10.11919/j.issn.1002-0829.215044>, 2015.