Supplemental Information for:

On the dynamics of ozone depletion events at Villum Research Station in the High Arctic

Jakob Boyd Pernov[1,2], Jens Liengaard Hjorth[1], Lise Lotte Sørensen[1], and Henrik Skov[1]

[1]Department of Environmental Science, iClimate, Arctic Research Center, Aarhus University, Roskilde, Denmark.
[2]Extreme Environments Research Laboratory, École Polytechnique Fédérale de Lausanne, 1951 Sion, Switzerland.

Correspondence to: Jakob Boyd Pernov (jakob.pernov@epfl.ch) and Henrik Skov (hsk@envs.au.dk)

S1 Machine learning modeling methodology

Here we describe the missing data imputation, the machine learning model, hyperparameter tuning, the ML explainability approach employed, and model evaluation metrics.

Before input into the ML model, missing data were imputed since ML models require no missing data in the input files. We imputed missing data using the median value for the hour of the day for that day of the year. This imputation approach allows us to account for changes occurring from early to late spring as well as diurnal changes, which would otherwise be overlooked if only using a single median for the spring months. This is especially important for variables that drastically change over this short period (e.g., temperature, RH, solar radiation). Table S1 lists the percentage of missing data before imputation for each variable. Wind speed and direction exhibited the highest percentage of missing data, with both missing ~21 %, therefore data imputation shouldn't adversely affect the results of the ML model. No feature engineering (standardization or normalization) was applied prior to modeling since the initial evaluation metrics were deemed sufficiently accurate. No temporal information (Julian day, day of year, hour of day) was included in the input variables.

The XGBoost model was selected as the model used in this study due to its accuracy, computational efficiency, and ability to handle collinearity amongst the input variables, which is important for meteorological variables. XGBoost is an ensemble machine learning algorithm using the gradient-boosting methodology on individual decision trees (which are weak learners) and then builds multiple decision trees that are sequentially added (Chen and Guestrin, 2016). This allows for the previous tree's errors to be learned by the next tree, therefore reducing the loss function while obtaining the best prediction. A regularized model formalization is used in the XGBoost model to improve computational efficiency and prevent over-fitting. The xgboost package (v1.6.2) was used and all ML modeling was implemented in a Python environment (v3.10.2).

Hyperparameter tuning is an essential part of ML which ensures optimal model performance. We utilized a Bayesian approach for exploring the optimum hyperparameter configuration, implemented though the Optuna (Akiba et al., 2019) library (v3.0.3). The hyperparameters included, the range of values explored, and the optimum values are listed in Supplementary Table 2. This study employed a 70/30

train/test split ratio. The objective of the hyperparameter tuning procedure is to maximize the mean recall score using 10-fold cross-validation of the training set. Tuning was performed for 1000 trials and the best parameters were selected. Hyperparameter values were sampled using the Tree-structured Parzen Estimator (TPE) algorithm (Bergstra et al., 2011) and trials were pruned using the Hyperband pruner (Li et al., 2018). The final set of hyperparameters was selected based on the compromise between overall performance (high recall scores) and agreement between the training and test set evaluation metrics using 10-fold cross-validation (prevention of over-fitting).

We employed SHapley Additive exPlanations (SHAP) values (Lundberg and Lee, 2017) which are based on Shapely values (Shapley, 1953), to assess the effect of the input variables on the model output. The SHAP approach is a model-agnostic methodology designed to assess input variable importance based on coalitional game theory (Molnar, 2022), where input variables are treated as "players" in a "game" (model framework) and SHAP aims to assess the players' contribution to the "payout" (model output). For each observation, the SHAP value represents an input variable's marginal contribution over the mean model output when considering all possible combinations of the input variables. SHAP values can be positive or negative, with positive values indicating a variable is more likely to contribute to an observation being predicted as an ODE while negative values mean a variable is more likely to contribute to an observation being labeled as a Non-ODE. It is important to note that SHAP values do not represent how well the input variables explain the behavior of our target variable in the natural environment but how well these variables explain the behavior of our target variable in our model, therefore SHAP values represent purely statistical relationships. SHAP can produce both local and global explanations contrary to other commonly used input variable importance methods (e.g., split count, gain, permutation importance) that only produce an estimate of global importance (Lundberg et al., 2019). The global importance for each feature is calculated as the mean of the absolute SHAP values for said input variable which gives an overview of the most important variables, however, this does not account for the relationship between the SHAP and input value (positive or negative relationship, linear or non-linear). Therefore, we assessed the relationship between the SHAP and ambient values by discretizing the ambient values into fifteen equally spaced bins and calculated the median and 25^{th}/75^{th} percentiles for each bin. These two approaches allow for the evaluation of the overall global importance as well as the relationship between ambient and SHAP values for each input variable. The SHAP approach was applied via the shap package (v0.41.0).

The ML model was evaluated using common metrics for a classification model, namely accuracy, recall, and Area Under Curve Receiver Operating Characteristics (AUC ROC). The accuracy is the fraction of correctly labeled data, both positive (ODEs) and negative (Non-ODEs), compared to the total number of data points (sum of ODEs and Non-ODEs) and ranges from 0 to 1. In other words, accuracy is the fraction of correctly predicted observations regardless of label (ODE vs Non-ODE). The recall (also defined as the true positive rate or sensitivity) is the fraction of correctly identified positive labels (ODEs identified by the ML model) compared to the total number of positive labels (total number of ODEs) and ranges from 0 to 1. In other words, recall is the fraction of ODEs correctly predicted. The ROC curve displays the performance of a classification model across different decision thresholds and is represented by a plot of the true positive rate versus the false positive rate. The AUC ROC is the area underneath the ROC curve and evaluates how well a model can discriminate between positive and negative labels across all decision thresholds (0.5 is the default threshold used in this study). The AUC ROC ranges from 0 to 1, with 0.5 representing random chance and 1 representing a perfect model. The accuracy gives an overview of the model performance for both labels (ODEs vs Non-ODEs), recall gives the model performance only for

positive labels (ODEs), and AUC ROC evaluates the model performance over different decision thresholds, together, these three metrics give a comprehensive view of the model's performance. These metrics were implemented using the scikit-learn package (v1.0.2).

Table S1. Percentage of missing data before imputation for 2007-2019.

Variable	% Missing before imputation
Above Mixed Layer	0
Pressure	12.76
Radiation	0
RH	18.269
Sea Ice	0
Temperature	13.07
Wind Direction	21.60
Wind Speed	21.49
Snow	0

Table S2. Overview of the hyperparameter optimization for the ML model.

Hyperparameter	Short name	Range	Optimal Value
Number of estimators	n_estimators	100-1000	550
Maximum tree depth	max_depth	3-7	5
L1 regularization	reg_alpha	0-10	8.1
L2 regularization	reg_lambda	0-10	1.8
Minimum child weight	min_child_weigh	1-10	8
Gamma	gamma	0-10	0.6
Learning rate	learning_rate	0.01-0.1	0.092
Subsample fraction	subsample	0.5-1.0	0.95
Column sample by tree fraction	colsample_bytree	0.5-1.0	0.9
Positive Label Scalar	scale_pos_weight	1-10	5

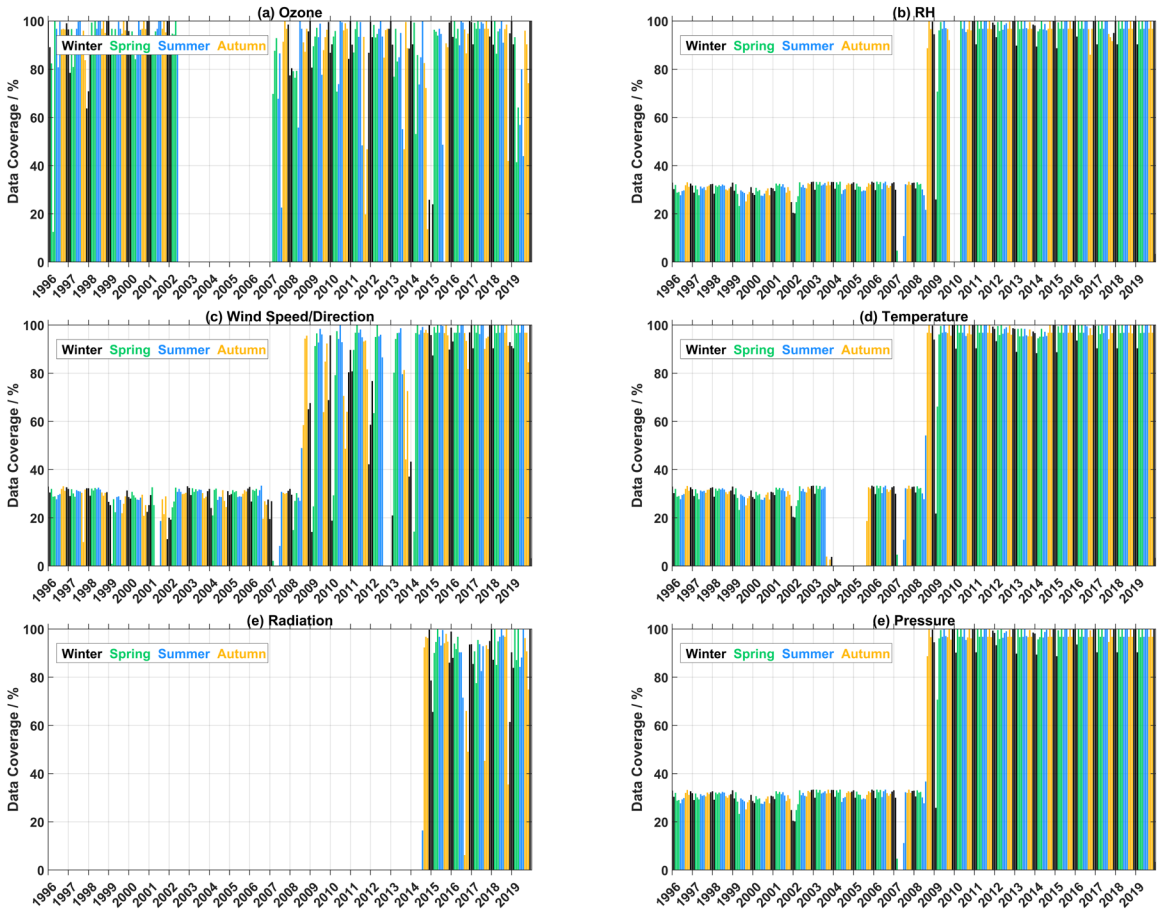

Figure S1. Data coverage over 1996-2019 for (**a**) ozone and (**b-e**) the meteorological variables as expressed as a percentage of available measurements relative to the total possible number of measurements for each month. Months are color-coded by season.

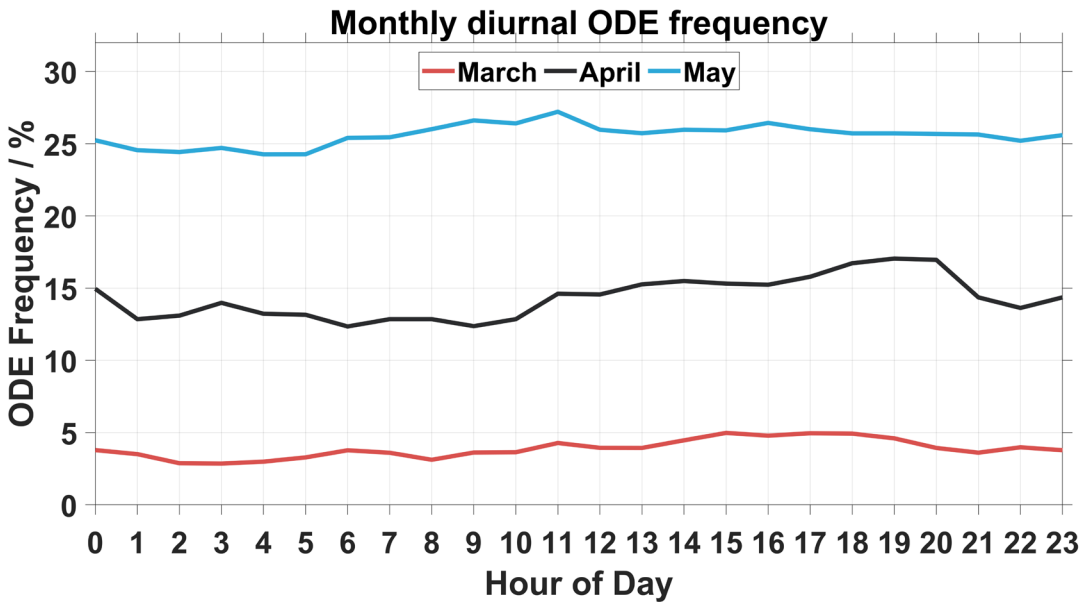

Figure S2. Diurnal ODE frequency by month.

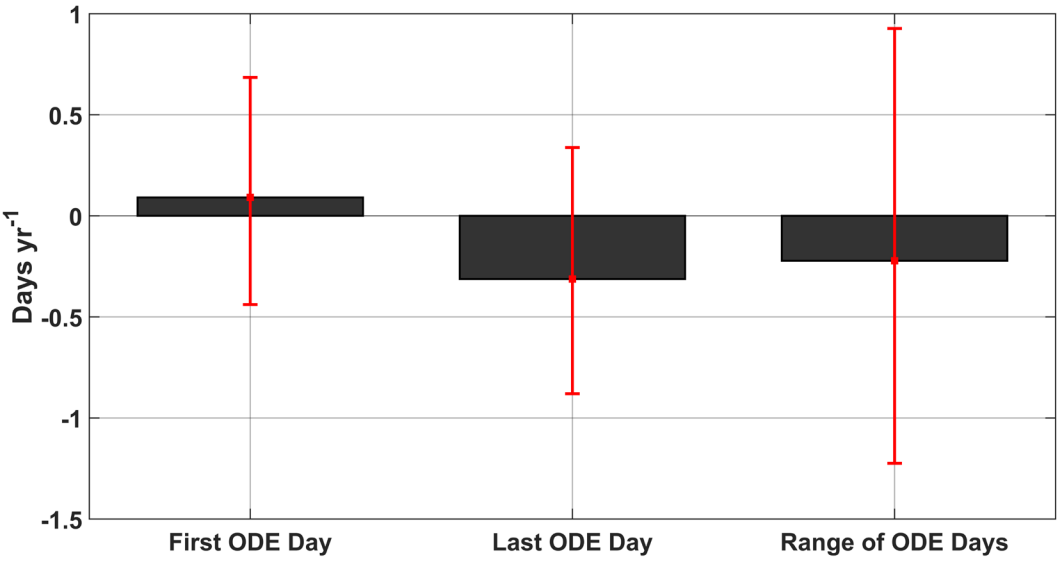

Figure S3. Trend analysis of the first ODE day (defined as the first day of the year with an ozone measurement < 10 ppbv), the last ODE day (defined as the last day of the year with an ozone measurement < 10 ppbv), and the range of the ODE season (last day of the year minus the first day of the year). The blue bars represent trends that are significant on the 95th % confidence level while the black bars are not. The red error bars represent the 95th % confidence intervals of the slope. The p values for first, last, and range of ODE days are 0.78, 0.20, and 0.65, respectively.

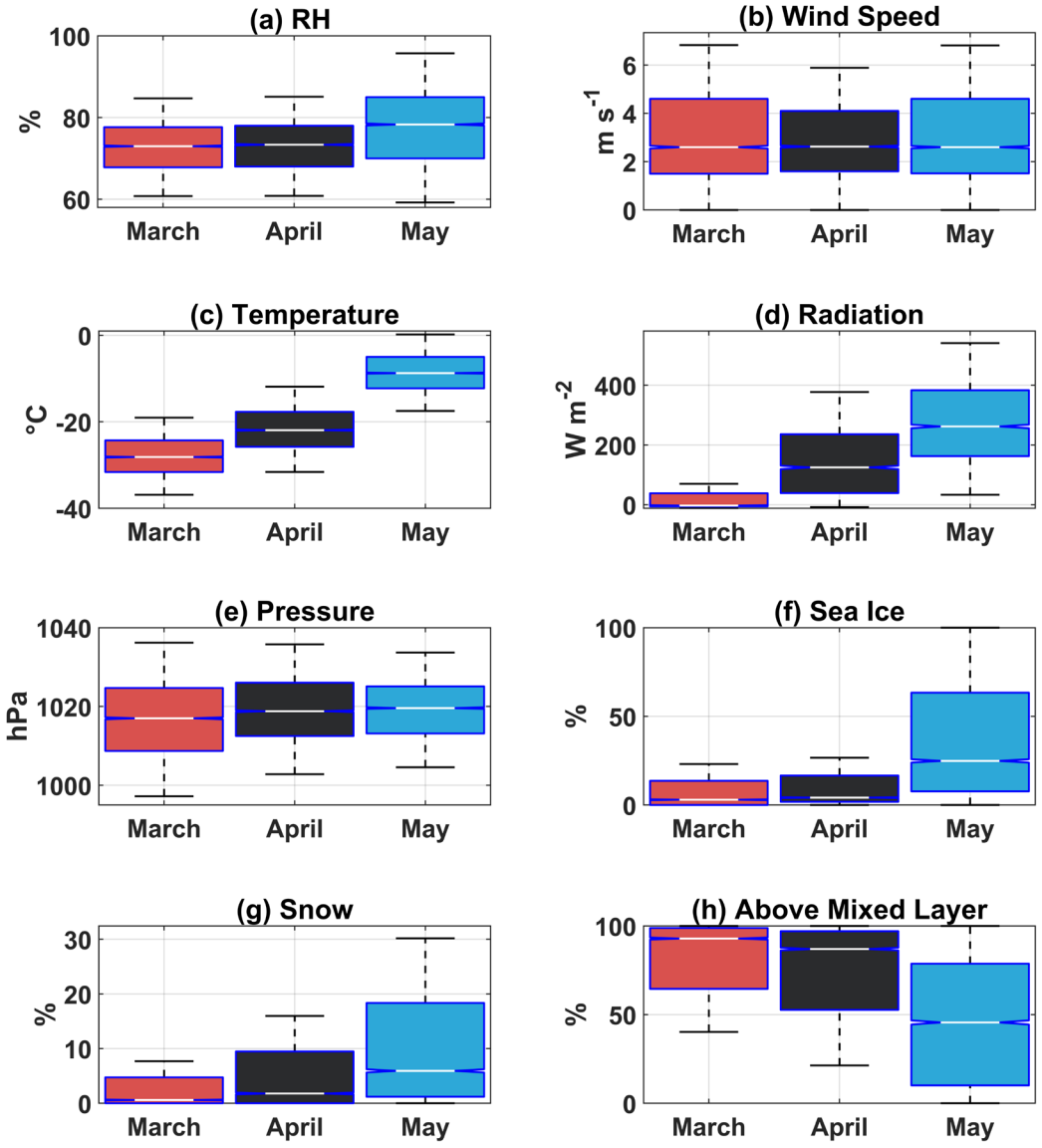

Figure S4. Distribution of meteorological and air mass history variables during each spring month including (**a**) RH, (**b**) wind speed, (**c**) temperature, (**d**) radiation, (**e**) pressure, (**f**) time over sea ice, (**g**) time over snow, and (**h**) time above the mixed layer. The line in the middle of the box represents the median, the boxes represent the interquartile range, the medians of boxes whose notches do not overlap differ with 95% confidence. For a description of how the time spent over different surface types is calculated see the methods section. All available data for each variable collocated with ozone measurements was used, resulting in different years used for each variable, with the minimum number of years included was 5 for radiation.

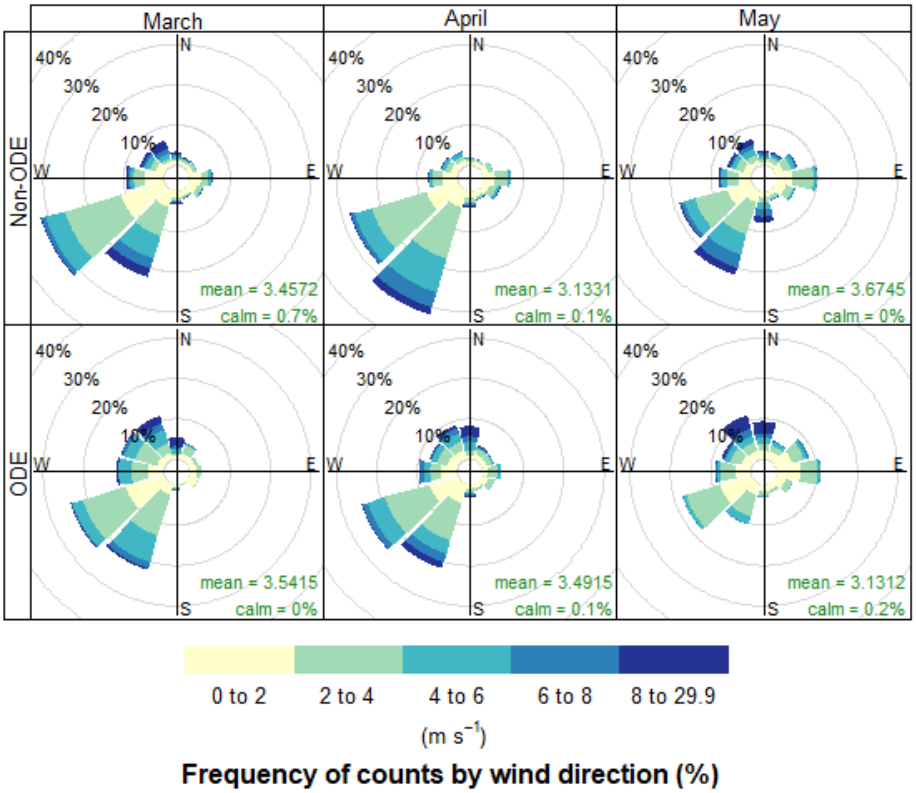

Figure S5. Wind roses for Non-ODEs (top row) and ODEs (bottom row) for the spring months. The mean wind speed and the percentage of time the wind speed is zero (or calm) is given in each panel.

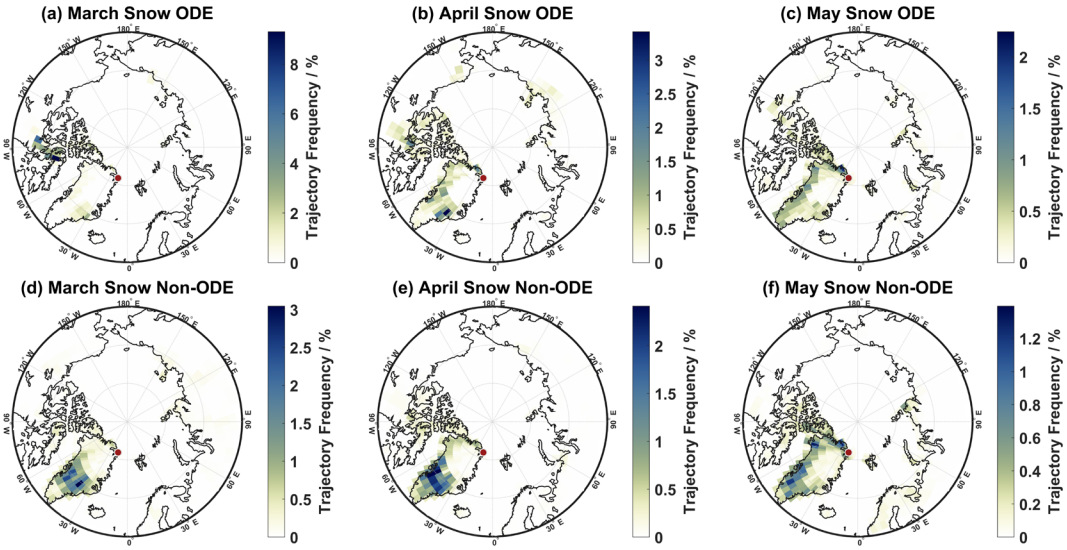

Figure S6. Trajectory frequency maps for trajectory steps below the mixed layer and over snow for (**a-c**) ODEs and (**d-f**) Non-ODEs during March, April, May at Villum (indicated by the red and white circle).

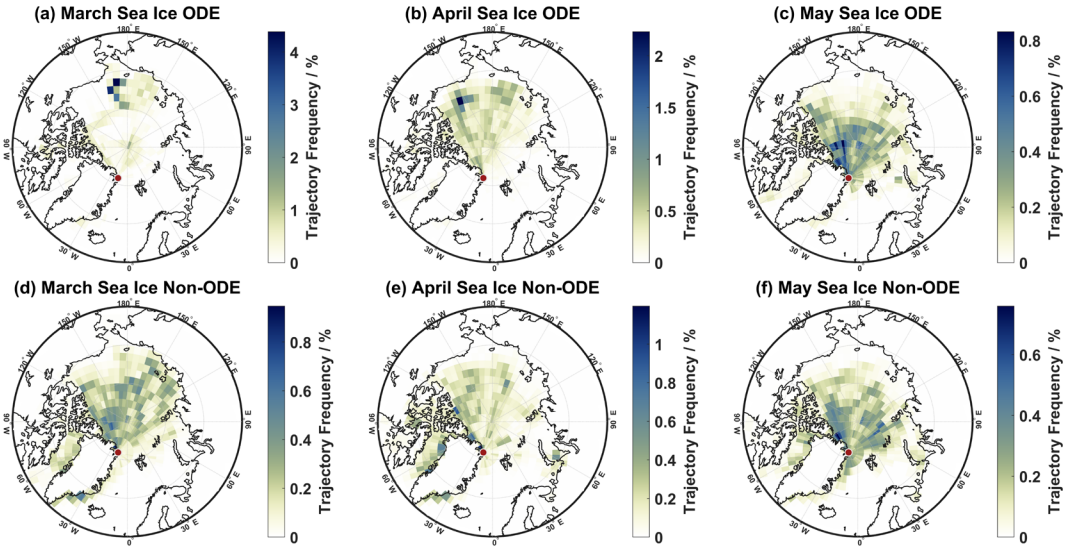

Figure S7. Trajectory frequency maps for trajectory steps below the mixed layer and over sea ice for (**a-c**) ODEs and (**d-f**) Non-ODEs during March, April, May at Villum (indicated by the red and white circle).

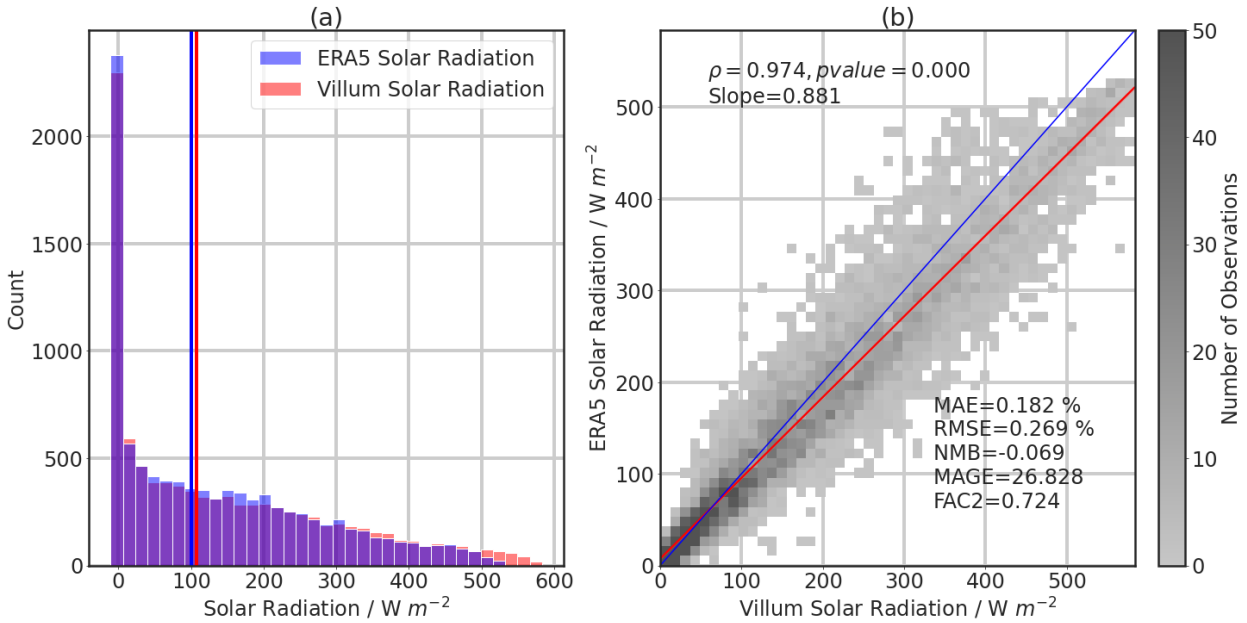

Figure S8. Comparison between observations of solar radiation and shortwave solar radiation downwelling from ERA5 Reanalysis, with (**a**) a histogram and (**b**) scatterplot. On the scatterplot, the spearman rank correlation coefficient along with its associated p-value are presented in the top left corner while the mean absolute error (MAE), root mean square error (RMSE), normalized mean bias (NMB), mean absolute gross error (MAGE), and fraction of modeled data with a factor of 2 (FAC2) of the observations are presented in the bottom right corner. The MAE and RMSE are given as percentages of the mean observational value.

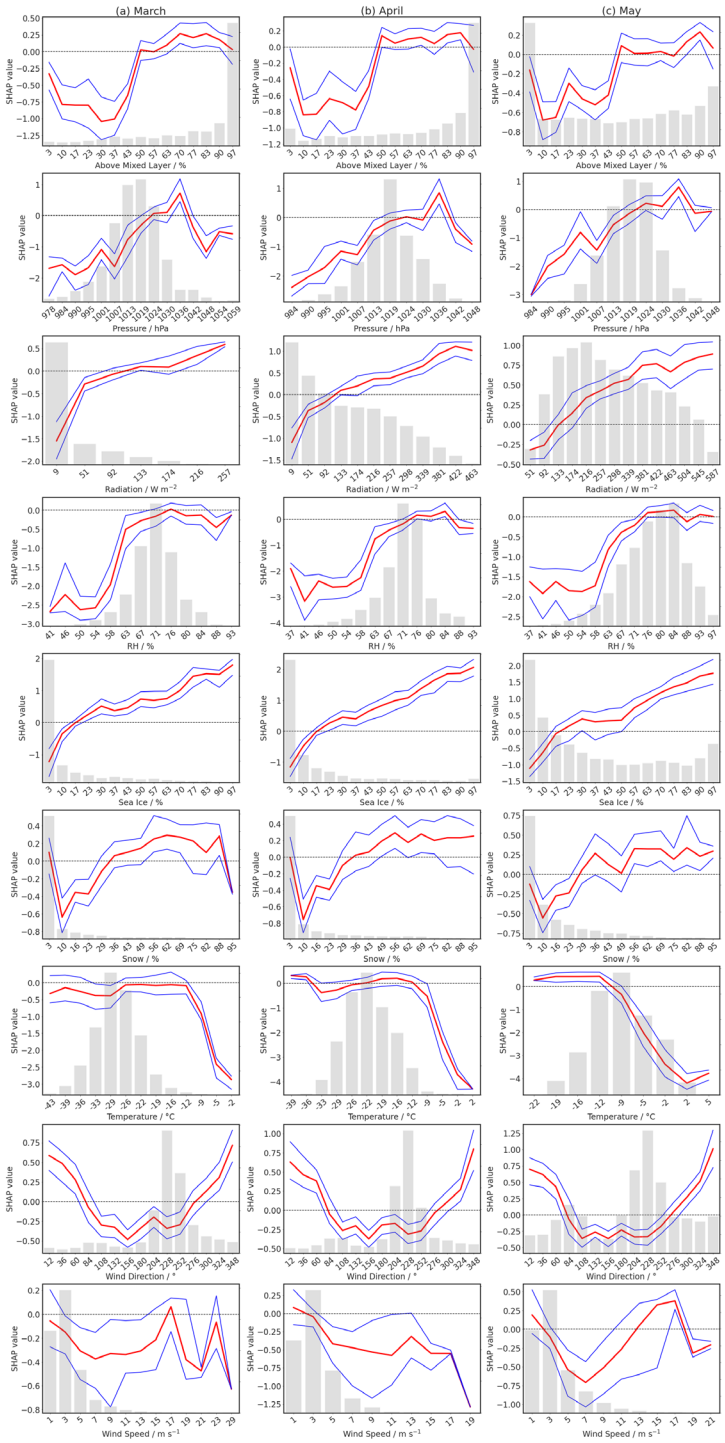

Figure S19. The relationships between SHAP and ambient values for (**a**) March, (**b**) April, and (**c**) May. 15 equally spaced bins were calculated for each feature, the median (red line) and IQR (blue lines) of the SHAP values were computed for each bin. The value listed on the x-axis is the midpoint of each bin. The light gray bars represent a histogram of the ambient values, whose axis was omitted for clarity. The bins are the same used in Fig. 9.

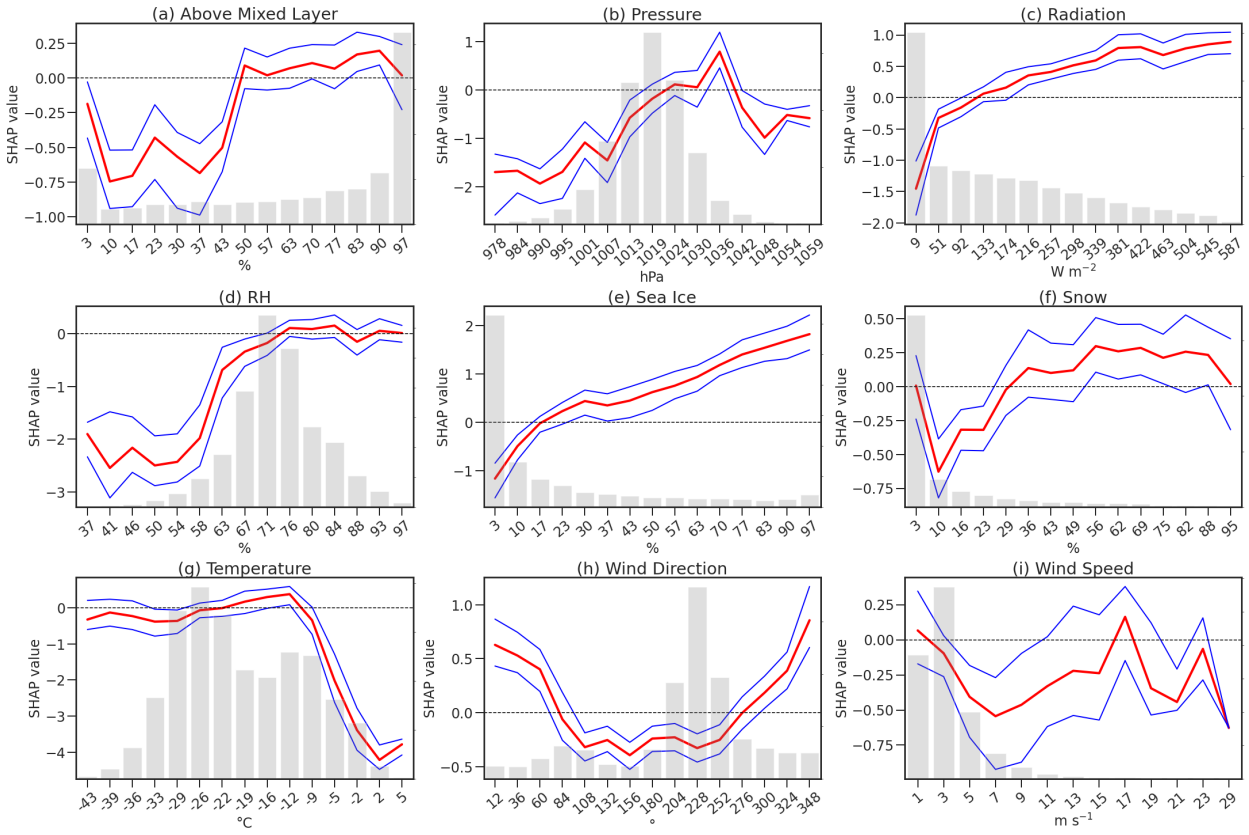

Figure S10. The relationships between SHAP and ambient values for March – May combined for (**a**) time above the mixed layer, (**b**) pressure, (**c**) radiation, (**d**) RH, (**e**) time air masses spent over sea ice, (**f**) time air masses spent over snow, (**g**) temperature, (**h**) wind direction, and (**i**) wind speed. Fifteen equally spaced bins were calculated for each feature, the median (red line) and IQR (blue lines) of the SHAP values were computed for each bin. The value listed on the x-axis is the midpoint of each bin. The light gray bars represent a histogram of the ambient values, whose axis was omitted for clarity.

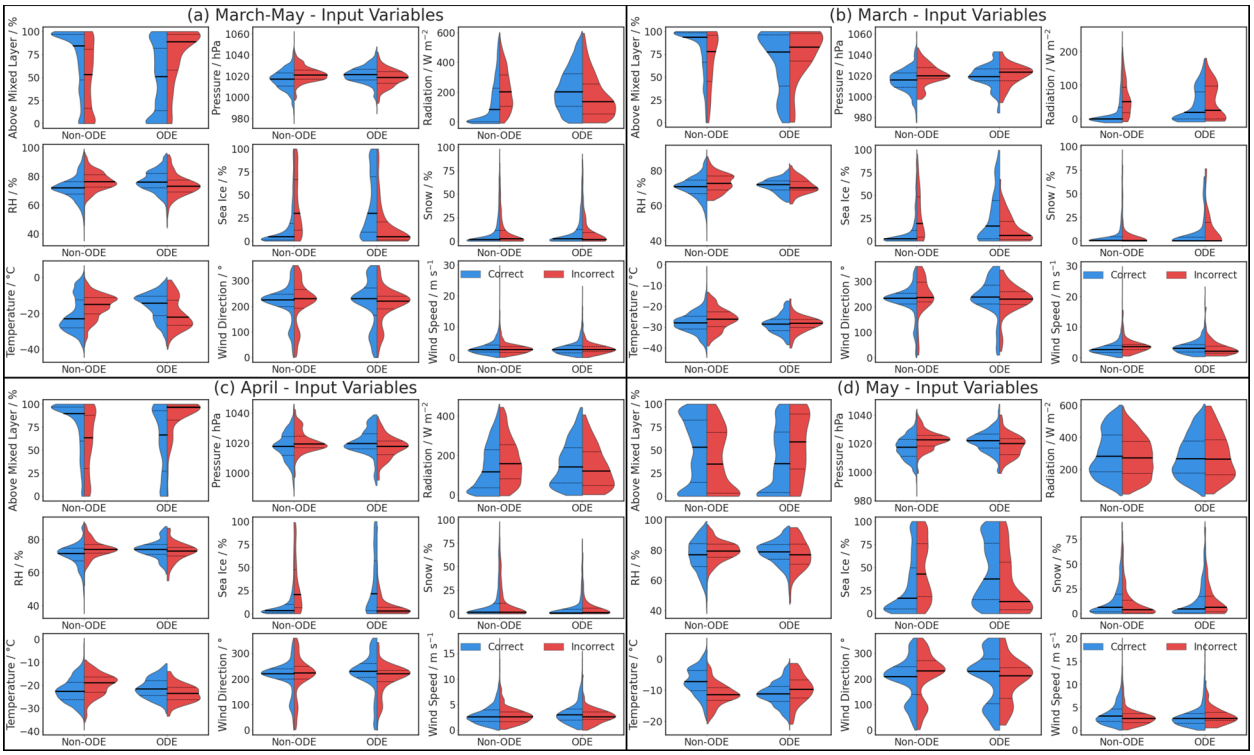

Figure S11. Distributions of input variables for Non-ODEs and ODEs for (**a**) March-May, (**b**) March, (**c**) April, (**d**) May, the thick black line represents the median and thin dashed lines represent the 25th and 75th percentiles. The distributions are color-coded by correct (blue) or incorrect (red) prediction by the ML model. All data over 2007-2019 was used to predict ODEs and was used in this analysis.

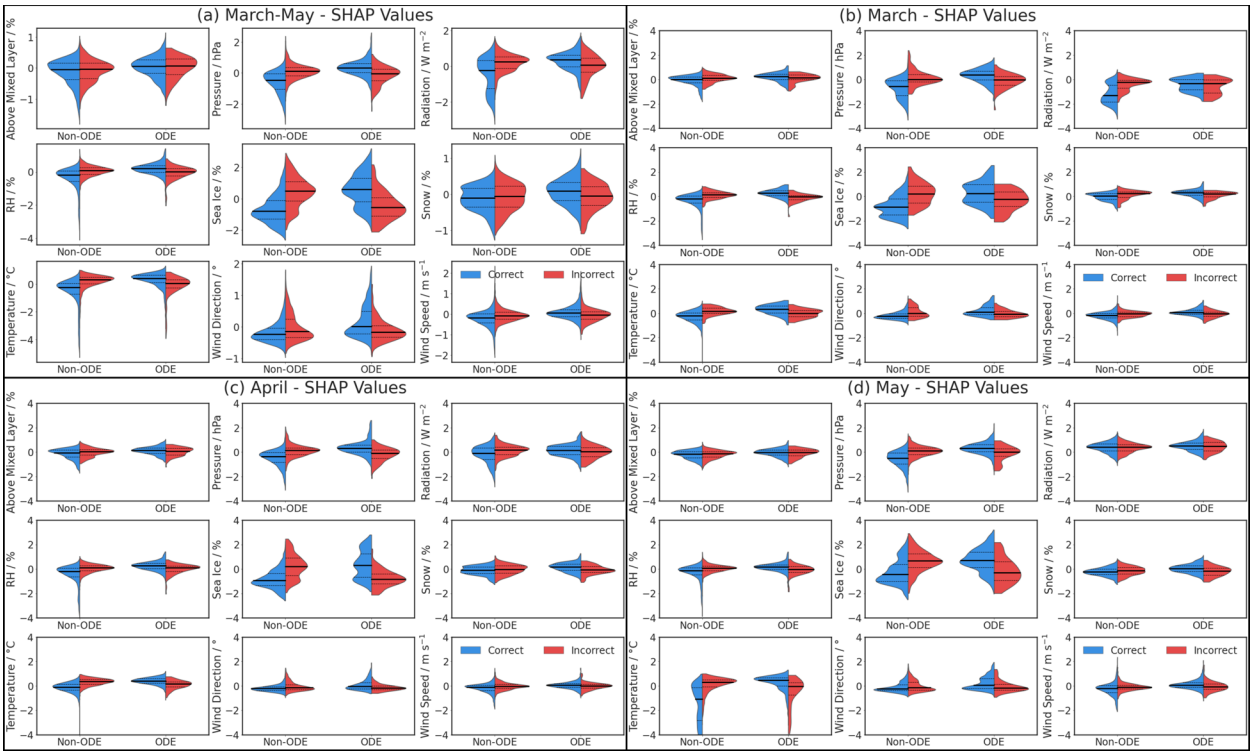

Figure S12. Distributions of SHAP values for Non-ODEs and ODEs for (**a**) March-May, (**b**) March, (**c**) April, (**d**) May, the thick black line represents the median and thin dashed lines represent the 25th and 75th percentiles. The distributions are color-coded by correct (blue) or incorrect (red) prediction by the ML model. All data over 2007-2019 was used to predict ODEs and was used in this analysis.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M.: Optuna: A Next-Generation Hyperparameter Optimization Framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, event-place: Anchorage, AK, USA, 2623–2631, https://doi.org/10.1145/3292500.3330701, 2019.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B.: Algorithms for Hyper-Parameter Optimization, in: Advances in Neural Information Processing Systems, 2011.

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, New York, NY, USA, event-place: San Francisco, California, USA, 785–794, https://doi.org/10.1145/2939672.2939785, 2016.

Lundberg, S. M. and Lee, S.-I.: A Unified Approach to Interpreting Model Predictions, in: Advances in Neural Information Processing Systems, https://doi.org/10.48550/arXiv.1705.07874, 2017.

Lundberg, S. M., Erion, G. G., and Lee, S.-I.: Consistent Individualized Feature Attribution for Tree Ensembles, 2019.

Molnar, C.: Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2nd ed., 2022.

Shapley, L. S.: 17. A Value for n-Person Games, in: Contributions to the Theory of Games (AM-28), Volume II, edited by: Kuhn, H. W. and Tucker, A. W., Princeton University Press, Princeton, 307–318, https://doi.org/doi:10.1515/9781400881970-018, 1953.