We divided our dataset into three groups: a training set, for learning; a validation set, for hyperparameter tuning; and a test set, to evaluate performance. We split the dataset by whole days instead of individual data points to prevent any spatial autocorrelation from inflating performance metrics. From 2,115 total days (comprising 3,434,703 altitude bins across 143 radar stations after filtering steps), we randomly selected 75% of days for training, 10% for validation, and 15% for testing.

We tuned model hyperparameters with grid searches across hyperparameter space (fig. S10). For our first search, we set the learning rate *eta* to 0.05 while varying maximum tree depth *max_depth* between 8-16. Trees of these depths are complex, but predicting bird migration across the entire United States from March to May at 30 different altitude bins is a complex problem. We used the *early_stopping_rounds* argument to stop the algorithm after 10 boosting iterations in which performance on the validation set failed to improve. Larger trees perform better on training data, but trees that are too large lower performance due to overfitting. We therefore used the validation dataset to select the best-performing value of maximum tree depth. We then tested the following modifications to additional parameters that can prevent overfitting: decreasing *subsample* from 1.0 to 0.70, increasing *min_child_weight* from 1 to 5, and increasing *gamma* from 0 to 1 or 10. We tried all 12 combinations of these modifications. The best combination of parameters was the following: *max_depth* = 12, *min_child_weight* = 5, *gamma* = 1, *colsample_bytree* = 1, and *subsample* = 0.7, Using the best combination of hyperparameters, we further lowered the learning rate to 0.01 and set *early_stopping_rounds* to 100 to determine the optimal number of boosting iterations for that learning rate. Lower learning rates decrease the contribution of each tree to the model, making the boosting algorithm more conservative and further preventing overfitting, but lower learning rates require more iterations. With this information, we fit a final model with learning rate = 0.01 on the combined training and validation sets. We then evaluated its performance on the test dataset (15% of data), which had been withheld from all training and validation. To assess performance, we calculated two metrics: root mean square error and the coefficient of determination (or R^2). We calculated R^2 by dividing the sum of squared errors by the total sum of squares, and then subtracting this value from 1. An R^2 value of 0 indicates that the model does not explain the data any better than a simple null model that predicts the mean for each observation, while a negative R^2 value indicates that the model explains the data worse than this null model.

In an XGBoost model, correlated or uninformative predictors generally have little negative effect; they will generally not be incorporated during tree construction. However, extraneous predictors increase computational time and data storage requirements, making the forecast system more unwieldy to operationalize. For this reason, we sought to remove uninformative predictors. Using the *xgboost* package, we calculated the gain, a predictor importance metric that quantifies how much a tree improves by adding a split on a given variable. Gain values are scaled to sum to 1. After the first grid search step, we identified and eliminated predictors with gain values less than 0.01 and restarted the tuning procedure. In this manner, we eliminated albedo, vertical velocity, convective available potential energy, total precipitation, and snow cover. This left 12 variables in the final model: ordinal date, height above ground level, latitude, longitude, air temperature, relative humidity, zonal wind, meridional wind, surface pressure, mean sea level pressure, visibility, and total cloud cover.

We trained and tested two further modifications to the final model: one which also included additional conditions variables from the previous night (winds, temperature, and surface pressure) and their 24-hour changes, and another which included these lagged weather variables