

Classifying Convective Storms Using Machine Learning

G. ELI JERGENSEN AND AMY MCGOVERN

University of Oklahoma, Norman, Oklahoma

RYAN LAGERQUIST

University of Oklahoma, and Cooperative Institute for Mesoscale Meteorological Studies, Norman, Oklahoma

TRAVIS SMITH

*University of Oklahoma, and Cooperative Institute for Mesoscale Meteorological Studies, and
National Severe Storms Laboratory, Norman, Oklahoma*

(Manuscript received 12 August 2019, in final form 10 December 2019)

ABSTRACT


We demonstrate that machine learning (ML) can skillfully classify thunderstorms into three categories: supercell, part of a quasi-linear convective system, or disorganized. These classifications are based on radar data and environmental information obtained through a proximity sounding. We compare the performance of five ML algorithms: logistic regression with the elastic-net penalty, random forests, gradient-boosted forests, and support-vector machines with both a linear and nonlinear kernel. The gradient-boosted forest performs best, with an accuracy of 0.77 ± 0.02 and a Peirce score of 0.58 ± 0.04 . The linear support-vector machine performs second best, with values of 0.70 ± 0.02 and 0.55 ± 0.05 , respectively. We use two interpretation methods, permutation importance and sequential forward selection, to determine the most important predictors for the ML models. We also use partial-dependence plots to determine how these predictors influence the outcome. A main conclusion is that shape predictors, based on the outline of the storm, appear to be highly important across ML models. The training data, a storm-centered radar scan and modeled proximity sounding, are similar to real-time data. Thus, the models could be used operationally to aid human decision-making by reducing the cognitive load involved in manual storm-mode identification. Also, they could be run on historical data to perform climatological analyses, which could be valuable to both the research and operational communities.

1. Introduction

Storm-mode classification is an important task for both real-time weather forecasting and climatological analysis. As the National Weather Service builds next-generation forecast systems that make use of automated technology [e.g., probabilistic hazard information, discussed in Gallo et al. (2017) and Rothfus et al. (2018)], a real time system that classifies storm-mode can help guide automated warnings, since storm-mode is correlated with hazards such as hail and tornadoes (Smith et al. 2012; Thompson et al. 2012). In addition, accurate

and automated storm-mode classification would allow for long-term climatologies. These climatologies could answer questions such as “how often do supercells occur at my location?” or “what is the most common convective mode at my location?” These questions could also be broken down by time of day, time of year, synoptic régime, etc. Furthermore, we could study trends in convective mode as a function of climate change or internal climate variability (e.g., the El Niño–Southern Oscillation).

The existence of different convective modes has been known for decades. Byers (1949) describes the life cycle of a single-cell thunderstorm and its structure and dynamics during each stage. This conceptual model has three stages: the cumulus stage, in which the updraft is shallow but deepening and there is no downdraft; the mature stage, in which the updraft is at maximum depth (often up to the tropopause) and is adjacent

 Denotes content that is immediately available upon publication as open access.

Corresponding author: Amy McGovern, amcgovern@ou.edu

DOI: 10.1175/WAF-D-19-0170.1

© 2020 American Meteorological Society. For information regarding reuse of this content and general copyright information, consult the [AMS Copyright Policy](#) (www.ametsoc.org/PUBSReuseLicenses).

to a well-developed downdraft; and the dissipating stage, in which the downdraft comes to dominate the low levels and chokes off the low-level inflow to the updraft, thus extinguishing the storm. This life cycle typically lasts 30–60 min (Wallace and Hobbs 2006, their section 8.3.2a). In Byers's conceptual model the single cell does not interact with other cells, except that its cold outflow may trigger the development of new cells. However, Byers acknowledges the existence of multicellular storms. Single-cell storms of the Byers type are usually not associated with severe weather (Wallace and Hobbs 2006, their section 8.3.2a).

Quasi-linear convective systems (QLCS) are a type of mesoscale convective system (Houze 2004). The strongest updrafts (storm cells) form a line or arc and are often adjacent to a large region of stratiform precipitation. The classical structure of a QLCS is leading line–trailing stratiform, in which the stratiform precipitation is behind the leading line (Houze and Hobbs 1982). QLCSs often begin with a single cell, which in its dissipating stage produces a downburst that propagates away as a gust front, along which new cells initiate (Johns 1993). The new cells may undergo the same process, producing an even longer gust front, until the system grows to lengths on the order of 100–1000 km. Rising air at the leading line propagates toward the back of the system, forming a mesoscale region of ascending front-to-rear flow. This is balanced by descending rear-to-front flow, which may form a rear-inflow jet, descending to the surface near the leading line and strengthening the gust front, which allows the system to persist. QLCSs usually last 6–12 h (Parker and Johnson 2000) and sometimes evolve into bow echoes or derechos, which can produce extreme surface winds (Coniglio et al. 2004). QLCSs sometimes produce tornadoes, but these tornadoes tend to be weaker than their supercellular counterparts (Thompson et al. 2012).

Supercells are storms with a strong updraft and collocated mesocyclone (Lemon and Doswell 1979), which is a vortex with diameter and depth from a few kilometers to 10 km (Stumpf et al. 1998). The formation of a mesocyclone depends on strong vertical wind shear in the prestorm environment (Lemon and Doswell 1979). Wind shear also creates an upward-directed perturbation pressure-gradient force, which strengthens the updraft, and shear carries hydrometeors away from the updraft as they move aloft. This latter effect allows horizontal separation between the updraft and precipitation-loaded downdraft, which prevents the downdraft from extinguishing the updraft and allows the storm to persist for several hours. Supercells typically move to the right of the environmental steering wind in the Northern Hemisphere, as a result of continual regeneration of the updraft on its right flank and dissipation on its left flank (Davies-Jones 2002). Supercells

are responsible for a majority of tornadoes and a large majority of violent tornadoes and often produce other types of severe weather, including straight-line wind and hail (Thompson et al. 2012).

Despite decades of research on different convective modes, few objective classification schemes existed before the installation of the Next-Generation Radar system (NEXRAD; Crum and Alberty 1993), completed in the mid-1990s. Fowle and Roebber (2003) classify storms as linear, multicellular, or isolated (single cell). A “linear” storm is one with a high-reflectivity area (radar reflectivity >40 dBZ) covering at least 500 km^2 , persisting for at least 3 h, and with a length-to-width ratio of at least 3. A “multicellular” storm is one that meets all these criteria except the length-to-width ratio, and an “isolated” storm is a high-reflectivity area covering less than 500 km^2 . Meanwhile, Trapp et al. (2005) classify storms as a cell, QLCS, or other. In their scheme a “cell” is a circular or elliptical region of nonzero reflectivity with maxima typically ≥ 50 dBZ, whereas a QLCS is a quasi-linear region of reflectivity ≥ 40 dBZ with length >100 km. Their “other” category mostly contains tornadic outer rainbands of tropical cyclones that have made landfall. Many studies have focused only on the classification of QLCSs. For example, Bluestein and Jain (1985) classify QLCSs into four types: broken line, back building, broken areal, and embedded areal (their Fig. 1). Meanwhile, Parker and Johnson (2000) classify QLCSs into those with trailing, leading, and parallel stratiform precipitation (their Fig. 4) while acknowledging that some QLCSs have no stratiform precipitation. Gallus et al. (2008) build on these works and classify storms into nine types: isolated cells, clusters of cells, nonlinear systems, and six types of QLCS (their Fig. 2). Last, objective supercell-detection schemes have generally used a threshold on the linear correlation between vertical velocity and vertical vorticity (Clark 1979; Weisman and Klemp 1984; Knupp et al. 1998) or on the updraft helicity (Kain et al. 2008; Sobash et al. 2011).

Smith et al. (2012, hereinafter S12) developed a classification scheme that contains five major categories: supercell, part of a QLCS, disorganized, linear hybrid, and marginal supercell. “Disorganized” storms are single cells and multicellular clusters that do not clearly achieve supercell or QLCS criteria; “linear hybrids” are right-moving supercells embedded in a QLCS; and “marginal supercells” are those with wind shear $<20 \text{ m s}^{-1}$ from 0 to 6 km above ground level (Thompson et al. 2003). As noted in S12, this scheme can be simplified to three categories: supercells, QLCS, and other, as in Trapp et al. (2005) but with two differences. First, S12 use a reflectivity threshold of 35 dBZ for all storms; second, their “other” category contains all

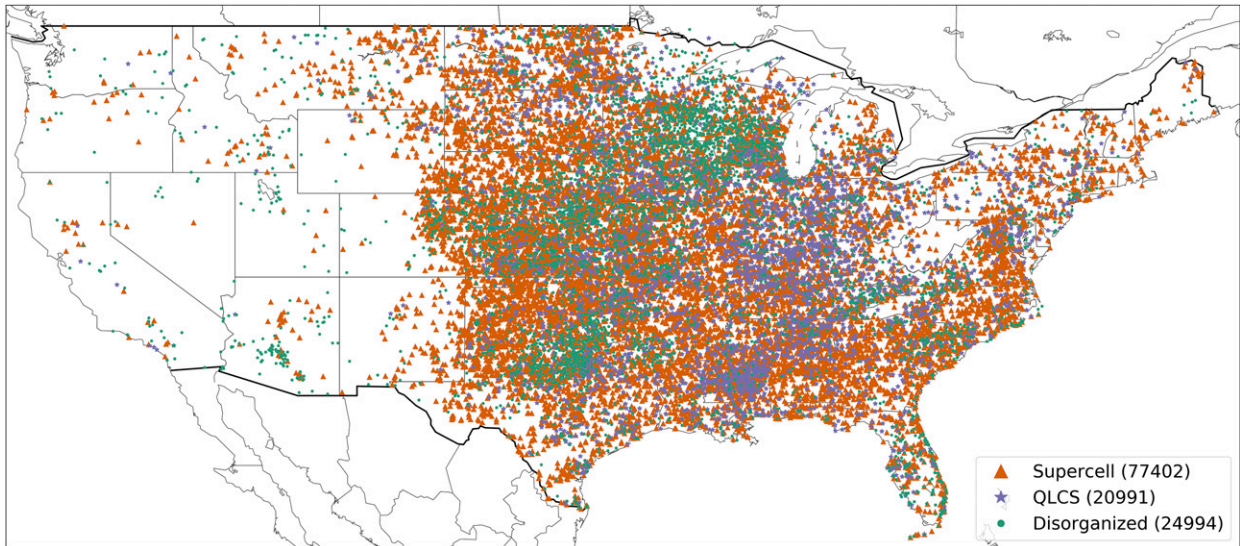


FIG. 1. Spatial distribution of storm objects with each convective mode in the dataset (2004–11).

disorganized single cells and clusters, not only rainbands in tropical cyclones. In general, each storm identified by S12 contains a single dominant updraft. This study adopts the simplified three-category scheme.

S12 classify only storms associated with a tornado, significant-severe hail (diameter ≥ 2 in. or 50.8 mm), or significant-severe wind gust (≥ 65 kt or 33.4 m s^{-1}). Henceforth, we will refer to these as “significant-severe storms.” Their dataset totals 22 901 storms from the years 2004–11 (Fig. 1), and all storms are classified manually. They note that, of the five major categories, linear hybrids and marginal supercells are the most difficult to identify, with some potential linear hybrids requiring multiple examinations by outside experts. In a personal communication with the authors, they estimate that it took the equivalent of 3.5 years of a person working full time to create these data. S12 note that, had they used severe thresholds (25.4 mm and 25.7 m s^{-1}) rather than significant-severe thresholds, there would have been enough storms to make the cost of labeling prohibitive.

These difficulties highlight the benefits of machine learning (ML), which can achieve human-level accuracy in image classification in a small fraction of the time (Quartz 2017). The success of ML-based image classification in meteorology has been less dramatic, likely because definitions of meteorological phenomena (e.g., linear hybrids) are less objective. Nonetheless, it has achieved notable successes in meteorology as well. For example, Wang et al. (2016) use ML to detect sea ice concentration in satellite imagery; Liu et al. (2016), to detect extreme-weather patterns in climate-model output; Chilson et al. (2019), to detect bird roosts in radar imagery; Lagerquist et al. (2019), to detect warm and cold

fronts in reanalysis data; and Wimmers et al. (2019), to classify tropical-cyclone intensity from satellite imagery.

Although this paper focuses specifically on the application of ML to storm-mode classification, it also contributes to a larger body of work on implementing and evaluating ML techniques to improve weather forecasting in a manner that is not focused on postprocessing model output. With this paper, we also hope to raise awareness of the potential for similar ML models to be trained on many forms of data (such as in situ observations, remote sensing, and numerical weather prediction output), applied to a wide array of meteorological tasks, and used as components of larger systems, such as the Warn-On-Forecast system (WoF; Stensrud et al. 2009; Lawson et al. 2018; Skinner et al. 2018). This work builds on Gagne et al. (2009) in which we demonstrated that an automated storm classification system could use single decision trees to classify hand-labeled storms.

2. Data

We use two datasets to create predictors for storm mode: the Multiyear Reanalysis of Remotely Sensed Storms (MYRORSS; Ortega et al. 2012) and the Rapid Update Cycle numerical weather model (RUC; Benjamin et al. 2004, 2016). MYRORSS is used to identify storm cells and extract within-storm radar statistics, while the RUC is used to create a proximity sounding, representing the near-storm environment.

MYRORSS is an archive of quality-controlled, composited data from all NEXRAD radars in the continental United States (CONUS). Quality control and compositing is done by the Warning Decision Support

TABLE 1. Radar predictors. Each statistic is computed for each variable, using all grid cells inside the storm object, resulting in 120 predictors. MESH and VIL are in units of millimeters, azimuthal shear is in inverse seconds¹, reflectivity is in reflectivity decibels (dBZ); and echo top is in kilometers above sea level. SHI is unitless.

Variables	Spatial statistics
Low-level azimuthal shear [max from 0 to 2 km above ground level (AGL)]	Min
Midlevel azimuthal shear (max from 3 to 6 km AGL)	5th percentile
18-dBZ echo top	25th percentile
40-dBZ echo top	Median
Max estimated hail size (MESH)	75th percentile
−20°C reflectivity	95th percentile
−10°C reflectivity	Max
0°C reflectivity	Mean
Column-max (composite) reflectivity	Std dev
Lowest-altitude reflectivity	Skewness
Severe-hail index (SHI)	
Vertically integrated liquid (VIL)	

System with Integrated Information (WDSS-II; Lakshmanan et al. 2007), a software package for the analysis and visualization of radar data. MYRORSS includes 12 variables (Table 1, Figs. 3–5, described in more detail below) on a CONUS-wide grid at 5-min time steps. The velocity-derived fields (azimuthal shear) are on a 0.005° latitude–longitude grid, and the reflectivity-derived fields (all others) are on a 0.01° grid. The RUC is a nonhydrostatic mesoscale model with a 13- or 20-km grid covering the full CONUS, 50 vertical levels, and a 30-s time step. The model is run hourly, and output is available at 1-h time steps and 37 pressure levels equally spaced from 1000 to 100 hPa. The RUC (and its modern successor, the Rapid Refresh) have been used extensively to study thunderstorm environments, including in the Storm Prediction Center’s mesoanalysis (Storm Prediction Center 2019).

Storm cells (e.g., outlines in Figs. 2–5) are identified in MYRORSS by an algorithm called segmotion (Lakshmanan and Smith 2010), which is also part of WDSS-II. Identification is done independently at each 5-min time step, using the extended-watershed algorithm (Lakshmanan et al. 2009). This algorithm identifies local maxima in column-maximum reflectivity of at least 40 dBZ, then grows these point maxima into polygons with two minimum sizes: 40 km² (small scale) and 200 km² (large scale). The polygons are tracked over time, independently at each scale, using a variant of the *K*-means clustering algorithm. The two scales are then merged: for each large-scale polygon *L* containing exactly one small-scale polygon *S* (it may contain all or part of *S*), *S* is replaced with *L*. All other large-scale polygons are thrown out. Each remaining polygon is considered to be a storm object (one storm

cell at one time step), and most polygons contain only a single dominant updraft, making our definition of a storm cell consistent with S12. See Fig. 2 for an example of segmotion-detected storm objects at one time step. All segmotion settings described in this paragraph, including the scale merger, are used in the quasi-operational ProbSevere system (Cintineo et al. 2014, 2018).

We create four types of predictors for each storm object: radar predictors, shape predictors, storm motion, and sounding predictors. First, we compute 10 statistics for each of 12 radar variables (Table 1, Figs. 3–5), using only MYRORSS grid cells inside the storm object. This results in 120 radar predictors. Second, we compute nine shape predictors from the polygon defining the storm outline: area, perimeter, eccentricity, orientation, solidity, extent, mean absolute curvature, bending energy, and compactness. Eccentricity and orientation are based on the ellipse with the same second moments as the storm outline. Solidity is the number of grid cells in the storm/the number of grid cells in the convex hull; extent is the number of grid cells in the storm/the number of grid cells in the bounding box; mean absolute curvature is a mean over all vertices; bending energy is the sum of squared curvatures/perimeter; compactness is $\text{perimeter}^2/(4\pi \times \text{area})$. Solidity, extent, and bending energy can be viewed as measures of how irregular the shape is; solidity and extent decrease, while bending energy increases, as the shape becomes more irregular. Third, we extract storm motion from the segmotion file, which uses a finite-difference estimate. Storm motion is a vector and is decomposed in two ways—into *x* and *y* components and into magnitude, sine, and cosine—resulting in five predictors.

Fourth, sounding predictors are based on the proximity sounding, taken from the RUC. No temporal or spatial interpolation is done: we take the sounding from the¹ nearest grid cell to the storm center at the latest RUC analysis before $t - 30$ min, where t is the valid time of the storm object. The 30-min offset generally prevents convective contamination of the sounding. Nearest-neighbor interpolation preserves physical consistency among the sounding variables (vertical profiles of temperature, humidity, wind velocity, and geopotential height), which more complicated interpolation methods, such as linear or cubic, would not necessarily do. Sounding predictors are computed with the SHARPPy software package (Blumberg et al. 2017).

¹The RUC is run every hour, and we use a fallback plan to handle missing data. If the 13-km grid is available at the most recent hour, we use the 13-km grid. Otherwise, if the 20-km grid is available, we use the 20-km grid. Otherwise, we leave sounding predictors empty for the given storm object.

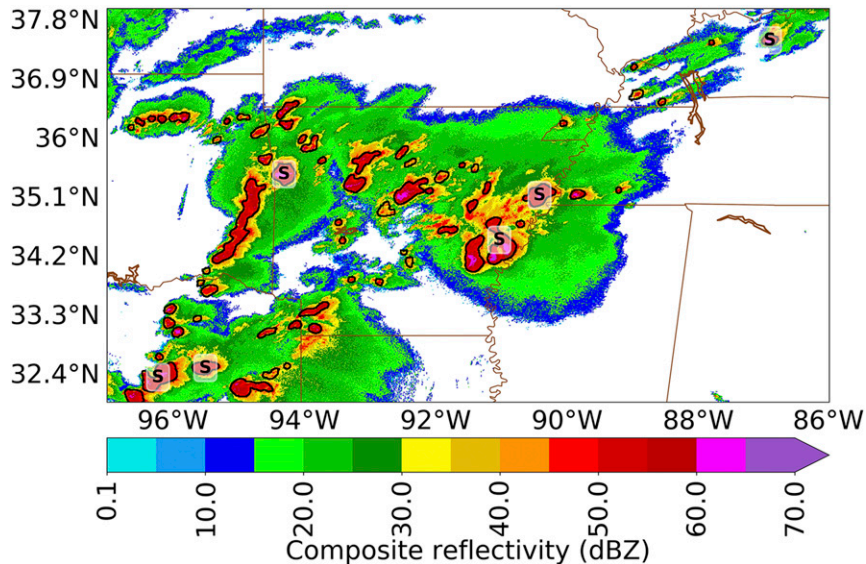


FIG. 2. Storm objects identified by the segmotion algorithm for one time step. Storm objects are outlined in black, and six of these storm objects have been labeled in the human dataset (all marked “S” for supercell).

Predictors that depend on actual storm motion are computed with the segmotion estimate, rather than sounding-inferred storm motion. Predictors such as storm-relative helicity are computed with right-mover/left-mover motion inferred from the sounding. Vectors output by SHARPPy are decomposed into five predictors each, as described in the above paragraph. The result is 207 sounding predictors, a complete list of which can be found in Table A1 of Lagerquist et al. (2017). In broad terms, sounding predictors include wind shear over different layers; mean wind velocity over different layers; mean storm-relative wind velocity over different layers; thermodynamic indices such as lapse rate, lifted index, and convective available potential energy over different layers; and other indices, such as the supercell composite parameter, significant-tornado parameter, wind-damage parameter, and others. Altogether, there are 341 predictors. Several example distributions for the ranges of the input predictors are shown in Fig. 6.

Our labels come from two datasets: S12 and H. Obermeier (2016, personal communication). As noted in section 1, S12 contains only significant-severe storms. Obermeier uses the same classification scheme as S12, but the Obermeier dataset contains both severe and nonsevere (mostly nonsevere) storms in the central United States² from April to December 2011. Because

² States include Texas, Oklahoma, Kansas, Nebraska, South Dakota, Louisiana, Arkansas, Missouri, Iowa, Minnesota, Illinois, and Wisconsin.

of the greater subjectivity in identifying linear hybrids and marginal supercells and their much smaller count of labeled storms, we relabel these storms as QLCS and supercell, respectively. This leaves three categories: supercell, QLCS, and disorganized.

To create target values for ML, we link these labels to storm objects created by segmotion. The “target value” is the correct answer (supercell, QLCS, or disorganized) provided to ML models during training. We link each label to the nearest storm object s^* within 20 km: the label must be inside the polygon or within 20 km of the nearest edge. If there is no storm object within 20 km, we throw the label out. We also assume that each label is valid for 10 min before and after its time stamp, so we link the label to all storm snapshots of s^* within 10 min. This 10-min offset allows the number of labeled storm objects to exceed the number of original labels. The final result is 77 402 objects labeled as supercells, 20 991 labeled as QLCS, and 24 994 labeled as disorganized (Fig. 1). Each storm object is considered to be a separate data point (example) for the ML models. The distribution shown in Fig. 1 does not reflect the true climatology, because most labels come from S12, which contains only significant-severe storms.

In this study we consider four subsets of the predictor variables: full (all 341), no azimuthal shear (“no-az-shear”; leaving 321 predictors), no sounding (leaving 129), and no azimuthal shear or sounding (“limited”; leaving 109). Azimuthal shear is omitted because it takes more time to process in MYRORSS than the reflectivity-based variables. Also, the quality of azimuthal shear decreases more quickly with distance from the nearest radar than does

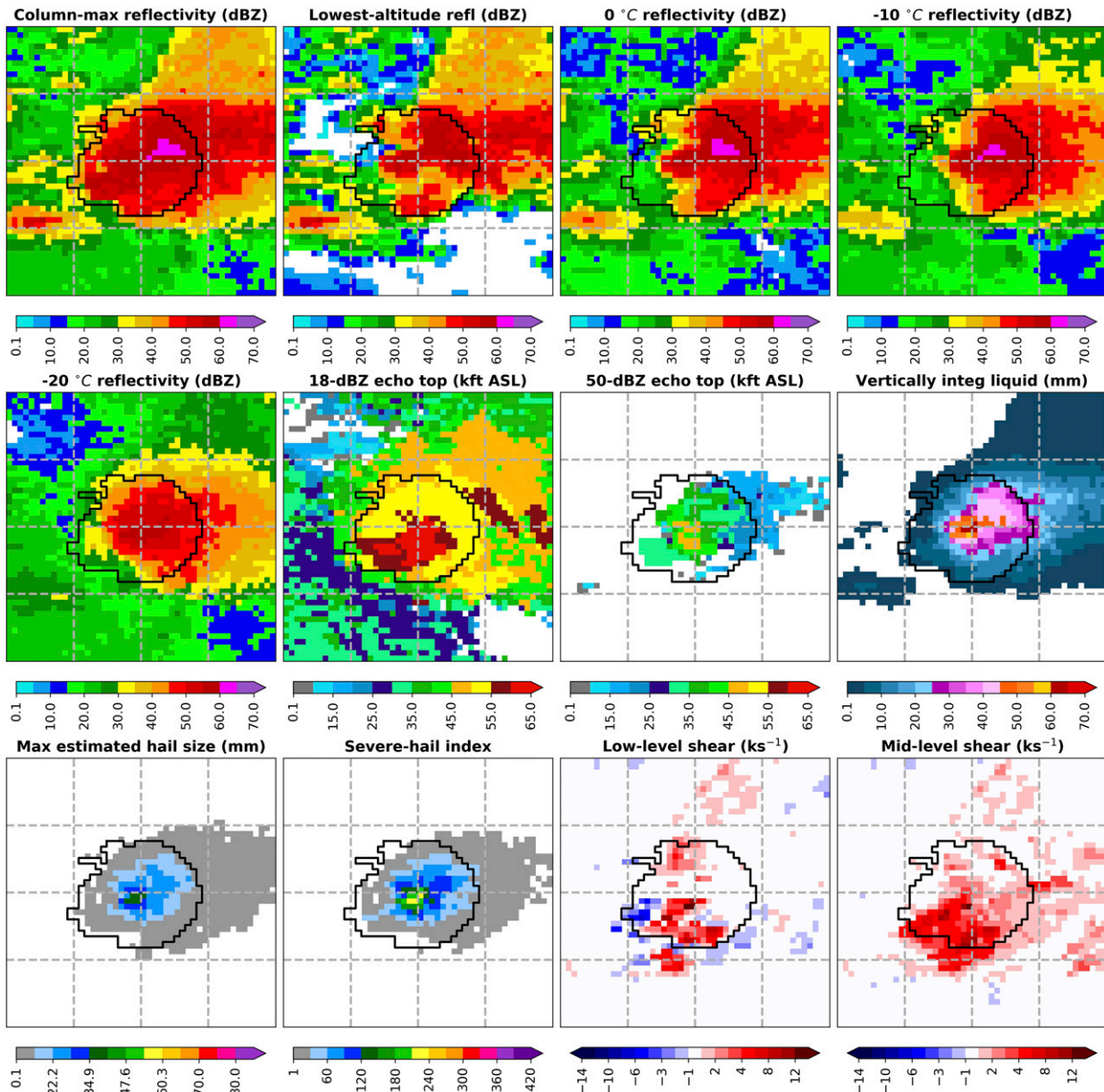


FIG. 3. Examples of the 12 MYRORSS variables for a randomly selected supercell.

reflectivity, so high-quality azimuthal shear is available for fewer storms. Sounding predictors are omitted because they take a long time to compute: once MYRORSS data have been processed, it takes milliseconds per storm object to compute radar predictors but ~ 1 s for sounding predictors. Thus, if we find that ML models perform equally well without one or more predictor types, an operational system could ignore them and perform just as well, with fewer computational resources required.

We hypothesize that removing any subset of predictors will be detrimental to model performance and

that removing azimuthal shear will be most impactful. Although high-quality azimuthal shear is not always available, it is generally a good indicator of supercells (high values in the mesocyclone), which dominate the dataset. If we were predicting future storm mode, sounding predictors might be more important, since they represent the near-storm environment, which largely determines the storm's evolution. However, our task is to classify storms at the present time, for which radar data should be more useful.

We apply five more transformations to the dataset, listed below.

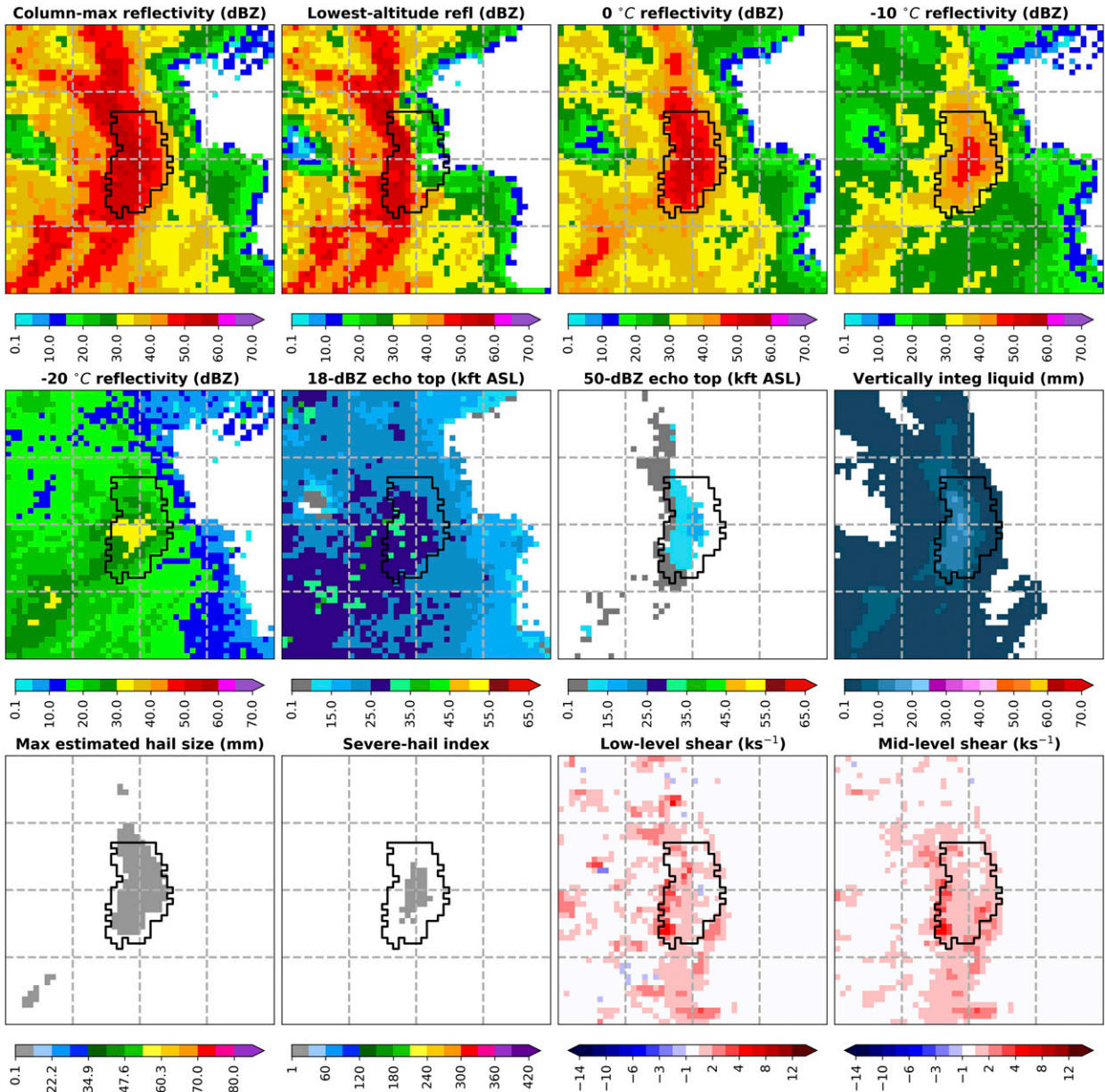


FIG. 4. Examples of the 12 MYRORSS variables for a randomly selected QLCS storm.

- 1) We remove examples with >100 missing predictors. This mostly removes examples with missing sounding data, which occur because the latest hourly RUC was unavailable.
- 2) We set any missing predictors left to -999 . This is necessary because we use the scikit-learn (Pedregosa et al. 2011) implementation for all ML models, which cannot handle missing values.
- 3) We create eight training/testing splits. In the k th split, the test set is year $2003 + k$ and the training set is all other years. Splitting by year eliminates

temporal autocorrelation between the training and testing sets, which ensures that they are statistically independent. Each ML model is trained eight times, once for each split, so that testing results can be reported for the entire dataset.

- 4) For each training/testing split, we normalize each predictor variable using Eq. (1). Variable x_{ij} is the unnormalized value of the j th predictor for the i th example; \tilde{x}_{ij} is the normalized value; \tilde{x}_j is the median of the j th predictor; $p_{75,j}$ is the 75th percentile; and $p_{25,j}$ is the 25th percentile:

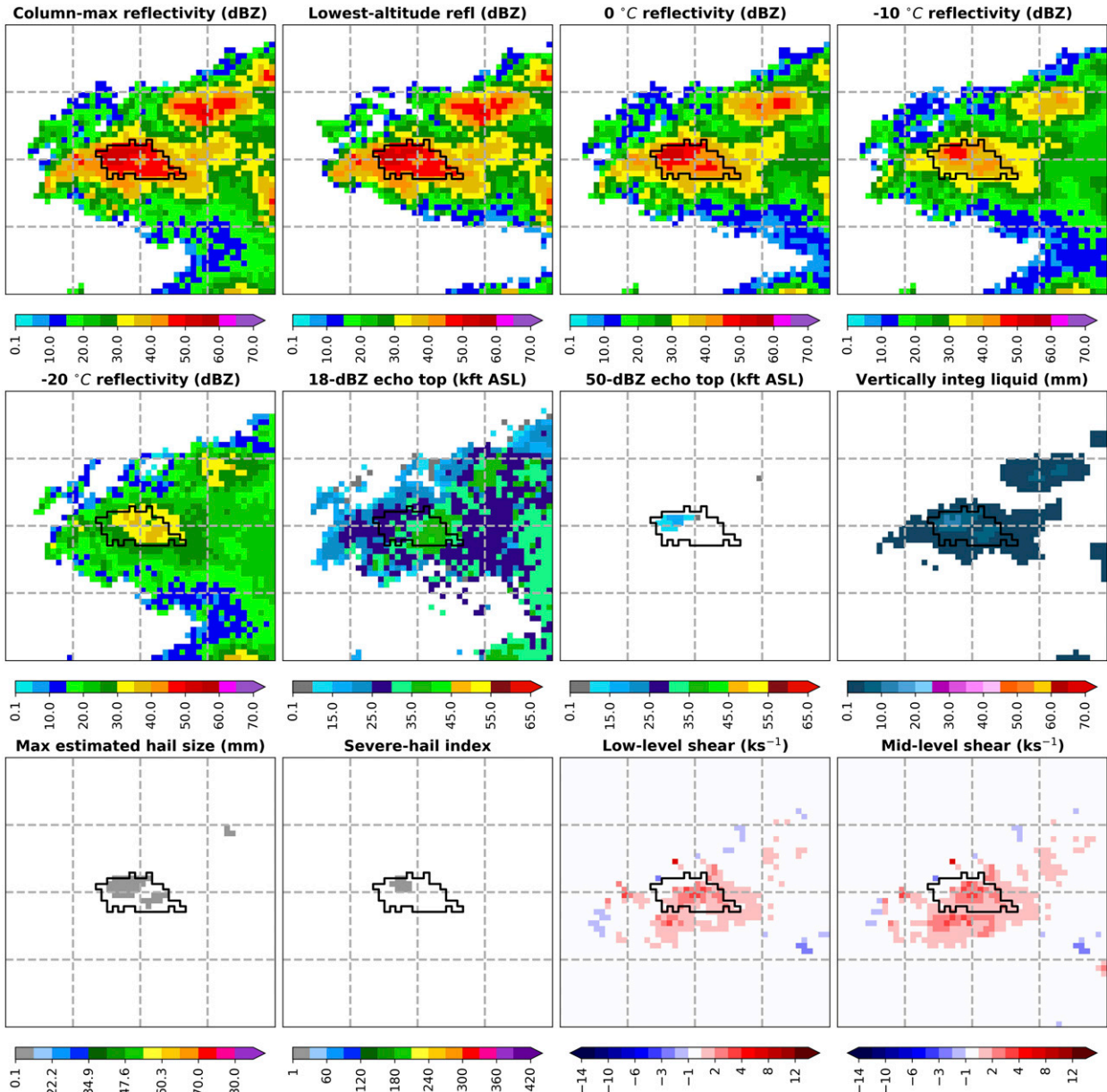


FIG. 5. Examples of the 12 MYRORSS variables for a randomly selected disorganized storm.

$$x'_{ij} = \frac{x_{ij} - \tilde{x}_j}{p_{75,j} - p_{25,j}}. \quad (1)$$

Normalization parameters (\tilde{x}_j , $p_{75,j}$, and $p_{25,j}$) are computed only on the training set and are used to normalize both the training and testing sets. This prevents the testing set from influencing the training procedure so that it may provide an independent assessment of the model's performance. We use Eq. (1) instead of the z score (the standard choice), because the z -score equation involves mean and standard

deviation, rather than median and interquartile range, which is $p_{75,j} - p_{25,j}$; the latter statistics are more resistant to outliers. This is especially important when the dataset uses -999 for missing values.

5) For each split, the training set is balanced to make the distribution 40% supercells, 30% QLCS, and 30% disorganized. When the distribution is heavily unbalanced, ML models tend to learn useful relationships only for the majority class and perform poorly on the minority classes. Balancing is done by both downsampling the majority class and upsampling the

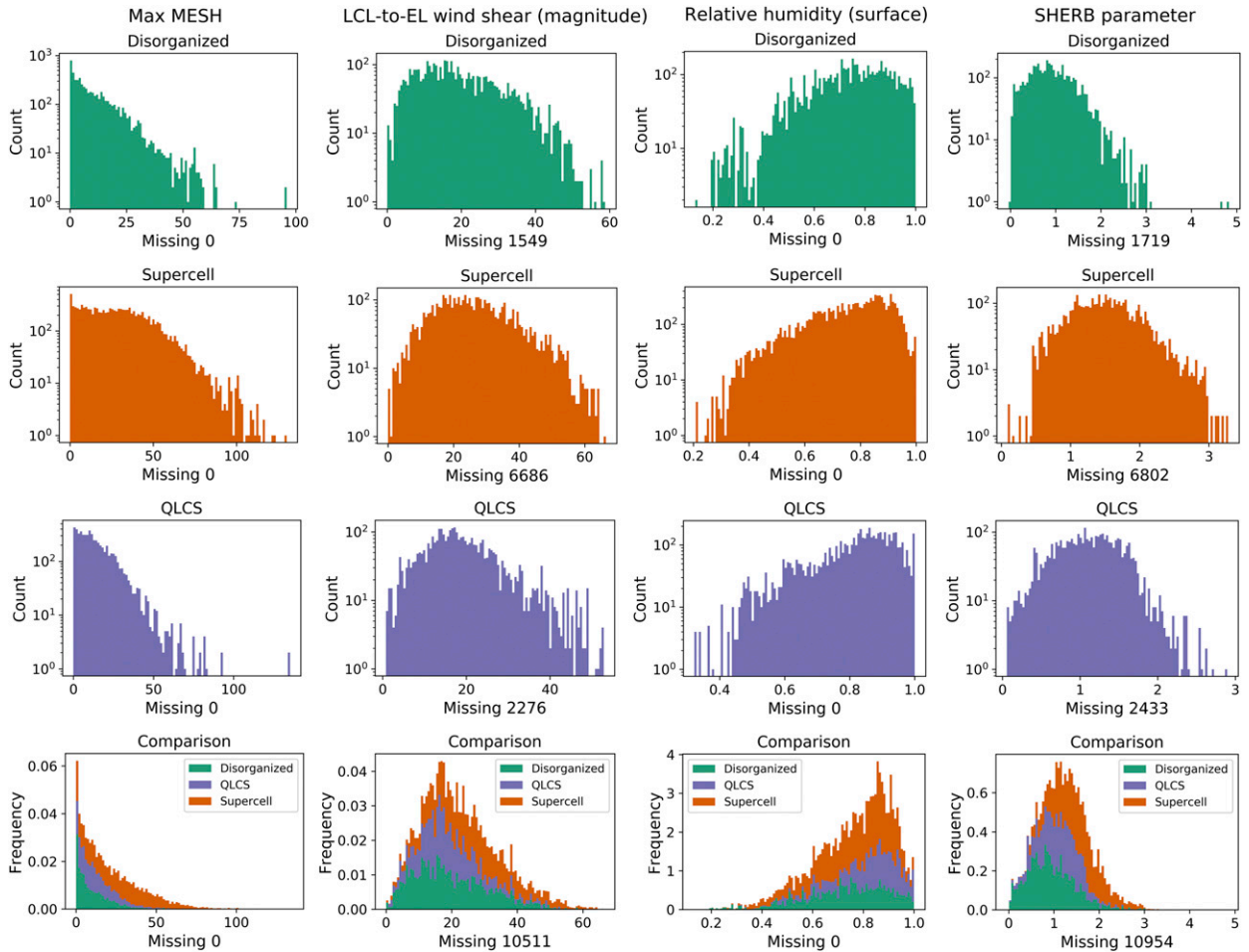


FIG. 6. Distributions of four predictor variables: (left) maximum MESH inside the storm, (left center) magnitude of wind shear from lifting condensation level to equilibrium level (LCL to EL), (right center) surface relative humidity, and (right) severe hazards in environments with reduced buoyancy (SHERB). For each predictor, a logarithmic frequency plot is shown for each class, as well as (bottom) a nonlogarithmic plot that compares the distributions across the three classes. The number of storm objects with missing values is included below each plot.

minority classes. Specifically, we randomly eliminate supercell examples, and randomly duplicate QLCS and disorganized examples, until the desired proportions are reached. Only the training set is balanced; the testing set comes from the actual data labels. This is necessary for demonstrating results that could work in the real world.

3. Machine-learning algorithms

This section briefly reviews the ML algorithms used. We use the scikit-learn (Pedregosa et al. 2011) implementation of all algorithms.

a. Logistic regression

Linear regression uses a weighted sum of the predictors, along with a bias weight, to predict a real number.

Logistic regression adapts linear regression to binary classification by applying a sigmoid function to the output. Equation (2) shows the resulting equation in terms of the predictors x_j and the learned weights β_j . Note that z is simply linear regression:

$$y = \frac{\exp(z)}{1 + \exp(z)}, \text{ where } z = \beta_0 + \sum_{j=1}^M \beta_j x_j. \quad (2)$$

Logistic regression can be adapted for nonbinary (e.g., three class) classification in several ways. We use the scikit-learn method SGDClassifier, which trains a different binary classifier [version of Eq. (2)] for each class.

During training, the weights are adjusted to minimize the cross-entropy between the predictions and true values under two additional regularizations. The L_1 (“lasso”) penalty encourages the model to produce

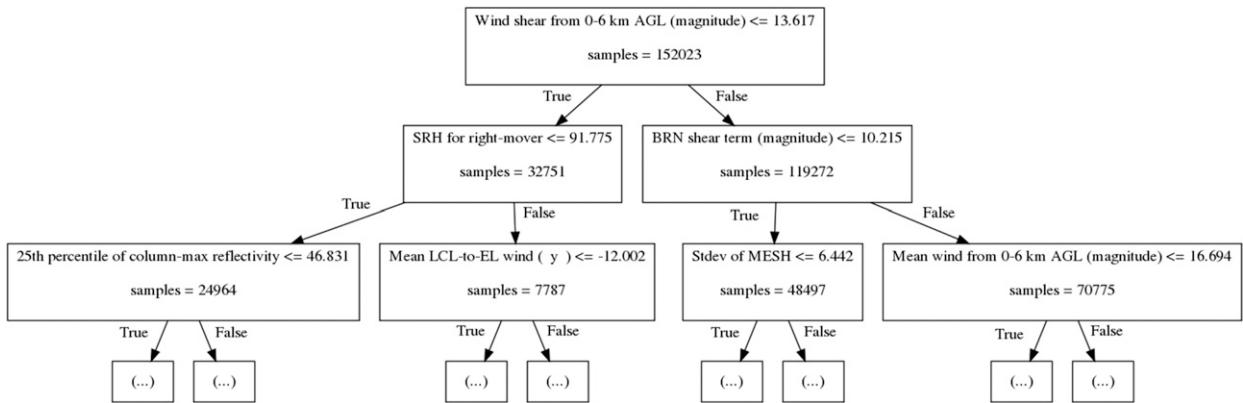


FIG. 7. Diagram of the first few decisions in a single decision tree. As more questions are asked, the node contains predominantly more examples of a single class. At the bottom leaf nodes, a prediction is made using the fraction of each class present at the leaf. This tree is one of the trees in the GBF that was trained over the full predictor set and therefore contains seven total layers.

weights of exactly zero by penalizing the sum of absolute values of the weights (Tibshirani 1996), and the L_2 (“ridge”) penalty encourages the remaining weights to be very small by penalizing the sum of squared weights (Hoerl and Kennard 1988). Equation (3) shows the resulting loss function (penalized cross-entropy), where the first sum is the cross-entropy between the predictions \hat{y}_i and true values y_i ; the second sum is the L_1 penalty; the final sum is the L_2 penalty; and λ_1 and λ_2 are the strengths of the respective penalties, which we choose using cross validation:

$$P = -\frac{1}{N} \sum_{i=1}^N [y_i \log_2(\hat{y}_i) + (1 - y_i) \log_2(1 - \hat{y}_i)] + \lambda_1 \sum_{j=1}^M |\beta_j| + \lambda_2 \sum_{j=1}^M \beta_j^2. \quad (3)$$

Applying both L_1 and L_2 penalties is called elastic-net regularization (Zou and Hastie 2005). Linear regression and variants thereof have been used in meteorology for decades (e.g., Kohler 1949; Malone 1955). Recent successes—the first two of which use elastic-net regularization—include predicting solar radiation (Aggarwal and Saini 2014), pollutant concentration (Suleiman et al. 2016), and convective initiation (Mecikalski et al. 2015).

b. Random and gradient-boosted forests

A decision tree consists of both branch nodes and leaf nodes (Quinlan 1986). Each branch node poses a yes-or-no question for one predictor (e.g., “is the reflectivity at least 65 dBZ?”), deciding whether the example is sent down the right or left branch. Leaf node n predicts the probability of each class, based on the examples in the training set that reached n . An example of the first few

questions asked by a decision tree is shown in Fig. 7. See Fig. 1 of McGovern et al. (2017) for an illustration of a complete tree.

The main disadvantage of decision trees is that they learn precise thresholds, which easily overfit the training data. This problem can be mitigated by ensembling many trees into a random forest or gradient-boosted forest (GBF). In a random forest (Breiman 2001), each tree is trained with a bootstrapped replicate of the training examples and each branch node is allowed to choose from only a small subset of the predictors. A bootstrapped replicate of N examples still contains N examples, but they are resampled with replacement from the original dataset, yielding approximately 63.2% of the unique examples on average (Efron 1979). Bootstrapping and predictor-subsetting ensure diversity among the trees, so while the individuals overfit, their biases often cancel out when ensembled. To make predictions from a random forest, the individual trees’ predictions are averaged.

In a GBF (Friedman 2002) the first tree is trained as usual. However, all other trees are iteratively trained to predict the error of the previous trees, emphasizing difficult examples more heavily. This encourages each tree to learn from the mistakes of previous trees. Also, the target variable for each tree after the first is an intricate function of the predictors and label, rather than the label itself, which allows GBFs to learn more complicated functions than random forests. In practice, GBFs generally make better predictions than random forests. However, GBFs are more computationally expensive, because the trees must be trained in series, whereas those in a random forest can be trained in parallel.

Random forests and GBFs have been applied successfully to predict convectively induced turbulence

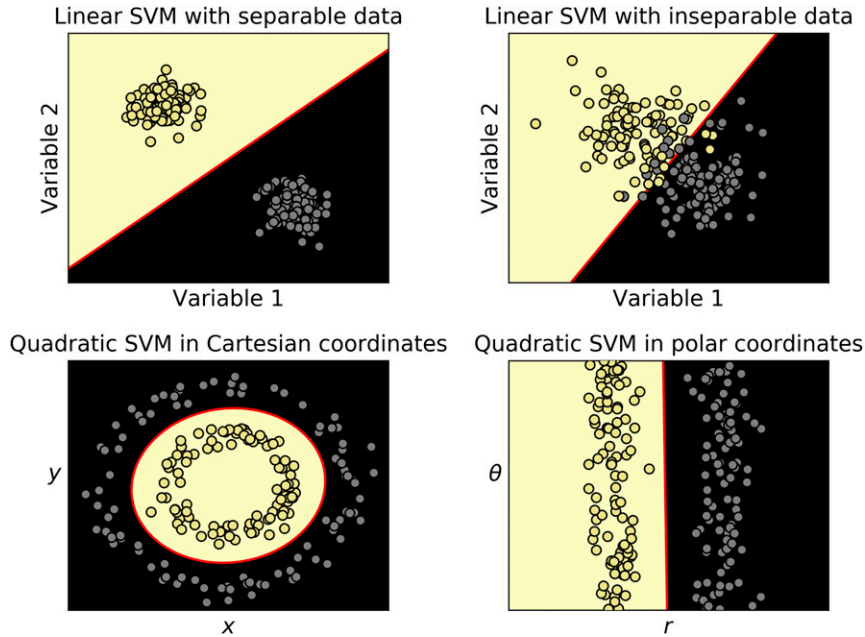


FIG. 8. SVMs used to separate synthetic data into two classes; the predictors are (top) variables 1 and 2, (bottom left) x and y , and (bottom right) r and θ . True classes are represented by yellow and gray dots; predicted classes are represented by beige and black regions. In the top-left diagram, a linear SVM is used to classify linearly separable data; in the top-right diagram, a linear SVM is used to classify non-linearly separable data; in the bottom-left diagram, a quadratic SVM is used to classify non-linearly separable data; the bottom-right diagram is the same as the bottom-left one but is in the space implicitly defined by the SVM’s quadratic kernel.

(Williams 2014), tornadogenesis (McGovern et al. 2014), solar radiation (McGovern et al. 2015), and damaging straight-line wind (Lagerquist et al. 2017); and to identify features such as drylines (Clark et al. 2015); and meso-scale convective systems (Haberlie and Ashley 2018).

c. Support-vector machines

Linear support-vector machines (SVM) were first developed by Vapnik (1963) for binary classification. Linear SVMs work by finding a hyperplane in the predictor space that best separates the two classes (i.e., most examples in class A are on one side of the hyperplane, while most examples in class B are on the other side). Schematics for a 2D predictor space are shown in Fig. 8. SVMs cannot be so easily visualized in our predictor space, which is anywhere from 109-D (the limited predictor set) to 341-D (full predictor set). The SVM also learns to maximize the “margin”—average Euclidean distance between the hyperplane and a correctly classified example. This is shown most clearly in the top left of Fig. 8: the line could be moved toward the bottom-right or top-left and still obtain 100% accuracy on the data shown, but maximizing the margin generally makes the SVM a better predictor for new data (which may fall outside of the two clusters shown).

Since most real-world data are not linearly separable, nonlinear kernels (Cortes and Vapnik 1995; Vapnik 1995) are often used to implicitly transform the predictor space. The linear kernel (which may be considered as “no kernel”) is defined in Eq. (4). Both \mathbf{x} and \mathbf{w} are predictor vectors for two examples, both of length M , where M is the number of predictors; $\mathbf{x} \cdot \mathbf{w}$ is the dot product ($x_1w_1 + x_2w_2 + \dots + x_Mw_M$); and c is a regularization term. The c is a hyperparameter, which encourages the model to overfit when too small and underfit when too large:

$$K_{\text{linear}} = \mathbf{x} \cdot \mathbf{w} + c. \tag{4}$$

For nonlinear SVMs in this study, we use the Gaussian kernel [Eq. (5)]. Here $\|\mathbf{x} - \mathbf{w}\|$ is the magnitude (Euclidean norm) of the difference between the two vectors, and σ (another hyperparameter) is the decay rate. As with c , small values of σ lead to overfitting and large values of σ lead to underfitting:

$$K_{\text{Gaussian}} = \exp \left[-\frac{1}{2} \left(\frac{\|\mathbf{x} - \mathbf{w}\|}{\sigma} \right)^2 \right] + c. \tag{5}$$

SVMs can be adapted for nonbinary classification in the same way as logistic regression (section 3a): by training one model for each class k , which discriminates

between k and all other classes (the one-vs-all approach). Although less popular in meteorology than the other ML algorithms discussed, SVMs have been used successfully to predict temperature (Radhika and Shashi 2009) and tornadoes (Trafalis et al. 2003; Adrianto et al. 2009).

4. Model interpretation

We use multiple interpretation methods to understand the relationships learned by ML models. For decision trees and forests, the standard interpretation method is impurity importance (Louppe et al. 2013). However, this method works only for trees and forests, and it overstates the importance of predictors that appear earlier in the tree (closer to the root node), as explained in McGovern et al. (2019). Instead, we use model-agnostic interpretation methods, which can be applied to any ML model. They are described briefly in the following sections, and more detailed descriptions are found in McGovern et al. (2019).

a. Permutation importance

Permutation importance measures the importance of each predictor x_j by how much the model error changes when statistical correlations between x_j and the target variable are broken. The model is first trained on nonpermuted data and then importance can be measured on either the training or testing set. Shuffling the values for a single predictor across the various observations breaks the statistical correlation between x_j and the target variable, so if x_j is important, error should increase. If error does not increase, this is a sign that, at least for the given model, x_j is unimportant. There are two versions of permutation importance: single-pass (Breiman 2001) and multipass (Lakshmanan et al. 2015), both implemented with parallelization in Jergensen (2019). In the single-pass version, each predictor is shuffled once and importance is calculated. In the multipass version, the most important variable remains shuffled while additional variables are shuffled. The multipass and single-pass algorithms often give different answers, especially when there is a linear correlation among the predictors. This issue is discussed further in McGovern et al. (2019).

b. Sequential selection

Sequential forward selection (SFS; Webb 2003, their section 9.2.3) is another model-agnostic approach to ranking predictor importance. The algorithm is outlined below, and M is again the total number of predictors.

- 1) Train M models with one predictor each. Compute the error of each one-predictor model on the testing set. Keep the model with the lowest error and call the newly added predictor x_1^* .
- 2) For each of the $M - 1$ remaining predictors x_j , train a model with x_1^* and x_j . Compute the error of each two-predictor model on the testing set. Keep the model with the lowest error and call the newly added predictor x_2^* .

This process continues until a stopping criterion is met. There are two key differences between permutation importance and SFS. First, to evaluate model error without x_j , SFS removes x_j from the model entirely, whereas permutation importance shuffles x_j randomly. Second, SFS retrains the model for each predictor set, whereas permutation importance uses a pretrained model. As a result, whereas permutation importance indicates the importance of a predictor to a particular model realization, SFS indicates the importance of the predictor to the model architecture. Both permutation importance and SFS can be computationally expensive, because the first computes model error $0.5M(M - 1)$ times and the second retrains the model up to $0.5M(M - 1)$ times. Retraining the model generally takes much longer, which makes SFS much more expensive. However, in practice the stopping criterion is usually reached early (after only a small minority of predictors has been added), which makes SFS tractable.

c. Partial-dependence plots

The above methods succinctly quantify the importance of each predictor, but they do not indicate how it is important. This problem is partially addressed by partial-dependence plots (PDP; Friedman 2001), which visualize the average prediction as a function of each predictor x_j . Specifically, x_j is fixed at a given value for all examples (leaving the other predictors untouched); the fixed dataset is passed through the model; and the resulting predictions are averaged over all examples. This process is repeated for many values of x_j , yielding a curve. Parts of the curve with a nonzero slope indicate where the model is sensitive to x_j , and the sign (positive or negative) of the slope indicates the direction of the relationship.

5. Results

This section summarizes the predictive performance and interpretation of all five ML algorithms: logistic regression with elastic-net regularization, random forest, GBF, and SVM with both linear and nonlinear kernels. Performance results are based on all eight years, 2004–11, but using only the testing year from each training/testing split (section 2). Interpretation results are based on 2010, from the split where 2010 is the testing year. Full interpretation methods could not be

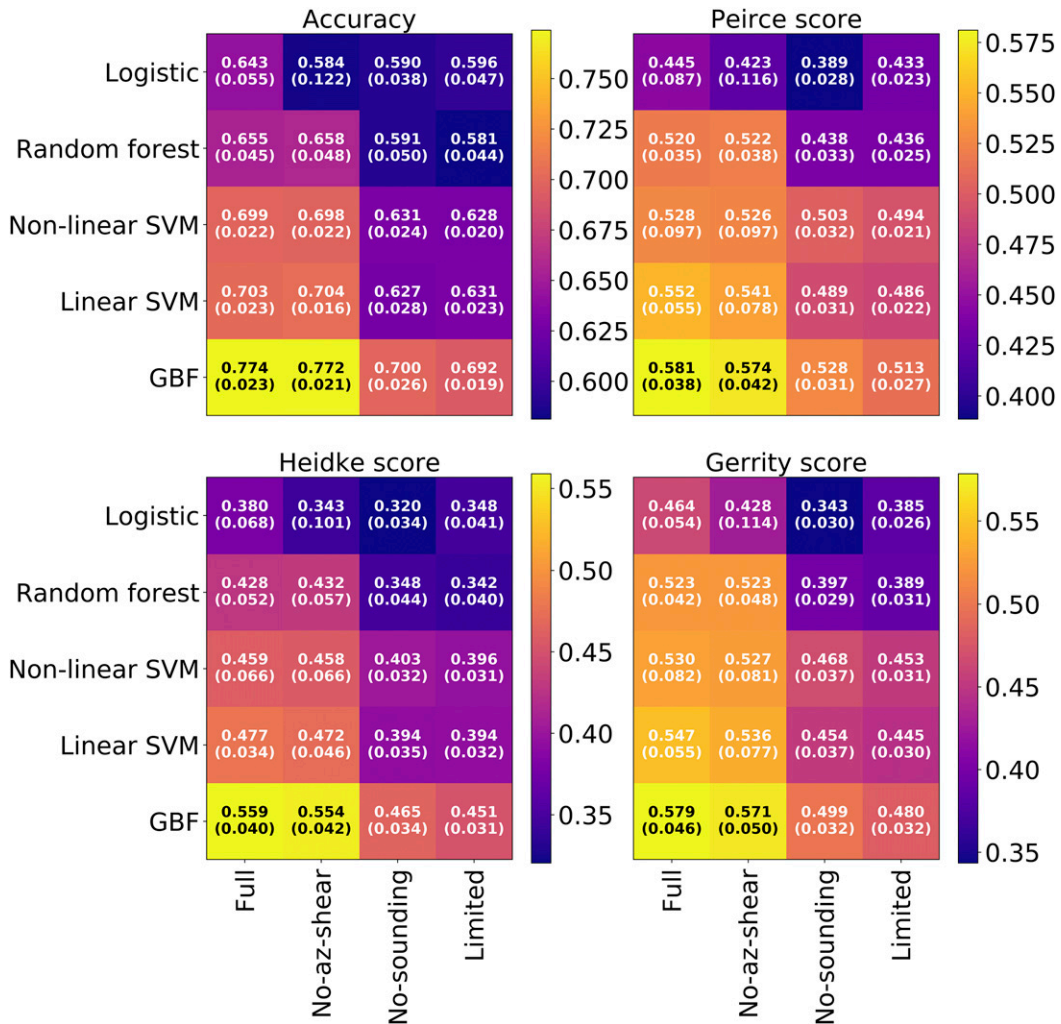


FIG. 9. Model scores on testing data. The numbers without parentheses are the mean, and those inside parentheses are the standard deviation over all eight training/testing splits. Lighter color indicates higher skill, as depicted by the color bars next to each plot.

run on the full eight years, because the methods are too computationally expensive. All models are implemented in version 0.20 of scikit-learn (Pedregosa et al. 2011). Implementation details, including hyperparameters (model settings such as penalty weights, number of trees in the forest, etc.), are found in the appendix.

a. Evaluation scores

Our main skill score is the Peirce score (Peirce 1884), which measures the improvement in accuracy over climatological guessing. In climatological guessing, if class k appears with frequency f in the training set, the predicted probability of class k is f for every example. We also compute accuracy, the Heidke score (Heidke 1926), which measures the improvement in accuracy over random guessing, and the Gerrity score (Gerrity 1992), which is similar to the Peirce and Heidke scores but

rewards correct predictions of the minority classes (QLCS and disorganized) more than the majority class (supercells). Accuracy ranges over $[0, 1]$; Heidke score ranges over $(-\infty, 1]$; and the Peirce and Gerrity scores range over $[-1, 1]$. Higher is better for all scores, and, for all but accuracy, positive values indicate skill (an improvement over the baseline).

b. Model performance

Figure 9 summarizes the performance of each model on each predictor set. On all predictor sets the model with the best performance (as measured by the Peirce score) is the GBF, followed by the linear SVM, non-linear SVM, random forest, and logistic regression. The performance of the GBF and linear SVM (second-best model) often differs by more than the standard deviation over the eight training/testing splits, but in general

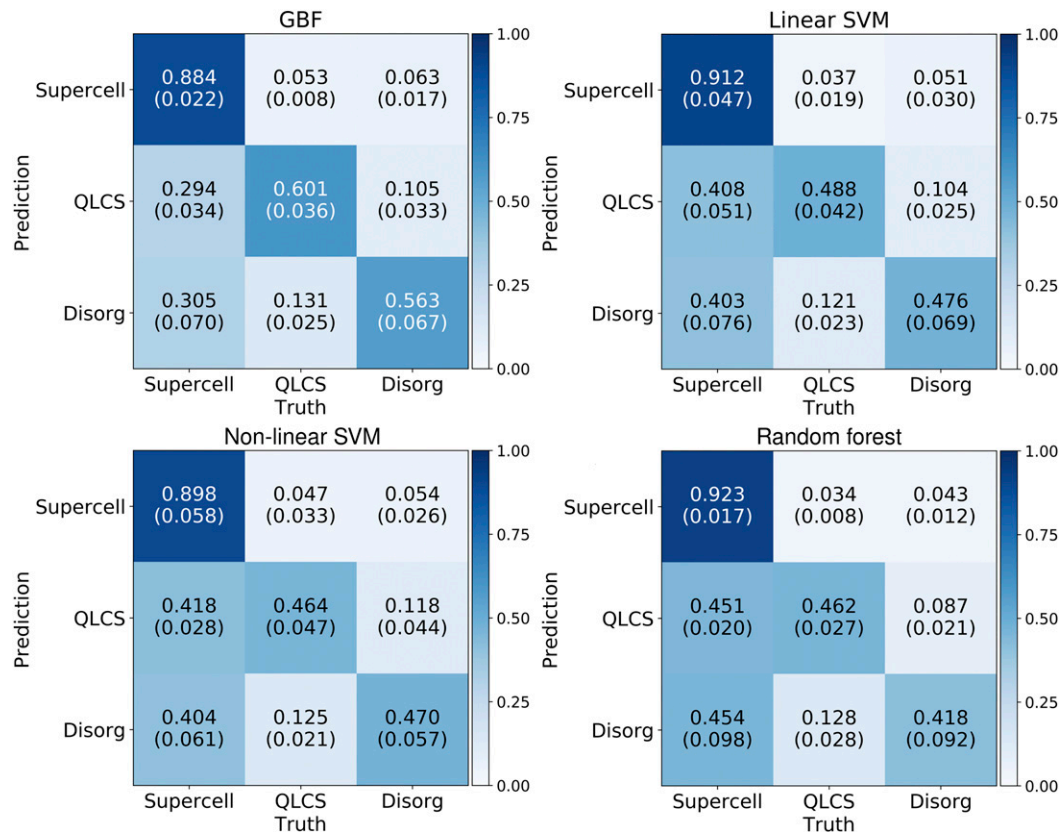


FIG. 10. Row-normalized contingency tables for the four best models on the full predictor set. “Row normalized” means that the sum across each row in each table is 1.0. Thus, the number at row i and column j is the conditional probability that the j th label is observed, given that the i th label is predicted. For example, if the number at row “QLCS” and column “Disorg” is 0.105, 10.5% of predicted QLCS storms are actually disorganized. The numbers in parentheses are the standard deviation over all eight training/testing splits. Darker color indicates a higher fraction of predictions for that true label.

no other model pairs differ strongly. The superiority of GBF over the SVMs suggests that the data are too complex for the classes (supercell, QLCS, and disorganized) to be separated by hyperplanes, even after transforming the predictor space with a nonlinear kernel. The superiority of GBF over the random forest suggests that sequential training, where the k th tree focuses on the most difficult examples for the first $k - 1$ trees, is important for this dataset.

For almost all models, the best performance is achieved with the full predictor set, followed by no-az-shear (missing 20 predictors), then no-sounding (missing 212 predictors), then limited (missing 232 predictors). This supports our hypothesis that model performance would decline whenever predictors are removed. Differences between the top two predictor sets (full and no-az-shear), as well as the differences between the bottom two sets (no-sounding and limited), are small. This suggests that azimuthal shear has little impact on model performance. However, differences between the

full and no-sounding sets are much larger. This suggests that the sounding yields a better predictor set than azimuthal shear. However, this should not be taken to mean that azimuthal shear is generally unimportant. One confounding factor is that the sounding includes 212 predictors, whereas azimuthal-shear statistics include only 20. The correct conclusion is that for this particular prediction task azimuthal shear can be safely ignored, which obviates the need for expensive data-processing (section 2).

Contingency tables for the top four algorithms (all but logistic regression), for both the full and limited predictor sets, are shown in Figs. 10–13. For both predictor sets, the GBF has the lowest success ratio³ for supercells but the highest success ratio for QLCS and disorganized (Figs. 10 and 12). In other words, a key advantage of the GBF is that its predictions of the minority classes are

³The “success ratio” for class k is the fraction of predictions of class k that are correct.

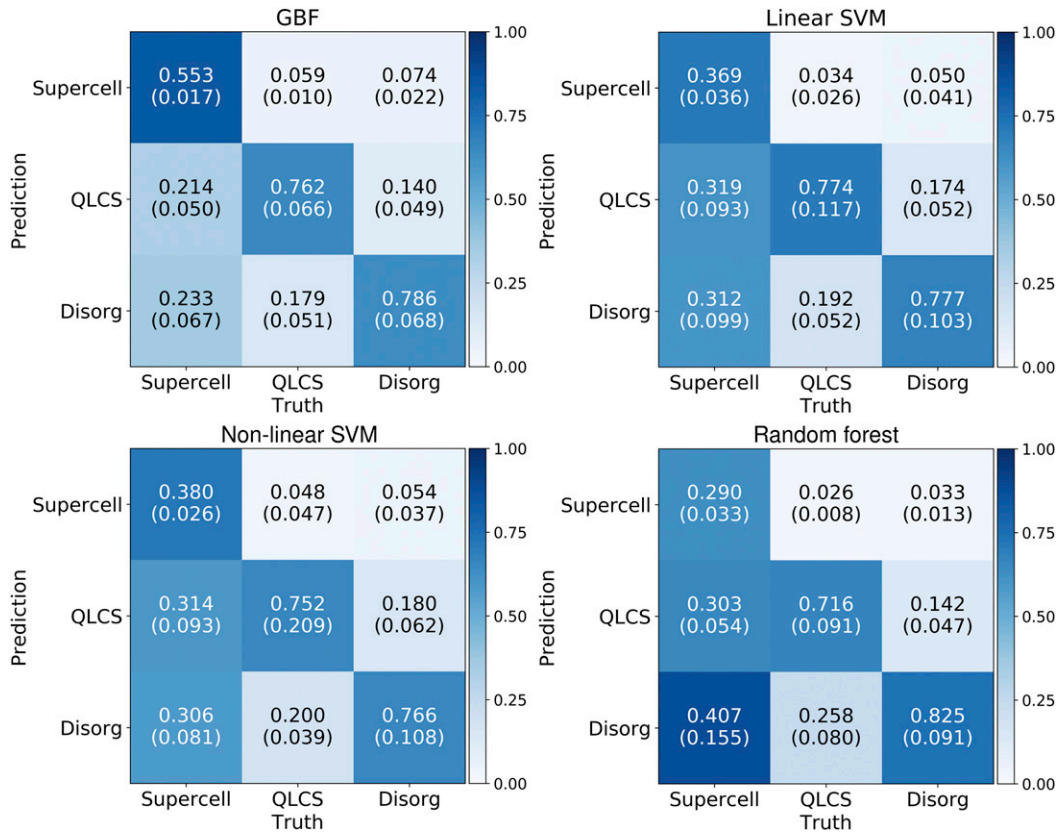


FIG. 11. As in Fig. 10, but column normalized. “Column normalized” means that the sum over each column in each table is 1.0. Thus, the number at row i and column j is the conditional probability that the i th label is predicted, given that the j th label is observed. For example, if the number at row “QLCS” and column “Disorg” is 0.140, 14.0% of disorganized storms are predicted to be QLCS.

more often correct. Also, for both predictor sets, the GBF has the highest probability of detection⁴ for supercells (Figs. 11 and 13).

c. Model interpretation

Results of multipass permutation importance and SFS are shown in Figs. 14–16. Ideally these figures would show the top four models, but the nonlinear SVM was too computationally expensive for model interpretation. The nonlinear SVM takes much longer to generate predictions, because this process involves a nonlinear transformation of the predictors, which are 341-dimensional vectors. The loss function for permutation importance and SFS is the negative Peirce score.⁵ We show only the top 10 predictors, for ease of viewing.

⁴The “probability of detection” for class k is the fraction of actual occurrences of class k that are correct.

⁵We multiply the Peirce score by -1 , since the loss function must be negatively oriented (where lower is better) and Peirce score is positively oriented.

According to permutation importance (Fig. 14), permuting one predictor almost never causes a significant decrease in performance. This makes sense, as there are 341 predictors and many are likely correlated. The two predictors causing a significant decrease, both for the linear SVM, are the microburst composite parameter (MCP; Entremont et al. 2018) and sine of effective-layer shear (ELS). As MCP increases, QLCS frequency increases and disorganized frequency decreases (Fig. 15a), possibly because QLCSs are intrinsically associated with downbursts (section 1), of which microbursts are a subcategory. As the sine⁶ of ELS increases, disorganized frequency increases while QLCS frequency decreases (Fig. 15b). This suggests that, when effective-layer shear is more southerly, disorganized storms are more likely and QLCSs are less likely. However, this is the weakest relationship of the partial dependency plots in Fig. 15 and therefore should be given less credence.

⁶The sine of a vector is $(y - \text{component})/\text{magnitude}$, or the fraction of its magnitude that points northward.

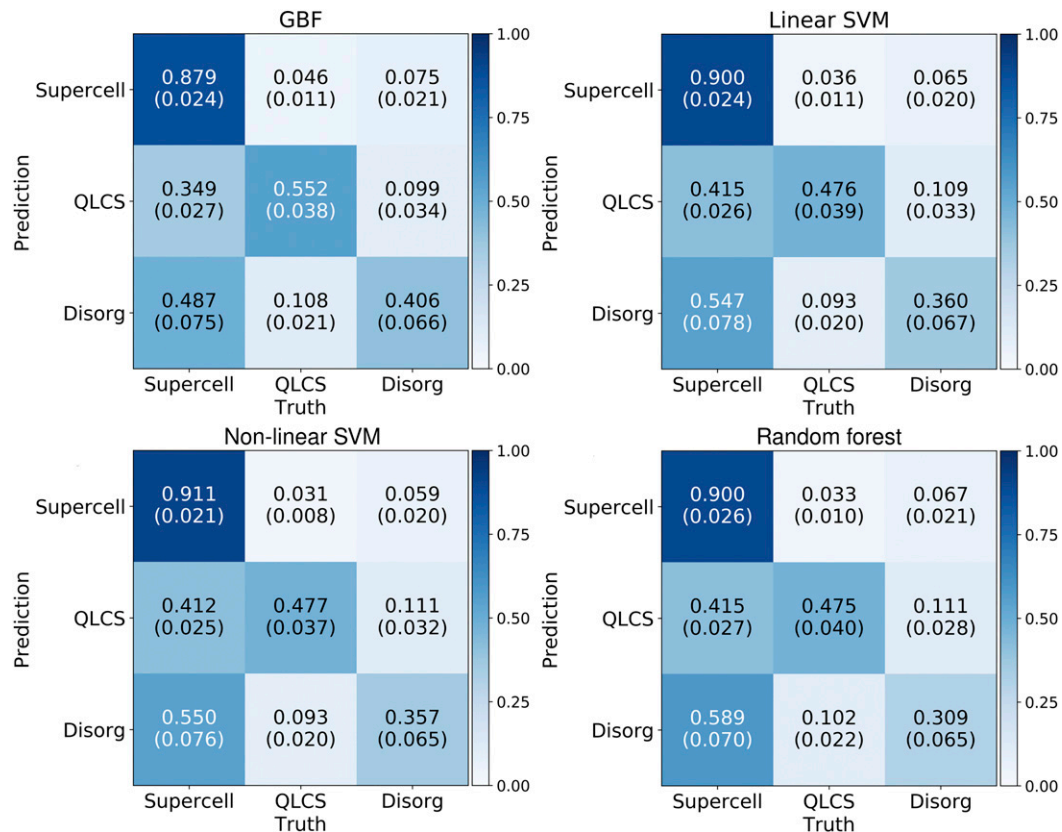


FIG. 12. As in Fig. 10, but for the limited predictor set.

More generally, permutation importance suggests that shape predictors are the most important type. The top 10 predictors for each model include 6 or 7 shape predictors, whereas shape predictors make up only 9 of 341 in the dataset. Supercell frequency increases strongly with storm age, while QLCS frequency decreases strongly (Fig. 15c), likely because supercells tend to be longer-lived while individual cells in a QLCS do not. QLCS frequency increases strongly with storm area (Fig. 15d), possibly because the tracking algorithm struggles with QLCSs and often includes multiple updraft cores in the same “storm cell.” Supercell frequency decreases with eccentricity, while QLCS frequency increases (Fig. 15e), possibly because supercells tend to be more circular and those in a QLCS tend to be more elongated. Last, QLCS frequency increases with compactness (Fig. 15f), defined as the object area/area of a circle with the same perimeter. More compact objects tend to be simpler, while less compact objects often have large intrusions or protrusions, such as a hook echo in a supercell.

SFS (Fig. 16) tells a different story. Results for the linear SVM have very wide confidence intervals (likely because the model is unstable with few predictors), so

this discussion will focus on the GBF and random forest. Common predictors between the two are skewness of azimuthal shear, components of environmental wind shear, and components of layer-averaged storm-relative wind (SRW). Partial-dependence plots for the GBF and random forest for these predictors are not shown because the dependence of these predictors across the domain was extremely flat (much flatter than Fig. 15b), suggesting that there is almost no dependence on these predictors. This is possibly because, within a GBF or random forest, the same predictor shows up in many different contexts (in different trees, applied to different subsets of the data, with different thresholds). The same predictor can have a different effect in each context, and these effects often nearly cancel out. As such, the increases or decreases in the prediction frequencies across the domain are insignificant and the PDPs yield inconclusive results. In light of this, we do not offer an interpretation of the GBF and random forest dependence on these predictors, as we wish to avoid overzealous interpretation of the PDP and predictor importance results.

SFS also indicates that echo top and hail statistics (MESH and SHI) are more important for the random

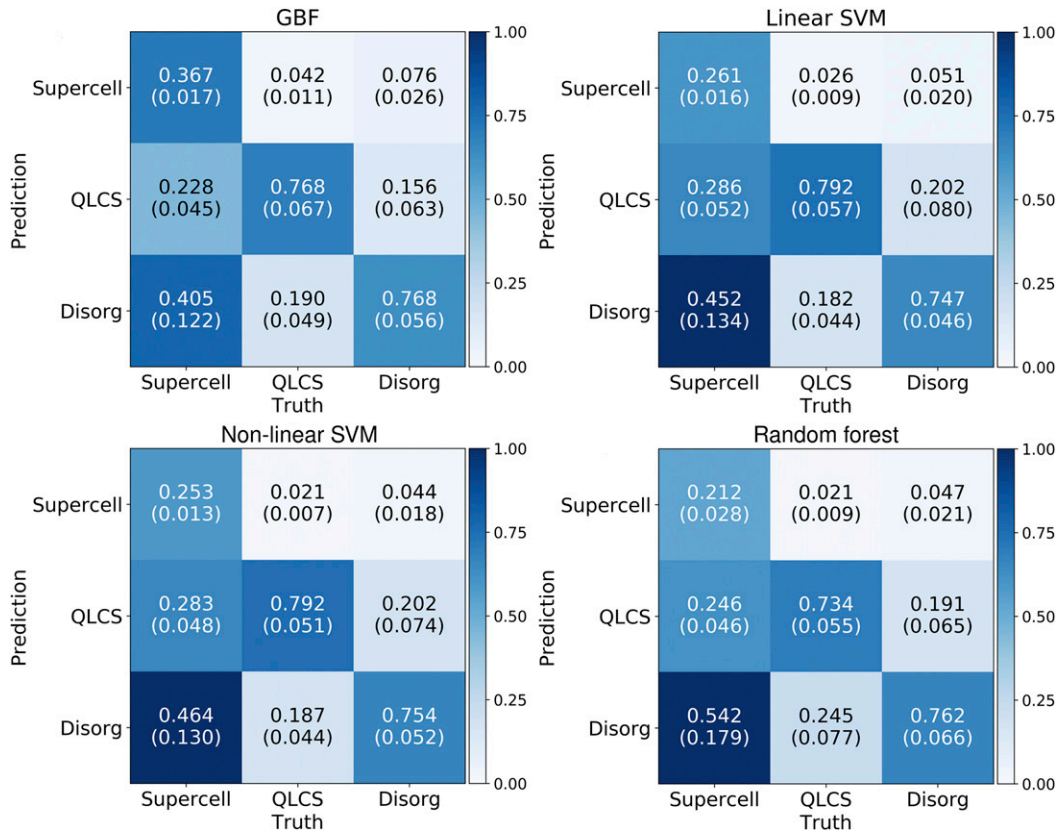


FIG. 13. As in Fig. 11, but for the limited predictor set.

forest, while column-maximum reflectivity and compactness (a shape parameter) are more important for the GBF. Since the GBF widely outperforms the random forest, one might conclude that variables in the GBF set but not in the random-forest set are generally better predictors of storm mode. However, we caution against this conclusion, since (i) according to permutation importance, the top 10 predictors for the GBF and random forest are much more similar; (ii) PDPs for both models are inconclusive and do not clearly show *how* the important predictors are important; and (iii) none of the interpretation methods considered here allow for direct numerical comparison between different models. It is plausible that another model could outperform the GBF with a completely different set of top predictors. This last point underscores the need, when interpreting ML models, to look for general trends across models and interpretation methods.

6. Summary and conclusions

We used several machine-learning algorithms to classify thunderstorms into three convective modes:

supercell, part of a QLCS, and disorganized. Our predictors included composited radar data from MYRORSS, a proximity sounding from the RUC model, and storm motion and shape derived from segmotion, a quasi-operational storm-tracking algorithm. We compared the five ML algorithms on four predictor sets, and the best configuration was GBF with the full predictor set, yielding an accuracy of 0.77 ± 0.02 and Peirce score of 0.58 ± 0.04 .

We also employed three ML-interpretation methods. First, according to permutation importance, shape predictors are vastly more important than radar or sounding predictors. PDPs for the linear SVM suggest that supercell frequency increases with storm age and decreases with storm area, eccentricity, and compactness. However, the relationships with storm area are likely an artifact of the tracking algorithm merging several QLCS updraft cores into one object. Second, according to SFS, environmental wind shear and storm-relative wind are generally important predictors, while echo top, column-maximum reflectivity, and hail statistics (MESH and SHI) are each important for only a subset of ML models. SFS results vary much more across models than permutation results,

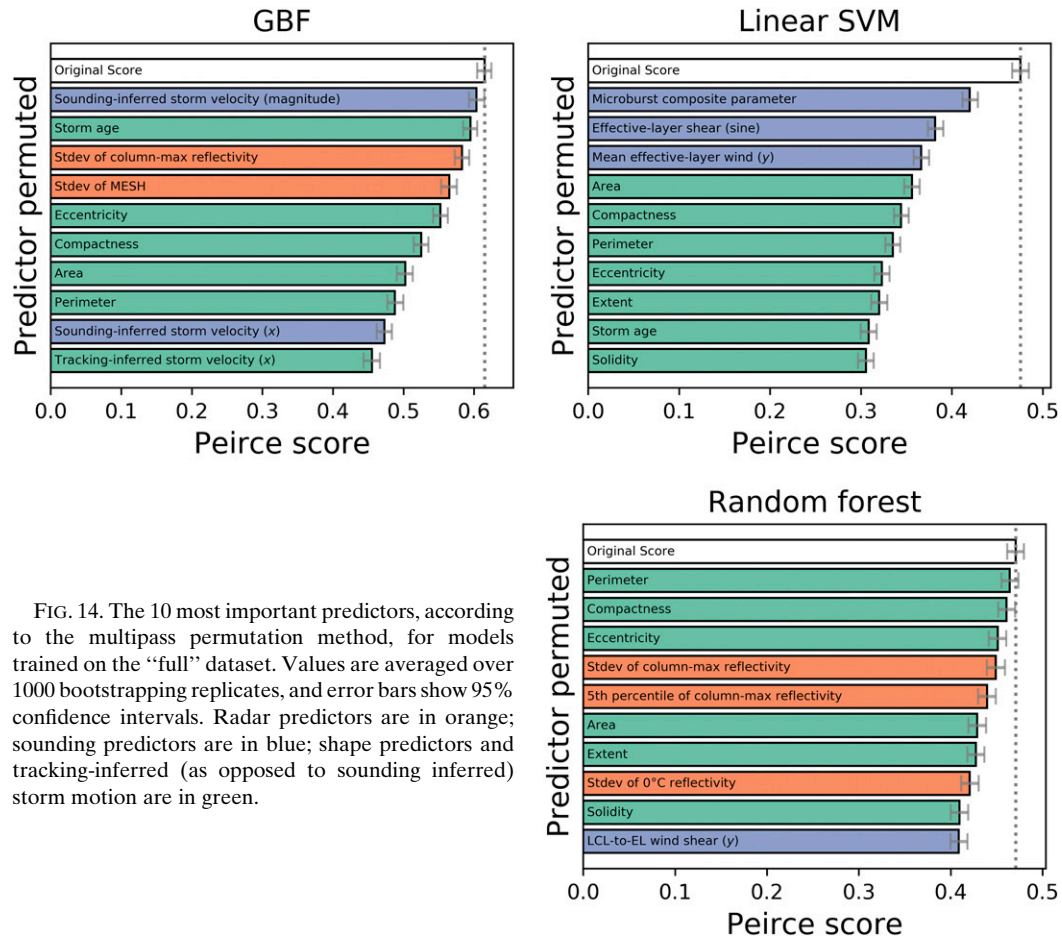


FIG. 14. The 10 most important predictors, according to the multipass permutation method, for models trained on the “full” dataset. Values are averaged over 1000 bootstrapping replicates, and error bars show 95% confidence intervals. Radar predictors are in orange; sounding predictors are in blue; shape predictors and tracking-inferred (as opposed to sounding inferred) storm motion are in green.

and the top predictors for SFS contain a more even mix of predictor types (radar, sounding, and shape), rather than focusing heavily on shape predictors. However, we consider the permutation results more valid, because (i) permutation does not require retraining, so results are based on the same models used elsewhere in the paper; (ii) SFS results cannot easily be cross-referenced with PDPs, because linear SVM is the only model that generates PDPs with perceptibly non-zero slopes but SFS results for the linear SVM have prohibitively large error bars. Thus, for predictors identified as important by SFS, it is difficult to assess *how* they are important.

Automated classification of convective mode could be useful in both operational forecasting and research. In operations, it would save the human labor involved in manually labeling storms and allow meteorologists to focus more energy on forecasting storm motion and attendant severe weather. Also, since convective mode is correlated with storm motion and severe weather (section 1), automated classification could help with these problems as well.

An automated classification system could form an important step in automated damage predictions or systems that automatically adjust storm motion predictions in light of storm modes. An automated classification system could also provide a possible input to severe weather prediction systems (e.g., tornado or lightning forecasting systems). On the research side, automated classification is most important for climatology analyses as an automated system would allow for the generation of labels for storms that were observed but not given a human label. This would increase the range of storms that are used to inform these analyses.

One disadvantage of our models is that they rely on radar observations, so they can classify only existing storms. The models also do not account for convective initiation, so they cannot predict convective mode for future storms. However, this problem could be addressed by training with simulated storms from convection-allowing numerical models. Such an ML model could potentially operate within the Warn-on-Forecast system (WoF; Stensrud et al. 2009;

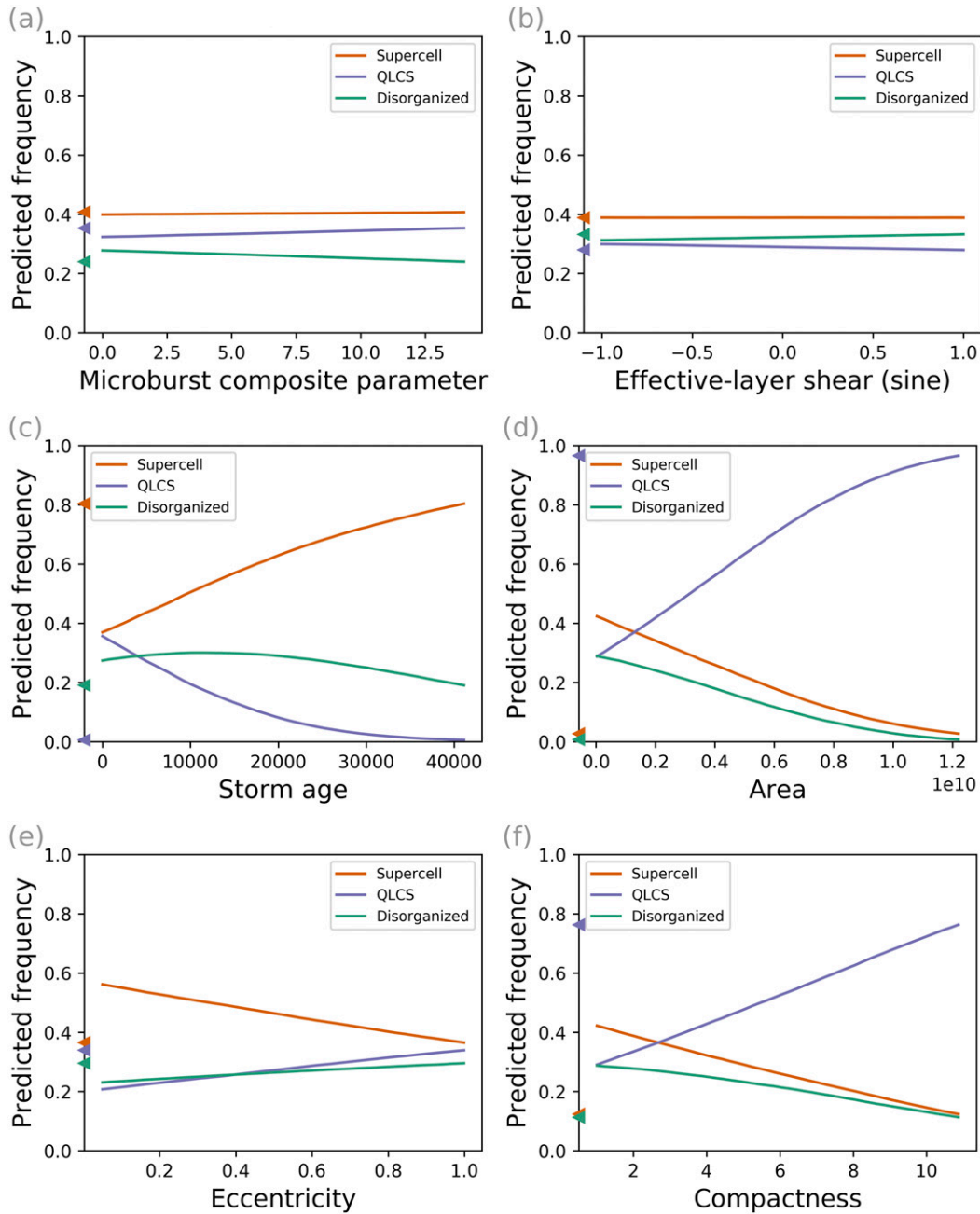


FIG. 15. Partial-dependence plots for the linear SVM. Triangles show the predicted frequencies for missing data. All predictors shown here are unitless, except for storm age (s) and area (m²).

Lawson et al. 2018; Skinner et al. 2018), which aims to provide short-term guidance for thunderstorms and severe weather.

In the near future we will apply convolutional neural networks (CNN), a type of deep-learning model, to this problem. The main advantage of CNNs is that they can learn from gridded data, which would obviate the need to compute radar and sounding statistics (we could

use raw storm-centered radar images and proximity soundings, instead). This approach often leads to better performance than hand-engineering predictors, because the hand-engineered predictors may exclude important relationships. Also, CNNs allow the interpretation outputs to be viewed in the same space as the input grids (McGovern et al. 2019), which is often more intuitive to humans.

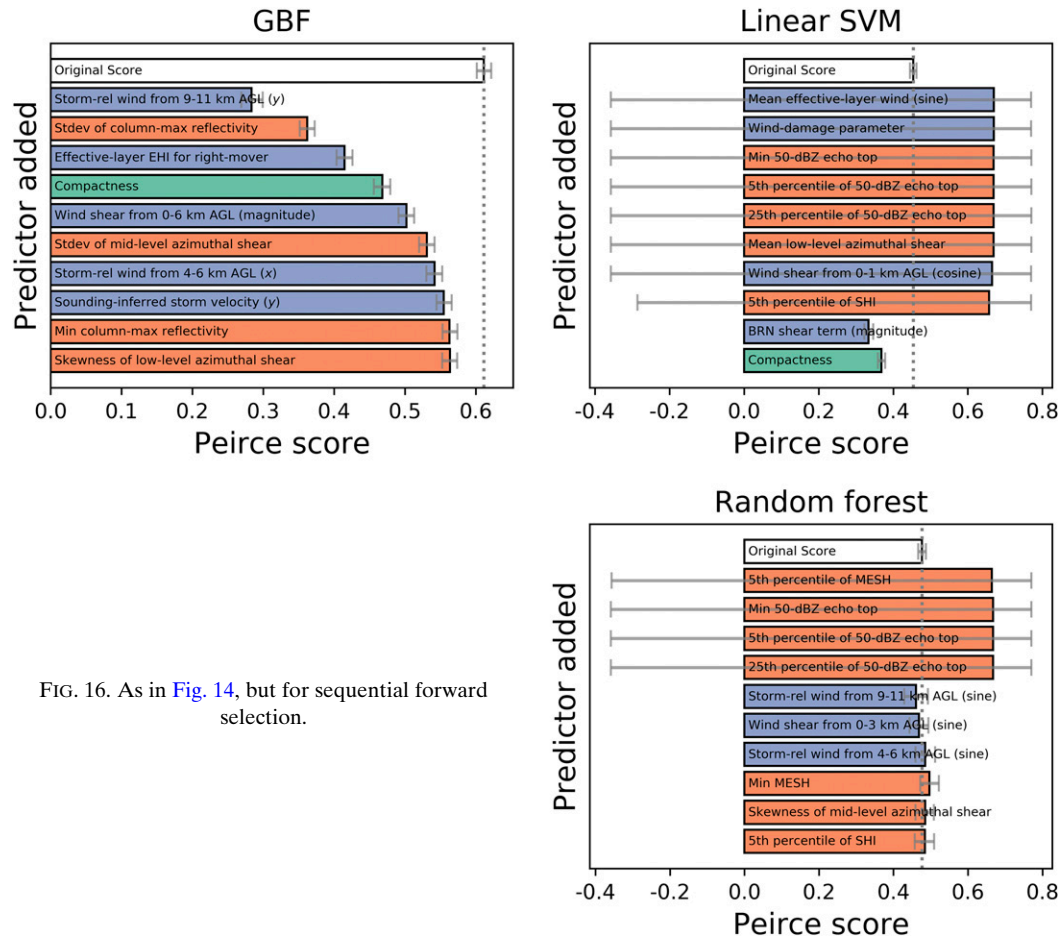


FIG. 16. As in Fig. 14, but for sequential forward selection.

Acknowledgments. The authors thank Rich Thompson and Bryan Smith for sharing their data with us and for the many hours of work put into creating the labels for each storm. We further thank the MYRORSS team and Holly Obermeier for the radar data and additional labels, respectively. This material is based upon work supported by the National Science Foundation under Grant EAGER AGS 1802627. Funding was also provided by the NOAA/Office of Oceanic and Atmospheric Research under NOAA–University of Oklahoma Cooperative Agreement NA16OAR4320115, U.S. Department of Commerce. Most of the computing for this project was performed at the University of Oklahoma Supercomputing Center for Education and Research (OSCER).

APPENDIX

Implementation Details

The best hyperparameters for each model were determined by a grid search (Goodfellow et al. 2016, their

section 11.4.3). We used the cross-validation approach described in section 2—except that we split data into training, validation, and testing rather than just training and testing. We chose the hyperparameters that yielded the best Peirce score on the validation data (averaged over all eight training/validation/testing splits). For all hyperparameters not mentioned here, we used the default values in version 0.20 of the “scikit-learn” software (Pedregosa et al. 2011). For the linear SVM, we used the LinearSVC method (<https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html>) with $c = 4.6$ in Eq. (4) and balanced class weights. Also, we solved the primal optimization problem (rather than dual), as primal optimization is better suited for datasets with more examples than features. For the nonlinear SVM, we used the SVC method (<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>) with $c = 400$ and $\sigma = 1.6 \times 10^{-5}$ in Eq. (5) and balanced class weights. For logistic regression, we used the SGDClassifier method (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html) with logarithmic loss, tolerance of 0.001, 1000 epochs, and $\lambda_1 = 0.1764$ and

$\lambda_2 = 0.0036$ in Eq. (3). The GBF and random forest were implemented with GradientBoostingClassifier (<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>) and RandomForestClassifier (<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>), both with 250 trees and maximum depth of seven splits.

REFERENCES

- Adrianto, I., T. B. Trafalis, and V. Lakshmanan, 2009: Support vector machines for spatiotemporal tornado prediction. *Int. J. Gen. Syst.*, **38**, 759–776, <https://doi.org/10.1080/03081070601068629>.
- Aggarwal, S. K., and L. M. Saini, 2014: Solar energy prediction using linear and non-linear regularization models: A study on AMS (American Meteorological Society) 2013–14 solar energy prediction contest. *Energy*, **78**, 247–256, <https://doi.org/10.1016/j.energy.2014.10.012>.
- Benjamin, S. G., and Coauthors, 2004: An hourly assimilation-forecast cycle: The RUC. *Mon. Wea. Rev.*, **132**, 495–518, [https://doi.org/10.1175/1520-0493\(2004\)132<0495:AHACTR>2.0.CO;2](https://doi.org/10.1175/1520-0493(2004)132<0495:AHACTR>2.0.CO;2).
- , and Coauthors, 2016: A North American hourly assimilation and model forecast cycle: The rapid refresh. *Mon. Wea. Rev.*, **144**, 1669–1694, <https://doi.org/10.1175/MWR-D-15-0242.1>.
- Bluestein, H. B., and M. H. Jain, 1985: Formation of mesoscale lines of precipitation: Severe squall lines in Oklahoma during the spring. *J. Atmos. Sci.*, **42**, 1711–1732, [https://doi.org/10.1175/1520-0469\(1985\)042<1711:FOMLOP>2.0.CO;2](https://doi.org/10.1175/1520-0469(1985)042<1711:FOMLOP>2.0.CO;2).
- Blumberg, W. G., K. T. Halbert, T. A. Supinie, P. T. Marsh, R. L. Thompson, and J. A. Hart, 2017: SHARPPy: An open-source sounding analysis toolkit for the atmospheric sciences. *Bull. Amer. Meteor. Soc.*, **98**, 1625–1636, <https://doi.org/10.1175/BAMS-D-15-00309.1>.
- Breiman, L., 2001: Random forests. *Mach. Learn.*, **45**, 5–32, <https://doi.org/10.1023/A:1010933404324>.
- Byers, H. R., 1949: Structure and dynamics of the thunderstorm. *Science*, **110**, 291–294, <https://doi.org/10.1126/science.110.2856.291>.
- Chilson, C., K. Avery, A. McGovern, E. Bridge, D. Sheldon, and J. Kelly, 2019: Automated detection of bird roosts using NEXRAD radar data and convolutional neural networks. *Remote Sens. Ecol. Conserv.*, **5**, 20–32, <https://doi.org/10.1002/rse2.92>.
- Cintineo, J., M. Pavlonis, J. Sieglaff, and D. Lindsey, 2014: An empirical model for assessing the severe weather potential of developing convection. *Wea. Forecasting*, **29**, 639–653, <https://doi.org/10.1175/WAF-D-13-00113.1>.
- , and Coauthors, 2018: The NOAA/CIMSS ProbSevere Model: Incorporation of total lightning and validation. *Wea. Forecasting*, **33**, 331–345, <https://doi.org/10.1175/WAF-D-17-0099.1>.
- Clark, A. J., A. MacKenzie, A. McGovern, V. Lakshmanan, and R. A. Brown, 2015: An automated, multiparameter dryline identification algorithm. *Wea. Forecasting*, **30**, 1781–1794, <https://doi.org/10.1175/WAF-D-15-0070.1>.
- Clark, T. L., 1979: Numerical simulations with a three-dimensional cloud model: Lateral boundary condition experiments and multicellular severe storm simulations. *J. Atmos. Sci.*, **36**, 2191–2215, [https://doi.org/10.1175/1520-0469\(1979\)036<2191:NSWATD>2.0.CO;2](https://doi.org/10.1175/1520-0469(1979)036<2191:NSWATD>2.0.CO;2).
- Coniglio, M. C., D. J. Stensrud, and M. B. Richman, 2004: An observational study of derecho-producing convective systems. *Wea. Forecasting*, **19**, 320–337, [https://doi.org/10.1175/1520-0434\(2004\)019<0320:AOSODC>2.0.CO;2](https://doi.org/10.1175/1520-0434(2004)019<0320:AOSODC>2.0.CO;2).
- Cortes, C., and V. Vapnik, 1995: Support-vector networks. *Mach. Learn.*, **20**, 273–297, <https://doi.org/10.1007/BF00994018>.
- Crum, T. D., and R. L. Alberty, 1993: The WSR-88D and the WSR-88D operational support facility. *Bull. Amer. Meteor. Soc.*, **74**, 1669–1687, [https://doi.org/10.1175/1520-0477\(1993\)074<1669:TWATWO>2.0.CO;2](https://doi.org/10.1175/1520-0477(1993)074<1669:TWATWO>2.0.CO;2).
- Davies-Jones, R., 2002: Linear and nonlinear propagation of supercell storms. *J. Atmos. Sci.*, **59**, 3178–3205, [https://doi.org/10.1175/1520-0469\(2003\)059<3178:LANPOS>2.0.CO;2](https://doi.org/10.1175/1520-0469(2003)059<3178:LANPOS>2.0.CO;2).
- Efron, B., 1979: Bootstrap methods: Another look at the jackknife. *Ann. Stat.*, **7**, 1–26, <https://doi.org/10.1214/aos/1176344552>.
- Entremont, C., E. Carpenter, B. Bryant, D. Cox, A. Wolverson, and J. Allen, 2018: Microburst composite parameter: A forecasting and analysis approach to determine favorable days for microbursts across the southern states. *Eighth Conf. on Transition of Research to Operations*, Austin, TX, Amer. Meteor. Soc., 1190, <https://ams.confex.com/ams/98Annual/webprogram/Paper333302.html>.
- Fowle, M. A., and P. J. Roebber, 2003: Short-range (0–48-h) numerical prediction of convective occurrence, mode, and location. *Wea. Forecasting*, **18**, 782–794, [https://doi.org/10.1175/1520-0434\(2003\)018<0782:SHNPOC>2.0.CO;2](https://doi.org/10.1175/1520-0434(2003)018<0782:SHNPOC>2.0.CO;2).
- Friedman, J. H., 2001: Greedy function approximation: A gradient boosting machine. *Ann. Stat.*, **29**, 1189–1232, <https://doi.org/10.1214/aos/1013203451>.
- , 2002: Stochastic gradient boosting. *Comput. Stat. Data Anal.*, **38**, 367–378, [https://doi.org/10.1016/S0167-9473\(01\)00065-2](https://doi.org/10.1016/S0167-9473(01)00065-2).
- Gagne, D. J., A. McGovern, and J. Brotzge, 2009: Classification of convective areas using decision trees. *J. Atmos. Oceanic Technol.*, **26**, 1341–1353, <https://doi.org/10.1175/2008JTECHA1205.1>.
- Gallo, B. T., and Coauthors, 2017: Breaking new ground in severe weather prediction: The 2015 NOAA/Hazardous Weather Testbed spring forecasting experiment. *Wea. Forecasting*, **32**, 1541–1568, <https://doi.org/10.1175/WAF-D-16-0178.1>.
- Gallus, W. A., Jr., N. A. Snook, and E. V. Johnson, 2008: Spring and summer severe weather reports over the Midwest as a function of convective mode: A preliminary study. *Wea. Forecasting*, **23**, 101–113, <https://doi.org/10.1175/2007WAF2006120.1>.
- Gerrity, J. P., Jr., 1992: A note on Gandin and Murphy's equitable skill score. *Mon. Wea. Rev.*, **120**, 2709–2712, [https://doi.org/10.1175/1520-0493\(1992\)120<2709:ANOGAM>2.0.CO;2](https://doi.org/10.1175/1520-0493(1992)120<2709:ANOGAM>2.0.CO;2).
- Goodfellow, I., Y. Bengio, and A. Courville, 2016: *Deep Learning*. MIT Press, 775 pp.
- Haberlie, A. M., and W. S. Ashley, 2018: A method for identifying midlatitude mesoscale convective systems in radar mosaics. Part I: Segmentation and classification. *J. Appl. Meteor. Climatol.*, **57**, 1575–1598, <https://doi.org/10.1175/JAMC-D-17-0293.1>.
- Heidke, P., 1926: Berechnung des Erfolges und der Güte der Windstärkevorhersagen im Sturmwarnungsdienst. *Geogr. Ann.*, **8**, 301–349, <https://doi.org/10.1080/20014422.1926.11881138>.
- Hoerl, A., and R. Kennard, 1988: Ridge regression. *Encyclopedia of Statistical Sciences*, S. Kotz, Ed., Vol. 8, John Wiley and Sons, 129–136.
- Houze, R., Jr., 2004: Mesoscale convective systems. *Rev. Geophys.*, **42**, RG4003, <https://doi.org/10.1029/2004RG000150>.
- , and P. Hobbs, 1982: Organization and structure of precipitating cloud systems. *Advances in Geophysics*, Vol. 24, Elsevier, 225–305.
- Jergensen, G., 2019: PermutationImportance. Github Python software library. <https://github.com/gelijergensen/PermutationImportance>.
- Johns, R. H., 1993: Meteorological conditions associated with bow echo development in convective storms. *Wea. Forecasting*,

- 8, 294–299, [https://doi.org/10.1175/1520-0434\(1993\)008<0294:MCAWBE>2.0.CO;2](https://doi.org/10.1175/1520-0434(1993)008<0294:MCAWBE>2.0.CO;2).
- Kain, J. S., and Coauthors, 2008: Some practical considerations regarding horizontal resolution in the first generation of operational convection-allowing NWP. *Wea. Forecasting*, **23**, 931–952, <https://doi.org/10.1175/WAF2007106.1>.
- Knupp, K. R., J. R. Stalker, and E. W. McCaul Jr., 1998: An observational and numerical study of a mini-supercell storm. *Atmos. Res.*, **49**, 35–63, [https://doi.org/10.1016/S0032-5910\(97\)93378-7](https://doi.org/10.1016/S0032-5910(97)93378-7).
- Kohler, M. A., 1949: On the use of double-mass analysis for testing the consistency of meteorological records and for making required adjustments. *Bull. Amer. Meteor. Soc.*, **30**, 188–195, <https://doi.org/10.1175/1520-0477-30.5.188>.
- Lagerquist, R., A. McGovern, and T. Smith, 2017: Machine learning for real-time prediction of damaging straight-line convective wind. *Wea. Forecasting*, **32**, 2175–2193, <https://doi.org/10.1175/WAF-D-17-0038.1>.
- , —, and D. Gagne II, 2019: Deep learning for spatially explicit prediction of synoptic-scale fronts. *Wea. Forecasting*, **34**, 1137–1160, <https://doi.org/10.1175/WAF-D-18-0183.1>.
- Lakshmanan, V., and T. Smith, 2010: Evaluating a storm tracking algorithm. *26th Conf. on Interactive Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, Atlanta, GA, Amer. Meteor. Soc., 8.2, https://ams.confex.com/ams/90annual/techprogram/paper_162556.htm.
- , —, G. Stumpf, and K. Hondl, 2007: The Warning Decision Support System—Integrated Information. *Wea. Forecasting*, **22**, 596–612, <https://doi.org/10.1175/WAF1009.1>.
- , K. Hondl, and R. Rabin, 2009: An efficient, general-purpose technique for identifying storm cells in geospatial images. *J. Atmos. Oceanic Technol.*, **26**, 523–537, <https://doi.org/10.1175/2008JTECHA1153.1>.
- , C. Karstens, J. Krause, K. Elmore, A. Ryzhkov, and S. Berkseth, 2015: Which polarimetric variables are important for weather/no-weather discrimination? *J. Atmos. Oceanic Technol.*, **32**, 1209–1223, <https://doi.org/10.1175/JTECH-D-13-00205.1>.
- Lawson, J. R., J. S. Kain, N. Yussouf, D. C. Dowell, D. M. Wheatley, K. H. Knopfmeier, and T. A. Jones, 2018: Advancing from convection-allowing NWP to warn-on-forecast: Evidence of progress. *Wea. Forecasting*, **33**, 599–607, <https://doi.org/10.1175/WAF-D-17-0145.1>.
- Lemon, L. R., and C. A. Doswell III, 1979: Severe thunderstorm evolution and mesocyclone structure as related to tornado-genesis. *Mon. Wea. Rev.*, **107**, 1184–1197, [https://doi.org/10.1175/1520-0493\(1979\)107<1184:STEAMS>2.0.CO;2](https://doi.org/10.1175/1520-0493(1979)107<1184:STEAMS>2.0.CO;2).
- Liu, Y., and Coauthors, 2016: Application of deep convolutional neural networks for detecting extreme weather in climate datasets. arXiv:1605.01156.
- Louppe, G., L. Wehenkel, A. Sutura, and P. Geurts, 2013: Understanding variable importances in forests of randomized trees. *Proc. 26th Int. Conf. on Neural Information Processing Systems*, Lake Tahoe, CA, Neural Information Processing Systems Foundation, 431–439.
- Malone, T. F., 1955: Application of statistical methods in weather prediction. *Proc. Natl. Acad. Sci. USA*, **41**, 806–815, <https://doi.org/10.1073/pnas.41.11.806>.
- McGovern, A., D. J. Gagne II, J. K. Williams, R. A. Brown, and J. B. Basara, 2014: Enhancing understanding and improving prediction of severe weather through spatiotemporal relational learning. *Mach. Learn.*, **95**, 27–50, <https://doi.org/10.1007/s10994-013-5343-x>.
- , —, J. Basara, T. M. Hamill, and D. Margolin, 2015: Solar energy prediction: An international contest to initiate interdisciplinary research on compelling meteorological problems. *Bull. Amer. Meteor. Soc.*, **96**, 1388–1395, <https://doi.org/10.1175/BAMS-D-14-00006.1>.
- , K. L. Elmore, D. J. Gagne II, S. E. Haupt, C. D. Karstens, R. Lagerquist, T. Smith, and J. K. Williams, 2017: Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bull. Amer. Meteor. Soc.*, **98**, 2073–2090, <https://doi.org/10.1175/BAMS-D-16-0123.1>.
- , R. Lagerquist, D. J. Gagne II, G. E. Jergensen, K. L. Elmore, C. R. Homeyer, and T. Smith, 2019: Making the black box more transparent: Understanding the physical implications of machine learning. *Bull. Amer. Meteor. Soc.*, **100**, 2175–2199, <https://doi.org/10.1175/BAMS-D-18-0195.1>.
- Mecikalski, J. R., J. K. Williams, C. P. Jewett, D. Ahijevych, A. LeRoy, and J. R. Walker, 2015: Probabilistic 0–1-h convective initiation nowcasts that combine geostationary satellite observations and numerical weather prediction model data. *J. Appl. Meteor. Climatol.*, **54**, 1039–1059, <https://doi.org/10.1175/JAMC-D-14-0129.1>.
- Ortega, K., T. Smith, S. Stevens, S. Williams, D. Kingfield, and R. Lagerquist, 2012: The Multi-Year Reanalysis of Remotely Sensed Storms (MYRORSS): Data processing and severe weather projects. *37th Conf. on Radar Meteorology*, Norman, OK, Amer. Meteor. Soc., 205, <https://ams.confex.com/ams/37RADAR/webprogram/Paper275486.html>.
- Parker, M. D., and R. H. Johnson, 2000: Organizational modes of midlatitude mesoscale convective systems. *Mon. Wea. Rev.*, **128**, 3413–3436, [https://doi.org/10.1175/1520-0493\(2001\)129<3413:OMOMMC>2.0.CO;2](https://doi.org/10.1175/1520-0493(2001)129<3413:OMOMMC>2.0.CO;2).
- Pedregosa, F., and Coauthors, 2011: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.
- Peirce, C. S., 1884: The numerical measure of the success of predictions. *Science*, **4**, 453–454, <https://doi.org/10.1126/science.ns-4.93.453-a>.
- Quartz, 2017: The Quartz guide to artificial intelligence: What is it, why is it important, and should we be afraid? Accessed 27 May 2019, <https://qz.com/1046350/the-quartz-guide-to-artificial-intelligence-what-is-it-why-is-it-important-and-should-we-be-afraid/>.
- Quinlan, J. R., 1986: Induction of decision trees. *Mach. Learn.*, **1**, 81–106, <https://doi.org/10.1007/BF00116251>.
- Radhika, Y., and M. Shashi, 2009: Atmospheric temperature prediction using support vector machines. *Int. J. Comput. Theory Eng.*, **1**, 55–58, <https://doi.org/10.7763/IJCTE.2009.V1.9>.
- Rothfusz, L. P., R. Schneider, D. Novak, K. Klockow-McClain, A. E. Gerard, C. Karstens, G. J. Stumpf, and T. M. Smith, 2018: FACETS: A proposed next-generation paradigm for high-impact weather forecasting. *Bull. Amer. Meteor. Soc.*, **99**, 2025–2043, <https://doi.org/10.1175/BAMS-D-16-0100.1>.
- Skinner, P. S., and Coauthors, 2018: Object-based verification of a prototype Warn-on-Forecast system. *Wea. Forecasting*, **33**, 1225–1250, <https://doi.org/10.1175/WAF-D-18-0020.1>.
- Smith, B. T., R. L. Thompson, J. S. Grams, C. Broyles, and H. E. Brooks, 2012: Convective modes for significant severe thunderstorms in the contiguous United States. Part I: Storm classification and climatology. *Wea. Forecasting*, **27**, 1114–1135, <https://doi.org/10.1175/WAF-D-11-00115.1>.
- Sobash, R. A., J. S. Kain, D. R. Bright, A. R. Dean, M. C. Coniglio, and S. J. Weiss, 2011: Probabilistic forecast guidance for severe thunderstorms based on the identification of extreme phenomena in convection-allowing model forecasts.

- Wea. Forecasting*, **26**, 714–728, <https://doi.org/10.1175/WAF-D-10-05046.1>.
- Stensrud, D. J., and Coauthors, 2009: Convective-scale warn-on-forecast system: A vision for 2020. *Bull. Amer. Meteor. Soc.*, **90**, 1487–1500, <https://doi.org/10.1175/2009BAMS2795.1>.
- Storm Prediction Center, 2019: Mesoscale analysis. NOAA, accessed 28 May 2019, <https://www.spc.noaa.gov/exper/mesoanalysis/>.
- Stumpf, G. J., A. Witt, E. D. Mitchell, P. L. Spencer, J. T. Johnson, M. D. Eilts, K. W. Thomas, and D. W. Burgess, 1998: The National Severe Storms Laboratory mesocyclone detection algorithm for the WSR-88D. *Wea. Forecasting*, **13**, 304–326, [https://doi.org/10.1175/1520-0434\(1998\)013<0304:TNSSLM>2.0.CO;2](https://doi.org/10.1175/1520-0434(1998)013<0304:TNSSLM>2.0.CO;2).
- Suleiman, A., M. R. Tight, and A. D. Quinn, 2016: Hybrid neural networks and boosted regression tree models for predicting roadside particulate matter. *Environ. Model. Assess.*, **21**, 731–750, <https://doi.org/10.1007/s10666-016-9507-5>.
- Thompson, R. L., R. Edwards, J. A. Hart, K. L. Elmore, and P. Markowski, 2003: Close proximity soundings within supercell environments obtained from the Rapid Update Cycle. *Wea. Forecasting*, **18**, 1243–1261, [https://doi.org/10.1175/1520-0434\(2003\)018<1243:CPSWSE>2.0.CO;2](https://doi.org/10.1175/1520-0434(2003)018<1243:CPSWSE>2.0.CO;2).
- , B. T. Smith, J. S. Grams, A. R. Dean, and C. Broyles, 2012: Convective modes for significant severe thunderstorms in the contiguous United States. Part II: Supercell and QLCS tornado environments. *Wea. Forecasting*, **27**, 1136–1154, <https://doi.org/10.1175/WAF-D-11-00116.1>.
- Tibshirani, R., 1996: Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc.*, **58B**, 267–288, <https://doi.org/10.1111/J.2517-6161.1996.TB02080.x>.
- Trafalis, T., H. Ince, and M. Richman, 2003: Tornado detection with support vector machines. *Int. Conf. on Computational Science*, Melbourne, Australia and St. Petersburg, Russia, Springer, 289–298.
- Trapp, R. J., S. A. Tessendorf, E. S. Godfrey, and H. E. Brooks, 2005: Tornadoes from squall lines and bow echoes. Part I: Climatological distribution. *Wea. Forecasting*, **20**, 23–34, <https://doi.org/10.1175/WAF-835.1>.
- Vapnik, V., 1963: Pattern recognition using generalized portrait method. *Auto. Remote Control*, **24**, 774–780.
- , 1995: *The Nature of Statistical Learning Theory*. Springer, 188 pp.
- Wallace, J., and P. Hobbs, 2006: *Atmospheric Science: An Introductory Survey*. Elsevier, 504 pp.
- Wang, L., K. A. Scott, L. Xu, and D. A. Clausi, 2016: Sea ice concentration estimation during melt from dual-pol SAR scenes using deep convolutional neural networks: A case study. *IEEE Trans. Geosci. Remote Sens.*, **54**, 4524–4533, <https://doi.org/10.1109/TGRS.2016.2543660>.
- Webb, A., 2003: *Statistical Pattern Recognition*. John Wiley and Sons, 514 pp.
- Weisman, M. L., and J. B. Klemp, 1984: The structure and classification of numerically simulated convective storms in directionally varying wind shears. *Mon. Wea. Rev.*, **112**, 2479–2498, [https://doi.org/10.1175/1520-0493\(1984\)112<2479:TSACON>2.0.CO;2](https://doi.org/10.1175/1520-0493(1984)112<2479:TSACON>2.0.CO;2).
- Williams, J. K., 2014: Using random forests to diagnose aviation turbulence. *Mach. Learn.*, **95**, 51–70, <https://doi.org/10.1007/s10994-013-5346-7>.
- Wimmers, A., C. Velden, and J. H. Cossuth, 2019: Using deep learning to estimate tropical cyclone intensity from satellite passive microwave imagery. *Mon. Wea. Rev.*, **147**, 2261–2282, <https://doi.org/10.1175/MWR-D-18-0391.1>.
- Zou, H., and T. Hastie, 2005: Regularization and variable selection via the elastic net. *J. Roy. Stat. Soc.*, **67B**, 301–320, <https://doi.org/10.1111/j.1467-9868.2005.00503.x>.