Deep Learning on Three-Dimensional Multiscale Data for Next-Hour Tornado Prediction

RYAN LAGERQUIST

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

AMY MCGOVERN

University of Oklahoma, Norman, Oklahoma

CAMERON R. HOMEYER

School of Meteorology, University of Oklahoma, Norman, Oklahoma

DAVID JOHN GAGNE II

National Center for Atmospheric Research, Boulder, Colorado

TRAVIS SMITH

Cooperative Institute for Mesoscale Meteorological Studies, Norman, Oklahoma

(Manuscript received 20 November 2019, in final form 30 April 2020)

ABSTRACT

This paper describes the development of convolutional neural networks (CNN), a type of deep-learning method, to predict next-hour tornado occurrence. Predictors are a storm-centered radar image and a proximity sounding from the Rapid Refresh model. Radar images come from the Multiyear Reanalysis of Remotely Sensed Storms (MYRORSS) and Gridded NEXRAD WSR-88D Radar dataset (GridRad), both of which are multiradar composites. We train separate CNNs on MYRORSS and GridRad data, present an experiment to optimize the CNN settings, and evaluate the chosen CNNs on independent testing data. Both models achieve an area under the receiver-operating-characteristic curve (AUC) well above 0.9, which is considered to be excellent performance. The GridRad model achieves a critical success index (CSI) of 0.31, and the MYRORSS model achieves a CSI of 0.17. The difference is due primarily to event frequency (percentage of storms that are tornadic in the next hour), which is 3.52% for GridRad but only 0.24% for MYRORSS. The best CNN predictions (true positives and negatives) occur for strongly rotating tornadic supercells and weak nontornadic cells in mesoscale convective systems, respectively. The worst predictions (false positives and negatives) occur for strongly rotating nontornadic supercells and tornadic cells in quasi-linear convective systems, respectively. The performance of our CNNs is comparable to an operational machine-learning system for severe weather prediction, which suggests that they would be useful for real-time forecasting.

1. Introduction

Tornadoes are one of the costliest weather disasters in the United States (Insurance Information Institute 2019). The National Weather Service (NWS) is responsible for issuing tornado warnings and generally issues warnings at lead times up to 30 min (Brooks and Correia 2018), with durations up to 60 min (Harrison and Karstens 2017). Although the skill of NWS tornado

Supplemental information related to this paper is available at the Journals Online website: https://doi.org/10.1175/MWR-D-19-0372.s1.

Corresponding author: Ryan Lagerquist, ryan.lagerquist@ou.edu

DOI: 10.1175/MWR-D-19-0372.1

warnings has generally improved over time, critical success index (CSI) and lead time have stagnated in the last decade (Brooks and Correia 2018). During this time, the amount of data available to forecasters has exploded—including dual-polarization radar observations, high-resolution satellite observations, and forecasts from convection-allowing models (CAM). However, none of these datasets explicitly resolves tornadoes, so they must still be translated into useful information by forecasters, which can lead to cognitive overload (Wilson et al. 2017). This problem can be alleviated by explicit tornado-modeling and postprocessing methods, the latter of which combine multisource data into explicit tornado predictions (Karstens et al. 2018).

Much work in this area falls under the Warn-on-Forecast initiative (WoF; Stensrud et al. 2009, 2013). The main goal of WoF is to shift the current warning paradigm from extrapolation based on current observations (warn on detection) to use of short-range CAM simulations. This effort includes creating explicit probabilistic tornado forecasts at 0–1-h lead times. CAMs typically have 1–4-km horizontal grid spacing, which allows them to explicitly resolve some thunderstorms but not individual hazards such as tornadoes.[1] However, CAMs do resolve midlevel and sometimes low-level mesocyclones, which are necessary precursors for supercell tornadogenesis (Davies-Jones et al. 2001; Markowski and Richardson 2009, 2014). Yussouf et al. (2015) and Wheatley et al. (2015) ran 3-km CAM ensembles for several tornado outbreaks, at lead times up to 1 h, and skillfully simulated the low-level mesocyclone in many storms. A longer-term goal of WoF is to establish "the feasibility of explicit ensemble probabilistic prediction of tornadoes" (Snook et al. 2019). With this aim, Snook et al. (2019) ran a 50-m CAM ensemble for one storm, with all members correctly producing a tornado and 4 of 10 members correctly producing winds of EF5 strength. The main shortcoming of 3-km models is their inability to explicitly resolve tornadoes, while 50-m models are too computationally expensive to run in real time (and may continue to be for decades).

For non-tornado-resolving CAMs, storm surrogates are often used to relate resolved quantities to tornado occurrence. The most popular surrogate is updraft helicity (UH; Kain et al. 2008), which is the height-integrated product of vertical velocity and vertical vorticity. To create a surrogate severe probability forecast (SSPF; Sobash et al. 2011), UH is thresholded to create a binary mask (0 or 1 at each grid point), which is then smoothed via kernel density estimation. The UH threshold and smoothing radius are chosen to maximize predictive skill, defined by how well the forecast matches "ground truth," which may consist of observed tornadoes or a proxy such as radar-derived rotation tracks. The main disadvantage of the former is underreporting bias in sparsely populated areas, discussed in section 3a; the main disadvantage of the latter is that in many places the radar network does not have sufficient resolution and low-level coverage to identify tornadoes.

Sobash et al. (2011) is the first study to use SSPF, predicting the probability of any severe weather.[2] In later work, Sobash et al. (2016b) applied SSPF to a CAM ensemble, outperforming the same technique applied to a deterministic CAM. Sobash et al. (2016a) was the first study to use SSPF to discriminate between tornadic and nontornadic storms, obtaining skillful next-day predictions at the larger smoothing radii (≥160 km). Gallo et al. (2016) used a similar approach but incorporated near-storm environment (NSE) variables, such as convective available potential energy (CAPE) and the significant-tornado parameter, which greatly reduced overprediction of tornadoes. While the aforementioned studies focused on 1–2-day lead times, SSPF is run annually on a 3-km CAM ensemble in the Hazardous Weather Testbed (HWT; Clark et al. 2012; Gallo et al. 2017), where the focus is on 0–3-h lead times, and skillfully predicts radar-derived rotation tracks (Skinner et al. 2018).

Another widely used postprocessing approach is machine learning (ML). One of the earliest efforts was the NSSL Severe Weather Potential algorithm (Kitzmiller et al. 1995), a linear-regression model that predicted any severe weather in the next 20 min. Marzban and Stumpf (1996) used neural networks to predict tornadogenesis for a given mesocyclone in the next 20 min. Lakshmanan et al. (2005) and Adrianto et al. (2009) used fuzzy logic and support-vector machines, respectively, to produce a spatiotemporal tornado-probability grid for the next 30 min. Because of computational limitations at the time, the aforementioned studies used only radar data as predictors. Gagne et al. (2012) trained a spatiotemporal relational random forest with radar data, surface observations, and NSE variables from a reanalysis to predict tornado probability for a given supercell.

[1] Physical models cannot resolve features with a length scale of less than $\sim 6\delta x$, where δx is the horizontal grid spacing. Thus, coarser-resolution (4 km) CAMs cannot resolve small thunderstorms. Similarly, the 50-m CAMs mentioned later in this paragraph cannot resolve small tornadoes.

[2] Hereinafter defined as a tornado, hail with diameter ≥ 25.4 mm, *or* wind gust ≥ 25.7 m s^{-1}.

They found that many of the best predictors came from the surface and NSE datasets. Cintineo et al. (2014, 2018) developed an operational algorithm called ProbSevere, which uses naïve Bayes to forecast any severe weather for a given storm. Their predictors are derived from radar, satellite, and lightning data, as well as NSE variables from the Rapid Refresh model. ProbSevere has run in the HWT for several years, receiving favorable feedback from forecasters. It has improved upon the median lead time of NWS tornado and severe-thunderstorm warnings but at the cost of a decrease in CSI (Cintineo et al. 2018).

Convolutional neural networks (CNN) are specially designed to learn from spatial grids and often contain many layers, which qualifies them as a deep-learning method (section 1.1.4 of Chollet 2018). In traditional ML, spatial grids must be transformed into scalar features, which become the direct inputs to the model. Examples are principal components, spatial statistics (such as means and standard deviations), and raw grid-point values (where each value in the grid is treated as a scalar feature, with no regard to spatial structure). Inevitably, since the transformation to scalar features is done as a preprocessing step rather than informed by ML, it does not optimally exploit the spatial information available. In contrast, spatial grids are fed directly into a CNN, which simultaneously learns to transform the grids into features and the features into predictions. This synergy generally improves skill (Krizhevsky et al. 2017; Dieleman et al. 2015; Silver et al. 2016) and reduces the amount of preprocessing needed, relative to traditional ML methods. CNNs have been used in atmospheric science to estimate sea ice concentration (Wang et al. 2016) and tropical-cyclone intensity (Wimmers et al. 2019) from satellite images, detect extreme-weather patterns in model output (Racah et al. 2017; Kurth et al. 2018; Lagerquist et al. 2019), replace subgrid-scale parameterizations in numerical models (Bolton and Zanna 2019), and improve the understanding and prediction of convective hazards (McGovern et al. 2019; Gagne et al. 2019). CNNs are becoming popular tools in the geosciences at large, and Reichstein et al. (2019) and Gil et al. (2019) have recently called for a vast expansion of our efforts to incorporate deep learning into the geosciences.

This paper describes the development and testing of CNNs to predict next-hour tornado occurrence. In addition to their ability to learn relevant spatial features at multiple scales, CNNs, like other ML methods, can effectively leverage data from multiple sources (Gagne et al. 2012; Cintineo et al. 2014, 2018). Specifically, the predictors used in this study (for each storm) are a storm-centered radar image, representing the storm itself, and a numerically modeled proximity sounding, representing the ambient environment through which the storm is moving. The two datasets have very different characteristics (i.e., radar data are 2D or 3D with high spatial resolution, while soundings are 1D with lower spatial resolution), which would present a major difficulty for non-ML-based postprocessing methods such as SSPF.

The rest of this paper is organized as follows. Section 2 briefly describes the inner workings of CNNs [a more thorough description is provided in Lagerquist et al. (2019), hereafter L19], section 3 describes the input data and preprocessing, section 4 describes experiments used to find the best CNNs, section 5 evaluates performance of the best CNNs, and section 6 summarizes and discusses future work.

2. Convolutional neural networks

As shown in Figs. 1 and 2, a CNN contains three types of specialized layers: convolutional and pooling layers, which turn the input maps into abstractions called "feature maps," and dense layers, which turn the feature maps into predictions. Maps received by the first convolutional layer (leftmost in Figs. 1 and 2) contain raw weather fields; maps received by deeper layers are feature maps, containing transformations of the raw fields. The number of feature maps increases with depth, which increases the number of features that can be learned. The convolution operator is defined in Eq. (4) of L19 and animated in our Fig. S1 in the online supplemental material. Convolution is both spatial and multivariate, so it encodes spatial patterns that combine all input variables.

Each convolutional layer applies two operations after the convolution itself: activation and batch normalization. Activation is a nonlinear function applied elementwise to the feature maps. Without activation functions, the CNN could learn only linear relationships, because convolution is a linear operation. A popular activation function is the leaky rectified linear unit (ReLU; Maas et al. 2013), used in this work. Activation is followed by batch normalization (Ioffe and Szegedy 2015), which is also applied elementwise to the feature maps. Batch normalization transforms each element to approximately a Gaussian distribution,[3] with mean of 0.0 and standard deviation of 1.0. This speeds up learning (Ioffe and Szegedy 2015) and alleviates the vanishing-gradient problem that arises in neural networks with many layers (see section 1 of L19 for a detailed discussion).

[3] Batch normalization is applied separately to each batch of N training examples. Thus, for element x of the feature maps (one variable at one grid point), it forces x to approximately a Gaussian distribution over the N examples.

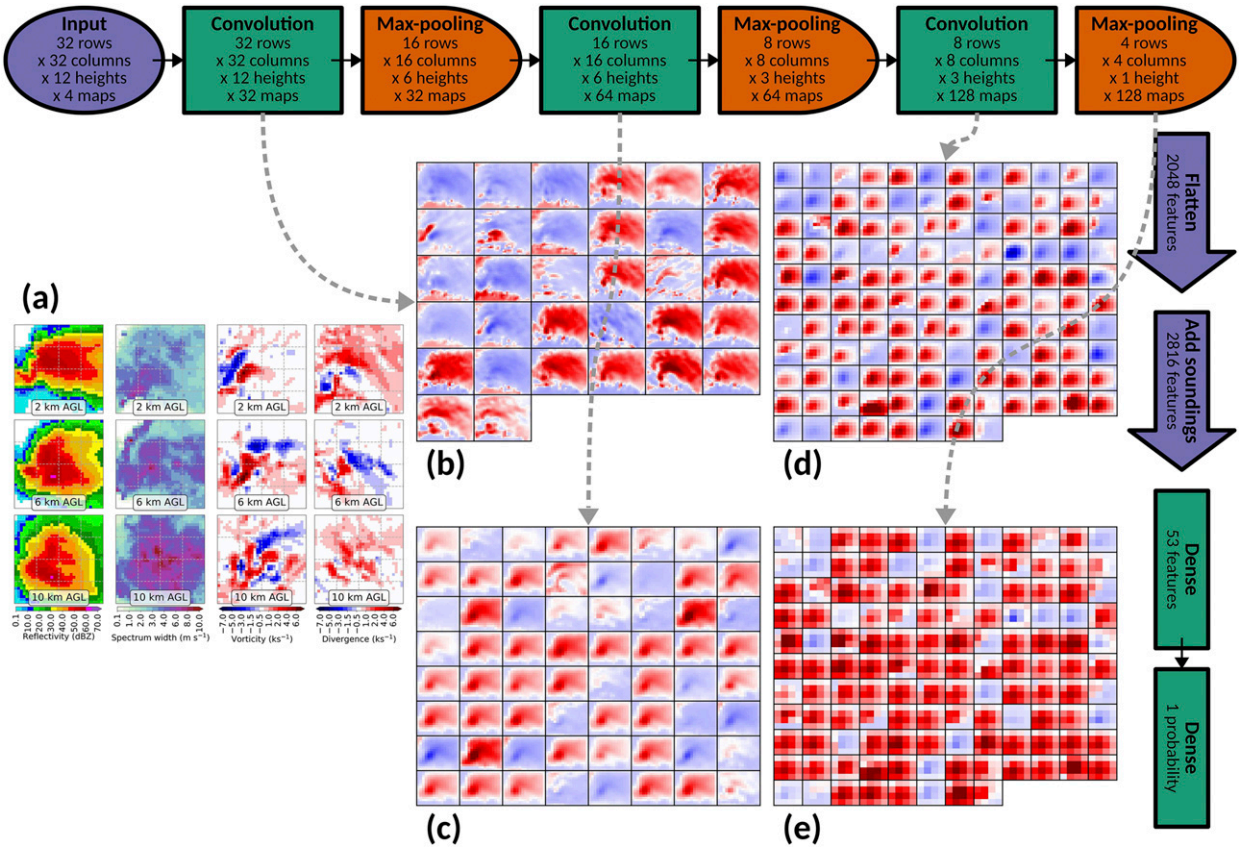

FIG. 1. Architecture of CNN trained with GridRad data. The feature maps in (b)–(e) are shown only for the lowest height and use a diverging color scheme, with negative values in blue and positive values in red. (a) Radar predictors, consisting of a $32 \times 32 \times 12$ grid with four maps. For the sake of brevity, only 3 of 12 radar heights are shown and sounding predictors are not shown. Also shown are feature maps produced (b) by the first convolutional layer, after activation and batch normalization; (c) as in (b), but by the second convolutional layer; (d) as in (b), but by the last convolutional layer; and (e) by the last pooling layer. The flattening layer transforms these maps into a vector of length 2048 ($4 \times 4 \times 1 \times 128$), which is concatenated with sounding features produced by 1D convolution and pooling (Fig. D1 in the online supplemental material). The dense layers transform this concatenated vector into representations of exponentially decreasing length ($2816 \rightarrow 53 \rightarrow 1$), and the final output is next-hour tornado probability.

A pooling layer downsamples feature maps, using either a maximum or mean filter. Each map is downsampled independently. Following common practice, this work uses a maximum filter with downsampling factor of 2, which halves the spatial resolution (doubles the grid spacing). For example, in Fig. 1, pooling layers increase the horizontal spacing from 1.5 to 3.0 to 6.0 km, which allows deeper convolutional layers to learn larger-scale features. This, combined with the fact that feature maps in deeper layers have passed through more convolutions and nonlinear activations, allows deeper layers to learn higher-level abstractions. The pooling operation is animated in Fig. S2 of the online supplemental material.

The dense layers (called ''hidden layers'' in chapter 6 of Goodfellow et al. 2016) transform feature maps into predictions. Since the dense layers are spatially agnostic, feature maps are flattened into a 1D vector before they are passed to the dense layers (Figs. 1 and 2). Each feature in one dense layer is a weighted sum of those in the previous layer. All dense layers except the last follow this linear transformation with leaky ReLU and batch normalization, like the convolutional layers. The last dense layer uses the sigmoid activation function (section 6.2.2.2 of Goodfellow et al. 2016), which forces the output to range over [0, 1], allowing it to be interpreted as a probability. The last dense layer does not use batch normalization, because this would force the outputs to a Gaussian distribution, which permits values outside [0, 1] and is therefore invalid for probabilities.

The convolutional and dense layers contain all adjustable weights in the CNN. These weights are initialized randomly and fit during training to minimize cross entropy [Eq. (1)]. In Eq. (1), p_i is the forecast tornado probability and y_i is the true label (1 if tornadic and 0 otherwise) for the ith example, N is the number of examples, and ε is the cross entropy, ranging over $[0, \infty)$.

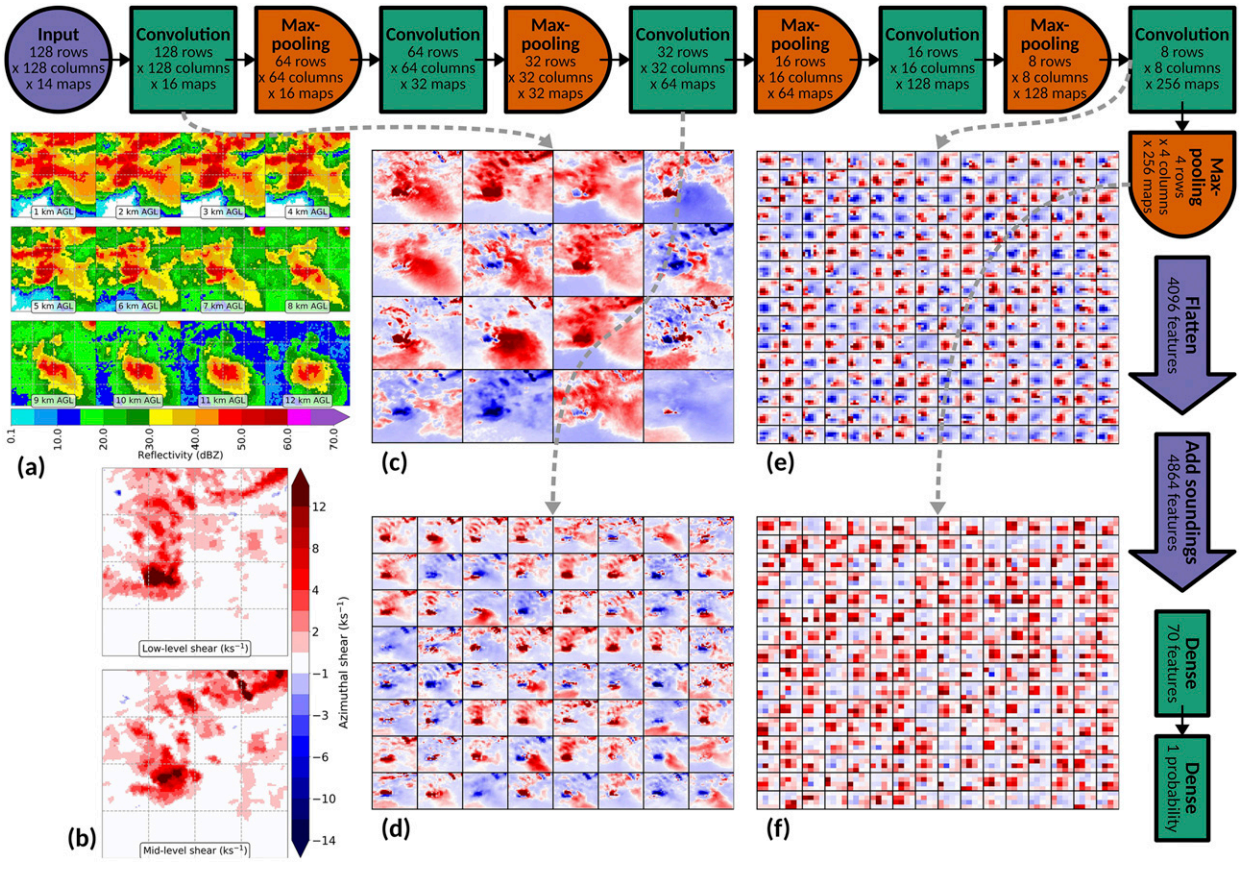

FIG. 2. Architecture of CNN trained with MYRORSS data. The feature maps in (c)–(f) use a diverging color scheme, with negative values in blue and positive values in red. (a),(b) Radar predictors, consisting of a 128 × 128 grid with 14 maps. Also shown are feature maps produced (c) by the first convolutional layer, after activation and batch normalization; (d) as in (c), but by the third convolutional layer; (e) as in (c), but by the last convolutional layer; and (f) by the last pooling layer. As in Fig. 1, the flattening and dense layers transform radar- and sounding-derived features into next-hour tornado probability.

Lower values mean that there is a better correspondence between predictions and labels:

$$\varepsilon = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log_2(p_i) + (1 - y_i)\log_2(1 - p_i)]. \qquad (1)$$

3. Data description and preprocessing

We use three datasets to create predictors: the Multiyear Reanalysis of Remotely Sensed Storms (MYRORSS; Ortega et al. 2012), Gridded NEXRAD Weather Surveillance Radar-1988 Doppler (WSR-88D) radar dataset (GridRad; Homeyer and Bowman 2017), and Rapid Refresh[4] numerical weather model (RAP; Benjamin et al. 2016). Characteristics of these datasets are summarized in Table 1. For each storm we use MYRORSS or GridRad to create a storm-centered radar image, representing the storm at forecast time, and the RAP to create a proximity sounding, representing the environment in which the storm will evolve over the next hour. Both types of information are critical to storm evolution and the development of hazards such as tornadoes. We use NWS storm reports (National Climatic Data Center 2020) to determine when, if at all, the storm is tornadic. We train separate CNNs with MYRORSS and GridRad and test both CNNs on the one year of overlap (2011; see Table 1). Using both datasets demonstrates the generalizability of our methods and results, especially patterns leading to the best and worst predictions (section 5).

Section 3a describes these datasets in more detail, while the remaining sections 3b–3d discuss preprocessing methods. Sections 3b and 3c describe storm tracking and tornado attribution, used to link tornadoes to storms, and section 3d describes the creation of predictors. Section A in the online supplemental material describes the estimation

[4] For initialization times before 1 May 2012, we use the Rapid Update Cycle (RUC; Benjamin et al. 2004), which was replaced by the RAP on 1 May 2012.

TABLE 1. Summary of raw input data. Here, MSL indicates above mean sea level.

Dataset	Time period	Time step	Horizontal spacing	Vertical levels
MYRORSS reflectivity	2000–11	5 min	0.01°	0.25, 0.50, . . . , 3.00 km MSL, then 3.5, 4.0, . . . , 9.0 km MSL, and then 10, 11, . . . , 20 km MSL
MYRORSS shear	2000–11	5 min	0.005°	Low level (max from 0 to 2 km AGL); midlevel (max from 3 to 6 km AGL)
GridRad	Selected days in 2011–18	5 min	0.0208°	0.5, 1.0, . . . , 7.0 km MSL and 8, 9, . . . , 22 km MSL
RAP	May 2012–present	1 h	13 or 20 km	100, 125, . . . , 1000 hPa
RUC	Apr 2002–Apr 2012	1 h	13 or 20 km	100, 125, . . . , 1000 hPa

of storm velocity, used for all preprocessing methods discussed in the main text, and supplemental sections B and C describe echo classification and storm detection, used to create the objects tracked by the algorithm in section 3b. The rest of this paper will use the term "storm object" to mean one storm cell at one time.

a. Data description

MYRORSS[5] contains quality-controlled, merged data from all WSR-88D (Crum and Alberty 1993) sites in the contiguous United States (CONUS). Each radar scans a different part of the atmosphere, and where multiple radars scan the same point, they generally have differing resolution and errors. Merging data from all radars allows the data to be represented on a common grid, and the merging algorithm includes quality-control measures that cannot be applied to single-radar data. The merging algorithm is part of the Warning Decision Support System–Integrated Information (WDSS-II; Lakshmanan et al. 2007), a software package for the visualization, analysis, and forecasting of thunderstorms and their attendant hazards. At each 5-min time step, MYRORSS contains a 3D reflectivity grid, plus 2D grids of low-level and midlevel azimuthal shear (the azimuthal derivative of radial velocity; Mahalik et al. 2019). Low-level shear is the maximum from 0 to 2 km above ground level (AGL), and midlevel shear is the maximum from 3 to 6 km AGL.

GridRad also contains merged data from WSR-88D radars. The main differences between MYRORSS and GridRad are different merging algorithms, time periods, and variables (Table 1). Although the public GridRad dataset (Bowman and Homeyer 2017) has 1-h time steps, we have obtained 5-min data for 147 days.[6] These days represent a variety of scenarios—including large tornado outbreaks, small outbreaks, and nonoutbreaks in all seasons. The spatial domain for each day is different and generally not CONUS-wide, but the domain usually covers a large portion of the CONUS, including most tornadoes on the given day and many nontornadic storms. At each 5-min time step, GridRad includes four variables on the 3D grid: reflectivity, spectrum width, vorticity (twice azimuthal shear), and divergence (twice radial shear).

The RAP is a nonhydrostatic mesoscale model with 13- or 20-km grid spacing and covers much of North America, including the full CONUS. The RAP is run every hour and produces forecasts at 1-h time steps, at 37 pressure levels spaced equally from 100 to 1000 mb. We use the RAP instead of another physical model because the RAP has a long and mostly complete archive (https://www.ncei.noaa.gov/thredds/catalog.html) and is commonly used in convective meteorology, including as the background field for the Storm Prediction Center (2020) mesoanalysis. The disadvantage of operational models like the RAP is that they change configurations over time (e.g., Table 1 of Benjamin et al. 2016), creating inhomogeneities that can negatively impact CNN performance. However, our goal in this work is to create CNNs that can be operationalized, so it is important to use operational data as much as possible (see discussion in section 6). Also, we have found that performance [specifically, the area under the receiver-operating-characteristic curve (AUC), defined in section 4] does not decline from the training to validation/testing data, which suggests that the CNNs do not overfit to particular RAP configurations. However, this does not rule out that a large configuration change in the future could adversely affect the CNNs.

A known issue with NWS tornado reports is that many tornadoes are unreported, especially in sparsely populated areas and at night (Doswell et al. 1999). This is why some researchers use other datasets as ground truth, such as radar-derived rotation tracks (e.g., Skinner et al. 2018). These tracks can directly resolve mesocyclones but not tornadoes, which is a major disadvantage, because most mesocyclones do not produce tornadoes

[5] Since the MYRORSS dataset is very large (~75 terabytes), it is not available for public download. However, the data are available upon request from author T. Smith.

[6] Hourly data are available publicly via the Research Data Archive (RDA; Bowman and Homeyer 2017), and 5-min data are available upon request from author C. Homeyer; 5-min data will eventually be publicly available on the RDA as well.

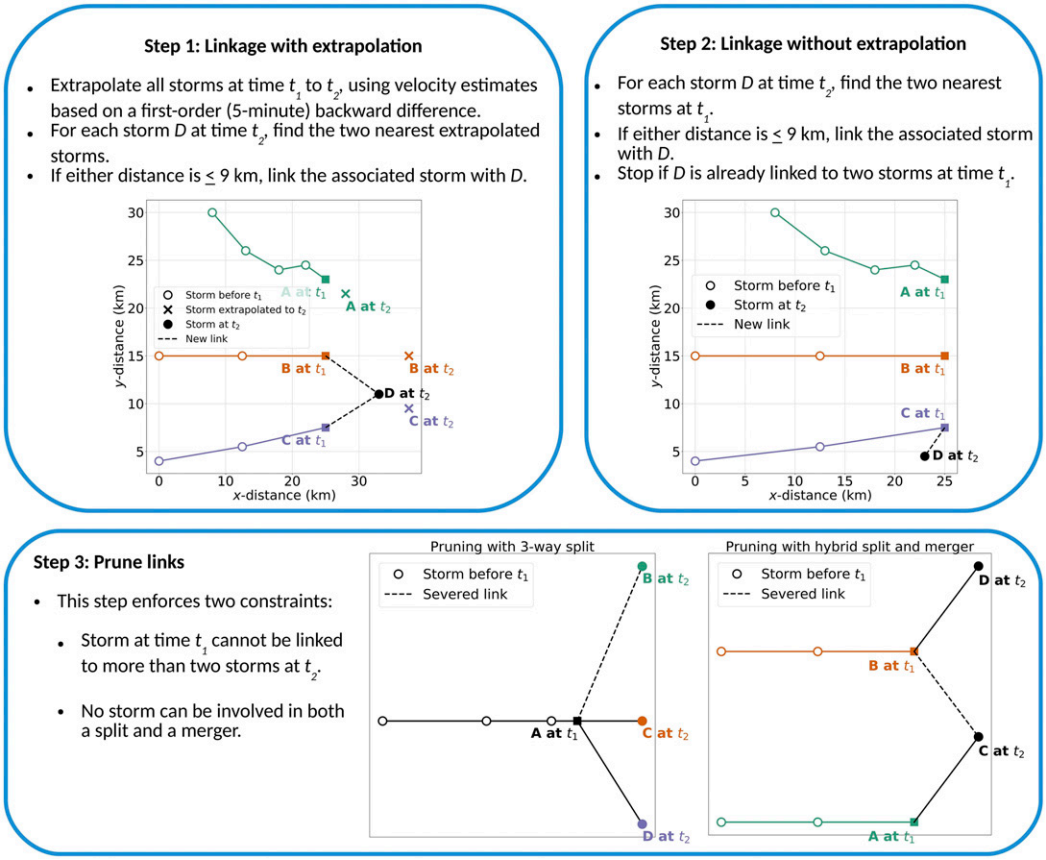

FIG. 3. Flowchart for the preliminary storm-tracking algorithm. In step 1, the extrapolated locations of storms B and C are within 9 km of storm D, so B and C are linked to D. Thus, storms B and C merge into D between times t_1 and t_2. In step 2, storm C at time t_1 is within 9 km of storm D at t_2, so the two are linked. In the three-way split shown for step 3, storm A is initially linked to three storms at the next time step; only the links to the two nearest storms are kept. In the hybrid split and merger shown for step 3, storm C at time t_2 is involved in both a split (storm B at t_1 into storms C and D at t_2) and a merger (storms A and B at t_1 into storm C at t_2). The linkage between storms B and C is severed, leading to the simplest solution (no splits or mergers).

(Trapp et al. 2005) and some tornadoes are not associated with mesocyclones. Thus, we use NWS reports despite their population bias. We also evaluate the CNNs for strong tornadoes only (enhanced Fujita scale EF2+; section 5), which have little to no population bias, because they generally last long enough and cause enough damage to be reported (Anderson et al. 2007; Elsner et al. 2013).

b. *Storm tracking*

Storms are tracked over time via two algorithms, called preliminary and final. The idea of two-stage tracking is motivated by WDSS-II, which uses "segmotion" (Lakshmanan and Smith 2010) for preliminary tracking and "w2besttrack" (Lakshmanan et al. 2015) for final tracking. Our preliminary tracking (Fig. 3) is applied separately to each pair of consecutive time steps, t_1 and t_2. Step 2 (linkage without extrapolation) serves as a "second chance" to link storms, which is useful because step 1 uses first-order backward differences to estimate storm velocity (supplemental section A), which are often noisy. Steps 1 and 2 both ensure that no more than two storms at t_1 can be linked to the same storm at t_2. Otherwise, more than two storms could merge into one over 5 min—which rarely, if ever, occurs in the real atmosphere. Step 3 (pruning) ensures that no more than two storms at t_2 can be linked to the same storm at t_1. Otherwise, one storm could split into more than two storms over 5 min, which is also implausible. The 9-km distance threshold in steps 1 and 2 was chosen subjectively. Based on visual inspection, smaller thresholds often split one storm into several tracks, while larger thresholds often connected several storms into the same track.

The main shortcoming of the preliminary algorithm is that it often "drops" a storm track for one time step,

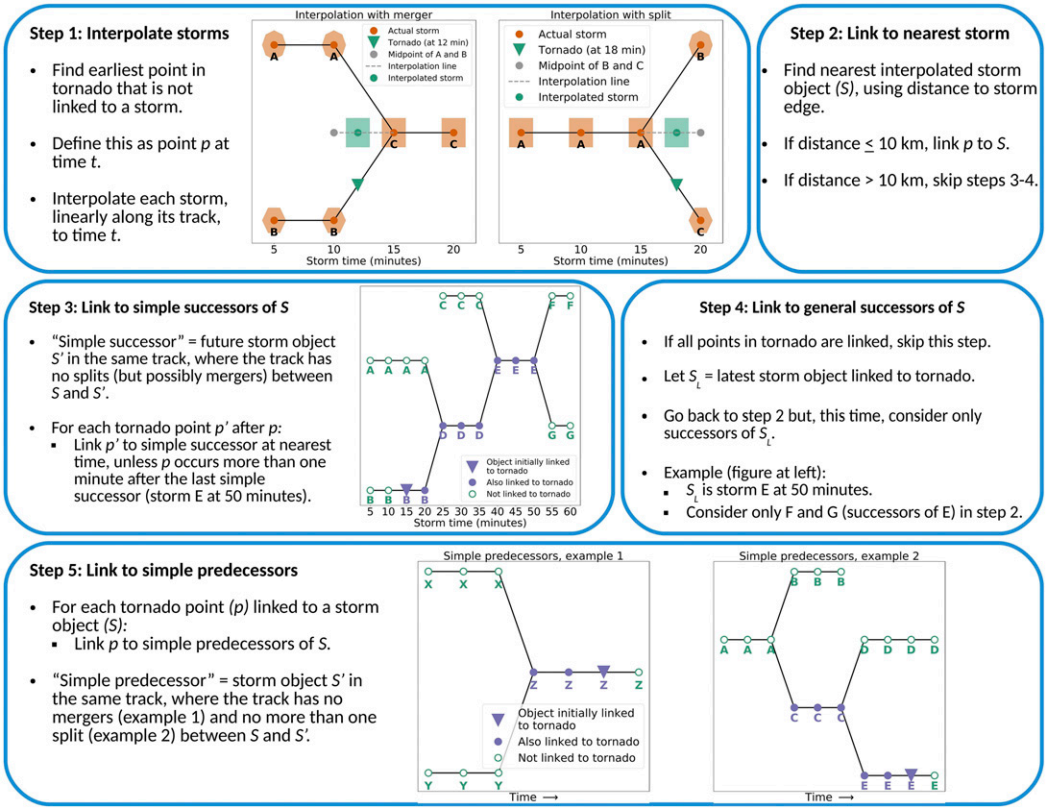

FIG. 4. Flowchart for tornado attribution. In step 1, each storm is interpolated to the time of the earliest unlinked point in the tornado p. The dark-orange/green dots are storm centers, and the surrounding light-orange/green polygons are storm outlines. The three cases for interpolation are explained in the main text. In step 2, p is linked to the nearest interpolated storm object S, as long as S is close enough. In step 3, p is also linked to simple successors of S. In step 4, if any point in the tornado is still unlinked, the algorithm returns to step 2 but considers only successors of the latest storm object linked to the tornado. In step 5, any tornado point that is already linked to a storm object is linked to simple predecessors of the given object.

thus splitting one actual track into two tracks with a 10-min gap. One common reason is that the storm drops below the echo-top threshold (supplemental section C) for one time step, leaving no storm object to track. Another is that, even though step 4 of the detection algorithm (supplemental section C) partly alleviates this, the storm center "jumps" erratically between time steps, causing the distance between successive centers to exceed the 9-km threshold.

The result is often two nearly collinear storm tracks with a small gap in the middle, like a piece of string that has been cut in half. The final tracking algorithm joins such pairs of tracks. The final algorithm is equivalent to the preliminary algorithm (Fig. 3), with four exceptions. First, the difference between t_1 and t_2 is two time steps (10 min). Second, the final algorithm is applied only to pairs of preliminary tracks in which one ends at t_1 and the other begins at t_2. Third, velocity estimates used in the final algorithm are third-order, rather than first-order, backward differences. Computing higher-order differences requires more computing time per track, but the final algorithm considers only a small number of preliminary tracks, which makes this computation feasible. Fourth, the final algorithm skips step 2 (linkage without extrapolation), because third-order velocity estimates are much less noisy than first-order estimates, which obviates the need for step 2 as a "second chance." Figures S3 and S4 of the online supplemental material show animated tracking output for the GridRad and MYRORSS data, respectively, on 26–27 April 2011.

c. *Tornado attribution*

The procedure shown in Fig. 4 is repeated for each tornado. It is more complex than most spatial linkage procedures, because the storm tracks include splits and mergers. Before step 1, the tornado is interpolated linearly to 1-min intervals between its start and end locations (the only locations included in NWS

tornado reports), yielding the "tornado points" mentioned in Fig. 4. In step 1, each storm is interpolated to the time of tornado point p, so that tornado-to-storm distances can be calculated. This interpolation uses the previous and next locations of each storm—by definition, at times t_1 and t_2, respectively. The three cases for storm interpolation are explained below for clarity:

1) If the storm undergoes a merger (shown in Fig. 4) between t_1 and t_2, interpolate from the midpoint of the two storm centers at t_1 to the actual storm center at t_2. Keep the storm boundary from t_2.
2) If the storm undergoes a split (shown in Fig. 4) between t_1 and t_2, interpolate from the actual storm center at t_1 to the midpoint of the two storm centers at t_2. Keep the storm boundary from t_1.
3) If the storm neither merges nor splits (not shown in Fig. 4), interpolate between the two actual storm centers. Keep the storm boundary from the time nearest the tornado point.

In step 2, the distance used is between the tornado point and the nearest storm edge. The 10-km threshold accounts for tornadoes in weak-echo regions. Each storm edge encloses an area with 40-dBZ echo top $\geq$ 4 km above sea level (section C), and smaller distance thresholds cause tornadoes in weak-echo regions to be unlinked. By visual inspection, all tornadoes >10 km from the nearest storm edge appear to be erroneous reports.

However, step 2 is not sufficient, because it does only spatial attribution (links to one storm object, which is one storm at one time). The ultimate goal of tornado attribution is to know, at each time, which storms are responsible for a tornado in the next hour. This requires temporal linkage, which is done by steps 3–5. Specifically, if a tornado is linked to storm object S in step 2, step 3 links the tornado to simple successors of S (future storm objects linked to S by any track without a split) that occur during the tornado's lifetime. In Fig. 4 this encodes the fact that storm B is tornadic from 15 to 20 min, storm D is tornadic from 25 to 35 min, and storm E is tornadic from 40 to 50 min. If necessary, step 4 links the tornado to nonsimple successors of S (those created by a split), because in such cases it is not obvious which of the two postsplit storms, if either, remains tornadic. In step 5, the tornado is linked to simple predecessors of S. This encodes the fact that certain storm objects are responsible for tornadoes in the future, even if they are not tornadic at their valid time. The definition of "simple predecessor," used in step 5, is explained below for clarity:

1) If storm C splits into storms D and E, then either D or E produces a tornado, the tornado is also attributed to C. See example 2 in Fig. 4.
2) If storm A splits into B and C, then C splits into D and E, then D or E produces a tornado, the tornado is *not* attributed to A. We assume that storm characteristics change enough over two splits that storm A only negligibly impacts the tornado potential of storm D or E. See example 2 in Fig. 4.
3) If storms X and Y merge into Z, then Z produces a tornado, the tornado is attributed to *neither* X nor Y. There is no obvious way to determine which of X and Y is primarily responsible for the characteristics of Z that led to tornadogenesis. See example 1 in Fig. 4.

After the procedure shown in Fig. 4, any storm object linked to a tornado in the next hour is labeled "yes"; all others are labeled "no." Among the storm objects labeled no, two types are removed from the dataset:

1) storms that merge into a tornadic successor (like X and Y in rule 3 above), because they cannot be confidently labeled yes or no (it is unclear which storm, if either, is responsible for tornadogenesis), and
2) storms with a successor in the next hour that passes within 10 km (the maximum tornado-to-storm distance) of the land boundary of the CONUS (NWS reports are generally not collected outside this area).

d. Creation of predictors

One data point (or example) for ML represents one storm object. The label (yes or no) indicates whether the storm is tornadic in the next hour, and the predictors are a storm-centered radar image and proximity sounding. Details are listed in Table 3. The storm-centered radar image comes from either MYRORSS or GridRad, and its horizontal center is the horizontal center of the storm object. The storm-centered radar image is on an equidistant grid with storm motion pointing to the right (in the $+x$ direction). Heights in the storm-centered grid are ground relative, whereas heights in the native (MYRORSS or GridRad) grid are sea level relative. We interpolate to ground relative because storms over high terrain have a lot of missing data near sea level, leading to poor CNN predictions. For both MYRORSS and GridRad, the storm-centered grid has a horizontal extent of 48 km $\times$ 48 km. Horizontal spacing of the storm-centered grid is chosen so that resolution is not lost during interpolation from the native grid. For example, the native grid for GridRad has 0.0208° spacing, leading to 2.31-km meridional spacing and at least 1.52-km zonal spacing everywhere south of the Canadian border. Thus, resolution is preserved by 1.5-km horizontal spacing in the storm-centered grid.

Note that the storm-centered radar image has three spatial dimensions for GridRad and two spatial dimensions

for MYRORSS (Table 3). Thus, the associated CNNs perform 3D and 2D convolution, respectively. The native MYRORSS dataset has 3D reflectivity and 2D azimuthal shear, so we tried training CNNs that perform 3D convolution for reflectivity and 2D convolution for azimuthal shear. However, the approach presented here led to better predictions.

To create proximity soundings, we use the following procedure for each example. Let the example be storm S at forecast time t. Note that proximity soundings are created separately for MYRORSS and GridRad examples.

1) Extrapolate S to time $t + 30$ min and find the nearest RAP grid point to the center of the extrapolated storm.[7] Call this grid point g.
2) Take the sounding from grid point g in the latest RAP analysis before t. This corresponds to previous-neighbor interpolation in time and nearest-neighbor interpolation in space.

Higher-order interpolation methods, such as linear and cubic, sometimes generate physical inconsistencies such as large supersaturations, because each variable in the sounding is interpolated separately. The simple method used preserves the entire sounding from one grid point at one time, which prevents such inconsistencies. Also, by using RAP data from at least 30 min before the extrapolated storm, convective contamination (where the sounding for storm S is influenced by storm S) is usually prevented.

Native RAP variables needed to create a sounding are u wind, v wind, temperature, and relative humidity. However, to train CNNs, we replace temperature in this set with virtual potential temperature θ_v and specific humidity. The θ_v is important because static stability is determined by its vertical profile alone (stable if θ_v increases with height and unstable otherwise), and specific humidity is important because it is the total mass concentration of water vapor. We interpolate all five variables to heights from 0 to 12 km AGL (Table 3). We use ground-relative heights here to prevent the use of underground height levels, which contain data extrapolated from the lowest atmospheric levels.

We split both MYRORSS and GridRad examples into training, validation, and testing data (Table 2). We leave a one-week gap between each pair of consecutive datasets, which ensures that the storms and synoptic patterns in one dataset are not temporally autocorrelated with those in another. The role of training data is to fit weights in the CNNs; the role of validation data is to fit hyperparameters, defined as ML settings that must be chosen before training and cannot be fit during training; and the role of testing data is to evaluate the final model on data independent of the training and validation.

TABLE 2. Training, validation, and testing periods for the MYRORSS and GridRad datasets.

Dataset	Time period
MYRORSS training	1 Jan 2005–24 Dec 2008
MYRORSS validation	1 Jan 2009–24 Dec 2010
MYRORSS testing	2011
GridRad training	1 Jan 2012–24 Dec 2014
GridRad validation	2015–18
GridRad testing	1 Jan 2011–24 Dec 2011

We normalize predictors via Eq. (2): x is the original value, $\bar{x}$ is the mean over training data, s is the standard deviation over training data, and z is the normalized value:

$$z = \frac{x - \bar{x}}{s}. \tag{2}$$

We apply the equation independently to each variable in each dataset (eight for MYRORSS examples and nine for GridRad examples; Table 3). It is crucial that $\bar{x}$ and s be computed with training data only. If validation/testing data were used, the normalized training data would contain information from the validation/testing data and the three sets would no longer be independent. Normalization ensures that all predictors have equal variance (1.0) in the training data, which prevents the CNN from unduly focusing on predictors with higher variance, which is often due to physical units. For example, in the GridRad training data, reflectivity has a variance of 212 dBZ^2 while vorticity has a variance of 3.0×10^{-7} s^{-2}.

4. Hyperparameter experiment (finding the best CNNs)

This section describes the experiment used to find the best CNN hyperparameters. The experiment takes the form of a grid search (section 11.4.3 of Goodfellow et al. 2016), which has the following procedure: 1) Decide the set of experimental hyperparameters and values to try for each (Table 4). 2) For each combination of values, train a model on the training data and evaluate it on the validation data. 3) Select the model that performs best on validation data. 4) Evaluate the selected model on testing data. Furthermore, all CNNs in this

[7] Note that $t + 30$ min is the midpoint of the 1-h-long prediction window. Extrapolation is done by assuming that the storm will maintain its current velocity, estimated by a third-order backward difference.

TABLE 3. Summary of processed input data ("images" = storm-centered radar images). One example for ML contains a proximity sounding and *either* an MYRORSS image or a GridRad image, with all variables listed in the rightmost column.

Dataset	Grid size	Horizontal spacing (km)	Heights (km AGL)	Variables
MYRORSS images	128 rows; 128 columns	0.375	—	Low-level azimuthal shear, midlevel azimuthal shear, and reflectivity at 1, 2, . . . , 12 km AGL
GridRad images	32 rows; 32 columns; 12 heights	1.5	1, 2, . . . , 12	Reflectivity, spectrum width, vorticity, and divergence
Proximity soundings	49 heights	—	0, 0.25, . . . , 12	u wind, v wind, relative humidity, specific humidity, and virtual potential temperature θ_v

work are trained with the Keras Python package (Chollet et al. 2020).

a. *Fixed hyperparameters (used for all CNNs)*

CNN training is subdivided into stages called epochs. In each epoch, multiple batches of training examples are presented to the CNN. After each batch, weights in the CNN are updated via the Adam optimizer (Kingma and Ba 2014), in an effort to minimize the loss [Eq. (1)]. After each epoch, the loss is computed for both training and validation data, which can be used to diagnose overfitting (when loss continues to decrease for the training data but begins to increase for the validation data). Specifically, the CNNs are trained for 100 epochs, with 32 batches per epoch and 1152 examples per batch. These numbers are chosen to be large enough that validation loss always converges to a minimum before training is complete. This occurs for every CNN in the experiment, usually well before the 50th epoch, at which point they begin to overfit. Thus, after training is complete, weights are restored to the epoch with minimum validation loss.

Examples in each training batch are drawn randomly from different time steps, which maximizes diversity within the batch—that is, ensures that most examples come from different storm cells and different synoptic situations—which reduces overfitting. Also, training data are undersampled so that each batch contains an equal number of tornadic and nontornadic examples. Without undersampling, the CNNs would have little incentive to predict probabilities $\gg 0$, because the small fraction of tornadic examples would have little effect on the loss. Undersampling is used only for training. When a CNN is evaluated on the validation or testing data, undersampling is not used, so the evaluation scores fully reflect the difficulty of predicting a rare event.

Complete details on CNN architecture are shown in section D of the online supplemental material. Note that the CNNs perform convolution and pooling over both radar images and proximity soundings.

b. *Experimental hyperparameters*

The experiment, conducted separately for the MYRORSS and GridRad models, involves four hyperparameters (Table 4). All hyperparameters control overfitting, which is a serious problem for tornado prediction, due to uncertainty in the predictors (radar and NWP data) and labels (tornado reports). The first hyperparameter is the number of dense layers, which is the main control on the number of CNN weights. The number of convolutional layers is fixed and chosen to yield a final domain size of 4–6 grid cells in each direction (Figs. 1 and 2). We have found that smaller domains (e.g., 2 grid cells per direction) do not contain enough spatial information to make skillful predictions, while larger domains (e.g., 10 grid cells per direction) lead to too many weights in the dense layers, which makes training more computationally expensive and increases the risk of overfitting.

The second hyperparameter is dropout rate d. For each training example and each dense layer, dropout (Hinton et al. 2012) randomly zeroes out fraction d of the layer's outputs.[8] This forces the weights in a given layer to evolve more independently, which reduces overfitting. Dropout is used only during training and for all dense layers except the last (the last dense layer has only one output, which is tornado probability). The third hyperparameter is L_2 weight λ, which controls the strength of L_2 regularization (Hoerl and Kennard 1970, 1988). In L_2 regularization, the term λ SSW is added to the loss function [Eq. (1)] during training, where SSW is the sum of squared weights in all convolutional layers.[9] This encourages the models to learn smaller weights. Large weights make the models unstable, causing sharp

[8] Dropout multiplies the remaining outputs by $1/d$ so that the sum remains the same.

[9] Regularization can also be applied to the dense layers, but, since we already use dropout for the dense layers, we believe that applying both types of regularization in tandem would be superfluous.

TABLE 4. Values attempted for hyperparameter experiment.

Hyperparameter	Values
Dropout rate	0.250, 0.375, 0.500, 0.625, and 0.750
L_2 weight	10^{-3}, $10^{-2.5}$, 10^{-2}, $10^{-1.5}$, and 10^{-1}
Data augmentation	On, off
No. of dense layers	1, 2

changes in the output (prediction) for small changes in the inputs (predictors).

The fourth hyperparameter is data augmentation (section 5.2.5 of Chollet 2018), the practice of applying small perturbations to the predictors while assuming that the label (tornadic or not) remains the same. This reduces overfitting by preventing the CNN from learning relationships that are too specific. We apply 17 perturbations to the radar image for each example:

- horizontal rotations (about the z axis) of +15°, −15°, +30°, and −30°;
- horizontal translations of three grid cells in eight directions spaced equally from 0° to 315°; and
- five additions of Gaussian noise, each with a standard deviation of 0.1 (since all predictors are normalized to the same scale [Eq. (2)], the magnitude of this effect is the same for all predictors).

The perturbations are applied separately, turning one example into 18 (the original example plus 17 perturbed ones). When data augmentation is turned on, each training batch contains 64 real examples (32 tornadic and 32 nontornadic), perturbed 17 times each to create a batch of 1152. When data augmentation is turned off, each training batch contains 1152 real examples (576 tornadic and 576 nontornadic). The exact perturbations were decided by a previous experiment (not shown).

After training, we rank the CNNs by AUC (Metz 1978) on the validation data. We also compute probability of detection (POD), false-alarm ratio (FAR), frequency bias, and CSI, defined in Eqs. (1)–(4) of Roebber (2009). All scores other than AUC are based on deterministic predictions, whereas the CNNs output probabilities. To convert from probabilistic to deterministic, we use the probability threshold that maximizes validation CSI (this threshold is different for each CNN). Thus, the probability threshold is treated as a hyperparameter: chosen on the validation data, then frozen.

5. Results

Figures E1 and E2 in the online supplemental material show performance of the 100 GridRad models on validation data. The best GridRad model uses data augmentation and has a dropout rate of 0.5 (the median value attempted), L_2 weight of 10^{-3} (minimum attempted), and two dense layers (maximum attempted). In general, the GridRad models perform best with two dense layers and data augmentation turned on. The other two hyperparameters, L_2 weight and dropout rate, have a much smaller effect on performance. This suggests that data augmentation is the most effective regularization method for GridRad data. Although data augmentation increases effective sample size much less than it increases nominal sample size (because the 17 perturbed examples created from each original example are highly correlated), it clearly increases effective sample size enough to improve predictions on independent data. Figures E3 and E4 in the online supplemental material show performance of the 100 MYRORSS models on validation data. The best MYRORSS model uses data augmentation and has a dropout rate of 0.75 (maximum attempted), L_2 weight of $10^{-2.5}$ (second-lowest attempted), and two dense layers (maximum attempted). As for GridRad, the MYRORSS models generally perform best with two dense layers and data augmentation turned on, but here the difference is less dramatic.

Figures 5a and 5b shows performance of the best MYRORSS and GridRad models on testing data. Each point in the receiver-operating-characteristic (ROC) curve or performance diagram (Roebber 2009) corresponds to one probability threshold. Event frequency (percentage of storms that are tornadic in the next hour) is 3.52% in the GridRad testing data and 0.24% in the MYRORSS testing data. AUC is significantly higher for the MYRORSS model, likely because the MYRORSS dataset contains more trivial correct nulls. Specifically, because the MYRORSS data are available for every day, while the GridRad data are available primarily for days with at least one tornado, the MYRORSS data contain more easy-to-predict nontornadic storms (e.g., storms in the middle of winter and other environments that are very nonconducive to tornadoes). AUC > 0.9 for both models, which is generally considered the threshold for "excellent" performance (e.g., Luna-Herrera et al. 2003; Muller et al. 2005; Mehdi et al. 2011). Because the ROC curve is insensitive to event frequency, this threshold can be used across prediction tasks. No such threshold exists for the performance diagram, which is highly sensitive to event frequency, as shown by the difference between MYRORSS and GridRad models. The best point in the performance diagram is the top right (where CSI = 1.0), and the worst point is the bottom left (where CSI = 0.0). CSI is sensitive to event frequency, because a high CSI requires a high POD and low FAR. In other words, to achieve a high CSI, the model must correctly predict a large fraction of events without producing

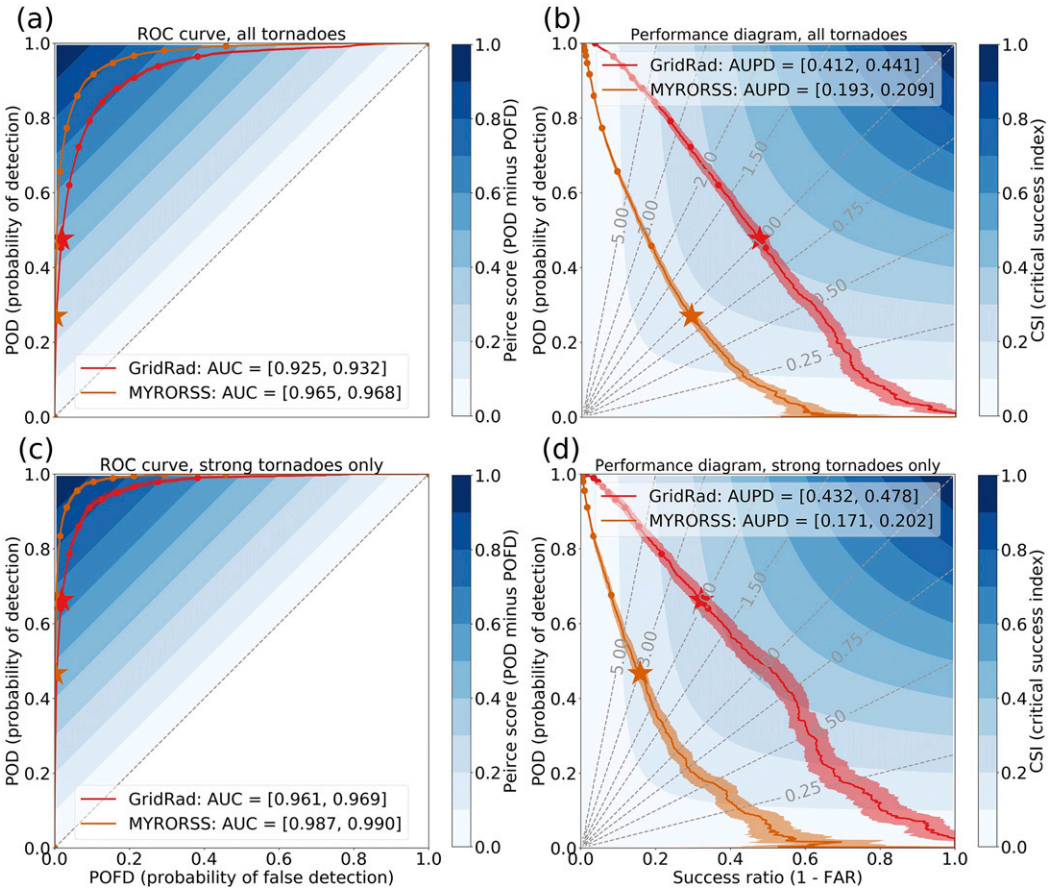

FIG. 5. Performance of MYRORSS and GridRad models on testing data. Dark lines show the mean, and light shading shows the 95% confidence interval, determined by bootstrapping 1000 times. The star corresponds to the probability threshold that maximizes CSI on the validation data. Dots correspond to probability thresholds p^* of 0.0, 0.1, . . . , 1.0. Shown are (left) the ROC curve for (a) all tornadoes and (c) strong tornadoes only (AUC is area under the curve, given as a 95% confidence interval; p^* increases from 0.0 at top right to 1.0 at bottom left) and (right) the performance diagram for (b) all tornadoes and (d) strong tornadoes only (AUPD is area under the curve, given as a 95% confidence interval; p^* increases from 0.0 at top left to 1.0 at bottom right).

a large number of false alarms, which becomes more difficult as the event frequency decreases. Testing and validation AUC for the MYRORSS model are approximately equal (cf. Fig. 5 with Fig. E3 in the online supplemental material), but for the GridRad model, testing AUC is ~0.04 higher than validation AUC (cf. Fig. 5 with Fig. E1 in the online supplemental material). This suggests that perhaps 2011 is an abnormally easy year for the GridRad model, which is a slight caveat. Why 2011 is not abnormally easy for the MYRORSS model as well requires future investigation.

The stars in Fig. 5 show the chosen probability threshold (that which maximizes validation CSI) for each model, and Table 5 shows contingency tables created with this threshold. This threshold nearly maximizes CSI on the testing data as well (shown in the performance diagram), but it leads to an unacceptably low POD (0.27 for MYRORSS and 0.48 for GridRad) for a costly event such as tornadoes. Considering that false negatives have a much greater cost than false positives, if one could assign numerical values to these costs, one could choose the probability threshold that minimizes cost. This would result in a lower threshold, causing the star to move toward the top right of the ROC curve and top left of the performance diagram—yielding a lower CSI and higher POD, POFD, FAR, and frequency bias.

Figures 5c and 5d contain testing results for strong tornadoes only (EF2+), which suffer from less underreporting bias, as discussed at the end of section 3a. Table 6 shows the corresponding contingency table for each model. The testing sets used in Figs. 5c and 5d are created by simply removing EF0 and EF1 tornadoes from those used in Figs. 5a and 5b. This decreases the GridRad event frequency to 1.36% and the MYRORSS

TABLE 5. Contingency tables for MYRORSS and GridRad models on all testing data. Probabilities (raw CNN output) are converted to deterministic forecasts using the CSI-maximizing probability threshold, 88.95% for the GridRad model and 95.18% for the MYRORSS model.

Forecast	Observation: Yes	Observation: No
GridRad model		
Yes	2193	2515
No	2418	123 829
MYRORSS model		
Yes	2642	6487
No	7221	4 143 227

TABLE 6. As in Table 5, but for strong (EF2+) tornadoes only.

Forecast	Observation: Yes	Observation: No
GridRad model		
Yes	1139	2455
No	601	123 889
MYRORSS model		
Yes	1254	6449
No	1342	4 143 265

event frequency to 0.06%. Assuming that model skill does not vary with tornado strength, one would expect this change to yield a similar ROC curve and worse performance diagram. However, the ROC curve improves significantly, and the performance diagram remains roughly the same. This suggests that model skill improves with tornado strength, which is likely caused by two factors. First, strong tornadoes are labeled more accurately, since they suffer from less underreporting bias; second, strong tornadoes generally have clearer signatures, especially in the radar image.

As mentioned in the introduction, an ML model called ProbSevere is currently used annually in the HWT, where it has received positive feedback from forecasters. ProbSevere forecasts the probability of any severe weather on a storm-by-storm basis. As shown in Fig. 6 of Cintineo et al. (2018), ProbSevere achieves a CSI of 0.27 and POD of 0.55 with 4.94% event frequency. Figure 5 suggests that the GridRad model could achieve the same CSI with a lower event frequency (3.52%) and higher POD (~0.7). Although the comparison is not completely fair (ProbSevere predicts all severe weather and uses a real-time version of MYRORSS, which is less quality controlled), we believe the evidence is sufficient to suggest that our models would be useful in an operational setting.

Figures 6 and 7 break down the testing performance of the two models by time. As the number of tornadic examples increases, POD generally increases while FAR generally decreases, causing CSI to increase [$\mathrm{CSI}^{-1} = \mathrm{POD}^{-1} + (1 - \mathrm{FAR})^{-1} - 1$]. In terms of AUC and CSI, both models perform best from April to July and during the afternoon and evening (the hours of 1800–0500 UTC, which end at 0559:59 UTC), when most tornadoes occur. In terms of AUC only, performance is excellent (>0.9) for most hours and months. A notable exception is the GridRad model during winter (December, January, and February). These months have the fewest examples (2644, 975, and 3029, respectively), and nearly the fewest tornadic examples (50, 18, and 82, respectively), of all months in the GridRad testing data. This is also true for the training data, which means that relationships learned during training were controlled mostly by the other (nonwinter) months, at the cost of performance in winter. The MYRORSS model also performs worst in winter. Both models have peaks in CSI for November and the hour of 1000 UTC (1000:00–1059:59 UTC). The peak in November can be explained by a peak in tornadic examples, but the one at 1000 UTC cannot. This peak is generally not significant (i.e., the confidence interval for 1000 UTC generally overlaps with confidence intervals for the adjacent hours), so it may be due merely to sampling error.

Figures 8 and 9 break down the testing performance of the two models by location, into 100-km grid cells. Grid cells with no tornadic examples are not shown, because this causes the scores to degenerate. Most examples and most tornadic examples occur in the southeast quadrant of the CONUS, with a secondary maximum in tornadic examples in the southern Great Plains. In most areas, the GridRad model has an AUC > 0.8 (considered "good" in the same papers that define 0.9 as "excellent") and the MYRORSS model has an AUC > 0.9. Areas with lower AUC generally have very few tornadic examples, thus very few examples with which to compute POD (one of the inputs to AUC), thus a large sampling error. For examples of this effect, see the three grid cells on the northern border of Kentucky in Fig. 8. Both models perform poorly west of the Rockies, which is likely due to a combination of few tornadoes and poor radar coverage. Complex orography and a less dense radar network ensure that many areas are covered by only one or two radars, leading to poor estimates of all variables. Many areas with few tornadoes in the GridRad data experience many tornadoes both climatologically (https://www.spc.noaa.gov/wcm/climo/alltorn.png) and in the MYRORSS data (Fig. 9). This suggests that expanding the GridRad dataset to include tornadic storms in these areas—such as eastern Colorado, the northern Great

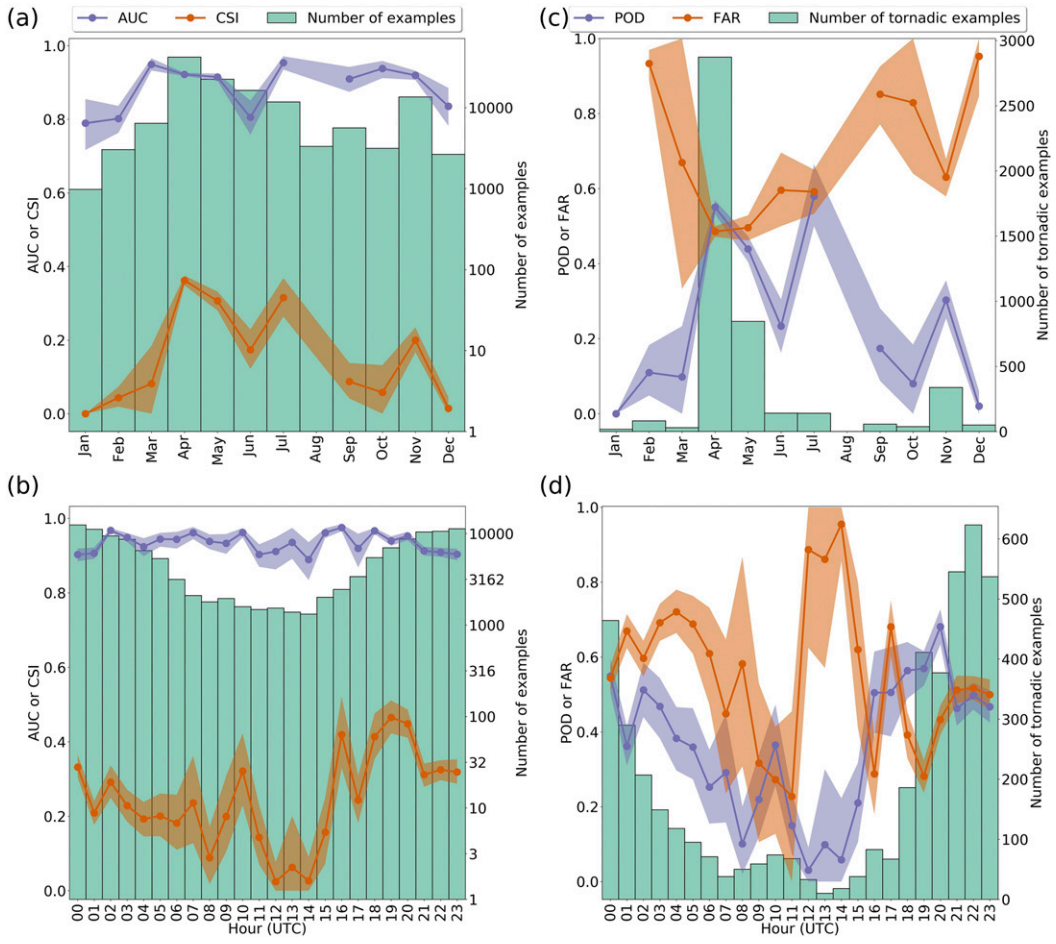

FIG. 6. Monthly and hourly performance of the GridRad model on testing data. Dark lines show the mean, and light shading shows the 95% confidence interval, determined by bootstrapping 1000 times. Shown are (left) AUC, CSI, and number of examples and (right) POD, FAR, and number of tornadic examples (or "events") (a),(c) by month and (b),(d) by hour.

Plains, and the Ohio River valley—would greatly improve performance there.

Figures 10–13 show extreme cases: the 100 best hits, worst false alarms, worst misses, and best correct nulls in the testing data (Table 7). These sets are constructed separately for the MYRORSS and GridRad models, with one constraint: a set cannot contain multiple examples from the same storm.[10] If the best hits or worst false alarms contain multiple time steps from one storm, only that with the highest probability is kept. Similarly, if the worst misses or best correct nulls contain multiple time steps from one storm, only that with the lowest probability is kept. Thus, each set contains examples from 100 different storm cells, which increases the diversity within each set by decreasing autocorrelation. The composites shown in Figs. 10–13 are created via probability-matched means (PMM; Ebert 2001), which retains spatial structure better than taking the simple mean at each grid point.

Figure 10 shows the radar fields for extreme GridRad cases. For the sake of brevity, only 3 of the 12 heights are shown: 2, 6, and 10 km AGL. These will henceforth be called low level, midlevel, and upper level. The "best hits" composite contains high reflectivity throughout the column, strong low-level convergence, and strong upper-level divergence, suggesting a strong updraft; high vorticity and spectrum width throughout the column, suggesting a strong mesocyclone; and a hook echo in the low-level reflectivity field, which suggests the presence of a rear-flank downdraft (RFD) and is a signature often associated with tornadoes

[10] Two storm objects are considered to be part of the same storm if their tracks are connected at all, by any number of splits or mergers.

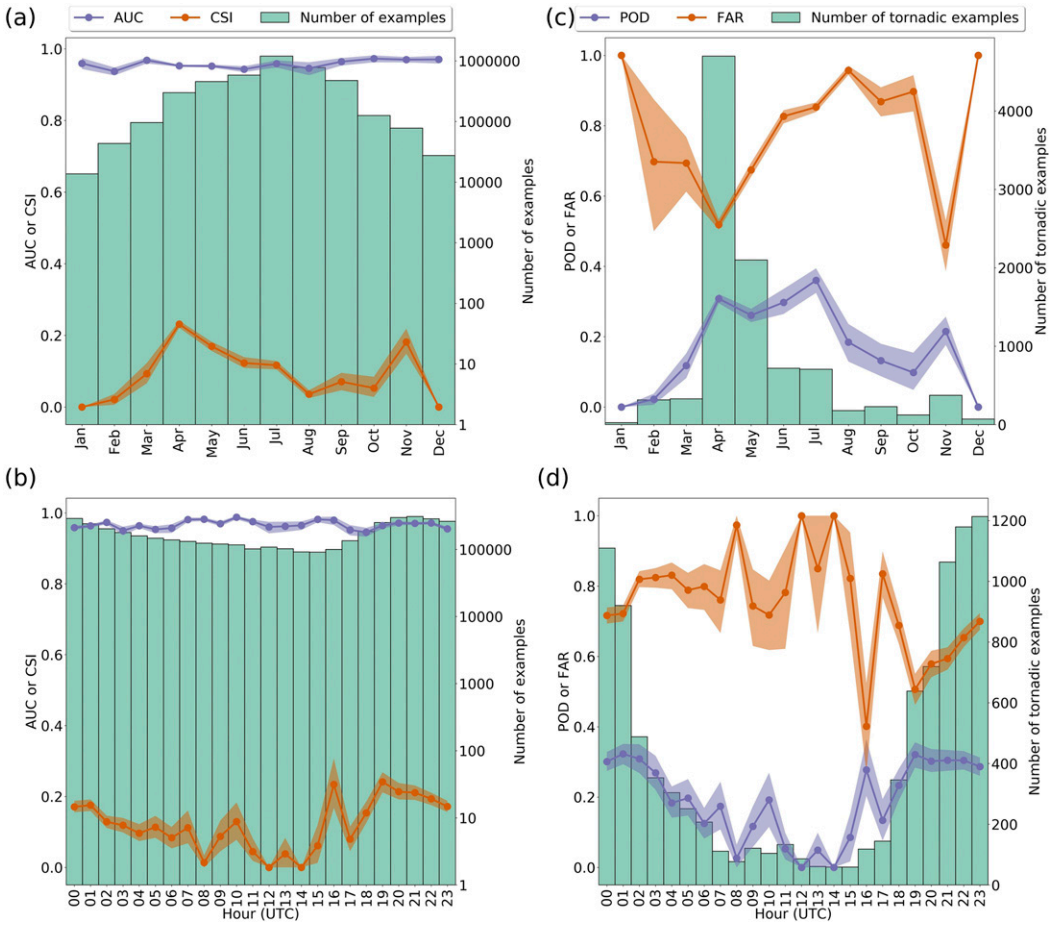

FIG. 7. As in Fig. 6, but for MYRORSS.

(Davies-Jones et al. 2001; Rhyzkov et al. 2005; Markowski and Richardson 2009, 2014). Overall, the storm appears to be a supercell, the type responsible for most tornadoes and especially EF2 + tornadoes (Table 2 of Smith et al. 2012). Also, the storm appears to be discrete (isolated from other storms), at least on its right and rear flanks. Although weak tornadoes are often produced by supercells in clusters, strong tornadoes are generally produced by discrete supercells (Table 2 of Smith et al. 2012).

The worst false alarms look very similar to the best hits, with one of the few notable differences being the absence of a low-level hook echo. We considered the possibility that these storms are not really false alarms—that is, produced unreported tornadoes—which is a known issue with the NWS storm reports, as discussed in section 3a. However, this explanation is implausible, because (i) most of these storms occur in the evening and near towns; (ii) 76 of the 100 storms are associated with an NWS tornado warning, and the NWS generally seeks out reports to verify warnings that they have issued. Rather, we believe that the similarity between the best hits and worst false alarms is due mainly to two factors. First, tornadoes are usually on the order of 100 m across, so GridRad and MYRORSS have insufficient resolution to represent tornadoes and other relevant circulations. Second, "tornado" and "nontornado" are discrete labels applied to a continuous spectrum of phenomena. Some funnel clouds very nearly reach the surface, while some tornadoes touch down for only a few seconds and produce minimal damage, but they are still labeled tornado and nontornado, respectively, which causes them to be treated as completely disparate phenomena.

The worst misses are shallow and elongated storms with smaller values of reflectivity, spectrum width, vorticity, low-level convergence, and upper-level divergence. By inspection of the 100 individual storms, we have found that most either (i) are weak storms that subsequently intensify rapidly, allowing them to produce tornadoes in the next hour, or (ii) are embedded in a quasi-linear convective system (QLCS). The second failure mode is more common than the first and is well known to

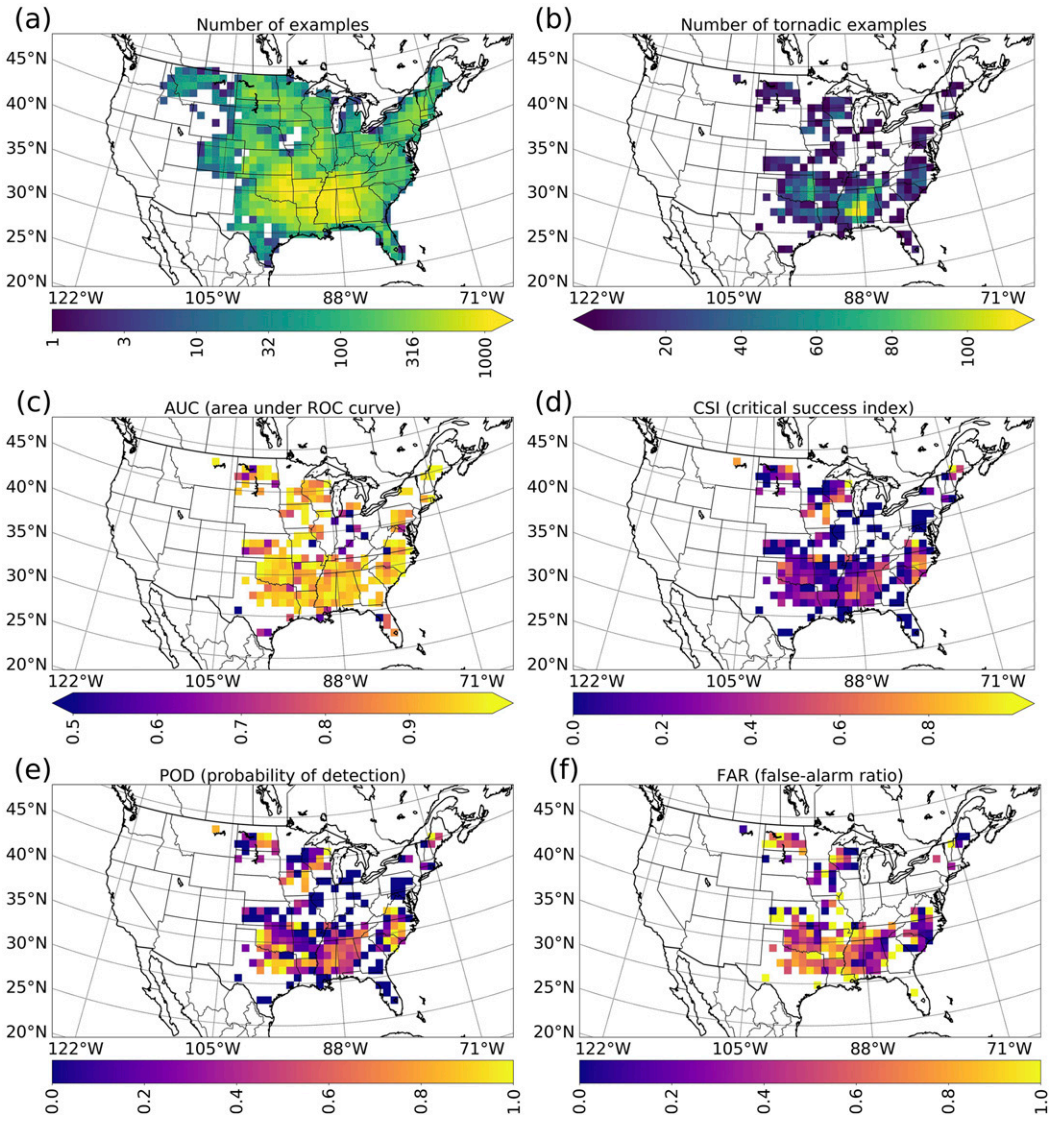

FIG. 8. Regional performance of the GridRad model on testing data: (a) number of examples, (b) number of tornadic examples (or "events"), (c) AUC, (d) CSI, (e) POD, and (f) FAR. Each grid cell is 100 km × 100 km.

meteorologists (Brotzge et al. 2013; Anderson-Frey et al. 2016). Note that the best correct nulls are generally weak and disorganized. By inspection of the 100 individual storms, we have found that most are nondominant cells in a mesoscale convective system (MCS). Also, 19 of the 100 storms occur in the outer rainbands of Hurricane Irene.

Figure 11 shows the radar fields for extreme MYRORSS cases. Overall, each composite is very similar to its counterpart in the GridRad data, except that the best hits for MYRORSS contain a much fainter hook echo. Conclusions based on inspecting individual storms also hold for the MYRORSS data. First, false alarms generally occur in the evening and near towns, and 47 of the 100 are associated with NWS tornado warnings, which suggests that most of these storms are truly nontornadic. Second, the worst misses are mostly QLCS cells, with some early-stage supercells. Third, the best correct nulls are mostly nondominant MCS cells.

Figure 12 shows the soundings for extreme GridRad cases. For all composites other than the best correct nulls, the soundings are much more similar than the radar images, which suggests that the radar images are generally more important for prediction. All three soundings have a slight surface-based temperature inversion, above which the profile is conditionally

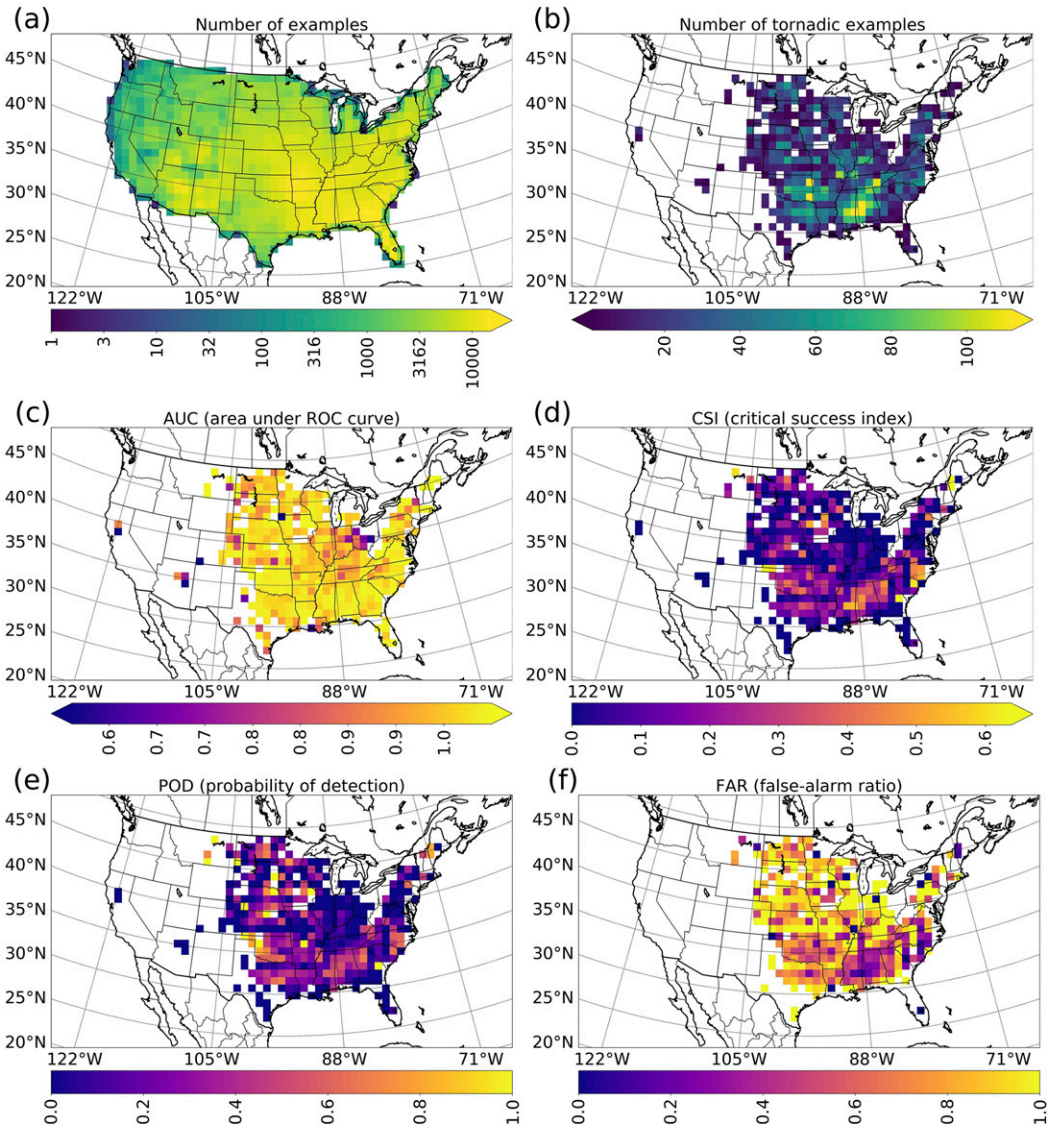

FIG. 9. As in Fig. 8, but for MYRORSS.

unstable up to the high troposphere; a dry nose around 700 hPa; and a strongly veering wind profile, with easterly surface winds veering sharply to southerly in the lowest 100 hPa and then to westerly in the high troposphere. This corresponds to high wind shear, which is crucially favorable for tornadoes (Markowski and Richardson 2009, 2014; Anderson-Frey et al. 2016). The surface-based inversion is usually caused by other storms that have recently moved over the area, as many storms occur in larger-scale convective systems. The main difference among the three soundings is that the worst misses have lower near-surface temperature and humidity, yielding less instability, and weaker winds aloft, yielding less wind shear. Both of these relationships are consistent with previous studies on the difference between supercell and QLCS environments (e.g., Tables 1–2 of Thompson et al. 2012). Compared to the other three composites, the best correct nulls have a stronger temperature inversion and much less low-level wind shear. By inspection of the 100 individual storms, many occur near fronts, causing the wind shift between 700 and 800 mb in the composite sounding.

Figure 13 shows the soundings for extreme MYRORSS cases. Each composite except the best correct nulls is very similar to its counterpart in the GridRad data. For the best correct nulls, the MYRORSS sounding has a much stronger temperature inversion, and much less near-surface moisture, than the GridRad sounding. The difference occurs because the MYRORSS dataset contains

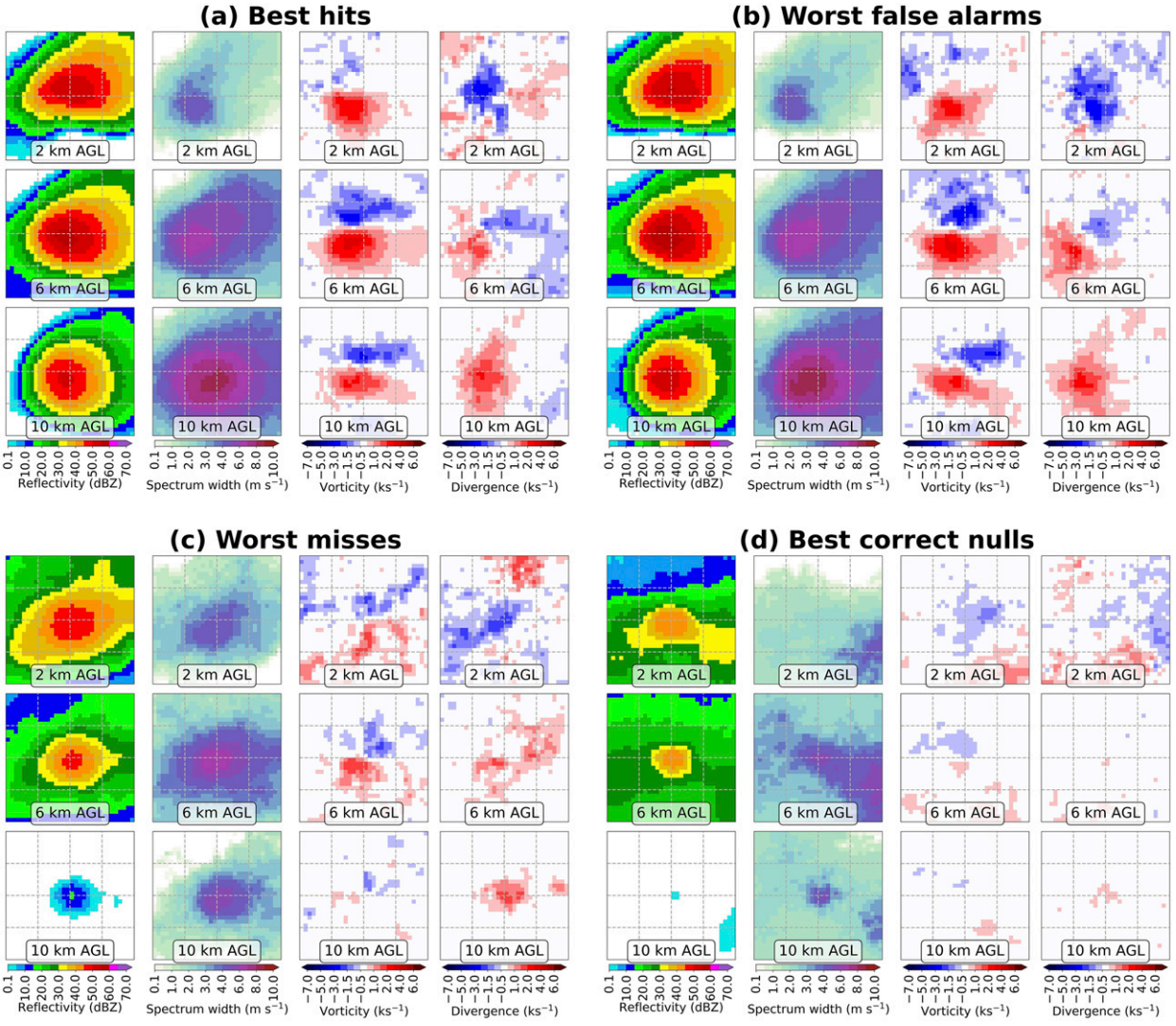

FIG. 10. Radar fields for extreme GridRad cases. Storm motion points to the right. Each image is a PMM composite over 100 examples.

many more nontornadic storms than GridRad, allowing for more extreme nontornadic cases.

6. Summary and future work

We used convolutional neural networks, a type of deep-learning method, to predict the probability that a storm will be tornadic in the next hour. The predictors were a proximity sounding and storm-centered radar image, the latter from either the MYRORSS or GridRad dataset. For both MYRORSS and GridRad, CNNs performed best when trained with data augmentation, the practice of applying small perturbations to the predictor fields while assuming that the label (tornadic or nontornadic) remains the same. When evaluated on independent testing data, the MYRORSS model achieved an AUC of 0.97 and CSI of 0.17, while the GridRad model achieved an AUC of 0.93 and CSI of 0.31. The difference in AUC is caused by a greater number of trivial nulls (easy-to-predict nontornadic storms) in the MYRORSS dataset, while the difference in CSI is caused by a lower event frequency in the MYRORSS dataset. Specifically, testing data for MYRORSS and GridRad have event frequencies of 0.24% and 3.52%, respectively. Results for strong tornadoes (EF2+) are slightly better than for all tornadoes, which suggests that model skill increases with tornado strength. Comparison with ProbSevere, an ML model currently used for operational severe weather prediction, suggests that our models would be useful operationally.

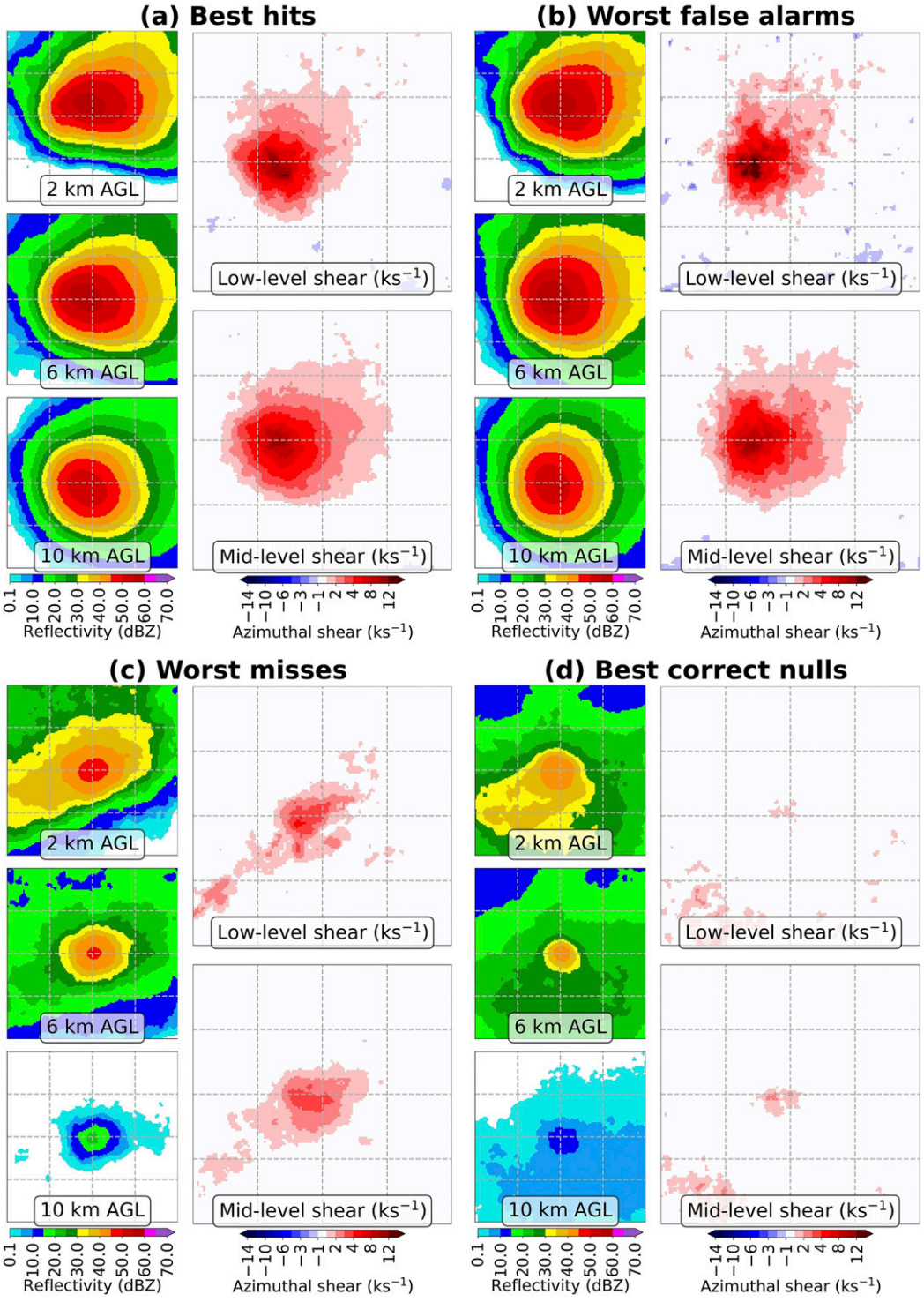

FIG. 11. As in Fig. 10, but for MYRORSS.

To better understand the models, we plotted storms yielding the best and worst predictions. For both the MYRORSS and GridRad models, the best hits are tornadic supercells; the worst false alarms are nontornadic supercells; the worst misses are mostly cells in quasi-linear convective systems, whereas some are early-stage supercells that subsequently undergo rapid intensification and tornadogenesis; and the best correct nulls are

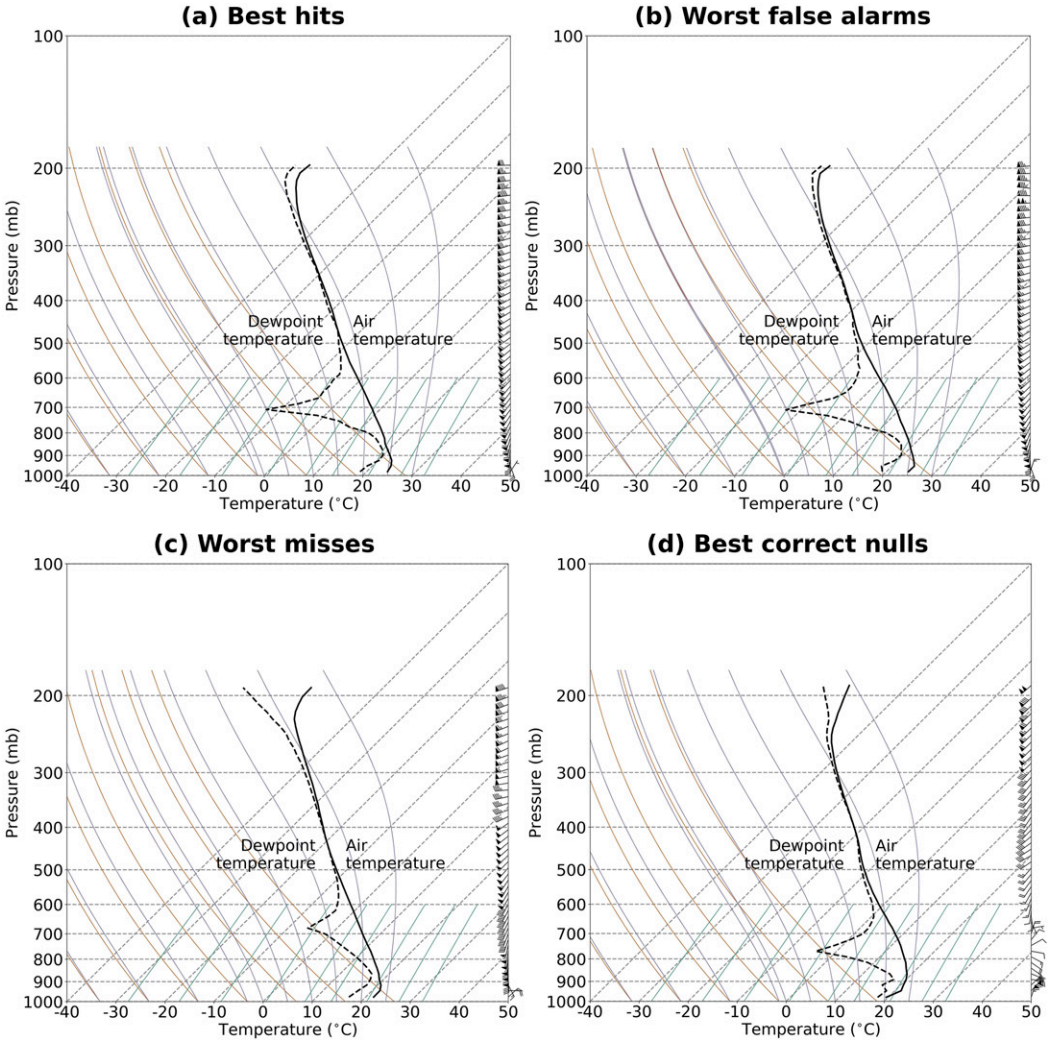

FIG. 12. Soundings for extreme GridRad cases. Each sounding is a PMM composite over 100 examples. The lowest data point is at the surface, and the spacing between subsequent points is 250 m.

mostly nondominant cells in mesoscale convective systems. Training models with both MYRORSS and GridRad data demonstrates the generalizability of our methods and results, especially patterns leading to the best and worst predictions. These patterns are very similar for the two models, even though they use different radar datasets and have quite different architectures (the MYRORSS model has five layers that perform 2D convolution, and the GridRad model has three layers that perform 3D convolution). Future work will use specialized ML-interpretation methods, such as those discussed in McGovern et al. (2019), to compare physical relationships learned by the two models.

In the future we also plan to adapt the models developed herein for an operational setting such as the HWT, where they could be evaluated in real time by forecasters. Although model development (training, validation, and testing) is computationally expensive, applying either trained CNN in real time takes ~5 min per radar time step on a desktop computer, including all preprocessing. This could easily be shortened by parallelizing across multiple cores and optimizing the code that creates proximity soundings, which takes more than half of the ~5 min. Radar data would come from the Multi-Radar Multi-Sensor (MRMS; Smith et al. 2016) dataset, the real-time version of MYRORSS, which is used by the ProbSevere model. The two datasets may be similar enough that the MYRORSS-trained model could be applied directly to MRMS data without retraining, but this will be investigated. Along with the model's predictions, we plan to show ML-interpretation output, such as saliency or class-activation maps (McGovern et al. 2019).

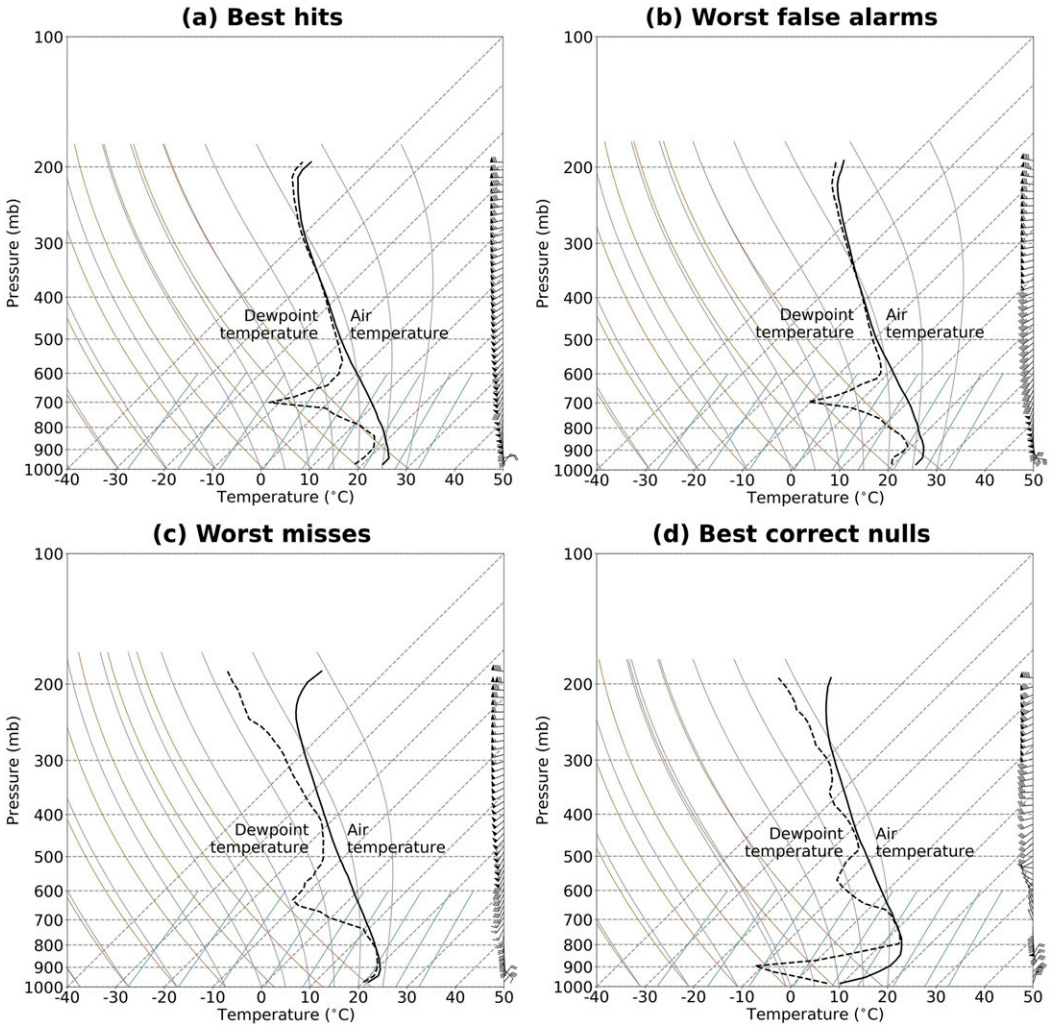

FIG. 13. As in Fig. 12, but for MYRORSS.

We believe that explaining the model's predictions in this way would increase forecaster trust and help them to identify failure modes (e.g., cases where the interpretation maps are physically unrealistic).

Future work will also include the following activities. The first is developing spatially and temporally dependent CNNs to improve predictions for spatial regions, times of day, and times of year that are currently poor. However, most of these regions and times have few tornadic examples with which to train a model, so performance improvements may be marginal. The second is expanding the scope of near-storm environment data used by the models. Instead of proximity soundings, we hope to train with full 3D and 4D data, which might allow the models to better capture relevant mesoscale features. The last two are improving the prediction of

TABLE 7. Definitions of extreme cases. "Probability" is the next-hour tornado probability forecast by the CNN. "Mean GridRad probability" is the mean forecast probability from the GridRad model over the 100 cases, and likewise for MYRORSS.

Set	Definition	Mean GridRad probability	Mean MYRORSS probability
Best hits	Tornadic examples with the *highest* probabilities	99.2%	99.6%
Worst false alarms	*Non*tornadic examples with the *highest* probabilities	98.8%	99.6%
Worst misses	Tornadic examples with the *lowest* probabilities	8.6%	11.9%
Best correct nulls	*Non*tornadic examples with the *lowest* probabilities	0.004%	0.04%

QLCS tornadoes and finding new ways to alleviate the rare-event problem so that a better balance of POD and FAR can be achieved.

Acknowledgments. Funding was provided by the National Science Foundation (Grant EAGER NSF AGS 1802267) and NOAA/Office of Oceanic and Atmospheric Research (NOAA–University of Oklahoma Cooperative Agreement NA16OAR4320115), U.S. Department of Commerce. Computing was done at the University of Oklahoma (OU) Supercomputing Center for Education and Research (OSCER), provided by OU, and Cheyenne (https://doi.org/10.5065/D6RX99HX), provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation. All plots were generated with matplotlib, version 3.1.1 (Hunter 2007). The National Center for Atmospheric Research (NCAR) is sponsored by the National Science Foundation.

REFERENCES

Adrianto, I., T. 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.

Anderson, C., C. Wikle, Q. Zhou, and J. Royle, 2007: Population influences on tornado reports in the United States. *Wea. Forecasting*, **22**, 571–579, https://doi.org/10.1175/WAF997.1.

Anderson-Frey, A., Y. Richardson, A. Dean, R. Thompson, and B. Smith, 2016: Investigation of near-storm environments for tornado events and warnings. *Wea. Forecasting*, **31**, 1771–1790, https://doi.org/10.1175/WAF-D-16-0046.1.

Benjamin, S., 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.

——, 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.

Bolton, T., and L. Zanna, 2019: Applications of deep learning to ocean data inference and subgrid parameterization. *J. Adv. Model. Earth Syst.*, **11**, 376–399, https://doi.org/10.1029/2018MS001472.

Bowman, K., and C. Homeyer, 2017: GridRad—Three-dimensional gridded NEXRAD WSR-88D radar data. NCAR Computational and Information Systems Laboratory Research Data Archive, accessed 11 June 2020, http://rda.ucar.edu/datasets/ds841.0/.

Brooks, H., and J. Correia, 2018: Long-term performance metrics for National Weather Service tornado warnings. *Wea. Forecasting*, **33**, 1501–1511, https://doi.org/10.1175/WAF-D-18-0120.1.

Brotzge, J., S. Nelson, R. Thompson, and B. Smith, 2013: Tornado probability of detection and lead time as a function of convective mode and environmental parameters. *Wea. Forecasting*, **28**, 1261–1276, https://doi.org/10.1175/WAF-D-12-00119.1.

Chollet, F., 2018: *Deep Learning with Python*. Manning, 361 pp.

——, and Coauthors, 2020: Keras. GitHub, https://github.com/fchollet/keras.

Cintineo, J., M. Pavolonis, 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., and Coauthors, 2012: An overview of the 2010 Hazardous Weather Testbed Experimental Forecast Program Spring Experiment. *Bull. Amer. Meteor. Soc.*, **93**, 55–74, https://doi.org/10.1175/BAMS-D-11-00040.1.

Crum, T., and R. 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.

Davies-Jones, R., R. Trapp, and H. Bluestein, 2001: Tornadoes and tornadic storms. *Severe Convective Storms*, C. Doswell, Ed., Amer. Meteor. Soc., 167–222.

Dieleman, S., K. Willett, and J. Dambre, 2015: Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Mon. Not. Roy. Astron. Soc.*, **450**, 1441–1459, https://doi.org/10.1093/MNRAS/STV632.

Doswell, C., A. Moller, and H. Brooks, 1999: Storm spotting and public awareness since the first tornado forecasts of 1948. *Wea. Forecasting*, **14**, 544–557, https://doi.org/10.1175/1520-0434(1999)014<0544:SSAPAS>2.0.CO;2.

Ebert, E., 2001: Ability of a poor man's ensemble to predict the probability and distribution of precipitation. *Mon. Wea. Rev.*, **129**, 2461–2480, https://doi.org/10.1175/1520-0493(2001)129<2461:AOAPMS>2.0.CO;2.

Elsner, J., L. Michaels, K. Scheitlin, and I. Elsner, 2013: The decreasing population bias in tornado reports across the central plains. *Wea. Climate Soc.*, **5**, 221–232, https://doi.org/10.1175/WCAS-D-12-00040.1.

Gagne, D., A. McGovern, J. Basara, and R. Brown, 2012: Tornadic supercell environments analyzed using surface and reanalysis data: A spatiotemporal relational data-mining approach. *J. Appl. Meteor. Climatol.*, **51**, 2203–2217, https://doi.org/10.1175/JAMC-D-11-060.1.

——, S. Haupt, and D. Nychka, 2019: Interpretable deep learning for spatial analysis of severe hailstorms. *Mon. Wea. Rev.*, **147**, 2827–2845, https://doi.org/10.1175/MWR-D-18-0316.1.

Gallo, B., A. Clark, and S. Dembek, 2016: Forecasting tornadoes using convection-permitting ensembles. *Wea. Forecasting*, **31**, 273–295, https://doi.org/10.1175/WAF-D-15-0134.1.

——, 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.

Gil, Y., and Coauthors, 2019: Intelligent systems for geosciences: An essential research agenda. *Commun. ACM*, **62**, 76–84, https://doi.org/10.1145/3192335.

Goodfellow, I., Y. Bengio, and A. Courville, 2016: *Deep Learning*. MIT Press, 800 pp.

Harrison, D., and C. Karstens, 2017: A climatology of operational storm-based warnings: A geospatial analysis. *Wea. Forecasting*, **32**, 47–60, https://doi.org/10.1175/WAF-D-15-0146.1.

Hinton, G., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2012: Improving neural networks by preventing co-adaptation of feature detectors. arXiv, 1207 (0580), https://arxiv.org/pdf/1207.0580.pdf.

Hoerl, A., and R. Kennard, 1970: Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55–67, https://doi.org/10.1080/00401706.1970.10488634.

——, and ——, 1988: Ridge regression. *Encyclopedia of Statistical Sciences*, S. Kotz, Ed., Vol. 8, John Wiley and Sons, 129–136.

Homeyer, C., and K. Bowman, 2017: Algorithm description document for version 3.1 of the three-dimensional gridded NEXRAD WSR-88D radar (GridRad) dataset. University of Oklahoma, 23 pp., http://gridrad.org/pdf/GridRad-v3.1-Algorithm-Description.pdf.

Hunter, J., 2007: Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.*, **9**, 90–95, https://doi.org/10.1109/MCSE.2007.55.

Insurance Information Institute, 2019: Facts + statistics: Tornadoes and thunderstorms. III, https://www.iii.org/fact-statistic/facts-statistics-tornadoes-and-thunderstorms.

Ioffe, S., and C. Szegedy, 2015: Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Int. Conf. on Machine Learning*, Lille, France, International Machine Learning Society, http://proceedings.mlr.press/v37/ioffe15.pdf.

Kain, J., 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.

Karstens, C., and Coauthors, 2018: Development of a human–machine mix for forecasting severe convective events. *Wea. Forecasting*, **33**, 715–737, https://doi.org/10.1175/WAF-D-17-0188.1.

Kingma, D., and J. Ba, 2014: Adam: A method for stochastic optimization. arXiv, 1412 (6980), https://arxiv.org/pdf/1412.6980v9.pdf.

Kitzmiller, D., W. McGovern, and R. Saffle, 1995: The WSR-88D severe weather potential algorithm. *Wea. Forecasting*, **10**, 141–159, https://doi.org/10.1175/1520-0434(1995)010<0141:TWSWPA>2.0.CO;2.

Krizhevsky, A., I. Sutskever, and G. Hinton, 2017: ImageNet classification with deep convolutional neural networks. *Commun. ACM*, **60**, 84–90, https://doi.org/10.1145/3065386.

Kurth, T., and Coauthors, 2018: Exascale deep learning for climate analytics. *Int. Conf. for High Performance Computing, Networking, Storage, and Analysis*, Dallas, TX, IEEE, https://doi.org/10.1109/SC.2018.00054.

Lagerquist, R., A. McGovern, and D. Gagne, 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: An objective method of evaluating and devising storm-tracking algorithms. *Wea. Forecasting*, **25**, 701–709, https://doi.org/10.1175/2009WAF2222330.1.

——, I. Adrianto, T. Smith, and G. Stumpf, 2005: A spatiotemporal approach to tornado prediction. *IEEE Int. Joint Conf. on Neural Networks*, Montreal, QC, Canada, IEEE, 1642–1647, https://doi.org/10.1109/IJCNN.2005.1556125.

——, T. Smith, 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.

——, B. Herzog, and D. Kingfield, 2015: A method for extracting postevent storm tracks. *J. Appl. Meteor. Climatol.*, **54**, 451–462, https://doi.org/10.1175/JAMC-D-14-0132.1.

Luna-Herrera, J., G. Martinez-Cabrera, R. Parra-Maldonado, J. Enciso-Moreno, J. Torres-Lopez, F. Quesada-Pascual, R. Delgadillo-Polanco, and S. Franzblau, 2003: Use of receiver operating characteristic curves to assess the performance of a microdilution assay for determination of drug susceptibility of clinical isolates of *Mycobacterium tuberculosis*. *Eur. J. Clin. Microbiol. Infect. Dis.*, **22**, 21–27, https://doi.org/10.1007/s10096-002-0855-5.

Maas, A., A. Hannun, and A. Ng, 2013: Rectifier nonlinearities improve neural network acoustic models. *Proc. 30th Int. Conf. on Machine Learning*, Atlanta, GA, International Machine Learning Society, 6 pp., http://robotics.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.

Mahalik, M., B. Smith, K. Elmore, D. Kingfield, K. Ortega, and T. Smith, 2019: Estimates of gradients in radar moments using a linear least squares derivative technique. *Wea. Forecasting*, **34**, 415–434, https://doi.org/10.1175/WAF-D-18-0095.1.

Markowski, P., and Y. Richardson, 2009: Tornadogenesis: Our current understanding, forecasting considerations, and questions to guide future research. *Atmos. Res.*, **93**, 3–10, https://doi.org/10.1016/j.atmosres.2008.09.015.

——, and ——, 2014: What we know and don't know about tornado formation. *Phys. Today*, **67**, 26–31, https://doi.org/10.1063/PT.3.2514.

Marzban, C., and G. Stumpf, 1996: A neural network for tornado prediction based on Doppler radar-derived attributes. *J. Appl. Meteor.*, **35**, 617–626, https://doi.org/10.1175/1520-0450(1996)035<0617:ANNFTP>2.0.CO;2.

McGovern, A., R. Lagerquist, D. Gagne, G. Jergensen, K. Elmore, C. 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.

Mehdi, T., N. Bashardoost, and M. Ahmadi, 2011: Kernel smoothing for ROC curve and estimation for thyroid stimulating hormone. *Int. J. Public Health Res.*, **Special Issue 2011**, 239–242.

Metz, C., 1978: Basic principles of ROC analysis. *Semin. Nucl. Med.*, **8**, 283–298, https://doi.org/10.1016/S0001-2998(78)80014-2.

Muller, M., G. Tomlinson, T. Marrie, P. Tang, A. McGeer, D. Low, A. Detsky, and W. Gold, 2005: Can routine laboratory tests discriminate between severe acute respiratory syndrome and other causes of community-acquired pneumonia? *Clin. Infect. Dis.*, **40**, 1079–1086, https://doi.org/10.1086/428577.

National Climatic Data Center, 2020: Index of /pub/data/swdi/stormevents/csvfiles. NOAA, accessed 11 June 2020, ftp://ftp.ncdc.noaa.gov/pub/data/swdi/stormevents/csvfiles.

Ortega, K., T. Smith, J. Zhang, C. Langston, Y. Qi, S. Stevens, and J. Tate, 2012: The Multi-Year Reanalysis of Remotely Sensed Storms (MYRORSS) project. *26th Conf. on Severe Local Storms*, Nashville, TN, Amer. Meteor. Soc., 74, https://ams.confex.com/ams/26SLS/webprogram/Handout/Paper211413/p4_74_ortegaetal_myrorss.pdf.

Racah, E., C. Beckham, T. Maharaj, S. Kahou, Prabhat, and C. Pal, 2017: ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. *Advances in Neural Information Processing Systems*, Long Beach, CA, Neural Information Processing Systems, 6932, https://papers.nips.cc/paper/6932-extremeweather-a-large-scale-climate-dataset-for-semi-supervised-detection-localization-and-understanding-of-extreme-weather-events.pdf.

Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat, 2019: Deep learning and process understanding for data-driven Earth system science. *Nature*, **566**, 195–204, https://doi.org/10.1038/s41586-019-0912-1.

Rhyzkov, A. V., T. J. Schuur, D. W. Burgess, and D. S. Zrnić, 2005: Polarimetric tornado detection. *J. Appl. Meteor.*, **44**, 557–570, https://doi.org/10.1175/JAM2235.1.

Roebber, P., 2009: Visualizing multiple measures of forecast quality. *Wea. Forecasting*, **24**, 601–608, https://doi.org/10.1175/2008WAF2222159.1.

Silver, D., and Coauthors, 2016: Mastering the game of Go with deep neural networks and tree search. *Nature*, **529**, 484–489, https://doi.org/10.1038/nature16961.

Skinner, P., 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., R. Thompson, J. Grams, C. Broyles, and H. 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.

Smith, T., and Coauthors, 2016: Multi-Radar Multi-Sensor (MRMS) severe weather and aviation products: Initial operating capabilities. *Bull. Amer. Meteor. Soc.*, **97**, 1617–1630, https://doi.org/10.1175/BAMS-D-14-00173.1.

Snook, N., M. Xue, and Y. Jung, 2019: Tornado-resolving ensemble and probabilistic predictions of the 20 May 2013 Newcastle–Moore EF5 tornado. *Mon. Wea. Rev.*, **147**, 1215–1235, https://doi.org/10.1175/MWR-D-18-0236.1.

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.

——, G. S. Romine, C. S. Schwartz, D. J. Gagne, and M. L. Weisman, 2016a: Explicit forecasts of low-level rotation from convection-allowing models for next-day tornado prediction. *Wea. Forecasting*, **31**, 1591–1614, https://doi.org/10.1175/WAF-D-16-0073.1.

——, C. S. Schwartz, G. S. Romine, K. R. Fossell, and M. L. Weisman, 2016b: Severe weather prediction using storm surrogates from an ensemble forecasting system. *Wea. Forecasting*, **31**, 255–271, https://doi.org/10.1175/WAF-D-15-0138.1.

Stensrud, D., 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.

——, and Coauthors, 2013: Progress and challenges with Warn-on-Forecast. *Atmos. Res.*, **123**, 2–16, https://doi.org/10.1016/j.atmosres.2012.04.004.

Storm Prediction Center, 2020: Mesoscale analysis pages. NOAA, accessed 9 March 2020, http://www.spc.noaa.gov/exper/mesoanalysis/.

Thompson, R., B. Smith, J. Grams, A. 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.

Trapp, R., G. Stumpf, and K. Manross, 2005: A reassessment of the percentage of tornadic mesocyclones. *Wea. Forecasting*, **20**, 680–687, https://doi.org/10.1175/WAF864.1.

Wang, L., K. Scott, L. Xu, and D. 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.

Wheatley, D., K. Knopfmeier, T. Jones, and G. Creager, 2015: Storm-scale data assimilation and ensemble forecasting with the NSSL Experimental Warn-on-Forecast System. Part I: Radar data experiments. *Wea. Forecasting*, **30**, 1795–1817, https://doi.org/10.1175/WAF-D-15-0043.1.

Wilson, K., P. Heinselman, C. Kuster, D. Kingfield, and Z. Kang, 2017: Forecaster performance and workload: Does radar update time matter? *Wea. Forecasting*, **32**, 253–274, https://doi.org/10.1175/WAF-D-16-0157.1.

Wimmers, A., C. Velden, and J. 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.

Yussouf, N., D. Dowell, L. Wicker, K. Knopfmeier, and D. Wheatley, 2015: Storm-scale data assimilation and ensemble forecasts for the 27 April 2011 severe weather outbreak in Alabama. *Mon. Wea. Rev.*, **143**, 3044–3066, https://doi.org/10.1175/MWR-D-14-00268.1.