An Automated Python Algorithm to Quantify Z_{DR} Arc and K_{DP}–Z_{DR} Separation Signatures in Supercells

MATTHEW B. WILSON[a] AND MATTHEW S. VAN DEN BROEKE[a]

[a] *University of Nebraska–Lincoln, Lincoln, Nebraska*

(Manuscript received 20 April 2020, in final form 4 December 2020)

ABSTRACT: Supercell thunderstorms often have pronounced signatures of hydrometeor size sorting within their forward-flank regions, including an arc-shaped region of high differential reflectivity (Z_{DR}) along the inflow edge of the forward flank known as the Z_{DR} arc and a clear horizontal separation between this area of high Z_{DR} values and an area of enhanced K_{DP} values deeper into the storm core. Recent work has indicated that Z_{DR} arc and K_{DP}–Z_{DR} separation signatures in supercell storms may be related to environmental storm-relative helicity and low-level shear. Thus, characteristics of these signatures may be helpful to indicate whether a given storm is likely to produce a tornado. Although Z_{DR} arc and K_{DP}–Z_{DR} separation signatures are typically easy to qualitatively identify in dual-polarization radar fields, quantifying their characteristics can be time-consuming and makes research into these signatures and their potential operational applications challenging. To address this problem, this paper introduces an automated Python algorithm to objectively identify and track these signatures in Weather Surveillance Radar-1988 Doppler (WSR-88D) data and quantify their characteristics. This paper will discuss the development of the algorithm, demonstrate its performance through comparisons with manually generated time series of Z_{DR} arc and K_{DP}–Z_{DR} separation signature characteristics, and briefly explore potential uses of this algorithm in research and operations.

KEYWORDS: Convective storms; Algorithms; Radars/Radar observations; Nowcasting

1. Introduction

Though supercell storms represent a small fraction of all thunderstorms, they are responsible for a disproportionate amount of severe weather reports in the United States. For instance, they were associated with 51% of total severe weather reports and 68.2% of tornado reports in the Midwest in one 2-yr study period despite representing only 22.8% of all storms examined (Duda and Gallus 2010). With the advent of dual-polarization (dual-pol) radar and the subsequent dual-pol upgrade of the Weather Surveillance Radar-1988 Doppler (WSR-88D) network, several dual-pol signatures have been identified in supercell storms that may be of use in determining a particular supercell's likelihood of producing severe wind, large hail, or a tornado (e.g., Kumjian and Ryzhkov 2008; Van Den Broeke et al. 2008; Romine et al. 2008). These signatures include tornadic debris signatures (Ryzhkov et al. 2005; Van Den Broeke and Jauernic 2014; Snyder and Ryzhkov 2015), hail signatures in the core and forward flank (Picca and Ryzhkov 2012; Dawson et al. 2014; Van Den Broeke 2016), differential reflectivity (Z_{DR}) columns associated with convective updrafts (Brandes et al. 1995; Kumjian et al. 2014; Snyder et al. 2015, 2017; Kuster et al. 2019), and the Z_{DR} arc along the supercell's forward-flank reflectivity gradient (Kumjian and Ryzhkov 2008, 2009; Dawson et al. 2014, 2015).

Among these supercell dual-pol signatures, initial studies (Kumjian and Ryzhkov 2008, 2009; Kumjian et al. 2010; Palmer et al. 2011; Crowe et al. 2012) have indicated that the Z_{DR} arc may show particular promise for use in operations, since it may be able to provide information about the low-level near-storm kinematic environment. The Z_{DR} arc is formed as precipitation particles falling through a supercell's forward flank are sorted by the storm-relative flow, with smaller drops taking longer to fall and thus being advected farther into the storm core by the storm-relative wind than larger drops (Kumjian and Ryzhkov 2008, 2009; Dawson et al. 2014, 2015). Since larger drops are more oblate and thus produce higher Z_{DR} values (Seliga and Bringi 1976), this results in a band of high Z_{DR} along a supercell's forward-flank reflectivity gradient as smaller drops are sorted out of this region (Fig. 1). Size-sorted melting hail may also contribute to enhancing Z_{DR} in a supercell's forward flank in addition to raindrop size sorting, albeit in an area slightly closer to the storm core than the traditional Z_{DR} arc (Dawson et al. 2014). Additionally, Dawson et al. (2015) found substantial positive correlations between the magnitude of the storm-relative flow causing size sorting in the arc and storm-relative helicity in all cases they examined except for idealized hodographs that had either no shear or perfect Beltrami flow. Thus, Z_{DR} arc size and intensity may be a useful proxy for changes in the storm-relative wind magnitude and related changes in environmental storm-relative helicity on scales smaller than those resolved by the radiosonde network (Kumjian and Ryzhkov 2008, 2009; Dawson et al. 2015).

Observational work by Van Den Broeke (2016, hereafter VDB16) examining Z_{DR} arc characteristics in 25 classic supercells in 12 different environments found that Z_{DR} arc width, areal extent, and mean Z_{DR} value were positively correlated with low-level bulk shear; however, other environmental variables such as midlevel relative humidity and the height of the level of free convection (LFC) also influenced the size and intensity of the Z_{DR} arc. The correlation between arc areal extent and low-level shear is hypothesized by VDB16 to be due to stronger size sorting by the storm-relative wind in higher-shear

Corresponding author: Matthew B. Wilson, mwilson41@huskers.unl.edu

DOI: 10.1175/JTECH-D-20-0056.1

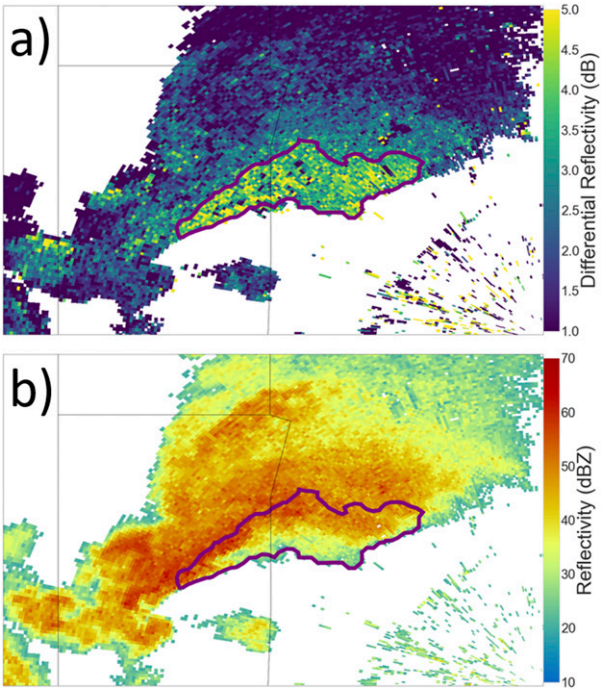

FIG. 1. Schematic of a Z_{DR} arc in a tornadic supercell on 30 Mar 2016 near Tulsa, Oklahoma: (a) the arc outlined in purple over a background of Z_{DR}; (b) the arc in relation to the reflectivity field.

environments, which would not only lead to a drop size distribution in the arc region dominated by large drops with high Z_{DR} but may also lead to a larger area over which a sufficient degree of size sorting occurs to produce Z_{DR} values above 3.5 dB in higher-shear environments. The increase in arc size in environments with low relative humidity in the low levels may be due to preferential evaporation of small drops as noted in the analysis of supercell hook echo drop-size distributions performed by Kumjian (2011), which helps skew the drop size distribution toward larger drops and produce higher Z_{DR} values apart from size sorting by the storm-relative wind. However, Z_{DR} enhancements due to evaporation may be fairly small compared to those from size sorting, with Kumjian and Ryzhkov (2010) finding only a 0.25 dB increase in Z_{DR} due to subcloud evaporation in the driest profiles in their simulations. Since the correlations found between arc metrics and environmental variables by VDB16 were in a fairly small sample of storms, and to the best knowledge of the authors little or no other modeling or theoretical work exists to further illuminate these potential correlations, further work examining a larger number of supercells in different environments may be useful in eliciting what Z_{DR} arc metrics can indicate to forecasters about a supercell's environment.

In addition to environmental information, changes in Z_{DR} arc metrics may also shed light on the progression of supercell and tornado life cycles. Palmer et al. (2011) observed a cyclic pattern of Z_{DR} arc evolution in a violently tornadic supercell during the 10 May 2010 tornado outbreak in Oklahoma, with the Z_{DR} arc extending back toward the hook echo leading up to tornadogenesis and weakening around tornado demise and occlusion, only to strengthen again as a new mesocyclone became established and produced another tornado. Kumjian et al. (2010) documented a similar pattern of evolution during the cycling process of a nontornadic supercell in central Oklahoma, with the Z_{DR} arc strengthening and extending back toward the hook echo leading up to mesocyclone occlusion and weakening following occlusion. Both Palmer et al. (2011) and Kumjian et al. (2010) hypothesize that storm-relative inflow weakens during occlusion as the occluding updraft weakens, disrupting the size sorting in the arc region and weakening the Z_{DR} arc. As the next mesocyclone begins to strengthen, storm-induced perturbations to the near-storm wind field may strengthen the storm-relative inflow again, leading to stronger size sorting in the arc region and a larger, more intense Z_{DR} arc. Z_{DR} arc behavior across tornado life cycles has also been investigated by Van Den Broeke (2017, hereafter VDB17), with arcs observed to grow larger and wider from tornadogenesis to tornado dissipation. VDB17, in line with prior work (Palmer et al. 2011; Crowe et al. 2012) hypothesized that low-level storm-relative inflow may be stronger during tornadic times than when a tornado is not present, leading to stronger size sorting and a larger and more intense Z_{DR} arc. However, Van Den Broeke (2020) compared Z_{DR} arc size and intensity in tornadic storms in the half hour prior to tornadogenesis to arc characteristics in nontornadic storms and did not find any notable differences in mean arc size and intensity between the pretornadic and nontornadic samples. Further study of trends in Z_{DR} arc characteristics leading up to tornadogenesis or tornadogenesis failure and over the mesocyclone cycling process in a larger sample of supercells may be useful in determining what the behavior of the Z_{DR} arc may tell forecasters about a storm's cycling process and how likely a storm is to produce a tornado.

Another size sorting signature in supercells that has been examined in previous studies is the separation between the Z_{DR} enhancement in the forward flank and the area of enhanced K_{DP} typically found within the storm core, known as the K_{DP} foot (Romine et al. 2008; Crowe et al. 2010, 2012). First examined in detail by Crowe et al. (2010) in three supercells in Hurricane Rita's rainbands, this separation is attributable to the same drop-size sorting that creates the Z_{DR} arc. As size sorting by the storm-relative wind advects smaller drops out of the Z_{DR} arc region, it reduces the liquid water content within the Z_{DR} arc and enhances it deeper in the storm core where the small drops end up. Since K_{DP} is strongly related to the liquid water content in a given radar volume, this reduces the K_{DP} within the Z_{DR} arc and may enhance it in the storm core, leading to a horizontal separation between areas of enhanced Z_{DR} and K_{DP} within the storm (Crowe et al. 2010). The magnitude of a vector drawn between the centroids of the K_{DP} and Z_{DR} enhancements is hypothesized in previous studies (Jurewicz and Gitro 2018; Loeffler and Kumjian 2018) to scale with the strength of the low-level storm-relative wind, and the orientation of that vector is thought to be parallel to the low-level storm-relative wind direction. Thus, it may also be a useful proxy for the low-level storm-relative wind and quantities related to it, such as low-level shear, storm-relative helicity, and storm inflow (Crowe et al. 2010, 2012).

Initial work on K_{DP}–Z_{DR} separation signatures often focused on the magnitude of the separation between regions of enhanced K_{DP} and Z_{DR}. Crowe et al. (2012) examined this signature in three different severe weather events and found that the separation between areas of enhanced Z_{DR} and K_{DP} was greater during tornadic periods in the tornadic storms examined and that areas of enhanced Z_{DR} and K_{DP} overlapped more during nontornadic periods and in storms that never produced tornadoes. The more pronounced K_{DP}–Z_{DR} separation at tornadic times was thought to be due to a linkage between more pronounced separation signatures and enhanced environmental storm-relative helicity. Martinaitis (2017) also found that a signature of horizontal separation between enhancements in Z_{DR} and K_{DP} was useful in differentiating between tornadic and nontornadic storms in convection associated with tropical cyclones in Florida when used in conjunction with analysis of the reflectivity and velocity fields. Ongoing work by Jurewicz and Gitro (2018) is dedicated to determining how useful the K_{DP}–Z_{DR} separation signature is in differentiating between tornadic and nontornadic supercells in a large sample of storms and developing ways to implement this signature in warning operations. Furthermore, Loeffler and Kumjian (2018) have developed a semiautomated algorithm to quantify the K_{DP}–Z_{DR} separation signature in tornadic nonsupercell storms, and Loeffler et al. (2020) applied this algorithm to supercells.

One parameter that recent work by Loeffler and Kumjian (2018) and Loeffler et al. (2020) has found to be particularly useful is the magnitude of the angle (Fig. 2) between a vector connecting the K_{DP} and Z_{DR} enhancement centroids and the storm motion vector, referred to as the separation orientation relative to storm motion. Loeffler and Kumjian (2018) found that separation orientations closer to 90° were found to be associated with larger values of low-level SRH for a given separation vector length, and Loeffler et al. (2020) found that tornadic supercells were much more likely to have separation orientations close to orthogonal than nontornadic supercells. Similarly, Homeyer et al. (2020) examined storm-centered radar composites for a large sample of tornadic and nontornadic supercells, and they found that the separation vector in the tornadic supercell composite was much closer to orthogonal to storm motion than the separation vector in the nontornadic composite. While Jurewicz and Gitro (2018) did not directly address the separation orientation, their work using separation vectors, surface winds, and storm motion vectors to construct simple hodographs for the near-storm environment also suggests that larger angles between the separation and storm motion vectors should be correlated with higher SRH and low-level shear. For brevity, separation orientation relative to storm motion will be referred to as the separation angle for the rest of this study. Manual calculation of this signature has also recently been featured as part of the National Weather Service Central Region's Tornado Warning Improvement Project (TWIP) training (Johnson 2018). However, since manually calculating the separation angle can be time-consuming and subjective, especially when dealing with multiple supercells, a fully automated algorithm that can calculate this signature may prove useful to operational forecasters.

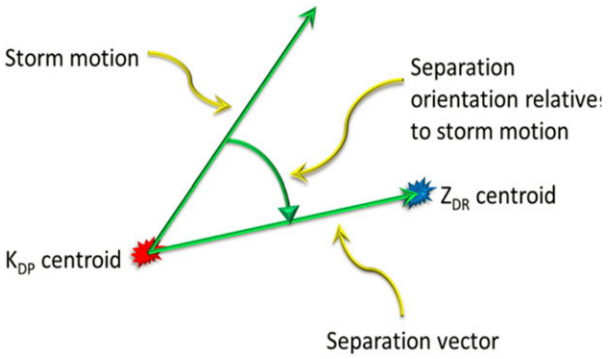

FIG. 2. Plan view of the separation vector and separation orientation relative to storm motion in an idealized storm (Fig. 4 from Loeffler and Kumjian 2018).

To support future work examining Z_{DR} arc and K_{DP}–Z_{DR} separation signatures in large samples of supercells, this study will introduce an open-source, automated Python algorithm to quickly and reliably identify and track these signatures and quantify their characteristics. In line with the previous work discussed in the background section above, this algorithm focuses on calculating five variables to quantify the characteristics of supercell size sorting signatures: Z_{DR} arc area, Z_{DR} arc intensity (quantified as the mean Z_{DR} value within the arc as well as the mean of the 10 pixels with the highest Z_{DR} values within the arc), the separation angle, and the length of the K_{DP}–Z_{DR} separation vector. Section 2 will describe how the algorithm works and the methodology used to calibrate Z_{DR} data for the algorithm. Section 3 will evaluate the algorithm's performance through comparisons of manually identified and algorithm-derived Z_{DR} arc and K_{DP}–Z_{DR} separation signature characteristics, discuss the algorithm's limitations, and examine the algorithm's performance in detail for a significantly tornadic supercell. Finally, section 4 will summarize the results and discuss future algorithm uses and improvements.

2. Data and methods

a. Z_{DR} calibration

To mitigate potential bias in the Z_{DR} data used to develop the algorithm described in this paper, scatterer-based Z_{DR} calibration was performed following the methodology of Ryzhkov et al. (2005), Picca and Ryzhkov (2012), and Van Den Broeke and Van Den Broeke (2015). This calibration technique makes use of the relatively consistent radar presentation of dry snow aggregates around 1.5 km above the environmental freezing level, which tend to have reflectivity values between 20 and 35 dBZ, copolar cross-correlation coefficient (CC) values above 0.99, and Z_{DR} values between 0.1 and 0.2 dB (Ryzhkov et al. 2005; Picca and Ryzhkov 2012; Van Den Broeke and Van Den Broeke 2015). To calibrate Z_{DR} using this technique, the average Z_{DR} of a region of dry snow aggregates 1.5 km above the freezing level is calculated first. Next, a reference Z_{DR} value of 0.15 dB, representing the Z_{DR} value that dry snow aggregates should theoretically display

on a well-calibrated radar, is subtracted from the average Z_{DR} value calculated in the previous step to create a calibration factor. This calibration factor is then subtracted from the Z_{DR} field as a whole to calibrate it. For the 51 storm dataset from VDB16 and VDB17 that is used for algorithm verification in section 3a, the manually calculated calibration factors used by VDB16 and VDB17 were also used when running the algorithm to maintain consistency with the Z_{DR} calibration in the dataset against which the algorithm is verified. For this dataset, calibration factors ranged from −0.75 to 0.71 dB, with a mean and median calibration factor of −0.09 dB. However, the mean and median values of the calibration factor were 0.30 and 0.28 dB. These values suggest that Z_{DR} bias may be substantial enough to warrant calibration in a nonnegligible fraction of cases when performing analyses involving cases from different radars or different days.

To simplify the calibration process for newer cases, a Python script has also been created to automatically calculate the Z_{DR} calibration factor. The script reads in 15 min of radar data from the center of a given case's analysis window using the Python ARM Radar Toolkit (PyART; Helmus and Collis 2016), along with a user-provided freezing level from a proximity sounding. It then calculates the average Z_{DR} for all radar pixels 1500 m (±500 m) above the freezing level that have reflectivity between 20 and 35 dBZ and CC above 0.99. A reference Z_{DR} value of 0.15 dB is then subtracted from this average Z_{DR} value, which produces a calibration factor to be subtracted from the case's Z_{DR} field. This script is available in the github repository for this paper (https://github.com/mwilson14/ZDRArcAlgorithm) for use with the algorithm.

b. Algorithm design

While recognizing a Z_{DR} arc is often intuitive for a human researcher or forecaster, programming a computer to identify the same signature is often more difficult. Thus, we must first precisely and objectively define what a Z_{DR} arc is. For the development of this algorithm, a Z_{DR} arc is defined as an area of Z_{DR} greater than 3.5 dB located on the inflow side of a supercell storm's forward flank, consistent with the threshold used to define the Z_{DR} arc core by VDB16. This 3.5 dB Z_{DR} threshold was initially chosen as a value that subjectively seemed to capture the most intense part of the Z_{DR} arcs in the supercell sample examined by VDB16. This threshold has been further evaluated in work presented by Van Den Broeke (2020), where 1.5, 2, 2.5, 3, 3.5, and 4 dB thresholds were tested for defining the Z_{DR} arc in several supercells. Thresholds that were too low (2 dB and below) were found to produce "arcs" that were much too large and included much of the storm area, while thresholds higher than 3.5 dB often missed arcs entirely. Overall, this sensitivity analysis indicated that the 3.5 dB threshold performed well in most of the storms examined. However, we do acknowledge that situations exist in which a lower or higher threshold would be useful for identifying Z_{DR} arcs (e.g., in tropical cyclone convection with overall low Z_{DR} values), and with that in mind we added the capability for this threshold to be modified by the user in the final version of the algorithm.

The first step in the process of the algorithm's process in identifying Z_{DR} arcs is to acquire and process radar data into a format that appropriate Python modules can ingest. This is done using the nexradaws Python module (https://github.com/aarande/nexradaws) to download archived WSR-88D data from an archive Unidata maintains in partnership with Amazon Web Services (available at https://aws.amazon.com/public-datasets/nexrad/), and extracting and gridding the necessary radar variables (Z_{HH}, Z_{DR}, K_{DP}, and CC) from the lowest tilt of each scan (usually 0.5°) onto a grid with a horizontal spacing of 493 m using PyART. PyART's gridding function has a number of possible settings, and for this analysis was configured to use a Barnes analysis scheme with a radius of influence that expands with distance from the radar to account for beam broadening to map the raw radar data to the grid. To identify the inflow side of the storm, the direction of the reflectivity gradient vector is calculated for all points where gridded reflectivity is greater than 20 dBZ, and the direction of a manually defined vector perpendicular to the storm's forward flank and pointing into the core is subtracted from it [referred to as the forward-flank downdraft (FFD) vector; Figs. 3a,b]. In the final version of the algorithm, this direction is entered by the user in a pop-up window prior to starting the algorithm. The gridded Z_{DR} field is masked in areas where this difference is greater than 120°, since these areas are likely not on the inflow side of the storm. Areas with CC values below 0.60 are also masked out in the Z_{DR} field, since these areas likely represent nonmeteorological scatterers and can create spurious Z_{DR} arc identifications (Figs. 3c,d).

The second step in the algorithm's workflow is to pick out areas that may be part of a Z_{DR} arc. To identify potential Z_{DR} arc objects, the Z_{DR} field is contoured at 3.25 dB, which has been reduced slightly from the 3.5 dB in the arc definition to account for smoothing-induced loss of higher Z_{DR} values as the data are gridded. Individual, closed polygons are extracted from the contoured Z_{DR} field using the Shapely Python module (available at https://github.com/Toblerity/Shapely/tree/master/docs/). The area, centroid, mean Z_{DR} value, maximum Z_{DR} value, mean reflectivity gradient value, mean reflectivity gradient direction relative to the FFD vector, mean CC, and mean reflectivity are then calculated and saved for each potential Z_{DR} arc polygon. Next, all Z_{DR} polygons identified in the previous step need to be associated with individual storms. To create storm objects, the reflectivity field is smoothed and the 45 dBZ contour is plotted and split into polygons in a similar manner to the Z_{DR} field. The algorithm plots the centroids of polygons with areas greater than 20 km^2 and saves them as storm objects that are tracked through subsequent radar scans. To deal with the possibility of supercells embedded within larger convective structures, the algorithm plots a 50 dBZ contour inside any polygons with areas greater than 300 km^2 and uses the centroids of any polygon(s) derived from that contour as the storm objects within that polygon. In cases with especially low or high reflectivity, the 45 and 50 dBZ thresholds can be adjusted up or down for more effective tracking. An example storm object is shown in Fig. 3d. Once storm objects are identified, Z_{DR} polygons are matched with the closest storm object to their centroid within a distance threshold of 30 km, and the distance and direction from the polygon to its corresponding storm object are saved for each polygon.

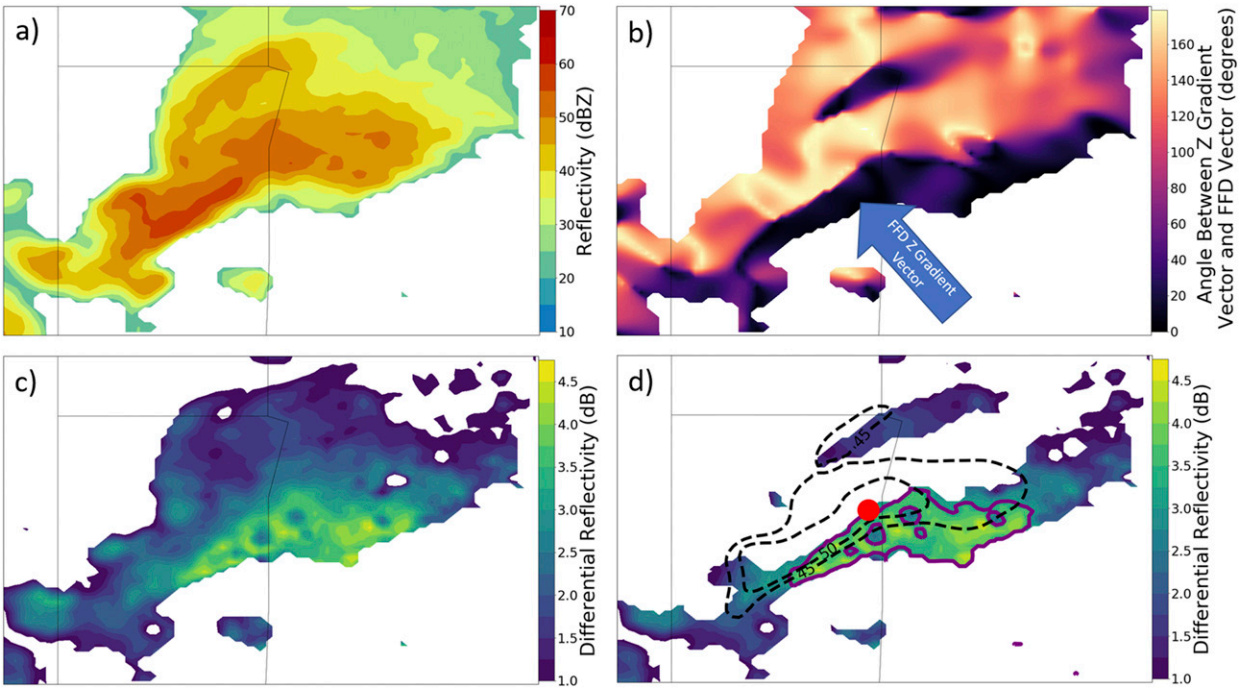

FIG. 3. Inputs for the Z_{DR} arc algorithm, showing (a) gridded reflectivity, (b) the FFD gradient vector (blue arrow) and reflectivity gradient direction relative to that vector, (c) the raw gridded Z_{DR} field, and (d) Z_{DR} field with data masked where the Z gradient direction relative to the FFD gradient vector direction is greater than 120° and CC less than 0.60, along with the 3.25 dB Z_{DR} contour (purple) and the associated storm object (dashed contours and red dot) for reference.

The third and final step in the algorithm's Z_{DR} arc identification process is to eliminate spurious arc detections and combine multiple arc objects associated with the same storm. At this point, a typical supercell with a Z_{DR} arc will often have multiple polygons associated with it, since it is not uncommon for a Z_{DR} arc to contain multiple noncontiguous regions of enhanced Z_{DR} interrupted by a hail signature or other areas of lower Z_{DR}. However, at this stage the algorithm frequently identifies polygons that represent areas of enhanced Z_{DR} away from the inflow side of the storm's forward flank in areas such as the rear of the echo appendage, the northern or northwestern side of the precipitation shield, or with a small cell moving into the main supercell's inflow region. Although areas of high Z_{DR} on the northern or northwestern flank of a storm may be the Z_{DR} arc of a developing left-moving supercell, since this algorithm is designed to focus on right-moving supercells these areas are excluded, since how they relate to the inflow characteristics of the right-moving storm may not be clear. An example of a cell with a real and a spurious polygon is shown in Fig. 4. Areas such as this are not part of the Z_{DR} arc, and thus a reliable method for removing these spurious polygons is needed.

The automated algorithm presented here uses a random forest classifier to accomplish this task. Random forest classifiers (Breiman 2001) are a type of machine learning algorithm that have shown promise working on similar classification problems, with recent work applying them to tasks as diverse as identifying and tracking mesoscale convective systems (MCSs) in regional reflectivity mosaics (Haberlie and Ashley 2018), improving the prediction of extreme precipitation events (Herman and Schumacher 2018), and forecasting the initiation of deep convection using satellite data and numerical model output (Mecikalski et al. 2015). Random forests work by training an ensemble of decision trees on manually labeled features (in this case, manually labeled arc and false detection objects) and a series of attributes of those features, with the goal being to use the attributes to accurately place the features

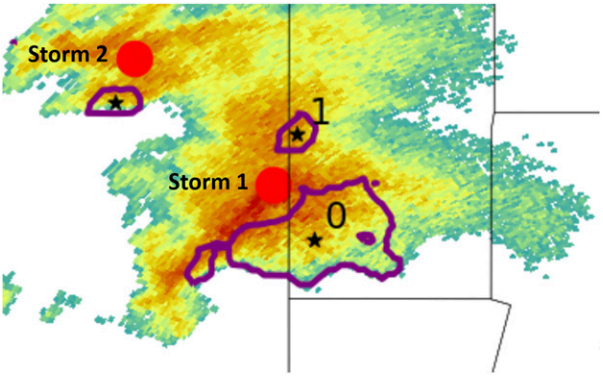

FIG. 4. An example of a supercell (labeled as "Storm 1," with the storm centroid marked with a red circle) with multiple Z_{DR} polygons (purple outlines/numbers, with centroids marked by small black stars) detected by the algorithm. In this case, object 0 is the arc, while object 1 is a spurious detection due to an area of high Z_{DR} in the northern part of the storm. "Storm 2" is a nonsupercell storm object with an associated patch of high Z_{DR} (purple outline).

TABLE 1. Variables saved for each potential arc object for use in the random forest algorithm.

Saved arc object variables
Arc area (km^2)
Storm centroid-arc object centroid distance (km)
Mean Z_{DR} value (dB)
Max Z_{DR} value (dB)
Mean CC value
Mean K_{DP} value (° km^{-1})
Mean Z value (dBZ)
Mean Z gradient direction relative to FFD vector direction (°)
Mean Z gradient value (dBZ km^{-1})
Angle between storm centroid-arc object centroid vector and FFD vector (°)
X component of storm centroid-arc object centroid vector (km)
Y component of storm centroid-arc object centroid vector (km)

in their manually labeled classes. Each decision tree starts by randomly picking one of the attributes and picking the attribute value that best splits the features into their correct classifications from a random subset of the attribute values. This is then repeated for several different attributes, creating a multilevel decision tree. The use of random subsets of the attributes to train each tree creates an ensemble of trees that produce slightly different outcomes. Since each tree by itself may not be an excellent classifier, the trees are combined into an ensemble to create a random forest, with the class indicated by the majority of the trees used as the output of the ensemble. The ensemble prediction produced by the random forest tends to be much more accurate than what any individual decision tree could produce on its own (Géron 2017).

To create a random forest algorithm that can differentiate between actual arc objects and false detections, a large training dataset of manually labeled candidate polygons is needed. To create this dataset, 20 supercell cases (10 tornadic, 10 nontornadic) were identified using the case selection criteria from VDB16 and VDB17: namely, each storm had to display typical supercell features such as a Z_{DR} arc, Z_{DR} column, and midlevel mesocyclone while remaining close enough to the radar to be sampled at altitudes below 1 km. These cases were then run through an initial version of the Z_{DR} arc detection and tracking algorithm. This script outputs a spreadsheet of the saved characteristics for all polygons associated with each storm (listed in Table 1), as well as plots of radar reflectivity with each potential arc polygon plotted and numbered (as shown for one storm in Fig. 4). Using the reflectivity images, each polygon was manually classified as an arc polygon or nonarc polygon based on whether it was located along the inflow side of the forward flank of its associated supercell. From the 20 supercell cases examined, this resulted in 1399 analysis times and 4724 manually labeled polygons, split between 2154 arc polygons and 2570 false detections. To ensure that the random forest algorithm would work in differentiating between actual Z_{DR} arc objects and false detections in situations where it would be useful to define the Z_{DR} arc core with a value of Z_{DR} different than 3.5 dB [for example, with low-topped supercells or supercells in tropical cyclone rainbands where drop sizes (and thus Z_{DR} values) tend to be smaller overall], the polygon mean Z_{DR} was not used as a predictor variable in the final random forest algorithm, and the polygon maximum Z_{DR} was normalized by each polygon's mean Z_{DR}.

Next, this training dataset was fed into a random forest classifier created in Python using the scikit-learn module (Pedregosa et al. 2011). This random forest classifier included 100 decision trees and used a 9:1 train–test split, meaning that 90% of the training dataset was randomly selected by scikit-learn to be set aside to train the random forest algorithm and the remaining 10% was used to test the performance of the resulting classifier. Since training the algorithm on one particular train–test split may give unrepresentative information on the algorithm's performance if the subset of the algorithm used for testing is particularly "easy" or "hard" for the random forest to classify, a technique known as k-fold cross validation was used to evaluate the algorithm's performance. In k-fold cross validation, the training dataset is divided into k subsets that are approximately equal in size. One of the subsets is reserved as the testing dataset, and the algorithm is trained on the remaining subsets and tested on the reserved subset. This is then repeated k times, with each subset being used as the training subset once, to generate k estimates of the model's performance (Kuhn and Johnson 2013). In this case, k was set to 10, and the model achieved an average probability of correct detection (POD) of 88.1% and a false alarm rate (FAR) of 11.4% on the training data. A final version of the random forest algorithm was then created by training the random forest on the entire training dataset. This final algorithm was tested on an independent testing dataset consisting of 1753 manually labeled Z_{DR} polygons from 51 supercells drawn from the supercell datasets used by VDB16 and VDB17. The random forest performed fairly well on this dataset, with a POD of 93.1% and a FAR of 17.9%, and captured almost all main arc objects in the traditional location along the forward-flank edge, as seen in Fig. 5. The FAR of 17.9% may seem high; however, many of the false alarm objects are fairly small patches of high Z_{DR} along the periphery of the arc or within the supercell's hook echo (green dots in Fig. 5), which may or may not be identified as part of the Z_{DR} arc in a given manual analysis. The final version of the random forest algorithm was then integrated into the Z_{DR} arc algorithm code to help remove spurious Z_{DR} polygons. Once the random forest algorithm has ideally removed any false arc detections for each storm, the remaining arc polygons are automatically combined into a single arc object for each storm in each radar scan. Z_{DR} arc characteristics (listed in Table 2) are then calculated for each arc object and saved for each radar scan. This results in a Pandas dataframe (McKinney 2010) containing time series of arc characteristics for each identified storm.

c. K_{DP}–Z_{DR} separation angle analysis

Since Loeffler and Kumjian (2018) and Loeffler et al. (2020) found the separation angle between the K_{DP}–Z_{DR} separation vector and the storm motion vector (Fig. 2) to be useful in distinguishing tornadic and nontornadic storms, an objective version of this calculation is included in the algorithm presented here. K_{DP} foot signatures are objectively identified in a

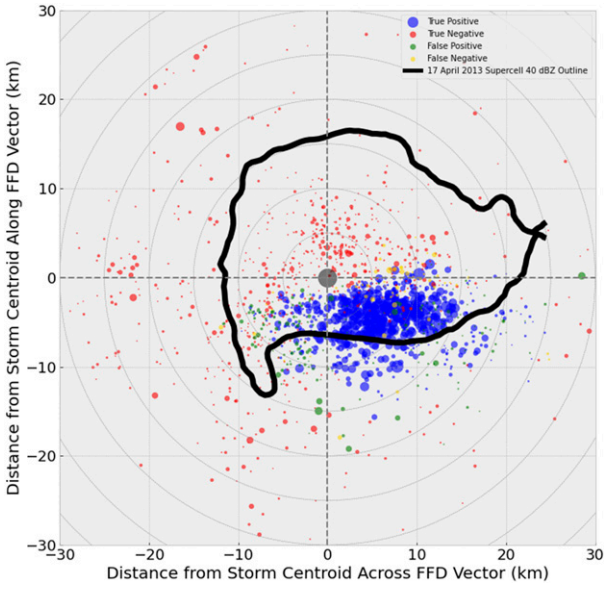

FIG. 5. Storm-relative plot of all 1752 Z_{DR} objects classified by the random forest from the 51 storm testing dataset. Correctly detected Z_{DR} arc object centroids are plotted in blue, false detection centroids are green, false negative centroids are yellow, and true negative centroids are red. For all objects, the dot size is proportional to the object area. The FFD vector points to the top of the plot, and the black outline is the time-averaged 40 dBZ outline of a supercell on 17 Apr 2013 near KFDR for reference. The gray dot in the middle is the storm centroid.

similar manner to how the initial Z_{DR} polygons are constructed. First, the K_{DP} field is masked where reflectivity is below 35 dBZ and contoured at 1.5° km^{-1}. The 1.5° km^{-1} K_{DP} threshold was chosen to agree with the threshold used to define the K_{DP} foot in previous studies (Romine et al. 2008; Crowe et al. 2012). This contour of K_{DP} is then broken into polygons using Shapely, and polygons are assigned to the closest storm object within 15 km. Multiple polygons on a single storm are combined into a single K_{DP} foot object and the centroid of this object is then used as the final K_{DP} foot centroid. For storms with both a K_{DP} foot object and a Z_{DR} arc object, a separation vector is then defined extending from the K_{DP} foot centroid to the Z_{DR} arc centroid. The separation angle is then calculated as the magnitude of the counterclockwise turning from the separation vector to the storm motion vector, as seen in Fig. 2. The storm motion direction is entered by the user in a text box when the algorithm is started, and it can be an observed storm motion from an ongoing storm or a predicted storm motion from a sounding.

TABLE 2. List of variables calculated and saved by the algorithm.

Saved storm object variables
Mean arc Z_{DR} value (dB)
Mean of 10 maximum Z_{DR} values in arc (dB)
Z_{DR} arc area (km^2)
K_{DP}–Z_{DR} separation vector length (km)
Separation angle magnitude (°)

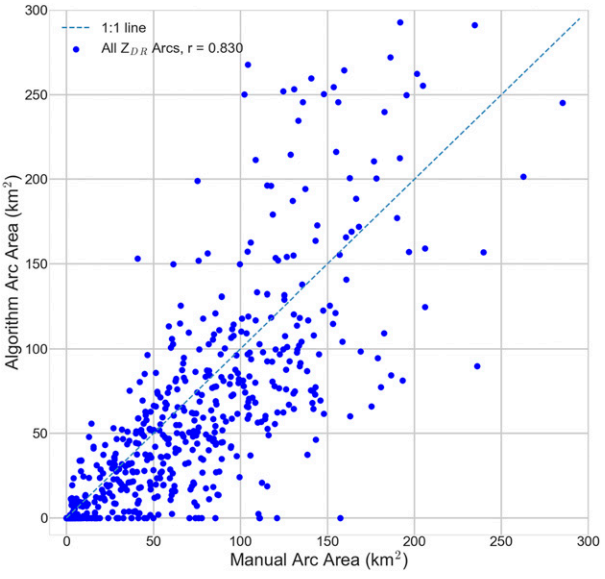

FIG. 6. Comparison between algorithm-derived and manual Z_{DR} arc areal extents. Dashed blue line is a 1:1 line along which a perfect match would fall.

The separation vector calculation presented here differs from that presented by Loeffler and Kumjian (2018) in that the separation vector is calculated between the centroids of the K_{DP} foot and Z_{DR} arc objects as defined above, while Loeffler and Kumjian (2018) calculate their separation vector between areas of enhanced K_{DP} and Z_{DR} using variable, storm-specific thresholds in a manually defined area of interest around each

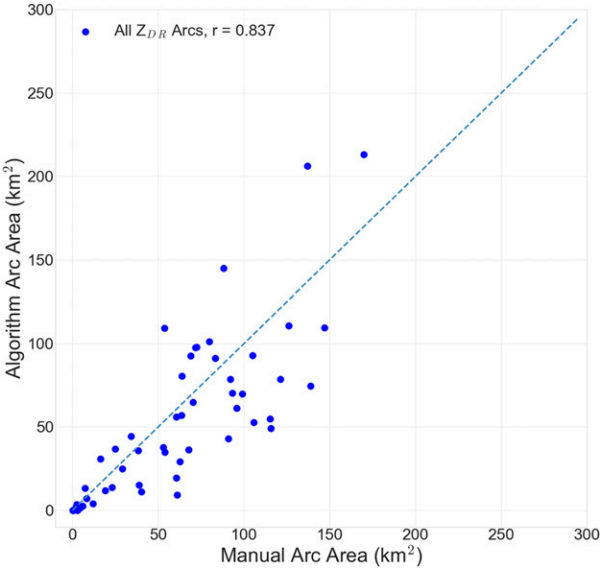

FIG. 7. Comparison between algorithm-derived and manual storm mean Z_{DR} arc areal extents. Dashed blue line is a 1:1 line along which a perfect match would fall.

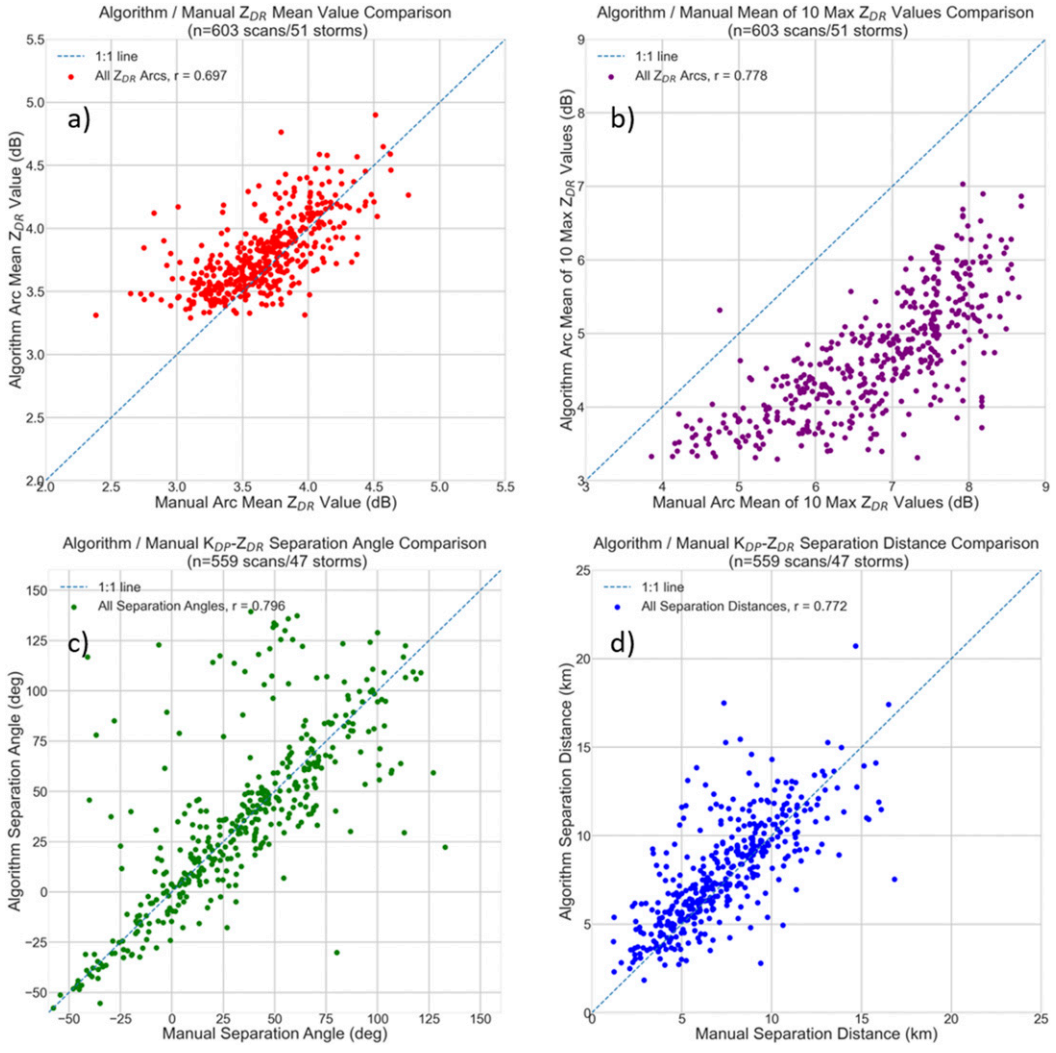

FIG. 8. As in Fig. 6, but for (a) mean arc Z_{DR} value, (b) the mean of the 10 maximum gridpoint Z_{DR} values in the arc, (c) separation angle, and (d) separation vector length.

storm. Both the threshold approach used in this algorithm for identifying K_{DP} and Z_{DR} enhancements and the dynamic approach used by Loeffler and Kumjian (2018) have their advantages and disadvantages. Exact thresholds have the advantage of being relatively simple to implement, easy to visualize and compare to features observable in raw radar data, and easy to verify against manual analyses. Dynamic approaches have the advantage of being able to analyze K_{DP}–Z_{DR} separation signatures even in environments where the Z_{DR} values in the arc do not reach a default threshold used to define the arc (3.25 dB in our case) and being unaffected by problems with Z_{DR} calibration. However, consistently quantifying Z_{DR} arc characteristics such as arc area or mean arc Z_{DR} value for work comparing storms in different environments would be difficult with such an approach, as the value of Z_{DR} that defines the arc would be constantly changing. Thus, an exact threshold was chosen for this work in order to make quantification of such characteristics possible and to allow a comparison with the manual Z_{DR} arc analyses from VDB16 and VDB17.

d. Algorithm user interface and output

The final version of the algorithm can be run using a graphical user interface written in Python. It requires users to enter a radar site, FFD angle, and storm motion estimate for real-time cases, and additionally information on a time window of interest for historic cases. In addition, options are provided to modify the thresholds used to define the Z_{DR} arc and K_{DP} foot objects as well as the reflectivity thresholds used for the tracking algorithm. Once these variables are set and the program is started, it automatically downloads level II radar files and runs them through the algorithm, saving output to a Pandas dataframe and a placefile that can be displayed in GR2 Analyst. On a typical laptop or desktop computer, the algorithm can process a radar scan in as short as 30 s after it is downloaded, meaning

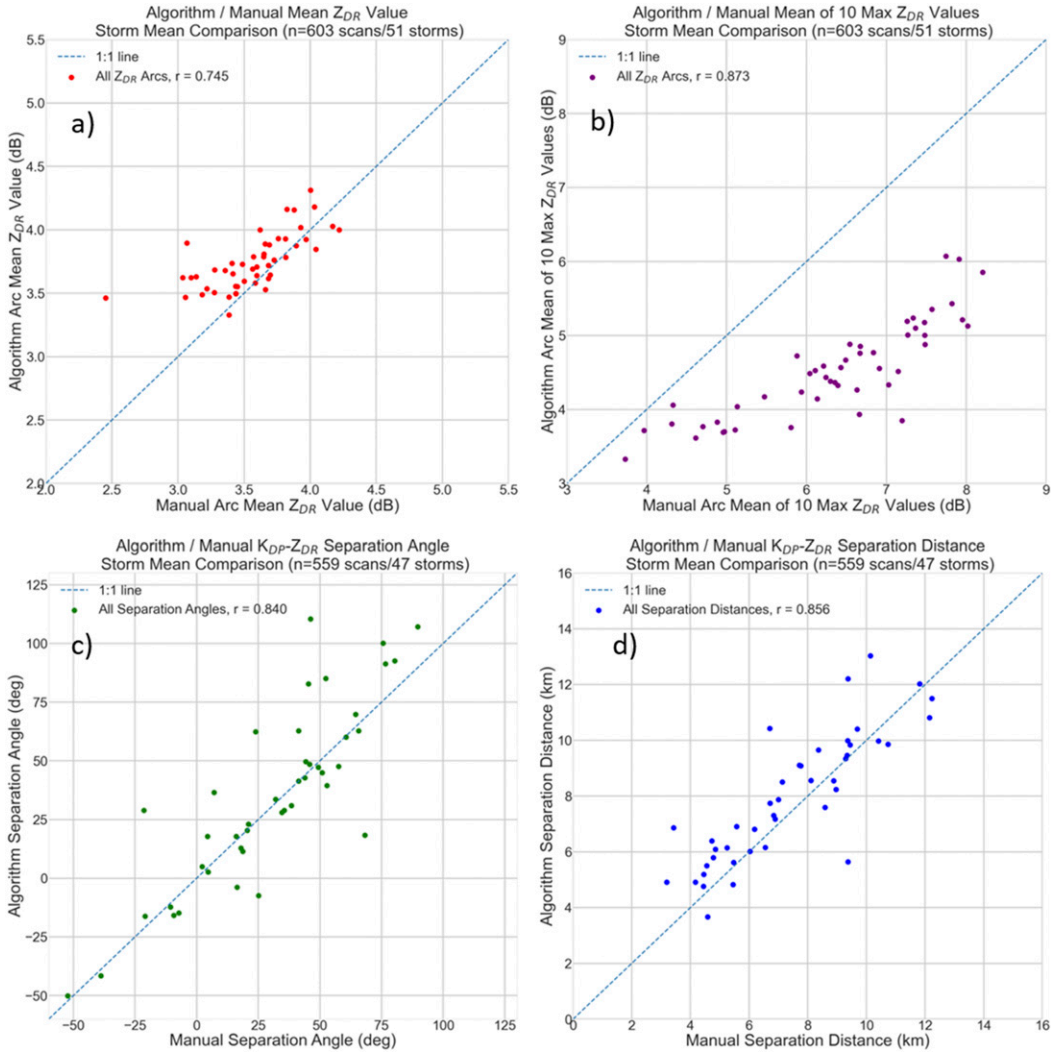

FIG. 9. As in Fig. 7, but for storm mean values for (a) mean arc Z_{DR} value, (b) the mean of the 10 maximum gridpoint Z_{DR} values in the arc, (c) separation angle, and (d) separation vector length.

that the placefile output can be viewed in near–real time in GR2 Analyst.

3. Results

a. Overall algorithm performance

The final Z_{DR} arc detection and tracking algorithm was run on the 51 storms from the VDB16 and VDB17 dataset, and the arc areas output by the algorithm were compared to the manual 3.5 dB arc areas for each storm calculated by VDB16 and VDB17 (Fig. 6a). Overall, the algorithm performed fairly well, with a correlation of Spearman's $r = 0.830$ between algorithm-calculated and manual arc areas (throughout this paper, r values and p values shown for correlations are calculated using a Spearman's rank-order correlation due to the non-Gaussian distribution of the Z_{DR} arc metrics, following VDB16). However, the algorithm struggled with some cases, overpredicting areal extents in many of the larger arcs and underpredicting smaller arc areas or failing to detect them entirely. Some of the mismatch between the algorithm and manual arc areas may be due to the inherent subjectivity of manual arc area calculations, especially in cases where the arc is small or ill defined. Comparing storm-average manual and algorithm arc areas produced a slightly larger correlation of $r = 0.837$ (Fig. 7a).

The algorithm was also validated against manual calculations for the arc intensity metrics (mean arc Z_{DR} and the mean of the arc's 10 maximum Z_{DR} pixels) and the separation angle. Since these values were not available from the VDB16 and VDB17 dataset, manual 3.5 dB Z_{DR} arc and K_{DP} foot polygons were drawn for each storm in Python by using the ipyleaflet module (https://ipyleaflet.readthedocs.io/en/latest/) to project the radar data (archived level II data from https://aws.amazon.com/public-datasets/nexrad/ for Z and Z_{DR}, archived level III data from NCEI for K_{DP}) onto an interactive map and draw the polygons on it. Once drawn, the latitude and longitude

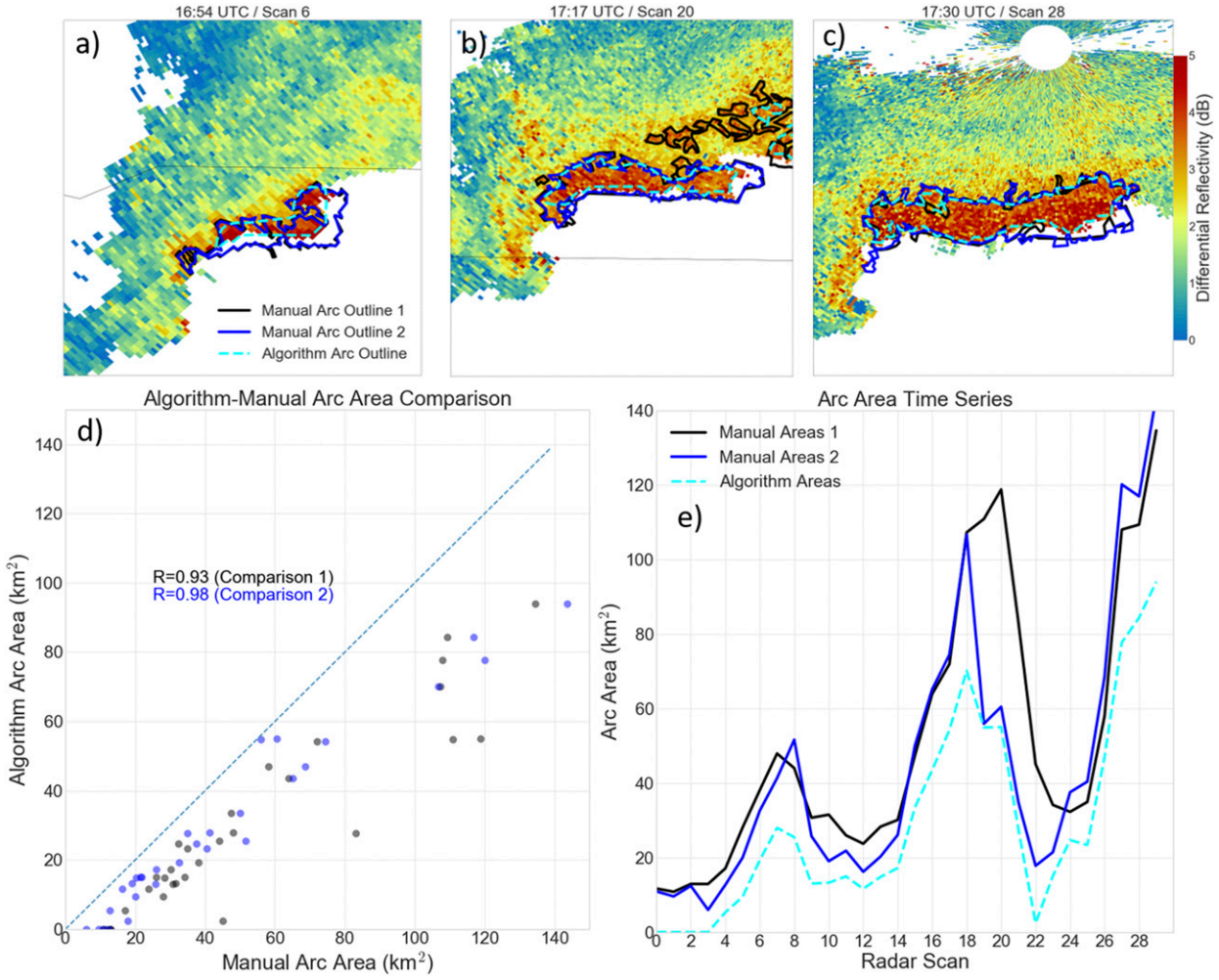

FIG. 10. A comparison of manual and algorithm Z_{DR} arc identifications for the 16 Dec 2019 DeRidder tornadic supercell, showing (a)–(c) manual (blue and black solid) and algorithm (dashed cyan) arc outlines for three selected times; (d) a scatterplot comparing manual and algorithm arc areas; (e) manual and algorithm-generated arc area time series.

coordinates of the outline of each polygon were then saved and loaded into another script that calculated the metrics to be verified using the manual arc and foot outlines, the original radar data, and (for the separation angle analysis) the observed storm motion for each storm. The mean Z_{DR} value for each manually drawn arc polygon was calculated as the mean Z_{DR} value for all pixels from the original radar scan included within the polygon, the mean of the 10 maximum arc Z_{DR} values was calculated as the mean of the 10 maximum Z_{DR} pixels within the arc polygon, the separation angle was calculated as the angle between the storm motion vector and the separation vector between the centroids of the manually drawn K_{DP} foot and Z_{DR} arc polygons, and the separation distance as the length of that vector. Four storms did not have the required level III K_{DP} data available in the NCEI level III archive and were thus not used for the separation angle or distance validation. As seen in Fig. 8, the algorithm performed fairly well for all four of these metrics, with particularly good performance demonstrated in calculating the separation angle and separation distance (Figs. 8c,d). A notable low bias in the algorithm values for the mean of the 10 maximum gridpoint Z_{DR} values in the arc (Fig. 8b) can likely be explained as a consequence of the gridding applied to the data used by the algorithm, which likely smooths out the impact of some of the individual pixels with very high Z_{DR} values in the raw data. As with the arc area verification, performance for all metrics improves when considering storm-average values instead of individual radar scans (Fig. 9).

Although this algorithm performed fairly well for supercell storms, it is important to note that it was only designed to operate on supercells. Areas of enhanced Z_{DR} in nonsupercell storms are fairly common, with patches of high Z_{DR} along the leading edge of squall lines and areas of high Z_{DR} beneath developing updrafts in relatively disorganized storms [such as those identified by the Thunderstorm Risk Estimation from Nowcasting Development via Size Sorting (TRENDSS) algorithm; Kingfield and Picca 2018] often identified as arcs if they are in a similar location relative to the storm centroid as the arc

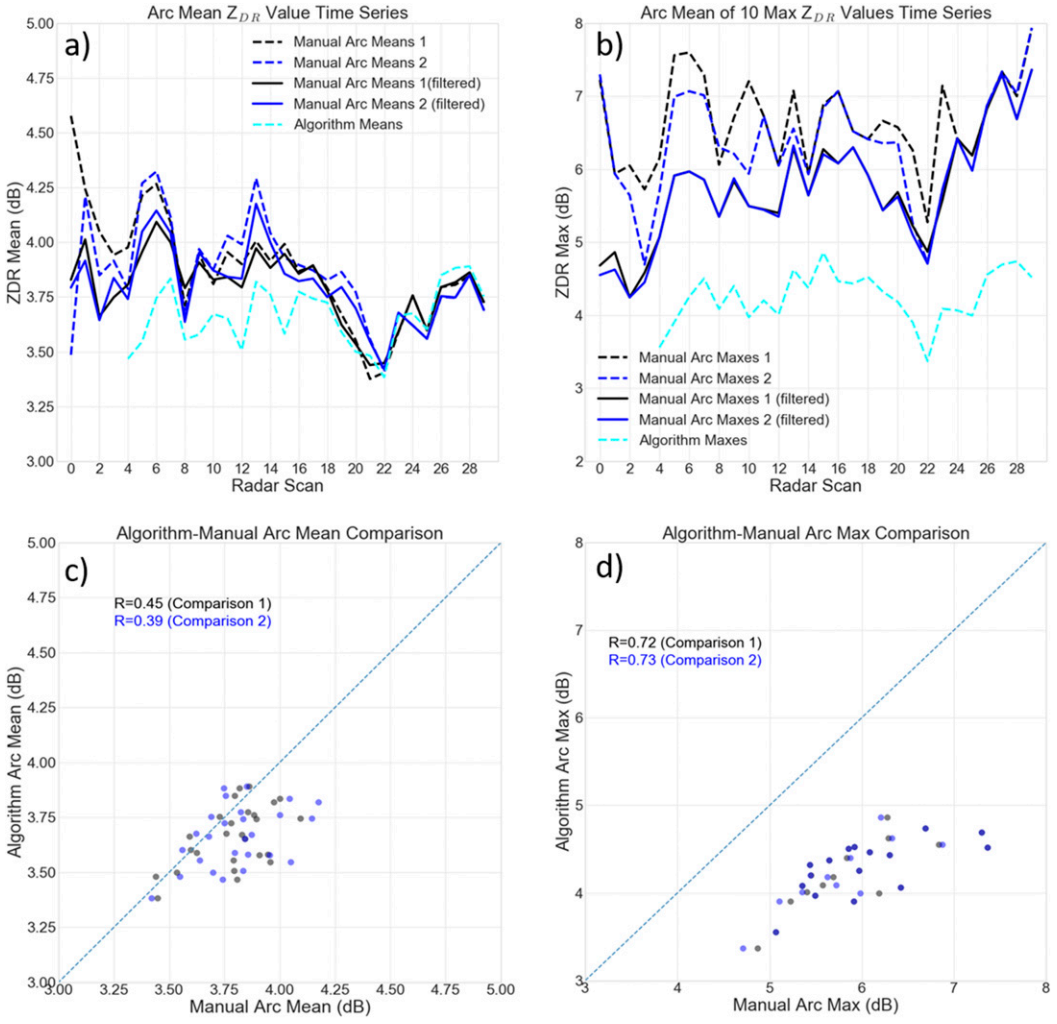

FIG. 11. A comparison of manual and algorithm Z_{DR} arc intensity metrics for the 16 Dec 2019 DeRidder tornadic supercell, showing (a) time series of manual and algorithm-calculated arc mean Z_{DR} values, (b) time series of the mean of the 10 maximum Z_{DR} values in manually analyzed and algorithm-identified arcs, (c) a scatterplot of manual and algorithm arc mean Z_{DR} values, and (d) a scatterplot of the mean of the 10 maximum Z_{DR} values in the manually drawn and algorithm-analyzed arc objects.

would be in a supercell. Some of these signatures may represent inflow enhancements similar to the Z_{DR} arc [especially in more organized convective modes, such as those noted by Crowe et al. (2012) in a tornadic quasi-linear convective system (QLCS)] while others may just be due to size sorting beneath developing updrafts. Future work will focus on better characterizing these signatures and determining which may be false detections and which may provide potentially useful information.

b. *Detailed case study: 16 December 2019*

To further explore the algorithm's performance, a detailed case study was conducted using WSR-88D data from a tornadic supercell near DeRidder, Louisiana, on 16 December 2019, which had a well-defined Z_{DR} arc for much of its lifetime. Both authors independently used level II Z_{DR} and Z_{HH} data to draw their own arc outlines in QGIS (QGIS Development Team 2019) and arc outline shapefiles were saved for each radar scan from 1643 to 1731 UTC 16 December 2019. To avoid potentially biasing the results toward the algorithm analysis, both sets of shapefiles were drawn before looking at any algorithm output for this storm. Arc area and intensity metrics were then calculated from each polygon as described in section 3a. These metrics were then compared to the algorithm-calculated metrics for this storm. A time series and scatterplot comparison of the manual and algorithm arc areas is shown in Figs. 10b and 10c, and plots of the arc outlines from the manual and algorithm analyses from several different times are shown in Fig. 10a. Figure 11 shows similar time series comparisons and scatterplots for mean arc Z_{DR} value and the mean of each arc's 10 maximum gridpoint Z_{DR} values.

Overall the algorithm calculated arc area quite well in this case, having correlations of $r = 0.93$ and $r = 0.98$ between the

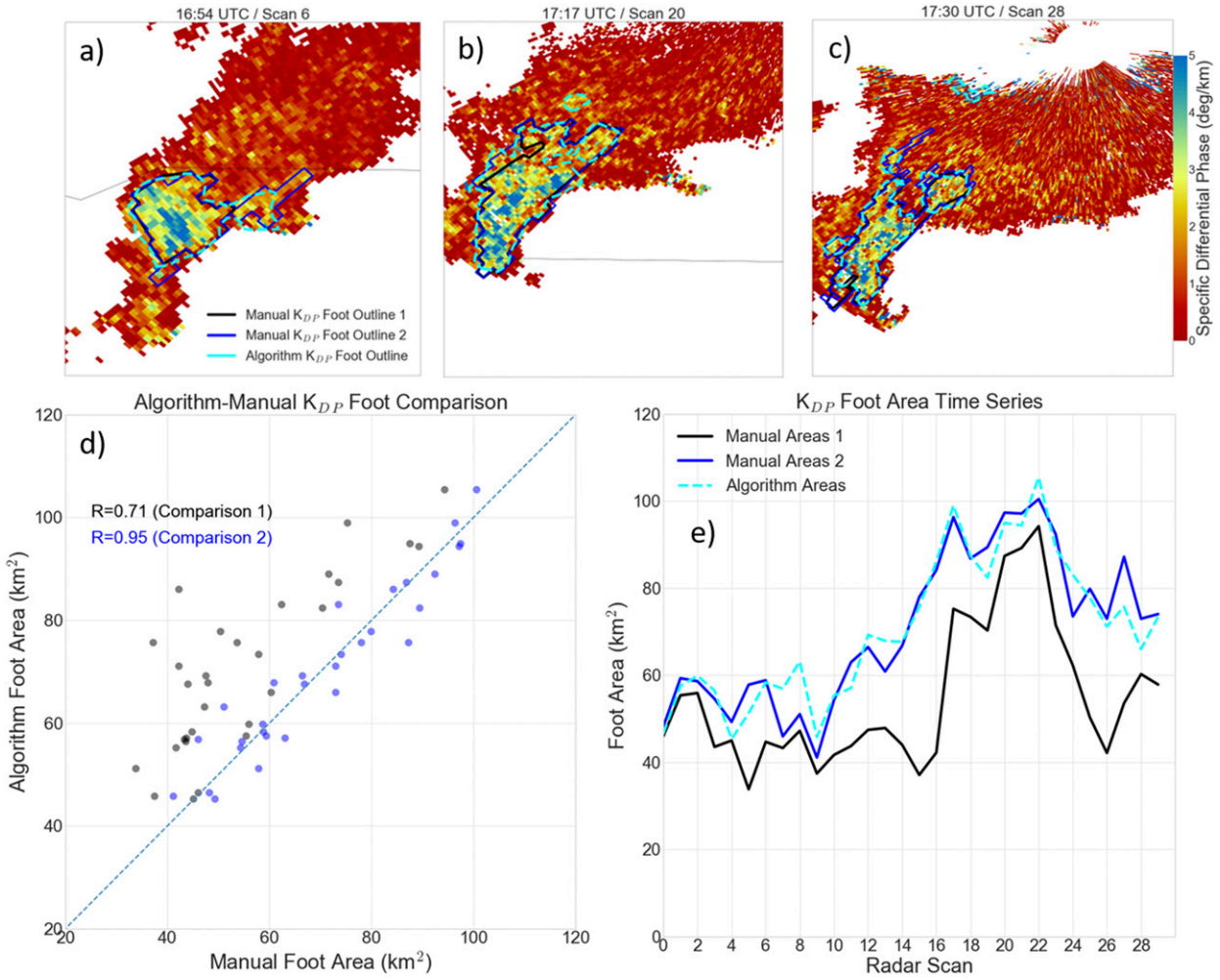

FIG. 12. A comparison of manual and algorithm K_{DP} foot identifications for the 16 Dec 2019 DeRidder tornadic supercell, showing (a)–(c) manual (blue and black solid) and algorithm (dashed cyan) K_{DP} foot outlines for three selected times; (d) a scatterplot comparing manual and algorithm K_{DP} foot areas; (e) manual and algorithm-generated K_{DP} foot area time series.

manual and algorithm arc areas and capturing temporal changes in the manual arc area time series well. Examination of arc identifications on individual radar scans shows that the algorithm struggles to properly identify the arc in the storm's early stages when the arc signature was smaller and more nebulous, but it performed well once the arc became large and clearly defined. The biggest differences between the algorithm and manual analyses often occurred when the Z_{DR} arc was relatively disorganized and split into multiple separate regions of higher Z_{DR} (such as in Fig. 10b); however, these situations produced the most substantial differences between the two manual analyses as well. Intensity changes in the arc were harder to capture, as reflected in the much lower correlations in the initial comparisons between the manual and algorithm values for these metrics ($r = 0.27$ and $r = 0.27$ for arc mean Z_{DR}, $r = 0.44$ and $r = 0.69$ for the mean of the 10 maximum Z_{DR} pixels in the arc, dashed blue and black time series in Fig. 11). Filtering the Z_{DR} pixels in the manual arc outlines to remove those with Z_{HH} values below 20 dBZ to reflect the removal of those pixels in the algorithm's quality control process greatly improved the correlations ($r = 0.45$ and $r = 0.39$ for arc mean Z_{DR}, $r = 0.72$ and $r = 0.73$ for the mean of the 10 maximum arc Z_{DR} values, Fig. 11). The overall low bias in the algorithm arc intensity metrics compared to the manual analysis is likely due to the gridding used in the algorithm calculations, which may smooth out higher Z_{DR} values by averaging them with other pixels within the radius of influence used for the Barnes analysis in PyART. Despite this low bias, the algorithm does capture the notable minimum in arc intensity well in both intensity metrics (Figs. 11a,b).

To evaluate the algorithm's performance in calculating the angle between the K_{DP}–Z_{DR} separation vector and the storm motion vector, manual K_{DP} foot outlines were again drawn in QGIS by both authors for each radar scan used in the Z_{DR} arc analysis above and saved as shapefiles. K_{DP}–Z_{DR} separation metrics were then calculated using the polygon centroids and observed storm motion as in section 3a. These values were then compared to algorithm-calculated separation angles as shown

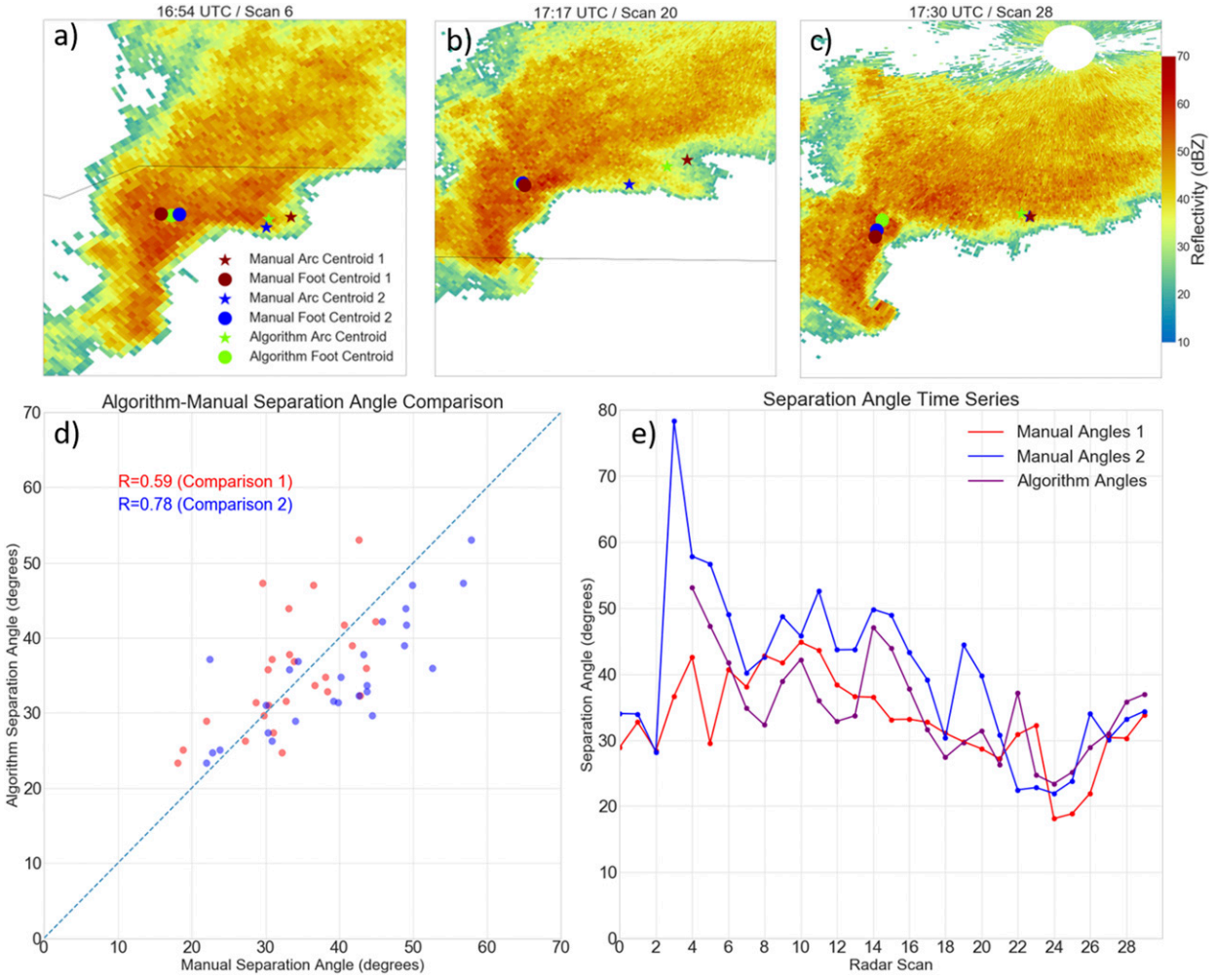

FIG. 13. A comparison of manual and algorithm K_{DP}–Z_{DR} separation signature identifications for the 16 Dec 2019 DeRidder tornadic supercell, showing (a)–(c) manual and algorithm K_{DP} foot and Z_{DR} arc centroids for three selected times; (d) a scatterplot comparing manual and algorithm separation angles; (e) manual and algorithm-generated separation angle time series.

in Fig. 13. The algorithm generally does well in identifying the location and size of the K_{DP} foot when compared to both manual analyses, as seen in Figs. 12a–c. However, it occasionally struggles to properly calculate the separation angle when the Z_{DR} arc is small, disorganized, or not detected by the algorithm, as shown at the beginning of the time series comparison in Fig. 13e. The algorithm and manual analyses also tend to agree fairly well on separation angle changes over time, with both manual analyses and the algorithm analysis capturing the gradual decrease in separation angle magnitude from scans 10 through 24 and the subsequent increase in separation angle (Fig. 13e). However, the algorithm tends to better match the short-term trends seen in the second manual analysis as opposed to the first.

c. *Algorithm limitations*

Although the automated Z_{DR} arc and K_{DP}–Z_{DR} separation signature detection and tracking algorithm described in this study performs well with most of the supercells in the dataset used to test it and often appears to qualitatively capture the extent of the Z_{DR} arc and K_{DP} foot well, some recurrent biases and limitations of the algorithm do exist, as shown in Fig. 14. Storms with large numbers of high-Z_{DR} pixels in their rear-flank downdraft regions occasionally have erroneous arc detections there (Fig. 14a), and these became particularly difficult to remove in cases where the region of high Z_{DR} in the arc was connected with high Z_{DR} values in the storm's rear (Fig. 14c). Since the algorithm identifies possible arc objects by breaking a contour of quality-controlled Z_{DR} at 3.25 dB into polygons and eliminates erroneous polygons with a random forest classifier, a polygon that contains both an actual arc and a spurious arc detection results in either an excessively large arc area if it is classified as an arc or a small or missing arc if it is classified as a nonarc area of high Z_{DR}. The algorithm also relies on a relatively rudimentary storm tracking algorithm to produce storm objects to which potential arc objects are assigned, and a missed or incorrectly tracked storm can occasionally cause arc objects to be lost or assigned to a storm other

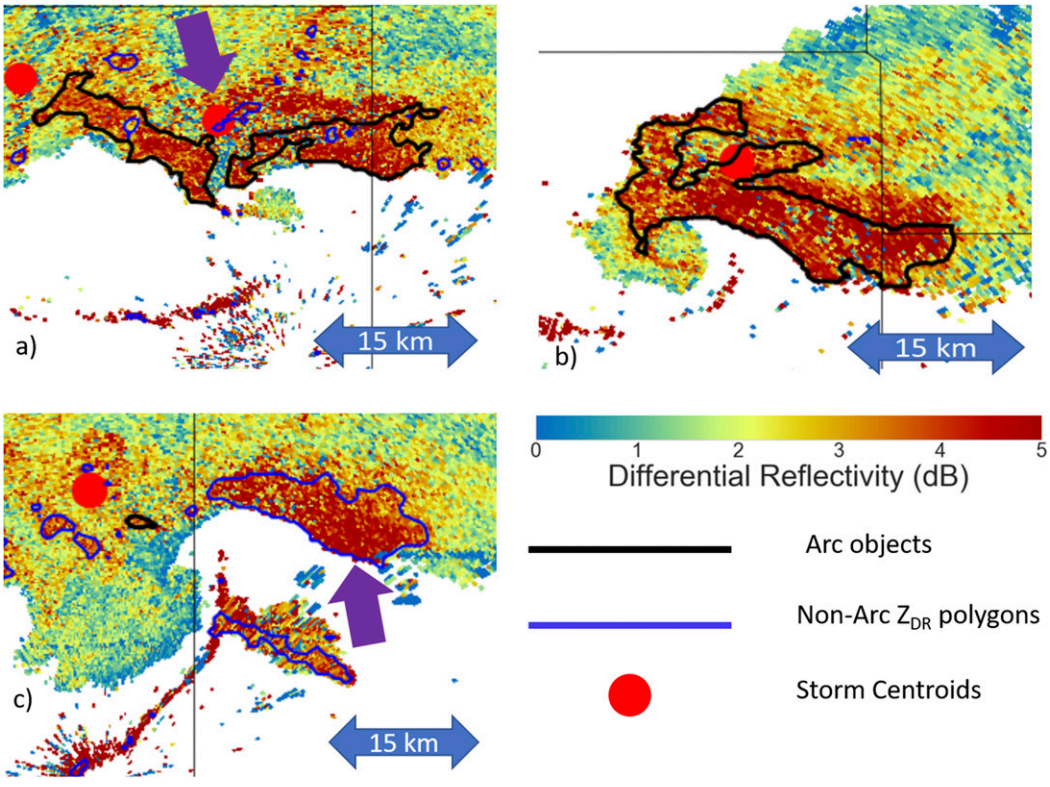

FIG. 14. Examples of Z_{DR} arc algorithm failure cases. (a) A manually defined Z_{DR} arc is incorrectly combined with a separate area of high Z_{DR} from a trailing storm. The purple arrow points toward the centroid of the storm of interest. (b) An area of high Z_{DR} in the storm's rear flank that is contiguous with the arc is erroneously included in the final arc object. (c) A well-defined arc (identified by the purple arrow) is misclassified by the random forest.

than the storm they are actually associated with. Arc objects on extremely large storms can also be missed when the random forest considers their distance from the storm centroid to be too great (Fig. 14b). Finally, the algorithm has been designed mainly for work with supercell storms and tends to produce spurious Z_{DR} arc and K_{DP} foot objects with nonsupercell storm modes. Future improvements to the algorithm, including the adoption of an improved storm tracking algorithm, should help mitigate these issues.

4. Summary

The Z_{DR} arc and K_{DP}–Z_{DR} separation signatures represent dual-polarization evidence of the degree of hydrometeor size sorting occurring in a supercell's forward flank. Previous work (Kumjian and Ryzhkov 2008, 2009; Kumjian et al. 2010; Palmer et al. 2011; Crowe et al. 2012; Loeffler and Kumjian 2018; Loeffler et al. 2020) indicates that these signatures may be useful for inferring the magnitude of low-level shear and storm-relative helicity in a supercell's environment, and potentially even in differentiating between tornadic and nontornadic storms. However, quantifying characteristics of these signatures can be time-consuming and can thus present difficulties in both research and operations. This paper presents an automated algorithm for the identification and quantification of Z_{DR} arc and K_{DP}–Z_{DR} separation signatures. A comparison of manual and algorithm-calculated Z_{DR} arc areas from the 51 supercells considered in this paper shows that this algorithm can reliably detect Z_{DR} arcs and that algorithm arc area calculations are similar to those produced by manual analysis of WSR-88D data. Furthermore, the detailed analysis of the 16 December 2019 DeRidder, Louisiana, supercell indicates that the algorithm may be able to accurately capture time trends in arc areal extent and the separation angle between the K_{DP}–Z_{DR} separation and storm motion vectors.

Future work will focus on using this algorithm to examine Z_{DR} arcs and K_{DP}–Z_{DR} separation signatures in a large sample of supercells. This analysis will be used to further explore whether these signatures exhibit any reliable trends before tornadogenesis or tornadogenesis failure and how these signatures vary in different near-storm environments. In addition, improvements will be made to the algorithm to reduce false Z_{DR} arc detections, especially in nonsupercell storms. Finally, the storm-tracking and object-identification frameworks used in this algorithm could be used to build automated algorithms to identify and quantify the characteristics of other supercell polarimetric signatures that may be useful in warning operations, such as Z_{DR} column area and depth or polarimetrically inferred hailfall area. The code for the algorithm presented here is available online (at https://github.com/mwilson14/ZDRArcAlgorithm), and suggestions for new features, improvements, or bug reports from the community are welcome.

Acknowledgments. The authors were supported by NSF Grant 11A-1539070 and NOAA Grant NA18OAR4590307. The second author was also supported by an academic appointment at the University of Nebraska–Lincoln. The authors would also like to thank Nicholas Humrich for helpful contributions to streamlining the algorithm code, Stephen Shield for advice on evaluating the random forest algorithm's performance, and the three anonymous reviewers for their feedback, which has greatly improved this manuscript.

Data availability statement. All Python scripts and data files used in the analysis described in this paper are available in a Github repository at https://github.com/mwilson14/ZDRArcAlgorithm, along with the algorithm code itself.

REFERENCES

Brandes, E. A., J. Vivekanandan, J. D. Tuttle, and C. J. Kessinger, 1995: A study of thunderstorm microphysics with multiparameter radar and aircraft observations. *Mon. Wea. Rev.*, **123**, 3129–3143, https://doi.org/10.1175/1520-0493(1995)123<3129:ASOTMW>2.0.CO;2.

Breiman, L., 2001: Random forests. *Mach. Learn.*, **45**, 5–32, https://doi.org/10.1023/A:1010933404324.

Crowe, C. C., W. A. Petersen, L. D. Carey, and D. J. Cecil, 2010: A dual-polarization investigation of tornado-warned cells associated with Hurricane Rita (2005). *Electron. J. Oper. Meteor.*, **11** (4), https://nwafiles.nwas.org/ej/pdf/2010-EJ4.pdf.

——, C. Schultz, M. Kumjian, L. Carey, and W. Petersen, 2012: Use of dual-polarization signatures in diagnosing tornadic potential. *Electron. J. Oper. Meteor.*, **13** (5), http://nwafiles.nwas.org/ej/pdf/2012-EJ5.pdf.

Dawson, D. T., II, E. R. Mansell, Y. Jung, L. J. Wicker, M. R. Kumjian, and M. Xue, 2014: Low-level Z_{DR} signatures in supercell forward flanks: The role of size sorting and melting of hail. *J. Atmos. Sci.*, **71**, 276–299, https://doi.org/10.1175/JAS-D-13-0118.1.

——, ——, and M. R. Kumjian, 2015: Does wind shear cause hydrometeor size sorting? *J. Atmos. Sci.*, **72**, 340–348, https://doi.org/10.1175/JAS-D-14-0084.1.

Duda, J., and W. Gallus, 2010: Spring and summer midwestern severe weather reports in supercells compared to other morphologies. *Wea. Forecasting*, **25**, 190–206, https://doi.org/10.1175/2009WAF2222338.1.

Géron, A., 2017: *Hands-On Machine Learning with Scikit-Learn and Tensor Flow*. 1st ed. O'Reilly, 543 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.

Helmus, J. J., and S. M. Collis, 2016: The Python ARM radar toolkit (Py-ART), a library for working with weather radar data in the Python programming language. *J. Open Res. Software*, **4**, e25, https://doi.org/10.5334/jors.119.

Herman, G. R., and R. S. Schumacher, 2018: Money doesn't grow on trees, but forecasts do: Forecasting extreme precipitation with random forests. *Mon. Wea. Rev.*, **146**, 1571–1600, https://doi.org/10.1175/MWR-D-17-0250.1.

Homeyer, C. R., T. N. Sandmæl, C. K. Potvin, and A. M. Murphy, 2020: Distinguishing characteristics of tornadic and nontornadic supercell storms from composite mean analyses of radar observations. *Mon. Wea. Rev.*, **148**, 5015–5040, https://doi.org/10.1175/MWR-D-20-0136.1.

Johnson, A., 2018: Assessing storm motion to maximize storm relative helicity: Part II: Case examples of using the separation vector orientation to qualitatively estimate SRH. YouTube, https://www.youtube.com/watch?time_continue=2&v=LCKeC2KxU7I&feature=emb_logo.

Jurewicz, M., and C. Gitro, 2018: Incorporating dual-polarization signatures into the tornado warning process: KDP/ZDR separation signals, ZDR arc considerations, and initial results of hook echo investigations. *Eighth Conf. on Transition of Research to Operation*, Austin, TX, Amer. Meteor. Soc., 12B2, https://ams.confex.com/ams/98Annual/webprogram/Paper330719.html.

Kingfield, D. M., and J. C. Picca, 2018: Development of an operational convective nowcasting algorithm using raindrop size sorting information from polarimetric radar data. *Wea. Forecasting*, **33**, 1477–1495, https://doi.org/10.1175/WAF-D-18-0025.1.

Kuhn, M., and K. Johnson, 2013: *Applied Predictive Modeling*. 1st ed. Springer, 600 pp.

Kumjian, M. R., 2011: Precipitation properties of supercell hook echoes. *Electron. J. Severe Storms Meteor.*, **5** (6), https://ejssm.org/ojs/index.php/ejssm/article/viewArticle/93.

——, and A. V. Ryzhkov, 2008: Polarimetric signatures in supercell thunderstorms. *J. Appl. Meteor. Climatol.*, **47**, 1940–1961, https://doi.org/10.1175/2007JAMC1874.1.

——, and ——, 2009: Storm-relative helicity revealed from polarimetric radar measurements. *J. Atmos. Sci.*, **66**, 667–685, https://doi.org/10.1175/2008JAS2815.1.

——, and ——, 2010: The impact of evaporation on polarimetric characteristics of rain: Theoretical model and practical implications. *J. Appl. Meteor. Climatol.*, **49**, 1247–1267, https://doi.org/10.1175/2010JAMC2243.1.

——, ——, V. M. Melnikov, and T. J. Schuur, 2010: Rapid-scan super-resolution observations of a cyclic supercell with a dual-polarization WSR-88D. *Mon. Wea. Rev.*, **138**, 3762–3786, https://doi.org/10.1175/2010MWR3322.1.

——, A. P. Khain, N. Benmoshe, E. Ilotoviz, A. V. Ryzhkov, and V. T. Phillips, 2014: The anatomy and physics of Z_{DR} columns: Investigating a polarimetric radar signature with a spectral bin microphysical model. *J. Appl. Meteor. Climatol.*, **53**, 1820–1843, https://doi.org/10.1175/JAMC-D-13-0354.1.

Kuster, C. M., J. C. Snyder, T. J. Schuur, T. T. Lindley, P. L. Heinselman, J. C. Furtado, J. W. Brogden, and R. Toomey, 2019: Rapid-update radar observations of Z_{DR} column depth and its use in the warning decision process. *Wea. Forecasting*, **34**, 1173–1188, https://doi.org/10.1175/WAF-D-19-0024.1.

Loeffler, S. D., and M. R. Kumjian, 2018: Quantifying the separation of enhanced Z_{DR} and K_{DP} regions in nonsupercell tornadic storms. *Wea. Forecasting*, **33**, 1143–1157, https://doi.org/10.1175/WAF-D-18-0011.1.

——, ——, M. Jurewicz, and M. M. French, 2020: Differentiating between tornadic and nontornadic supercells using polarimetric radar signatures of hydrometeor size sorting. *Geophys. Res. Lett.*, **47**, e2020GL088242, https://doi.org/10.1029/2020GL088242.

Martinaitis, S. M., 2017: Radar observations of tornado-warned convection associated with tropical cyclones over Florida. *Wea. Forecasting*, **32**, 165–186, https://doi.org/10.1175/WAF-D-16-0105.1.

McKinney, 2010: Data structures for statistical computing in Python. *Proc. Ninth Python in Science Conf.*, Austin, TX, SciPy, 56–61, https://doi.org/10.25080/Majora-92bf1922-00a.

Mecikalski, J. R., J. K. Williams, C. P. Jewett, D. Ahijevych, A. LeRoy, and J. 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.

Palmer, R. D., and Coauthors, 2011: Observations of the 10 May 2010 tornado outbreak using OU-PRIME: Potential for new science with high-resolution polarimetric radar. *Bull. Amer. Meteor. Soc.*, **92**, 871–891, https://doi.org/10.1175/2011BAMS3125.1.

Pedregosa, F., and Coauthors, 2011: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Picca, J., and A. Ryzhkov, 2012: A dual-wavelength polarimetric analysis of the 16 May 2010 Oklahoma City extreme hailstorm. *Mon. Wea. Rev.*, **140**, 1385–1403, https://doi.org/10.1175/MWR-D-11-00112.1.

QGIS Development Team, 2019: QGIS geographic information system. Open Source Geospatial Foundation Project, https://qgis.osgeo.org.

Romine, G. S., D. W. Burgess, and R. B. Wilhelmson, 2008: A dual-polarization-radar-based assessment of the 8 May 2003 Oklahoma City area tornadic supercell. *Mon. Wea. Rev.*, **136**, 2849–2870, https://doi.org/10.1175/2008MWR2330.1.

Ryzhkov, 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.

Seliga, T. A., and V. N. Bringi, 1976: Potential use of radar differential reflectivity measurements at orthogonal polarizations for measuring precipitation. *J. Appl. Meteor.*, **15**, 69–76, https://doi.org/10.1175/1520-0450(1976)015<0069:PUORDR>2.0.CO;2.

Snyder, J. C., and A. V. Ryzhkov, 2015: Automated detection of polarimetric tornadic debris signatures using a hydrometeor classification algorithm. *J. Appl. Meteor. Climatol.*, **54**, 1861–1870, https://doi.org/10.1175/JAMC-D-15-0138.1.

——, ——, M. R. Kumjian, A. P. Khain, and J. Picca, 2015: A Z_{DR} column detection algorithm to examine convective storm updrafts. *Wea. Forecasting*, **30**, 1819–1844, https://doi.org/10.1175/WAF-D-15-0068.1.

——, H. B. Bluestein, D. T. Dawson, and Y. Jung, 2017: Simulations of polarimetric, X-band radar signatures in supercells. Part II: Z_{DR} columns and rings and KDP columns. *J. Appl. Meteor. Climatol.*, **56**, 2001–2026, https://doi.org/10.1175/JAMC-D-16-0139.1.

Van Den Broeke, M. S., 2016: Polarimetric variability of classic supercell storms as a function of environment. *J. Appl. Meteor. Climatol.*, **55**, 1907–1925, https://doi.org/10.1175/JAMC-D-15-0346.1.

——, 2017: Polarimetric radar metrics related to tornado life cycles and intensity in supercell storms. *Mon. Wea. Rev.*, **145**, 3671–3686, https://doi.org/10.1175/MWR-D-16-0453.1.

——, 2020: A preliminary polarimetric radar comparison of pretornadic and nontornadic supercell storms. *Mon. Wea. Rev.*, **148**, 1567–1584, https://doi.org/10.1175/MWR-D-19-0296.1.

——, and S. T. Jauernic, 2014: Spatial and temporal characteristics of polarimetric tornadic debris signatures. *J. Appl. Meteor. Climatol.*, **53**, 2217–2232, https://doi.org/10.1175/JAMC-D-14-0094.1.

——, and C. A. Van Den Broeke, 2015: Polarimetric radar observations from a waterspout-producing thunderstorm. *Wea. Forecasting*, **30**, 329–348, https://doi.org/10.1175/WAF-D-14-00114.1.

——, J. M. Straka, and E. N. Rasmussen, 2008: Polarimetric radar observations at low levels during tornado life cycles in a small sample of classic southern plains supercells. *J. Appl. Meteor. Climatol.*, **47**, 1232–1247, https://doi.org/10.1175/2007JAMC1714.1.