# Using Machine Learning Techniques for Supercell Tornado Prediction with Environmental Sounding Data

CyberTraining: Big Data + High-Performance Computing + Atmospheric Sciences

Brice Coffer<sup>1</sup>, Michaela Kubacki<sup>2</sup>, Yixin Wen<sup>3</sup>, Ting Zhang<sup>4</sup>, Research assistant: Carlos A. Barajas<sup>5</sup>, Faculty mentor: Matthias K. Gobbert<sup>5</sup>

<sup>1</sup>Department of Marine, Earth, and Atmospheric Science, North Carolina State University <sup>2</sup>Department of Mathematics, Middlebury College

<sup>3</sup>Cooperative Institute for Mesoscale Meteorological Studies, University of Oklahoma, and NOAA/National Severe Storms Laboratory, Norman, Oklahoma

<sup>4</sup>Department of Mathematics and Computer Science, McDaniel College

<sup>5</sup>Department of Mathematics and Statistics, University of Maryland, Baltimore County

Technical Report HPCF-2020-18, hpcf.umbc.edu > Publications

#### Abstract

Tornadoes pose a forecast challenge to National Weather Service forecasters because of their quick development and potential for life-threatening damage. The use of machine learning in severe weather forecasting has recently garnered interest, with current efforts mainly utilizing ground weather radar observations. In this study, we investigate machine learning techniques to discriminate between nontornadic and tornadic storms solely relying on the Rapid Update Cycle (RUC) sounding data that represent the pre-storm atmospheric conditions. This approach aims to provide for early warnings of tornadic storms, before they form and are detectable by weather radar observations. Two machine learning methods tested in our project are Random Forest (RF) and Convolutional Neural Network (CNN). Performance testing of RF using various ranges of hyperparameters results in an overall accuracy score of 70.14%, but the accuracy of significantly tornadic class prediction is only 23.84%. The CNN model results in an overall accuracy score of 67.84%, but the accuracy for significantly tornadic storms is only 26.69%. The higher accuracy in the RF and CNN models for the majority class of nontornadic supercells suggests that the imbalanced dataset is a meaningful contributor to the lower accuracy for tornadic storms. After applying the simple method of randomly undersampling (oversampling) the majority (minority) class, the accuracies of significantly tornadic class prediction of RF and CNN are enhanced to 65.85% and 36.01%, respectively. Future work should investigate alternative methods of dealing with imbalanced datasets in a CNN, including more sophisticated undersampling/oversampling techniques.

**Key words.** Tornado prediction, Rapid Update Cycle (RUC), environmental sounding data, random forest classifier, convolutional neural network (CNN).

## 1 Introduction

Tornadoes can develop quickly, cause severe damage across a large spatial area, and create lifethreatening conditions, thus posing a forecast challenge to National Weather Service (NWS) forecasters. Proximity soundings derived from model analysis data [27–29] are currently used operationally to monitor the probability of tornadogenesis. The Significant Tornado Parameter (STP) was developed as one of these tools to aid operational forecasters in the tornado forecasting process [27], and it is probably the most widely used environmental proxy for tornadic thunderstorms. The STP has been updated after more supercell research and deeper understanding of the connection between tornadogenesis and the environment factors. The majority of studies of severe weather exclusively rely on commonly used parameters that describe the environment (e.g., how much instability or wind shear is in the atmosphere).

The goal of this work is to explore the use of machine learning techniques to predict significant tornadoes using only Rapid Update Cycle (RUC) sounding data that represent the pre-storm atmospheric conditions. This approach aims to provide for early warnings of tornadic storms, before they form and are detectable by ground weather radar observations.

Two machine learning methods are considered in our project: Random Forest (RF) and Convolutional Neural Network (CNN). (i) RF is an ensemble method that uses randomized decision trees as its base models and grows many decision trees depending on a collection of random variables. A new object will be classified using the binary recursive-partitioning algorithm in the decision tree forest. RF will give estimates of what features are best in the classification and help to interpret machine learning results. (ii) CNN is a class of neural network which can take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. CNNs use relatively little pre-processing compared to other image classification algorithms with an advantage of independence from prior knowledge and human effort in feature design. CNNs are mainly used for processing data that has a grid pattern, and designed to automatically learn spatial hierarchies of features [33]. (iii) The imbalanced nature of our dataset leads to underprediction of significantly tornadic storms in both the RF and the CNN. To address the imbalanced dataset issue, approaches to randomly undersample the dominant nontornadic cases and to oversample the rare significantly tornadic cases are applied in this study.

(i) Performance testing of RF using various ranges of hyperparameters results in an overall accuracy score of 70.14%, but the accuracy of the significantly tornadic category is only 23.84%. Feature importance analysis of the RF model indicates the v-wind variable scored significantly higher than other variables, while the pressure variable importance scores consistently lower than other variables across all height levels. (ii) Performance testing of a CNN with three 1D convolutional layers trained for 100 epochs has an overall accuracy score of 67.84%, but the accuracy of the significantly tornadic category is only 26.69%. The results show that CNN outperforms current forecasting parameter in terms of predicting whether or not a supercell storm will generate some type of tornadic event but performs poorly on predicating significant tornadic events in comparison. (iii) Applying random undersampling (RUS) to RF produces the highest class accuracy for the significantly tornadic category of all models considered herein, reaching a class accuracy of 65.86%. However, it also results in the overall model accuracy decreasing as the accuracy of original majority classes, weakly/nontornadic, decreases significantly. In contrast, RF with Random oversampling (ROS) increases the accuracy of significantly tornadic events more modestly while maintaining about the same accuracy for other two classes. Random undersampling (RUS) applied to CNN also improves the accuracy for significantly tornadic events at the steep expense in the accuracy for nontornadic events. Applying random oversampling (ROS) to CNN exhibits the best balance of increased accuracy in the significantly tornadic category with less impact on other accuracy measures. With 36.01% class accuracy in the significant tornado category, this exceeds the accuracy of the base RF, random oversampling RF, and the base CNN model without random sampling.

The remainder of this work is organized as follows. In Section 2, we discuss the basic physics of tornadic and nontornadic supercells. Section 3 discuss the details of sounding data we use in this study. Section 4 introduces the details of RF and CNN, the two machine learning techniques, and the forecast skill metrics. Section 5 presents the detailed results of RF and CNN, RF feature analysis, and the random sampling of the imbalanced data. Finally, in Section 6, we present conclusions and discuss future work.



Figure 2.1: Typical supercell appearance on both radar (left) and visually while storm chasing (right). The radar image is reproduced from [20]. (left) shows radar reflectivity from Doppler Radar on Wheels (DOW) on June 09, 2010 in Goshen County, Wyoming. Wind vectors and areas of horizontal convergence are derived from multiple mobile radars (DOW 6 & 7). (right) shows a photograph of a supercell on March 12, 2012 in Greer County, Oklahoma. Yellow arrows illustrate the rotation of the mesocyclone.

# 2 The role of supercell storms in tornado forecasting

On average, approximately 1,200 tornadoes are reported in the U.S. each year, resulting in roughly 80 deaths and millions of dollars in damage. Timely and accurate predictions of these severe weather events are key to mitigating casualties. However, tornadoes are very difficult to predict, as most severe weather will not lead to tornadogenesis. Furthermore, the false alarm rate for tornado warnings in the U.S. is around 75% [7]. Such a high false alarm rate is deeply problematic, not only because false alarms result in unnecessary and sometimes costly precautions, but also because frequent false alarms may negatively impact the public trust in tornado forecasting.

Since roughly 80% of tornadic events originate from supercell storms [31], the research conducted herein will focus on identifying if a supercell will become tornadic, and, in the event that a tornado forms, whether the resulting event will be weak (F0–F1) or severe (F2–F5) in nature. Supercells are isolated storms that exhibit a rotating updraft, or mesocyclone, due to wind shear. The strength of its updraft allows a supercell to self-sustain for longer periods of time than other types of thunderstorms. In addition to tornadoes, supercells are also known to produce large hail, strong nontornadic winds, flash flooding, and dangerous lightning, and tornadoes. Most strong to severe tornadoes occur in conjunction with supercell storms [21].

Radar images allow forecasters to identify supercell storms by identifying the mesocyclonic structure. We present a sample radar image of a supercell storm in Figure 2.1, along with a photograph of a supercell. In both the radar and the photograph, rotation of the storm cell is evident, but this is not sufficient to determine whether or not a tornado will form.

Predicting which supercell storms will result in tornadogenesis is much more difficult, as only one out of every four supercells generate tornadic events [31]. In other words, while most tornadoes originate from supercells, most supercells are nontornadic. Therefore, forecasters must rely on both real-time radar data along with observational data of the storm's surrounding environment to make a prediction. Often, forecasters will use the Significant Tornado Parameter (STP), a composite index calculated using multiple parameters known to be favorable in tornadic supercells [11]. This index is given by

$$STP = \frac{MLCAPE}{1500 \text{ J kg}^{-1}} \times \frac{2000 - MLLCL}{1000 \text{ m}} \times \frac{200 + MLCIN}{150 \text{ J kg}^{-1}} \times \frac{EBWD}{20 \text{ m s}^{-1}} \times \frac{SRH500}{75 \text{ m}^2 \text{ s}^{-2}}, \quad (2.1)$$

where mixed-layer (ML) convective available potential energy (CAPE) is a measure of instability in the atmosphere, convective inhibition (CIN) is the amount of energy that prevents an air parcel from realizing said instability, lifted condensation level (LCL) is the height of the base of the cloud, effective bulk wind difference (EBWD) is the vector magnitude of the change in wind speed and direction with height, and storm-relative helicity (SRH) in the 0 to 500 m layer above the ground is a proxy for the potential of cyclonic updraft rotation in right-moving supercells. These ingredients are known to be favorable for supercells and tornadoes and have successfully been used in operational forecasting for over 15 years [27]. However, given the high false alarm rate in tornado prediction, this index alone is not sufficient.

## 3 Data

The severe weather event database<sup>1</sup> used in this study is that of [26] and [29], except expanded to include the years 2005–2017 for tornadic thunderstorms and 2005–2015 for nontornadic thunderstorms (i.e., all available severe weather events in the current SPC convective mode database). All tornado, significant hail (sighail), and significant wind (sigwind) reports are filtered for the largest magnitude report per hour on a 40 km spacing Rapid Update Cycle (RUC) model analysis grid and then assigned to the closest analysis hour. Sub-significant hail/wind events or null cases (i.e., storms without severe weather reports) are not considered due to the difficulty of subjective case identification associated with what would be an overwhelming sample size [26]. We argue that, fundamentally, discerning the differences between tornadic and significantly severe, nontornadic storms is the most interesting and challenging forecast problem.

Each severe report is assigned a storm mode classification based on archived level II WSR-88D data from NCEI, as discussed in-depth in [26]. In this study, only the right-moving (cyclonic) supercells (RMs) are considered. Nearly 90% of all significant tornadoes occur with RMs [26], while 95% of all fatalities and 92% of injuries occur with significant tornadoes. The RMs classification includes three sub-classifications: discrete cell, cell in cluster, and cell in line. This results in 9,355 tornadoes, 3,788 sigwind, and 7,051 sighail events. These 20,194 right-moving supercells consist of 10,839 significantly severe nontornadic<sup>2</sup> cases, 7,743 weakly tornadic (F0–F1) tornadic damage cases, and 1,612 significantly tornadic (F2–F5) tornado damage cases. Approximately 54% of the dataset is nontornadic because approximately 85% of supercells in nature are nontornadic [30].

Environmental base-state data corresponding to each severe report are obtained from archived vertical profiles from the SPC's mesoscale surface objective analysis [5], which itself uses the RUC model<sup>3</sup> as the background environment [4]. Profiles are interpolated to isobaric surfaces with 25-hPa vertical resolution (e.g., 1000, 975, 950, 925, etc. hPa).

<sup>&</sup>lt;sup>1</sup>Compared to *Storm Data*, this dataset has a higher standard of quality control. Careful temporal or spatial adjustments are made to a small portion of the event database to correct report errors [26].

<sup>&</sup>lt;sup>2</sup>Hereafter, "nontornadic" is understood to refer to significantly severe nontornadic supercells (i.e., storms that produce hail  $\geq 2$  in (5.04 cm) diameter hail and convective wind gusts  $\geq 65$  kt (33.4 m s<sup>-1</sup>)).

<sup>&</sup>lt;sup>3</sup>The Rapid Refresh (RAP) model replaced the RUC model at 12Z 1 May 2012.

## 4 Methods

### 4.1 Random Forest

The first machine learning method that we exploit in our project is the Random Forest algorithm. It was proposed by Leo Breiman and developed to be used for either a categorical response variable, referred as "classification", or a continuous response, referred as "regression" [6]. As the name suggests, Random Forest is an ensemble method that uses randomized decision trees as its base models. Random Forest grows many decision trees with each tree depending on a collection of random variables. To classify a new object from an input, the algorithm puts the input vector down each of the decision tree in the forest using the binary recursive partitioning algorithm [12]. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest) as the classification of the given input.

Random Forest adds additional randomness to the model while growing the trees. The "randomness" is injected into the algorithm in two ways: (i) in each iteration, it takes a new random sub-sample of the given dataset so that it uses a different training set, and (ii) it uses a random subset of features to determine how to split on each tree node. It gives estimates of what features are best in the classification. This results in a wide diversity that generally results in a better model. Thus, there are three parameters than may be tuned to improve the accuracy of classification:

- 1. the number of randomly selected input variables/features chosen at each tree node,
- 2. the number of trees in the forest,
- 3. tree size, which can be measured by the maximum number of leaf nodes.

Typical Random Forests are not very sensitive to the number of selected input variables, so finetuning is not necessary and overfitting effects due to choice of this parameter is relatively small [12,13]. As stated in [6], the number of trees to use can be chosen as large as desired, without fear of increasing the generalization error. However, user should keep in mind that although increasing the number of trees will decreases variance in predictions, it also (linearly) increases the training time. The original Random Forests models [6] suggest using very large tree. But in recent research [25], a classification example with a forest of large trees shows overfitting. Users can tune either the number of trees or the size of trees to avoid overfitting.

Our dataset is imbalanced since the class of significantly tornadic is much smaller than the other two classes. The imbalanced dataset brings a great challenge to the classification algorithm. The Random Forest model has an effective method, which is weighting the classes, to give balanced results in imbalanced data. This method changes the weight that each class has when calculating the "impurity" score of a chosen split point. Impurity measures how mixed the groups of samples are for a given split in the training dataset and is typically measured with Gini or entropy. The calculation can be biased so that a mixture in favor of the minority class is favored, allowing some false positives for the majority class. Another approach to addressing the problem of class imbalance is to randomly resample the training dataset. The two main approaches to randomly resampling an imbalanced dataset are to delete examples from the majority class, called undersampling, and to duplicate examples from the minority class, called oversampling. We explore both undersampling and oversampling in this project.

In some applications, a variable importance analysis is needed to reduce dimensionality of the problem, which simplifies the classification process, and in some cases improves results. Random Forest uses an unusual but intuitive measure of variable importance [12]. To measure the importance

of a variable k, the out-of-bag predictions of both variable k and a permutation of variable k are obtained. Then, a measure of variable importance for the observation is acquired based on the difference between error rate of those predictions. Averaging over observations from the same class gives class-wise variable importance for the classification algorithm.

### 4.2 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of artificial neural network proposed by Hubel and Wiesel in 1960s inspired by the research of neurons used for local sensitive orientation-selective in cat's visual system. CNN is mainly used for processing data that has a grid pattern, such as images, and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns [33].

CNN is typically comprised of three types of layers (or building blocks): convolutional layers. pooling layers, and fully-connected layers. The first two, convolutional and pooling layers, perform feature extraction, whereas the third, a fully connected layer, maps the extracted features into the final output, such as classification. The convolutional layer plays a key role in a CNN. The neurons in a convolutional layer are locally connected, rather than fully connected, which allow for the extraction of local features from input data. Once a local feature is extracted, the positional relationship between it and other features also will be determined [19]. One layer feeds its output into the next layer. The pooling layer then simply performs downsampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation. For each neuron in every layer we apply an element wise non-linearity called an "activation function". which enables our network to learn more complex functions. Which activation function we use largely depends on the range of our input data, the level of computational performance we desire. and whether we apply the activation in between layers or on the final layer. For the connections between layers we typically use functions such as leaky and non-leaky rectified linear activation function (ReLU) and even hyperbolic tangent [24]. The final layer of the network receives a special activation function such as sigmoid or softmax which produces classification scores for the given input data.

The parameters, such as kernels, are optimized to minimize the difference between outputs of CNN and ground truth labels through training processes. Over a series of epochs of training, the CNN is able to distinguish between dominating and certain low-level features in the inputs and classify the inputs into human proposed categories.

The objective of a neural network is to have a final model that performs well both on the data that we use to train it, such as the training dataset, and the new data on which the model will be used for predictions. The model learns information from given examples and attempts to develop generalized ideas so that it may make accurate future predictions on unseen data. A model that performs extremely well on the training dataset but does not perform well on unseen examples is an overfit model. An overfit model can be easily diagnosed by monitoring the performance of the model during training by evaluating it on both a training dataset and on a validation dataset. There are two typical ways to avoid overfitting in a CNN model. One is to train the network on more examples and the other is to change the complexity. It is more common to focus on methods that constrain the size of the weighs in a network to gain a more stable model that is less sensitive to statistical fluctuations in the input data.

Class imbalance is a common problem in real life applications of deep learning based classifiers, such as our research problem. Methods for addressing class imbalance in CNN can be divided into two main categories [16]. The first category is data level methods that operate on the training data set and change its class distribution [8]. This type includes oversampling and undersampling.

The former one simply introduces more samples in the minority class. A recent research, SMOTE, applied oversampling method by augmenting artificial examples created by interpolating neighboring data points to overcome the overfittiung issue [9]. The later one, as opposed to oversampling, removes examples randomly from majority classes until all classes have the same number of examples. The other category keeps the training dataset unchanged and adjust the training algorithms. We apply the two data level methods in this project.

#### 4.3 Forecast skill metrics

A rigorous assessment of each algorithm is accomplished using a  $3 \times 3$  contingency table (also called a confusion matrix) [14, 15]. Since there are three predictands (i.e., nontornadic, weakly tornadic, and significantly tornadic), a multi-class contingency table is required. Several metrics are presented, including probability of detection (POD; or "hit rate"), the false alarm ratio (FAR), the critical success index (CSI), and the true skill statistic (TSS, also known as the Pierce skill score) [32]. Each of these will help elucidate which algorithms are the most skillful at discriminating between nontornadic and significantly tornadic supercells. There often is a fine balance between POD and FAR. Ideally, one wants the Probability of Detective (POD) to be as high as possible, indicating fewer false predictions. However, it is undesirable to greatly increase the POD at the expense of a low FAR (and vice-versa). Critical Success Index (CSI) and True Skill Statistic (TSS) provide a measurement indicating how well-balanced the forecast model is between overand under-prediction. Higher scores in these areas indicate a more optimal balance. For example, the TSS highlights parameters that maximize POD while minimizing probability of false detection (POFD; or "false alarm rate"). The TSS is defined by

$$TSS = \frac{ad - bc}{(a+c)(b+d)},\tag{4.1}$$

where a is the sum of correct forecasts of a significant tornado report, b is the sum of false alarms, c is the sum of missed significant tornado reports, and d is the sum of correct null forecasts. TSS is also equivalent to the difference between the POD and the POFD [14], meaning that parameters with high TSS have an optimal combination of detecting events without misidentifying nulls.

### 5 Results

In this section, we present the analysis and results of several predictive Random Forest Classification and Convolutional Neural Network models applied to the RUC sounding dataset. Results are organized as follows. In Section 5.1, we describe the preprocessing the data underwent prior to being used in our models. Next, we present the results from two Random Forest models, as well as feature importance analysis, in Section 5.2. Section 5.3 contains the results of a simple and a complex Convolutional Neural Network (CNN) model. Performance for both the Random Forest and CNN models is impacted by the imbalanced nature of the RUC sounding dataset, in which the the majority of storms are nontornadic (54%) and only a small minority (8%) are significantly tornadic. In Section 5.4, we work to address this issue using the techniques of oversampling and undersampling.

The hardware used in all of the enclosed computational studies is part of the UMBC High Performance Computing Facility (hpcf.umbc.edu). The studies use both a CPU node with two 18-core Intel Xeon Gold 6140 Skylake CPUs (2.3 GHz clock speed, 24.75 MB L3 cache, 6 memory channels) and 384 GB memory and a GPU node containing four NVIDIA Tesla V100 GPUs connected by



Figure 5.1: Height readings at each level are variable in the sounding data as depicted in (a). The average height for each sounding level is given by the black dotted line. Final data representation can be visualized as a 3D grid of dimension  $20194 \times 37 \times 6$  and is depicted in (b).

NVLink. The software used is Python 3.7.6 along with the following packages: scikit-learn (v. 0.23.dev0), imbalanced-learn (v. 0.6.2), TensorFlow (v. 2.1.0), and Keras (v. 1.1.0).

### 5.1 Data Preprocessing

In order use the RUC soundings in various machine learning approaches, the data first needed to be pre-processed. Each storm has a vertical profile of pressure, temperature, dewpoint temperature, relative humidity, both components of the horizontal velocity (u-wind and v-wind), and height above ground level at 25 hPa increments up to 100 hPa. Because some severe reports occur at higher elevations, with corresponding lower surface pressure, there are a variable number of vertical levels among the 20194 soundings, illustrated in Figure 5.1 (a). For example, vertical level number 20 has height range 4831 to 8085 m. In order for the machine learning algorithms to learn from common structures in the data, there needs to be a uniform structure to the height grid. Therefore, we compute the average height at each level and performed a linear interpolation of each sounding variable (i.e., pressure, temperature, etc.) to a common height coordinate. This resulted in a 20194 × 37 × 6 array of variables (samples × levels × variable), with height now implicit within each vertical profile, illustrated in Figure 5.1 (b).

#### 5.2 Random Forest

We built and tested a Random Forest model using the machine learning package sklearn<sup>4</sup> in Python. For our base model, hyperparameter settings are as follows: class\_weight = balanced, max\_depth = 200, n\_estimators = 200 (all other parameters use their default settings). We use the 'balanced' setting for class weights because we have imbalanced data: 54% nontornadic, 38% weakly tornadic, and 8% significantly tornadic. The balanced setting weights the classes inversely proportional

<sup>&</sup>lt;sup>4</sup>https://scikit-learn.org/stable/

(a) Confusion Table			
	Actual		
Predicted	[0]	[1]	[2]
[0]	1837	550	62
[1]	307	986	168
[2]	28	32	69

· · ·	•	•	
Class	Total	Predicted	Accuracy
[0]	2172	1837	84.58%
[1]	1568	986	62.88%
[2]	299	69	23.08%

Table 5.1: Confusion table and individual class accuracy for Random Forest base model. Class [0] refers to nontornadic, class [1] refers to weakly tornadic, and class [2] refers to significantly tornadic.

to their frequency of occurrence. Using these settings, and a training-testing split of 80-20, the Random Forest model predictions on the testing data results in an overall accuracy score of 71.6%.

Performance testing using various ranges of hyperparameters indicate that the model is performing optimally given the current settings and the current form of the input data. In testing a range of max\_depth settings from 10 to 400, the minimum accuracy score was 64.79% (for max\_depth = 10), the maximum was 71.6% (for max\_depth = 200), the average accuracy score was 69.78%, and the standard deviation was  $\pm 1.65\%$ . We also experiment with increasing the size of the forest (n\_estimators), varying choices between 100 to 2000. There were no significant gains in accuracy for larger forests, and the best accuracy was observed at around n\_estimators = 200. Similarly, experiments with varying the parameters min\_samples\_leaf and min\_samples\_split showed no significant gains beyond using the default settings.

**Detailed Accuracy Analysis.** We present a more detailed breakdown of accuracy results in Tables 5.1 (a) and (b). Because we have three classes: [0]-nontornadic, [1]-weakly tornadic, and [2]-significantly tornadic, the total accuracy score does not illustrate how well our model performs predicting the various classes of storms. Table 5.1 (a) presents the confusion table and Table 5.1 (b) gives the accuracy scores by class, comparing model predictions with actual events in each class. As evident in both tables, the Random Forest predictor performs poorly on the significantly tornadic class. These tables demonstrate that the Random Forest model is best at predicting nontornadic storms, and worst at predicting significantly tornadic storms.

To put our results in context, we compute and compare forecasting skill scores for the Random Forest (RF) model to current tornado forecasting, using the operational version of the Significant Tornado Parameter [11], and present them in Tables 5.2 (a) and (b). We compute these skill scores in two ways: tornadic vs. nontornadic (classes [1,2] vs. class [0]) and significantly tornadic vs. weakly/nontornadic (class [2] vs. class [0,1]). Examining Table 5.2 (a), we see that the Random Forest model outperforms current forecasting in terms of predicting whether or not a supercell storm will generate some type of tornadic event.

However, we argue that skill scores comparing the classes of significantly tornadic to weakly/nontornadic are of more importance in real forecasting, since significant tornadoes account for the majority of damage and human causalities. Misdiagnosing a significant tornado as a nontornadic event is substantially more catastrophic to the public. In Table 5.2 (b), we see that the Random Forest model skill scores are less desirable when viewed in comparison to current forecasting methods. The main exception is that RF has a lower FAR. This is expected, given that Tables 5.1 (a) and (b) indicate that our model is biased towards nontornadic events.

**Feature Importance Analysis.** The ability to easily and efficiently perform feature importance analysis was our primary motivation for using the Random Forest model on this data set. Feature

	Scoi	re	<b>RF</b> Prediction		Forecasting	
	POI	D	0.67		0.56	
	FAF	2	0.21		0.37	
	CSI		0.57		0.42	
	TSS	5	0.52		0.23	
(b	$\overline{\mathbf{o}}$ ) Sig	nif	icantly vs. We	ak	ly/Nontornad	ic
S	core	R	F Prediction		$\mathbf{STP}$	
Ρ	OD		0.24		0.68	
F.	AR		0.53		0.71	
$\mathbf{C}$	SI		0.19		0.25	
$\mathbf{T}$	SS		0.22		0.45	

(a) Tornadic vs. Nontornadic

Table 5.2: Skill scores for Random Forest model compared to current forecasting ability. We consider two cases: tornadic vs. nontornadic (classes [1,2] vs. [0]) and Significantly tornadic vs. Nontornadic (class [2] vs. class [0]). **POD** stands for probability of detection, **FAR** for false alarm ratio, **CSI** for critical success index, and **TSS** for true skill statistic.

importance analysis can lead to insights regarding our data, can lead to model improvements, and will play a role in the construction of our Convolutional Neural Network (CNN) model that we will discuss further in Section 5.3. There are multiple ways to perform feature importance analysis. For this research, feature importance for a given decision tree is defined as the decrease in node impurity weighted by the probability of reaching that node. Once that tree-specific feature importance is calculated for each tree in our forest, we average the results to compute a feature importance score. This is done automatically using the sklearn function feature importances... The cumulative sum of all feature importance scores is one. Higher feature scores indicate higher importance.

Recall that our data has 222 features, however, these features are split equally across six variables, temperature, dewpoint, humidity, u-wind, v-wind, and pressure, corresponding to thirtyseven standardized heights ranging from 10 m to 16.3 km. Since each feature is linked to a specific height, we plotted the feature importance scores of each variable across the range of heights in Figure 5.2, along with the cumulative feature importance scores for each variable, in which we add up each variable's feature importance scores for every height. These figures indicate the v-wind variable scored significantly higher than other variables, specifically at heights below 4 km. We also note that the first few humidity readings were of more importance than subsequent humidity readings. The pressure variable score consistently lower than other variables across all height levels.

These feature importance results are consistent with known key features of tornado formation. Low-altitude relative humidity is a decent predictor of downdraft coldness. Lower relative humidity allows for more evaporation and typically colder downdrafts. This colder, denser air at the surface resists being converged and stretched into a tornado. Along these lines, low-altitude vertical wind shear is well correlated with the strength of the convergence and stretching by the supercell on developing vortices. The combination of high low-altitude relative humidity and vertical wind shear make tornadogenesis more likely. On tornado outbreak days, the lower troposphere can be so humid that cloud bases are just a few hundred meters above the ground, and the wind shear can be so extreme that winds can vary by 20 m/s within the lowest 1 km [21].

**Revised Random Forest Model and Feature Importance Analysis.** Given the results of our feature analysis, we built a revised Random Forest model without the pressure variable. We



Figure 5.2: Feature importance scores and variable contributions from Random Forest model.

(a) Confusion Table			
	Actual		
Predicted	[0]	[1]	[2]
[0]	1803	570	59
[1]	324	958	171
[2]	38	44	72

(b	) Accuracy	By	Class
----	------------	----	-------

Class	Total	Predicted	Accuracy
[0]	2165	1803	83.28%
[1]	1572	958	60.94%
[2]	302	72	23.84%

Table 5.3: Confusion table and individual class accuracy for Random Forest model without pressure variable. Class [0] refers to nontornadic, class [1] refers to weakly tornadic, and class [2] refers to significantly tornadic.

drop pressure due to its overall lower importance score. Figure 5.3 gives the updated Feature importance scores and cumulative contributions for each variable, illustrating the same trends discussed above for the original model.

The resulting accuracy score of this exercise is 70.14%, approximately 1.46% decrease in accuracy from our base model. Upon examining the confusion table and class accuracy scores in Tables 5.3 (a) and (b), we see that the Random Forest model built by dropping the pressure variable performs comparably to our original model.



Figure 5.3: Feature importance scores and variable contributions from Random Forest model without pressure variable.

(a) Simple CNN Confusion Table			
	Actual		
Predicted	[0]	[1]	[2]
[0]	1873	953	211
[1]	323	557	94
[2]	11	11	6

(b) Accuracy By Class

( )	v	v	
Class	Total	Predicted	Accuracy
[0]	2207	1873	84.87%
[1]	1521	557	36.62%
[2]	311	6	1.91%

Table 5.4: Confusion table and individual class accuracy for the Convolutional Neural Network simple model. Class [0] refers to nontornadic, class [1] refers to weakly tornadic, and class [2] refers to significantly tornadic.

## 5.3 Convolutional Neural Network

We built and tested a one-dimensional Convolutional Neural Network model using the machine learning package Keras (https://keras.io) in Python. For our base model, the network was created using two 1D convolution networks, both with 64 filters. This is followed by a 1D MaxPooling layer and then two dense layers, with ReLU and softmax activation functions, respectively. The base model is compiled with Analysis Data model (ADaM) optimization for stochastic gradient descent and the categorical crossentropy loss function. Based on the feature importance results from the Random Forest model in Section 5.2, pressure was removed as a training variable. After training for 100 epochs and using a training-testing split of 80-20, the Convolutional Neural Network model predictions on the testing data result in an overall accuracy score of 60.83%. As discussed in Section 5.2, the total accuracy does not illustrate how well our model performs predicting the various classes of storms. Table 5.4 (a) presents the confusion table and Table 5.4 (b) gives the accuracy scores by class, comparing model predictions with actual events in each class. Evident in both tables, the simple CNN model performs especially poorly on the significantly tornadic class and most accurately at predicting nontornadic storms.

A more complex CNN model is also constructed. This model consists of three 1D convolution networks, with increasing number of filters (32, 64, and 128). The activation function for each of the layers is Leaky ReLU ( $\alpha = 0.1$ ). Each layer is followed by a batch normalization layer and a 1D MaxPooling layer. Finally, there are two dense layers, with Leaky ReLU and softmax activation functions, respectively. This more complex model is compiled with stochastic gradient descent optimization and the mean squared error loss function. After training for 100 epochs and using a training-testing split of 80-20, the Convolutional Neural Network model predictions on the testing data results in an overall accuracy score of 67.84%. Table 5.5 (a) presents the confusion table and Table 5.5 (b) gives the accuracy scores by class, comparing model predictions with actual events in each class. The complex model considerably improves the accuracy for tornadic and significantly tornadic supercells, at the expense of a slight decrease in accuracy for nontornadic supercells.

Similar to Section 5.2, we put our results in context by computing and comparing forecasting skill scores for the CNN to current tornado forecasting, using the operational version of the Significant Tornado Parameter [11], and present them in Tables 5.6 (a) and (b). We compute these skill scores in two ways: tornadic vs. nontornadic (classes [1,2] vs. class [0]) and significantly tornadic vs. weakly/nontornadic (class [2] vs. class [0,1]). The CNN outperforms current forecasting parameter in terms of predicting whether or not a supercell storm will generate some type of tornadic event. However, the CNN performs considerably poorer when comparing significantly tornadic supercells in comparison to current forecasting methods, which is not too surprising since the Significant Tornado Parameter was specifically designed and tuned to differentiate significant tornadic supercells from nontornadic supercells.

(a) Complex CNN Confusion Table			
	Actual		
Predicted	[0]	[1]	[2]
[0]	1774	567	75
[1]	390	885	153
[2]	43	69	83

(b) Accuracy By Class

( )	v	v	
Class	Total	Predicted	Accuracy
[0]	2207	1774	80.38%
[1]	1521	885	58.19%
[2]	311	83	26.69%

Table 5.5: Confusion table and individual class accuracy for the Convolutional Neural Network complex model. Class [0] refers to nontornadic, class [1] refers to weakly tornadic, and class [2] refers to significantly tornadic.

$(a) \mathbf{I}$	(a) Iornaule vs. romornaule				
Score	e CNN Prediction	<b>Forecasting</b>			
POD	0.65	0.56			
FAR	0.27	0.37			
CSI	0.53	0.42			
TSS	0.45	0.23			
(b) Sig	nificantly vs. Weak	ly/Nontornadic			
Score	<b>CNN</b> Prediction	STP			
POD	0.27	0.68			
FAR	0.57	0.71			
CSI	0.19	0.25			
maa	0.92	0.45			

#### (a) Tornadic vs. Nontornadic

Table 5.6: Skill scores for the Convolutional Neural Network complex model compared to current forecasting ability. We consider two cases: tornadic vs. nontornadic (classes [1,2] vs. [0]) and Significantly tornadic vs. Nontornadic (class [2] vs. class [0]). Currently forecasting skill scores were calculated from [32].

### 5.4 Randomly Undersampling/Oversampling the Classes

The higher accuracy in the Random Forest and CNN models for the majority class of nontornadic supercells suggests that the model is much better trained for this storm type, likely because it is training on much more data. As an initial attempt to combat this issue, during the training phase, either the majority class was undersampled or the minority class was oversampled using the RandomUnderSampler/RandomOverSampler function in the machine learning toolbox Imbalanced Learn<sup>5</sup> in Python.

The Random Forest model using a training-testing split of 80-20 results an overall accuracy score of 60.48% for the random undersampling (RUS) and an accuracy score of 68.43% for the random oversampling (ROS) on the testing data. Table 5.7 presents the confusion tables and the accuracy scores by class, comparing model predictions with actual events in each class for both the random undersampling (RUS) and random oversampling (ROS). The random undersampling (RUS) of the majority class greatly improves the accuracy for significantly tornadic supercells, resulting in an increase from 23.84% to 65.85%. However, this comes at a steep expense as it decreases accuracy for nontornadic supercells from 83.28% to 69.78% and decreases the accuracy for weakly tornadic supercells from 60.94% to 47.02%. Fewer training examples of these two majority classes is the main reason for this decrease in accuracy. Comparatively, the random oversampling (ROS) also increases

<sup>&</sup>lt;sup>5</sup>https://pypi.org/project/imbalanced-learn/

	Actual					
Predicted	[0]	[1]	[2]			
[0]	1478	380	31			
[1]	403	749	81			
[2]	237	237 464				
(c) <b>RF w</b> /	ROS (	Confusion Table				
		Actual				
Predicted	[0]	[1]	[2]			
[0]	1782	596	78			
[1]	303	867	150			
[2]	47	101	115			

(a) **RF w/ RUS Confusion Table** 

(b) Accuracy By Class

· · /	•	•	
Class	Total	Predicted	Accuracy
[0]	2118	1478	69.78%
[1]	1593	749	47.02%
[2]	328	216	65.85%

(	(d)	Accuracy	By	Class

Class	Total	Predicted	Accuracy
[0]	2132	1782	83.58%
[1]	1564	867	55.43%
[2]	343	115	33.53%

Table 5.7: Confusion table and individual class accuracy for the Random Forest model with random undersampling of the majority class (i.e., Class [0]). Class [0] refers to nontornadic, class [1] refers to weakly tornadic, and class [2] refers to significantly tornadic.

the accuracy for significantly tornadic supercells from 23.84% to 33.53%, while maintaining about the same accuracy for the nontornadic class and leads to a small decrease in accuracy for the weakly tornadic class by about 5%. From these results, we conclude that undersampling may be more suitable for imbalanced datasets when the prediction accuracy of the minority class is most important.

Using the complex CNN, training for 100 epochs with a training-testing split of 80-20, the Convolutional Neural Network model predictions on the testing data resulted in an overall accuracy score of 55.73%. Table 5.8 (a) presents the confusion table and Table 5.8 (b) gives the accuracy scores by class, comparing model predictions with actual events in each class for both the random undersampling (RUS) and random oversampling (ROS). The random undersampling of the majority class further improves the accuracy for significantly tornadic supercells, however this comes at a steep expense in the accuracy for nontornadic supercells. The lower accuracy for nontornadic supercells is almost certainly due to fewer training samples of this class. Comparatively, the random oversampling also increases the accuracy for significantly tornadic supercells, but without severely decreasing the accuracy for nontornadic supercells. This is consistent with previous literature that oversampling is the a more robust technique for dealing with imbalanced datasets than undersampling [17].

In summary, these results suggest that the imbalanced dataset is a meaningful contributor to the lower accuracy for tornadic storms, however the simple method of randomly undersampling (oversampling) the majority (minority) class can yield undesirable effects as well. Future work should investigate alternative methods of dealing with imbalanced datasets in a CNN, including more sophisticated undersampling/oversampling techniques [18] or using a Generative Adversarial Network (GAN) to create synthetic sounding data for tornadic supercells.

### 5.5 Comparison between Random Forest and CNN

To compare the two machine learning techniques, Random Forest and CNN, we summarize the accuracy and skill score results for all models constructed herein in Tables 5.9 (a) and (b).

Random undersampling (RUS) applied to Random Forest produces the highest class accuracy for the significant tornado class when compared to all other models. At the same time, RUS results

	Actual					
Predicted	[0]	[1]	[2]			
[0]	1249	431	64			
[1]	818	885	129			
[2]	140	205	118			
(c) CNN w/ ROS Confusion Table						
(c) OI II W	/ 1000		Tubion Tubic			
		Ac	ctual			
Predicted	[0]	Ac	ctual [2]			
Predicted [0]	[0] 1700	Ac [1] 511	[2]			
[0] [1]	[ <b>0</b> ] 1700 431	Addition           [1]           511           874	[2] 75 124			

(a) CNN w/ RUS Confusion Table

(b)	Accuracy	By	Class
-----	----------	----	-------

· · /			
Class	Total	Predicted	Accuracy
[0]	2207	1249	56.59%
[1]	1521	885	58.19%
[2]	311	118	37.94%

(	Έ)	Accuracy	$\mathbf{Bv}$	Class
	· •	/ moouracy	<b>_</b> ,	Clubb

Class	Total	Predicted	Accuracy
[0]	2207	1700	77.03%
[1]	1521	874	57.46%
[2]	311	112	36.01%

Table 5.8: Confusion table and individual class accuracy for the Convolutional Neural Network complex model with random undersampling of the majority class (i.e., Class [0]). Class [0] refers to nontornadic, class [1] refers to weakly tornadic, and class [2] refers to significantly tornadic.

in an approximate 14% decrease in class accuracy in both the nontornadic and weakly tornadic categories in comparison to the base Random Forest model. Upon examining the forecasting skill scores for significantly tornadic vs. weakly/nontornadic (class [2] vs. class [0,1]), given in Table 5.9 (b), the Random Forest RUS model has essentially the same skill in POD, FAR, CSI, and TSS as the current STP forecasting scores, without any preconceived notions of what the data looks like.

Random oversampling (ROS) applied to CNN exhibits the best balance of increased accuracy in the significantly tornadic category with less adverse impact other accuracy measures. With 36.01% class accuracy in the significant tornado category, this exceeds the accuracy of the base Random Forest, ROS Random Forest, and Base CNN model performance for this class. While there is a decrease in nontornadic and weakly tornadic class accuracy when compared to the base CNN model, this decrease is significantly less severe than what occurs with random undersampling (RUS) of CNN. CNN with ROS has a higher Probability of Detection (POD) score than the base model of CNN, with a smaller increase in false alarm ratio (FAR) than CNN with RUS. Both the Critical Success Index (CSI) and True-Skill Statistic (TSS) scores improve when ROS is applied to the base CNN model. Based on these skill score measures, while less accurate than current STP forecasting, this model has similar performance to the base CNN model, but with improved accuracy in the significantly tornadic class.

# 6 Conclusions and Future Work

Accurate prediction of significant tornadoes using machine learning algorithms is a relatively new and challenging data science problem. Solutions to this problem could, in time, provide a useful tool in severe weather forecasting and may provide additional insight into conditions surrounding tornado formation. The work conducted herein represents an initial exploration into using environmental sounding data that describes the pre-storm atmospheric conditions of a supercell to create and train Random Forest classification and Convolutional Neural Network models for tornado prediction. We present several Random Forest (RF) and Convolutional Neural Network (CNN) models for tornado prediction, each model created and trained using Rapid Update Cycle (RUC) sounding data for supercell storms. Feature analysis of the Random Forest model indicates that the pressure variable has little impact on the classification process, which is consistent with known key physical

	Randor	n Forest		CNN		
Accuracy	Base	RUS	ROS	Base	RUS	ROS
Nontornadic	83.28%	69.78%	83.58%	80.38%	56.59%	77.03%
Weakly-Tornadic	60.94%	47.02%	55.43%	58.19%	58.18%	57.45%
Significantly-Tornadic	23.84%	65.85%	33.53%	26.69%	37.94%	36.01%
Total Accuracy	70.14%	60.48%	68.43%	67.84%	58.26%	65.12%

### (a) Comparison of Accuracy Scores

### (b) Comparison of Forecast Skill Metrics

Random Forest				$\mathbf{CNN}$			Forecasting
Score	Base	RUS	ROS	Base	RUS	ROS	STP
POD	0.24	0.65	0.34	0.27	0.38	0.36	0.68
FAR	0.53	0.76	0.56	0.57	0.75	0.65	0.71
CSI	0.19	0.21	0.23	0.19	0.18	0.21	0.25
TSS	0.22	0.47	0.29	0.23	0.29	0.30	0.45

Table 5.9: Comparison metrics for Random Forest and CNN models (both without the pressure variable). The Base Random Forest model refers to the revised model presented in Section 5.2. The Base CNN model refers to the complex CNN model presented in Section 5.3. The Forecast Skill metrics represent the Significantly Tornadic vs. Weakly/Nontornadic Skill Scores compared with current forecasting ability. STP refers to the Significant Tornado Parameter from Equation (2.1).

attributes of tornado formation. Initial results point to the imbalanced dataset as being a barrier for obtaining high accuracy in predicting the most severe types tornadic events. While random oversampling and undersampling offer some improvement to these results, this comes at a cost in other accuracy measures.

Future work should consider other techniques for handling imbalanced data, including: oversampling techniques involving the generation of new samples (e.g., SMOTE, ADASYN), undersampling using different selection rules (e.g. Near Miss, Nearest Neighbors), undersampling by generation of new samples (K-means clustering), and combinations of over- and under-sampling. Another technique that may also improve performance is to use a Generative Adversarial Network (GAN) to synthetically create more data for the significantly tornadic class, creating a more balanced training dataset. In addition to addressing the imbalanced data, expanding the dataset from one-dimensional sounding profiles into three-dimensional profiles is another possible avenue for improvement in tornado prediction using machine learning.

# Acknowledgments

This work is supported by the grant "CyberTraining: DSE: Cross-Training of Researchers in Computing, Applied Mathematics and Atmospheric Sciences using Advanced Cyberinfrastructure Resources" from the National Science Foundation (grant no. OAC-1730250). The hardware used in the computational studies is part of the UMBC High Performance Computing Facility (HPCF). The facility is supported by the U.S. National Science Foundation through the MRI program (grant nos. CNS-0821258, CNS-1228778, and OAC-1726023) and the SCREMS program (grant no. DMS-0821311), with additional substantial support from the University of Maryland, Baltimore County (UMBC). See hpcf.umbc.edu for more information on HPCF and the projects using its resources. Co-author Carlos Barajas additionally acknowledges support as HPCF and CyberTraining RA.

# References

- Carlos A. Barajas. An Approach to Tuning Hyperparameters in Parallel: A Performance Study Using Climate Data. M.S. Thesis, Department of Mathematics and Statistics, University of Maryland, Baltimore County, 2019.
- [2] Carlos A. Barajas, Matthias K. Gobbert, and Jianwu Wang. Performance benchmarking of data augmentation and deep learning for tornado prediction. In 2019 IEEE International Conference on Big Data (Big Data), pages 3607–3615. IEEE, 2019.
- [3] Charlie Becker, Will D. Mayfield, Sarah Y. Murphy, Bin Wang, Carlos Barajas, and Matthias K. Gobbert. An approach to tuning hyperparameters in parallel: A performance study using climate data. Technical Report HPCF-2019-13, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2019.
- [4] Stanley G. Benjamin, Dezsö Dévényi, Stephen S. Weygandt, Kevin J. Brundage, John M. Brown, Georg A. Grell, Dongsoo Kim, Barry E. Schwartz, Tatiana G. Smirnova, Tracy Lorraine Smith, et al. An hourly assimilation-forecast cycle: The RUC. Mon. Wea. Rev., 132(2):495–518, 2004.
- [5] P. D. Bothwell, J. A. Hart, and R. L. Thompson. An integrated three-dimensional objective analysis scheme in use at the Storm Prediction Center. In 21st Conf. on Severe Local Storms, San Antonio, TX, 2002. Amer. Meteor. Soc.
- [6] L. Breiman. Random forests. Machine Learning, 45:5–32, 2001.
- [7] J. Brotzge, S. Erickson, and H. Brooks. A 5-yr climatology of tornado false alarms. Wea. Forecasting, 26(4):534-544, 2011.
- [8] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249—-259, Oct 2018.
- [9] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. J. Artif. Intell. Res., 16:321–357, 01 2002.
- [10] François Chollet. Deep Learning with Python. Manning, 2018.
- [11] Brice E. Coffer, Matthew D. Parker, Richard L. Thompson, Bryan T. Smith, and Ryan E. Jewell. Using near-ground storm relative helicity in supercell tornado forecasting. *Wea. Fore-casting*, 34(5):1417–1435, 2019.
- [12] Adele Cutler, David Cutler, and John Stevens. Random forests. Machine Learning, 45:157–176, 2011.
- [13] Ramon Diaz-Uriarte and Sara Alvarez. Gene selection and classification of microarray data using random forest. BMC Bioinformatics, 7:3, 02 2006.
- [14] Charles A. Doswell, III, Robert Davies-Jones, and David L. Keller. On summary measures of skill in rare event forecasting based on contingency tables. *Wea. Forecasting*, 5(4):576–585, 1990.

- [15] Charles A. Doswell, III and David M. Schultz. On the use of indices and parameters in forecasting severe storms. *Electon. J. Severe Storms Meteor.*, 1(3), 2006.
- [16] H. He and E. A. Garcia. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9):1263–1284, 2009.
- [17] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. Intell. Data Anal., 6:429–449, 11 2002.
- [18] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.
- [19] Tianyi Liu, Shuangsang Fang, Yuehui Zhao, Peng Wang, and Jun Zhang. Implementation of training convolutional neural networks. arXiv preprint arXiv:1506.01195, 2015.
- [20] Paul Markowski, Yvette Richardson, James Marquis, Joshua Wurman, Karen Kosiba, Paul Robinson, David Dowell, Erik Rasmussen, and Robert Davies-Jones. The pretornadic phase of the Goshen County, Wyoming, supercell of 5 june 2009 intercepted by VORTEX2. Part I: Evolution of kinematic and surface thermodynamic fields. *Mon. Wea. Rev.*, 140(9):2887–2915, 2012.
- [21] Paul M. Markowski and Yvette P. Richardson. Tornadogenesis: Our current understanding, forecasting considerations, and questions to guide future research. Atmospheric Research, 93(1):3–10, 2009. 4th European Conference on Severe Storms.
- [22] Amy McGovern, Kimberly L. Elmore, David John Gagne, Sue Ellen Haupt, Christopher D. Karstens, Ryan Lagerquist, Travis Smith, and John K. Williams. Using artificial intelligence to improve real-time decision-making for high-impact weather. Bull. Amer. Meteor. Soc., 98(10):2073–2090, 2017.
- [23] Amy McGovern, Ryan Lagerquist, David John Gagne, G. Eli Jergensen, Kimberly L. Elmore, Cameron R. Homeyer, and Travis Smith. Making the black box more transparent: Understanding the physical implications of machine learning. *Bull. Amer. Meteor. Soc.*, 100(11):2175– 2199, 2019.
- [24] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. ArXiv e-prints, 11 2015.
- [25] Mark Segal and Yuanyuan Xiao. Multivariate random forests. WIREs Data Mining and Knowledge Discovery, 1(1):80–87, 2011.
- [26] Bryan T. Smith, Richard L. Thompson, Jeremy S. Grams, Chris Broyles, and Harold E. Brooks. Convective modes for significant severe thunderstorms in the contiguous United States. Part I: Storm classification and climatology. Wea. Forecasting, 27(5):1114–1135, 2012.
- [27] Richard L. Thompson, Roger Edwards, John A. Hart, Kimberly L. Elmore, and Paul Markowski. Close proximity soundings within supercell environments obtained from the Rapid Update Cycle. *Wea. Forecasting*, 18(6):1243–1261, 2003.
- [28] Richard L. Thompson, Corey M. Mead, and Roger Edwards. Effective storm-relative helicity and bulk shear in supercell thunderstorm environments. Wea. Forecasting, 22(1):102–115, 2007.

- [29] Richard L. Thompson, Bryan T. Smith, Jeremy S. Grams, Andrew R. Dean, and Chris Broyles. Convective modes for significant severe thunderstorms in the contiguous United States. Part II: Supercell and QLCS tornado environments. Wea. Forecasting, 27(5):1136–1154, 2012.
- [30] Richard L. Thompson, Bryan T. Smith, Jeremy S. Grams, Andrew R. Dean, Joseph C. Picca, Ariel E. Cohen, Elizabeth M. Leitman, Aaron M. Gleason, and Patrick T. Marsh. Tornado damage rating probabilities derived from WSR-88D data. *Wea. Forecasting*, 32(4):1509–1528, 2017.
- [31] Robert J. Trapp, Gregory J. Stumpf, and Kevin L. Manross. A reassessment of the percentage of tornadic mesocyclones. Wea. Forecasting, 20(4):680–687, 2005.
- [32] Daniel S. Wilks. *Statistical Methods in the Atmospheric Sciences*, volume 100. Academic Press, 2011.
- [33] Rikiya Yamashita, Mizuho Nishio, Richard Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9, 06 2018.