

Real-time enhanced tornado warning product guidance based on tornado debris signature height
and population density

By

Alexander Cooke

Approved by:

Michael E. Brown (Major Professor)

Michelle E. Saunders

Andrew Mercer (Committee Member/Graduate Coordinator)

Rick Travis (The Dean, College of Arts & Sciences)

A Thesis

Submitted to the Faculty of

Mississippi State University

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in Geosciences with a concentration in Applied Meteorology

in the Department of Geosciences

Mississippi State, Mississippi

May 2025

Copyright by

Alexander Cooke

2025

Name: Alexander Cooke

Date of Degree: May 16, 2025

Institution: Mississippi State University

Major Field: Geosciences with a concentration in Applied Meteorology

Major Professor: Michael E. Brown

Title of Study: Real-time enhanced tornado warning product guidance based on tornado debris
signature height and population density

Pages of Study: 83

Candidate for Degree of Master of Science

This research aims to develop an operational program for enhancing real-time tornado warning capabilities by integrating tornado debris signature (TDS) analysis with population impact assessment. The proposed Python-based tool will ingest Level II radar data to identify and analyze TDS, estimating tornado intensity based on maximum TDS height. For high-intensity tornadoes, the program will project the potential impact area using storm motion vectors and integrate this with population density data. The system will assess whether the situation meets National Weather Service criteria for enhanced warning products, providing forecasters with rapid, objective guidance for critical warning decisions. The program's performance will be validated using historical tornado cases. Expected outcomes include reduced forecaster cognitive load during severe weather events and consistent recommendations for enhanced tornado warnings. Limitations such as radar distance constraints and time lags in TDS formation are acknowledged.

Keywords: tornadoes, debris, radar

ACKNOWLEDGEMENTS

The author wishes to acknowledge and thank Professor Michael Brown, Professor Andrew Mercer, and Professor Michelle Saunders for their guidance, patience, and generosity of time and knowledge.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
LIST OF TABLES	vi
LIST OF FIGURES	viii
LIST OF SYMBOLS, ABBREVIATIONS, AND NOMENCLATURE	x
CHAPTER	
I. INTRODUCTION	1
II. LITERATURE REVIEW	3
III. RESEARCH OBJECTIVES AND HYPOTHESIS	11
IV. METHODOLOGY AND ANTICIPATED LIMITATIONS	13
4.1 Methodology	13
4.2 Population Analysis	20
4.3 Anticipated Limitations	22
4.3.1 Radar Distance	22
4.3.2 Time for TDS to Appear	22
4.3.3 QLCS Tornadoes	22
4.3.4 Available Debris	23
4.4 Data Management	23
V. PROGRAM STRUCTURE	24
5.1 Introduction	24
5.2 Global Imports and Configuration	24
5.3 Functions and Utilities	25
5.3.1 process_chunk	25

5.3.2	<code>rename_duplicates</code>	26
5.3.3	<code>format_size_mb</code>	26
5.3.4	<code>create_velocity_colormap</code>	26
5.3.5	<code>add_motion_polygon</code>	27
5.3.6	<code>calculate_polygon_vertices</code>	27
5.3.7	<code>is_point_in_polygon</code>	28
5.3.8	<code>calculate_parallel_distance</code>	28
5.4	Custom Dialogs and Windows	29
5.4.1	<code>CoupletCategoryDialog</code>	29
5.4.2	<code>RadarDisplayWindow</code>	30
5.4.3	<code>CityImpactsWindow</code>	31
5.4.4	<code>Debris3DWindow</code>	32
5.5	Main Application: <code>RadarViewer</code>	33
5.5.1	Structure and Purpose	33
5.5.2	Key Data Attributes	33
5.5.3	Case Management	34
5.5.4	Loading Radar Data from AWS	34
5.5.5	Displaying a Radar Frame	35
5.5.6	Plotting Methods	36
5.5.7	Couplet Detection Workflow	36
5.5.8	Debris Height Workflow	36
5.6	Algorithm Options	37
5.7	Additional Functions	38
5.7.1	Manual Addition of Couplets	38
5.7.2	Population Analysis	38
5.7.3	Visual Filters	40
5.7.4	3D Viewer	41
5.8	Full Interface	42
VI.	OBSERVED LIMITATIONS	43
6.1	Range Folding	43
6.2	Debris Contamination	43
6.3	Proximity to Radar	44
6.4	Exceedingly Wide Signals	44
6.5	CC Melting Layer	44
6.6	Low-Significant Far Tornadoes	45
6.7	Overfiltering	45
6.8	Low CC Inflow Area	46
6.9	Delay Between Damage and Signature Appearance	46
6.10	Discussion of Limitations	47
VII.	RESULTS	49

7.1	Overall Results	49
7.2	Category-Specific Results	51
7.3	NWS Verification Metrics	59
7.4	Correlation Plots	61
7.5	Bootstrap Analysis	66
VIII.	DISCUSSION AND FUTURE DIRECTIONS	69
8.1	Summary and Interpretation of Key Findings	69
8.1.1	Intensity Classification Performance	69
8.1.2	Lead Time Advantages	70
8.1.3	Warning Classification Performance	70
8.2	Operational Implications and Applications	71
8.2.1	Integration with Current Warning Systems	71
8.2.2	Forecaster Decision Support	72
8.2.3	Standardization and Consistency Benefits	72
8.3	Current Limitations and Challenges	72
8.3.1	Technological and Meteorological Constraints	72
8.3.2	Classification and Detection Challenges	73
8.3.3	Verification and Validation Limitations	74
8.3.4	Algorithm Enhancements	75
8.3.5	Additional Data Integration	76
8.3.6	Population Analysis Improvements	76
8.3.7	Operational Implementation Improvements	77
8.3.8	Testing and Validation	78
8.4	Broader Implications and Future Vision	78
8.4.1	Impact on Warning Philosophy	78
8.4.2	Societal Impacts and Benefits	78
8.4.3	Long-term Research Directions	79
8.5	Conclusion	79
IX.	REFERENCES	81

LIST OF TABLES

4.1	Geographic distribution of selected cases	16
7.1	Overall Verification Metrics (Warning Times)	49
7.2	Detection Level Proportions	50
7.3	Overall Lead Time Statistics (Mean and Standard Deviation)	50
7.4	Overall Lead Time Statistics (Quantiles)	50
7.5	Category-Specific Verification Metrics (Warning Times)	51
7.6	TDS Height Detection Level Analysis by Category	52
7.7	Emergency Stats for V_{rot} (Mean and Standard Deviation)	53
7.8	Emergency Stats for V_{rot} (Quantiles)	53
7.9	Height Statistics (Mean and Standard Deviation) for Tornado Emergencies	53
7.10	Height Statistics (Quantiles) for Tornado Emergencies	53
7.11	Lead Time Statistics (Mean and Standard Deviation) for Tornado Emergencies	54
7.12	Lead Time Statistics (Quantiles) for Tornado Emergencies	54
7.13	PDS Stats for V_{rot} (Mean and Standard Deviation)	55
7.14	PDS Stats for V_{rot} (Quantiles)	55
7.15	PDS Stats for Height (Mean and Standard Deviation)	55
7.16	PDS Stats for Height (Quantiles)	55
7.17	PDS Lead Time Statistics (Mean and Standard Deviation)	56

7.18	PDS Lead Time Statistics (Quantiles)	56
7.19	Normal Stats for V_{rot} (Mean and Standard Deviation)	56
7.20	Normal Stats for V_{rot} (Quantiles)	56
7.21	Normal Stats for Height (Mean and Standard Deviation)	57
7.22	Normal Stats for Height (Quantiles)	57
7.23	Lead Time Statistics (Mean and Standard Deviation)	57
7.24	Lead Time Statistics (Quantiles)	57
7.25	Overall Verification Metrics (NWS Scores - Issue Times)	59
7.26	Category-Specific NWS Scores (Issue Times)	60

LIST OF FIGURES

4.1	Algorithm flowchart	18
4.2	Population analysis flowchart	19
5.1	Color Maps for the Radar Plots	26
5.2	Warning polygons generated by the function seen in the reflectivity plot (distances from radar site are shown along axes in kilometers)	27
5.3	This code checks for loaded cities inside a polygon.	28
5.4	The manual couplet dialog	29
5.5	City Impact Window	31
5.6	Three-dimensional debris tracker	32
5.7	Radar display controls	35
5.8	Algorithm options section	37
5.9	Visual filter options	40
5.10	Figure 5.9 zoomed on area of interest	40
5.11	3D debris viewer window	41
5.12	Full program interface	42
5.13	Full program interface with population window	42
7.1	Histogram of EF Rating vs Debris Height	61
7.2	Correlation between EF rating and debris height: (a) Full dataset, (b) EF 2-5 subset	62
7.3	Correlation between EF rating and debris height for only significant tornadoes . . .	64

7.4	Bootstrap distributions for 10,000 samples	66
7.5	Associated bootstrap confidence intervals	66

LIST OF SYMBOLS, ABBREVIATIONS, AND NOMENCLATURE

This section provides definitions for the technical terms and symbols used throughout this thesis.

AWS Amazon Web Services

AWIPS Advanced Weather Interactive Processing System

Boto3 AWS SDK for Python

BR Base Reflectivity

BV Base Velocity

CC Correlation Coefficient

CSI Critical Success Index

dBZ Decibels of Z (reflectivity)

EBWD Effective Bulk Wind Difference

EDEX Environmental Data Exchange

EF Enhanced Fujita (scale)

EF0-1 Enhanced Fujita scale ratings 0-1 (weak tornadoes)

EF2-3 Enhanced Fujita scale ratings 2-3 (significant tornadoes)

EF4-5 Enhanced Fujita scale ratings 4-5 (violent tornadoes)

Emergency A warning corresponding to a violent tornado threatening a populated area

ESRH Effective Storm Relative Helicity

ETA Estimated Time of Arrival

FAR False Alarm Ratio

GIS Geographic Information System

GLM Geostationary Lightning Mapper

GOES Geostationary Operational Environmental Satellite

GUI Graphical User Interface

IBW Impact-Based Warning

JSON JavaScript Object Notation

MLCAPE Mixed Layer Convective Available Potential Energy

MLCIN Mixed Layer Convective Inhibition

MLLCL Mixed Layer Lifting Condensation Level

Normal Warning A tornado warning corresponding to a weak tornado

NWS National Weather Service

PDS Particularly Dangerous Situation

POD Probability of Detection

Python-AWIPS Python implementation for interfacing with AWIPS

QLCS Quasi-Linear Convective System

qIDSS Quantitative Impact-Based Decision Support Services

REF Reflectivity

RHO Correlation Coefficient (alternate notation)

SciPy Scientific Python (computational library)

SRV Storm-Relative Velocity

STP Significant Tornado Parameter

TDS Tornado Debris Signature

UI User Interface

USGS United States Geological Survey

VEL Velocity

WarnGen Warning Generator (in AWIPS)

WSR-88D Weather Surveillance Radar - 1988 Doppler

ZDR Differential Reflectivity

r^2 Coefficient

V_{rot} Rotational Velocity, defined as $\frac{|v_{max}| - |v_{min}|}{2}$

v_{max} Maximum velocity

v_{min} Minimum velocity

CHAPTER I

INTRODUCTION

Tornadoes pose a significant threat to life and property across much of the United States. The ability to accurately assess tornado intensity in real-time is crucial for effective warnings and public safety. However, this task remains one of the most challenging aspects of operational meteorology. Recent advancements in radar technology, particularly the ability to detect and analyze tornado debris signatures (TDS), offer new opportunities for improving tornado intensity estimation, impact prediction, and ultimately enhancing real-time warnings.

The enhanced Fujita (EF) scale, used to rate tornado intensity based on damage surveys, cannot be directly applied in real-time warning situations. This limitation has led to a critical gap in our ability to communicate the potential severity of an ongoing tornado event. The current impact-based warning (IBW) system, implemented by the National Weather Service, attempts to address this gap by providing qualitative descriptors of potential tornado impacts.

Recent studies have shown promising correlations between various TDS characteristics and tornado intensity. For instance, Emmerson et al. (2020) found that maximum TDS height is strongly correlated with peak wind speeds. Similarly, rotational velocity (v_{rot}) and the significant tornado parameter (STP) have been linked to tornado intensity (Smith et al., 2020). These findings suggest that a combination of radar-derived parameters and environmental data could provide a robust basis for real-time tornado intensity estimation, which is supported by the National Weather Service's

operational guidance on using such parameters. Nonetheless, the tornadic environment is extremely taxing for the operational meteorologist, and any automated program that can reduce workload is worthy of investigation.

This research aims to develop a novel method for real-time tornado impact assessment using TDS characteristics and population data. The primary goals of this study are to:

- Develop an algorithm that relates tornado debris signature (TDS) height, storm motion, and downstream population density and assesses expected impact.
- Implement this algorithm in a Python-based operational program that recommends tornado warning enhancement in situations when a strong or violent tornado is indicated and is threatening a highly populated area(s).
- Evaluate the performance of this system using past cases.

By achieving these objectives, this research has the potential to enhance the current tornado warning process by reducing forecaster cognitive load.

CHAPTER II

LITERATURE REVIEW

Tornado intensity estimation in real-time remains a critical challenge in operational meteorology. Recent advancements in radar technology, particularly the ability to detect and analyze tornado debris signatures (TDS), have opened new avenues for improving tornado intensity estimation and impact prediction. A TDS is defined as a valid velocity signature collocated with a correlation coefficient (CC) value below 0.90 and a reflectivity value above 30 dBZ. A differential reflectivity (ZDR) value near zero is not necessary but does increase confidence.

The identification and analysis of tornado debris signatures have become crucial in estimating tornado intensity. Bodine et al. (2013) found that TDS parameters tend to correlate with enhanced Fujita (EF) scale ratings from damage surveys. Their research revealed that during tornado intensification, certain parameters (such as 90th percentile reflectivity) tend to increase, while others (like 10th percentile correlation coefficient) tend to decrease. Importantly, they noted that maximum TDS height and volume typically occur 5-10 minutes after periods of significant damage. This time lag is indicative of the idea that some TDS characteristics might be useful for confirming tornado intensity rather than providing advance warnings; however, given the frequently above-average track length and duration of significant (EF2-3) and violent (EF4-5) tornadoes, they may hold value for advance warning based on early-life damage. The study also showed that violent tornadoes tend to exhibit much higher TDS heights compared to weak (EF0-1) and significant

tornadoes. This finding is of particular importance for identifying the most dangerous tornadoes in operational scenarios.

Building on this work, Emmerson et al. (2017) analyzed TDS associated with a range of significant and violent tornadoes. Examining 581 tornadoes rated EF2 or higher between May 2010 and December 2017 showed an impressive correlation ($r^2 = 0.59$) between TDS height and estimated maximum wind speeds. Furthermore, TDS height was a capable discriminator between significant and violent tornadoes, with all violent tornadoes showing TDS heights in excess of 15,000 feet above ground level. This suggests that TDS height could be a valuable tool for identifying high-end tornadoes in near-real-time. Adding azimuthal shear to the model in tandem with TDS height produced a linear r^2 value of 0.71 for predicting peak wind speeds (as estimated by NWS damage surveyors), showing impressive performance.

Gibbs (2016) explored the TDS for tornado intensity estimation, with a similar finding that showed a maximum height threshold of 10,000 ft having skill in discriminating EF2+ tornadoes. Combining TDS height with other parameters improved predictive accuracy. While TDS height showed lower skill than rotational velocity (v_{rot} , defined as $\frac{|v_{max}| - |v_{min}|}{2}$), it was not without value, as TDS height can be of particular use where v_{rot} fails: at distances far from the radar where interrogation of low-level features is not possible due to beam height.

While TDS can provide important information, v_{rot} has been shown to be a primary determinant of tornado intensity in multiple studies. Smith et al. (2020) found a strong correlation between v_{rot} and tornado intensity. Combining this with significant tornado parameter (STP) value and population density, they devised a multi-tier scale of tornado intensity and impact. STP is defined as:

$$STP = \frac{MLCAPE}{1500} \cdot \frac{2000 - MLLCL}{1000} \cdot \frac{ESRH}{150} \cdot \frac{EBWD}{20} \cdot \frac{200 + MLCIN}{150} \quad (2.1)$$

where:

- MLCAPE: mixed layer convective available potential energy
- MLLCL: mixed layer lifting condensation level
- ESRH: effective storm relative helicity
- EBWD: effective bulk wind difference
- MLCIN: mixed layer convective inhibition

and:

- The second term set to 1.0 when $MLLCL < 1000$ m and set to 0.0 when $MLLCL > 2000$ m
- The fourth term set to 30 when $EBWD > 30 \text{ ms}^{-1}$ and set to 0.0 when $EBWD < 12.5 \text{ ms}^{-1}$
- The fifth term set to 1.0 when $MLCIN > -50 \text{ kg}^{-1}$ and set to 0.0 when $MLCIN < -200 \text{ kg}^{-1}$

The Significant Tornado Parameter (STP) is a composite parameter that attempts to distill the multitude of ingredients favorable for right-moving supercells to produce EF2-EF5 tornadoes into a single value. Values above 1 are considered indicative of considerable potential for such events. As can be seen from the parameters, the STP considers environmental instability, moisture, available streamwise vorticity, shear, and inhibition to updraft growth, all major considerations when diagnosing supercellular environments.

The STP has been shown (Thompson et al. 2017) to be a useful predictor of tornado intensity, particularly in combination with v_{rot} . This is of particular interest, as most intensity estimation approaches use only radar-based interrogation, whereas STP is an environmental parameter that is known in advance of tornado or even storm development. Because v_{rot} is a real-time parameter, STP is a pre-existing synoptic-level parameter, and population density is pre-existing, the approach

of leveraging these three pieces of data offers strong real-time potential. It also highlights the importance of awareness of local population distributions and near-storm synoptic environments in anticipating tornado intensity and impact. While the approach of this research will not use STP, STP's known correlation to tornado intensity offers promising avenues for future combinations with other parameters, for determining appropriate environment in which to use the program, and for creating suitable null cases.

Gibbs (2016) identified specific v_{rot} thresholds that demonstrated impressive skill in discriminating between weak (EF0-1) and strong/violent (EF2+) tornadoes. A v_{rot} of 40 kts with the presence of a TDS (of any height) showed high skill, with a probability of detection (POD) of 0.915 and false alarm ratio (FAR) of 0.190 for EF2+ tornadoes. This is strong quantitative evidence for using v_{rot} operationally for real-time tornado intensity estimation. A higher v_{rot} threshold results in lower POD (though lower FAR as well). This could be partially due to beam height issues resulting in less representative v_{rot} values as compared to the true near ground-level v_{rot} . Thompson et al. (2017) further illustrated the importance of v_{rot} , showing a strong correlation between tornado damage rating probabilities and low-level v_{rot} . There was a strong negative correlation between tornado damage rating probability and circulation diameter. The study also found regional variation in tornado probability distributions due to damage indicator density, indicative of the need for local customization in operational usage.

Smith et al. (2020) further showed the correlation between tornado intensity and STP. The study also found that long-duration, high- v_{rot} (≥ 70 kts) scenarios in high-STP (≥ 6) environments were generally associated with the most intense tornadoes, adding not just a magnitude but a temporal perspective to intensity estimation.

There have been several approaches to creating real-time tools and methodologies for tornado intensity estimation. Karstens et al. (2023) used v_{rot} and STP to create percentile-based estimates of maximum wind speeds along a tornado path to create a probabilistic framework, of particular use given the current National Weather Service approach to communicating uncertainty. Applying the model to 115 tornadoes between 2020 and 2022 that provided 6,425 damage indicators, they found the model generally predicted peak tornado intensity within one EF scale rating value. Similarly, Mosier et al. (2022) created a web-based tool using v_{rot} , maximum STP within 80 km, population density, v_{rot} duration, and TDS presence. The tool provides both estimated EF scale rating and estimated peak wind speed, plus Impact-Based Warning (IBW) tags. Testing the tool on 99 tornadoes between 2020 and 2022, the authors found the maximum estimated wind speed identified by after-event damage assessment was within the predicted range about two thirds of the time. More than 80% were within 10 mph of the predicted range. Nonetheless, the drawback of the tool is the need for manual interrogation and data entry, adding workload in the operational environment. Meanwhile, Smith et al. (2024) created a quantitative impact-based decision support services (qIDSS) tool that estimates maximum wind speed range based on v_{rot} and STP. Using this method per tornado showed better performance than per scan, indicating the importance of tracking the temporal evolution of tornado and storm characteristics instead of relying on single scans, as these do not contain any sort of evolutionary information on a storm or tornado and can also suffer from contamination issues that will be filtered out in a multi-scan approach. These tools show significant promise in improving real-time evaluation of tornado intensity, while also highlighting the significant complexity of tornado dynamics and the consequent difficulties in predicting intensity and impact.

The National Weather Service incorporated many of the aforementioned research findings into their operational Impact-Based Warning (IBW) system. Current NWS guidance uses a tiered approach to tornado warnings based on v_{rot} , STP, and TDS presence, along with population density considerations (National Weather Service 2024), with tags applied based on a combination of factors. For example, the "catastrophic" tag requires a confirmed tornado (via visual confirmation or a TDS), $v_{rot} \geq 70$ kts, and $STP \geq 6$. TDS identification criteria require a velocity circulation signature at the lowest tilt and a correlation coefficient below 0.90 collocated with reflectivity above 30 dBZ, with near-zero differential reflectivity offering increased confidence. The training also heavily emphasizes consideration of environmental factors, particularly STP. For quasi-linear convective systems (QLCS), requirements are slightly modified. Pitfalls include things like vertical sidelobe contamination and false TDS appearances in inflow areas. Increased confidence is given by time and height continuity of signatures.

Smith et al. (2024) found good overlap between their aforementioned qIDSS tool and NWS warnings. Nonetheless, their tool showed potential for reducing FAR of "catastrophic" tags, underscoring the potential for operational improvement through use of such tools. Still, the need for manual interrogation and data input could be difficult in a heavy-workload operational environment.

Despite the success of these techniques and tools, there are still challenging limitations. Radar range and sampling can impact the accuracy of measurements, particularly at long distances from the radar site. The advantage of the TDS height method is that while lower-end tornadoes (associated with lower TDS heights) may not be detectable via this method at greater distances from the radar, the tornadoes of highest interest will be the least affected by this issue. Furthermore, the inherent limitations of the EF scale and available damage indicators can adversely affect verification efforts

and thus, model design (Gibbs 2016, Thompson et al. 2017, Smith et al. 2020). As mentioned, the Thompson et al. (2017) study highlighted regional differences due to density of damage indicators. Operationally, one of the most significant challenges is the time lag between tornado touchdown and the appearance of a detectable TDS and the additional time lag until maximum TDS height is reached. Bodine et al. (2013) noted it usually takes 5-10 minutes for a TDS to appear on radar once a tornado begins ingesting debris.

Future directions include improved integration of multiple data sources, including dual-polarization radar, mobile radar, and environmental data. Machine learning shows promise, while near-term and real-time high-resolution numerical model output can improve accuracy of environmental parameters. Regionally customized algorithms could produce better local accuracy as well. Lastly, increased automation and lower reliance on manual interrogation and data input can help reduce forecaster load and increase operational adoption.

SUMMARY OF LITERATURE

Recent advancements in radar technology coupled with similar advancements in research and operational practices have significantly improved real-time tornado intensity estimation, though challenges remain. Multiple studies have identified v_{rot} as a strong indicator of tornado intensity (Smith et al. 2020 and Gibbs 2016). TDS characteristics have also shown strong correlations with tornado intensity (Bodine et al. 2013, Emmerson et al. 2017, Gibbs 2016). The Significant Tornado Parameter has also been identified as a strong predictor (Smith et al. 2020 and Thompson et al. 2017). These findings have led to the development of a range of tools and techniques for real-time tornado intensity estimation, some of which have been incorporated into National Weather Service operational practices.

Nonetheless, several challenges remain in operational practice, including radar range and sampling issues, environmental mischaracterization, the use of manual methods in periods of high workload, and the inherent verification limits of the EF scale and availability of damage indicators. Despite these challenges, the identified parameters show significant promise for improving tornado warnings and public safety. The development of an automated tool could both reduce real-time operational workload and potentially improve the consistency and accuracy of tornado intensity estimates, addressing an ongoing need in severe weather forecasting and warning operations.

CHAPTER III

RESEARCH OBJECTIVES AND HYPOTHESIS

The primary objective of this research is to develop an operational program that enhances real-time tornado warning capabilities by integrating tornado debris signature (TDS) analysis with population impact assessment in a manner that helps to reduce operational workload.

The proposed program will ingest Level II radar data to identify and locate TDS, scanning all tilts to determine the maximum TDS height. Using established correlations between TDS height and tornado intensity, the program will then estimate the likely Enhanced Fujita (EF) scale rating of the tornado. If the tornado is determined to be of sufficient intensity (EF2 or greater), the program will then project the potential impact area using the radar-derived storm motion vector. It will then integrate this projection with GIS population data to evaluate both population density and total population at risk in the projected path. Once this analysis is complete, the program will assess whether the situation meets the established National Weather Service Impact-Based Warning standards for enhanced warning products and indicate to the user if these standards are met. The program aims to provide forecasters with rapid, objective, and automated guidance for critical warning decisions.

The author hypothesizes the combination of TDS height and downstream population density can reliably identify a substantial proportion of situations warranting tornado emergency warnings. Specifically, the author predicts that tornadoes with TDS heights above an established threshold

threatening areas of critical population density will correlate strongly with historical situations where tornado emergencies were issued or, in hindsight, should have been.

Secondary objectives, depending on the complexity of the implementation, include integrated dynamically updated GIS population data for the most up-to-date analysis, allowing for user-defined thresholds to adapt to local or situational needs (whether meteorological- or impact-based), and allowing for user-defined storm motion vectors to account for forecaster expertise and/or deviant storm motion.

This research hopes to contribute to the ongoing efforts to improve tornado warning efficacy by providing targeted and consistent warnings for some of the most dangerous events.

The program's performance will be systematically evaluated using standard verification metrics, including Probability of Detection (POD), False Alarm Ratio (FAR), and Critical Success Index (CSI). Lead time comparisons between the automated program and current NWS warning practices will be conducted to assess potential operational benefits, with particular focus on whether TDS height-based warnings can provide earlier detection for significant and violent tornadoes.

The analysis will evaluate performance differences across tornado intensity categories (EF0-1, EF2-3, EF4-5) to determine where the method demonstrates greatest utility and reliability. Additionally, statistical validation through bootstrap analysis will be performed to determine whether performance differences are statistically significant. This comprehensive evaluation framework will provide a scientific basis for assessing the program's potential contribution to operational meteorology beyond the technological development itself.

CHAPTER IV

METHODOLOGY AND ANTICIPATED LIMITATIONS

4.1 Methodology

The proposed operational program was developed using Python, leveraging its robust ecosystem for scientific computing and data analysis. The methodology for creating this program involved several key components, each addressing a crucial aspect.

The first step in this process was data ingestion and preprocessing. A module was developed to take in Level II radar data using the Python-AWIPS system, extracting the relevant radar products, including reflectivity, velocity, and correlation coefficient. Following data ingestion, the program focused on Tornado Debris Signature (TDS) identification. While the National Weather Service uses a variety of preexisting tools for TDS identification, this research implemented a custom algorithm based on the formal NWS operational criteria. These criteria are well-established and standardized in NWS impact-based warning operational guidance (National Weather Service, 2024), requiring the presence of a velocity couplet at the lowest radar tilt, correlation coefficient below 0.90, and collocation with reflectivity above 30 dBZ. A function was developed to scan all elevation tilts and determine the maximum TDS height, which is not part of standard NWS operational procedures but serves as the core innovation of this research. This was done by first establishing a valid TDS at the lowest tilt, then checking higher tilts for CC drops (indicating lofted tornadic debris). The program's custom implementation allows for automated processing,

the additional TDS height analysis, and integration with the population analysis components. In order to increase efficiency, the program first filters out all bins with reflectivity below 30 dBZ or CC above 0.9. These bins are filtered out from all plots, including velocity, reducing the number of velocity bins checked during couplet detection. Once the filtered dataset of velocity points was established, every bin within the user-set detection radius of a bin was checked, and the vector difference of the velocity values is calculated. If this difference was above the user-set threshold, a couplet was plotted between the two bins. While using single bins might increase the chances of noise creating false detections, the multiple requirements of the prefiltering vastly reduced the size of the velocity dataset and did well in preventing this.

The EF Scale rating was then estimated via a lookup table based on established research to correlate the maximum TDS height with the likely rating. If the rating was above an established threshold (the 8,000 ft EF2+ threshold established by NWS IBW guidance), the function activated the population analysis. For tornadoes that triggered the population analysis, the program projected the potential impact area using the storm motion vector. The projection included error bounds to account for uncertainties in both storm motion and track width.

Next, the program integrated a GIS model to load and query population density data using a CSV file of approximately 31,000 US cities with their respective coordinates and population (Simple Maps, 2024). A function was created to calculate the number of areas above the established threshold in the projected impact area. Next, the program used these outputs in tandem with TDS parameters to make warning product enhancement recommendations. The program then outputted all relevant population centers and allowed the user to copy these for easy inclusion in AWIPS. For the purposes of this research, a population center was defined as an incorporated municipality,

census-designated place, or unincorporated community with a recorded population in the U.S. Census data or recognized population estimates. To improve usability, a graphical user interface was designed to allow easy operation of the program. Configuration options were implemented, including user-adjustable thresholds.

Finally, the program's performance was validated using a set of historical tornado cases with known outcomes and damage surveys. Cases were first found using the Iowa State University Iowa Environment Mesonet archive of NWS-issued tornado warnings. Because nationwide deployment of dual-polarization radar was not completed until 2016, cases were temporally restricted to 2016-2024. In order to maintain relative geographic uniformity, cases were spatially restricted to five adjacent National Weather Service County Warning Areas: Memphis, Tennessee; Birmingham, Alabama; Huntsville, AL; Jackson, Mississippi; and Peachtree City, Georgia. Once this spatial and temporal prefiltering was completed, randomly selected subsets of cases for testing were taken from three supersets: tornado emergencies, tornado warnings with a particularly dangerous situation tag, and tornado warnings not associated with either (deemed "normal" warnings). This resulted in three test sets: 22 tornado emergencies, 50 PDS tornado warnings, and 50 normal tornado warnings. For each case, the program was fed the relevant archived Level II radar data and compared against the issued warnings and outcomes.

	Emergency Cases	PDS Cases	Normal Cases
Birmingham, AL	5	13	5
Huntsville, AL	2	5	3
Jackson, MS	4	14	22
Memphis, TN	6	12	12
Peachtree City, GA	5	6	8

Table 4.1: Geographic distribution of selected cases

Performance metrics were then calculated, including the false alarm ratio (proportion of cases where the program triggered enhanced warnings that were unwarranted by eventual tornado intensity), probability of detection, critical success index, and difference in program lead time from forecaster lead time. The Critical Success Index (CSI) was selected as a primary verification metric for this study due to its comprehensive evaluation of warning performance. Unlike Probability of Detection (POD) or False Alarm Ratio (FAR) alone, CSI incorporates both missed events and false alarms into a single measure that ranges from 0 to 1, with higher values indicating better overall performance. This metric is particularly valuable for evaluating severe weather warning systems because it penalizes both types of error that impact public safety: failing to warn for actual tornadoes and issuing unnecessary warnings that could lead to warning fatigue. Unlike the Heidke Skill Score (HSS) or True Skill Statistic (TSS), CSI excludes correct negatives from its calculation. This exclusion can be useful for tornado verification where correct negatives would overwhelmingly dominate the statistic due to the rare nature of tornadic events, potentially masking meaningful differences in detection performance. Additionally, defining appropriate null cases for tornado warnings presents significant challenges—the vast spatial and temporal domain where tornadoes

don't occur would create an artificially large set of correct negatives that don't meaningfully test the algorithm's discrimination ability. By focusing exclusively on cases where warnings were issued or tornadoes occurred, CSI provides a more stringent and relevant assessment of performance in the critical scenarios that matter most for public safety. That being said, developing a true null case to assess the ability of the program to distinguish non-tornadic radar data from tornadic radar data is important; this is discussed in section 8.3.3.

Regressions between debris height and EF rating were also calculated, with Spearman rank-order correlation coefficients and p-values calculated. Final statistics included both performance of the program and comparison to current methods.

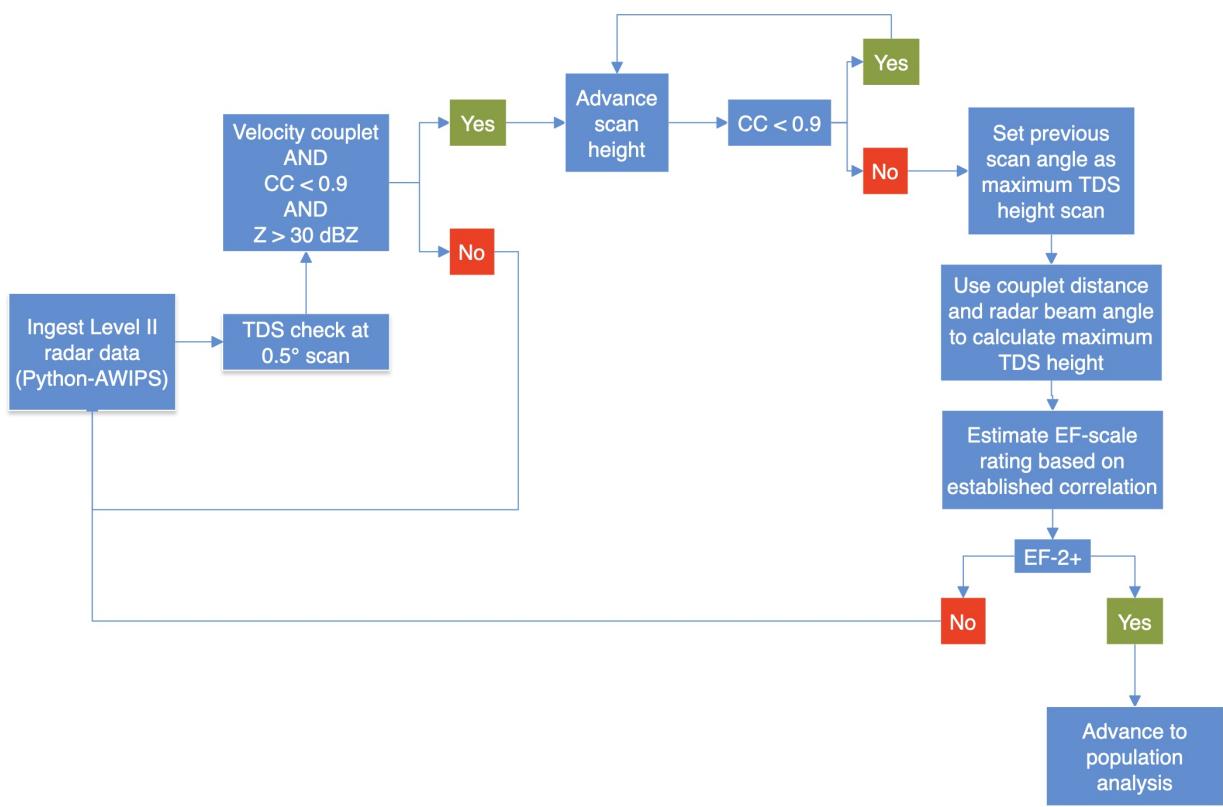


Figure 4.1: Algorithm flowchart

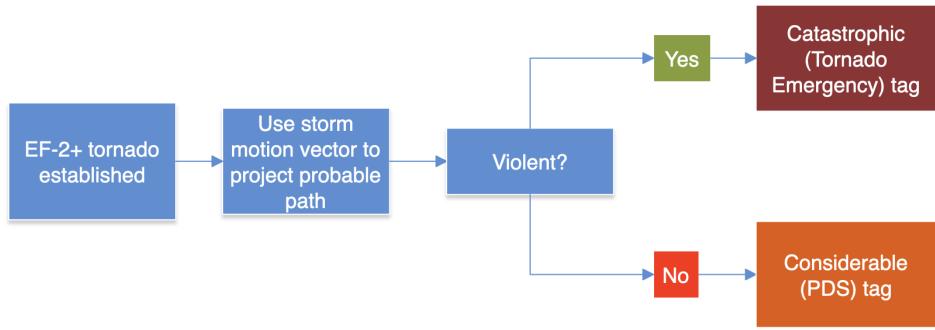


Figure 4.2: Population analysis flowchart

4.2 Population Analysis

While the debris height analysis forms the meteorological foundation of this research, the population impact assessment is equally critical to the program's operational value.

For tornadoes meeting the threshold for strong or violent intensity (EF2 or greater), the program calculated a forward projection of the potential impact area using the storm motion vector. This projection created a trapezoidal polygon that extends from 10 km behind the detected couplet to the distance the storm would travel in 60 minutes based on its current speed and direction. The polygon widens from 20 km at the base to 30 km at the far end to account for track variability and increasing uncertainty with time. This polygon construction was based on standard NWS construction.

The population database consisted of approximately 31,000 U.S. cities with their geographic coordinates and population estimates (Simple Maps, 2024). Each location in this database represented an established community, ranging from small towns to major metropolitan areas. The program queried this database to identify all population centers that fall within the projected impact polygon.

A population center was defined in this research as any incorporated municipality, census-designated place, or unincorporated community with a recorded population. To focus on areas where impacts would affect significant numbers of people, the program applied a user-adjustable minimum population threshold to accommodate different regional population distributions and operational requirements.

For each identified population center in the storm's path, the program calculated the estimated time of arrival (ETA) based on the distance and the storm's motion vector. This temporal component is critical for prioritizing warnings and emergency response. The population centers and their ETAs

were then presented in a sortable table interface that allows operational users to quickly assess which communities face the most imminent threats and where the greatest population impacts may occur. With a single click, users could copy the list of cities to the clipboard so that it can be quickly pasted into the AWIPS WarnGen function. This allowed for both increased consistency and efficiency in warning creation and dissemination.

4.3 Anticipated Limitations

While the project shows promise, it is not without limitations.

4.3.1 Radar Distance

Research shows that the minimum maximum TDS height for violent tornadoes is typically around 15,000 feet, with EF3 tornadoes ranging from about 3,000 to 28,000 feet. This places limitations on radar distance, as precipitation mode tilt angles range from 0.5 to 19.5 degrees. This means signatures of interest must be at least 9 miles from the radar site, optimally a bit farther to distinguish from overlapping lower ratings.

4.3.2 Time for TDS to Appear

Once a TDS appears, it is because the tornado is already causing damage. Typically, it takes approximately 5-10 minutes for a TDS to appear once debris has been ingested. If a tornado starts in a rural area, this may still allow for reasonable lead time, but in cases where a tornado touches down in or near an urban area (such as the 2011 Joplin tornado), this method will likely lag too much to be of use.

4.3.3 QLCS Tornadoes

QLCS tornadoes represent a significant challenge for radar operators, as they often appear and dissipate very quickly, sometimes even between radar scans. As such, they will often be of a duration insufficient to make this method of use. On the other hand, they are rarely of an intensity that would warrant consideration of an enhanced tornado warning.

4.3.4 Available Debris

There may be cases in which a tornado is of sufficient intensity to warrant consideration of a tornado emergency, but the debris it has ingested is not sufficient in either quantity or quality to produce the needed radar signature. For example, if a violent tornado traverses relatively barren land, it may ingest only a bit of uniform soil, which may not produce the necessary reflectivity or correlation coefficient values for this method. Furthermore, it is not uncommon for a robust updraft to keep debris lofted well after a tornado has dissipated. As such, this could trigger false positives in the system.

4.4 Data Management

The project requires both radar data and GIS population data. Both live and archived radar data will be obtained via the NSF Unidata public EDEX server via the Python-AWIPS protocol. Static population data will be retrieved from USGS. If live data acquisition becomes feasible, a viable source will be determined.

Results will be analyzed on a case-by-case basis, comparing lead time (in terms of product enhancement recommendations) given by the algorithm as opposed to those issued in real-time by the National Weather Service. From these randomly selected cases (from the larger data set of historical tornado emergencies), summary statistics and skill statistics will be calculated, and overall performance compared to manual product enhancement will be determined. Cases in which a tornado emergency was not issued but a violent tornado struck a population-dense area will also be examined.

CHAPTER V

PROGRAM STRUCTURE

5.1 Introduction

This Python program provides a comprehensive radar-viewing interface with both graphical and algorithmic functionalities. It uses several powerful libraries, including Tkinter for the GUI, Matplotlib for plots, MetPy for radar file parsing, Boto3 for AWS interaction (downloading radar data), and Numpy/Scipy for numerical and scientific computations. Its principal objectives are:

- To download and manage NEXRAD Level 2 radar data from AWS.
- To provide an interactive user interface (UI) for visualizing base reflectivity, correlation coefficient, base velocity, and storm-relative velocity.
- To detect and display rotational couplets, estimate their possible severity (via EF-rating proxies based on debris height), and attempt to track debris columns in 3D space.

A csv file of approximately 31,000 cities (Simple Maps, 2024) and their associated populations and coordinates is used for the population analysis function. This CSV file is static, containing a list of US cities by name, their populations as of approximately October 2024, and their coordinates. The program is laid out in a single file that defines a series of helper functions, classes, and a main RadarViewer class (which initializes the entire graphical interface).

5.2 Global Imports and Configuration

At the start, the script imports its main dependencies:

- **tkinter** and **ttk** for the GUI.

- **matplotlib** and its submodules (`pyplot`, `FigureCanvasTkAgg`, `NavigationToolbar2Tk`, etc.) for plotting radar data in embedded Tkinter windows.
- **metpy.io.Level2File** and **metpy.plots.ctables** for reading and color-tabling radar data from NOAA NEXRAD Level 2 files.
- **boto3**, **botocore**, **Config** for AWS S3 interaction (to download the NEXRAD data).
- **numpy**, **pandas** for numerical and tabular data handling.
- **concurrent.futures.ProcessPoolExecutor** and **threading** for parallel processing of couplet detection.
- **scipy.ndimage** for label-based segmentation and filtering data in 2D arrays.
- **json** for storing/loading user “cases” from external files.

A number of smaller imports handle advanced plotting elements like 3D transformations, patches, polygons, path effects, and so forth. The script also predefines `cites.csv` for caching city data.

5.3 Functions and Utilities

5.3.1 `process_chunk`

This function is meant to run in a separate process (via `ProcessPoolExecutor`) to detect couplet centers in a piece of the velocity data. It:

1. Converts the `max_range_mi` from miles to kilometers, storing it in `max_range_km`.
2. Iterates over the indices in `chunk`.
3. Skips data points outside the maximum range or with invalid velocity values (NaNs).
4. For each valid velocity “`max_vel`,” it looks for other velocities “`min_vel`” in a local search window, ensuring the azimuth difference is not too large and the distance is not beyond `max_distance_km`.
5. Computes rotational velocity “ $v_{rot} = \frac{|max_vel - min_vel|}{2}$ ”, and if that surpasses `vrot_threshold_ms`, it records the geometric center of the couplet.
6. Manages a dictionary `couplet_centers` to store only the largest `vrot` per approximate location, ensuring each center is uniquely identified.
7. Returns a final list of all the best couplet centers found within that chunk.

5.3.2 `rename_duplicates`

When loading JSON data with repeated keys, standard Python would overwrite earlier keys.

This function is used as an `object_pairs_hook` so that repeated keys get a suffix (-1, -2, etc.) to preserve all data.

5.3.3 `format_size_mb`

A utility that converts a size in bytes to megabytes, returning a string with one decimal place.

5.3.4 `create_velocity_colormap`

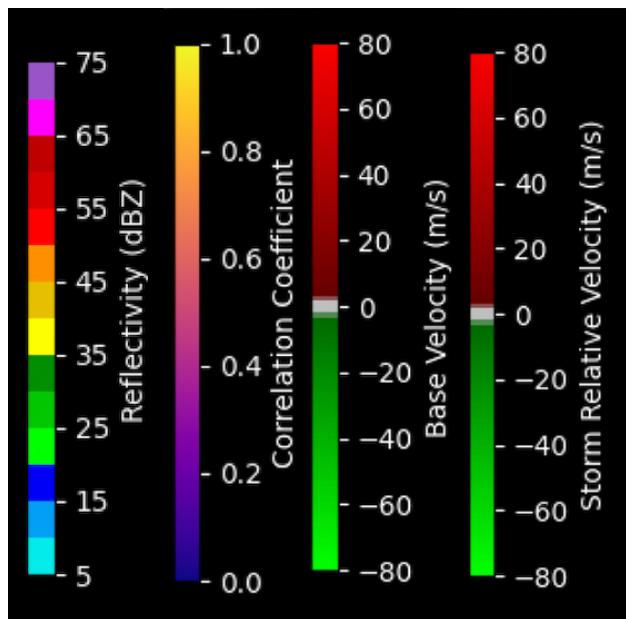


Figure 5.1: Color maps for the radar plots

This function constructs a custom colormap spanning from greenish for inbound velocities through grays for near-zero to red for outbound velocities.

5.3.5 add_motion_polygon

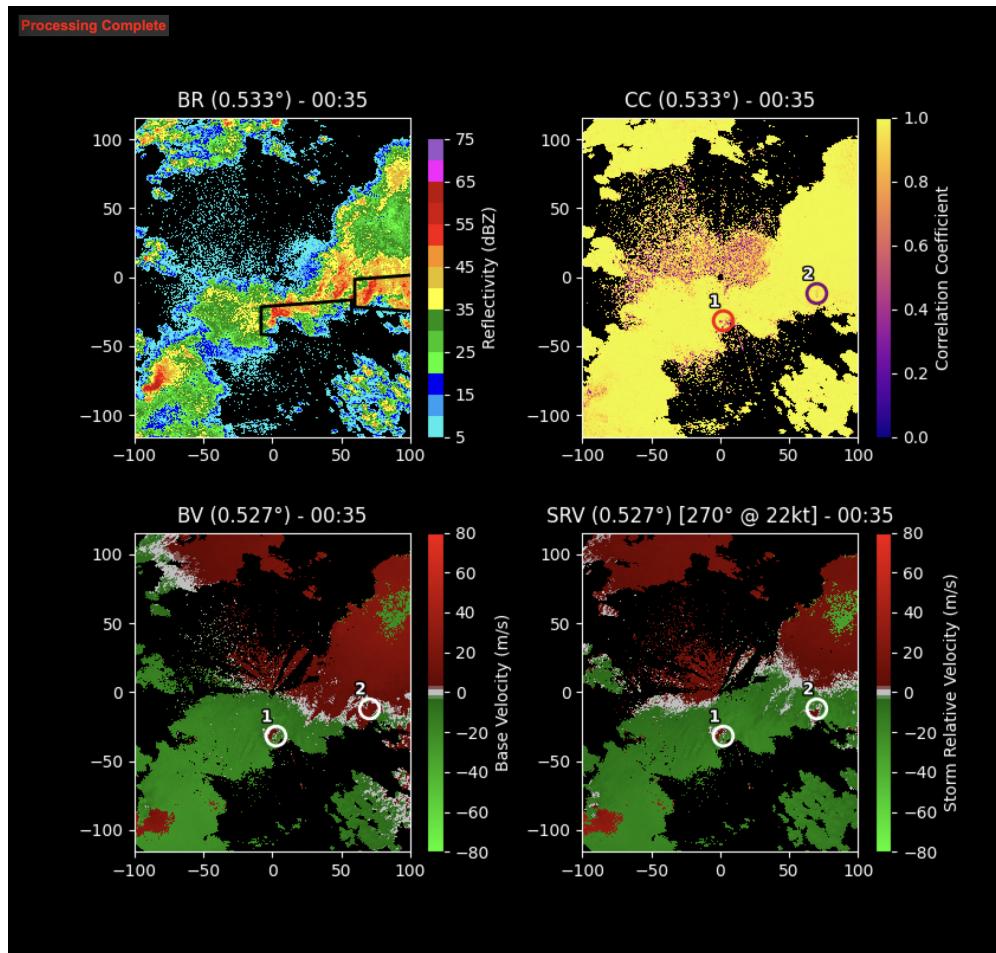


Figure 5.2: Warning polygons generated by the function seen in the reflectivity plot (distances from radar site are shown along axes in kilometers)

Creates a polygon that extends along the storm's motion vector. The polygon is determined by the function `calculate_polygon_vertices`, which is itself defined based on AWIPS parameters. The polygon has a higher z-order so it appears above the main radar data.

5.3.6 calculate_polygon_vertices

Given a storm direction, speed, and an initial couplet location (x, y), this computes the polygon corners. The shape is a trapezoidal area that extends behind the couplet point 10 km and ahead

by the distance the storm will cover in an hour based on its storm motion vector. The widths are 20 km at the base, enlarging to 30 km at the far side. This helps highlight potentially impacted areas downstream of a rotating storm and serves as the search area for the population analysis function.

5.3.7 `is_point_in_polygon`

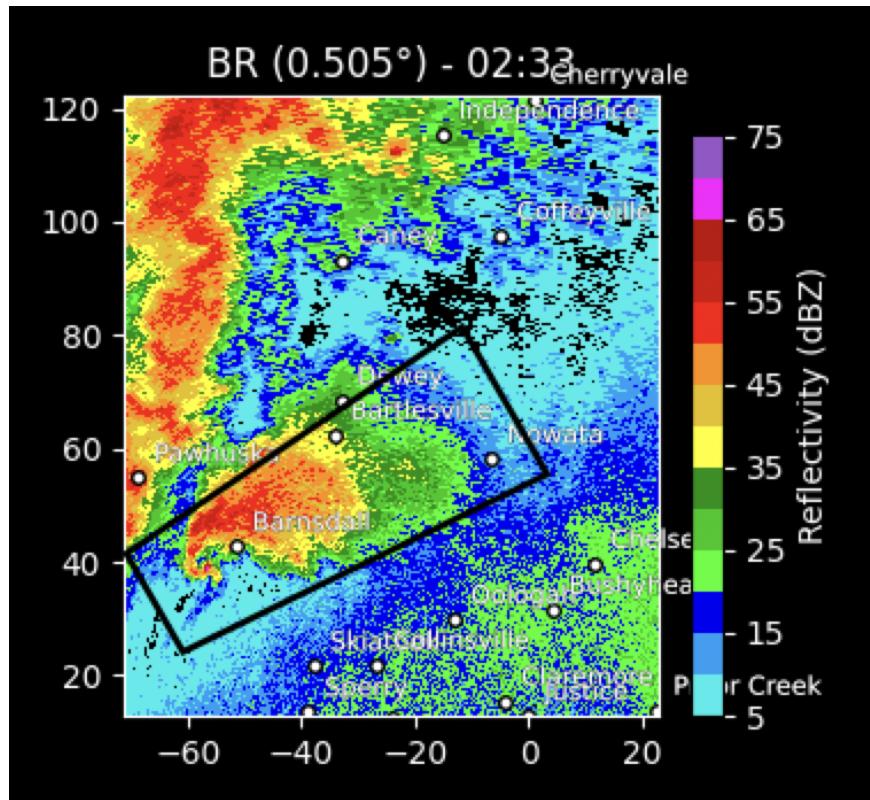


Figure 5.3: This code checks for loaded cities inside a polygon.

This determines if (x, y) lies inside a polygon defined by an array of vertices. This is used to see if a city is within a storm motion polygon.

5.3.8 `calculate_parallel_distance`

This projects the vector from the couplet centroid to the city onto the storm motion axis. This yields a distance in the direction of storm motion.

5.4 Custom Dialogs and Windows

5.4.1 CoupletCategoryDialog

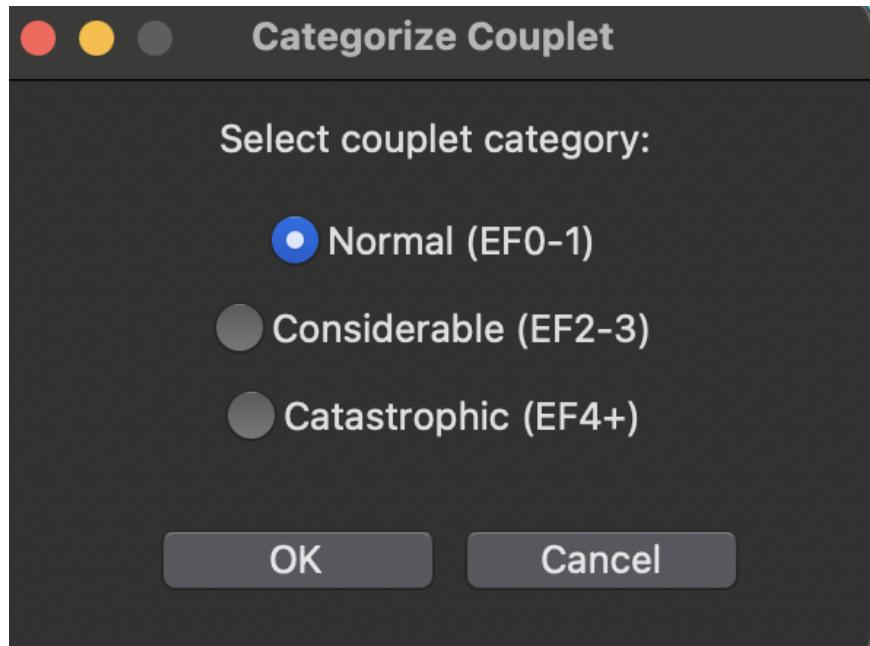


Figure 5.4: The manual couplet dialog

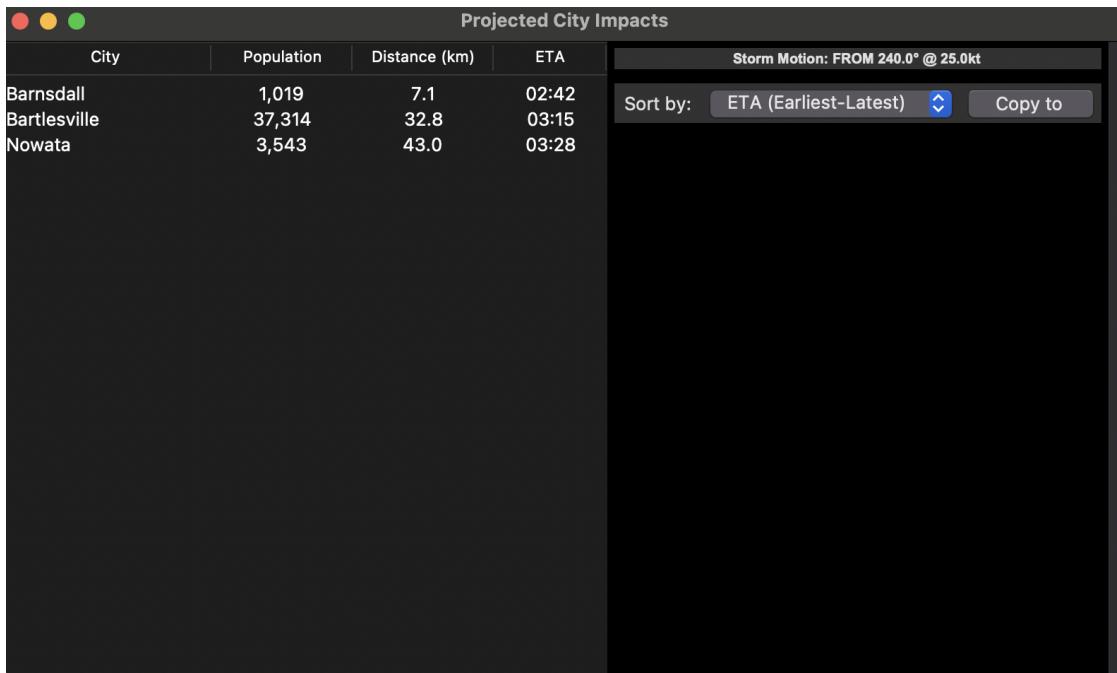
A small pop-up dialog for the user to manually label a newly added couplet as “Normal (EF0-1),” “Considerable (EF2-3),” or “Catastrophic (EF4+).” This returns the chosen category along with the (x, y) location. If the program fails to detect a couplet or TDS, this allows the user to right-click and add one at the severity level of their choice. Doing this adds it to the couplet list and leverages the storm motion vector to perform the same population analysis as automatic couplets. This lets users still take advantage of the efficiency of the automatic population analysis even if the program fails to detect a couplet.

5.4.2 RadarDisplayWindow

This is a crucial class that creates a separate Tkinter top-level window to display and interact with the radar data plots. It:

- Builds a `matplotlib` figure with four subplots, each for a different radar product: Base Reflectivity (BR), Correlation Coefficient (CC), Base Velocity (BV), and Storm-Relative Velocity (SRV).
- Controls reading labels at the bottom (showing values at the mouse cursor's position: reflectivity in dBZ, velocity in m/s, correlation coefficient, etc.).
- Allows manual addition of couplets by right-clicking (a context menu triggers `add_couplet_at_cursor`).
- Maintains a `Treeview` table listing detected couplets, with columns for the couplet number, rotational velocity, debris column height, and EF rating estimate. Also, it can highlight “Considerable” or “Catastrophic” couplets with flashing text and markers.
- Implements synchronized zooming and panning across all four subplots in `on_mouse_release` and `on_scroll`.
- Has an embedded progress label for ongoing computations (like couplet detection).
- Has logic for “flashing” intense couplets to catch the operator’s attention.
- Employs `draw_couplet_markers(...)`, which draws circles and text labels for each couplet, controlling color (yellow, red, or purple) depending on EF rating.
- Employs `update_couplet_table(...)`, which fully repopulates the `Treeview` widget when the user changes the scan time.
- Uses `flash_warning_text(...)`, which toggles the color of text and markers for “Considerable” or “Catastrophic” couplets every second, similar to flashing them on the plots.
- Implements `on_couplet_double_click(...)`, which triggers an analysis of city impacts.
- `manual_couplets` (dictionary) that accumulates user-added couplets keyed by time string.

5.4.3 CityImpactsWindow



The screenshot shows a window titled "Projected City Impacts". The window has a dark background and a light gray header bar. The header bar contains the title "Projected City Impacts" and a status message "Storm Motion: FROM 240.0° @ 25.0kt". Below the header is a table with four columns: "City", "Population", "Distance (km)", and "ETA". The table contains three rows of data:

City	Population	Distance (km)	ETA
Barnsdall	1,019	7.1	02:42
Bartlesville	37,314	32.8	03:15
Nowata	3,543	43.0	03:28

Below the table is a control bar with a "Sort by:" dropdown set to "ETA (Earliest-Latest)" and a "Copy to" button.

Figure 5.5: City Impact Window

When the user double-clicks a couplet, the system can calculate which cities might be impacted via the motion polygon. Those cities and their ETA (in minutes) appear in this table window.

Features:

- Sorting controls: sort by name, population, or earliest-latest time of arrival.
- A “Copy to Clipboard” button to copy the uppercase city list for quick dissemination, matching standard NWS warning formatting.

5.4.4 Debris3DWindow

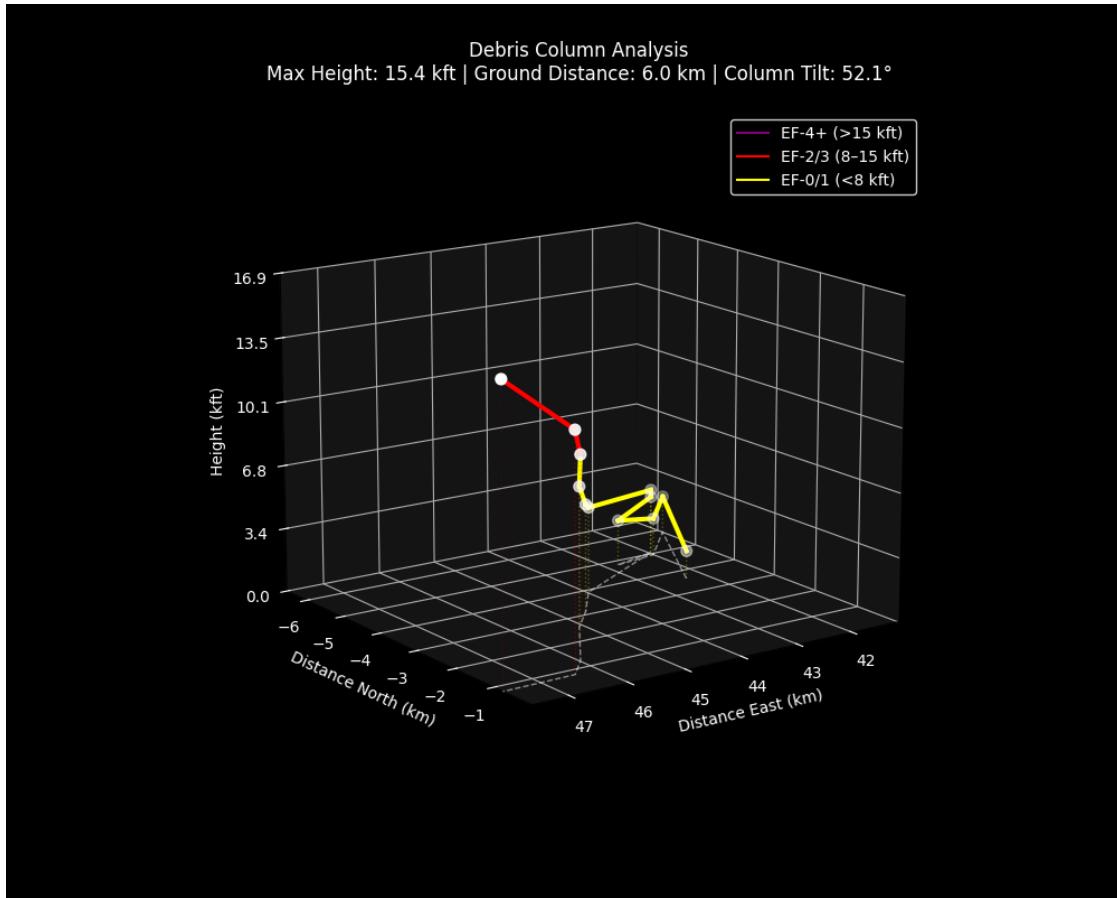


Figure 5.6: Three-dimensional debris tracker

Provides a `matplotlib` 3D figure to visualize a “debris column,” i.e., vertical extension of debris as indicated by correlation coefficient across different tilts. It:

- Creates a single `Axes3D` subplot.
- Plots lines and markers that represent the track from ground level up to maximum height. Each marker’s position is given by the centroid of its associated CC cluster.
- Colors lines by approximate EF rating thresholds (yellow = EF0–1, red = EF2–3, purple = EF4+).
- Adds a legend that clarifies the color-coded significance.
- Dynamically sets axis limits and labels to maintain a good perspective view.

5.5 Main Application: RadarViewer

5.5.1 Structure and Purpose

The RadarViewer handles:

1. Configuration of user input controls for radar site, date/time, product selection, and display preferences (e.g., filtering by reflectivity or correlation coefficient thresholds for visualization purposes).
2. Downloading data from AWS S3 for a chosen `start--end` time range and radar site.
3. Listing available times in a Listbox.
4. Stepping forward/backward in frames or auto-playing them.
5. Adjusting parameters for couplet detection.

5.5.2 Key Data Attributes

- `motion_polygons`: dictionary to store polygons in the reflectivity view for each time. These are the polygons used for population analysis and are similar to an NWS warning polygon.
- `radar_sites`: mapping from site ID (like '`KABR`') to a string describing that location. This allows users to select either a radar site code or a city and state.
- `available_times`: a list of time strings (HH:MM) representing the frames loaded from AWS. This shows users all available radar scan times.
- `available_products`: a dictionary keyed by time string; each entry includes '`products`' (the codes found) and '`tilts`' (unique tilt angles found). This shows the user which products (among reflectivity, base velocity, and correlation coefficient) are available. Note that the program calculates storm relative velocity by using the vector difference between the base velocity product and storm motion vector. This also shows all available tilts to the user.
- `couplet_positions_cache`: a dictionary that caches detected couplets for each time slice. Doing this prevents couplet detection from running again when a user scrubs away and back to a scan time that has already been analyzed.
- `debris_points_cache` and `couplet_tracking_cache`: hold advanced 3D and tracking data. This holds data on CC-drop centroids for both height analysis and three-dimensional plotting.
- `cases`: for saving and loading configuration states (start date, site, etc.) from an external JSON file. This allows users to access historical cases without having to manually enter the parameters every time.

5.5.3 Case Management

The program allows a user to save a “case” (radar site, start time, end time, product selections, etc.) to a JSON file for quick reloading. The relevant functions handle reading/writing that JSON, plus ensuring repeated keys don’t cause collisions (via `rename_duplicates`).

5.5.4 Loading Radar Data from AWS

This:

- Retrieves the `year/month/day/site` prefix from the user’s input date and site, then lists objects from the NOAA NEXRAD Level 2 bucket.
- Filters files by time range (start and end HHMM).
- Downloads each file in small chunks, updating the UI progress bar and status label as it accumulates data.
- Loads each result with `metpy.io.Level2File`, reading available tilts and sub-products (REF, RHO, VEL, etc.).
- Collects available times and available products for each time slice.

After finishing, it populates the `Listbox` with all loaded times.

5.5.5 Displaying a Radar Frame

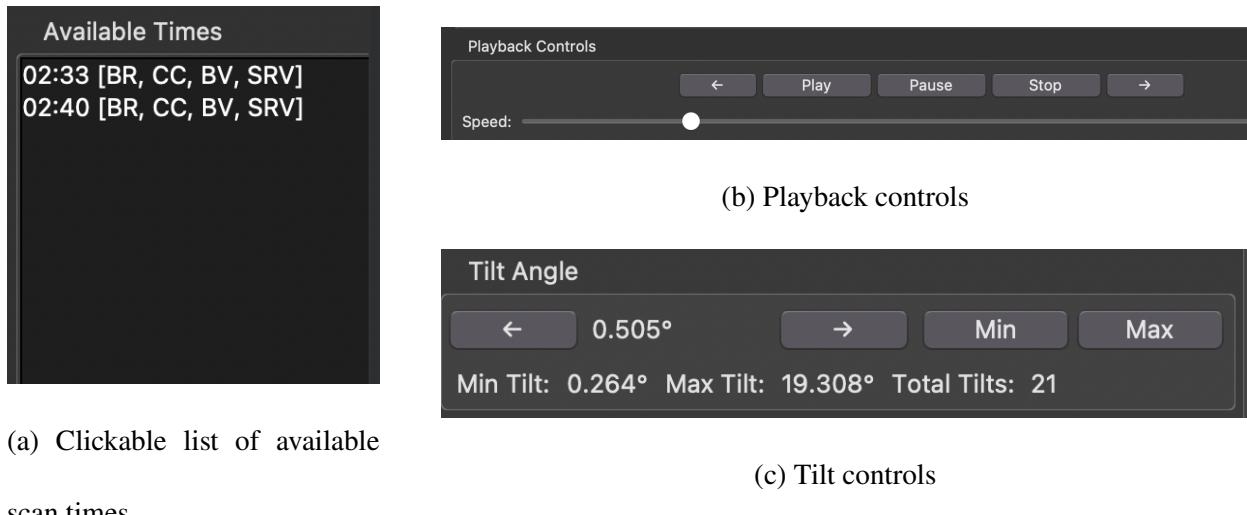


Figure 5.7: Radar display controls

Once data is loaded, the user can select a time from the Listbox or use the playback controls.

This:

- Ensures there is a `RadarDisplayWindow`, creating one if needed.
- Clears the axes or re-initializes them for a fresh plot.
- Retrieves the relevant `Level2File` object from `radar_data_cache`.
- Gathers the list of products (BR, CC, BV, SRV) that were found for this time.

Calls specialized sub-plotting methods:

- `plot_reflectivity`
- `plot_correlation`
- `plot_velocity`
- `plot_storm_relative_velocity`

It then:

- Resets or updates the `couplet_table` if couplets exist in `couplet_positions_cache`.

- Respects the user’s city display choice, plotting city markers and names if turned on.
- Optionally triggers couplet detection automatically, depending on the “Auto-calculate couplets” setting.

5.5.6 Plotting Methods

Each of these obtains a specific product from the `Level2File` sweeps, builds a 2D array (azimuth vs. range), then calls `pcolormesh` to paint it in the appropriate color scale. Additional logic handles filtering out data that fails reflectivity or CC thresholds. The relevant axes object is updated, a colorbar is attached, and the axis is labeled with a suitable title.

5.5.7 Coupleット Detection Workflow

Most of the logic for detecting a rotational couplet occurs in `def detect_couplets`. It:

1. Aligns reflectivity and CC data to the velocity grid.
2. Create a mask that eliminates values outside thresholds or that are invalid.
3. Identify contiguous clusters of valid velocity data.
4. For each cluster, creates a parallel `process_chunk` job to examine all $(az_idx, range_idx)$ gate pairs in the cluster, searching for inbound/outbound velocity pairs that meet rotation thresholds.
5. Recombine results and remove duplicates within a certain minimum separation distance.
6. For each final couplet, run `analyze_couplet_height` to attempt a vertical extension analysis (using CC across multiple tilts) to guess a debris height. This is done by searching for the centroid of low-CC clusters in the area of the couplet at each tilt, then iterating to the next tilt. This is then used to infer an EF rating level.
7. Store these final results in the `couplet_positions_cache` for the relevant time slice.

5.5.8 Debris Height Workflow

- Once a couplet is established at the lowest velocity tilt, the program begins the debris height search.

- The program begins with the lowest CC tilt. It searches for a cluster of low CC bins within a 5 km radius of the couplet centroid. A cluster is defined as a collection of CC bins that are connected (defined as being adjacent in a 3 by 3 grid) in which all bins have a CC value less than 0.9, usually 15 bins.
- If a cluster is found, the program stores its centroid, then iterates to the next tilt, using a 5 km radius around the previous centroid to search for a cluster.
- Once the program reaches a tilt at which no cluster is found, it uses the previous tilt and its centroid's distance from the radar to determine its height.
- This height is then correlated to a likely EF scale range based on NWS and literature guidance.

5.6 Algorithm Options

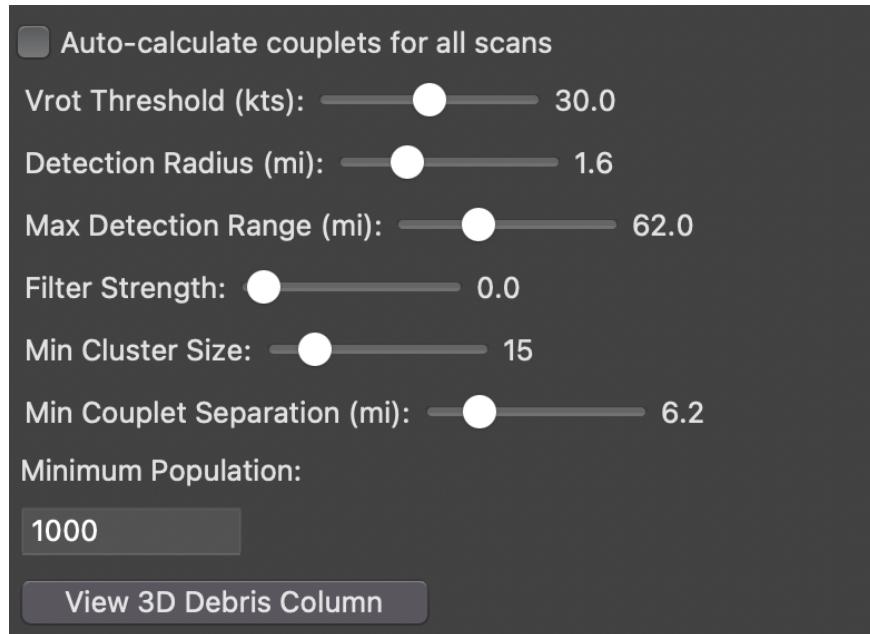


Figure 5.8: Algorithm options section

- v_rot threshold: this allows the user to specify the minimum rotational velocity to establish the presence of a couplet.
- Detection radius: this defines the maximum distance by which two velocity values can be separated to be considered part of the same couplet.
- Max Detection Range: this defines the maximum distance from the radar at which the program will search for couplets.

- Filter Strength (deprecated): this applied a filter to velocity data to reduce noise. It attempted to reduce false positives by applying a median filter to velocity bins, which assigns each bin the median value of the set of points including that bin and its adjacent bins. However, this significantly decreased the POD.
- Min Cluster Size: This determines the number of adjacent low-CC bins the program must find to determine a cluster.
- Minimum Couple Separation: This determines the minimum separation between couples in order to reduce duplicates. The maximum strength couplet is assigned when duplicates exist.
- Minimum Population: This determines the minimum population a city in the database must have to be included in population analysis.

5.7 Additional Functions

5.7.1 Manual Addition of Couplets

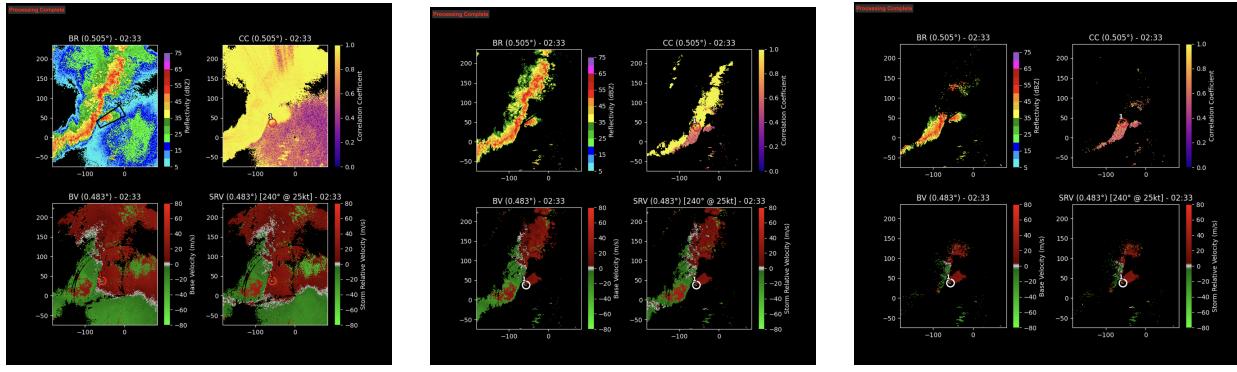
As the program cannot successfully detect all couplets, users who believe a couplet exists and have an estimate of the associated tornado's strength can right-click on any plot and manually add a couplet. The program will ask where it is a normal (EF0-1), PDS (EF2-3), or Emergency (EF4-5) couplet, then populate the couplet table with the manual couplet. This allows the user to quickly generate a population analysis even if the program does not automatically pick up on a couplet. How a user estimates maximum strength will depend on NWS operational guidelines, but current procedure involves evaluating rotational velocity, the presence of a TDS, and STP. Users may also use the tilt function to estimate maximum CC drop height, though this is not in current NWS guidelines.

5.7.2 Population Analysis

The population analysis feature helps assess the potential impact of severe weather on populated areas. When the program or user identifies a rotation couplet (potential tornado signature), they can double-click on it to see which cities lie in the projected storm path. The code calculates

a storm motion polygon based on the storm's direction and speed, then identifies cities from a database (uscities.csv) that fall within this polygon. The analysis filters cities based on a minimum population threshold (defaulting to 10,000 but user-adjustable), calculates the estimated time of arrival (ETA) for each city, and presents this information in a sorted table. The system accounts for the storm's motion vector to project accurate arrival times, allowing the user to prioritize warnings and resources for the most populated areas in harm's way. Users can sort this list by ETA, population, or alphabetically. They can then, in one click, export to the clipboard a list of all the cities in the selected sorting order in the same formatting (all caps) as a typical warning, which can then be pasted into AWIPS' WarnGen function instead of manually typing a list, potentially saving time and increasing coverage during time-critical warning situations.

5.7.3 Visual Filters

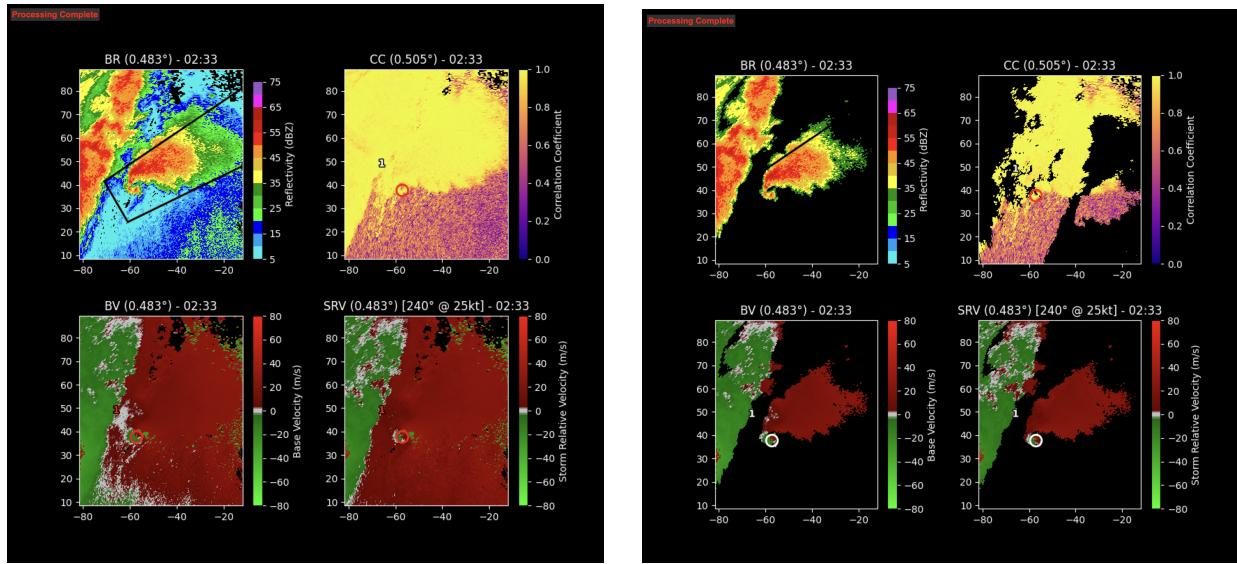


(a) Standard Display

(b) >30 dBZ ref

(c) >30 dBZ ref, <0.9 CC

Figure 5.9: Visual filter options



(a) Unfiltered

(b) >30 dBZ ref

Figure 5.10: Figure 5.9 zoomed on area of interest

While the NWS requirements for couplet existence (collocated reflectivity of at least 30 dBZ and collocated CC less than 0.9) are hardcoded for the sake of couplet detection, the user can

optionally enable visual masks that filter out all reflectivity data below 30 dBZ and/or all CC data greater than 0.9, allowing users to more easily locate and focus on relevant storms and features.

5.7.4 3D Viewer

The 3D debris viewer provides users with a comprehensive visualization of tornado debris signatures detected by radar. When a user selects a rotation couplet of interest, they can view its associated debris column in three dimensions, helping to understand the vertical structure of the tornado and mesocyclone. Using the x,y coordinates of each CC cluster centroid and its associated height, the debris column is plotted and ground distance covered and column tilt angle are calculated. Different segments are color-coded in a manner corresponding to their associated estimated EF rating.

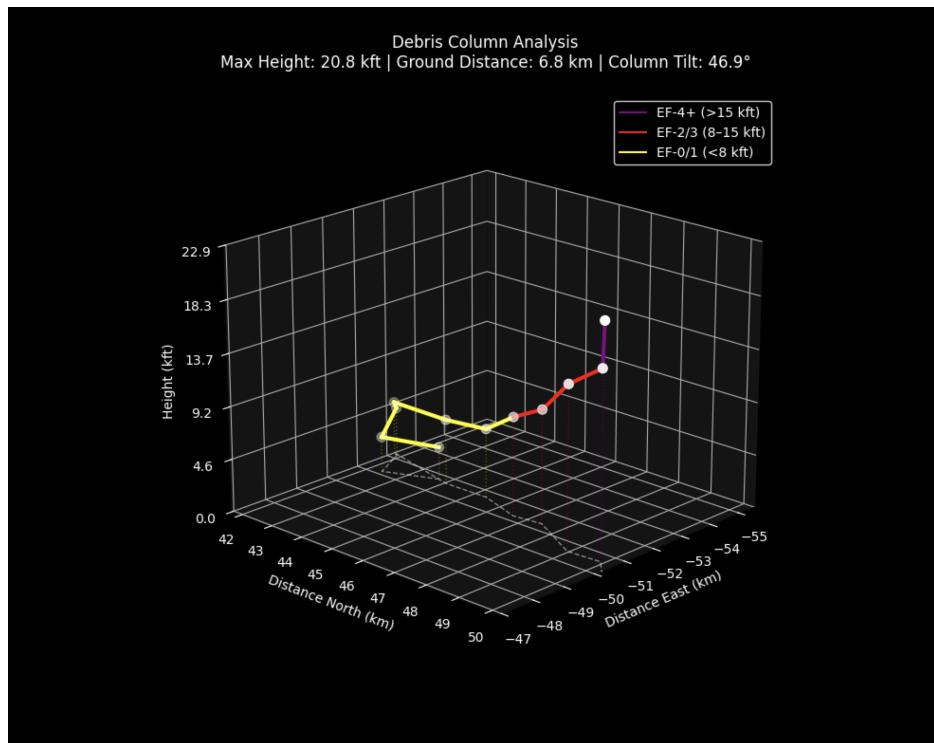


Figure 5.11: 3D debris viewer window

5.8 Full Interface

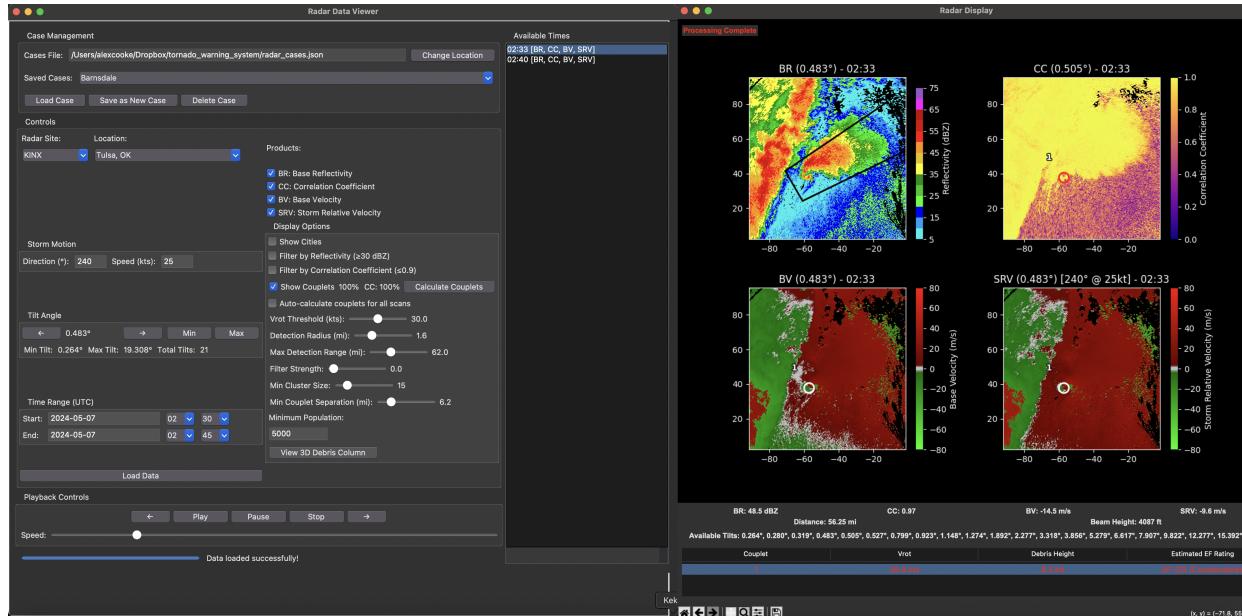


Figure 5.12: Full program interface

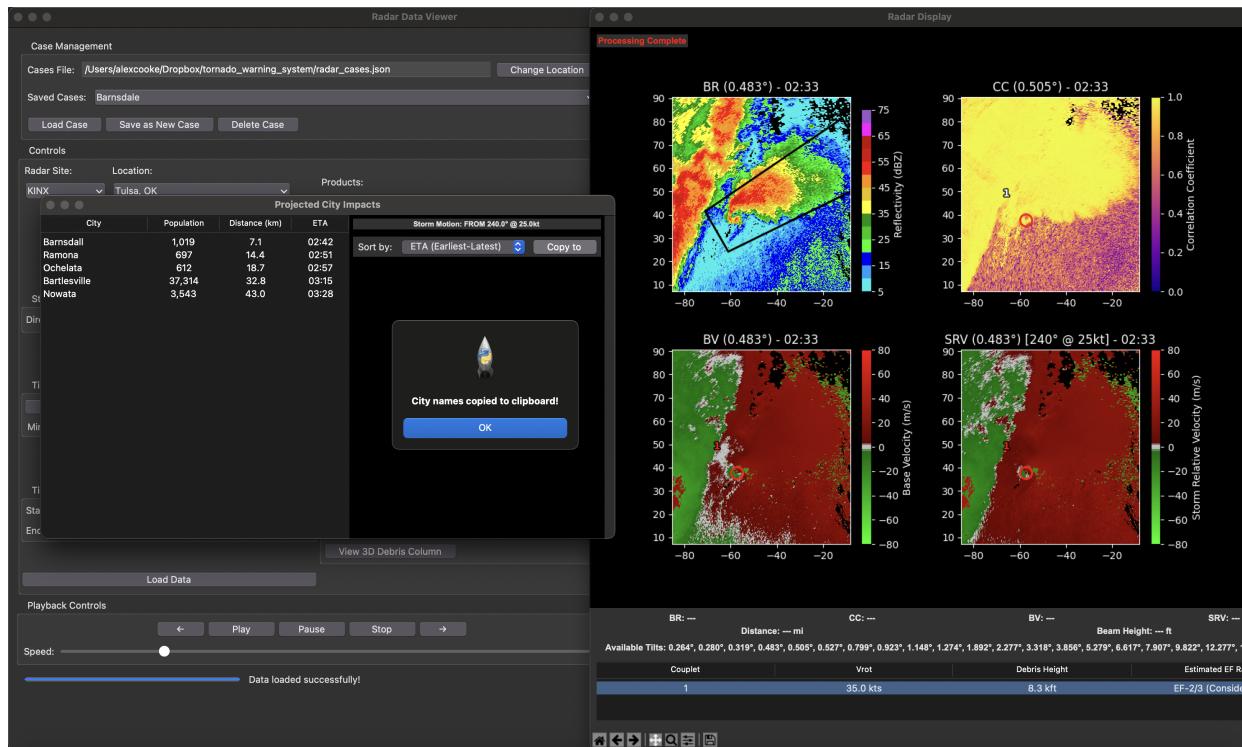


Figure 5.13: Full program interface with population window

CHAPTER VI

OBSERVED LIMITATIONS

6.1 Range Folding

Range folding occurs when radar signals return from distances beyond the maximum unambiguous range, creating misleading velocity data that can trigger false couplet detections. The program attempts to handle this by applying range limits and filtering, but range-folded velocities often appear as alternating bands of inbound and outbound motion that can be misinterpreted as rotation or that can reduce the apparent robustness of a couplet. These artifacts pose significant challenges when trying to identify genuine tornado signatures, especially in complex environments where actual rotation may be occurring simultaneously with range folding effects.

6.2 Debris Contamination

Debris contamination becomes particularly problematic in cyclic supercell environments where multiple tornado lifecycles occur in close proximity. As debris from an earlier tornado falls out or is carried downstream, it can create low correlation coefficient (CC) signatures that are incorrectly associated with new developing rotations. This debris fallout can linger for extended periods and drift considerable distances, contaminating CC analysis for subsequent tornadoes. The algorithm may struggle to differentiate between fresh lofting associated with an active tornado and lingering debris from previous tornadoes, potentially leading to overestimation of debris heights or misattribution of debris characteristics to the wrong circulation.

6.3 Proximity to Radar

When tornadoes form very close to the radar site, the low-level velocity data becomes unreliable due to the shallow beam angle. Issues like ground clutter are more pronounced near the radar, adding sometimes significant noise to the data, and maximum height of the beam is limited, reducing the program's ability to estimate the debris column height of stronger tornadoes.

6.4 Exceedingly Wide Signals

Exceedingly wide signals challenge the algorithm's ability to isolate discrete circulation features. When rotation covers a broad area, such as in a mesocyclone or large tornado, the detection thresholds designed for more compact signatures may fail to capture the full extent of the circulation. The program's method of searching for velocity couplets within specific distance thresholds works best with tight, well-defined rotation patterns, but struggles with diffuse or complex rotational structures. In particular, at extreme distances, radar bins may be so large that cluster size needs to be reduced, which, in turn, increases the false alarm rate.

6.5 CC Melting Layer

The CC melting layer false positive issue stems from the natural reduction in correlation coefficient where precipitation transitions from ice to liquid, creating signatures that can resemble debris to the algorithm. This melting layer typically appears as a ring of lower CC values around the radar at the altitude where atmospheric temperature crosses the freezing point. The algorithm may incorrectly interpret these natural meteorological features as potential tornado debris, especially when they intersect with velocity signatures.

6.6 Low-Significant Far Tornadoes

Though the algorithm is specifically designed for EF2-5 tornadoes, low-end significant tornadoes typically have debris columns around 8,000 to 10,000 feet. As beam height increases with range, it can potentially overshoot these shallow debris columns altogether.

6.7 Overfiltering

Because different radar products are often collected with slightly different parameters, they must be spatially aligned before processing. The alignment involves interpolation and resampling to match a common data grid (velocity, in this case). During interpolation, sharp gradients or small-scale features present in the original data might be smoothed out or diluted. Small but significant rotation signatures or debris patterns could be weakened or lost entirely once resampled to align with other products.

If one product has missing or poor-quality data in certain areas, the alignment process might propagate these gaps across all products after filtering criteria are applied. For example, if correlation coefficient data is noisy in a particular region, aligning velocity data and then applying CC-based filtering could unnecessarily remove valid velocity data.

The aligned products might have different native resolutions, causing finer-detail products to lose information when matched to coarser products. This effectively applies an additional smoothing filter beyond what's explicitly defined in the filtering parameters. When multiple filtering criteria from different aligned products are combined (like requiring specific reflectivity thresholds AND correlation coefficient values AND velocity patterns), the intersection of these constraints can create an overly restrictive filter that eliminates borderline but potentially significant meteorological signals.

This cascading effect of alignment and multi-product filtering can be particularly problematic for detecting weak or distant tornadoes, where signatures are already subtle and further data reduction through alignment processing might push them below detection thresholds.

This is likely the most significant limitation of the algorithm, as even robust signatures often have a majority of their data filtered out. This rarely entirely eliminates a couplet, but it can filter out maximal velocity values, reducing recorded rotational velocity and forcing a reduction in detection threshold, causing an increase in false alarms.

6.8 Low CC Inflow Area

Low correlation coefficient values in inflow areas create confusion because strong inflow regions of supercells often display reduced CC values without necessarily indicating debris. These low CC signatures in inflow regions result from a diversity of particle types, sizes, and orientations being drawn into the storm, rather than debris lofting. The algorithm may misinterpret these naturally occurring low CC regions as potential debris signatures when they are near velocity couplets in the inflow region, leading to false tornado identifications in areas where tornadoes haven't yet formed. In fact, forecasters are specifically trained to be aware of this pitfall. In particularly narrow storms or storms where the mesocyclone is near the edge, the program's detection radius can overlap into the inflow region and mistake it for debris.

6.9 Delay Between Damage and Signature Appearance

It often takes 5-10 minutes between a tornado causing damage and an appreciable radar signature appearing. If a tornado touches down close to major population center, the algorithm may not have enough time to identify the tornado debris signature. For example, the 2011 Joplin

EF5 touched down a 5:34 pm local time and entered the outskirts of the city by 5:41 pm. In such cases, the program will likely not have sufficient data nor provide sufficient lead time.

6.10 Discussion of Limitations

Many of the limitations described above represent fundamental challenges inherent to radar-based tornado detection that operational meteorologists already contend with routinely. Range folding, debris contamination in cyclic supercells, proximity issues near radar sites, CC melting layer artifacts, low CC values in inflow areas, and the delay between tornado formation and TDS development are all well-documented challenges in operational meteorology. NWS forecasters receive extensive training to recognize and account for these limitations when making warning decisions.

What distinguishes the algorithm's experience of these limitations from human forecasters is primarily the lack of contextual understanding and adaptability. While experienced meteorologists can draw on pattern recognition skills, environmental awareness, and the integration of multiple data sources to overcome many of these challenges, the algorithm is constrained by its predefined parameters and processing methods. For example, meteorologists might recognize range folding or melting layer signatures through their characteristic patterns and spatial positioning, whereas the algorithm must rely solely on threshold-based filtering.

The overfiltering issue represents a limitation more unique to the automated nature of the program. Human forecasters viewing radar data in AWIPS or similar systems are seeing minimally processed data products and can mentally integrate information across multiple fields without the data loss that occurs during the algorithm's alignment and filtering processes. Similarly, the fixed

detection parameters for rotation width create a limitation that human forecasters overcome through adaptive interpretation based on experience.

Despite these limitations, the program demonstrates considerable skill in detecting significant and violent tornadoes, as shown in the results section. The algorithm's performance metrics suggest that while these limitations do impact overall detection capabilities, they do not prevent the system from providing valuable supplemental guidance for high-impact events. The complete absence of "too low" classifications across all tornado categories indicates that when the program does identify a tornado, it reliably communicates the appropriate level of threat, albeit with some tendency toward overclassification.

Future iterations of the algorithm could address some of these limitations through machine learning approaches that better mimic the pattern recognition capabilities of experienced forecasters or through the integration of additional data sources that provide context currently missing from the radar-only analysis.

CHAPTER VII

RESULTS

7.1 Overall Results

Metric	Value
POD	0.691
FAR	0.0946
CSI	0.644
Hits	67
Misses	30
False Alarms	7

Table 7.1: Overall Verification Metrics (Warning Times)

The comprehensive verification metrics presented in Table 7.1 show overall performance. With a Probability of Detection (POD) of 0.69, the system successfully identified over two-thirds of actual tornadoes, while maintaining an impressively low False Alarm Ratio (FAR) of 0.095. This balance yielded a Critical Success Index (CSI) of 0.644, indicating good overall skill. In absolute terms, the algorithm correctly detected 67 tornadoes, missed 30 events, and generated only 7 false alarms across the evaluation period. For the purposes of this analysis, a hit is considered a detection that is at or above the final EF rating, representing sufficiently serious categorization of the threat.

A miss is considered either a failure to detect or a detection level below the final EF rating. A false alarm is a detection of a nonexistent tornado.

Metric	Value
Correct	0.701
Too High	0.299
Too Low	0

Table 7.2: Detection Level Proportions

The detection level analysis in Tables 7.2 highlights a clear intensity classification bias in the algorithm. Approximately 70% of tornado detections correctly classified intensity levels, while about 30% overestimated tornado strength. Notably, there were no instances of underestimation (0% "Too Low"), revealing a systematic bias toward overclassification. This pattern suggests refinement opportunities to reduce overclassification while maintaining the desirable property of never underestimating tornado intensity.

Metric	Value
Mean Lead Time	3.939
Standard Deviation	15.395

Table 7.3: Overall Lead Time Statistics
(Mean and Standard Deviation)

0%	25%	50%	75%	100%
-39.0	-4.00	2.50	11.25	46.0

Table 7.4: Overall Lead Time Statistics
(Quantiles)

The lead time statistics in Tables 7.3 and 7.4 provide crucial insight into the algorithm's practical utility for public safety. The system delivered an average warning time of approximately 3.9 minutes earlier than NWS warning issuance, though with substantial variability (standard deviation of 15.4 minutes). The median lead time of 2.5 minutes indicates that half of all detected tornadoes received less than 3 minutes of improvement, while the 25th percentile value of -4 minutes shows that a significant portion of events showed poorer performance by the program than by manual NWS detection.

7.2 Category-Specific Results

Emergency		PDS		Normal	
Metric	Value	Metric	Value	Metric	Value
POD	1.0	POD	0.875	POD	0.415
FAR	0.636	FAR	0.143	FAR	0.292
CSI	0.364	CSI	0.764	CSI	0.354
Hits	8	Hits	42	Hits	17
Misses	0	Misses	6	Misses	24
False Alarms	14	False Alarms	7	False Alarms	7

Table 7.5: Category-Specific Verification Metrics (Warning Times)

The category-specific verification metrics in Table 7.5 reveal striking performance differences across tornado categories that provide useful insight into the program's applicability to different scenarios. For Emergency-level tornadoes, the system achieved perfect detection (POD=1.0) but at the cost of numerous false alarms (FAR=0.636), resulting in a relatively low CSI of 0.364.

PDS (Particularly Dangerous Situation) tornadoes showed the most balanced performance with strong detection rates ($POD=0.875$) and limited false alarms ($FAR=0.143$), yielding the highest CSI of 0.764. Normal-level tornadoes proved most challenging to detect, with a POD of only 0.415, though false alarms remained moderate ($FAR=0.292$). This pattern likely reflects the radar signature characteristics of each category, with more intense tornadoes generating clearer, more distinctive patterns that algorithms can more confidently identify.

Emergency		PDS		Normal	
Metric	Value	Metric	Value	Metric	Value
Correct	1.0	Correct	0.738	Correct	0.471
Too High	0.0	Too High	0.262	Too High	0.529
Too Low	0.0	Too Low	0.0	Too Low	0.0

Table 7.6: TDS Height Detection Level Analysis by Category

Table 7.6 extends the intensity classification analysis by category, showing that the algorithm correctly classified 100% of detected Emergency tornadoes, approximately 74% of detected PDS tornadoes, and about 47% of detected Normal tornadoes. The systematic overclassification bias increases as intensity decreases, showing the algorithm's tendency to perform better with more intense tornadoes. Interestingly (especially as compared to NWS warnings), the program has a complete lack of underestimation.

Metric	Value
Mean V_{rot}	42.963
Standard Deviation V_{rot}	16.958

Table 7.7: Emergency Stats for V_{rot} (Mean and Standard Deviation)

0%	25%	50%	75%	100%
15.60	32.225	50.050	53.950	60.30

Table 7.8: Emergency Stats for V_{rot} (Quantiles)

Tables 7.7 and 7.8 detail the rotational velocity (v_{rot}) statistics for Emergency category tornadoes. The mean v_{rot} of 42.96 knots is above the NWS threshold for a normal warning but below the threshold for a catastrophic (emergency) tag (70 kts). This is not surprising given the overfiltering limitation (section 6.7), and the author expects mean v_{rot} would increase were this limitation solved and increased velocity data made available to the algorithm. Quantile stats paint a slightly better picture, with the median value at approximately 50 knots, a robust though not ideal value.

Metric	Value
Mean Height	19737.5
Standard Deviation	3288.047

Table 7.9: Height Statistics (Mean and Standard Deviation) for Tornado Emergencies

0%	25%	50%	75%	100%
15400	18175	19200	20825	26500

Table 7.10: Height Statistics (Quantiles) for Tornado Emergencies

Tables 7.9 and 7.10 paint a better picture. With a mean height of 19,737.5 feet and a standard deviation of 3,288 feet, these events generate substantial debris plumes. The minimum height observed was 15,400 feet, with the median at 19,200 feet and the maximum reaching 26,500 feet. This means that all detected violent tornadoes displayed debris column heights above the

established threshold of 15,000 ft, underscoring the reliability of the method for this subset of tornadoes.

Metric	Value
Mean Lead Time	12.143
Standard Deviation	19.30

Table 7.11: Lead Time Statistics (Mean and Standard Deviation) for Tornado Emergencies

0%	25%	50%	75%	100%
-9.0	0.0	3.0	22.5	46.0

Table 7.12: Lead Time Statistics (Quantiles) for Tornado Emergencies

Tables 7.11 and 7.12 examine lead time advantage specifically for Emergency tornadoes as compared to NWS warning issue times (positive values indicating increased warning over NWS warnings). The mean lead time advantage of 12.1 minutes with a standard deviation of 19.3 minutes indicates better advance warning for these severe events compared to the overall average. This mean is skewed by some major outliers, with the quantiles showing a more measured evaluation. Nonetheless, 75% of emergency cases equaled or improved upon NWS performance, simultaneously showing the program's potential and that it is a supplement, not a replacement.

Metric	Value
Mean V_{rot}	34.929
Standard Deviation V_{rot}	13.682

Table 7.13: PDS Stats for V_{rot} (Mean and Standard Deviation)

0%	25%	50%	75%	100%
9.2	26.7	34.0	42.7	67.1

Table 7.14: PDS Stats for V_{rot} (Quantiles)

Tables 7.13 and 7.14 detail the rotational velocity statistics for PDS (Particularly Dangerous Situation) tornadoes. The mean v_{rot} of 34.93 knots with a standard deviation of 13.68 m/s shows these are significant circulation signatures, though less intense than Emergency cases. Again, the mean and median are below the NWS thresholds for a considerable (PDS) tag (40 kts with TDS and STP>1 or 50 kts without TDS and STP>1). The first quartile also underscores limitation 6.7 and the need for better velocity data.

Metric	Value
Mean Height	11813.79
Standard Deviation	2685.773

Table 7.15: PDS Stats for Height (Mean and Standard Deviation)

0%	25%	50%	75%	100%
8100	9300	12100	13900	18400

Table 7.16: PDS Stats for Height (Quantiles)

Similar to Emergency cases, all values exceed the established 8,000 ft threshold, with most remaining below the 15,000 ft ceiling that marks the transition to Emergency status. The existence of approximately 20 % of tornadoes above the 15,000 ft threshold could be explained by either imperfect algorithm performance or variability in EF ratings as compared to true tornadic potential.

For example, a tornado may have the winds and updraft strength to loft debris to an EF4 level, but it may only hit structures that allow it to achieve an EF3 rating.

Metric	Value
Mean Lead Time	4.095
Standard Deviation	15.210

Table 7.17: PDS Lead Time Statistics (Mean and Standard Deviation)

0%	25%	50%	75%	100%
-29.0	-4.0	3.5	9.0	40.0

Table 7.18: PDS Lead Time Statistics (Quantiles)

Tables 7.17 and 7.18 show a notable but more modest lead time advantage for PDS warnings, averaging about a 4-minute increase, but showing a greater proportion of detections that perform worse than NWS warnings, again underscoring a supplemental role.

Metric	Value
Mean V_{rot}	25.9
Standard Deviation V_{rot}	17.012

Table 7.19: Normal Stats for V_{rot} (Mean and Standard Deviation)

0%	25%	50%	75%	100%
15.60	16.275	18.750	25.425	65.10

Table 7.20: Normal Stats for V_{rot} (Quantiles)

Tables 7.19 and 7.20 present rotational velocity statistics for Normal category tornadoes. The mean v_{rot} of 25.9 kts indicates generally weaker circulation compared to the more severe categories, as expected. The quantiles show a minimum of 15.6 knots, a median of only 18.75 knots, but some higher-end cases reaching 65.1 knots. Due to limitation 6.7, the need for lower thresholds is expected. The upper quartile extending well into the considerable range and even near the violent

range can be indicative of tornadoes that are rather intense but do not hit anything that allow them to achieve higher ratings. With a larger data set and more refined algorithm, this could possibly be used to estimate the proportion of tornadoes capable of higher levels of damage than they cause.

Metric	Value
Mean Height	5400
Standard Deviation	4466.382

Table 7.21: Normal Stats for Height (Mean and Standard Deviation)

0%	25%	50%	75%	100%
1000	1525	4500	7950	13700

Table 7.22: Normal Stats for Height (Quantiles)

Tables 7.21 and 7.22 detail the debris height patterns for Normal tornadoes. With a mean height of just 5,400 feet and a large standard deviation of 4,466.38 feet, these events show substantially lower debris elevation and much greater variability than the more severe categories. Heights ranged dramatically from 1,000 feet to 13,700 feet, with a median of 4,500 feet. Nonetheless, approximately 75% of tornadoes were under the established 8000-feet threshold.

Metric	Value
Mean Lead Time	0.176
Standard Deviation	13.630

Table 7.23: Lead Time Statistics (Mean and Standard Deviation)

0%	25%	50%	75%	100%
-39	-5	2	6	19

Table 7.24: Lead Time Statistics (Quantiles)

Tables 7.23 and 7.24 examine lead time advantage for Normal tornadoes. The mean lead time advantage of just 0.18 minutes in tandem with the median of 2 minutes and normal POD supports

the idea that the program should be limited to use for significant tornadoes, where it achieves both far better POD and lead time advantage values.

7.3 NWS Verification Metrics

Metric	Value
POD	0.938
FAR	0.261
CSI	0.815
Hits	97
Misses	6
False Alarms	16

Table 7.25: Overall Verification Metrics (NWS Scores - Issue Times)

Table 7.25 presents performance metrics using National Weather Service (NWS) verification standards based on issue times rather than warning times. Like the program's overall statistics, a hit is considered a detection that is at or above the final EF rating, representing sufficiently serious categorization of the threat. A miss is considered either a failure to detect or a detection level below the final EF rating. A false alarm is a detection of a nonexistent tornado.

NWS achieved an impressive POD of 0.938, though with a higher FAR of 0.261, resulting in a strong CSI of 0.815. These figures represent 97 correctly detected events, only 6 misses, but 16 false alarms. The higher detection rate compared to Table 7.1 suggests that the algorithm performs better when evaluated on issue times rather than specific warning timing, indicating potential value in operational forecast environments.

Emergency (NWS)		PDS (NWS)		Normal (NWS)	
Metric	Value	Metric	Value	Metric	Value
POD	0.625	POD	0.854	POD	1.0
FAR	0.737	FAR	0.268	FAR	0.349
CSI	0.227	CSI	0.651	CSI	0.651
Hits	5	Hits	41	Hits	41
Misses	3	Misses	7	Misses	0
False Alarms	14	False Alarms	15	False Alarms	22

Table 7.26: Category-Specific NWS Scores (Issue Times)

Table 7.26 breaks down NWS performance metrics by tornado category. The Emergency category showed moderate detection capability (POD 0.625) but high false alarms (FAR 0.737), resulting in a low CSI of 0.227. The PDS category maintained strong detection (POD 0.854) with reasonable false alarm rates (FAR 0.268), yielding a good CSI of 0.651. Interestingly, the Normal category achieved perfect detection (POD 1.0) but with higher false alarms (FAR 0.349), also resulting in a CSI of 0.651.

As compared to Table 7.5, we see the program has better POD, FAR, and CSI for Emergency tornadoes and better FAR and thus better CSI for PDS tornadoes, with far worse performance for Normal tornado warnings. This, in tandem with the improved lead time for significant tornadoes, is indicative of the tool's usefulness as a supplemental program for detection of higher-intensity tornadoes, as well as its efficiency benefits.

7.4 Correlation Plots

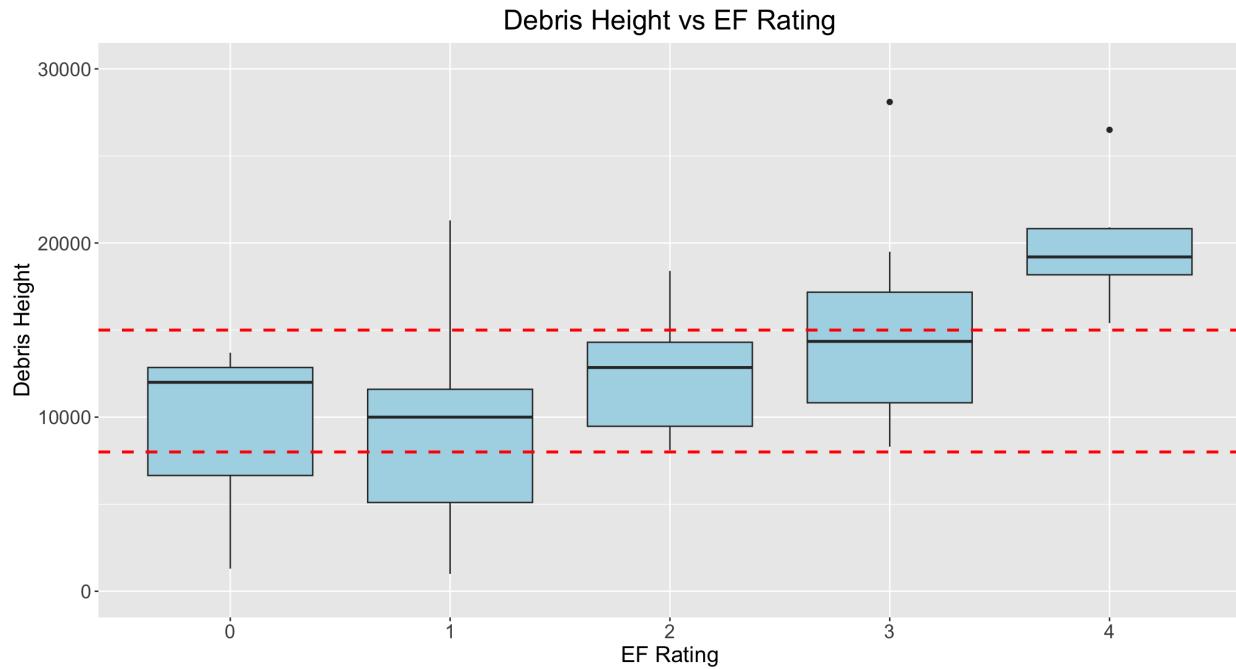
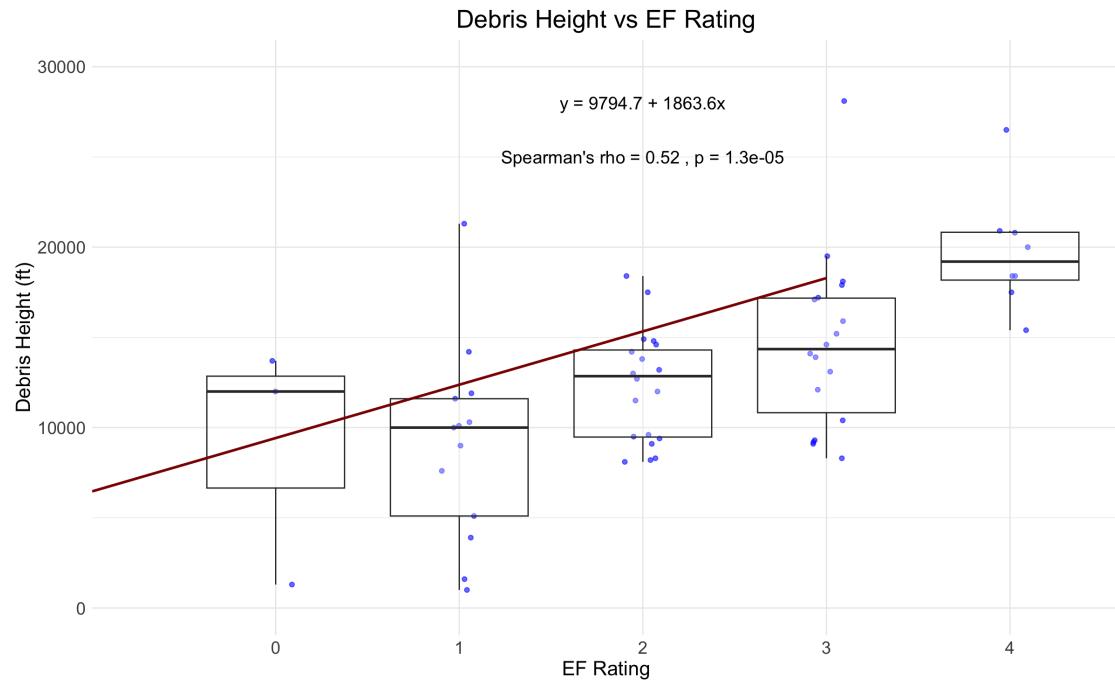
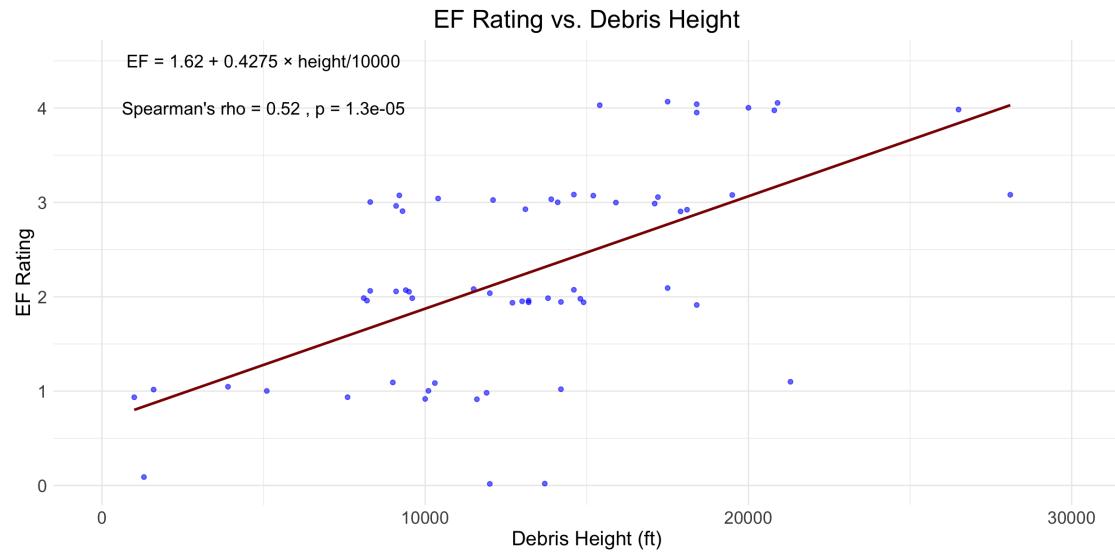


Figure 7.1: Histogram of EF Rating vs Debris Height

Figure 7.1 shows histograms of EF rating versus debris height (with two outliers not shown for better visual quality), with the dashed red lines representing the 8,000 and 15,000 ft thresholds. As can be seen, the program struggles with Normal (weak) tornadoes, while PDS (EF2-3) tornadoes generally lie within the desired 8,000-15,000 ft range. Violent (EF4-5) tornadoes lie entirely within the desired >15,000 ft range (note: no EF5 tornadoes were in the sample, as the most recent EF5 tornado occurred in 2013, before the full deployment of dual-polarization radar).



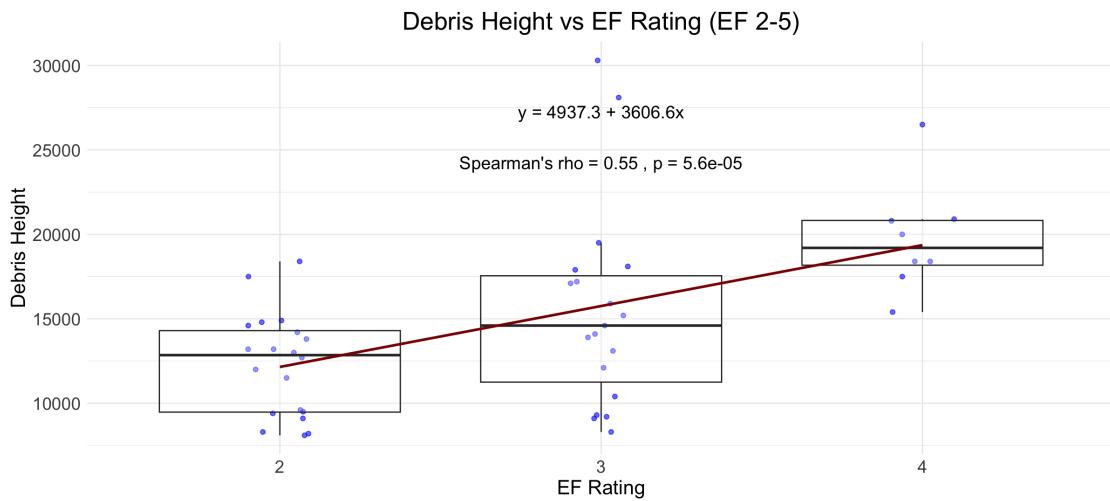
(a) Linear correlation of EF rating and debris height (jitter added for ease of visualization)



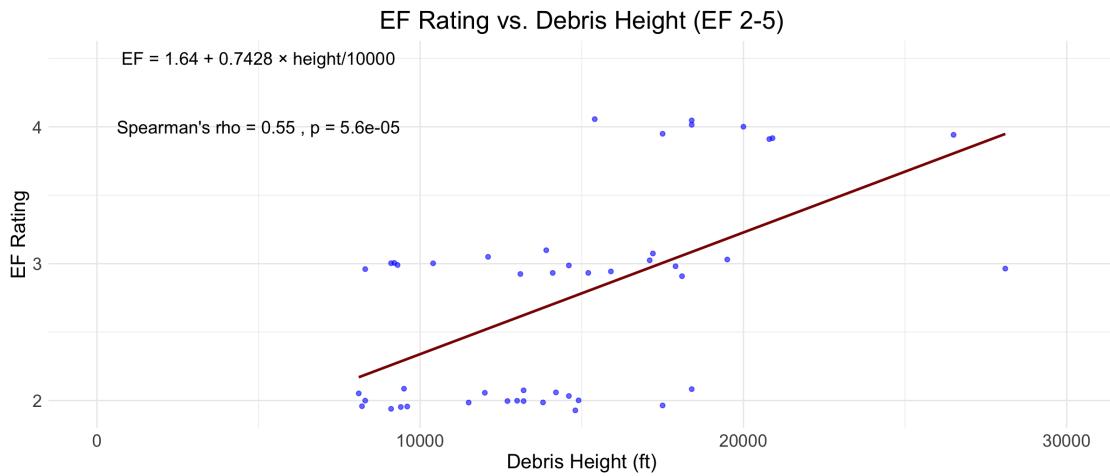
(b) Linear correlation of EF rating and debris height (jitter added for ease of visualization)

Figure 7.2: Correlation between EF rating and debris height: (a) Full dataset, (b) EF 2-5 subset

Figures 7.2(a) and (b) shows a linear correlation of EF rating and debris height, yielding a Spearman rank correlation coefficient of 0.52. Nonetheless, as can be seen, the poor performance of the program on weak tornadoes results in poor correlation for significant tornadoes. This, in turn, reduces the ability to predict EF rating based on debris height should one wish to opt for a continuous predictor rather than categorical threshold, which may be desirable, particularly when dealing with borderline cases.



(a) Linear correlation of EF rating and debris height (jitter added for ease of visualization)



(b) Linear correlation of EF rating and debris height (jitter added for ease of visualization)

Figure 7.3: Correlation between EF rating and debris height for only significant tornadoes

Figures 7.3 (a) and (b) show that restricting the correlation to significant tornadoes improves the fit for those cases and provides a more useful estimator equation that softens the abrupt categorical shift created by using discrete ratings.

7.5 Bootstrap Analysis

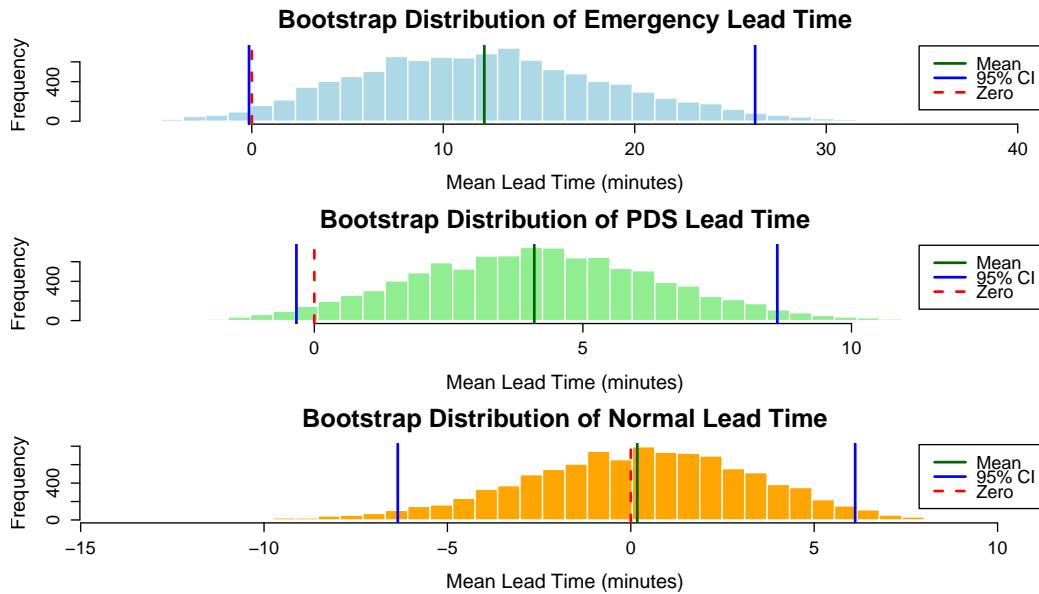


Figure 7.4: Bootstrap distributions for 10,000 samples

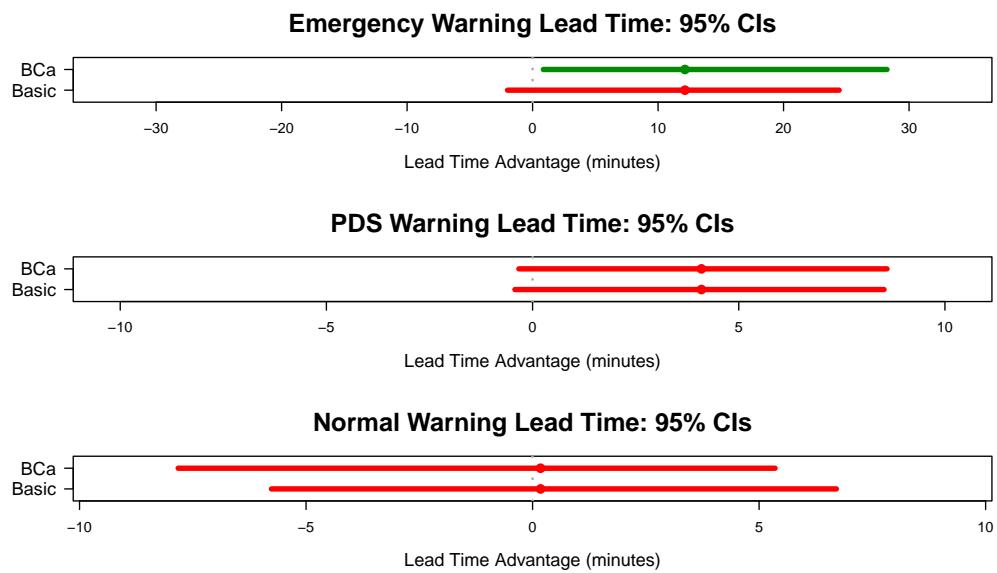


Figure 7.5: Associated bootstrap confidence intervals

The bootstrap confidence intervals reveal important insights about the performance of the tornado warning detection program compared to the NWS system. Most notably, the Emergency category (EF 4-5 tornadoes) demonstrates a promising mean lead time advantage of approximately 12 minutes. While the basic bootstrap interval for this category includes zero (showing non-significance), the more advanced BCa interval excludes zero and is entirely positive (ranging from approximately 0.9 to 28 minutes). This suggests a statistically significant improvement in detecting the most severe tornadoes. In contrast, both PDS (EF 2-3) and Normal (EF 0-1) categories show non-significant results, with both interval types containing zero, indicating no reliable detection advantage over NWS for these less severe tornado categories.

The BCa (bias-corrected and accelerated) bootstrap method used here is particularly important given the characteristics of this dataset. Unlike basic bootstrap intervals that make strong normality assumptions, BCa intervals correct for both bias and changing variance across the sampling distribution. This correction is especially valuable for the Emergency category, which has a small sample size ($n=8$) and a possibly skewed distribution. BCa bootstraps intervals are particularly valuable for small samples. This methodological choice provides greater confidence in the finding that the program offers a real advantage for the most severe tornadoes.

The discrepancy between interval methods for the Emergency category is particularly informative. The Basic interval, ranging from approximately -2 to 24 minutes, suggests uncertainty about whether the program consistently outperforms NWS. However, the BCa interval's shift rightward (0.9 to 28 minutes) indicates that after accounting for distribution characteristics, we can be confident that the true advantage is positive ($p=.05$). This pattern doesn't appear in the PDS and Normal categories, where both interval types show similar ranges containing zero, reinforcing that any

lead time advantages for less severe tornadoes are likely minimal or inconsistent. Nonetheless, the fact that zero is near the edge of both PDS CIs indicates that with some refinement, a level of improvement that creates a statistically significant advantage may be within reach.

These findings have important practical implications. The program appears most valuable for the scenario where it matters most—providing earlier warnings for the most destructive EF 4-5 tornadoes. This could translate to crucial additional minutes for people to seek shelter during the most dangerous events. The lack of significant improvement for less severe tornadoes suggests that resources might best be focused on optimizing detection of high-end events, where the program demonstrates its clearest advantage.

CHAPTER VIII

DISCUSSION AND FUTURE DIRECTIONS

8.1 Summary and Interpretation of Key Findings

This research developed and validated a Python-based operational program designed to enhance tornado warning capabilities by integrating tornado debris signature (TDS) height analysis with population impact assessment. The performance metrics revealed several significant patterns that provide insight into the program's operational utility and limitations.

8.1.1 Intensity Classification Performance

The most striking finding is the program's categorical performance gradient across tornado intensity levels. For violent (EF4+) tornadoes, the algorithm achieved perfect detection (POD=1.0), albeit with a high FAR, nonetheless demonstrating exceptional reliability for the most dangerous tornadic events. The strong correlation between maximum TDS height and EF rating for these high-end tornadoes confirms the robustness of debris height as a proxy for intensity in violent tornadoes.

For significant (EF2-3) tornadoes, the program maintained strong detection capability (POD=0.875) with minimal false alarms (FAR=0.143), yielding the highest overall critical success index (CSI=0.764) of any category. This balanced performance suggests the program provides optimal utility for this middle category, which represents tornadoes that pose serious threats but may not always trigger the highest warning levels.

Weak (EF0-1) tornadoes proved most challenging, with lower detection rates (POD=0.415) and higher misclassification percentages. This pattern aligns with the physical limitations of debris detection for weaker tornadoes, which typically loft less debris to lower heights, creating less distinctive TDS patterns. The significantly increased variability in debris height measurements for this category (standard deviation of 4,466 ft compared to 3,288 ft for violent tornadoes) further supports this interpretation.

8.1.2 Lead Time Advantages

The lead time analysis revealed a compelling pattern: the program provided the greatest warning time advantages precisely for the most dangerous tornado categories. With a mean lead time advantage of 12.1 minutes for violent tornadoes, the algorithm demonstrated substantial potential to extend warning periods for the events where additional time is most critical for public safety. This finding has particular significance for emergency management and public safety applications.

The positive correlation between tornado intensity and lead time advantage likely reflects the more distinctive radar signatures generated by violent tornadoes, allowing the algorithm to identify them earlier in their lifecycle. This interpretation is supported by the distribution of TDS heights, with all violent tornadoes in the sample exhibiting debris columns above 15,000 ft, the established threshold.

8.1.3 Warning Classification Performance

The program's systematic bias toward overclassification can likely be improved with further parameter and algorithm refinement. The complete absence of underestimation (0% "Too Low"

across all categories) ensures the program never minimizes the threat level, though this comes at the cost of some unnecessary escalation, particularly for weak tornadoes.

When compared to NWS performance metrics, the program demonstrated superior POD, FAR, and CSI for Emergency tornadoes and better FAR and CSI for PDS tornadoes, suggesting its particular utility as a supplemental program for high-impact events. These statistics can be customized to a specific user's needs to a degree through the algorithm detection controls in the user interface. The comparative underperformance for weak tornadoes further reinforces the conclusion that the program's optimal application is for significant and violent tornado events.

8.2 Operational Implications and Applications

8.2.1 Integration with Current Warning Systems

The performance characteristics of this program suggest it could serve as a valuable supplemental system within the existing warning framework rather than as a standalone replacement. Its strength in detecting and correctly classifying significant and violent tornadoes, combined with its lead time advantages for these categories, makes it particularly suitable for augmenting and validating forecaster judgments about enhanced warning products like Tornado Emergencies and PDS warnings.

The program's population analysis component adds further operational value by rapidly generating lists of threatened communities and their estimated times of arrival (ETAs). This capability directly addresses the operational challenge of efficiently compiling impact information during high-stress warning scenarios, potentially reducing cognitive load and allowing forecasters to focus on other critical aspects of the warning process.

8.2.2 Forecaster Decision Support

Rather than automating warning decisions, this program is best positioned as a decision support system that provides objective guidance while preserving forecaster judgment. The algorithm's recommendations could serve as one factor in the warning decision process, particularly valuable in ambiguous situations or when multiple threatening storms demand attention simultaneously.

For Tornado Emergency scenarios, where the criteria include both meteorological factors (confirmed violent tornado) and impact considerations (threat to populated areas), the program's integrated approach to assessing both dimensions provides particular utility. By objectively evaluating debris signature height alongside population density, the algorithm helps standardize this otherwise subjective evaluation process.

8.2.3 Standardization and Consistency Benefits

The automation of population impact analysis also addresses a known challenge in warning operations: the application of population considerations in warning decisions. By standardizing this process and providing explicit user-determined population thresholds, the program supports more equitable warning services across geographic regions.

8.3 Current Limitations and Challenges

8.3.1 Technological and Meteorological Constraints

Several fundamental limitations constrain the program's effectiveness in certain scenarios. The radar distance issue remains problematic, as beam height increases with range from the radar, potentially causing the system to miss relatively shallow debris columns from significant tornadoes at greater distances. While this limitation affects all radar-based tornado detection approaches, it poses a particular challenge for debris height-based classification.

Temporal limitations also significantly impact performance. The 5-10 minute lag between tornado formation and TDS development means the program cannot provide initial detection advantages. This constraint is particularly problematic for short-lived tornadoes and those that form in or near population centers, where advance warning is most critical.

The observed overfiltering problem represents another significant technical limitation. The alignment and interpolation of radar products with different native resolutions can result in the loss or dilution of important signals, potentially reducing measured rotational velocities and affecting classification accuracy. This issue likely contributes to the lower-than-expected mean v_{rot} values observed across all categories.

8.3.2 Classification and Detection Challenges

Several pattern recognition challenges affect the program's performance. The confusion between low correlation coefficient signatures in inflow regions and actual debris signatures can trigger false positives, particularly in narrow storms or those where the mesocyclone is positioned near the edge of the storm. This phenomenon likely contributes to the relatively high false alarm rates observed for Emergency warnings.

For QLCS tornadoes, the rapid development and dissipation cycles often occur faster than the radar scanning interval, making these events particularly challenging to detect and classify accurately. This limitation is inherent to the radar technology rather than the algorithm itself but nonetheless affects operational utility. Future work could investigate how storm mode affects the program's performance. Depending on radar and convection location, scanning mode, and temporal staggering, it might be possible to use multiple radar sites to reduce the effective scan interval.

The CC melting layer issue introduces another classification challenge, as the natural reduction in correlation coefficient at the freezing level can mimic debris signatures when intersecting with velocity features. While the current implementation attempts to address this through spatial constraints, it remains a source of potential false positives.

Lastly, the program is prone to spurious signals in several situations. Although they are normally obviously illegitimate to a trained meteorologist, they can still be visually distracting. Fixing the overfiltering problem will allow the minimum v_{rot} threshold to be raised, which will help to eliminate many of these. Other approaches like cluster shape analysis and dynamic distance-based thresholds could help to further reduce these signals.

8.3.3 Verification and Validation Limitations

The imperfect nature of EF scale ratings as ground truth presents a fundamental verification challenge. Since damage-based EF ratings are contingent on the presence of appropriate damage indicators, tornadoes in sparsely populated areas may receive lower ratings than their true intensity would warrant. This limitation affects both the training and validation process, potentially biasing the correlation between debris height and "true" tornado intensity.

The relatively small sample size of violent tornadoes (n=8) in the evaluation dataset also limits the confidence in performance statistics for this critical category. While the perfect detection rate is encouraging, broader validation with a larger sample would strengthen confidence in the program's performance for these rare but high-impact events.

Another issue regarding validation was the lack of a true null case. Because cases were chosen from preexisting NWS tornado warnings due both to the relatively spatial and temporal rarity of tornado conditions and the desire to compare performance, all radar environments were of the

nature the program was designed for. To fully evaluate the program's performance, a true null case is needed to establish the program's ability to reject radar returns that are not characteristic of tornadoes, such as normal stratiform precipitation. This would require establishing benign conditions, such as cases in which the STP = 0.

8.3.4 Algorithm Enhancements

Several specific algorithm improvements could address the identified limitations. A more sophisticated filtering approach that preserves velocity extrema while removing noise would help mitigate the overfiltering problem, potentially improving Vrot measurements and classification accuracy. Implementing an adaptive cluster size threshold based on distance from the radar could also enhance detection capability for distant tornadoes.

Incorporating machine learning techniques represents a promising avenue for improvement. A supervised learning approach using historical tornado cases could potentially identify more complex relationships between radar parameters and tornado intensity than the current threshold-based system. This approach could integrate multiple factors beyond debris height, potentially including:

- Temporal evolution of TDS characteristics
- Three-dimensional structure of the debris column
- Spatial relationships between reflectivity, velocity, and polarimetric signatures
- Environmental parameters like STP

A machine learning approach could also help address the classification challenges for weak tornadoes by identifying subtler signature patterns distinct from non-tornadic signatures.

8.3.5 Additional Data Integration

Integrating additional data sources could substantially enhance the program's performance. Real-time STP analysis, as suggested by Smith et al. (2020), would provide valuable environmental context that could improve classification accuracy, particularly for borderline cases.

Incorporating multi-radar integration would address some of the range-dependent limitations by using data from multiple radars where coverage overlaps. This approach would be particularly valuable in regions with dense radar coverage, allowing for more complete three-dimensional analysis of debris columns.

Lightning data integration represents another promising enhancement. Research has shown relationships between lightning flash rate trends and mesocyclone strengthening (Schultz et al., 2017). Incorporating real-time lightning mapping array data or GLM data from GOES satellites could provide additional lead time before debris signatures appear. Other tornado precursors, such as the descending reflectivity core, also show promise.

Allowing the user to select the storm mode would be a useful improvement, as operational meteorologists will often slightly adjust parameter thresholds for QLCS scenarios versus supercellular environments. This could allow for automatic shifts of the algorithm thresholds, or in the case of more advanced methods being employed (such as machine learning), switch to entirely new detection paradigms.

8.3.6 Population Analysis Improvements

There are several ways the population analysis could be improved. First, in rural areas, where population is more spread out, there are fewer concentrated population centers, which, in turn, decreases the number of named places output by the program. This may decrease the ability for the

program to communicate areas and populations in danger from a tornado. Adding in an additional function, such as county analysis and output, could help to bridge this gap.

Another improvement would be handling major metropolitan areas and their associated suburbs. If a tornado nears such an area, the population analysis may name several or even dozens of suburbs associated with the area, which may be too much text for a warning, in which efficiency of communication is key. A potential solution is either manually designating metropolitan areas and their associated suburbs or setting a certain lower threshold of population as a major metropolitan area and designating all population centers within a certain radius as suburbs. The text output could then be modified to display something like “[metropolitan area] and all suburbs” instead of listing each individual population center.

8.3.7 Operational Implementation Improvements

Several enhancements could improve the program’s operational utility. Developing an AWIPS-compatible version would facilitate seamless integration with existing NWS workflows. Integration with other popular radar programs, such as GR2Analyst, could also be beneficial. These implementations could include configurable thresholds to accommodate regional variations in tornado characteristics and population density.

Implementing a probabilistic output framework would better align with the NWS’s movement toward explicitly communicating forecast uncertainty. Rather than discrete category assignments, the system could provide probability distributions for different intensity levels, allowing forecasters to make more nuanced warning decisions. The EF rating-debris height regression is a first step in this direction.

8.3.8 Testing and Validation

Expanded testing across diverse geographical regions would help assess the program's generalizability. Tornado characteristics vary significantly between regions (e.g., Southeast vs. Plains), and validation across these different environments would ensure robust performance nationwide.

Operational testing in simulated and real warning environments would provide crucial insights into human-machine interaction aspects. Understanding how forecasters interpret and apply the program's guidance in high-stress scenarios would inform interface improvements and training materials.

Gathering data on couplet distance from radar and analyzing performance by distance could yield valuable insights on both how to use the program operationally and how to improve it, particularly when it comes to decreasing false positives.

8.4 Broader Implications and Future Vision

8.4.1 Impact on Warning Philosophy

This research contributes to the ongoing evolution of tornado warning philosophy from a binary (warning/no warning) approach toward a more nuanced, impact-based framework. By objectively quantifying both meteorological intensity and population impacts, the program supports the trend toward more specific communication of tornado threats.

8.4.2 Societal Impacts and Benefits

With a mean lead time advantage of 12.1 minutes for violent tornadoes, this program has the potential to help forecasters reduce casualties from the most dangerous events. The population analysis component addresses important equity concerns in warning services. By standardizing the consideration of population impacts, the program helps ensure that all communities receive

appropriate warnings based on objective threat assessment rather than subjective evaluations or resource constraints.

8.4.3 Long-term Research Directions

Investigating the relationship between TDS vertical structure and tornado dynamics represents another frontier. The three-dimensional structure of debris columns likely contains additional information about tornado intensity, duration, and evolution that could further improve warning capabilities.

Further investigation of other radar phenomena associated with tornadogenesis and tornado intensity could refine and/or augment the algorithm.

8.5 Conclusion

This research has demonstrated that a Python-based operational program integrating tornado debris signature height with population impact assessment can enhance real-time tornado warning capabilities, particularly for significant and violent tornadoes. The program's strong performance for significant tornadoes and lead time advantages for high-impact events highlight its potential value as a supplemental warning system.

While several limitations constrain its effectiveness, particularly for weak tornadoes and in certain meteorological scenarios, the identified enhancements and future directions provide a clear path toward improved performance. Machine learning approaches, additional data sources, and probabilistic frameworks represent particularly promising avenues for advancement.

Beyond its direct operational applications, this research contributes to the broader evolution of tornado warning philosophy and practice. By objectively quantifying both meteorological and

impact factors, the program supports the ongoing transition toward more specific, impact-based warning communications.

Ultimately, this work demonstrates that radar-derived debris characteristics can provide valuable real-time insights into tornado intensity and potential impacts, offering a promising approach for enhancing public safety during severe weather events. With continued refinement and operational integration, programs of this nature have the potential to improve the nation's tornado warning system, reducing casualties from these devastating natural hazards.

REFERENCES

- [1] Amazon Web Services, “Boto3: AWS SDK for Python,” 2024, available online at <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>.
- [2] Amazon Web Services, “Botocore: Low-level, core functionality of boto3,” 2024, available online at <https://github.com/boto/botocore>.
- [3] D. J. Bodine, M. R. Kumjian, R. D. Palmer, P. L. Heinselman, and A. V. Ryzhkov, “Tornado damage estimation using polarimetric radar,” *Weather Forecasting*, vol. 28, no. 1, 2013, pp. 139–158.
- [4] S. Emmerson, S. E. Nelson, and R. L. Thompson, “Using characteristics of tornadic debris signatures to estimate tornado intensity,” *19th Annual Meeting*, Boston, MA, 2020, American Meteorological Society.
- [5] S. W. Emmerson, S. E. Nelson, and A. K. Baker, “A comprehensive analysis of tornado debris signatures associated with significant tornadoes from 2010-2017,” *28th Conf. on Severe Local Storms*, Seattle, WA, 2017, American Meteorological Society, Paper 11.4.
- [6] J. Gibbs, “A skill assessment of techniques for real-time diagnosis and short-term prediction of tornado intensity using the WSR-88D,” *Journal of Operational Meteorology*, vol. 4, no. 13, 2016, pp. 170–181.
- [7] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array availablemung with NumPy,” *Nature*, vol. 585, no. 7825, Sept. 2020, pp. 357–362, available online at <https://doi.org/10.1038/s41586-020-2649-2>.
- [8] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, 2007, pp. 90–95, available online at <https://matplotlib.org/>.
- [9] Iowa State University, “Iowa Environmental Mesonet,” 2024, available online at: <https://mesonet.agron.iastate.edu/>. Accessed: December 15, 2024.
- [10] C. D. Karstens, B. Saba, B. T. Smith, R. L. Thompson, S. A. Erickson, and I. L. Jirak, “A method for adding intensity information to NWS preliminary tornado damage paths,” *103rd Annual Meeting*, Denver, CO, 2023, American Meteorological Society.

- [11] F. Lundh, “An introduction to tkinter,” 1999, available online at www.pythonwarecom/library/tkinter/introduction/index.htm.
- [12] Matplotlib development team, “Matplotlib Toolkits,” 2024, available online at: <https://matplotlib.org/stable/tutorials/toolkits/index.html>.
- [13] R. M. May, S. C. Arms, P. Marsh, E. Bruning, J. R. Leeman, K. Goebbert, J. E. Thielen, Z. S. Bruick, and M. D. Camron, “MetPy: A Python Package for Meteorological Data,”, 2024, available online at <https://unidata.github.io/MetPy/latest/index.html>.
- [14] R. M. Mosier, B. Smith, R. Thompson, and C. Karstens, “A tool for real-time estimation of tornado damage intensity,” *30th Conference on Severe Local Storms*, Santa Fe, NM, 2022, American Meteorological Society.
- [15] National Oceanic and Atmospheric Administration, “Damage Assessment Toolkit,”, 2024, available online at: <https://apps.dat.noaa.gov/stormdamage/damageviewer/>. Accessed: December 15, 2024.
- [16] National Weather Service, *Impact-based tornado warning guidance*, Training document, National Weather Service, 2024, available online at https://training.weather.gov/wtd/courses/rac/warnings/IBWcontent/story-content/external_files/IBW%20Tornado%20Guidance.pdf.
- [17] Pandas development team, “pandas-dev/pandas: Pandas,”, Feb. 2020, available online at <https://doi.org/10.5281/zenodo.3509134>.
- [18] Python Software Foundation, “Python programming Language,”, 2024, available online at <https://www.python.org/>.
- [19] Simple Maps, “United States Cities Database,”, 2024, available online at: <https://simplemaps.com/data/us-cities>. Accessed: December 10, 2024.
- [20] B. T. Smith, R. L. Thompson, C. D. Karstens, J. S. Grams, A. R. Dean, R. M. Mosier, and A. D. Lyons, “Preliminary evaluation of a real-time diagnostic tornado damage intensity estimation tool used at the Storm Prediction Center,” Unpublished manuscript, 2024.
- [21] B. T. Smith, R. L. Thompson, D. A. Speheger, A. R. Dean, C. D. Karstens, and A. K. Anderson-Frey, “WSR-88D tornado intensity estimates. Part II: Real-time applications to tornado warning time scales,” *Weather Forecasting*, vol. 35, no. 6, 2020, pp. 2493–2506.
- [22] R. L. Thompson, B. T. Smith, J. S. Grams, A. R. Dean, J. C. Picca, A. E. Cohen, E. M. Leitman, A. M. Gleason, and P. T. Marsh, “Tornado damage rating probabilities derived from WSR-88D data,” *Weather Forecasting*, vol. 32, no. 4, 2017, pp. 1509–1528.
- [23] Tornado Archive, “Data Explorer,”, 2024, available online at: <https://tornadoarchive.com/home/tornado-archive-data-explorer/>. Accessed: December 15, 2024.

- [24] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, and P. van Mulbregt, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,”, 2020, available online at <https://scipy.org>.