ASCAT Wind Estimation at 2.5 km Resolution Supported by Machine Learning Rain Detection

Joshua Benjamin Kjar

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

David G. Long, Chair
Willie K. Harrison
D.J. Lee

Department of Electrical and Computer Engineering

Brigham Young University

ABSTRACT

ASCAT Wind Estimation at 2.5 km Resolution Supported
by Machine Learning Rain Detection

Joshua Benjamin Kjar
Department of Electrical and Computer Engineering, BYU
Master of Science

The Advanced Scatterometer (ASCAT) is a C-band scatterometer designed to be less sensitive to rain contamination than other higher frequency scatterometers. However, the radar backscatter is still affected by rain which increases error during wind estimation. The error can be reduced in rainy conditions by combining a rain backscatter model with the existing wind only (WO) backscatter model to perform simultaneous wind and rain (SWR) estimation. I derive and test several 2.5 km resolution rain backscatter models for ASCAT data which are used with the WO model to estimate the near surface winds. Various rain models optimal for different purposes are discussed. The best rain model for estimating wind speed lowers the root mean square error (RMSE) in the presence of rain by 13.6% when compared to using the WO model alone. The rain model which best predicts rain rates has a RMSE of 7.9 mm/h.

A neural network (NN) is designed to discriminate the presence of rain using ASCAT's backscatter measurements. Such a NN enables the SWR algorithm to be used only on rainy samples and thus improves estimation. By removing all samples identified by the NN as rain, the WO algorithm's speed estimate improved by 2.83%.

Keywords: ASCAT, radar, wind, rain, geophysical model function, machine learning, neural net

ACKNOWLEDGMENTS

Thank you Dr. Long for several years worth of excellent advice, counseling and encouragement.

To my school friends: Thank you for lifting me up when life got me down. Your example, support, patient explaining and re-explaining of "simple concepts" have created memories I will always cherish. Thank you Nate, Sheridan, James, Adam and the rest of the MERS crew.

To my wife Abby: Thank you for always welcoming me home with a hug after a long day's work.

To my son Calvin: This is for you. I'll be loving you always.

TABLE OF CONTENTS

Title Page . . . . . **i**

Abstract . . . . . **ii**

Acknowledgments . . . . . **iii**

Table of Contents . . . . . **iv**

List of Tables . . . . . **vii**

List of Figures . . . . . **viii**

Chapter 1 Introduction **1**
1.1 Wind Estimation by Scatterometry 1
1.2 Machine Learning 2
1.3 Thesis Layout 3

Chapter 2 Background **4**
2.1 GMFs and the Retrieval Process 4
2.2 Basic Rain GMF Framework 6
2.3 SWR retrieval 7
2.4 Parameterization Restrictions 7

Chapter 3 ASCAT, TRMM and ECMWF Collocated Data Set **9**
3.1 Data Sources 9
3.2 UHR Process 10
3.3 Data Collocation 11
3.4 Backscatter's Relation to Wind Speed and Rain 11

Chapter 4 Rain GMF 1: Constant Surface Rain Rate **15**
4.1 Solving for Rain Induced Attenuation 15
4.2 Fitting the Basic Rain GMF Coefficients 16
4.2.1 Gain 16
4.2.2 Effective Backscatter 18
4.3 Empirical Test 19
4.3.1 Wind Speed Results 20
4.3.2 Rain Results 22
4.4 Stochastic Inferences 22
4.4.1 PDF of H_f 25
4.4.2 PDF of α in terms of PDF of H_f 25
4.5 CSRR Conclusion 28

Chapter 5 Rain GMF 2: Effective Rain Attenuation . . . 31
5.1 Solving for Rain Induced Attenuation . . . 31
5.2 Empirical Test . . . 32
5.2.1 Wind Speed Results . . . 32
5.2.2 Rain Results . . . 32
5.3 Stochastic Inferences . . . 33
5.4 ERA Conclusion . . . 36

Chapter 6 Rain GMF 3: Effective Distance . . . 39
6.1 Solving for Rain Induced Attenuation . . . 39
6.2 Peculiarities of α . . . 40
6.3 Empirical Results . . . 40
6.4 Stochastic Inferences . . . 40
6.5 ED Conclusion . . . 41

Chapter 7 Rain GMF 4: Data-Derived WO Backscatter . . . 43
7.1 Solving for Wind Induced Backscatter . . . 43
7.2 New Fitting Method for the Basic Rain GMF Coefficients . . . 43
7.3 Empirical Results . . . 49
7.3.1 Wind Estimation . . . 49
7.3.2 Rain Estimation . . . 50
7.3.3 Performance Using Dry Samples . . . 51
7.4 Stochastic Inferences . . . 52
7.5 DDWB Conclusion . . . 53

Chapter 8 Machine Learning Background . . . 55
8.1 Machine Learning and Neural Networks . . . 56
8.2 NN Structure . . . 57
8.3 Basic Issues Preventing Successful NNs . . . 59
8.3.1 Overfitting and Underfitting . . . 59
8.3.2 Observing Overfitting and Underfitting . . . 60

Chapter 9 NN-based Rain Detection Applied to QuikSCAT . . . 63
9.1 Preliminary Tests using QuikSCAT Data . . . 63
9.1.1 QuikSCAT . . . 64
9.1.2 Data set . . . 65
9.2 Performance Metrics . . . 65
9.2.1 Performance of the Estimators . . . 65
9.2.2 Performance of the NN . . . 66
9.3 Handling NN Complexity . . . 69
9.3.1 Shape . . . 70
9.3.2 Regularization . . . 71
9.4 Minority Oversampling . . . 72
9.5 Training Parameters . . . 73

9.5.1 Software Platform . . . 74
9.5.2 Training Duration and Learning Rate . . . 74
9.5.3 Input Conditioning . . . 75
9.5.4 Activation Functions . . . 76
9.6 Input and Output Data . . . 77
9.6.1 Data Handling . . . 77
9.6.2 Data Amount . . . 77
9.6.3 Data Type . . . 78

Chapter 10 NN-based Rain Detection Applied to ASCAT . . . 81
10.1 Design Processes . . . 81
10.1.1 Use of Neighboring WVCs and Training Duration . . . 82
10.1.2 Minority Oversampling . . . 83
10.1.3 Adjusting Size . . . 85
10.1.4 Structure of Best Design . . . 85
10.2 Testing Contamination and Overfitting . . . 86
10.2.1 Data Sameness Amongst Neighboring WVCs . . . 87
10.2.2 Handling Near Neighbor Sameness to Control Overfitting . . . 88
10.2.3 Performance After Contamination Correction . . . 90
10.3 Conclusion . . . 91

Chapter 11 Conclusion . . . 92
11.1 Rain GMF Summary . . . 92
11.2 NN Summary . . . 94
11.3 Future Work . . . 95

References . . . 97

LIST OF TABLES

7.1 Equations tested for modeling $\alpha(R)$ and $\sigma_e(R)$ 45

LIST OF FIGURES

2.1 σ^0 as a function of relative azimuth angle 5

3.1 The ASCAT and TRMM collocations over the global oceans. 12
3.2 ASCAT σ^0 across wind speed 13
3.3 ASCAT σ^0 across rain rate 14

4.1 Fitting of PIA as a function of rain rate for CSRR rain GMF 17
4.2 Linear fit of coefficients mapping rain rate to PIA for CSRR rain GMF 18
4.3 Epanechniov and histogram filtered σ_e across rain rate 20
4.4 Wind speed RMSE for CSRR rain GMFs across incidence angle 21
4.5 SWR and WO estimated wind speeds compared with ECMWF for CSRR rain GMF 22
4.6 Rain rate RMSE for CSRR rain GMF 23
4.7 Satellite geometry 24
4.8 Normalized histogram of freezing height 26
4.9 Visualization for CSRR stochastic study 27
4.10 Boundaries of f_α along a sweep of rain rates. 28
4.11 Difference between the reported TRMM PIA values and the estimated TRMM PIA values 29

5.1 Wind speed RMSE for ERA rain GMF across incidence angle 33
5.2 Rain rate RMSE for ERA rain GMF 34
5.3 Range of ERA gain coefficient for ASCAT 36
5.4 Heat and contour map of σ^0 across speed compared to CMOD5 estimate 37
5.5 Noise floor versus change caused by gain term 38

6.1 ED calculated PIA across rain rate 41
6.2 Effective distance across rain rate 42

7.1 σ_w estimated by raw (DDWB) and synthetic (CMOD5) methods 44
7.2 $\sigma_e(R)$ across rain rate assuming $\alpha(R) = 1$ 46
7.3 Rain GMF modeling α and σ_e terms for bug affected loss function 47
7.4 Rain GMF modeling α and σ_e terms for bug corrected loss function 48
7.5 Wind speed RMSE for DDWB rain GMF across incidence angle 50
7.6 Estimated performance range for 4 difference DDWB rain GMFs 51
7.7 Errors of fit and wind estimation errors for DDWB and CMOD5 rain GMFs . . 53

8.1 Illustration of rain influence in wind retrieval 55
8.2 Basic example of NN structure 58
8.3 Model complexity and affects of underfitting and overfitting 61
8.4 Overfitting and underfitting viewed with the validation 62

9.1 Example ROC curve 69
9.2 Example of how the AUC is calculated 70

10.1 F1-scores across epochs for different NN input configurations 82
10.2 Concentration of minority data's affect to different scoring metrics 84
10.3 Keras code for the best 1x1 NN configuration. 86
10.4 Flow of data through the 7x7 configuration NN 87
10.5 Exploration of patch "sameness" 89
10.6 Training and validation loss for "patch sameness" corrected data 90

11.1 Wind speed RMSE for best performing rain GMFs 93
11.2 Rain RMSE best rain estimating rain GMF 94

CHAPTER 1. INTRODUCTION

1.1 Wind Estimation by Scatterometry

After radar became ubiquitous during WWII, succeeding generations of scientists found ways to use radar for a variety of applications. As early as 1960 it was shown that wind produces unique changes in radar backscatter over the ocean [1]. In 1978 NASA launched the Seatsat-A scatterometer, which showed that satellites could be used to gather ocean wind measurements, enabling wind measurements to be gathered over much larger areas in shorter periods of time.

This research deals with the Advanced Scatterometer (ASCAT) launched by the European Organization for the Exploitation of Meteorological Satellites. ASCAT is an active scatterometer which transmits a pulse of energy and measures the signal that returns back towards the scatterometer from the surface. The return signal which is called backscatter, or σ^0, is used in surveillance radars to detect objects. In the case where the sensor is observing an object significantly larger than its beam, such as the ocean, σ^0 reflects the roughness of the object's surface.

Satellite measurements of wind speed over the ocean are made possible by small capillary waves created by wind. The transmitted signal wavelength is on the same scale (centimeters) as the capillary waves, which induces Bragg scattering which is highly sensitive to direction. ASCAT, like many other scatterometers, exploits this sensitivity by acquiring multiple measurements of the same surface location at varying azimuth angles, which enables estimation of wind direction.

The relationship between wind velocity and radar backscatter is usually tabulated empirically in a geophysical modeling function (GMF). Most GMFs are only modeled in the absence of rain, making wind estimates less accurate when rain is present. Although

ASCAT uses a C-band frequency which is less affected by rain, [2] and [3] show that rain causes significant error to the wind estimation product.

In this thesis I derive a rain GMF which works with the existing wind only (WO) GMF to lower the wind estimation error in the presence of rain. The process of using both the WO and rain GMFs to estimate wind speeds and rain rates is called the simultaneous wind and rain (SWR) retrieval. A successful rain GMF is created that lowers the root mean squared error (RMSE) of wind estimation in the presence of rain from 3.02 to 2.61 m/s.

1.2 Machine Learning

In addition to developing a C-band SWR GMF, I also create a neural network (NN) which enables the use of the SWR GMF on global data. The SWR algorithm at Ku-band is only optimal over data samples that are influenced by rain, otherwise the WO retrieval is generally optimal [4]. I suspect the same pattern holds for the C-band SWR model, which makes the SWR model useful if the presence of rain is already known. Because global high resolution rain data is infrequent, it is desirable to estimate the occurrence of rain using data from the ASCAT sensor alone. The neural net detects the presence of rain using the scatterometer's own data so the SWR model can be used appropriately.

NNs are a subset of machine learning, which is the study of algorithms that "learn" from large sets of data and can be taught to perform a wide range of tasks from playing chess to controlling social media content. Previous research done at Ku-band, which is significantly more sensitive to rain, was able to use complex statistical analysis to detect rain [4]. By comparison, C-band instruments are far less affected by rain, making a detection system more of a challenge.

Various parameters for constructing the NN are explored. A preliminary NN is build for a Ku-band scatterometer's data (QuikSCAT). Several QuikSCAT NNs were effective, and subsequent ASCAT NNs were likewise developed which appeared to detect rain with considerable accuracy. Unfortunately, a subtle form of contamination among the testing data proved to be the cause of the NN's "accuracy". NNs constructed with uncontaminated data sets had greatly reduced performance. The NNs were unsuccessful at lowering the total wind speed error by select rain samples for the SWR algorithm.

It was found that by entirely removing samples the NN labeled as rainy reduced the RMSE of speed by 2.68%. This, however, was at the cost of removing 6% of all the data, even though only 10.6% of that data contained rain.

1.3 Thesis Layout

This thesis is split into two major sections which discuss the rain GMF and machine learning portions of the research. Chapters 2 through 7 detail the rain GMF research while Chapters 8 through 10 handle the machine learning aspect.

Chapter 2 discusses a few background processes like the general approach to a rain only GMF and the wind retrieval processes. The data set used is explored in Chapter 3, after which four different approaches to creating a rain GMF are discussed in Chapters 4 through 7.

Chapter 8 introduces key concepts concerning neural networks and Chapter 9 presents a preliminary "best case" study of NN rain detection using QuikSCAT. Chapter 10 explores rain classification networks on ASCAT data. The rain GMF and machine learning research are combined in a comprehensive study in Chapter 11.

CHAPTER 2. BACKGROUND

Traditional WO GMFs map a set of input parameters, such as wind speed, relative wind direction, and incidence angle to an expected σ^0. Previous research done by H. Hersbach et al., created a WO GMF for C-band scatterometers [5] called CMOD5. This GMF is called a "wind only" model because it does not attempt to account for rain presence, nor does it try to estimate rain levels. Since CMOD5 is the only WO GMF employed in this thesis, I also refer to it simply as the WO GMF. The objective of this research is to create a useful rain GMF to be used in conjunction with the existing WO GMF.

This section discusses the geophysical inverse problem and how to handle it, the basic framework from which rain GMFs are based on, and the restrictions to the parameterization of rain GMF coefficients.

2.1 GMFs and the Retrieval Process

Solving the wind speed vector or rain rate from the scatterometers σ^0 measurements, also known as retrieval, is more complex than merely applying an inverse GMF. For the WO GMF, σ^0 values are not unique, meaning that different combinations of wind speed, direction, and incidence angles can result in the same σ^0 value. Because the GMF is not unique, reversing the GMF to solve for wind speed and direction given σ^0 is an ill posed estimation problem. Solving for X, given a Y for $Y = X^2$ is an example of an ill posed problem since the sign of X can not be determined, leaving us with multiple ambiguous solutions. In a similar manner, inverting the WO GMF results in an ambiguous solution, see Fig. 2.1. To handle this geophysical inverse problem the WO retrieval process uses a maximum likelihood algorithm and multiple σ^0 measurements of the same surface location collected from different azimuth angles, also called flavors, to estimate wind speeds and directions [6].

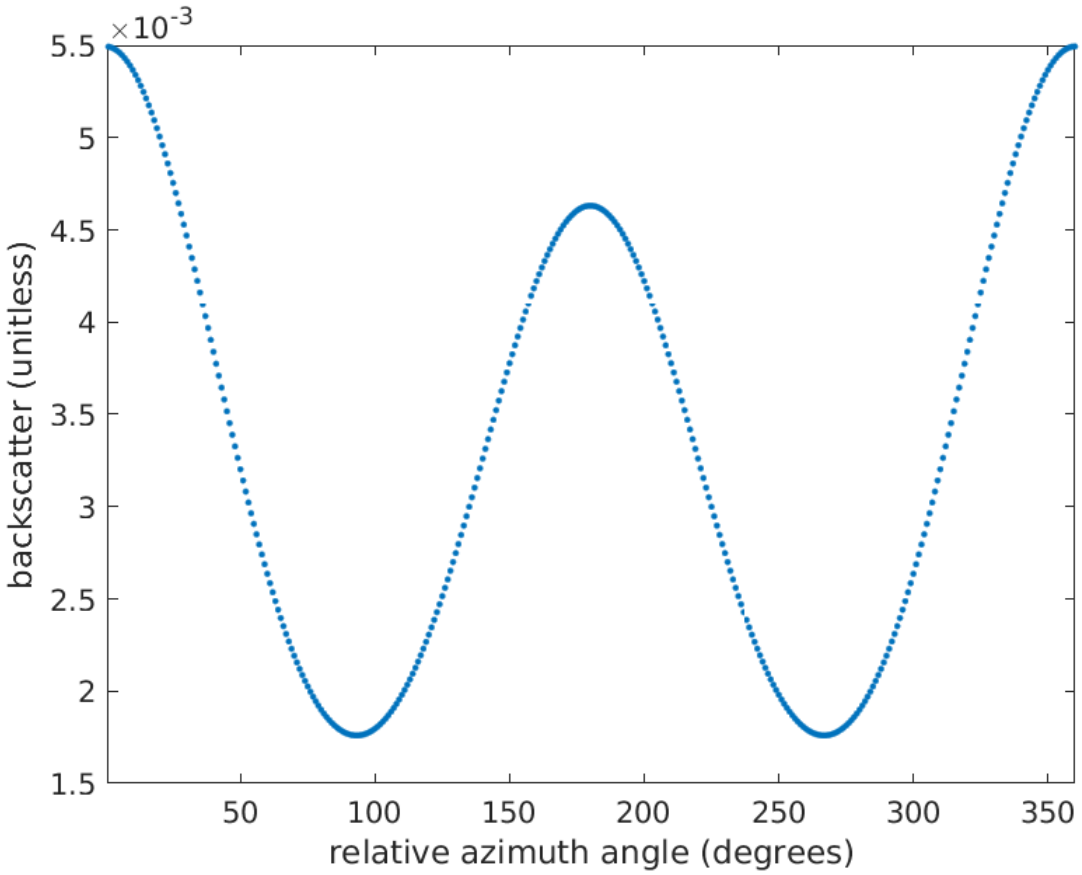


Figure 2.1: Example of WO GMF backscatter for a fixed incidence and wind speed across a varying relative azimuth angle. Relative azimuth angle is the difference between the wind and the radar azimuth directions.

Maximum likelihood estimation is performed by first defining the probability density function (pdf) of a noisy σ^0 as a normal distribution written as,

$$p(z|s,d;\theta,\rho) = \frac{1}{\sqrt{2\pi V}} e^{-\frac{1}{2}(z-GMF(s,d;\theta,\rho))^2/V}, \tag{2.1}$$

where z is the measurement, s represents wind speed, d is the relative angle between azimuth angle and wind direction, θ is incidence angle, and ρ is the polarization (vertical only in the case of ASCAT). V denotes the measurement variance which is equal to $(K_p)^2 z^2$. The geophysical modeling function $GMF(s,d;\theta,\rho)$ returns σ^0 given the input wind speed and direction at a specific incidence angle and polarization.

The measurements z are treated as independent so the joint pdf of the set $\mathbf{z}$ of N measurements can be written as,

$$p(\mathbf{z}|s,d) = \prod_{n=1}^{N} p(z_i|s,d;\theta_i). \tag{2.2}$$

Maximum likelihood estimation attempts to find the most likely speed and direction given the N measured σ^0 values. This is done by maximizing the log-likelihood function $J(s,d)$,

$$J(s,d) = -\sum_{i=1}^{N} \left\{ \ln V_i + \frac{1}{2}(z - GMF(s,d;\theta_i))^2/V_i \right\}. \tag{2.3}$$

Another helpful form appears when the variance terms and constants are dropped. In this case

$$J(s,d) = -\sum_{i=1}^{N} \left\{ (z - GMF(s,d;\theta_i))^2 \right\}. \tag{2.4}$$

Notice that maximizing $J(s,d)$ is equivalent to minimizing the sum of squared error between the measured σ^0 and the backscatter value estimated by the WO GMF.

2.2 Basic Rain GMF Framework

Rain affects the surface radar backscatter caused by wind in ways that both diminish and increase the signal while over the ocean. Rain in the atmosphere attenuates signals transmitted through it. At the same time, scattering from airborne rain increases the backscatter. Rain impacting the surface of the ocean also creates a rougher surface which increases the backscatter. A simple phenomenological model is used to account for each of these effects. It was also used by [4] when exploring the creation of an ASCAT GMF at 25 km resolution. The model is given by

$$\sigma^0 = (\sigma_w + \sigma_{sr})\alpha + \sigma_r, \tag{2.5}$$

where σ^0 is the observed backscatter, σ_w is the backscatter caused by the ocean surface roughness induced by wind, σ_{sr} is the additional backscatter from the ocean surfaced made by rain, α is the rain attenuation, and σ_r is the backscatter caused by airborne water.

The model can be simplified to

$$\sigma^0 = \sigma_w \alpha + \sigma_e, \tag{2.6}$$

where terms are combined into an effective σ_e term.

The rain GMFs explored in Chapters 4 through 6 use Eqs. 9.5 and 2.6 as their basic frameworks. These rain GMFs use different methods to estimate the basic parameters in Eq. 9.5.

2.3 SWR retrieval

SWR retrieval uses Eq.2.6 and another maximum likelihood estimation to more accurately estimate the wind speed and direction. The α and σ_e terms are assumed to be functions of rain rate r given an incidence angle. By applying the gain and additive terms to the GMF and solving the objective function, the maximum likelihood SWR objective function is found to be

$$J(s,d,r) = -\sum_{i=1}^{N} \left\{ \ln V_i + \frac{1}{2}(z - GMF(s,d;\theta_i)\alpha(r;\theta) - \sigma_e(r;\theta))^2 / V_i \right\}. \tag{2.7}$$

2.4 Parameterization Restrictions

One must be cautious with the parameters used to model the rain GMF because these same parameters need to be estimated during the SWR retrieval processes. Maximum likelihood estimation has error boundaries which only increase as more parameters are estimated, especially when the parameters are of no interest, or are nuisance parameters [7]. As such, it is helpful to limit the number of parameters being estimated. In this case, the parameters used to create the rain model are restricted to rain rate and incidence angle, giving the estimation problem only one more parameter to estimate. This restricted parametrization is partially due to historic code implementation. Future work could improve this modeling by allowing the gain and additive terms to vary given wind speed or other parameters.

When there exists parameters affecting the rain model besides rain rate and incidence angle, these parameters are treated as random variables (RVs). These RVs introduce noise into the GMF which in turn can degrade SWR performance.

CHAPTER 3. ASCAT, TRMM AND ECMWF COLLOCATED DATA SET

The data used for this study comes from the collocation of measurements of ASCAT SZF backscatter wind products, the Tropical Rain Measuring Mission (TRMM) satellites precipitation radar data set, and the European Centre for Medium-Range Weather Forecasts (ECMWF) 2A23 wind data set. Collocated measurements are those that happen over the same area at approximately the same time. This chapter provides a brief overview of the different sources of data, the processes which increases ASCAT resolution, and the collocation processes.

3.1 Data Sources

ASCAT is a fan beam design scatterometer with six fixed position antenna beams operating at 5.255 GHz. The six beams are symmetrically placed on the right and left of the satellite. C-band scatterometers, like ASCAT, have been shown to be less responsive to rainfall than Ku-band sensors such as TRMM [8]. Because rain has less of an influence over the the C-band signal, rain can be more challenging to model and estimate in C-band than for Ku-band. ASCAT flies on the Metop satellites and is processed into a 2.5 km Ultra High Resolution (UHR) using BYU's UHR processes [9]. Backscatter at each sample collocation was measured by the three beams on the left or right side of the sensor. These three different backscatter measurements are called flavors, which are split up into forward, mid and aft positions. The study uses ASCAT data taken over the year 2009.

The TRMM satellite is used to collect rain information that is used to identify where ASCAT samples are affected by rain. As the name suggests, TRMM is limited to tropical latitudes. It looks down at the Earth from a much smaller incidence angle and at a higher frequency (13.8 GHz) than ASCAT. It's incidence angle, and Ku-band frequency make TRMM much more responsive to airborne water than ASCAT. TRMM data products have roughly

4.5 km resolution and represents the rain profile as a column. TRMM data products also include an estimate of freezing height, and path integrated attenuation.

ECMWF near-surface wind data is used to describe wind information that is used as a benchmark of accuracy. ECMWF's numerical wind predictions are synthesised using inputs from various sensors. ECMWF data has has a time resolution of 6 hours and a spatial resolution of 1 degree in latitude and longitude, which is approximately 111 km. The wind vector field is interpolated in space and time to the locations of ASCAT wind vector cells (WVCs) because of ECMWF's low time and space resolution. It is possible that this resolution difference is a source of error compared to the higher resolution scatterometer wind measurements.

3.2 UHR Process

Standard ASCAT scatterometer products achieve resolution by using range gating. These products are created by "slice" (SZF) and 25 km "cell" measurements which are different shaped resolution cells. The slice measurements are long and thin, and have significant overlap with other measurements. The UHR processes takes advantage of this overlap and the spatial measurement response to reconstruct the surface σ^0 into a posting resolution of 2.5 km [9].

Although higher resolution data is desirable because it can be used to study mesoscale wind features in tropical cyclones and convective storms, modeling at UHR leads to some complications. First, because rain can start falling anywhere below the freezing height, which can be upwards of 4 km or higher, ASCAT measurements at high incidence angles can project rain fall occurring over one resolution cell into neighboring cells. The horizontal spreading of the rain signal can contaminate neighboring wind vector cells.

Second, strong rain events which are smaller than the large 25 km WVC have a diffused effect on the overall radar backscatter. At higher resolution, this averaging effect is likely to have a weaker effect. Potentially this could create a divide between UHR and the standard 25 km resolution rain models, making them noninterchangeable. This is further complicated by the non-linear wind retrieval processing.

3.3 Data Collocation

Collocation is the process of matching data from different satellites that observed the same location on earth at approximately the same time. Because TRMM and ASCAT come from satellites with different orbits, their data needs to be collocated in order to be compared. In this study the ASCAT collocation points must occur within 10 minutes of a TRMM sample and spatially be within 0.01° in latitude and longitude of the reported TRMM position.

One year's worth of ASCAT and TRMM collocation data provides nearly global coverage and over 15 million UHR collocations, with over 1.1 million of those affected by rain. Global coverage is desirable, however, since TRMM only covers the tropics the resulting rain GMF is not fully tuned for global data, but only between positive and negative 40° longitude.

3.4 Backscatter's Relation to Wind Speed and Rain

Radar backscatter increases with increased wind speeds. This is evident in the collocated data set. Fig. 3.2 shows a 2D histogram of the radar backscatter versus wind speed. There is a clear positive correlation between σ^0 and wind speed.

Rain also has an impact on the radar backscatter. Higher rain rates tend to have higher radar backscatter values in the same way that higher wind speed does.

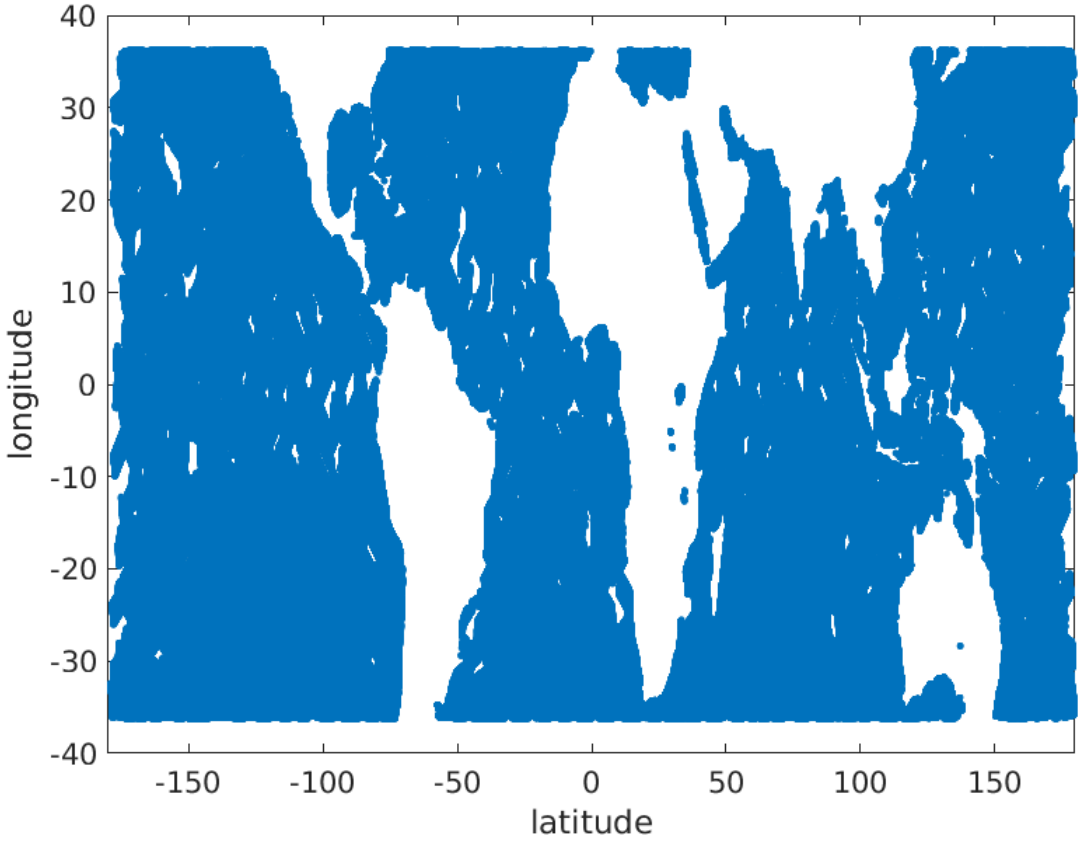


Figure 3.1: The ASCAT and TRMM collocations over the global oceans.

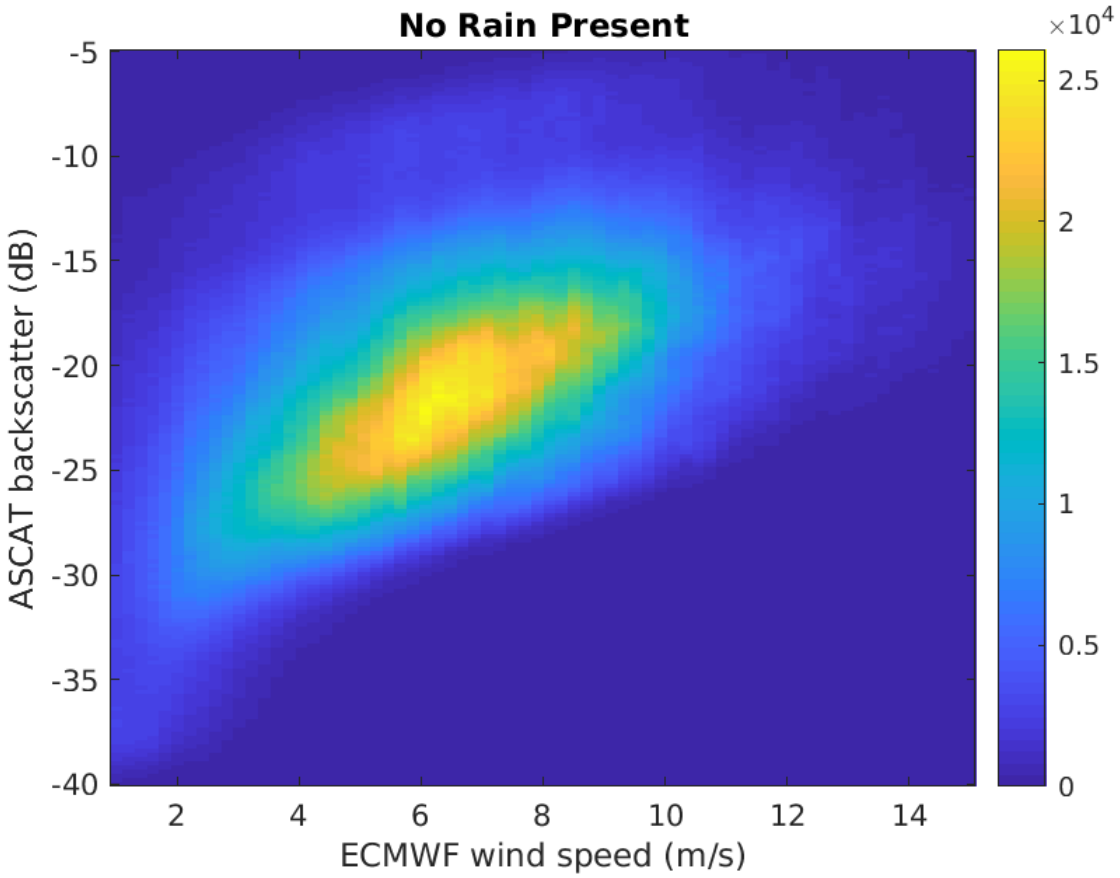


Figure 3.2: 2D histogram of wind speed versus radar backscatter under the condition that no rain is present. The presence of rain is determined using TRMM data.

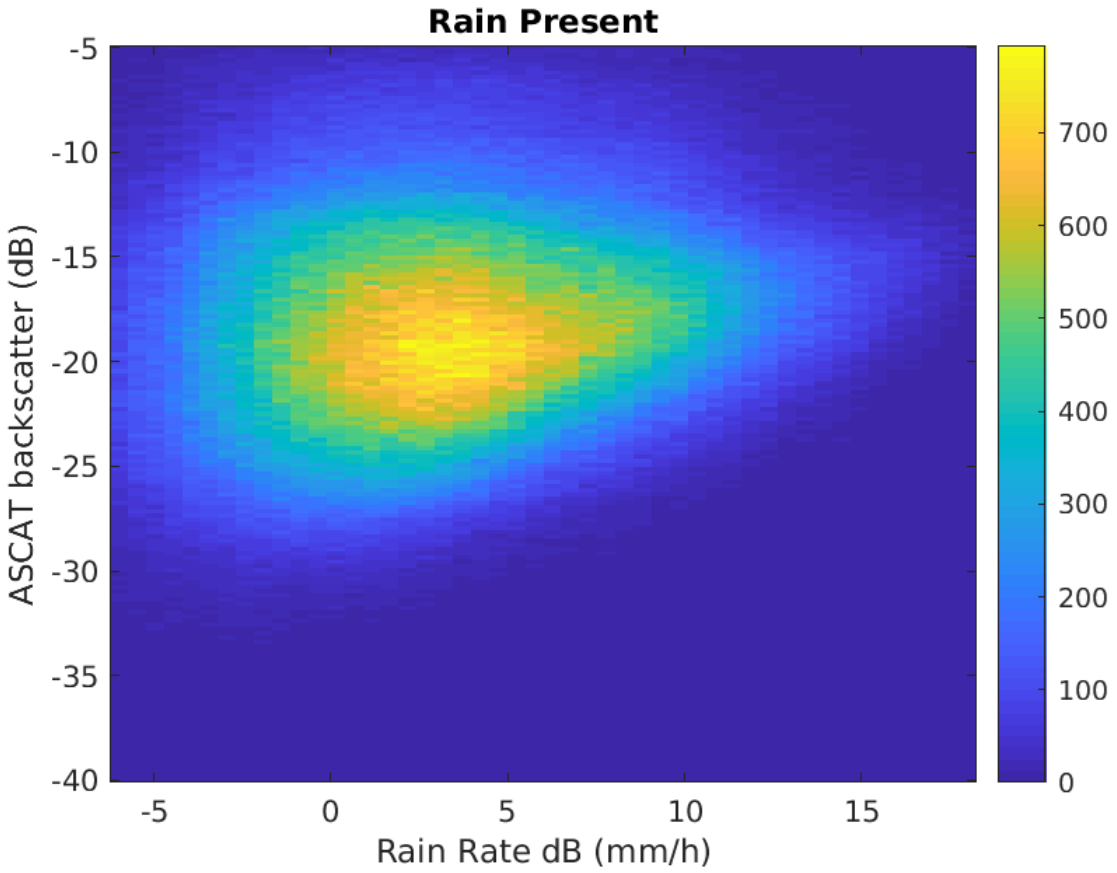


Figure 3.3: 2D histogram of rain rate and radar backscatter under the condition that rain is present. Compared to Fig. 3.2

CHAPTER 4. RAIN GMF 1: CONSTANT SURFACE RAIN RATE

In this thesis several rain GMFs are considered. The first is called the "constant surface rain rate" (CSRR) GMF. It attempts to estimate the attenuation α of ASCAT's signal using only the surface rain rate. It is assumed that the primary cause of this attenuation is rain falling at an approximately constant vertical rate from a specific height. The following sections discuss how the basic rain model coefficients are solved, fitting these coefficients, CSRR's performance on test data, a stochastic exploration of the observed results, and an overall summary.

4.1 Solving for Rain Induced Attenuation

Attenuation caused by rain is a function of frequency and rain rate [8]. The TRMM sensor reports a path integrated attenuation (PIA) which is the inverse of α. However, ASCAT operates at 5.255 GHz and TRMM operates at 13.8 GHz. Also, TRMM and ASCAT have different incidence angles. Thus a method to estimate C-band rain attenuation that accounts for ASCAT's different geometry and frequency is needed.

Instead of directly using the TRMM PIA, the CSRR uses the reported surface rain rate and the Internal Telecommunications Union (ITU) rain attenuation model [10] to estimate the attenuation per kilometer, or γ. The pathway through the rain layer is approximated as a constant rain rate from freezing height down to the ocean surface. The freezing height is the point at which airborne water freezes; hence, no rain occurs above this point. Since ice has a much smaller effect on the return signal, it can be ignored above the freezing height. TRMM's 2A23 data set includes an estimate of freezing height which is extrapolated from surface temperature measurements [11]. Snow and hail are treated as having negligible effect in comparison to rain attenuation because rain is more common and causes greater attenuation [8]. This is discussed further in Section 4.4.

After estimating the distance ASCAT's signal traveled through the rain layer using simple geometry, see Fig. 4.7, the freezing height and ASCAT's incidence angle are than multiplied by γ, to obtain a bidirectional estimate of the rain induced PIA. This estimate is $1/\alpha$ in Eq. 2.6. Next, σ_w is calculated using the CMOD5 function which uses ECMWF's estimate of oceanic wind speed and direction. Adjusting Eq. 2.6, the last unknown variable σ_e is solved for,

$$\sigma_e = \sigma^0 - \sigma_w \alpha \tag{4.1}$$

where σ^0 is the observed ASCAT backscatter, σ_w is the WO GMF backscatter estimate, α is the gain coefficient from Eq. 2.6, and σ_e is the effective backscatter which accounts for increase in backscatter due to rain.

4.2 Fitting the Basic Rain GMF Coefficients

Using collocated σ^0, wind and rain data points, the model coefficients which fulfill Eq. 2.6 are solved. The resulting coefficients are smoothed by fitting to a low order polynomial to create a function which returns the coefficient values σ_e and α given a rain rate and incidence angle.

Several methods are considered for the fitting processes. It is assumed that the model which most accurately matches the α and σ_e terms in the presence of rain will estimate wind speed the most accurately. The metric of mean squared error (MSE) is used to measure accuracy. It was observed early on that lower incidence angle samples are much more noisy than higher incidence angle samples, so that only samples which had an incidence angle of 30° or more are used.

4.2.1 Gain

The α coefficient, which symbolizes a gain between 0 and 1, has a mean of 0.986 and standard deviation of 0.04. The best fit (MSE of 5.37e-10) was found by dividing the data into 30 groups by incidence angle, and then fitting each group to a exponential curve as a function of rain rate as in Fig. 4.1.

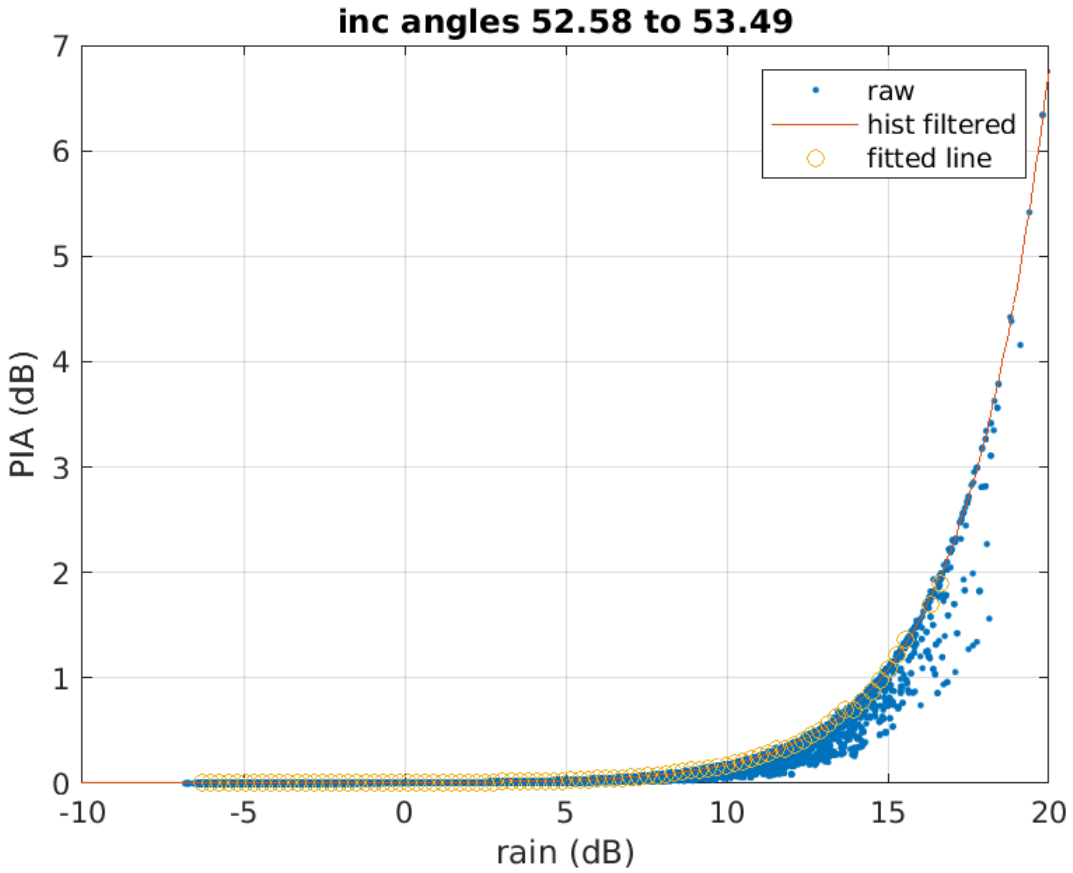


Figure 4.1: Scatter plot of PIA (dB) by rain (dB) showing an exponential relationship which become noticeably more noisy as rain values get larger.

For a given incidence bin, the PIA is first determined in log space, changing the curve in Fig. 4.1 from exponential to linear. Next, the data is put into a 2-D histogram function which bins the data into a 100 by 100 2-D histogram. The most densely populated bin for each column of like-valued rain rate is then determined and used in the GMF table. This effectively filters the data. The 2-D histogram filter outputs are fit by a first-order polynomial. This is repeated for each incidence angle bin.

The resulting set of first-order polynomial coefficients are fit along the average incidence angle of each incidence bin. This allows the estimate of α to be smooth as a function of rain rate and incidence angle.

For comparison's sake, another fitting that only considers rain rate and not incidence angle is created. This rain-only fitting results in a MSE of 7.27e-5 which is considerably

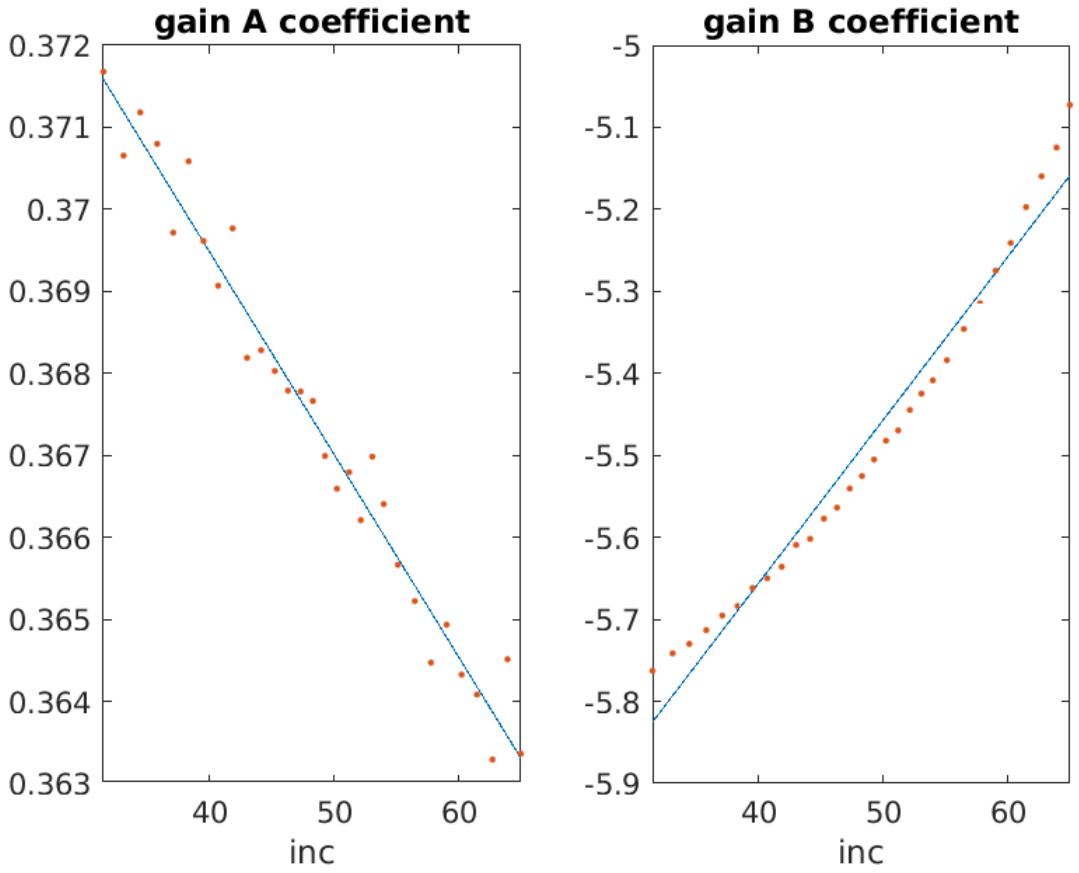


Figure 4.2: A and B coefficients which satisfy $\log(\text{PIA}) = A * \text{rain} + B$ for each incidence bin and their smooth line approximation.

higher than the fitting that uses both incidence angle and rain rate (a fit error of 5.37e-10). Thus fitting α by incidence angle and rain rate is significantly more accurate than fitting by rain rate alone.

4.2.2 Effective Backscatter

The process used to fit σ_e is very similar to the process for α, with a few variations to cope with its higher noise content. Because σ_e has values which are negative, it can not be fit in log space.

Two filtering methods are experimented with to reduce variance. The first is the 2-D histogram filtering used with the gain, which returns a curve that appears to be unbiased through the most densely populated regions of a density plot. The second type of filtering is

done using an Epanechnikov filter which appears to put the fitted curve slightly above the most densely populated regions, but better fits the general shape of the data at higher rain rates.

Three different filter types are testing by creating fits for each. First, a 2-D histogram filter. Second, a Epanechnikov filter, and finally a composite that uses the 2-D histogram filter for the lower incidence angles where it visually appears to fit the best, and Epanechnikov over the rest.

The relationship between rain rate, incidence angle and σ_e is more complex than with the gain. Plots of σ_e across rain, like Fig. 4.3, are similar to exponential curves, but they dip slightly below zero before rising exponentially. As the curve changes with higher incidence angles, the slight dip becomes less shallow and closer approximates an exponential curve. Also, there appears to be a cosine-like ripple along the incidence angle axis which increase in amplitude with larger rain rates. Because of this peculiar shape, I found the best results came when I first filtered the data for each incidence bin, and than ran the 2-D matrix of rain rate by incidence angle, through a small span Lowess filter. The Lowess filter created a smooth surface of the data without erasing the unique features which are difficult to fit.

An Epanechnikov filter followed by a Lowess filter resulted in the lowest MSE of 1.1758e-4. The histogram and the composite filters followed by a Lowess filter resulted in a MSE of 1.1970-4 and 1.1875e-4, respectively. This demonstrates that even though the composite and 2-D histogram filtering visually appeared to be the better fits, the Epanechniov was superior overall.

For comparison sake, another fitting was also created called "rain fitting". Instead of dividing up the data by incidence angle, this fit only takes into consideration the rain rate. It results in a MSE of 1.1876e-4. This suggests that fitting by rain rate and incidence angle increases the accuracy of the model, but only slightly.

4.3 Empirical Test

The CSRR GMFs are empirically tested by running the wind retrieval algorithms over the collocated data where rain is present. The wind speed is estimated using both WO and SWR retrieval. The SWR retrieval algorithm uses the WO estimates as a starting point

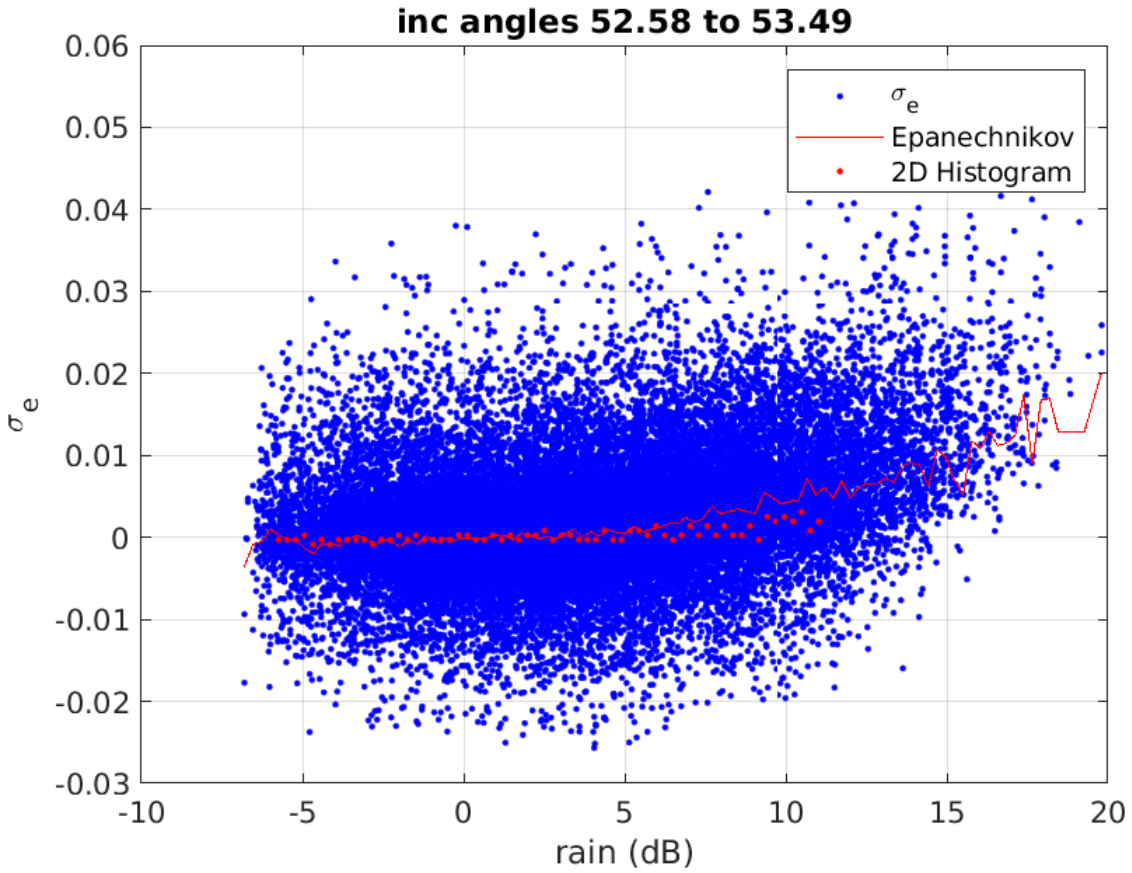


Figure 4.3: Noisy σ_r compared to Epanechnikov and histogram filter outputs. The histogram output is slightly below the Epanechnikov output at higher rain rates, however, when viewed using a density plot, the histogram filter runs directly through highest density regions.

and then runs a gradient descent processes. This approach has been shown to be effective with Ku-band rain models [4]. Because several different rain GMFs are tested in the SWR retrieval, the results are distinguished by the filtering method performed: Epanechnikov, the compound method, and rain only.

The SWR retrieval processes also returns an estimate of rain rate. However, as shown in subsequent sections, the estimated rain rates are erroneous to the point of uselessness.

4.3.1 Wind Speed Results

The RMSE between ECMWF and the estimated wind speed is calculated for both WO and SWR retrieval over data where rain is present. The RMSE is calculated for eight

incidence ranges which are plotted in Fig. 4.4. The WO retrieval performed the best across all incidence angles with a cumulative RMSE of 3.02. The rain fitted GMF is the next best with a cumulative RMSE of 3.04. The compound filtering, which appears to be visually the most representative, is third with a RMSE of 3.05. The fitting which most accurately reflects the α and σ_e terms, Epanechnikov, performed the worst with a RMSE of 3.07. Note that except for the no incidence angle model, for angles greater than 45°, all these models perform worse than using WO model.

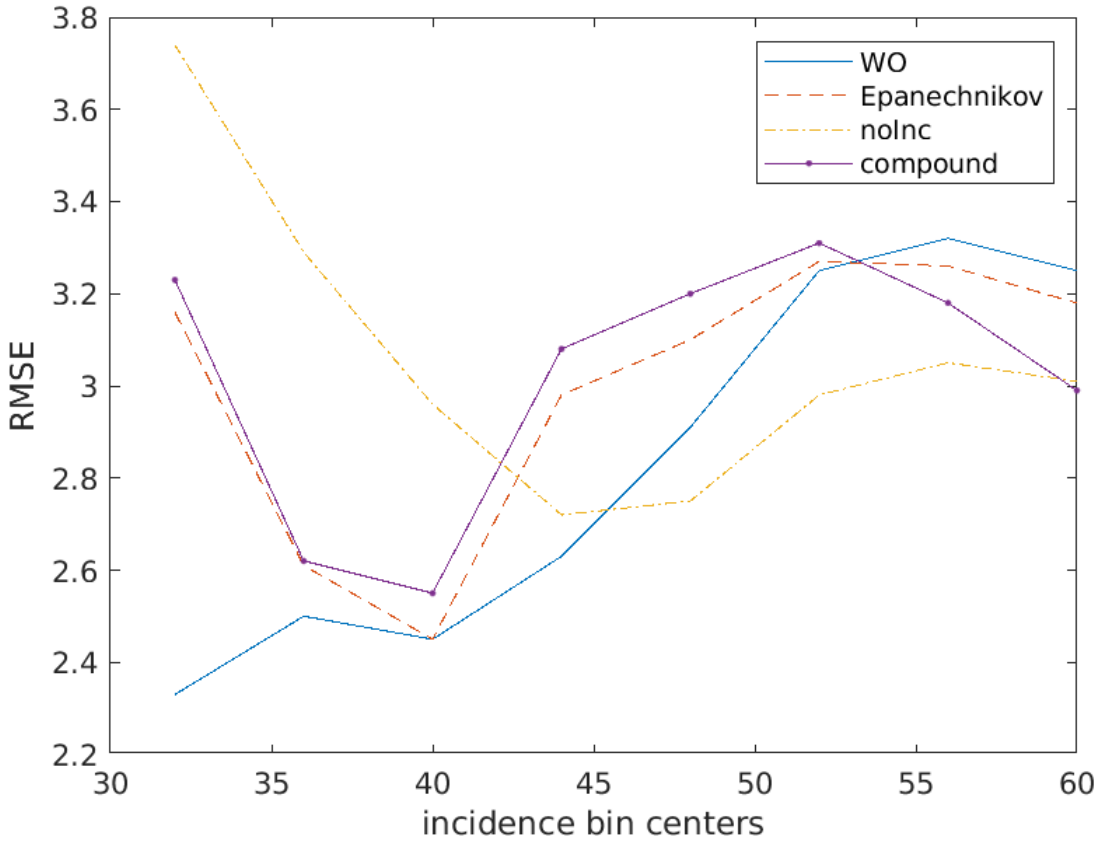


Figure 4.4: RMSE between ECMWF and estimated wind speeds. WO retrieval cases outperform all other versions below an incidence angle of roughly 45°. Above 45°, the SWR retrieval, which does not use incidence angle to fit the rain model, performs the best.

These results demonstrate that the method of estimating ASCAT's PIA using TRMM's reported surface rain rate and the ITU equations, is highly inaccurate. It is shown that the rain GMF which best reflects the σ_e and α terms of Eq. 2.6 (Epanechnikov), performs the

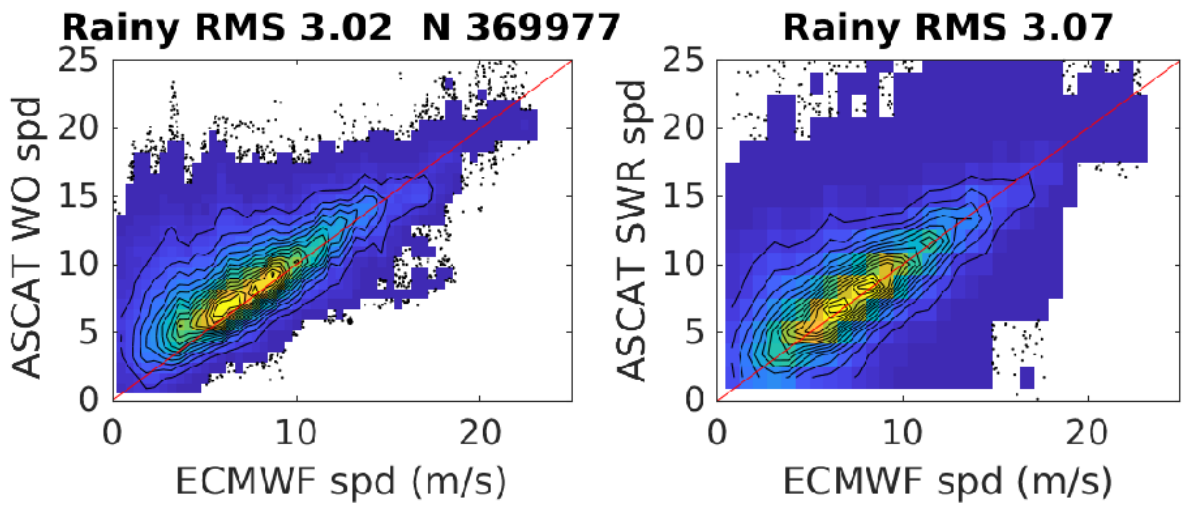


Figure 4.5: Cumulative empirical results across all incidence angles for both WO (left) and Epanechnikov (right) GMFs. WO retrieval has a narrow spread and is biased high. SWR using Epanechnikov modeling is unbiased, but performs worse and has a much larger spread.

worst during SWR retrieval. It seems reasonable that better fits to the σ_e and α terms should lead to better wind estimation if the underlying assumptions are accurate, but this appears to not be the case.

4.3.2 Rain Results

C-Band backscatter is less influenced by rain. As such, I expect the ability of rain estimation at C-Band frequencies to be less effective than estimation made by a Ku-Band sensor. The empirical results back this up.

For the best performing GMF, rain estimates across all incidence angles average a RMSE error 15.7 mm/h. This error generally drops off with increasing incidence angle, as demonstrated in Fig. 4.6.

4.4 Stochastic Inferences

The final rain GMF in Section 4.3 uses only rain rate and incidence angle; therefore it is important that the information used to build the rain GMF are either rain rate, incidence

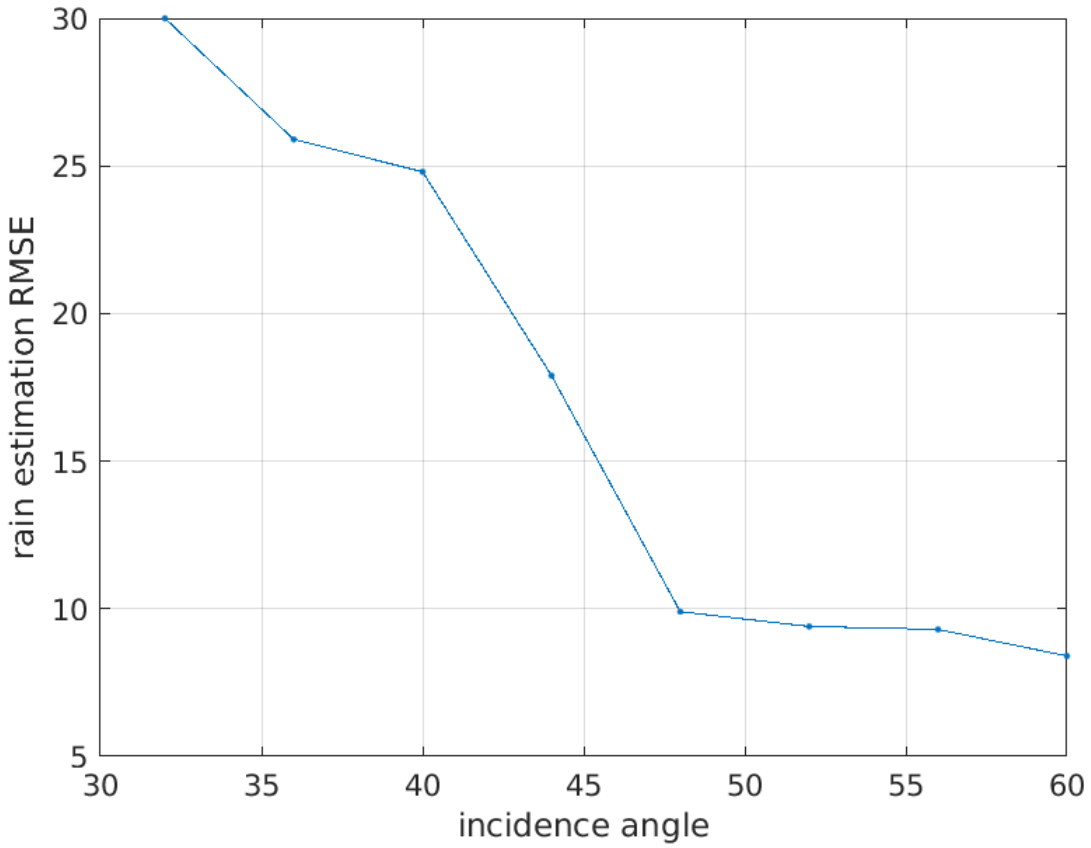


Figure 4.6: Average rain estimation RMSE for a GMF fitting only by rain rate and not by incidence angle. Rain estimations RMSE diminishes with increases in incidence angle. However, because the RMSE is so high, I decided to focus my attention on minimizing the RMSE of wind speed instead of rain.

angle, or some variable which can be described well by either. For this reason, any other variable is a random variable that adds noise, pursuant to discussion in Section 2.4.

Freezing height is a variable used to model the ASCAT PIA that has a weak correlation of 0.054. Therefore, freezing height is treated as a random variable. For contrast, the correlation between rain rate and estimated ASCAT PIA is 0.921.

To observe the effects of freezing height (H_f) on α, I start by defining γ in terms of H_f. Only H_f introduces noise since sensor noise is considered negligible. This noise is also passed down to σ_e, however, discussion of α is sufficient to demonstrate why the CSRR method works poorly.

The distance the signal travels through rain is dependant on freezing height by,

$$D = \frac{H_f}{\sin(90^\circ - \phi_i)}, \tag{4.2}$$

where D is the distance the signal travels through rain, H_f is the freezing height, and ϕ_i is the measurements incidence angle.

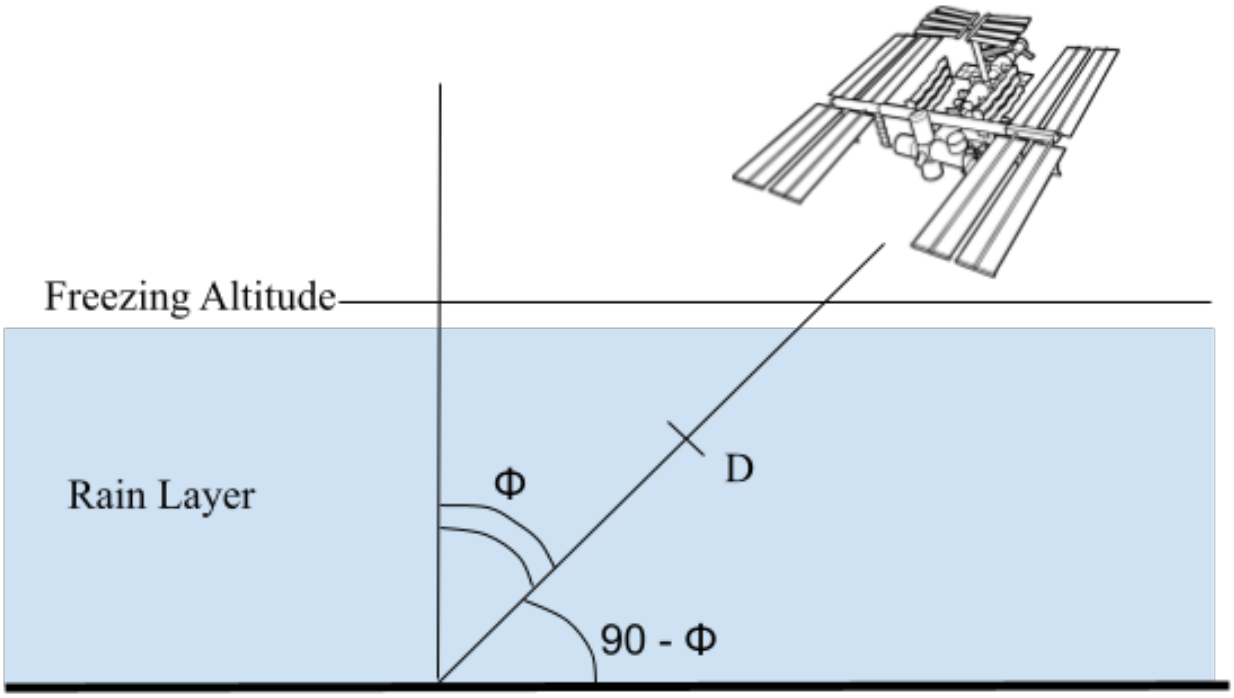


Figure 4.7: Satellite sensor geometry which relates incidence angle (Φ) and freezing height (H_f) to the total distance which the signal travels through rain.

By multiplying the round trip distance by γ, which is in units of dB of attenuation per km, the ASCAT PIA and α are respectively found to be,

$$PIA = \gamma \frac{2H_f}{\sin(90^\circ - \phi_i)}, \tag{4.3}$$

$$\alpha = 10^{-PIA/10}. \tag{4.4}$$

PIA accounts for the signal attenuation as the signal travels to and from the ocean's surface. To simplify the next steps Eq. 4.4 is rewritten as

$$\alpha = 10^{-\beta H_f}, \tag{4.5}$$

$$\beta = \frac{\gamma}{5\sin(90^\circ - \phi_i)}, \tag{4.6}$$

where β can range between 0 and 0.6698 due to the ASCAT's incidence angle range (23° - 65°) and an assumed maximum rain rate of 200 mm/h.

4.4.1 PDF of H_f

Freezing height as reported by TRMM has a difficult distribution to model. The PDF of freezing height from one year of TRMM data is shown in Fig. 4.8. The data appears to be sporadically divided between 3 groups: a tight Gaussian distribution centered around 2.68 km, a common uniform distribution from 3 to 5.2 km, and a distribution from 1.4 to 2.5 km.

To simplify the distribution, I treat the RV H_f as uniform between 1.4 and 5.2 km, $U(1.4, 5.2)$. This is, or course, an imperfect description of the H_f distribution, but it is sufficient to demonstrate key issues.

4.4.2 PDF of α in terms of PDF of H_f

I start the derivation with the CDF of α. The CDF_α is defined as

$$F_A(\alpha) = P(A < \alpha), \tag{4.7}$$

where $A = 10^{-\beta H_f}$. H_f is defined on the range of 0 to infinity. Notice that β is always positive and nonzero when rain is occurring. This means Y is monotonically decreasing as shown in Fig. 4.9 and can be rewritten as,

$$F_A(\alpha) = P\left(H_f > \frac{-\log_{10}(\alpha)}{\beta}\right), \tag{4.8}$$

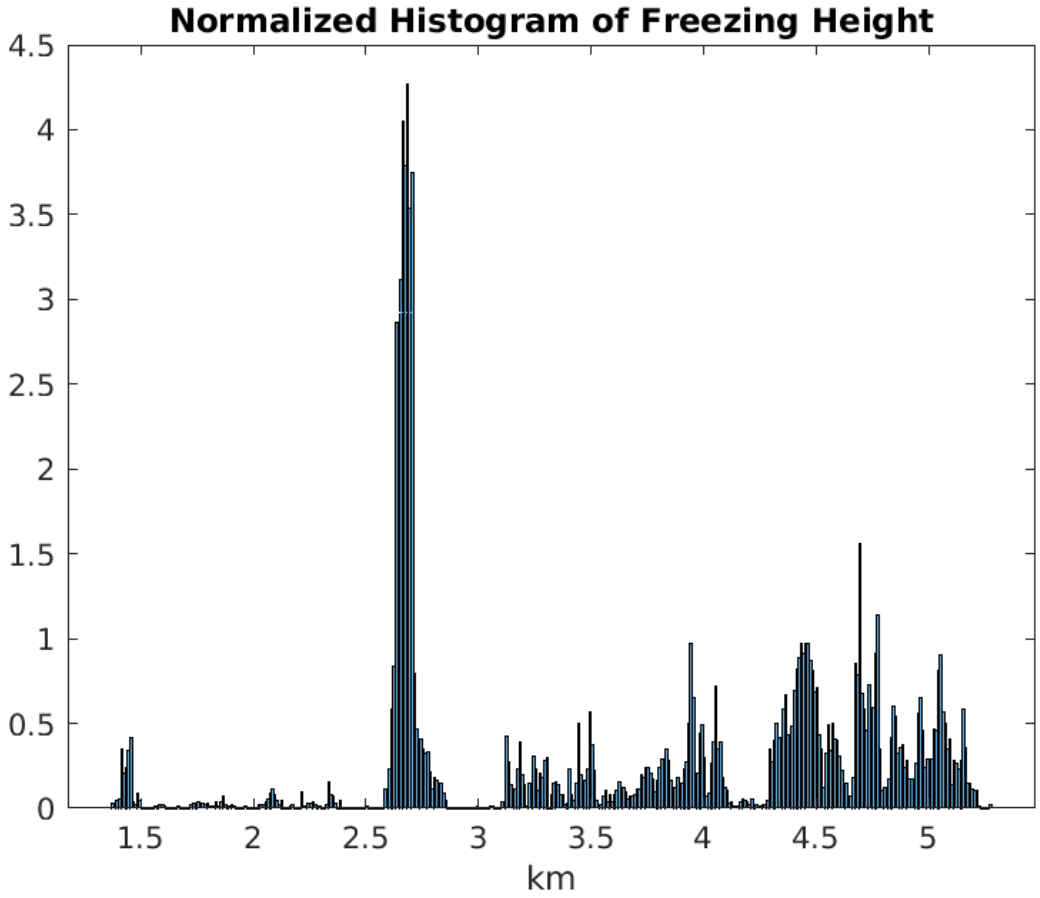


Figure 4.8: Normalized histogram of collocated freezing heights computed by ground temperature measurements taken by TRMM. The histogram shows preference for 2.68 km height, but is sporadic between 1.4 to 5.2 km.

which can be described by H_f's CDF as,

$$F_A(\alpha) = 1 - F_{H_f}\left(\frac{-\log_{10}(\alpha)}{\beta}\right). \tag{4.9}$$

By taking the derivative of F_A with respect to α it is found that,

$$f_\alpha(\alpha) = \frac{f_H(\frac{-\log_{10}(\alpha))}{\beta})}{\beta\alpha\ln(10)}. \tag{4.10}$$

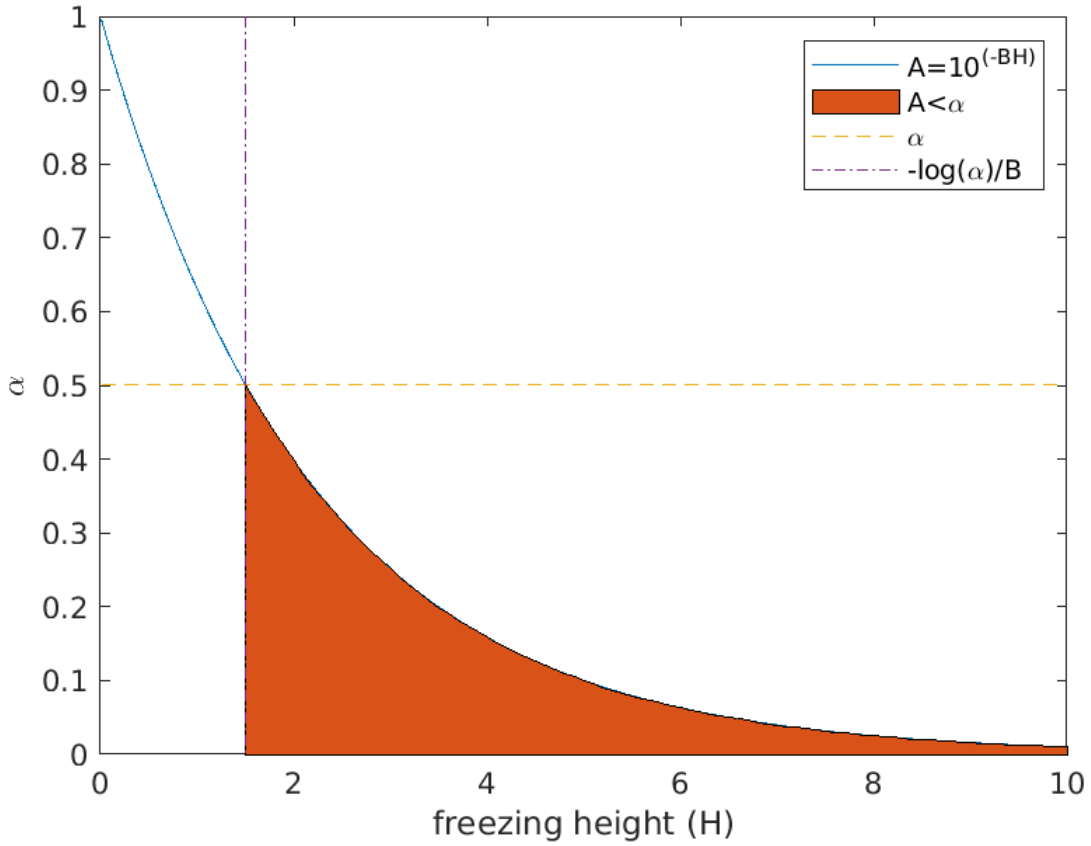


Figure 4.9: Visual relating H_f and α. To be the below the line α, H_f needs to be greater than $-\log(\alpha)/\beta$.

Since f_H is said to be uniform between 1.4 to 5.2 km, f_H is equal to 0.2632 over the bounds of $10^{-5.2\beta} < \alpha < 10^{-1.4\beta}$, so that Eq. 4.10 simplifies to

$$f_\alpha(\alpha) = \frac{0.2632}{\beta\alpha\ln(10)}. \tag{4.11}$$

It is insightful to notice how this distribution changes according to β, which is a function of rain rate. Observing solely the boundaries of the distribution along a sweep of rain rate, it is seen in Fig. 4.10 that increasing rain rates rapidly result in large uncertainty with regards to gain. As such, the derived α and σ_e values become less reliable as rain rates increase.

In summary, Gain, or α for each rain rate and incidence angle combination has a significant amount of uncertainty due to variability in the freezing height which is not

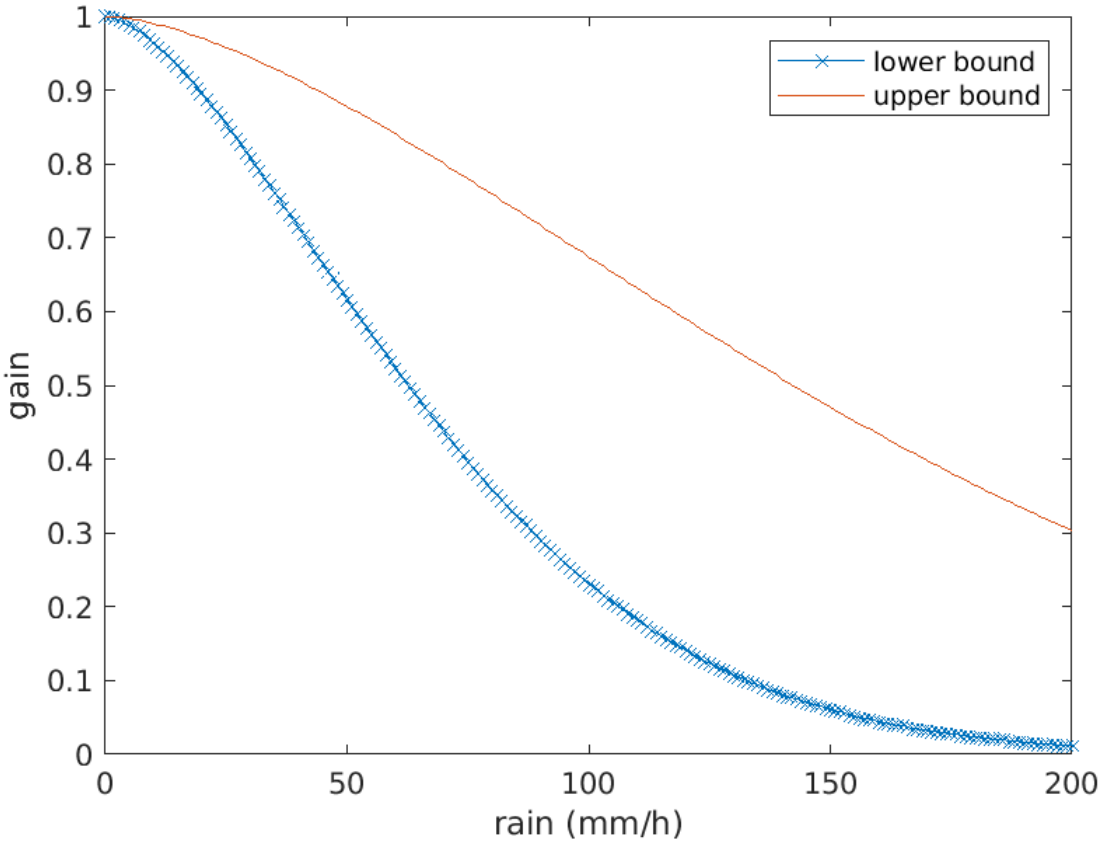


Figure 4.10: Boundaries of f_α along a sweep of rain rates.

modeled in the rain GMF. This uncertainty becomes more problematic with higher rain rates and peaks in uncertainty around 114 mm/h.

4.5 CSRR Conclusion

I have shown that the CSRR method is ineffective. This method estimates ASCAT's PIA by assuming the surface rain rate is constant from sea level to freezing height, and than utilizes the ITU equations to estimate attenuation given the TRMM reported rain rate. Some of the rain models outperformed the WO model during empirical testing over a range of incidence angles, but none were able to do so over all.

The CSRR method likely failed for a number of reasons. In Section 4.4.2, it was shown that the freezing height variable gave the gain term an very large and unpredictable range. The CSRR assumptions are put to the test by using the same method to estimate

TRMM's reported PIA given the surface rain rate. Because TRMM was the sensor which originally reported a PIA, it is easy to compare the CSRR estimate of TRMM frequency PIA against the measured TRMM PIA.

The difference between the reported and estimated TRMM PIA is -0.71 on average with a variance of 3.15. This is a significant error given that the average TRMM PIA value is 0.99. The RMSE for the TRMM PIA estimation is an 1.91. It is shown in Fig. 4.11 that the error distribution has a left sided tail. The estimate of TRMM PIA using the CSRR methodology is biased higher than the reported TRMM PIA. This suggests that the CSRR method will overestimate the PIA of a given frequency.

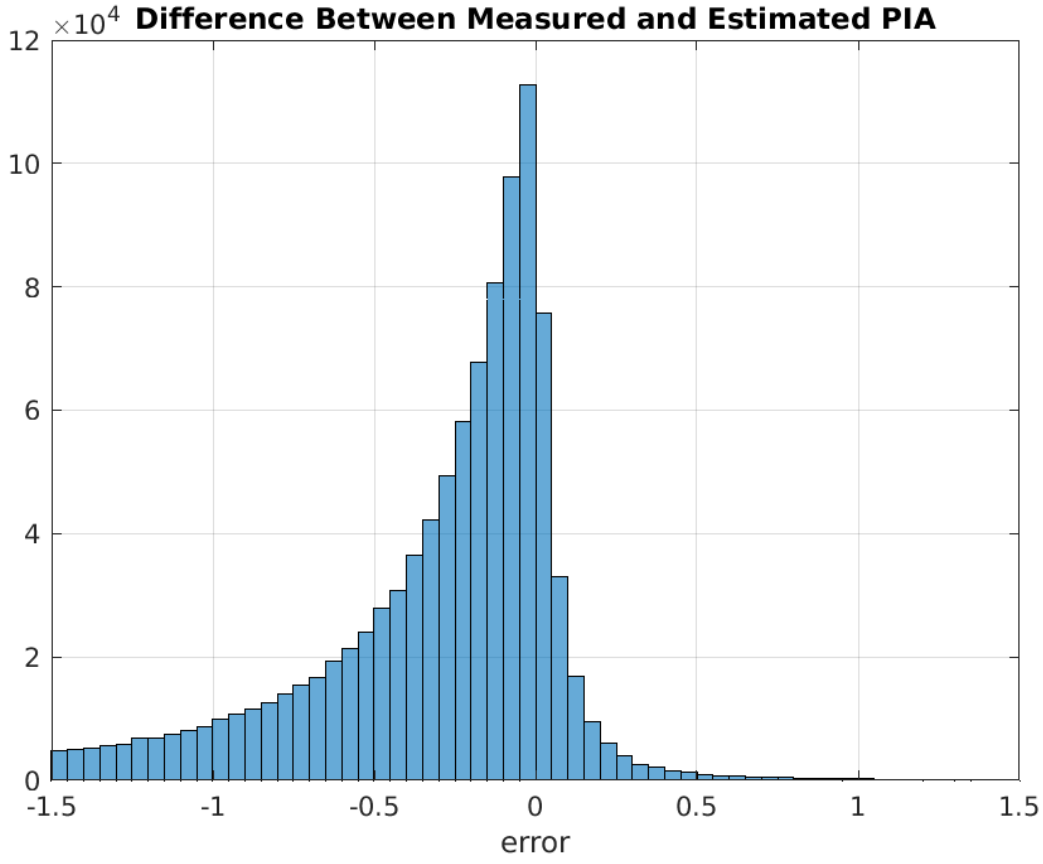


Figure 4.11: Difference between the reported TRMM PIA values and the estimated TRMM PIA values, which are made assuming the surface rain rate is constant from the sea level to the freezing height, and using the ITU equations.

The CSRR method overestimates the PIA parameter and introduces an unacceptable about of uncertainty into the model. The CSRR method was unable to outperform the WO model over all incidence angles and is not a productive way to gather the parameters needed to make a rain GMF which will enable better wind estimation in the presence of rain.

CHAPTER 5. RAIN GMF 2: EFFECTIVE RAIN ATTENUATION

This chapter develops another rain GMF. The "effective rain attenuation" (ERA) GMF uses the same base equation as the CSRR GMF (Eq. 2.6), but solves for the gain coefficient in way that appears to be more accurate. This new method provides insight into rain's influence at C-band. Using this method, it can be observed that the attenuation caused by rain is negligible at C-band, and the resulting GMF considerably improves the SWR retrieval performance. This chapter discusses the new method for gain estimation, the empirical test results, and statistical justification for the results observed.

5.1 Solving for Rain Induced Attenuation

The ERA method differs from previous models by using TRMM's reported PIA value to calculate attenuation, instead of only using the rain rate. TRMM's reported PIA is calculated by integration through the various layers of atmosphere. Its derivation is much more complex than CSRR's which assumed that the surface rain rate was constant throughout. The complexities of TRMM PIA calculation are explored in [12]. These details are beyond the scope of this thesis, except to point out that it is a more complex estimate of TRMM's PIA.

TRMM's reported PIA is defined as the integrated attenuation due to precipitation [12]. Because rain is the most common precipitation, the ITU rain attenuation equations [10] are used to solve for an "effective rain rate". The effective rain rate is the rain rate, which, if rain falls at a constant rate from freezing height down to the surface of the ocean, gives the same PIA as the one reported by TRMM when calculated using the ITU equations. The effective rain rate is solved by rearranging the ITU equation as,

$$R_{eff} = \left(\frac{P_T}{k_T D_T} \right)^{\frac{1}{a_T}} \tag{5.1}$$

where R_{eff} is the effective rain rate, P_T, is the TRMM-reported PIA, D_T is the distance the signal travels from the TRMM sensor through the rain, and a_T and k_T are coefficients determined by the frequency and incidence angle of TRMM.

After solving for R_{eff}, ITU equations are applied at the C-band frequency which has coefficients k_A and a_A. The resulting estimate the PIA for C-band frequency is,

$$P_A = D_A k_A R_{eff}^{a_A}, \tag{5.2}$$

where P_A is the PIA estimate for ASCAT and D_A is the distance the ASCAT signal is transmitted through the rain.

5.2 Empirical Test

After fitting the α and σ_e terms using the Epanechnikov and lowess filter method determined from the CSRR models, the resulting ERA models are empirically tested. These tests show that the ERA model is more accurate than both the WO and CSRR models. Unfortunately, rain estimation continues to be poor.

5.2.1 Wind Speed Results

The most efficient ERA GMF scores a cumulative RMSE of 2.81 m/s. For comparison, the WO and most accurate CSRR scores are 3.02, and 3.04 m/s respectively.

Fig. 5.1 shows the RMSE of two CSRR, one ER, and the WO models distributed by incidence angle. In general the accuracy of wind speed estimation in rainy conditions decreases with higher incidence angles, and the effective rain rate is superior to other CSRR and WO models for all but the highest incidence angles.

5.2.2 Rain Results

Rain estimation continues to be poor, similar to the CSRR GMFs, as seen in Fig. 5.2. The rain RMSE is so high that I choose to measure GMF performance by wind speed

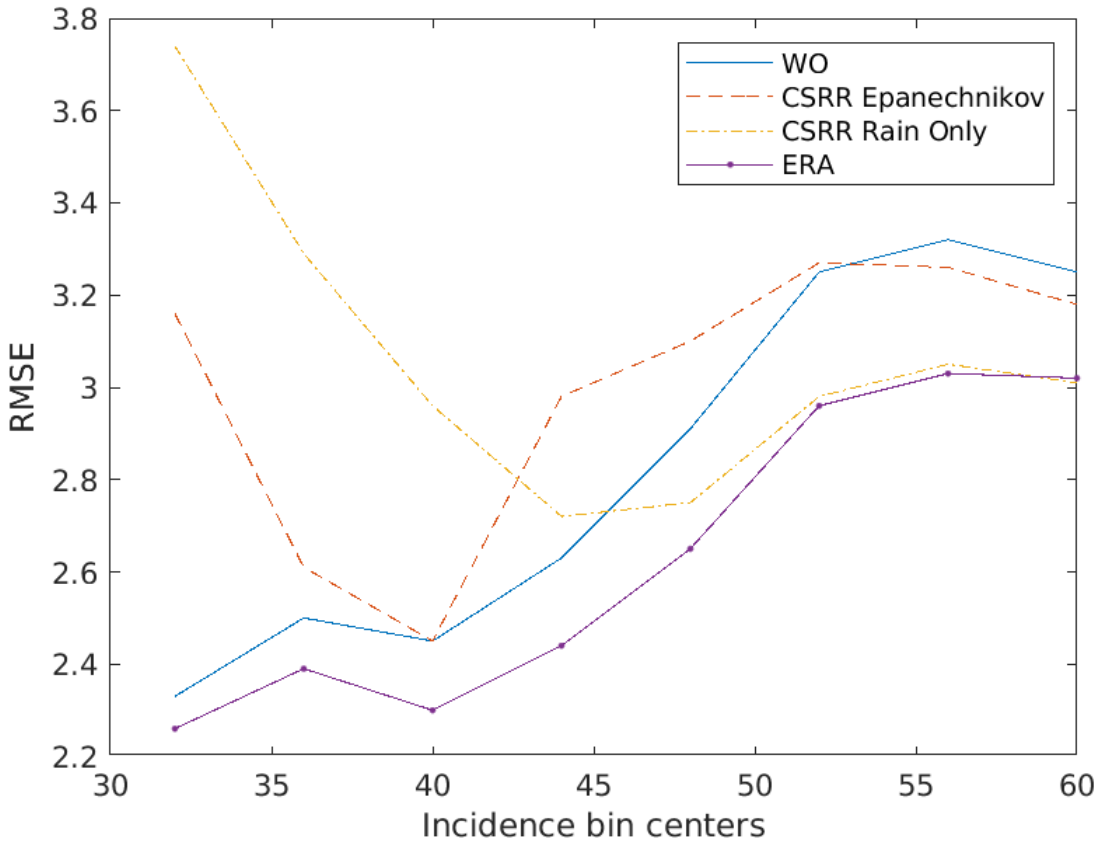


Figure 5.1: RMSE between ECMWF and retrieved values found using various models. Epanechnikov and rain only represent CSRR models which performed best over a range of incidence angles. The ERA model has a lower RMSE and performs better than all other models for all incidence angles, except at the very highest incidence angle where a version of CSRR barely outperforms ERA.

only. Thus in practice, the rain estimate is a nuisance parameter that is estimated, but likely should not be used.

5.3 Stochastic Inferences

Similar to the previous stochastic inferences made in Section 4.4, the random variable of freezing height (H_f) is approximated as a uniform distribution between 1.4 and 5.2 km. Instead of solving the probability density function for ASCAT's gain coefficient, I focus on the boundaries of H_f for the extremes of 1.4 and 5.2 km. This method adequately demonstrates why the ERA method is less affected by unmodeled RVs.

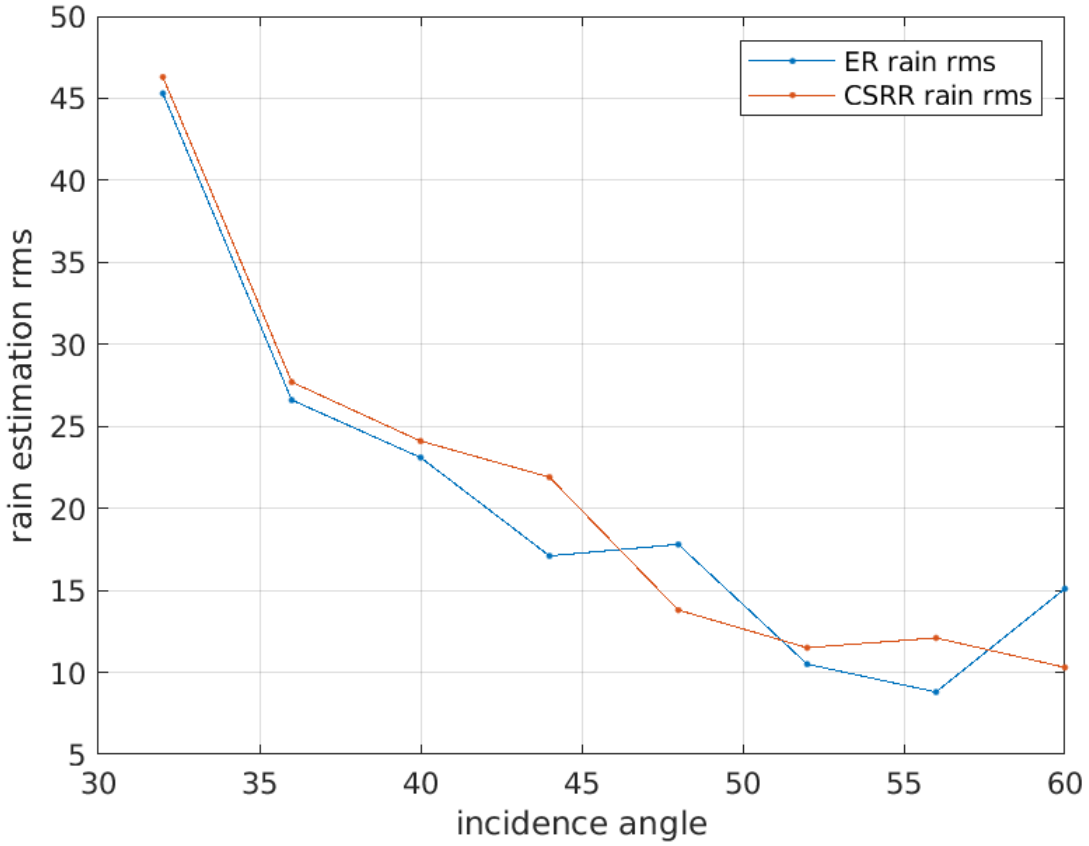


Figure 5.2: RMSE of rain estimation for both ERA and CSRR GMFs. It appears that even though ERA methodology improved wind speed estimation compared to the CSRR method, it has not improved rain estimation.

By substituting the definition for R_{eff} from Eq. 5.1 into Eq. 5.2, and defining the distance terms as $\frac{2H_f}{\sin(90-\theta)}$ where θ is the incidence angle of TRMM or ASCAT respectively, the equation for ASCAT's PIA is,

$$P_A = \left(\frac{2H_f}{\sin(90-\theta_A)}\right) k_A \left(\frac{P_T \sin(90-\theta_T)}{2H_f k_T}\right)^{\frac{\alpha_A}{\alpha_T}}. \tag{5.3}$$

Freezing height terms can be combined and the gain (α) can be solved as,

$$\alpha = 10^{-H_f^{\beta} G/10}, \tag{5.4}$$

where

$$\beta = 1 - \frac{a_A}{a_T} \tag{5.5}$$

$$G = \left(\frac{k_A 2^{\beta}}{\sin(90 - \theta_A)}\right)\left(\frac{\sin(90 - \theta_T)P_T}{k_T}\right)^{1-\beta}. \tag{5.6}$$

Fig. 5.3 shows the upper and lower boundaries of the random variable α when freezing height is considered uniformly distributed between 1.4 and 5.2 km. Notice the minute dynamic range of α which appears to be reasonably static regardless of the TRMM PIA. The TRMM PIA term along the x axis behaves similar to rain rate and represents rain rates ranging from 0 to 120 mm/h. Although the x axis of Fig. 5.3 is slightly different from Fig. 4.10, Fig. 5.3 clearly shows that uncertainty in the ERA method of calculating α is much smaller than the uncertainty from the CSRR method. Fig. 5.3 shows that gain varies only slightly as a result of the RV H_f. This variance seems to be largely independent of rain rate. It also shows that even with extremely large rain events, the ASCAT gain coefficient is hardly altered, hovering close to 1. This suggests that ASCAT σ^0 experiences little to no atmospheric attenuation due to rain and that effects which add to the total σ^0 dominate rains affects to backscatter. This can also be demonstrated using the σ^0 noise floor.

Because the CMOD5 model is imperfect, the limited accuracy of CMOD5 provides a noise floor on the estimate of σ_e. Fig. 5.4 shows how the dry measured σ^0 values differ from the CMOD5 estimate of σ^0. Also, Fig. 5.4 applies a box and whisker chart to data between 5.9 and 6.1 m/s. This chart shows the wide variance in the measured σ^0 values which contributes to the noise floor. When rain-caused adjustments to σ^0 are smaller than this noise floor these adjustments are insignificant.

The noise floor is calculated by taking the MSE between CMOD5's estimate of σ^0 and the raw observed σ^0 when not in the presence of rain. The estimated noise floor is 7.04e-5.

The noise floor is compared with the change to σ_w caused by gain, as calculated using the ERA method. The largest change induced by gain is when gain is equal to 0.999. The resulting MSE between σ_w and $\alpha\sigma_w$ when applied to all samples is 4.4e-10, which is far below the noise floor so the rain-caused gain is insignificant. Fig. 5.5 demonstrates the change induced by gain along a range of σ_w values and demonstrates that they are always well below the noise floor. Because the change caused by the gain term is well below the

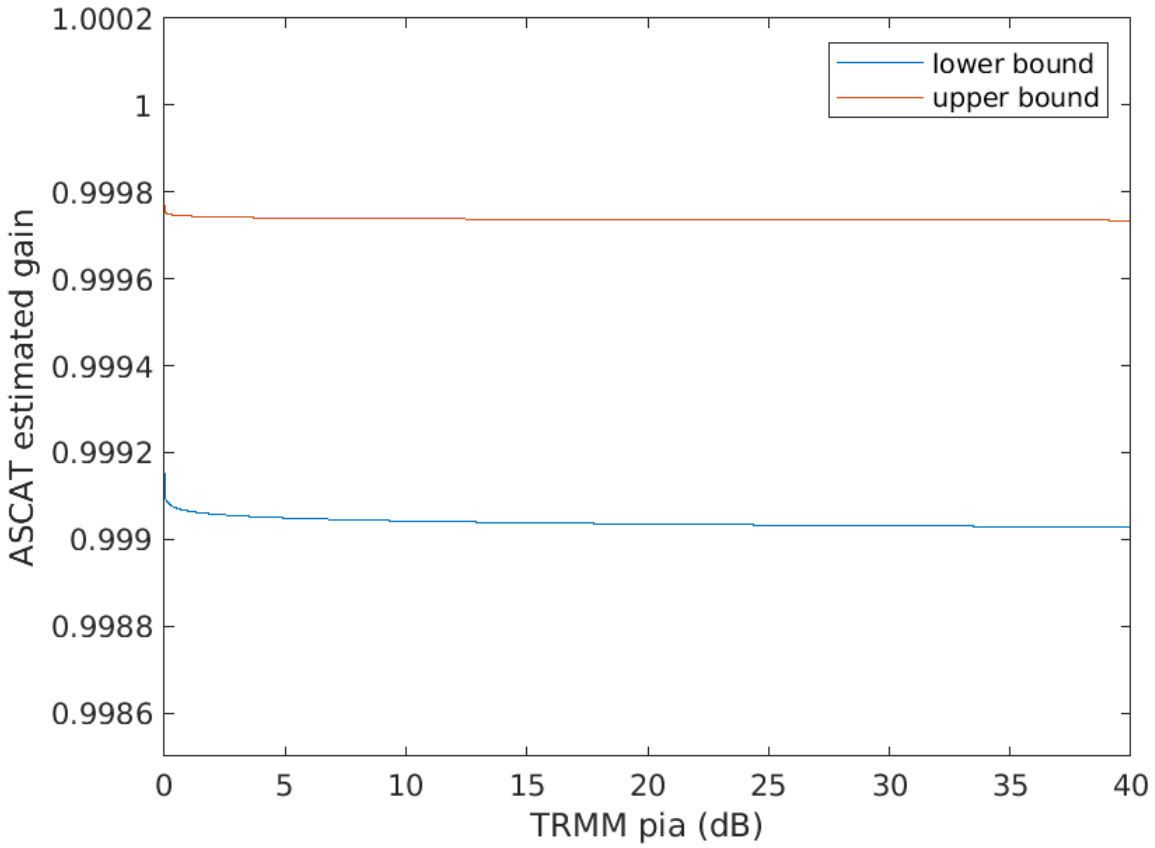


Figure 5.3: Estimated gain coefficient for ASCAT. The red and blue lines represent the boundaries of the gain PDF under the specific incidence angle conditions. This figure demonstrates that gain is only marginally affected by randomness in the freezing height. This is calculated using fixed incidence angles of 0° and 45° for TRMM and ASCAT respectively.

noise floor of dry σ^0 values, the gain term can thus be approximated as one. This coincides with the insight from [13] that rain surface perturbations are the dominating factor of rain's effects on the backscatter at C-band.

5.4 ERA Conclusion

The ERA is found to retrieve wind estimates better than the WO and CSRR models. This could be due to the weaker RV effects caused by freezing height, or because TRMM's PIA estimate is more accurate than the ITU equations under CSRR assumptions.

C-band attenuation due to rain is shown to be smaller than the noise in the σ^0 signal. The change this attenuation causes is so small that empirical tests show barely any change

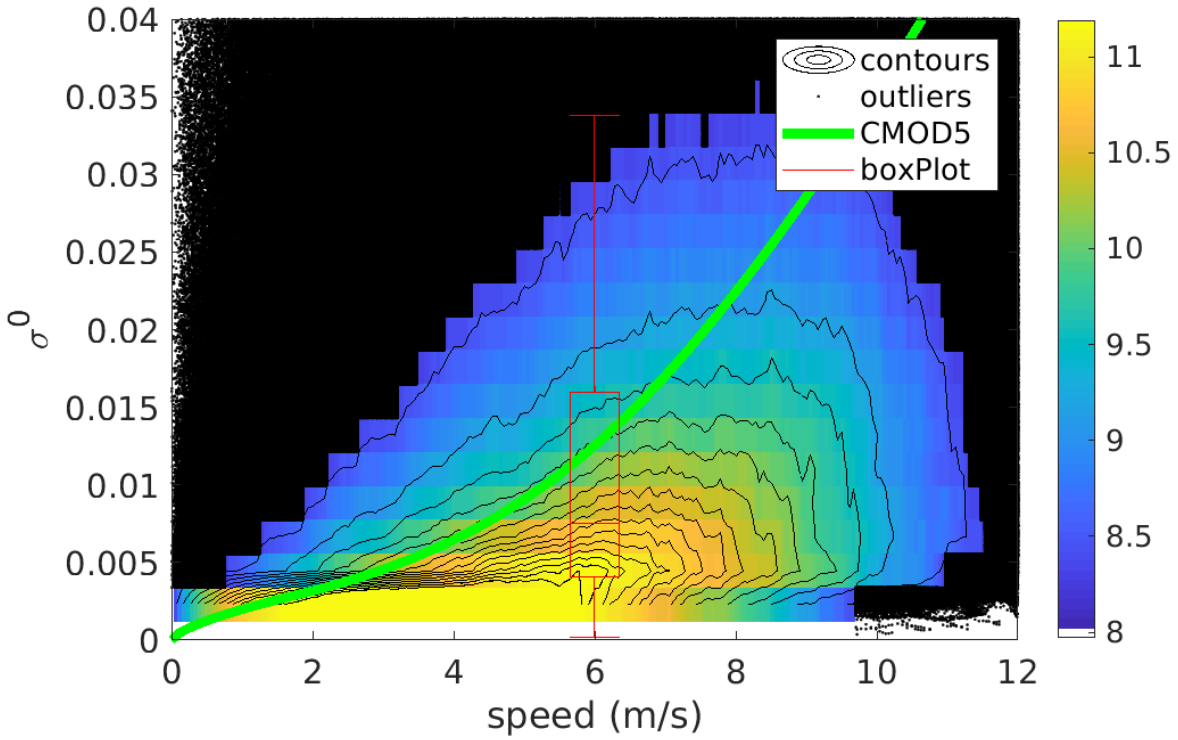


Figure 5.4: Heat and contour map of measured non-rainy σ^0 value at different wind speeds. The green line shows the the CMOD5 estimate of σ^0 at a fixed incidence angle of 47.7° and a relative direction of 0°. To demonstrate the distribution of a band of data, a box and whisker chart is shown for wind speeds between 5.9 and 6.1 m/s. This figure shows the variance of dry measured σ^0 values and how these samples vary from the CMOD5 estimate.

in performance when gain is assumed to always equal one, this suggests that standard model from Eq. 9.5 can be rewritten as,

$$\sigma^0 = \sigma_w + \sigma_e, \tag{5.7}$$

where σ_e is redefined as $\sigma_{sr} + \sigma_r$ as used in Eq. 9.5. Possibly, this could be simplified even further to

$$\sigma^0 = \sigma_w + \sigma_{sr}. \tag{5.8}$$

If gain is one, meaning the signal is not lessened at all as it travels through rain, than there would be no amount of signal radiating back towards the sensor off of the air-borne rain. If this is true, than σ_r is zero, and σ_e is directly equal to the increased backscatter caused by the rain roughening the ocean surface, σ_{sr}. In conclusion, the ERA GMF is a better

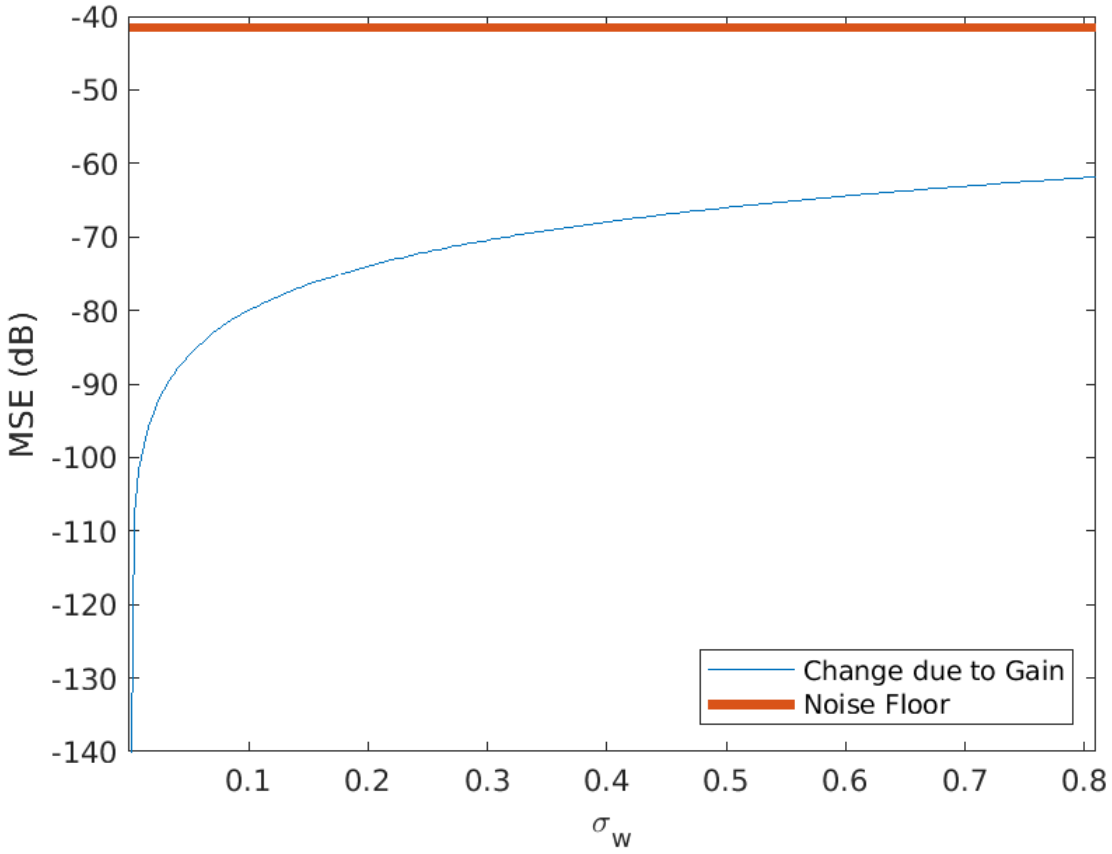


Figure 5.5: Noise floor versus the most extreme change in σ_w caused by the gain term. This change is expressed as the MSE between σ_w and $\alpha\sigma_w$. This plot shows that regardless of the size of σ_w's size, all changes caused by the gain term fall far below the noise floor. Thus, the gain term is deemed inconsequential, along with the randomness caused the RV freezing height.

modeling function than CSRR and is able to estimate wind speeds more accurately than the WO GMF in the presence of rain.

CHAPTER 6. RAIN GMF 3: EFFECTIVE DISTANCE

This chapter considers another rain GMF modelling approach. The equations used to solve for α under the ERA method can be solved without the use of freezing height. This is desirable because the freezing height behaves like a random variable adding noise to the GMF. However, solving for ASCAT PIA without using the freezing height leads to some peculiar behaviors in α that do not match the physical understanding of the term. While it is expected that lower rains result in lower PIA values, some higher PIA values are observed at lower rain rates. This may be occurring because of increased noise at lower rain rates. The effective distance (ED) method developed here, like ERA, estimates the gain to be near one at all times and applies the same fitting method. Thus the empirical results are essentially the same as discovered in Chapter 5.

6.1 Solving for Rain Induced Attenuation

The effective distance method uses the ITU equation to estimate rain attenuation given a frequency, rain rate, and incidence angle. The reported surface rain rate is assumed to be vertically constant from the ocean surface to a certain unspecified distance. The ITU equation for TRMM can be rewritten to solve for an effective distance the signal travels through the rain layer as

$$D_{eff} = \frac{P_T}{(k_T R)^{a_T}}, \tag{6.1}$$

where P_T is the PIA for TRMM, k_T and a_T are coefficients specified by TRMM's incidence angle and frequency, R is the surface rain rate, and D_{eff} is the effective distance.

This effective distance can than be inserted into the ITU equation for ASCAT to solve for ASCAT's PIA as

$$P_A = \frac{P_T}{(k_T R)^{a_T}} (k_A R)^{a_A}. \tag{6.2}$$

This method of computing PIA is referred to as the "effective distance GMF" because the D_{eff} term describes the effective freezing height expected under ITU conditions. The effective distance term is defined because it helps us understand some of the peculiar aspects of α's distribution.

6.2 Peculiarities of α

In the ED method the gain term α and its inverse, the PIA, have peculiar relationships to increasing levels of rain. As with all previous methods, PIA is low for low rain rates. As rain increases there is a clear increase in PIA. The same trend is seen in α, but only for rain rates above 1 mm/h, see Fig. 6.1

While this behavior is not explored here, I point out that it is similar to the behavior seen with effective distance. In Fig. 6.2 the heat and contour map for effective distance and rain shows the same decreasing-then-increasing trend centered a little above 1 mm/h rain. Effective distance uses ITU equations to undo TRMM's more complex PIA estimate. It is possible that subtle differences between theses two equations, along with increased noise at lower rain rates, combine to create a higher PIA at low rain rates.

6.3 Empirical Results

Using the same fitting techniques as used in Chapter 5, it is observed that the ED scores are nearly identical to the ERA scores. Automatically setting the gain term to be 1 results in nearly the same cumulative MSE for both the ERA and ED methods. This suggests that the α term for both these methods, is so close to 1 it does not matter how rain rate and incidence angle are mapped to α, because the effect is so small.

6.4 Stochastic Inferences

Because this approach uses no RV terms (see Section 2.4), there is no explicit noise model for the system even though errors may be present in the measurements and model inputs.

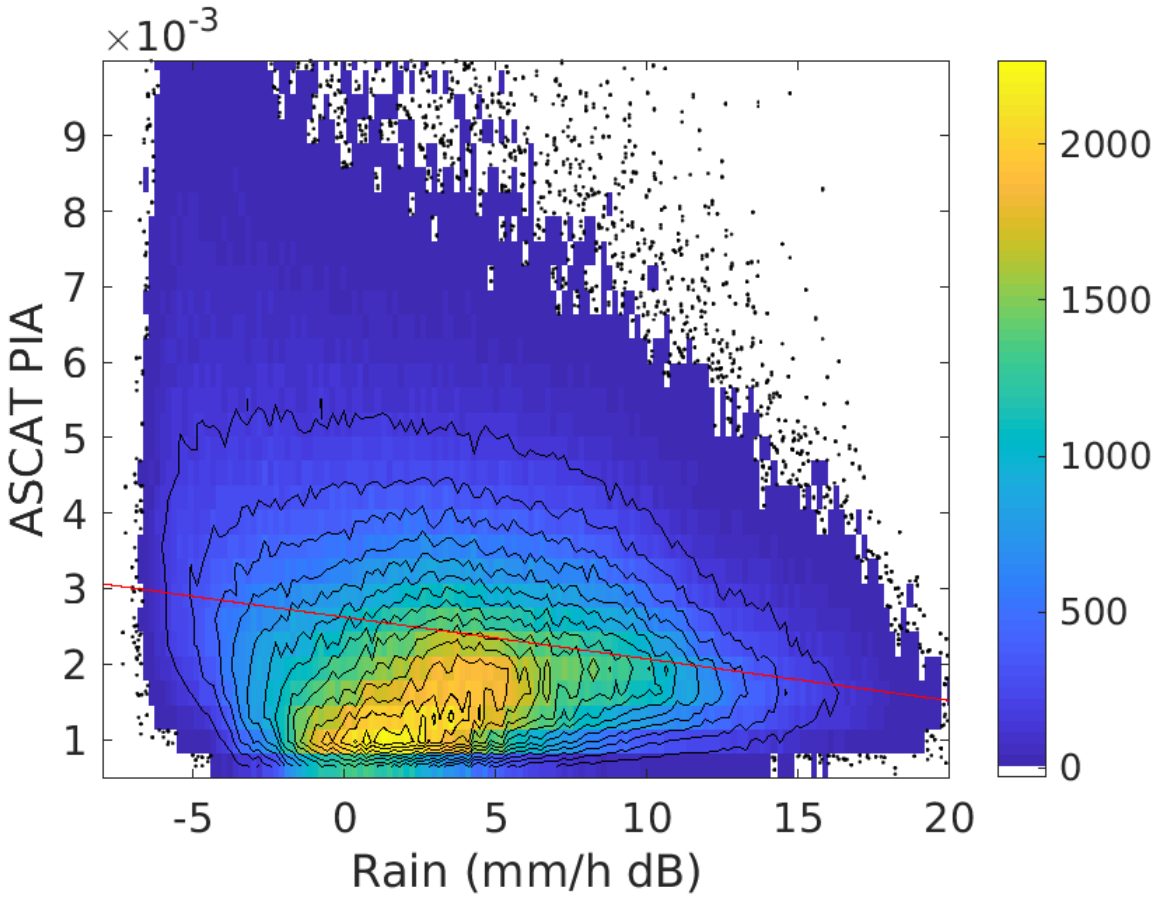


Figure 6.1: Heat and contour map demonstrating the relationship between PIA and rain rate. A red line indicates the first order polynomial fitting of the data. PIA decreases between -5 and 0 dB rain, and than increases after 0dB. Because increasing rain rates usually correlate to increasing PIA, it is expected that PIA continuously increases with increasing rain rates. This is not reflected by the data.

6.5 ED Conclusion

The ED GMF performs almost identically to the ERA GMF. Findings with the ED and ERA GMF suggest that the α term is unneeded, and that Eq. 2.6 can be simplified to Eq. 5.8. As with the ERA GMF, the ED GMF successfully reduced overall wind speed RMSE by 6.95% in the presence of rain.

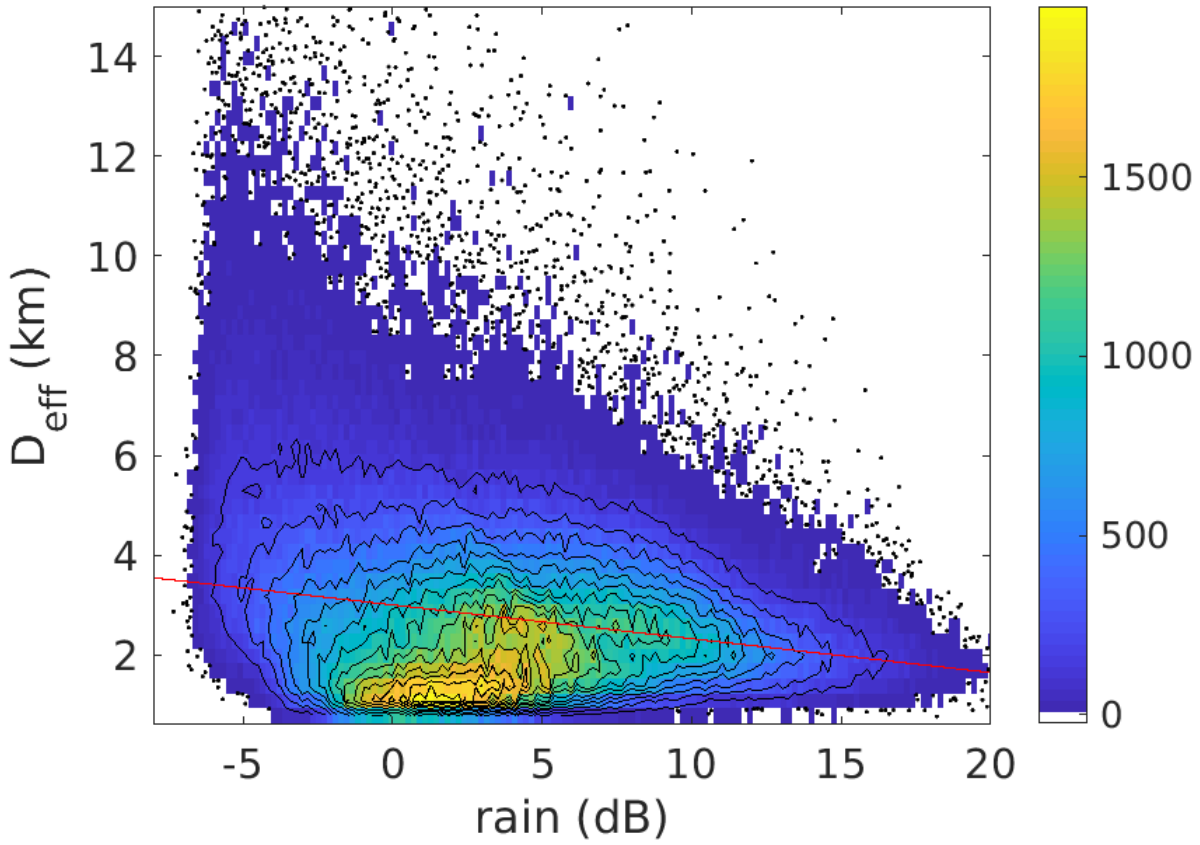


Figure 6.2: Heat and contour map demonstrating the relationship between effective distance and rain rate. A first order polynomial fitting of the data is shown as a red line. The relationship between rain and effective distance closely matches the relationship seen in Fig. 6.1. This may suggest that the peculiarities seen in Fig. 6.1 is largely due to the behavior of effective distance.

CHAPTER 7. RAIN GMF 4: DATA-DERIVED WO BACKSCATTER

The Data-Derived WO Backscatter (DDWB) GMF method developed in this chapter shifts the focus from the α gain term, which has been shown to be arbitrarily close to one, towards solving for the σ_w term (see Eq. 2.6) which has up until now been estimated using the CMOD5 function. A new fitting method is developed which is more accurate than previous methods. The wind retrieval performance over areas of rain is significantly improved.

7.1 Solving for Wind Induced Backscatter

The DDWB method solves for σ_w by collecting the rain free, or WO, samples of collocated data and binning them according to wind speed and incidence angle. The measured radar backscatter is defined as σ_w because gain and σ_e are presumed to be 1 and 0 respectively when there is no rain present. By averaging the binned backscatter values, a look up table ordered by speed and incidence angle is created that replaces the CMOD5 function.

Because CMOD5 is a well tested algorithm, the DDWB σ_w is expected to be very similar to the one estimated by CMOD5. With the exception of very low wind speeds, CMOD5 and DDWB estimates of σ_w are similar as shown in Fig. 7.1. CMOD5 and DDWB have mixed results with empirical testing. In some cases, the CMOD5 version of the data outperforms the DDWB, and in others the DDWB version is superior. This will be discussed further in the context of the new fitting method in Section 7.2.

7.2 New Fitting Method for the Basic Rain GMF Coefficients

During this stage of research, a more accurate method was discovered for fitting the basic rain GMF coefficients in Eq. 2.6. Instead of first filtering the data and then fitting a curve or surface to the output, this method models σ_e and α as a functions of rain rate and uses an unrestricted, nonlinear Nelder-Mead minimization over all available rainy data to

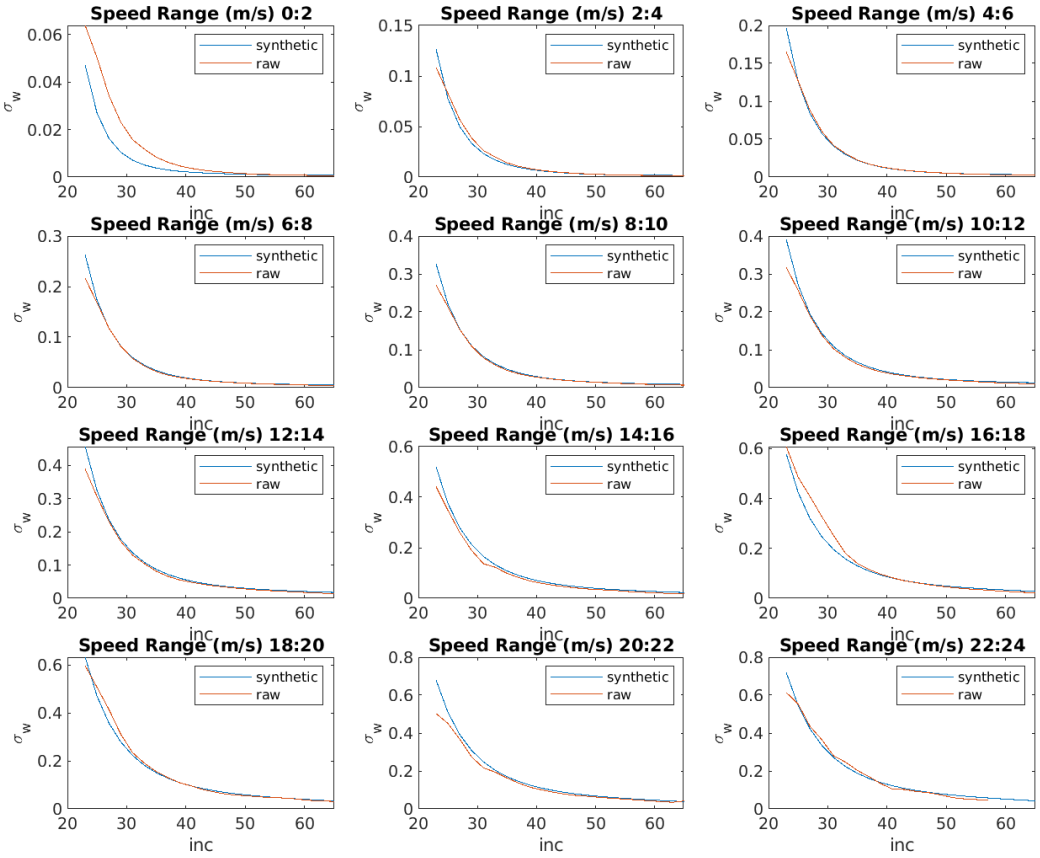


Figure 7.1: σ_w estimated by raw (DDWB) and synthetic (CMOD5) methods. This is displayed as a function across incidence angle and speed. Notice that synthetic and raw data are very similar except in the case of low wind speeds.

estimate the parameters which model α and σ_e from Eq. 2.6. In Matlab, this minimization is done using the `fminsearch()` function [14]. The search is done separately for each incidence region, and then the resulting parameter values are modeled by a simple linear fit across incidence angle. This fitting of the coefficients across the incidence angle allows the DDWB GMF to be smooth across rain rate and incidence angle.

I first test the minimization fitting using CMOD5 to model σ_w. This enables the new fitting method to be compared with previous methods before introducing a new element that can affect performance. After the CMOD5 version is tested, the DDWB version is then tested and compared.

Selecting the $\alpha(R)$ and $\sigma_e(R)$ Equations

The loss score equation minimized by `fminsearch()` has a significant impact on the retrieval result. I model the loss score (S) as the mean sum of squared error (MSSE) as

$$S = \frac{1}{N}\sum\{\sigma_{raw} - [\alpha(R)\sigma_w + \sigma_e(R)]\}^2, \tag{7.1}$$

where σ_{raw} is the sampled σ^0 value, and σ_w is the wind only backscatter which is derived from either CMOD5 or DDWB. N is the total number of rainy samples used in this fitting.

I suspected that the model which could achieve the lowest MSSE would also produce the best wind retrieval product. This proved to not always be strictly true, but this principle guides the initial step of selecting several model equations.

The relationship between σ_e and rain rate , assuming gain is equal to one, appears to be a slow growing exponential, as seen in Fig. 7.2. As such, the equations explored model an exponential growth. Many equations are tested to model $\alpha(R)$ and $\sigma_e(R)$. Most proved to be ineffective, Table 7.1 shows the equations which created effective rain GMFs.

Table 7.1: Equations tested to model α and σ_e. E1 is the same as E2 except that E1 uses the sum of errors squared minimization function, (Eq. 7.2). E2 through E5 use the sum of squared error minimization function, (Eq. 7.1).

Equation ID	$\alpha(R)$	$\sigma_e(R)$
E1	R^{-x_1}	$R^{x_2} - 1$
E2	R^{-x_1}	$R^{x_2} - 1$
E3	1	$R^{x_2} - 1$
E4	1	$x_1 R + x_2$
E5	1	xR

The new method of fitting is 7.8% to 11% more accurate at modeling σ_e than the most accurate fitting method previously used, the Epanechnikov and lowess filters. The models fit using `fminsearch()` resulted in a MSSE scores between 7.68e-4 to 7.97e-4. The method of using Epanechnikov and lowess filters, as done in Chapters 4 through 6, results in a MSSE of 8.66e-4. It is expected that greater accuracy in the modeling should result in more accurate wind retrieval.

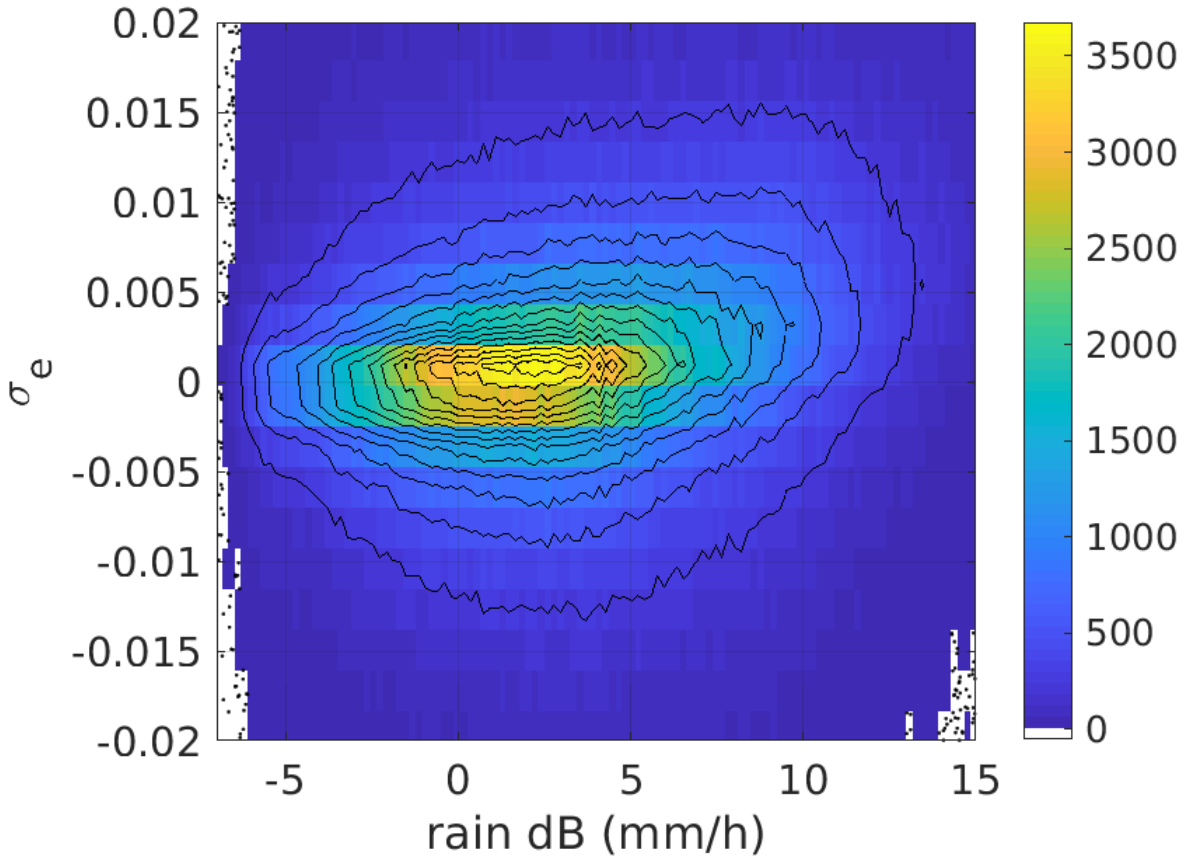


Figure 7.2: $\sigma_e(R)$ across rain rate assuming $\alpha(R) = 1$

This expectation is not realised because the rain model with the best wind retrieval performance is E1 which has the highest SSE. Interestingly, the E1 fit of $\alpha(R)$ and $\sigma_w(R)$, shown in Fig. 7.3 was affected by a coding bug. This bug was discovered part way through the testing. Nevertheless the resulting GMF scored a wind speed RMSE of 2.61 m/s and so is included here.

The bug was a misplaced parenthesis in the MSSE calculation. The error was first summed across all samples, and then squared, allowing positive and negative errors to cancel each other out,

$$S = \frac{1}{N}\left[\sum\{\sigma_{raw} - [\alpha(R)\sigma_w + \sigma_e(R)]\}\right]^2, \tag{7.2}$$

E1 gave the best performing rain GMF and showed that the expected relationship between MSSE and wind retrieval performance does not hold. When the bug was corrected

for E2, the MSSE dropped from 7.97e-4 to 7.71e-4 but the cumulative wind speed error rose from 2.61 m/s to 3.35 m/s.

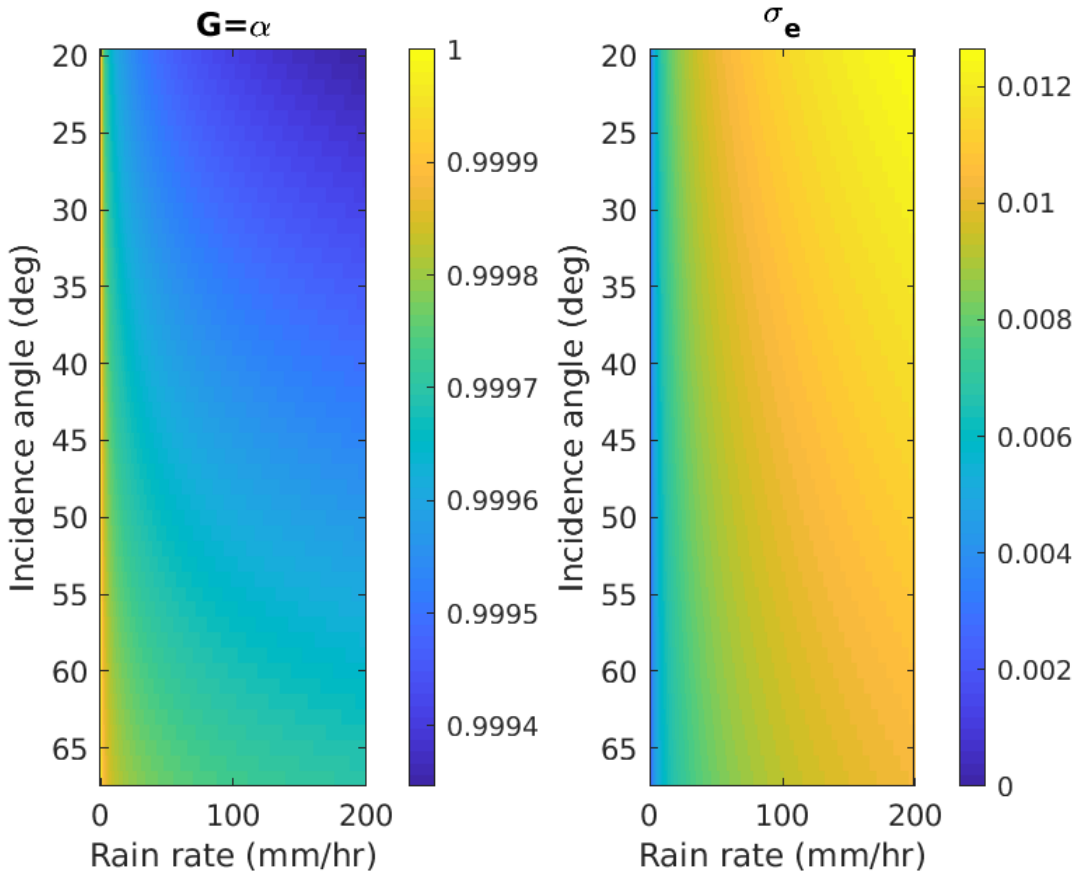


Figure 7.3: Rain GMF modeling α and σ_e terms as functions of rain rate and incidence angle. This particular GMF performed wind retrieval more accurately than all other DDWB GMFs and scored a wind speed RMSE of 2.61 m/s. However, this particular GMF had an error in the minimizing equation which used the squared sum of errors instead of the sum of squared errors.

This behavior, though complex, is possible because the SWR retrieval algorithm applies a non-linear maximum likelihood estimate. This means that just because a GMF fits the best going forward (weather conditions to σ^0) it does always fit the best going backwards (σ^0 to weather conditions).

A potential reason why E2 performed worse than E1 might be understood by comparing the GMF values. Fig. 7.4 shows the E2 GMF created without the bug. Notice that

Fig. 7.4 demonstrates a significantly larger dynamic range for α and σ_e than Fig. 7.3. It is possible that the E2 GMF is fitting the data to powerfully, meaning the counteracting effects of α and σ_e are larger than necessary. This results in noise being magnified. A similar issue is discussed in Section 9.3.2.

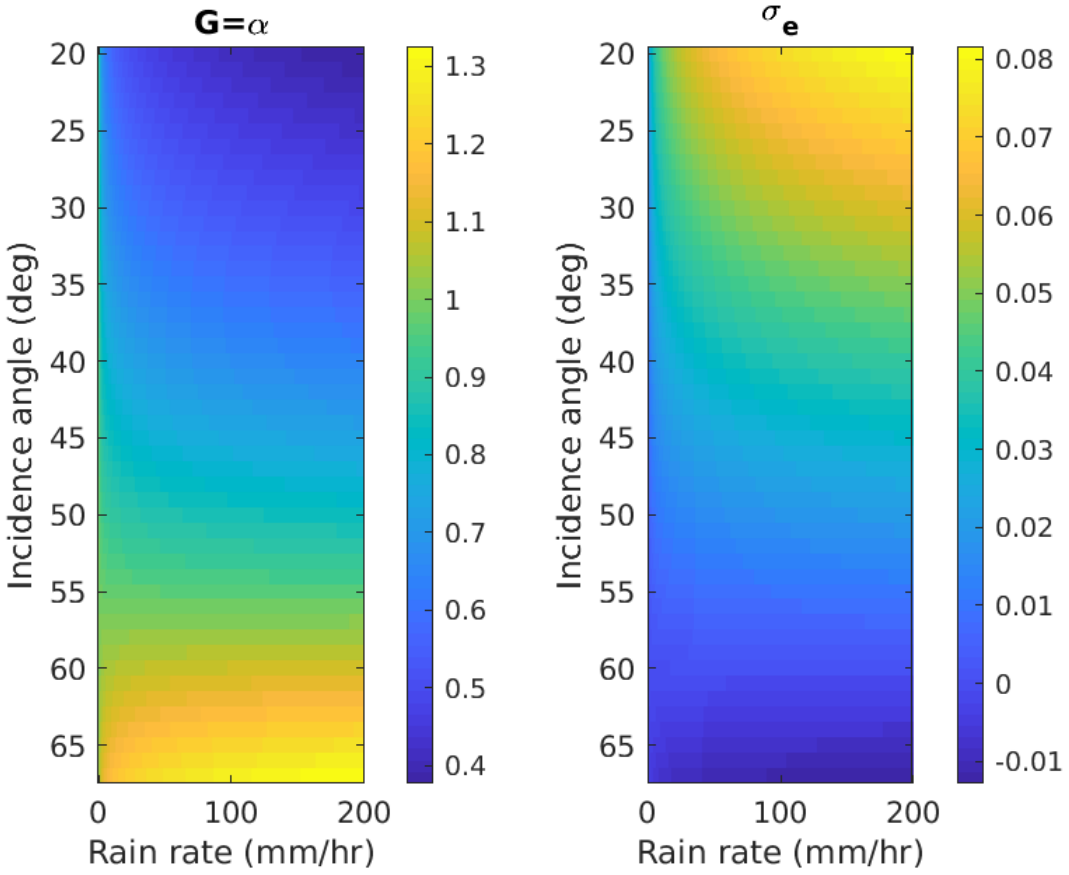


Figure 7.4: Rain GMF modeling α and σ_e terms as functions of rain rate and incidence angle. This particular GMF scored wind speed RMSE of 3.35 m/s. This GMF is the bug-corrected form of Fig. 7.3.

The other rain GMFs demonstrate a similar pattern where the worse performing GMFs had larger dynamic ranges. The other equations, in order of dynamic range, are: E3, E4, E5. Their dynamic ranges are 0.04, 0.6, 0.8. These equations have wind retrievals errors of 2.75, 2.92 and 2.87 m/s respectively. These same functions fit over slightly different circumstances result in widely varying GMFs which support this trend of increasing wind speed estimation error with larger dynamic ranges in the rain GMF. It is noticed, however,

that the E1 GMF has a dynamic range of 0.12 for σ_e which is larger than the 0.04 range of E3. Clearly, dynamic range does not solely govern performance.

7.3 Empirical Results

7.3.1 Wind Estimation

This section discusses the wind estimation performance for two distinct cases. First we discuss the DDWB case where σ_w is solved using dry test samples. The second case is where σ_w is solved using CMOD5. The CMOD5 method is how σ_w has been solved for all previous rain GMF's. Both the DDWB and CMOD5 methods achieve significant improvement to the wind RMSE.

In general, the rain GMFs which consistently performed the best for DDWB are E1 and E3. E1 and E3's best RMSE scores using DDWB are 2.61 and 2.62, respectively. These scores are highly sensitive to the methodology of the minimization fitting. For example, E1 and E3 score best only when the rain value is scaled up by 5; otherwise, E1 and E3 score 2.94 and 2.75, respectively. E1 and E3 also require the incidence fitting to be done using only incidence angles greater than 30°.

The CMOD5 performance was likewise sensitive. The best score for CMOD5 type data was 2.63, using a form of E1 where the coefficients are only fit by the data points with incidence angles higher than 63°. When CMOD5 and DDWB types of data were tested under like conditions, there appeared to be no definite pattern as to which data set was superior. Similar trends between equation types occurred for DDWB data as well as CMOD5, but small changes in the fitting method would improve the performance of one and diminish the performance of another.

It was noticed that slight adjustments to the fitting of coefficients across the incidence angles resulted in significant fluctuations in the score. Likely, the slight difference in CMOD5 and DDWB data resulted in slightly different fittings which caused the peculiar performance preferences.

Although the fittings were highly sensitive to small changes in the methodology, the new fitting function and the DDWB resulted in significant improvements to wind retrieval performance, see Fig. 7.5.

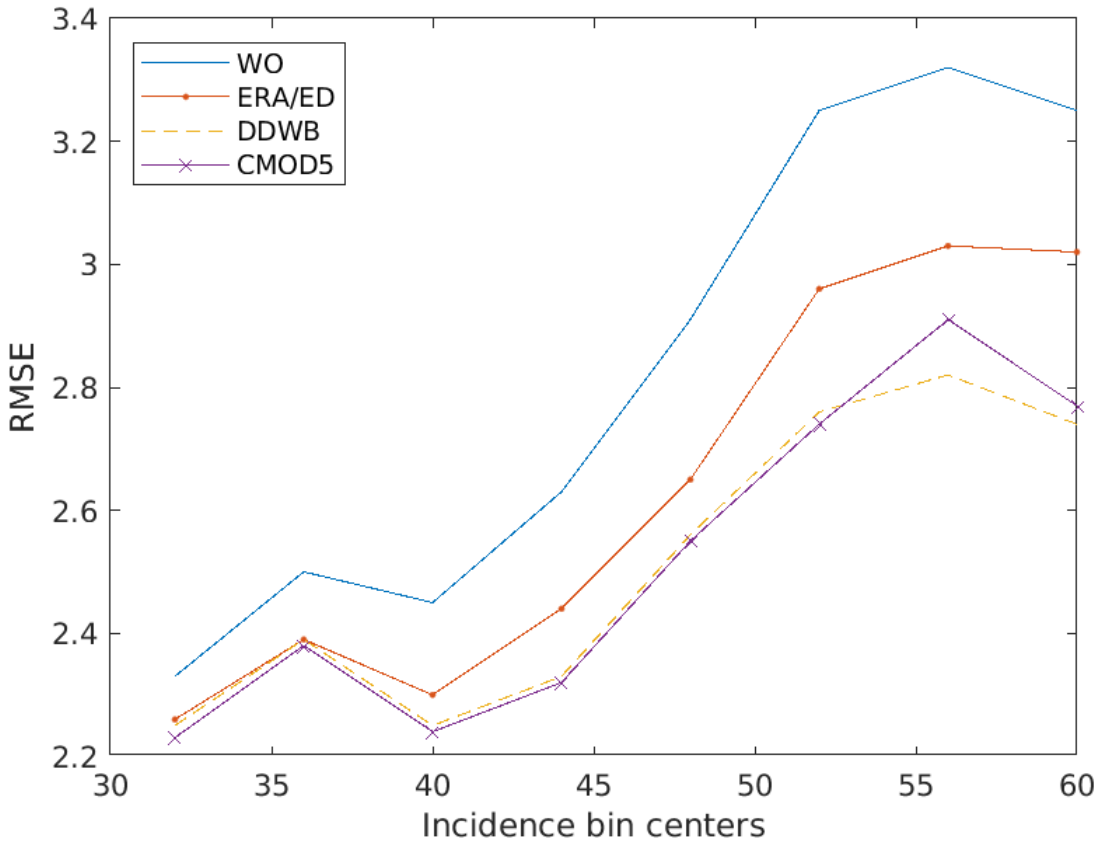


Figure 7.5: RMSE between ECMWF and retrieved wind speeds in the presence of rain for several different rain GMFs and the WO GMF. The ERA and ED rain GMFs outperform the WO GMF in the presence of rain for any incidence angle. Likewise, the DDWB and CMOD5 GMFs created in Chapter 7 outperform ERA and ED.

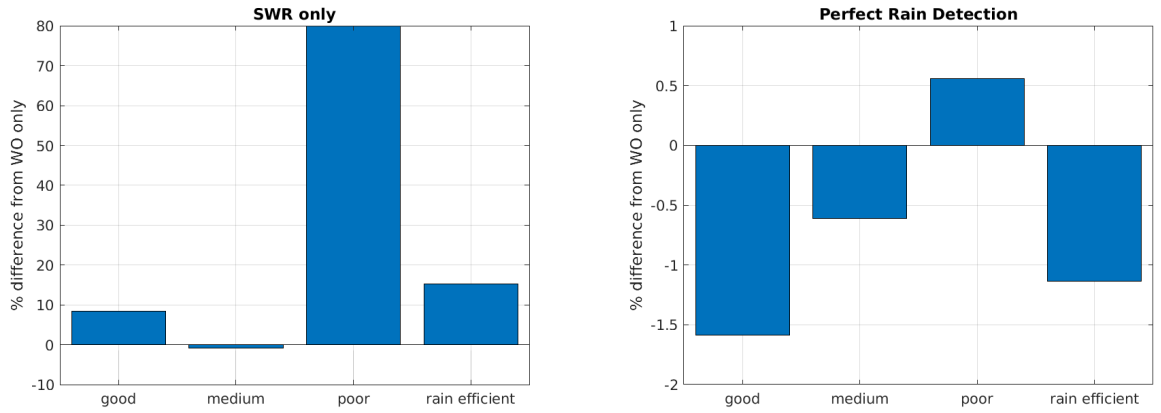
7.3.2 Rain Estimation

The average rain RMSE for the previous GMFs discussed was around 19 mm/h. For most of the GMFs created using the minimization technique, the rain RMSE was above 30 mm/h. However, a few rain GMFs were outliers. One in particular was able to score a 7.9

RMSE for rain, and a wind RMSE of 2.86. Although a RMSE of 7.9 mm/h is far from ideal, it is low enough that this particular GMF might be desirable for rain estimation.

7.3.3 Performance Using Dry Samples

Because a perfect rain classifier is unreasonable to expect, I investigate the performance of a few of these GMFs on both dry and rainy samples to see how the total wind error is affected. Fig. 7.6 shows the estimated range-of-performance for 4 different types of rain GMFs developed using the `fminsearch()` fitting. These are distinguished as good, medium, poor, and rain-efficient. The good GMF gave the minimum wind speed error (2.61). Medium (2.81) and poor (3.15) also also reflect the wind speed performance compared to the WO case which scores a RMSE of 3.02. Rain-efficient is the GMF which had the lowest rain estimation error and scored a wind RMSE of 2.86.



(a) (b)

Figure 7.6: Percent difference of wind speed RMSE for the SWR and WO systems. The figure on the left uses a SWR system where all data samples are classified as rainy and processed using SWR. The figure on the right shows a SWR system where the rain detection was done perfectly and SWR is only used on samples containing rain. These two extremes show the range of performance we can expect for the four rain models presented.

The y axis for both figures shows the % difference from the WO case, which is when rain is ignored. The figure on the left shows the percent difference from the WO case when the SWR GMF is used on all samples. This behavior is the worse-case scenario for the rain

detector and "should" achieve the highest error. The figure on the right shows the change when the rain detector is ideal. Any rain detection system should achieve a total RMSE error between the value on the left and on the right. The medium rain GMF, however, does not follow this trend, but it still provides a useful framework for decision making.

The good rain GMF improves the total wind speed estimation by 1.5% given a perfect rain detector. However, a less idea detector can result in a total error 8.4% above the WO case. If the rain detector is not effective, the medium performance rain GMF is a desirable option. Even with no rain detector and every sample processed by SWR, the total error is reduced by 0.8%. However, with the reduced risk comes reduced reward as the medium rain GMF does not score as well the good for a perfect rain detector.

The poor GMF performs as expected. Rain-efficient, like the good rain GMF, also has high risk and high reward. This type of rain GMF can lower total error by 1.1%, but if the rain detector is poor the total error may be increased by up to 15%.

In summary, a particular rain GMF can be selected according to the confidence in the rain detection systems performance. Specifically, I suggest choosing between the good rain GMF, which is optimal for an accurate rain detection system, and the medium rain GMF, which performs well regardless of the detectors performance. Finally, if the user desires rain information, there is a rain-efficient GMF which produces the best rain estimates and still gives good wind speed estimates.

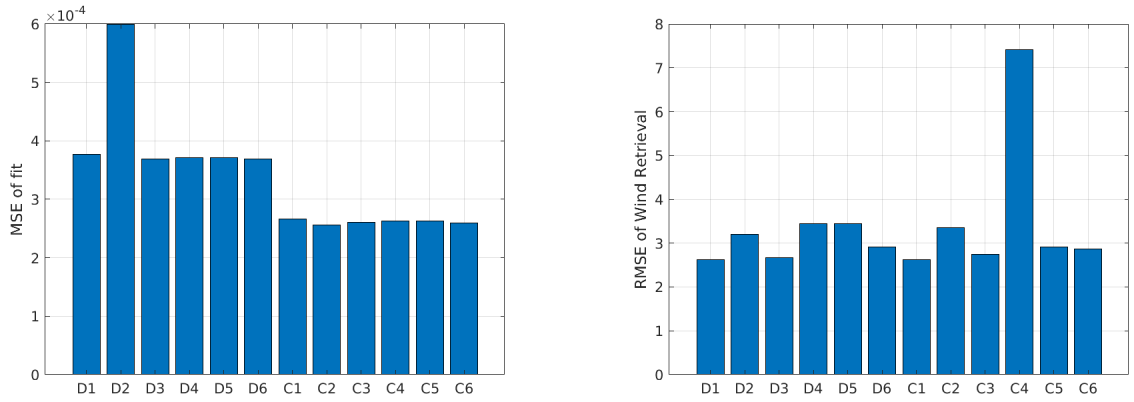
7.4 Stochastic Inferences

As with the ED rain GMF, the DDWB rain GMF has no noise model because it uses no random variables to solve for σ_e and α (see Section 2.4). It is interesting, however, to notice that the assumption concerning the equation type used to model σ_e and σ was shown to be false. I assumed that whichever equation could best fit the data would result in the best retrieval scores. Fig. 7.7 shows the MSE of the fit next to the RMSE of the wind retrieval.

In part (a) of Fig. 7.7, there is a clear distinction between models made using DDWB data, which are numbered D1 through D6 on the left, and those made using CMOD5 data, numbered C1 to C6 on the right. Models which were made using DDWB data fit the raw

data less accurately because the resolution of σ_w is binned into ranges of 2 m/s wind speeds and 2 degrees incidence angle. By comparison, the CMOD5 has a smooth fitting alongside speed and incidence angle. Even though DDWB constructs σ_w directly from the raw data, equations tend to fit less effectively over DDWB data than CMOD5 data.

Part (b) of Fig. 7.7 shows the RMSE of the wind retrieval for same models in part (a). It is interesting to notice that high MSE of fit does not correlate to a better RMSE of wind speed retrieval.



(a) (b)

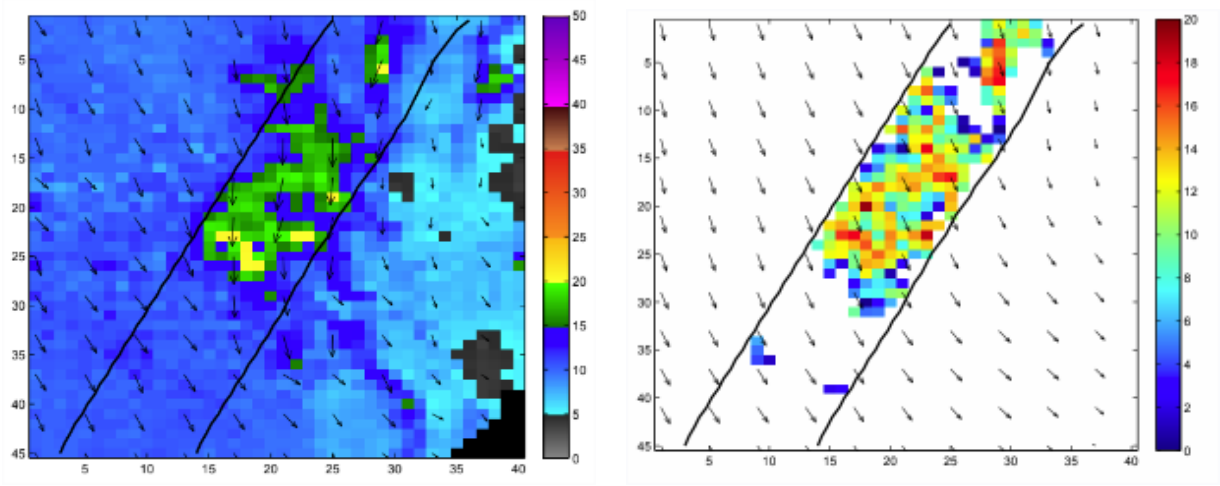
Figure 7.7: The figure on the left (a) shows the MSE of the rain models compared to the observed backscatter return. The figure on the right shows the wind RMSE for each rain model. Those rain models which used DDWB to model σ_w are shown to have much higher MSE values (labeled as D1 through D6). Rain models which used CMOD5 (labeled C1 through C6) have a cleaner fit to the measured backscatter data and thus have a lower MSE of fit. Even though DDWB rain models fit "worse", the resulting RMSE of wind retrieval presented in the figure on the right, shows that DDWB and CMOD5 have similar retrieval performance values.

7.5 DDWB Conclusion

Even though the DDWB methodology does not significantly improved the rain GMF, the minimization method of fitting σ_e and α to the raw data has. Fig. 7.5 demonstrates the performance improvement that the new fitting method creates. Both the original CMOD5

method and the DDWB method for modeling σ_w are shown. It is unclear if DDWB data can model the σ_w term in a way that results in the better wind retrieval than using CMOD5.

Rain, as discussed earlier, reduces the accuracy of wind estimation by disrupting the capillary waves and usually increasing backscatter. This gives the false appearance of higher wind speeds in regions where rain is falling over the ocean. This can be seen in Fig. 8.1 which shows estimated wind speed along side a map of rain. This increase in σ^0 is why SWR wind retrieval should be used over areas of rain contamination to reduce wind estimation error.



(a) (b)

Figure 8.1: Illustration of rain influence in wind retrieval. (a) Retrieved wind speeds using the WO GMF in knots. There is a peculiar spike in wind speed is peculiarly out of place. The black line represents the TRMM swath. (b) Rain rates in mm/h as reported by TRMM. High rain rates correlate with higher retrieved wind speeds. These figures come from [4]

The challenge for scatterometery is to know where to use specialized rain GMFs when rain is present. Without this knowledge, SWR wind retrieval is impractical for wind retrieval because of it's poor performance over dry samples. The current practice is to use the WO

GMF and treat rain as a contamination over approximately 2-6% of the ocean. If possible, the rain-contaminated winds are flagged and discarded.

By properly applying the SWR GMF in rainy regions where it outperforms the WO GMF, the total wind estimation error can be lowered and rainy samples preserved. Previous research done by [4] created a Bayesian estimating system which used various inputs to estimate which GMF would produce the lowest error without knowing rain conditions before hand. My research attempts a similar task, but instead by using machine learning tools, specifically neural nets (NNs), to preclassify conditions as raining or non-raining.

NNs are algorithms trained to utilize details in information without being explicitly told what those details are. For example, a well made NN can identify images of cats and dogs without being explicitly told what features are "catlike" or "doglike". This is accomplished by using a large set of labeled samples to "train" a collection of weights and biases in the NN. Training iteratively adjusts the weights and biases to step the prediction error towards the minimum. In this respect, a rain classification NN is trained and attempts to use traits in scatterometery data to identify the locations of rain. Once rain locations are isolated, they can be processed using the GMF retrieval, which results in minimum error.

This chapter introduces NNs, their structure, and the basic concepts of overfitting and underfitting.

8.1 Machine Learning and Neural Networks

Machine learning (ML) is the study of algorithms which can learn from experience [15]. While an algorithm or a computer is not a living being, the term "learning" is used because ML mimics the natural processes of learning which uses repetition and feedback that guides a learner toward a desirable outcome. The basic format for most ML techniques is to use sample data fed through a model to produce an output. Then comparing that output with the desired output, a function adjusts the decision making model in a way that lowers the error in a processes called backpropagation. This updating and adjusting processes is called training.

This research focuses on a specific type of ML called a neural network (NN). Just like in the brain, where tiny neurons are activated in response to neighboring neurons being

activated, a neural network is composed of many "neurons", or nodes, which interact to create a model.

The limitless number of configurations for these nodes and how they interact with each other, make NNs extremely powerful but challenging to work with. Powerful, because given enough nodes, they can, in theory, mimic any possible function without being told explicitly what the function is [15]. NNs can also make complex decisions quickly, making them valuable tools in real time computing.

NNs can also be challenging to work with because of their flexibility. NNs are often referred to as "black boxes" because the way the data is handled inside the NN can be so complex and interconnected that observing the actual parameters inside a NN often tell you nothing about its performance. Only by observing the data handled by the NN, can one tell how well the NN is actually working. Building and improving a NN is a process of trial and error which has no clearly defined methodology that is guaranteed to minimize the error, but is guided by some basic principles.

8.2 NN Structure

The structure of a NN is discussed using the example NN in Fig. 8.2 which takes in an image and categorizes the picture as a dog, fish or cat. The depiction is an over simplified net structure which probably would not perform this well. Its purpose is to introduce the key structural components of a NN.

The NN in this example is layered like a cake. The first layer is the input layer which takes in the images RGB pixel values. These values then move down to the next layer, following the solid black lines. In hidden layer 1, each node (the blue circles) receives inputs, adds it to a bias value and then weights the output by w_n. This output is then fed along to the connected nodes in hidden layers 2 and 3, and finally to the output layer which reports that the most likely label for the image is a dog. The NN in Fig. 8.2 is a simple feed forward NN because the information only moves forward from input nodes to output nodes. Also, the layer structure is called "fully connected" because every node in one hidden layer connects to all nodes in the next.

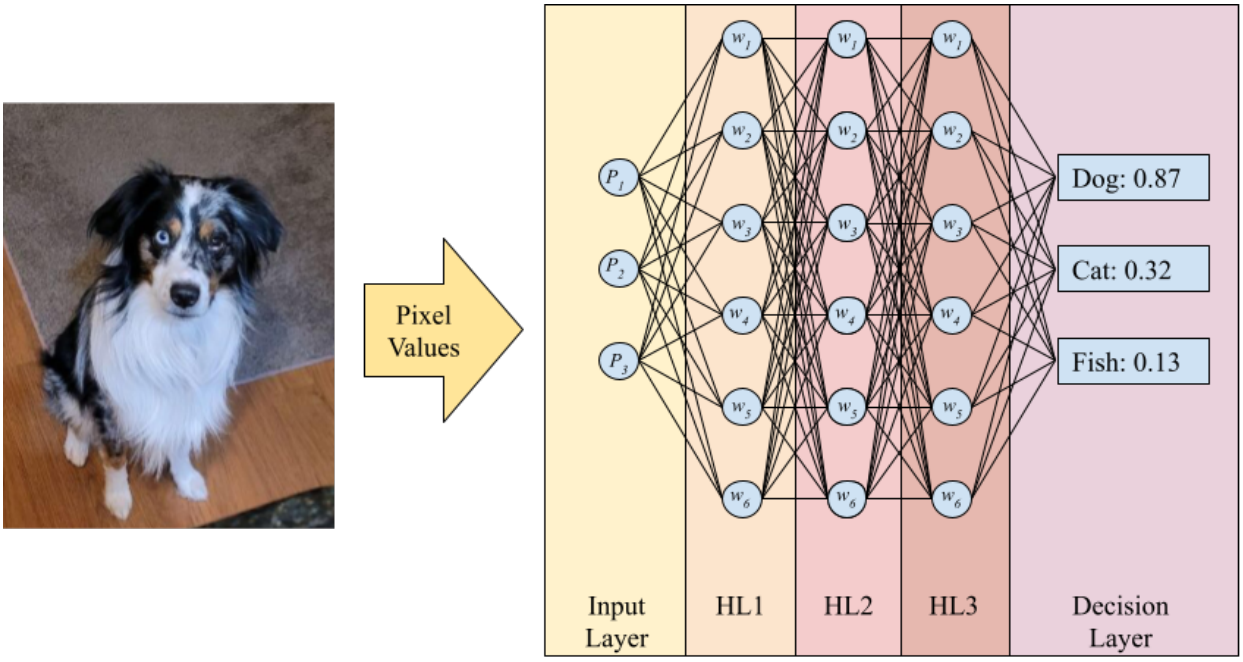


Figure 8.2: Example of a trained NN correctly categorizing an image of a dog. The input image is broken up and fed into the input layer as pixel values. These values are passed to the first hidden layer (HL1), each node (blue circle) sums the values being passed to it (indicated by the solid black line), next it adds this score to a tuned bias which is then multiplied by a tuned weighting. This pattern continues for 2 more hidden layers and culminates in the final decision layer. When all the nodes in one layer are connected to the following layer, it is called a fully connected layer. Another name for this type of net is a multilayered perceptron.

Fig. 8.2 shows a finished NN, but how is one created? This is accomplished by using many labeled sample images and updating the weights and biases using an optimization algorithm. Typically training starts with random Gaussian noise for the weights and biases. Naturally, this untrained NN performs terribly at first. A particular training algorithm uses a gradient descent approach, it observes how the output changes with small fluctuations to the weights and biases. The algorithm then selects the largest gradient descent and starts to "walk" the error down towards a low point. Given enough samples and a well designed NN architecture, the NN starts to "learn" and lowers the training error.

While manipulating the weights and biases to walk the error down towards a local minima is relatively straight forward and heavily supported by many libraries, controlling what the NN actually learns is a challenge. In the end, the NN needs to be able to operate

accurately on new data never seen by the NN before. How to control and guide the learning processes is discussed at greater length in Chapter 9.

8.3 Basic Issues Preventing Successful NNs

Overfitting and underfitting are broad ways to describe the performance of a NN. Striving to train a model which fits somewhere between these two descriptions usually yields the most desirable NN. This section describes what both overfitting and underfitting are, and how it can be observed.

8.3.1 Overfitting and Underfitting

Overfitting and underfitting can be more easily understood in the case of fitting a line to a group of points. If a tenth order polynomial line is fit to noisy data points that follow a linear curve, the fit tends to mimic the noise contained in the samples and not just the underlying linear curve. Although a tenth order polynomial may have a lower squared error between the line and the data points, a better fit is a simple line which describes the general trend of the data, not the noise. When an overly complex model is used to fit data which follows a simpler trend it is called overfitting. NNs can also overfit data. They can "learn" aspects of the training data set that do not apply generally, like noise. As such, the NN seems to train remarkably well, but when it encounters new data it performs poorly.

Underfitting is the opposite of overfitting. If the architecture of a NN is not complex enough for the problem it is training on, it is likely to not perform well enough. Underfitting is similar to fitting a parabolic grouping of data points with a straight line. No matter how much the slope or y-intercept of the line is adjusted, it never drops the error between the line and the data in a satisfactory manner.

Sometimes the network structure can be sufficiently complex but still be underfit. This happens when the NN finds a local minimum in the error score during training in the error score and gets stuck. In my case, this can be as simple as labeling every sample as non-rainy, which was a common failure mode found during preliminary trials because the NN

is correct 96% of the time if it assumes rain never happens. This is due to the infrequency (4%) of rain.

8.3.2 Observing Overfitting and Underfitting

How the NN performs on generic, everyday data it sees during its operation is the real test of a NN. How the NN performs on future unknown data is called the general performance. The true general performance of any NN is never fully realized because it would require using all the samples the NN is ever going to operate over in the future, which cannot be known. Instead the general performance is approximated by testing the NN on data which it has never seen during the training processes. It is assumed that future samples are reasonably similar to the "test" samples.

If the general performance (estimated by the test data) and the training performance are both poor, the model likely has underfitting issues, or is stuck in a local minima. One of the challenges in working with NNs is that there is not a great way to know when the performance has dropped to an optimal level. Where traditional estimation and detection algorithms have Cramer-Rao and other boundaries which let us know the borders of optimality, ML is trained and adjusted in an ad hoc manner. NNs are universal approximators [16], meaning that given enough resources and the correct structuring of the net, a NN can mimic any possible function. This suggests that at the very least, a NN can be expected to perform as well as a traditional estimator if trained properly and given enough resources. Unfortunately, in practice, the only way to know how well a NN is able to perform is to try to build one that performs that well.

Now for overfitting: if the training performance is good, but the general performance is much worse, the model may be overfitting the data. This means that the NN has learned characteristics of the data inherent to the training data, but which do not apply generally, such as noise. This is represented in Fig. 8.3, which shows general and training loss across a range of model complexities. Generalization Loss, or error, is the value which a well constrained NN is attempting to drive to zero. The best NN design minimizes the generalization loss.

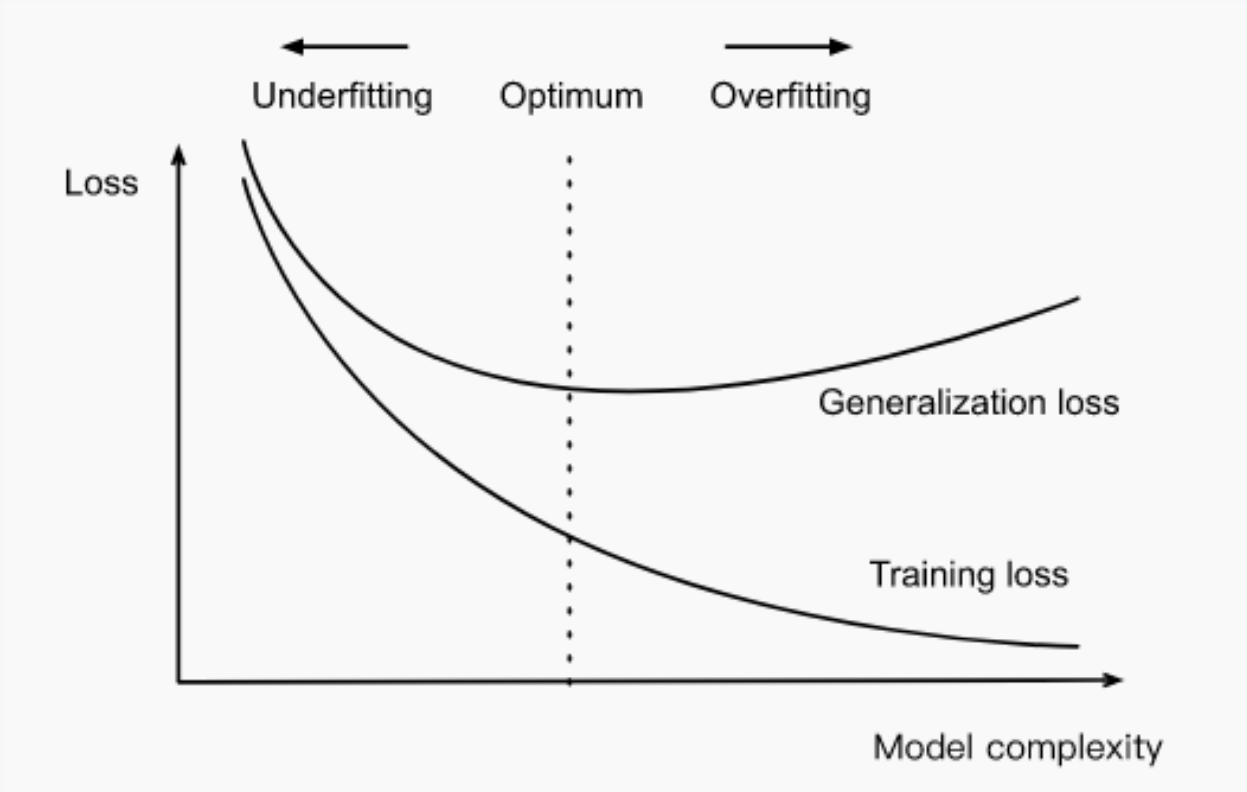


Figure 8.3: Visual representation of the effects of overfitting and underfitting on loss, also see Fig. 8.4 for further description. Underfitting is when the loss is high (poor performance) for both the general and training cases because the model is too simple. As the model becomes more complex, it can start to learn the general traits in the data which correlate to the desired outputs. When complexity goes too far the generalization loss actually starts to increase again. This overfitting implies that the model is starting to fit things like noise in the training data set, or that the training data set is not a good sampling of regular data which the net is going to estimate on in the future. Since the generalization loss cannot actually be known, it is estimated using the testing and validation data sets.

To obtain a more nuanced understanding of the NN's behavior, another small set of testing data is set aside called the validation set. This data set is fed through the NN periodically during the training process, but is not used to update the weights and biases. It is like taking a snapshot of how the testing data performs at different stages of the training process. When the NN is being trained, the loss approaches a minima. By plotting the loss of the training and validation data as in Fig. 8.4, one can see if the NN loss is dropping or not. This plot also helps determine how severe cases of underfitting or overfitting are relative to the loss decrease during the learning processes.

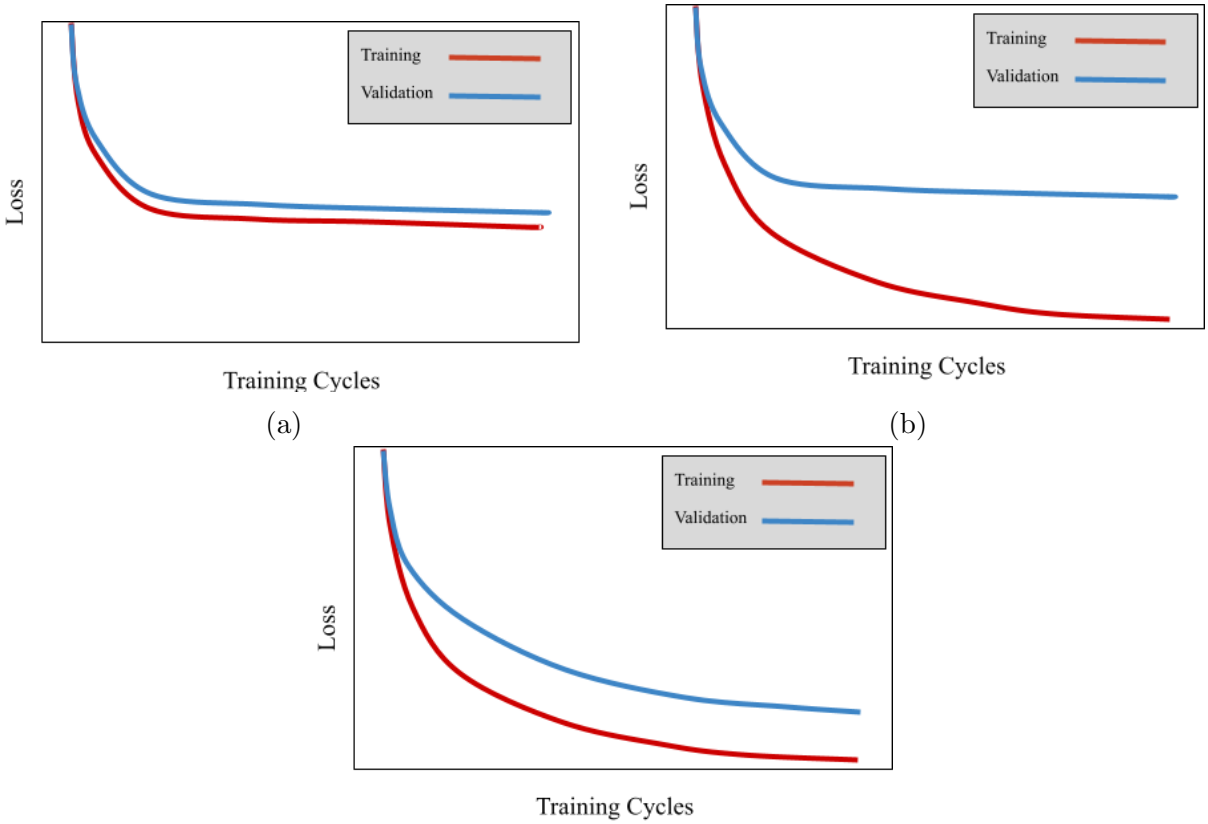


Figure 8.4: Training and validation loss plotted over multiple epochs, or training cycles. As the model trains, the error starts to lower. In (a) this model is underfitting and is not able to lower the loss error for both the training and validation data. Overfitting is happening in (b) where the training loss continues to drop, but the validation loss appears to be stagnant. In (c) both the training and testing losses are decreasing, indicating a good fitting. Notice that the validation loss usually does not drop lower than the training loss. This is quite normal. Compare with Fig. 8.3.

CHAPTER 9. NN-BASED RAIN DETECTION APPLIED TO QUIKSCAT

Constructing a NN requires us to make many decisions. The number of neurons in each hidden layer is chosen, the number of hidden layers, and the type of each hidden layer. Choices must be made for a loss functions, the optimization algorithms, and training rates. Choices abound regarding how to input the data itself. Some decisions have little affect on the performance of the net, while others can have drastic results.

Our research attempts to stick to the most basic and common methods and practices available; however, even the most basic methods have an large number of tunable parameters and decisions. This is handled by making decisions that follow the advice given in [15] and by sweeping along individual parameters to see its influence over the performance.

This chapter first discusses a NN based on QuikSCAT scatterometer data and why it was used prior to ASCAT sensor data. Next, a handful of the key principles learned from the preliminary tests with QuikSCAT are discussed. These are applied to an ASCAT NN in Chapter 10.

9.1 Preliminary Tests using QuikSCAT Data

Before a NN, which allows us to use the ASCAT SWR GMF in an effective manner, can be created, it is helpful to first understand the basics of ML. Because of this I chose to first attempt a rain-detecting NN using a signal that is more affected by the presence of rain, making detection easier than it would have been using ASCAT's C-band data.

To attempt this "best case" experiment I use data from the Ku-band scatterometer called SeaWinds which is mounted on the Quick Scatterometer satellite. QuikSCAT operates at Ku-band which experiences greater attenuation from rain and can estimate rain more accurately than lower frequency sensors like ASCAT [17]. Rain GMFs had previously been developed for QuikSCAT making it a useful sensor to start with. QuikSCAT also had an

additional GMF called the rain only (RO) GMF which specializes in estimating rain rates when rain's influence dominates the backscatter so much that wind has little influence. QuikSCAT's SWR estimator has been shown to effectively reduce rain-induced wind vector error and also produce a consistent estimate of rain in [18] and [19].

The purpose of the preliminary research is to explore the use of NNs for rain detection. If NNs are a viable solution, than it is desirable to determine a reasonable starting structure for an ASCAT NN. Further, a good start structure for an ASCAT NN is expected to be reasonably similar to the parameters discovered for QuikSCAT.

9.1.1 QuikSCAT

QuikSCAT is a rotating pencil beam scatterometer. Pencil beam sensors use narrower beams and mechanically rotate them to gather data in a helical pattern along the ocean surface. QuikSCAT has azimuth angles ranging a full 360 degrees and fixed incidence angles for two beams. Because of the rotation, each valid ground location is sampled four times. These are referred to as flavors. Thus, for every UHR pixel there are four σ_0's, four varied azimuth angles, and four relatively constant incidence angles. These samples are spaced irregularly due to the rotation of the beams.

As with ASCAT σ^0 values, the QuikSCAT σ^0 values are processed by the UHR algorithm to a 2.5 km resolution. The high resolution QuikSCAT data, like ASCAT, is collocated with TRMM measurements. A shorter collocation time window of just one minute is used because it was found to improved the performance of the network.

QuikSCAT also has characteristics that can make rain detection challenging. For example, because the four flavors occur at slightly different times, physical events are able to travel between the flavors. In a worst case scenario there is approximately 4.5 minutes between the sampling of the first flavor and the fourth [4], this means that a fast moving storm can smear itself across several wind vector cells (WVC).

QuikSCAT, also has a steep incidence angle. Because rain can occur up to an altitude of 6 km [4], the incidence angle can horizontally spread the effect of high altitude rain up to 6 km away from the reported TRMM value —well beyond the resolution size of a single WVC.

9.1.2 Data set

Collocating the QuikSCAT data set with TRMM and ECMWF proved to be a challenging task. The amount of collocated data was limited by the overlapping lifetimes of the scatterometer TRMM. Under these restriction, around 15 million UHR samples were collected over the year 2009. The data set was constrained to a year because the processing platforms used for the NN are only able to handle about a year's worth of data at a time. Also, constraining the data set to multiples of years might help control for any seasonal changes to the data distributions, which could bias the NN in some way.

9.2 Performance Metrics

One of the early discoveries from the preliminary research was the need for specific metrics to determine the effectiveness of a NN. This is challenging because rain's occurrence over the ocean is relatively rare, making the data set imbalanced. In addition to the complexity of representing the performance of the NN, I found out that a more complex solution was needed for choosing the optimal GMF for retrieval. A more complex way to rank the performance of the estimators is needed because the WO, SWR, and RO GMFs all report different types of data which are not easily comparable with one another. This section first addresses the solution for determining the optimal GMF and than discusses several of the tools used to more accurately describe the performance of the NN.

9.2.1 Performance of the Estimators

Determining which estimator performs the best is a challenging issue. The WO estimator does not estimate the rain rate and the RO estimator does not estimate the wind speed which makes it difficult to compare all three estimators against one another. For an extended discussion on this challenge see [4].

One method of determining optimality was to first decide between an estimator that reports wind speed and one which doesn't. This was determined selecting RO if it gave the most accurate rain estimate. Then for samples where RO is not the best rain estimator, a weighted error is used to select between the WO and SWR estimators. For my purposes, the

WO estimator "reports" a zero rain rate by default. The weighted error is the normalized sum of both the wind and rain error. It is calculated by multiplying the rain squared error by the ratio of rain and speed variances which gives the rain error equal weighting to the wind speed error.

This two step filtering tends to give an unfair advantage to the RO and WO estimators. Later research determined that a better metric is to judge by wind speed alone because the rain estimates for ASCAT are so poor they can be treated as nuisance parameters.

9.2.2 Performance of the NN

NNs tested with imbalanced data sets can overlook key nuances when judged solely by the percent of test samples correctly labeled. For example, a NN which labels all samples as dry, or non-rainy, is correct about 96% of the time, which may appear good, but is a poor detector. Overall accuracy also does not clearly show us when a minor category is being overrun by mislabeled majority data.

It was quickly learned that the metric of comparison should be carefully selected because of the imbalanced data set. Also, it is desirable to condense a set of statistics down to one or two value so NNs can be compared consistently.

Through research and exploration, I discovered that the most effective metrics are: the receiver operating characteristic curve (ROC) and its the area under curve (AUC), precision, recall, and the F1 score. These are common terms in the area of machine learning, but occasionally go by different names in more traditional electrical engineering fields. As such, each of these are covered in the following sections.

First, consider a few basic probability definitions concerning rain over the ocean. R is defined as the presence of rain. The absence of rain in an area is notated R'. The same notation, but lower case, is used to refer to the detection estimates. For example, the probability that it is raining is written as $P(R)$, the probability that the NN labels a sample as rainy is written $P(r)$. By extension, the probability of correctly labeling an area where no rain is present is written $P(r'|R')$. N is the number of samples tested.

Precision

Precision is the ratio of the number of samples correctly classified as category X, to the total number of samples classified as category X. It can be helpful to understand this metric as the accuracy, or the purity, of all samples labeled X. Precision applies to both rainy and dry samples. For example, the precision of the dry category is

$$P_{\text{dry}} = \frac{NP(r'|R')}{NP(r'|R') + NP(r'|R)} = \frac{P(r'|R')}{P(r')}, \tag{9.1}$$

where P_{dry} is the precision of dry samples.

This metric normalizes the accuracy of a label by the likelihood of estimating that label. For my tests this metric proved to be enlightening because the number of rain samples is small and missing rainy samples does not have a large affect on the overall accuracy.

Recall or True Positive Rate

Recall is the percent of a category that has been accurately classified. Another name for recall is the True Positive Rate (TPR). Eq. 9.2 defines the recall of rain. If the rain recall rate is very low, the estimator is discovering very few of the rain samples.

$$\text{TPR}_{\text{rain}} = R_{\text{rain}} = \frac{P(r|R)}{P(R)} \tag{9.2}$$

False Positive Rate

False Positive Rate (FPR) is the percent of a category that is classified incorrectly. This measurement combined with the TPR sums to one. For example, the FPR of rain detection is written as

$$\text{FPR}_{\text{rain}} = \frac{P(r|R')}{P(R)} = 1 - \text{TPR}_{\text{rain}}. \tag{9.3}$$

F1-score

Precision and recall have an loose inverse relationship. Generally, one can increase precision at the expense of recall, or increase recall at the expense of precision. This can make an overall comparison between classifiers difficult. The F1-score combines precision and recall into a single metric for easier comparison. The F1-score of dry data is

$$F_{\text{dry}} = 2\frac{P_{\text{dry}}R_{\text{dry}}}{P_{\text{dry}} + R_{\text{dry}}}, \tag{9.4}$$

where P_{dry} and R_{dry} are the precision and recall of values for dry.

ROC

The ROC curve indicates how the TPR and FPR change with varying prediction thresholds. Binary classifiers, like the NN, report how likely a positive classification is with a decimal number between zero and one with one being a very high likelihood. A threshold is created that classification probabilities needs to be above in order to be classified as a specific category. The ROC curve plots the scope of possible TPR and FPR rates at which the classifier can perform.

A good classifier is able to achieve high TPR while maintaining low FPR. A very poor classifier that performs a random guess has a ROC that is a diagonal line between position 0,0 and 1,1, see 9.1. In contrast, a perfect estimator touches the upper left corner of the ROC space.

AUC

While the ROC curve provides significant amounts of information about the performance of a classifier, it gives too much information to easily compare the performance of one classifier against another. Since the ROC curve is generally a smooth shape, the ROC's information can be compressed by taking the area beneath the curve as shown in Fig. 9.2. The "area under the curve" (AUC) helps compress the ROC down to a single number which enables easy comparison of classifiers. A perfect classifier has an AUC of 1, whereas a per-

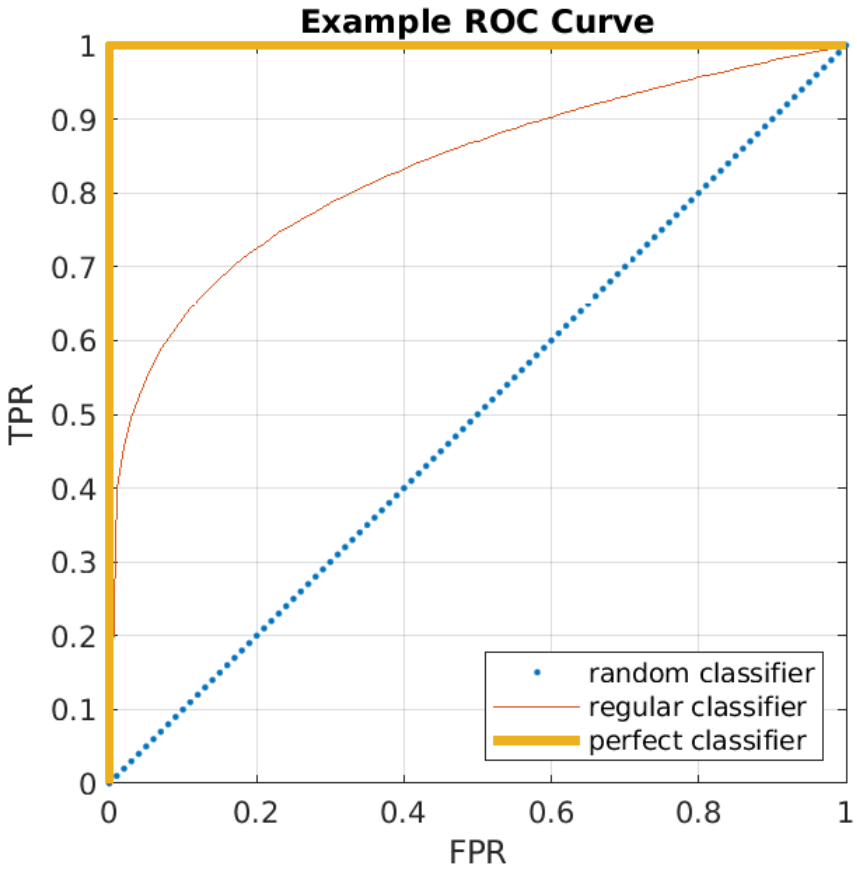


Figure 9.1: The ROC plots for 3 detectors. The diagonal line represents a random classifier, the yellow line represents a perfect classifier. Usually binary classifiers reside somewhere between the two, such as the regular classifier illustrated.

fectly random classifier has an AUC of 0.5. Any value of AUC below 0.5 is invalid because it represents a classifier that can segregate the two classes, but mislabels them.

9.3 Handling NN Complexity

There is a sweet spot in the training that is a balance between overfitting and underfitting. Initial tests severely underfitted because the nets were not structured or trained to reach deep enough complexity. When the research switched over to more complicated methods it was found that some parameters control complexity effectively, while some had no effect at all. This section describes the methods used to control the complexity of the NN and summarizes which are effective for classifying rain.

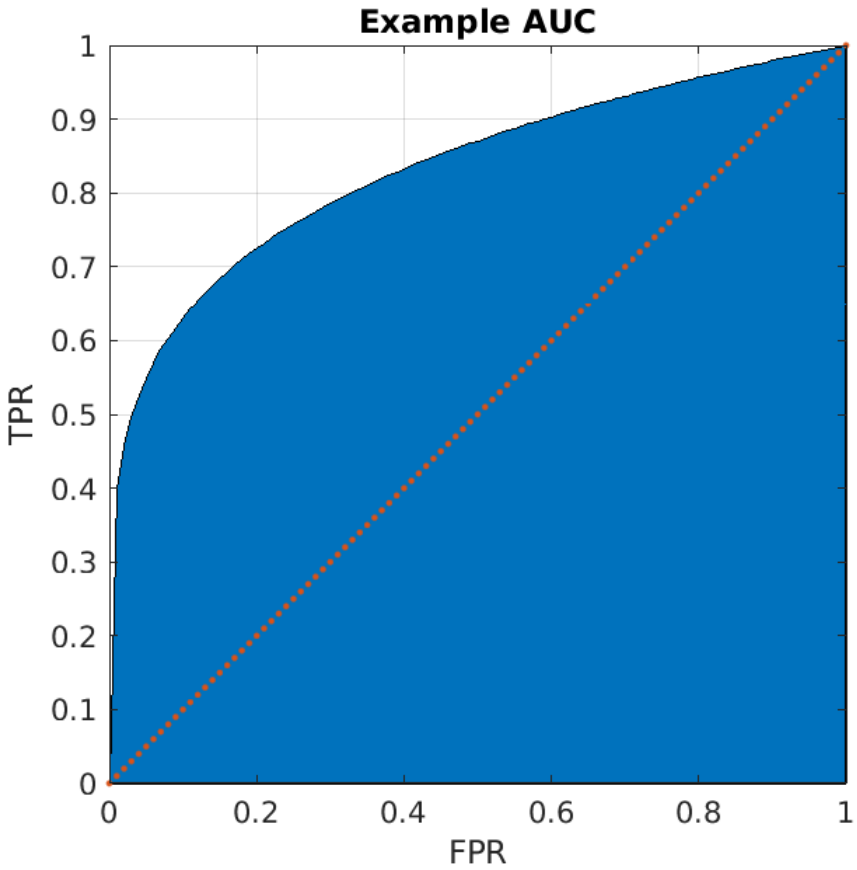


Figure 9.2: Illustration of how the AUC is calculated. AUC is the area underneath the ROC curve and succinctly describes the quality of the classifier. Perfect classifiers have an AUC of 1. Regular classifiers have AUC < 1. This has an AUC around 0.837.

9.3.1 Shape

The shape of the NN refers to its depth (number of layers) and width (how many neurons each layer has). Wider and deeper NNs have more tunable parameters which can be used to characterise a feature. If a NN is restricted to only one layer, the result is a linear NN. With more layers, or more neurons in a layer, the complexity increases.

Layers can be different widths and types. NN shape affects various aspects of the NN. A typical design pattern in [15] is to start wide and then taper off to more narrow layers. Sometimes this is done multiple times with specialized layers between. Because wider layers are more complex, this pattern can be understood as allowing the NN to first explore many sub features. The following shallower layers force the NN to condense these

small patterns into larger ones, and finally make a decision based upon the cumulative patterns noticed higher up in the NN structure. My experiments saw no real difference in performance following this pyramid-like structure. However, having too few hidden layers led to less effective NNs.

As for layer types, I choose to focuses primarily on fully connected layers. Other types of layers restrict connections in some way, or sometimes reinsert the original input mid-way through a stack of layers. I choose to use fully connected layers because fully connected layers have weights attached to each node that make it possible for a fully connected layer to mimic many different types of layers if these layers are properly trained. My research works with different shapes of feed forward, fully connected layers for these reasons.

Early tests used a single layer and up to 100 neurons. These tests never outperformed WO method of never selecting rain because the NNs were too simply and underfit the data. As the complexity was increased the NN went into the failure mode of "never guessing rain" (NGR). This suggests that the NN was getting stuck at a local minima and the NN needed more guidance during the training processes. It was found that a NN shape with around 3 to 4 hidden layers and 100 to 1000 neurons would provide a decent amount of complexity before throwing the NN into the NGR mode.

9.3.2 Regularization

Regularization is the term used to describe methods to combat overfitting and guide the NN during training. This section focuses on two: dropout and weight decay.

Dropout is a method to help a NN not become too dependant on a specific neurons during training. This is done by selecting a percentage of neurons in a layer and multiplying their output by zero during training. The neurons which are zeroed out is picked at random and switches during each training iteration. This forces the NN to become more generalized, or less complex, which reduces overfitting.

A dropout layer behaves like an array of ones and zeros which is slipped between layers in the NN [20]. A practical method is to place dropout layers between fully connected layers, and have the percentage zeros increase with increased depth. Some example NNs use dropout rates of up to 50% [15]. After training, the dropout layers are removed. These

layers act like braces to help the NN grow in a better generalized way, but are not part of the finished product.

Our research found that using dropout layers effectively prevented the NN from overfitting and enabled better performance. The effect of dropout layers is best when the dropout rate starts small in early layers and than increases. The difference in performance changes slightly with different dropout rates, however, it does not change performance so drastically as to warrant much tuning. I start my dropout layers at around 20% and than working up to 50%.

Another method of regularization is weight decay. This is motivated by the intuition that for any function, assigning all parameters to zero puts the function in its simplest form. As such, the complexity of a function can be approximated by measuring the distance between a functions parameter's and the zero weight case. Weight decay reduces complexity by lightly decreasing the weights, bringing them closer to zero at each stage of the training. This diminishes out-of-control weight growth which a NN can experience when overfitting. Also, it prevents the NN from making extreme decisions due to a single neuron. During testing, weight decay did not effect overfitting. As a result I chose to forgo weight decay.

9.4 Minority Oversampling

Rain happens so infrequently that NNs can become biased against labeling any samples as rainy. Minority oversampling is a way to counteract this bias. Oversampling is training the NN using more rain samples than actually occurred in nature. The resulting NN is more likely to label samples as rainy with the regular testing data set. The trade off is that the NN is also more likely to have false positives where it mislabels a sample as rainy. Using minority oversampling enables a balance between increased false positives and increased true positives that can raise the overall performance.

I considered three methods of minority oversampling: limiting majority samples, repeating minority samples, and the synthetic minority oversampling technique (SMOTE). Limiting majority samples means randomly choosing as many majority samples as there are minority samples. This, however, effectively reduces the data set to 8% of it's original size, which is undesirable. This method was quickly dropped.

Repeating samples means replicating minority samples to achieve a desired concentration. This proved to be an effective method and helped to prevent the NN from arriving at the NGR failure. Great care was taken to make sure the replicated samples only happened after the data was divided into testing and training data sets so that no replicated data would be seen in testing.

SMOTE is a more sophisticated form of repeated samples. SMOTE creates new minority samples which closely resemble original samples, but are not exact copies [20]. The benefit of using SMOTE is that the NN gets to learn general trends from minority-like data, which has the potential to more general than learning the same trends twice from the same data.

SMOTE works by finding minority samples similar to one another and projecting a line between their parameter values. The new sample is made of parameters which are chosen randomly along this connecting line, thereby making a new sample which has characteristics similar to the original data. By not being exactly the same as the original samples, these SMOTE samples make more robust NNs [20].

In practice there was little difference in performance between SMOTE and repeated samples. Because the difference seemed insignificant, I choose to use SMOTE instead of repeated samples because, in principle, it should be more effective at preventing overfitting.

In the case of a NN with 2 hidden layers containing 256 neurons each and operating over the final form of the NN and data it was found that minority oversampling is highly effective. Without minority oversampling the NN started to overfit around 30 epochs and only achieved F1-scores of 0.625 and 0.026 for dry and rainy samples respectively. When minority oversampling was employed the net was able to be trained for well over a thousand epochs without overfitting and the F1-scores reached 0.974 and 0.465.

9.5 Training Parameters

Some of the conditions set for training are critical to the performance of the NN, others have very little effect. This section describes several parameters and how they affect the NN performance. Also, this section describes a few training decisions which helped speed up the training processes.

9.5.1 Software Platform

The NN training code was implemented on a Google Colab platform using TensorFlow libraries. Colab was chosen as the platform because it connects directly to free graphics processing units (GPU) on Google's servers. GPUs are designed to do simple math very quickly. Traditionally GPUs are for graphics; however, they are also ideal for NNs where training and updating are done by vector math. The training speed increased roughly four fold with the use of GPUs.

Colab is able to access data stored in a user's Google Drive making it easy to access large sets of test data without having a lengthy upload every time the NN is trained. Colab, however, has some draw backs. Because GPUs are generally expensive, Colab has limitations to how long codes can be run on them before the system times out.

9.5.2 Training Duration and Learning Rate

Training duration refers to how many times the set of weights are iteratively updated (tuned). Duration is measured in epochs, or one run through the entire set of training data. The learning rate controls how powerfully the weights are adjusted during each backpropagation of the error. The algorithm that updates the weights and biases is called the optimizer. In terms of the gradient decent, the learning rate controls how large of steps the NN takes while searching for a minimum and the training duration determines how many steps it takes.

Training duration and learning rate are intertwined and need to be balanced. The training needs to be finish in a reasonable time, and find the lowest minimum, not just local minima. This requires a larger step size and fewer epochs. On the other hand, it is undesirable to miss the minimum error by constantly stepping too far over it. So smaller steps and more epochs are desirable. The research originally stopped training after about 10 epochs which underfit the data and gave us poor performance and little insight into the parameters being manipulated. Fig 10.1 demonstrates how increased training duration can result in better performance for ASCAT. QuikSCAT, by comparison, tends to flatten out faster than ASCAT, but the underlying principle is the same.

When written using Keras, the NN can use the Adam optimizer [20]. This optimizer starts with a quicker learning rate and then slows things down as the NN approaches a minimum. This is done automatically in the Colab system with little input required of user.

Most NN training set ups also provide the option to train until a desired threshold is met. For example, a NN can be trained until the error of the validation set is below some minimum. However, because the run time required is often longer than what Google Colab allows for their free GPU services, it was desirable to train for a given number of epochs instead. The NN can be saved after a number of epochs, and than training can be restarted in a new Google Colab run. My research found that with around 700,000 training samples, the training system could do 300 epochs before timing out. In most cases, after about 1200 epochs the performance stopped improving and training could be terminated.

Allowing training to run too long can introduce overfitting [21]. I avoid this issue by implementing early stopping which ends training when the validation data set losses stops dropping for a number of epochs.

9.5.3 Input Conditioning

NN performance can be improved by preconditioning the input data. This is also called normalization. Normalization is implemented by transforming the data into a Z-score as

$$Z = \frac{x - \mu}{\sigma}, \tag{9.5}$$

where x is the observed value, μ is the sample mean, and σ is the standard deviation of the samples.

Normalization is effective because it puts all the inputs on a common scale and range. This prevents any one parameter, say the incidence angle, from have more influence over the NN just because angle values are usually larger than backscatter values.

Using the same NN structure described in Sec. 9.4 with the SMOTE replicated data and 100 epochs, the NN falls into the NGR failure mode and scores F-1 scores of 0.983 and 0.000 without input conditioning. When the input is conditioned the same NN structure yields 0.973 and 0.458 F-1 scores.

9.5.4 Activation Functions

The NN introduced in Fig. 8.2 left out a key component for effective NNs which I now focus on. The example in Fig. 8.2 is entirely linear —neurons being added together and multiplied by weights. However, multiple linear layers can be rewritten as a single linear layer with no added complexity. The stacked linear layers essentially collapse down to a single layer and provide no added depth to the NN.

The trick to achieving the added complexity from adding multiple layers, is to put a non-linear element between each layer. The nonlinear layers are called "activation functions", and they prevent stacks of linear layers from compressing down to just a single layer. Activation functions need to be differentiable so they can be used during the gradient decent processes. They can also be used to reduce complexity by constricting their output to a more narrow range of numbers. The Sigmoid function for example, squeezes any input to fit between -1 and 1.

There are many different types of activation functions: ReLU, Sigmoid, Tanh, and more [15]. Each has their own characteristics that work best in different circumstances. Several different activation functions were tested and no significant change in performance was observed. Thus, I decided to use one of the most common activation functions called the ReLU function.

The ReLU function is defined as

$$\text{ReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x < 0. \end{cases} \tag{9.6}$$

ReLU is computationally simple as it merely zeros out any negative inputs. It's derivative, which is needed to update the weights, is zero where input is negative and 1 where it is positive.

9.6 Input and Output Data

One of the most influential factors that controls the performance of a rain classification NN is the input and output data used to build it. This section discusses a few basic principles regarding data and several methods attempted.

9.6.1 Data Handling

Because there is limited control over what the NN actually learns during training it is critical that the data is handled carefully. It is especially important to avoid cross contamination which happens when samples used for training are also found in the testing data set. Cross contamination makes the testing data set scores artificially high.

To avoid contamination I split 20% of the data off to form a testing data set, and the remaining portion is used for training the NN. Keeping these data sets separate avoids NNs that operate "well" during training and testing, but poorly on new data. Unfortunately, however, for a large portion of the research, a form of cross contamination still occurred because the data was split in a way which allowed very similar samples to go into the testing and training data sets. This is discussed further in Sec. 10.2. This affected all of the QuikSCAT tests and most of the ASCAT tests.

9.6.2 Data Amount

Generally, using more data to train a NN is best. More data usually makes sure that the characteristics which identify objects are better understood by the NN. There are some limitations to the data size for QuikSCAT. QuikSCAT was launched in 1999 and exceeded design expectations by operating until 2009 when a bearing failed and it started to operate in a modified mode. TRMM was launched in 1997 and operated until 2015, giving the research a collection window ranging from 1999 to 2009. Because this research is preliminary to ASCAT research I decided to merely limit the data set to a single year. I limit the data in this way for simplicity, and also because Google Colab has finite storage and using more than a year's worth of collocations usually causes the system to crash.

9.6.3 Data Type

For my experiments, the input data could be any type of parameter derived from the ASCAT sensor: time, date, location, backscatter, or WO estimate, for example. A basic understanding of rains affect on backscatter and the goal to keep the NN generalized limits the input list to the backscatter signal, sensor geometry (azimuth and incidence angle), and the wind and rain estimates from the WO, RO, and SWR estimators. The outputs are limited to rain flags, rain rates, wind speeds, and the optimal GMF.

There is a large degree of flexibility in the ways a NN's input and output data can be used. This flexibility makes working with NNs both versatile and challenging. I define three characteristics which can be used to distinguish the types of input and output data.

Near Neighbor WVCs

Our first distinction for the data is the decision to use a region of data, or only the data from single WVC. The logic behind using a larger region of data to detect rain is that the NN might be able to make a better decision with more input data. This might also allow us to find regions of rain in the output, instead of just single occurrences.

There are some down sides to using the near neighboring cells. It would require a more complex NN to handle this sort of spatial comparison. Also, this makes preparing the data significantly more challenging. Although, UHR data is uniformly gridded, the overlapping swath with TRMM results in irregular diamond like patches.

The alternative to using the near neighboring WVCs is to use a single collocation point. Setting up this problem is far simpler, but potentially misses out on some information which could be used to improve performance.

In the case of QuikSCAT, the best results came when a 7x7 square grid of input data was used to estimate the center WVC's rain conditions. This required an original NN structure which kept each individual WVC separate for several layers and than combined them all together. The logic behind this was to force the NN to first processes the information contained in each WVC and than compare information about each WVC against other

WVCs. Without this layering structure the 7x7 NN performs only as well as a single WVC input type.

Input: GMF Estimates v Input σ^0

The next distinction refers specifically to the input data. Does the NN use the raw collected σ^0 or the estimates of the GMFs, specifically SWR, WO, and RO, which are functions of σ^0 and the sensor geometry? Alternatively, the NN could use a combination of both input types. Preliminary tests were effective at eliminating other input data methods which proved to be less effective.

Because the GMF estimates are functions of σ^0 and the sensor geometries, a sufficiently complex NN using σ^0 as an input should "theoretically" be able to recreate the wind retrieval processes. In this way, all the information contained in the GMF estimates are also in the raw σ^0. It should be pointed out that "theoretically" does not imply practically, but it suggests that using σ^0 as an input should supply at least the same amount of information as using the GMF estimates. However, as with the input conditioning, how the data is input can have a significant consequence on the performance of the network.

To eliminate more tunable parameters and speed up the research processes, I decide to only test the σ^0 for ASCAT training. Also, in the case of ASCAT, there is only the WO and SWR GMFs to work with. Furthermore, any change to the SWR GMF, which was created after the ASCAT NN, would require the ASCAT NN to be reconstructed.

Output: Optimal GMF v Rain detection

The last distinction refers specifically to the output. Does the NN select the optimal GMF, or detect the presence of rain? I attempted a few other types of outputs where the NN predicted the "true" wind and rain rate, but this goes beyond the scope of the research as the NN would need to subsume both the GMF and the ambiguity selection algorithms.

Ideally the SWR or RO GMF is always optimal when rain is present and WO GMF is optimal when no rain is present. Based on results presented later, this is only the case 79.9% of the time when optimality is defined as having the most accurate wind speed. As

such, even with a perfect rain detector, the total detection error will not be minimum. Using the optimal GMF as the output would likely result in a lower total error. Also, depending on how optimality is defined, using the optimal GMF output would make it possible to collect significantly more training data because there would be no need for TRMM data. Regardless, there are drawbacks to using this output scheme that make it ineffective.

First, any changes to the ASCAT rain GMF requires a change to the NN as well. Having the two tied together makes it difficult to work on and improve each independently. Also, the ASCAT SWR GMF was not created until after the NN research, making the estimators not an output option during the development. In addition, models of ASCAT's SWR were found to actually estimate wind speeds better over dry samples and rainy ones. In this way, detecting the presence of rain also preserves the usefulness of this research and it the method I choose to use.

CHAPTER 10. NN-BASED RAIN DETECTION APPLIED TO ASCAT

There are several key principles taken from the QuikSCAT preliminary research which are applied to ASCAT. First, precision and recall and other more helpful metrics are used to determine progress in the NN design. Next, the NN is kept to a simple shape and then slowly made more complex so long as overfitting is not an issue and the performance continues to improve. Also, drop out layers, the Adam optimizer, and saving the NN to allow for longer trainings all improve the overall performance. Normalizing the input data as a Z-score also helps overall performance. Use of a GPU speeds up the training processes greatly.

This chapter briefly discusses the training and optimizing processes for determining a good ASCAT NN configuration. An issue with the data set which greatly reduces the performance of the NN is considered. Possible solutions and the performance is discussed.

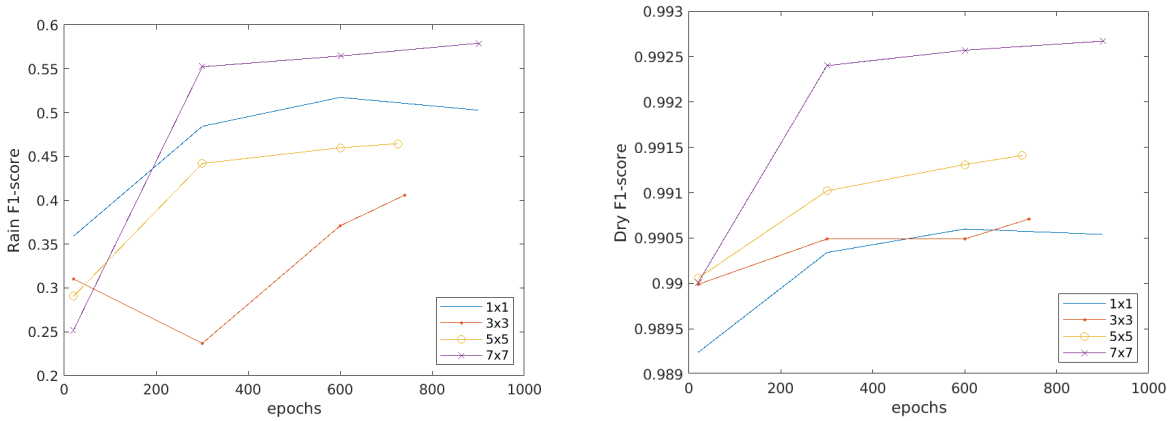
10.1 Design Processes

Because of the sensor differences between ASCAT and QuikSCAT, it was not expected that the exact same NN setup which worked best for QuikSCAT would also work best for ASCAT. It is assumed, however, that the working NN structure for QuikSCAT is a good starting point for ASCAT.

In this section I first sweep along the number of nearest neighboring WVCs used as an input and determine that the 1x1 and 7x7 configurations perform best. The NN structure which performed the best for QuikSCAT is used. Next, I explore how SMOTE affects both configurations and find that a 17.5% concentration of SMOTE data results in the highest PR AUC for the 1x1 configuration, while a 10% concentration works best for the 7x7.

10.1.1 Use of Neighboring WVCs and Training Duration

ASCAT is observed to train slower than QuikSCAT. After about 20 epochs, QuikSCAT usually stops improving. ASCAT, on the other hand, can be trained for over 1000 epochs before the loss becomes a flat line. As such, I found it important with ASCAT to train to at least 300 epochs before making decisions regarding performance. This issue arose early for the neighboring WVCs tests because after only 20 epochs the worst performing NN used a 7x7 configuration. After 300 epochs, however, the 7x7 configuration proved to be significantly better than the other configurations, as seen in Fig. 10.1. Interestingly, the 1x1 configuration also proved to be effective.



(a) (b)

Figure 10.1: Dry and Rainy F1-scores for a range of epochs. Several different configurations of using the neighboring WVCs are plotted. The 3x3 NN, for example, uses the ASCAT data from the 8 neighboring WVCs in addition to the center WVC which it is predicting the rain for. This figure shows that after 300 epochs, the 7x7 gridding proves to be the most effective method. Also, the longer training (more epochs) result in more accurate NNs.

I decided to stop my configuration size at 7x7 because this method doesn't allow the 3 WVCs closest to the edge of the ASCAT swath to be processed. This happens because all neighboring WVCs are used to estimate the presence of rain in the center cell. A more complex NN might be developed in future research which takes in the entire swath and identifies the rain flags for every WVC instead of just the center one. This would change the

NN from a multiple input single output (MISO) system to a multiple input multiple output (MIMO) system.

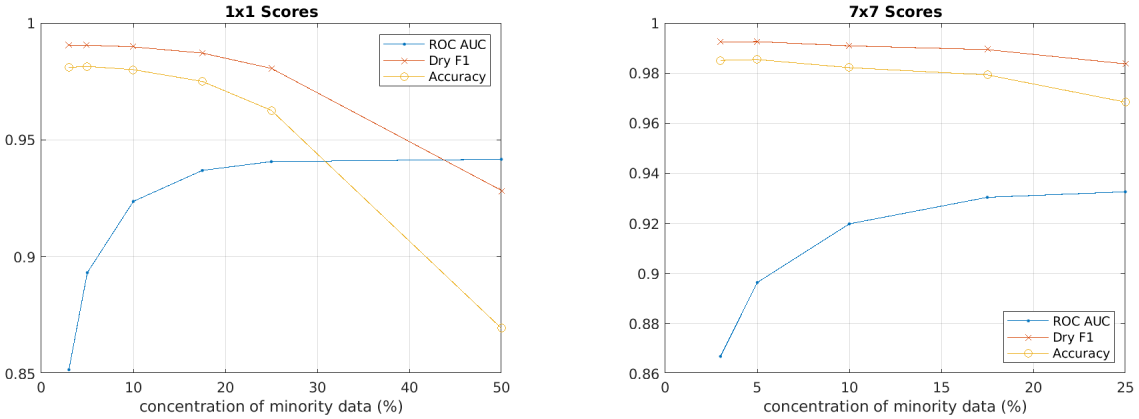
I chose to use a MISO system for simplicity because the ASCAT and TRMM collocation swath is an irregular shape with data gaps that make traditional NN methods used on constant shaped images more difficult.

10.1.2 Minority Oversampling

Section 10.1.1 showed that the 7x7 and 1x1 NN configuration worked best, so only these combinations are tested with the minority oversampling technique SMOTE. During tests with SMOTE, it was noticed that increasing the concentration of minority samples created using SMOTE had complex effects on the ROC AUC, PR AUC, F1 score, and the overall accuracy, see Fig. 10.2. The tests from Section 10.1.1, in contrast, had a consistent behavior among the performance metrics. The F1, accuracy, ROC, and PR AUC scores all increased or decreased together.

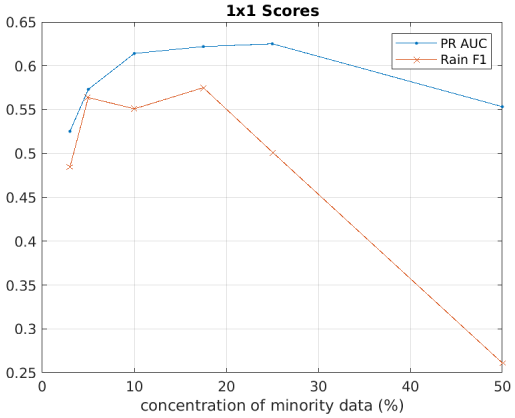
The divergence of these metrics suggests that a design decision needs to be made regarding which metric is optimised. I decided to pursue the increased performance of rain detection, as opposed to dry detection, because the WO case is so common that removing rain samples only improves the overall RMSE of wind speed by 4.6%. Using the SWR GMF on a rainy sample improves the wind speed RMSE of that sample by 13.6% on average. The rain flag can be used by future researchers also. The downside of optimizing for rain estimation is that more false positives lower the overall accuracy of the NN.

I decide to maximize the PR AUC metric because [22] suggests that it is a better indicator of performance for binary classification than the ROC AUC due to the data set's imbalance. As such, the best performance for the 1x1 and 7x7 configurations is 0.622 and 0.636 PR AUC, respectively. These are trained with data sets that contain 17.5% and 10% minority samples respectively.

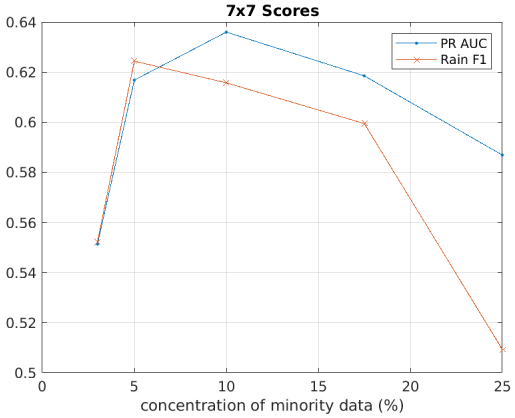


(a)

(b)



(c)



(d)

Figure 10.2: ROC and PR AUC, rain and dry F1-score, and accuracy for varying concentrations of SMOTE minority data. A 1x1 and 7x7 configuration are trained for 300 epochs and the resulting values shown. Changing concentrations of SMOTE minority data has varying effects for each metric. The ROC AUC always increases with higher concentrations of SMOTE data. The PR AUC has a maximum around the 10 or 17%. The accuracy and Dry F1-score decrease continuously, whereas the rain F1-score peaks around 5 and 17%.

10.1.3 Adjusting Size

Several NN shapes are explored to help improve the performance. It was found for the 1x1 case that adding another hidden layer and tapering the number of neurons to start with 2048 and work down to 256, resulted in a significantly more effective NN. These changes increased the PR AUC from 0.6391 to 0.6892.

For the 7x7 case, it was found that increasing the complexity for the layers which processes the single WVC inputs improved the PR AUC. By adding another layer to processes the single WVC and increasing the number of neurons for these layers, the PR AUC rose from 0.6560 to 0.6892.

10.1.4 Structure of Best Design

Because the 1x1 and 7x7 configuration score very close to one another, I present both as viable solutions for rain detection. The 7x7 configuration is significantly more complicated to implement as it needs to use the WVCs neighboring it. Each type scored PR AUC values of 0.6891.

Best Structure of 1x1 NN

The 1x1 NN is made of 4 hidden layers and a softmax output layer. The first four layers contain 2048, 1024, 512, and 256 neurons respectively. Each layer is fully connected. Drop out layers are implemented between the layers. This simple design uses data that has the minority portion over represented by SMOTE. Minority data makes up 17.5% of the testing data. Data samples are normalized.

Best Structure of 7x7 NN

The 7x7 NN has similar preprocessing done to the training data and operates at 10% concentration for the minority samples. The 7x7 NN structure, however, is much more complex. Fig. 10.4 demonstrates the flow of the data but shows a 3x3 configuration for visual simplicity. Each WVC and it's data is processed individually. These outputs are than concatenated along with the output of convolution layer which operates on the entire data

```
n1=2048
n2=1024
n3=512
n4=256
def build_model():
    model = keras.Sequential([
        Dense(n1, activation=tf.nn.relu, input_shape=[inputShape]),
        Dropout(.2),
        Dense(n2, activation=tf.nn.relu),
        Dropout(.3),
        Dense(n3, activation=tf.nn.relu),
        Dropout(.4),
        Dense(n4, activation=tf.nn.relu),
        Dropout(.5),
        Dense(2, activation='softmax')
    ])
```

Figure 10.3: Keras code for the best 1x1 NN configuration.

set. Next, the concatenated output is than run through more convolution layers. This flow of the data, ideally, constricts the NN to learn about spatial relationships between the WVCs. A variety of alternative configurations were tried and found to be less effective.

The convolution layers for the upper track are two fully connected layers, each of 256 neurons, with drop out layers sandwiched between them. The 2 hidden layers for each individual pixel have 25 neurons each. The convolution after the concatenation uses one layer with 256 neurons before it passes the data to the output layer.

10.2 Testing Contamination and Overfitting

To be thorough in my testing, I gathered previously unused data from another year to use as withheld data to validate the current performance. When the best 1x1 NN was tried on the new set data, a drastic decrease in the NN's performance occurred and the PR AUC changed from 0.6893 to 0.1550. This led to the discovery that the neighboring WVCs are similar enough to one another that simply shuffling all the UHR WVCs and than dividing the result into testing, training, and validation data sets, leads to a contaminated testing

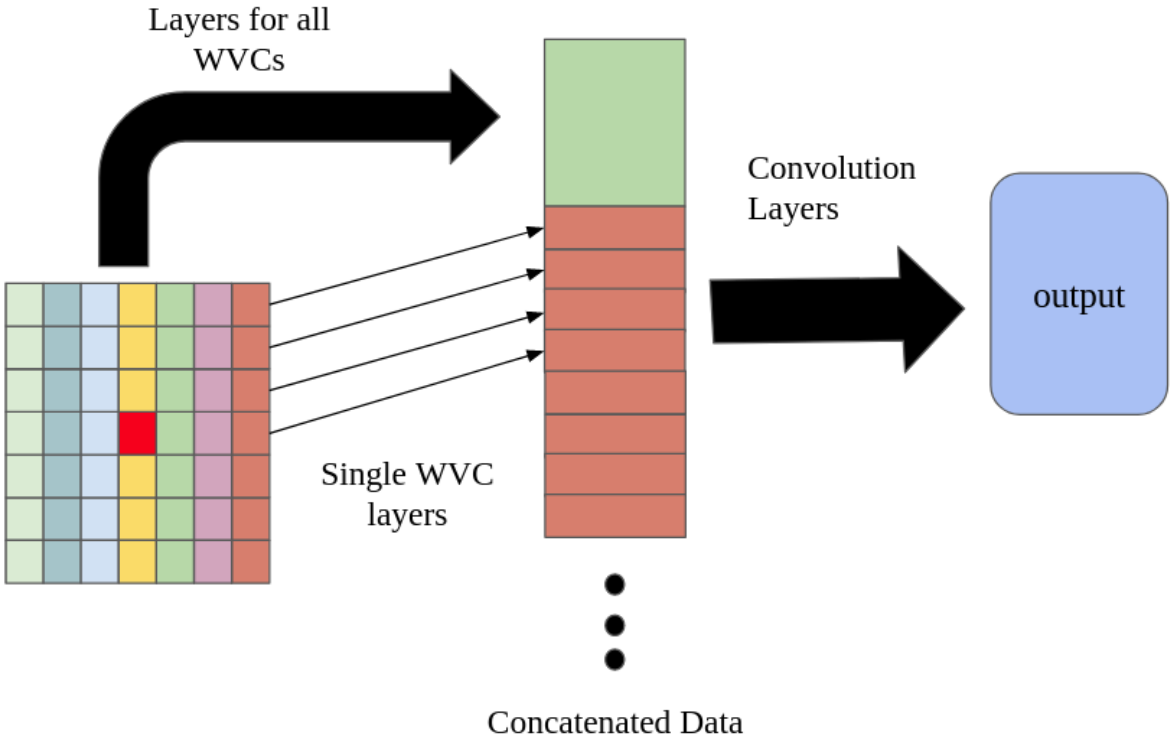


Figure 10.4: Flow of data through the 7x7 configuration NN. The 7x7 grid represents represent the data from individual WVCs surrounding and including the center WVC. Along the upper track (bold array), a simple multi layer perceptron processes the entire set using fully connected layers. The lower track (thin arrows) takes the data values from each individual WVC and extrapolates meaning from them. Both tracks converge and are concatenated into one block which is further processes by fully connected convolution layers.

set. As a result the NN was overfitting the data, but this overfitting was not detected. This section demonstrates the near neighbor similarity and its effects.

10.2.1 Data Sameness Amongst Neighboring WVCs

The collocation processes starts by finding TRMM and ASCAT samples over the ocean which occurred at nearly the same place and time. These collocations happen in patches because the TRMM and ASCAT satellites intersect just a few times each day. Each collocation patch contains hundreds of 25 km resolution ASCAT samples. Using the UHR processes, the high resolution WVCs are broken down from a 25 km resolution to a 2.5 km

resolution. As a result, neighboring WVCs of the same collocation patch can be very similar to one another.

This sameness becomes an issue when the data is split up into testing, training, and validation data sets. When neighboring WVCs in the testing data samples are in the training data set, the samples are similar enough that they can be nearly identical, making the test too easy because the NN was trained on nearly identical data. As a result, the model performance appears to be extremely good, but when data from a new collocation patch is tested by the network, it performs poorly.

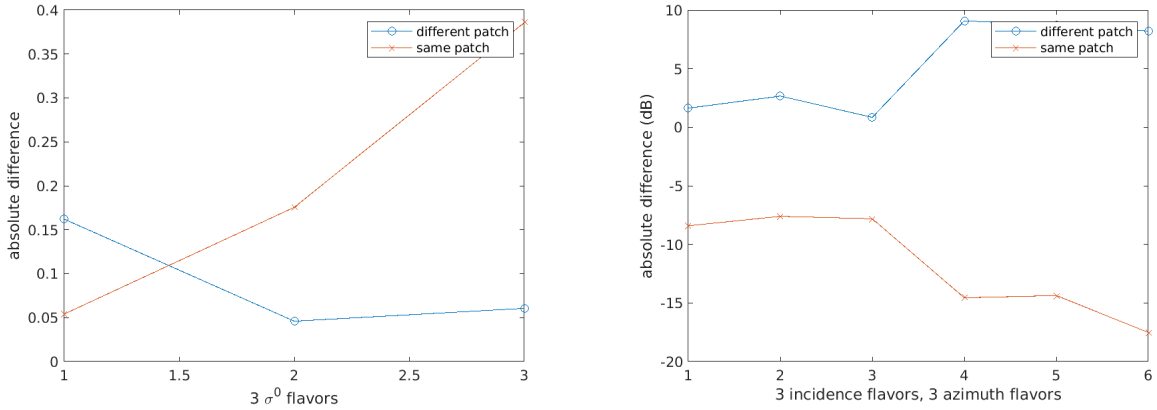
An experiment is created to explore how similar UHR WVCs are to their nearby neighbor cells from the same collocation patch. I compare this with the sameness seen among all other samples from different patches.

Sameness, in this case, is quantified by taking the absolute difference between the 3 flavors of σ^0, 3 flavors of azimuth and incidence, rain rate, and the wind vector broken into U and V parts. The 12 differences are next normalised by their parameters mean and summed to a single value which enables a WVC "sameness" score to be compared against another. Smaller sameness scores correlate to more similar samples.

Inside the original collocation patch, the most similar samples had a score of 0.156. The most similar sample outside of the same collocation patch had a score of 0.709, which was about about 4.5 times larger. Fig. 10.5 shows the absolute error for the most similar WVC originating from the same patch and the most similar from the remaining patches. Except in the case of the σ^0 flavors, the absolute error is significantly smaller for the WVC originating from the same patch. This suggests that UHR WVCs from the same patch are highly similar to one another and thus may contaminate the testing data if not separated.

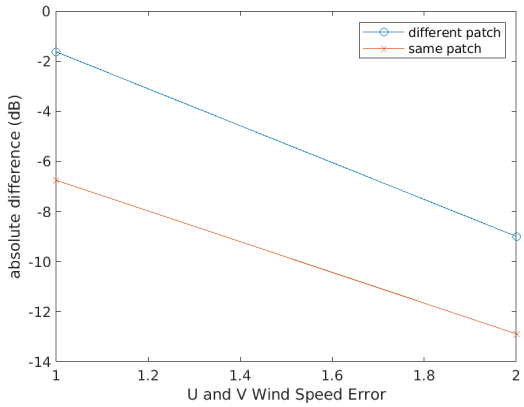
10.2.2 Handling Near Neighbor Sameness to Control Overfitting

Near neighbor sameness is handled by splitting up the data into test, train, and validation data sets using whole collocation patches. Using the same ASCAT collocated data set as has been used, this results in 555 patches set aside for training data, 133 patches set aside for testing data, and 38 patches set aside for validation.



(a)

(b)



(c)

Figure 10.5: The absolute difference between a sample WVC and the most similar WVC in the same collocation patch. This is compared with the most similar WVC in all other collocation patches. It is shown that WVCs from the same patch can be very similar to one another, whereas WVCs from different collocation patches are not as similar. (a) shows the absolute difference for the 3 σ^0 flavors. (b) shows the absolute difference for the 3 incidence angles and 3 azimuth angles in dB. (c) shows the absolute difference for the U and V components of wind speed.

Large portions of the data can be redundant because samples from a single patch are often very similar. Because of this redundancy the "effective" size of the training set is significantly smaller than the number of UHR samples.

10.2.3 Performance After Contamination Correction

By dividing up the data for testing, training, and validation according to individual patches of data, the deceiving impact of the near neighbor sameness is removed. The validation data set for the 1x1 configuration clearly shows the overfitting which occurs during training. Fig. 10.6 shows the loss for the validation data set along side the loss for the training data set. The rapid separation of these two losses shows overfitting. This behavior continued even when a very simplified 1x1 NN structure was attempted.

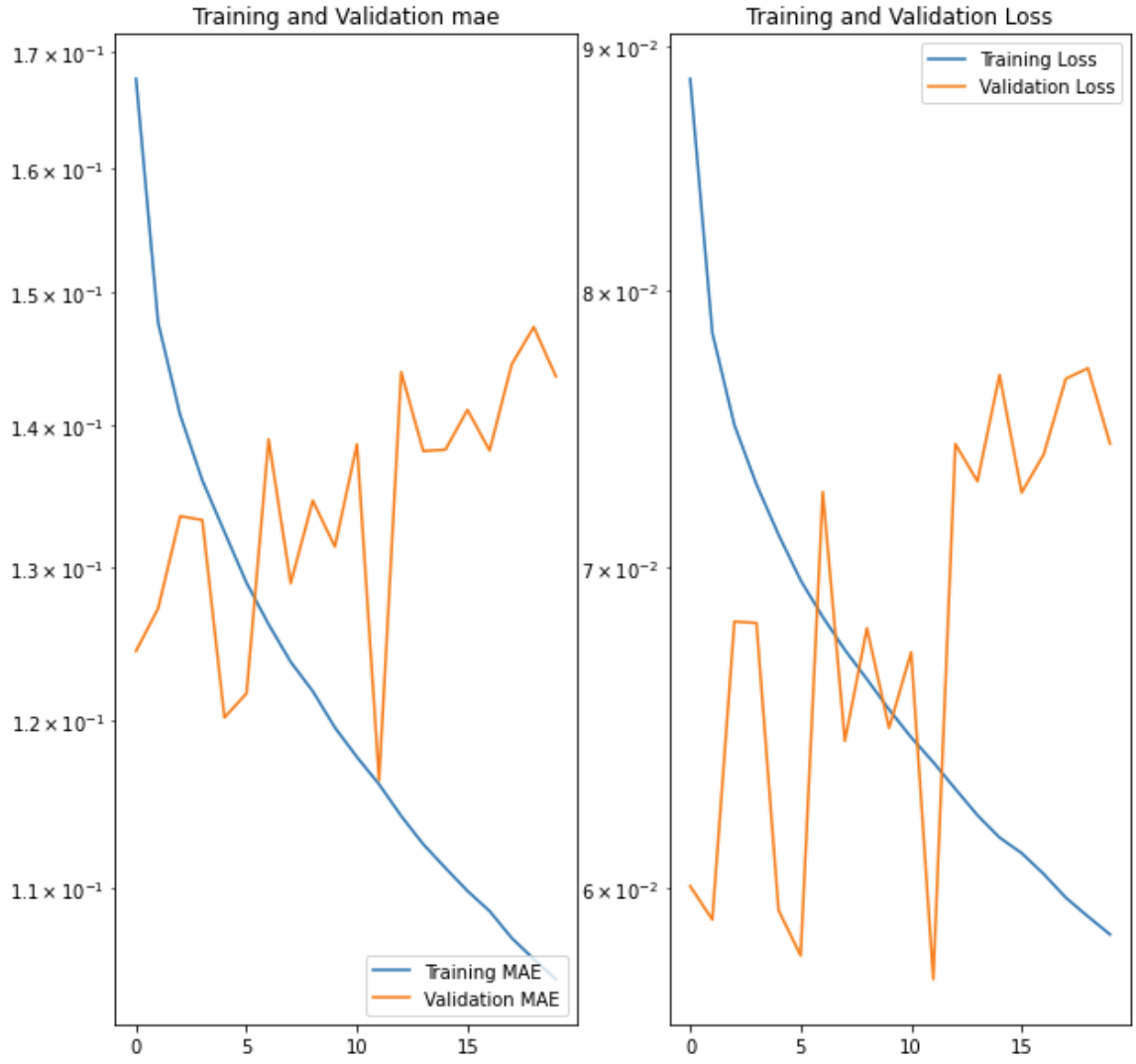


Figure 10.6: MAE and MSE training and validation losses for near neighbor contamination corrected data. Note that the decreasing training loss has little relation to the validation loss.

This behavior also occurs for the 7x7 case. Using the same data set as before, but splitting up the data to avoid contaminating the testing data set the PR AUC score drops by 75.7% and 77.1% for the 1x1 and 7x7 configuration, respectively. There are many possible reasons for this reduction in performance. Most likely this is due to the "effective" size of the testing data set being too small. Likely, increasing the number of patches which the training data set draws from will increase the training performance.

10.3 Conclusion

Although the ASCAT NN initially appeared to work very effectively, this was due to contamination in the testing data brought on by the similarities between neighboring UHR WVCs from the same collocation patch. NNs designed for rain detection using ASCAT data separated by patches have poor performance. Future rain detection NNs will likely perform better if the amount of "effective" data is increased.

CHAPTER 11. CONCLUSION

A rain GMF, and NN are explored in an effort to provide more accurate global UHR wind estimation using ASCAT. The rain GMF accounts for the effect of rain on radar backscatter return and allows for better estimation of wind speed when in the presence of rain. The NN attempts to identify the presence of rain using only data from ASCAT. This enables the rain GMF to be used most effectively to reduce the total wind speed estimation error. While the rain GMF was successful at lowering the wind speed error of rainy samples by 13.6%, rain detection when coupled with this rain GMF increased the total wind speed RMSE by 0.23%. Had the NN been a perfect estimator, the wind speed RMSE would have dropped by 1.5%. Although the NN's performance is poor, by discarding all samples labeled as rain by the NN, the wind speed RMSE decreases by 2.83%. This comes at the cost of loosing 6% of ASCAT data.

11.1 Rain GMF Summary

A simple model is used to define the influences which affect radar backscatter,

$$\sigma^0 = \sigma_w \alpha(R;\theta) + \sigma_e(R;\theta), \tag{11.1}$$

where σ^0 is the measured backscatter, σ_w is the backscatter from wind induced capillary waves on the ocean surface, and α and σ_e are functions of rain rate and incidence angle which account for rain's influence on backscatter. The most effective rain GMFs fit $\sigma_e(R;\theta)$ and $\alpha(R;\theta)$ using an unrestricted nonlinear Nelder-Mead minimization. It was found that σ_w could be modeled directly by the sampled σ^0 or CMOD5 and still achieve very high performance. Methods of achieving this high performance depend on how σ_w is modeled.

The performance of wind estimation and rain detection was found to vary with incidence angle. As shown in Fig. 11.1 and Fig. 11.2, wind estimation performs best at low incidence angles while rain estimation performs best at high incidence angles. Fig. 11.1 shows how the DDWB and CMOD5 model of σ_w can both create rain GMFs which greatly reduce wind speed error. Fig. 11.2 shows that the rain-efficient GMF is significantly more accurate at rain estimation than other rain GMFs, particularly at lower incidence angles.

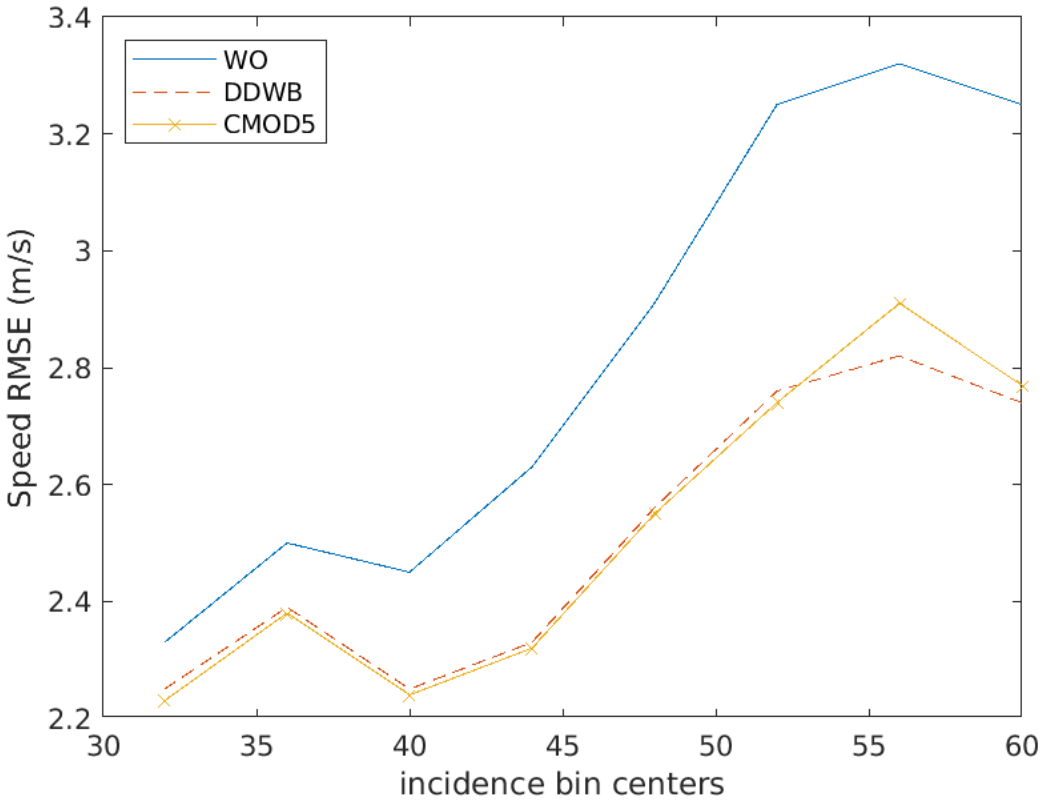


Figure 11.1: Wind speed RMSE for estimates derived from the WO model alone, and two SWR models using DDWB and CMOD5 respectively. The two SWR models share the same trend along the incidence angle as the WO model, but both reduce the error. Only samples affected by rain are tested here.

Several different rain GMFs were found that can lower the wind speed RMSE. The lowest RMSE for wind speed was 2.61 m/s, which is 13.6% better than using the WO GMF only. Nearly all rain GMFs created have rain RMSE values that are too high to be useful.

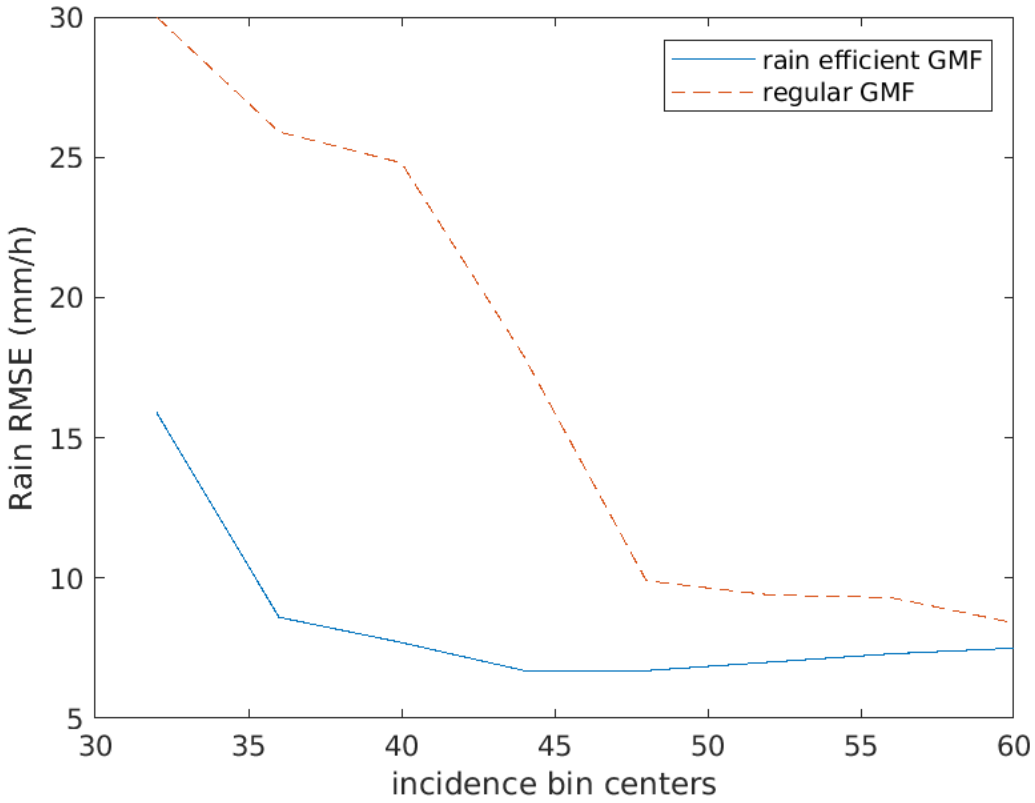


Figure 11.2: Rain RMSE for an average rain GMF and the rain-efficient GMF which is an outlier for rain estimation.

Though less than ideal, a few rain GMFs have significantly better rain estimation, the best of which yields a rain RMSE of 7.9 mm/h.

11.2 NN Summary

I tested the system's performance by using the best performing 1x1 NN to sort test samples into rainy and non-rainy classifications. NN performance is quantified by the PR AUC metric. The WO and SWR retrieval processes are applied to the non-rainy and rainy labeled samples, respectively. When the data set (data from a different year than the training data) is processed using only the WO GMF, the wind speed RMSE is 1.326 m/s and represents the basis case I compare the systems against. The wind speed RMSE of the entire data set when processed using the NN and optimal rain GMF was 1.329 m/s. As discussed in Section 7.3.3, an ideal NN used with this rain GMF can produce a 1.5% improvement

to the overall wind speed RMSE, but because the NN is suboptimal this method increases error instead of lowering it.

Using the "medium" rain GMF, which estimates wind speed better than the WO algorithm even without rain present, the NN is still found to be unhelpful. The medium performing rain GMF discussed in Section 7.3.3 was able to reduce the RMSE by 0.51% when samples labeled as rain are processed by SWR and the rest are processed by the WO algorithm. The wind speed RMSE, however, is reduced by 0.8% when the NN is disregarded and the medium SWR GMF is applied to all samples.

Although the NN was not effective working with SWR, it can still be used to lower the wind speed RMSE. By discarding all samples identified by the NN to contain rain, the remaining data, after being processed by the WO algorithm, has a wind speed RMSE 2.83% lower than if no data had been removed. The cost of this processes is that discarding the samples labeled as rainy removes 6% of all the data, even though only 10.6% of the removed data was correctly identified. For comparison, had the NN been ideal, the wind speed RMSE would lower by 4.58% at the cost of removing all rainy samples which make up 2.2% of the testing data set.

11.3 Future Work

There are several aspects of rain detection and the SWR GMF which should be explored in the future:

1. Rain Detecting NN Adjustments

   The scope of this thesis is limited by finite time and limited experience. Rain detection might be improved by increasing the "effective" size of the data set by including more collocation patches. Also, performance might be improved by treating entire collocation patches as "images" and detecting the location of rain inside them.

2. Changing the Purpose of the NN

   The goal of the NN was to locate the presence of rain. It is likely that the RMSE for wind speed can be reduced if the NN attempts to identify the ideal GMF for a WVC. NNs that estimate the wind speed and rain rate directly might also be worth studying.

3. Rain-Focused SWR Models

   Because of poor rain estimation performance, rain RMSE did not play an influential part during the design processes for a rain GMF. It is very likely that better rain estimation is possible if it is made a priority.

4. SWR Performance Parameterization

   ASCAT SWR models are judged by the effectiveness of the wind speed for the closest ambiguity. This, however, overlooks the directionality of the estimate. This also overlooks any affects which the SWR model might have on the ambiguity selection processes.

REFERENCES

[1] F. T. Ulaby, R. K. Moore, and A. K. Fung, *Microwave Remote Sensing.* Artech House, 1981, vol. 1. 1

[2] R. F. Contreras and W. J. Plant, "Surface effect of rain on microwave backscatter from the ocean: Measurements and modeling," *Journal of Geophysical Research: Oceans*, vol. 111, no. C8, 2006. 2

[3] C. Nie, "Wind/rain backscatter modeling and wind/rain retrieval for scatterometer and synthetic aperture radar," Ph.D. dissertation, Brigham Young University, 2008. [Online]. Available: https://www.proquest.com/dissertations-theses/wind-rain-backscatter-modeling-retrieval/docview/2504521782/se-2 2

[4] M. P. Owen, "Scatterometer contamination mitigation," Ph.D. dissertation, Brigham Young University, Provo, UT, 2010. 2, 6, 20, 55, 56, 64, 65

[5] H. Hersbach, A. Stoffelen, and S. de Haan, "An improved C-band scatterometer ocean geophysical model function: CMOD5," *Journal of Geophysical Research: Oceans*, vol. 112, no. C3, 2007. 4

[6] NSCAT Science Working Team Subcommittee on Geophysical Model Functions, "Current progress in Ku-band model functions," no. MERS 96-002. Brigham Young University Department of Electrical and Computer Engineering, August 1996. 4

[7] L. L. Scharf, *Statistical Signal Processing: detection, estimation, and time series analysis.* Addison-Wesley, 1990. 7

[8] W. A. H. Mark A. Richards, James A. Scheer, *Principles of Modern Radar.* SciTech, 2010, vol. 1. 9, 15

[9] D. G. Long, J. B. Luke, and W. Plant, "Ultra high resolution wind retrieval for seawinds," in *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No. 03CH37477)*, vol. 2. IEEE, 2003, pp. 1264–1266. 9, 10

[10] International Telecommunication Union, "Recommendation for specific attenuation model for rain for use in prediction methods," no. ITU-R P.838-3. ITU Radiocommunication Assembly, 2005. 15, 31

[11] TRMM Precipitation Radar Team, "Tropical rainfall measuring mission (TRMM), precipitation radar algorithm instruction manual for version 6." NASA, 2005. 15

[12] R. M. Toshio Iguchi, Shinta Seto, "GPM/DPR level-2, algorithm theoretical basis document," 2010. [Online]. Available: https://gpm.nasa.gov/sites/default/files/2022-06/ATBD_DPR_V07A.pdf 31

[13] C. Nie and D. G. Long, "A C-band wind/rain backscatter model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 3, pp. 621–631, 2007. 36

[14] Matlab, "Fminsearch," 1984. [Online]. Available: https://www.mathworks.com/help/matlab/ref/fminsearch.html 44

[15] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv preprint arXiv:2106.11342*, 2021. 56, 57, 63, 70, 71, 76

[16] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608089900208 60

[17] D. W. Draper and D. G. Long, "Evaluating the effect of rain on SeaWinds scatterometer measurements," *Journal of Geophysical Research: Oceans*, vol. 109, no. C12, 2004. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2002JC001741 63

[18] D. Draper and D. Long, "Simultaneous wind and rain retrieval using seawinds data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 7, pp. 1411–1423, 2004. 64

[19] D. W. Draper, "Wind scatterometry with improved ambiguity selection and rain modeling," Ph.D. dissertation, Brigham Young University, 2003. [Online]. Available: https://www.proquest.com/dissertations-theses/wind-scatterometry-with-improved-ambiguity/docview/305344849/se-2 64

[20] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://keras.io 71, 73, 75

[21] Udacity, "Going further with CNNs," 2022. [Online]. Available: https://learn.udacity.com/courses/ud187/lessons/4a041ac9-3bb9-43d6-8a02-4fa912626028/concepts/721ed0c1-85b3-4091-b84d-e37b0a6f436a 75

[22] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, no. 3, pp. 1–21, 03 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0118432 83