

An Exploration of Neural Networks in Enhanced  
Resolution Remote Sensing Products

Jordan Paul Brown

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  
Greg Nordin  
William Harrison

Department of Electrical and Computer Engineering  
Brigham Young University

Copyright © 2019 Jordan Paul Brown  
All Rights Reserved

## ABSTRACT

### An Exploration of Neural Networks in Enhanced Resolution Remote Sensing Products

Jordan Paul Brown

Department of Electrical and Computer Engineering, BYU  
Master of Science

Scatterometry and radiometry are used to obtain measurements of Earth properties with extensive spatial coverage at daily or near-daily temporal resolution. Their measurements are used in many climate studies and weather applications, such as iceberg tracking, ocean wind estimation, and volumetric soil moisture measurements. The spatial resolution of these data products ranges from a few kilometers to tens of kilometers. Techniques to enhance the spatial resolution of these products help reveal finer scale features, but come at the cost of increased noise.

This thesis explores the application of neural networks as a possible method to handle the noise and uncertainty in enhanced resolution scatterometer and radiometer data products. The specific sensors discussed are the Advanced Scatterometer (ASCAT) and its Ultrahigh Resolution (UHR) winds, and the Soil Moisture Active Passive (SMAP) radiometer and its soil moisture measurements. ASCAT UHR winds have already been validated in previous studies [1], but inherent ambiguity in the wind retrieval model couples with higher noise levels to decrease overall accuracy. Neural networks are tested as an alternate modeling method to possibly improve the accuracy compared with the current method. It is found that the feed forward neural networks tested are able to accurately estimate winds in most calculations, but struggle with the same ambiguity that occurs in the current model. The neural networks handle this ambiguity inconsistently, which results in worse overall network performance compared to the current wind retrieval method.

For the SMAP soil moisture measurements, the radiometer form of the Scatterometer Image Reconstruction algorithm is validated as a method to enhance resolution. While the increased noise at higher resolution does worsen overall accuracy, the performance remains within about  $0.04 \text{ cm}^3 \text{ cm}^{-3}$  RMSE of a validated soil moisture product, suggesting that fine scale features revealed as resolution is enhanced are accurate. Corrections to the soil moisture extraction model used in these tests could further improve these results. Neural networks are then applied and compared with the theory-based approach to extract soil moisture from the brightness temperature measurements, and are found to give slightly more accurate results than the theoretical model, though with somewhat higher error variance.

Keywords: scatterometer, radiometer, neural network, backscatter, radar cross section, brightness temperature

## ACKNOWLEDGMENTS

I would first like to acknowledge my wife Brooke and baby Wesley, as well as my parents and siblings for their love and support. I would also like to acknowledge Dr. Long and all of my friends in the lab for helping me be imMERSEd in a great graduate experience.

## TABLE OF CONTENTS

<b>TABLE OF CONTENTS</b> . . . . .	<b>iv</b>
<b>LIST OF TABLES</b> . . . . .	<b>vi</b>
<b>LIST OF FIGURES</b> . . . . .	<b>vii</b>
<b>Chapter 1 Introduction</b> . . . . .	<b>1</b>
1.1 Introduction to Neural Networks . . . . .	1
1.2 Scatterometry and Wind Retrieval . . . . .	3
1.3 Radiometry and Soil Moisture . . . . .	4
1.4 Summary of Results . . . . .	7
1.5 Thesis Organization . . . . .	7
<b>Chapter 2 Background</b> . . . . .	<b>9</b>
2.1 ASCAT UHR Wind Retrieval . . . . .	9
2.2 SMAP Soil Moisture Retrieval . . . . .	12
2.2.1 rSIR Algorithm . . . . .	14
2.2.2 Soil Moisture Extraction . . . . .	15
2.2.3 Soil Moisture Algorithm Limitations . . . . .	18
2.3 Neural Networks . . . . .	18
2.3.1 Information Extraction . . . . .	19
<b>Chapter 3 Neural Networks Applied to ASCAT UHR Wind Retrieval</b> . . . . .	<b>28</b>
3.1 Network Selection and Validation . . . . .	28
3.1.1 Theory . . . . .	28
3.1.2 Training . . . . .	31
3.1.3 Network Testing . . . . .	33
3.1.4 Analysis . . . . .	40
3.2 Extraction of the Neural Network Transfer Function . . . . .	43
<b>Chapter 4 Neural Networks in SMAP Soil Moisture Retrieval</b> . . . . .	<b>47</b>
4.1 rSIR Enhanced Resolution Soil Moisture . . . . .	47
4.1.1 Validation Method . . . . .	47
4.1.2 Validation Results and Analysis . . . . .	50
4.2 Network Selection and Validation . . . . .	51
4.2.1 Training . . . . .	52
4.2.2 Network Testing . . . . .	52
4.2.3 Analysis . . . . .	56
<b>Chapter 5 Conclusion</b> . . . . .	<b>59</b>
5.1 Conclusion . . . . .	59
5.2 Contributions . . . . .	60

5.3 Future Work . . . . .	62
<b>REFERENCES . . . . .</b>	<b>64</b>

## LIST OF TABLES

3.1	RMSE for wind speed neural networks with nine inputs . . . . .	35
3.2	RMSE for wind speed neural networks with 33 inputs . . . . .	36
3.3	RMSE for wind direction neural networks with nine inputs . . . . .	40
3.4	RMSE for wind direction neural networks with 34 inputs . . . . .	43
4.1	RMSE for the theory-based soil moisture extraction method at several resolutions . . .	50
4.2	Mean error for the theory-based soil moisture extraction method at several resolutions .	51
4.3	RMSE for the neural network soil moisture extraction at 36 km resolution . . . . .	56
4.4	RMSE for the neural network soil moisture extraction at 3 km resolution . . . . .	58

## LIST OF FIGURES

1.1	A chart of different neural network types . . . . .	2
1.2	A visualization of the Metop-A satellite . . . . .	4
1.3	An example drought danger map of the United States . . . . .	5
1.4	A visualization of the SMAP instrument . . . . .	6
2.1	The ASCAT beam geometry on the ground . . . . .	10
2.2	Signal interactions with the ocean surface . . . . .	10
2.3	Example ASCAT UHR wind product . . . . .	12
2.4	A visualization of the SMAP radiometer footprint . . . . .	13
2.5	A SMAP brightness temperature resolution enhancement example . . . . .	15
2.6	Breakdown of brightness temperature sources . . . . .	16
2.7	Example of brightness temperature and resultant soil moisture. . . . .	19
2.8	A visual representation of a feed forward neural network . . . . .	20
2.9	Neural network node expanded view . . . . .	23
3.1	The hyperbolic tangent activation function . . . . .	29
3.2	The boxcar-like function, made from hyperbolic tangents . . . . .	30
3.3	Function approximation using the boxcar-like functions . . . . .	31
3.4	Training data used for the wind retrieval neural networks . . . . .	32
3.5	Example wind speed maps from NWP models, CMOD5 processing, and neural network models . . . . .	37
3.6	A second example of wind speed maps from NWP models, CMOD5 processing, and neural network models . . . . .	38
3.7	A visualization of the target pixel and neighboring pixels used in the 33 and 34 input neural networks . . . . .	39
3.8	A comparison of wind direction results from ECMWF NWP, CMOD5 with ambiguity selection, a 9 input neural network, and a 34 input neural network . . . . .	41
3.9	A second comparison of wind direction results from ECMWF NWP, CMOD5 with ambiguity selection, a 9 input neural network, and a 34 input neural network . . . . .	42
3.10	Example values for a node in a wind speed neural network . . . . .	45
3.11	Example values for the logic-rule approximation of the node in Figure 3.10 . . . . .	45
4.1	Soil moisture resolution comparison maps of eastern Africa . . . . .	48
4.2	Soil moisture resolution comparison maps of central South America . . . . .	49
4.3	Example ascending passes from soil moisture neural network training data . . . . .	53
4.4	Example descending passes from soil moisture neural network training data . . . . .	54
4.5	Comparison of a 36 km neural network derived soil moisture map with a reference map . . . . .	55
4.6	Comparison of a 3 km neural network derived soil moisture map with a reference map . . . . .	57

## **CHAPTER 1. INTRODUCTION**

Over recent decades, remote sensing has become increasingly popular in weather prediction and the study of natural hazards and climate features [2]. Scatterometry contributes through the use of active satellite-based sensors which measure microwave radar scattering off the earth's surface, and radiometry uses passive satellite-based sensors which measure the microwave thermal emission from the earth. Scatterometers and radiometers enable global weather and climate measurements with daily or near-daily temporal coverage, at spatial resolutions from a few kilometers to a few tens of kilometers. Improved signal processing algorithms in recent years have enabled more accurate, higher resolution measurements than previously possible. In both scatterometry and radiometry, however, the enhanced resolution comes at the cost of increased noise, which leads to increased uncertainty and more inaccuracies in the derived data products. This thesis explores the application of neural networks to the derived enhanced resolution data products, specifically wind and soil moisture, and their potential to improve product accuracy.

### **1.1 Introduction to Neural Networks**

Artificial neural networks, often simply referred to as neural networks, are biologically-inspired computing systems designed by a flexible, automated means to model a relationship between input data and target outputs. Some common applications include image feature or pattern recognition, data classification, and mathematical model approximation. There are different types of neural networks, but all consist of nodes which represent the input, intermediate, and output values throughout the network; weighted paths connecting the nodes; and functions which are applied to the values at each node, modifying the value in some desired way. The nodes are organized into layers, the first of which is known as the input layer, the last is known as the output layer, and all layers in between are called hidden layers. The networks are adjusted for better performance in each application by varying the number of nodes and layers, tuning the weights and bias terms

associated with each connection between nodes, modifying the pattern of connections between nodes, and changing the type of activation function being applied.

A few common forms of neural networks can be seen in Figure 1.1. These example network frameworks each have uses in different modeling applications and are described, along with several others not shown, in [3]. The networks in this thesis are all based on the feed-forward framework, which is commonly used in modelling numerical functions. Feed forward neural networks are explained in greater detail in Chapter 2, further motivating their use in this application.

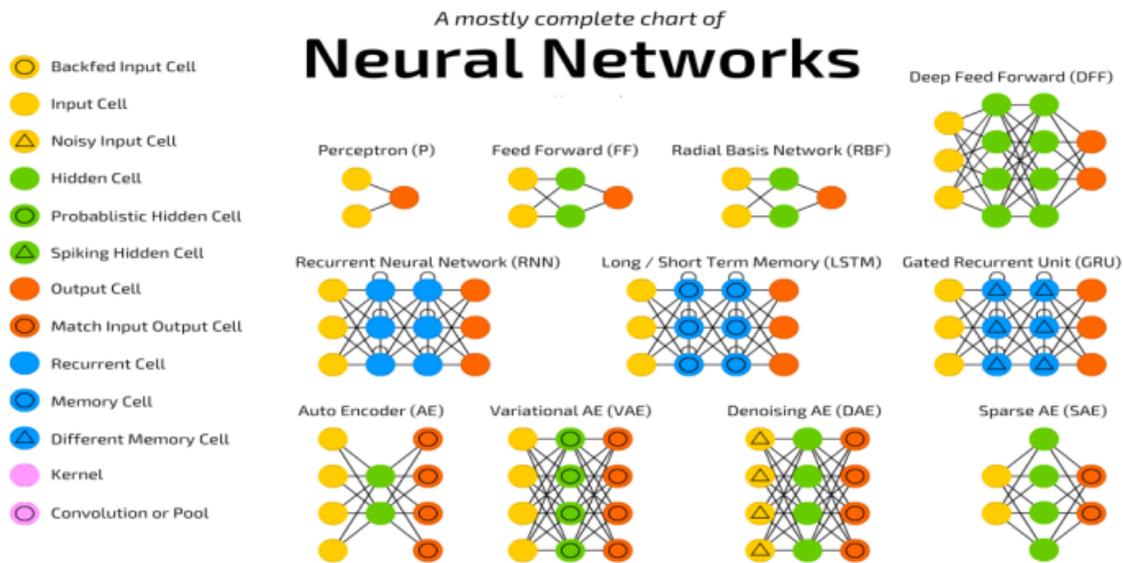


Figure 1.1: A chart of a few of the different neural network frameworks available. Beyond the different frameworks, the numbers of nodes, layers, etc. can be varied in order to find a network which fits the relevant application. This chart was taken from [4].

Once the specific architecture of the network is selected, the weights and biases are initialized, and a set of training data (example inputs and corresponding target outputs) is introduced to the network. The network then iterates through the training set, adjusting its model weights and biases after each iteration to more closely estimate the target outputs. This is continued for a desired amount of time, or until a desired model accuracy is reached.

Theoretically, neural networks have the potential to model any calculable relationship to any desired degree of accuracy, given enough nodes and layers. This is an idea developed further in Chapter 2. However, despite their impressive modelling power, neural networks are somewhat

of a black-box approach, meaning that the actual information learned by the network is difficult to understand or represent in a more transparent way, especially as we add nodes and layers to the architecture. They also can vary significantly given different initializations, training data, and other training parameters. For many applications, these facts do not pose a problem as long as the network output performance is accurate; for others, this discourages the use of neural networks, as the difficulty to replicate their models or predict their behavior in interpolation and extrapolation leaves too much uncertainty.

This thesis explores different aspects of applying feed forward neural networks to the wind and soil moisture retrieval problems. I examine the potential accuracy of neural network derived models in these areas, as well as the possibility of extracting information from neural networks to inform future modeling improvements.

## **1.2 Scatterometry and Wind Retrieval**

Wind near the sea surface plays a large role in weather and wave prediction, but obtaining sufficient in situ measurements for these applications at decent temporal and spatial resolutions is impractical, if not impossible. In 1974, this problem was addressed through the use of remote sensing, when the S-193 scatterometer on Skylab successfully demonstrated the potential of radar in retrieving near-surface wind speed from space. This led to the use of the first satellite-based scatterometer intended for wind retrieval, carried on the Seasat satellite launched in 1978 [2]. Since then, wind retrieval scatterometer systems and algorithms have continued to improve, resulting in wind products with increased resolution and accuracy.

One such instrument currently used in scatterometer wind retrieval is known as the Advanced Scatterometer (ASCAT), which was included aboard three separate systems (Metop-A launched in 2006, Metop-B launched in 2012, and Metop-C launched in 2018) from the European Space Agency (ESA) and European Organization for the Exploitation of Meteorological Satellites (EUMETSAT). The Metop satellite design is visualized in Figure 1.2. Together, the instruments have provided nearly continuous radar data contributing to studies of ice, soil moisture, and wind since 2007. Recent resolution enhancement advancements in wind data specifically have led to new products like the Ultra High Resolution (UHR) ASCAT wind product [1]. The ASCAT UHR winds give us a better look at small scale wind patterns, but also introduce increased noise. This

extra noise and uncertainty introduced in enhanced resolution wind products, together with some inherent ambiguity in the wind retrieval model, can lead to increased noise relative to the lower-resolution counterparts.

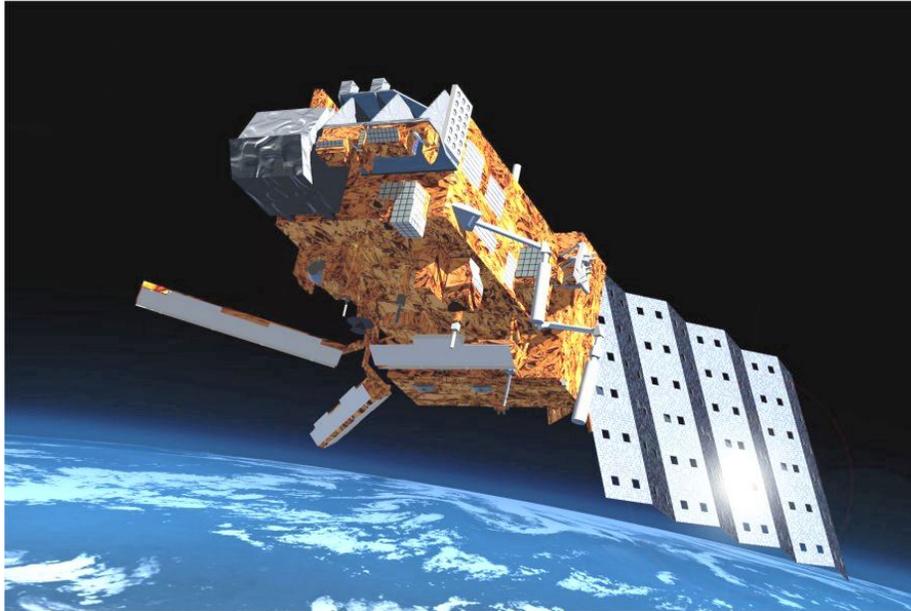


Figure 1.2: A visualization of the Metop-A satellite, with the first ASCAT on board. ASCAT consists of the three beam-like antennas in a fan-beam shape on the bottom of the system, with processing components inside the main body of the satellite. This figure was taken from [5].

This thesis explores the application of neural networks in an attempt to improve scatterometer wind retrieval with two different methods. The first is to use the networks in a traditional way to see if we can train a network to perform more accurately than the current ASCAT wind retrieval model. The second is to train a simple neural network to recreate the wind retrieval model with sufficient accuracy, then attempt to extract information from the network to inform model improvements.

### 1.3 Radiometry and Soil Moisture

Spaceborne radiometers have been used slightly longer than scatterometers, with the first earth-observing spaceborne sensor on board the Russian satellite Cosmos 243 in 1968 [2]. Radiometer data has been used in a wide variety of weather and climate studies, as well as military

applications. The introduction of synthetic-aperture radiometers, or interferometers, in the 1990s provided the possibility for much higher resolutions in radiometry than previously provided, which improved the usefulness of derived data products like sea surface salinity and soil moisture [2].

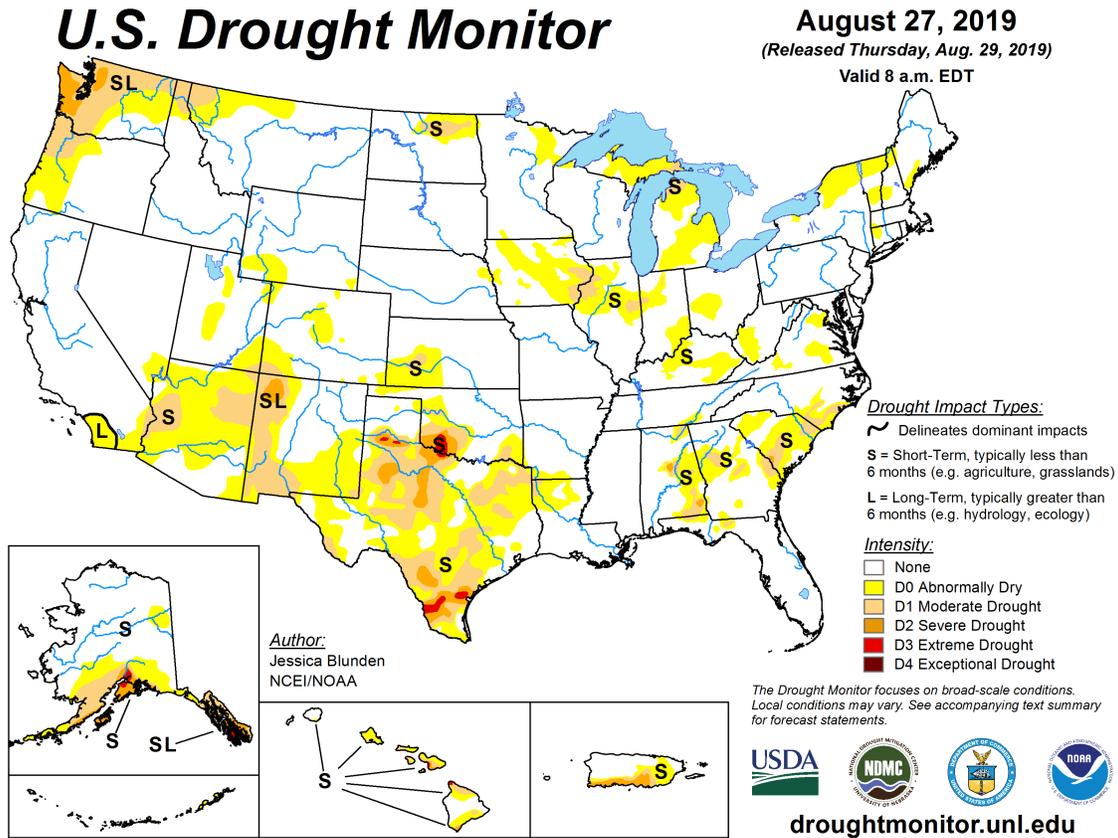


Figure 1.3: An example map of droughts in the United States. Soil moisture data from SMAP can be used to improve the resolution and accuracy of maps like this. This figure was taken from [6].

Later in 2007, an increased interest in soil moisture retrieved through remote sensing was evident in an Earth Science Decadal Survey performed by the National Research Council. The council described among its research priorities for the next decade a need for improved flood prediction and drought monitoring (example in Figure 1.3), improved weather forecasting, and enhanced understanding of the role of surface water in the energy and carbon cycles [7]. This led to the creation of the Soil Moisture Active Passive (SMAP) mission in 2008, with the SMAP satellite

finished and launched in 2015. The SMAP team addresses the priorities mentioned previously by tracking global soil moisture levels daily through scatterometry and radiometry.

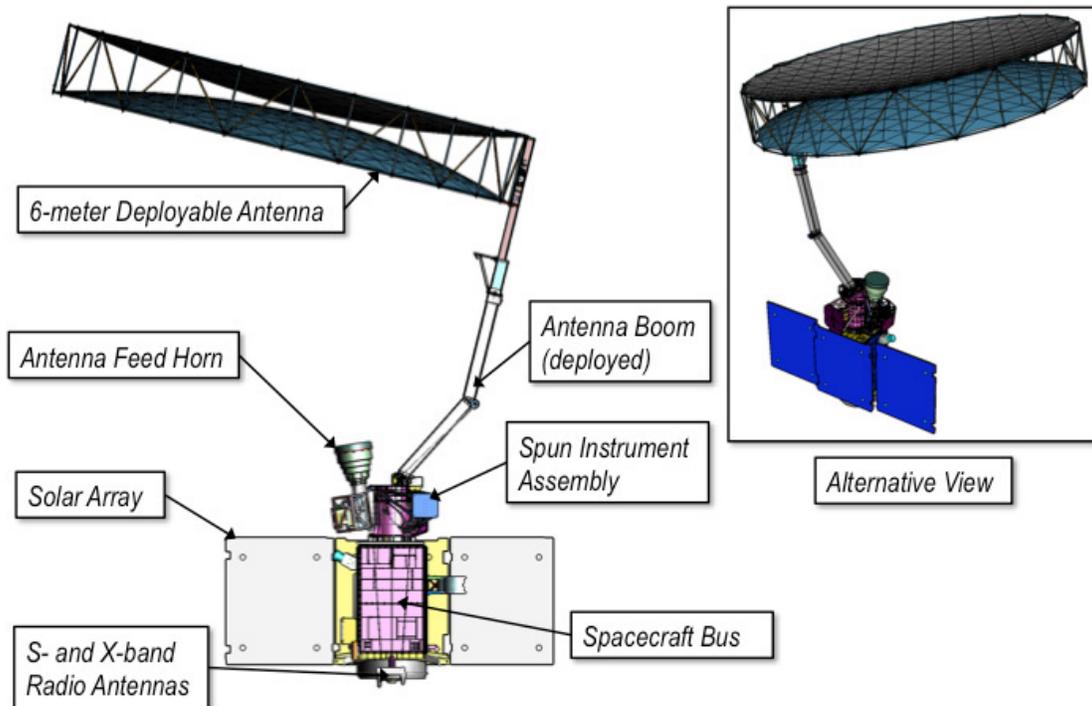


Figure 1.4: An illustration of the SMAP system from [8].

The SMAP instrument, visually represented in Figure 1.4, is unique in that it was designed to use both a radar (active) and radiometer (passive) specifically for soil moisture retrieval. This design allows SMAP to exploit the benefits of both methods of soil moisture retrieval. Generally, radar-based soil moisture retrieval offers higher resolution images, while radiometer-based retrieval offers higher accuracy and sensitivity. Unfortunately, the radar on the SMAP system failed on July 7, 2015, shortly after the instrument started returning data, and is no longer in use. Thus current SMAP data processing methods use only passive data. The baseline resolution of the resulting data products is that of the passive data, much greater than the mission goals with the active contribution [7]. This failure motivated an interest in image resolution enhancement techniques in order to continue the data products at the original desired resolutions.

One resolution enhancement method is known as the radiometer form of the Scatterometer Image Reconstruction algorithm (rSIR), developed at Brigham Young University, and described in detail in [9]–[11]. While the technique offers promise in achieving or surpassing the resolution goals of the SMAP mission, it also increases noise and uncertainty which carries over into the derived soil moisture product. The uncertainty in the higher resolution soil moisture products motivates this exploration of the application of neural networks to the problem. In this thesis, I show the application of the rSIR algorithm to the original brightness temperature data, and the resultant enhanced resolution data. I then compare two methods of extracting soil-moisture from the enhanced resolution data: one is a theory-based approach, and the other is a neural network modelling approach.

#### **1.4 Summary of Results**

In the case of wind retrieval, it is found that feed forward neural networks struggle with consistently resolving ambiguity. They were able to accurately model the training data as well as certain test regions, but had occasional high error in areas of new wind patterns or high noise. It is also found that, for both wind speed and wind direction, the information extracted from an accurately trained neural network using the method in [12] is too convoluted to provide any extra clarity to the underlying model.

For soil moisture extraction, neural networks were found to have slightly better overall accuracy than a theory-based method at both 36 km and 3 km resolution. However, there was more variation in the error, which increased at the higher resolution.

#### **1.5 Thesis Organization**

This chapter has given a brief introduction to the topics discussed in this thesis, which are described in more detail through the remaining chapters. Chapter 2 is a background chapter separated into three sections: one discusses the theory behind neural networks and the information extraction algorithm I use, the second describes instrument and algorithm information for the scatterometry wind retrieval problem, and the third describes instrument and algorithm information for the radiometry soil moisture retrieval and resolution enhancement problems. Chapter 3 shows my

application of neural networks to ASCAT UHR winds, and discusses the potential of extracting information from those networks to improve current models. Chapter 4 describes the enhanced resolution soil moisture product resulting from the rSIR algorithm, and compares the extraction of soil moisture using traditional methods versus neural networks. Chapter 5 describes conclusions drawn and the contributions of this thesis to the remote sensing field, then discusses possible future work on these topics.

## CHAPTER 2. BACKGROUND

This chapter presents background information and theory on the instruments and algorithms used throughout this thesis. The chapter is separated into three sections: first I describe the ASCAT wind retrieval algorithm, second I describe the SMAP passive soil moisture retrieval algorithm and method to enhance resolution, and finally I present some background on neural networks along with a description of the neural network information extraction algorithm.

### 2.1 ASCAT UHR Wind Retrieval

The Advanced Scatterometer was developed by ESA and EUMETSAT, and was designed specifically to measure ocean winds [13], though its uses have been expanded to things like storm, sea ice, and iceberg tracking, and soil moisture estimation [1], [2], [13]–[18]. It is a fan-beam scatterometer, with six vertically-polarized antennas that produce elongated coverage patterns on the earth with fore, mid, and aft azimuth angles for each of the two swaths (see Figure 2.1). The fore pattern is at  $45^\circ$  of the along-track direction, the mid pattern is at  $90^\circ$ , and the aft pattern is at  $135^\circ$ . These multiple azimuth looks allow for multiple backscatter measurements at a single point, which is required for accurate wind retrieval results [19]. ASCAT also operates in C-band at a frequency of 5.255 GHz, which enables all-weather, day-and-night measurements.

To retrieve the wind direction and speed from the backscatter measurement at a given location in the swath, a model is applied which is known as a geophysical model function (GMF). The overall idea of a GMF is to take advantage of wind interactions with the ocean waves, and in turn the signal's interactions with those waves. Generally, more wind leads to a rougher ocean surface, which increases the backscatter received by the antenna. Wind direction also changes the direction of waves on the ocean surface, which align differently with the incoming signal and vary the amount of backscatter. These ideas are illustrated in Figure 2.2.

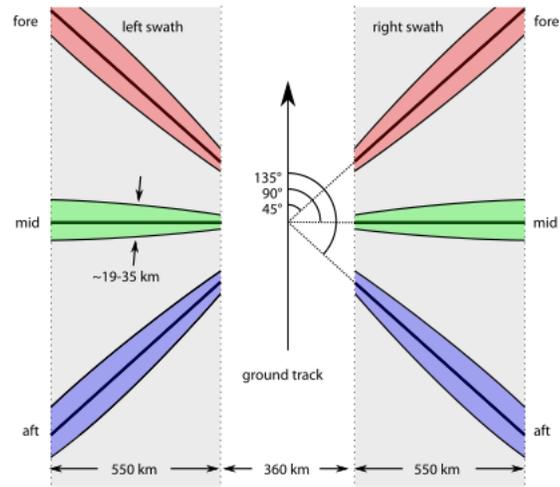


Figure 2.1: The ASCAT beam geometry as it appears on the ground. This figure is taken from [20].

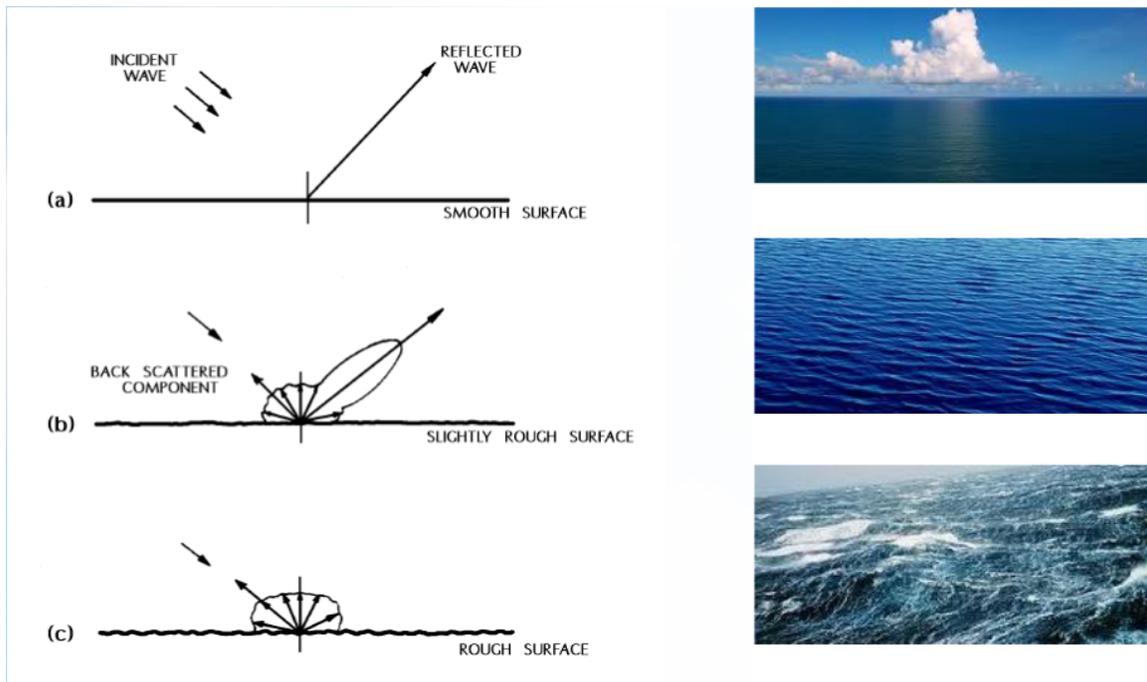


Figure 2.2: Visualization of a scatterometer signal interacting with waves to decrease or increase backscatter. This figure is taken from [21].

The specific model currently used by ASCAT ocean wind products is the CMOD5 GMF [19], defined by the equations

$$\sigma_0 = B_0(1 + B_1 \cos \phi + B_2 \cos 2\phi)^{1.6} \quad (2.1)$$

$$B_0 = 10^{a_0 + a_1 v} f(a_2 v, s_0)^\gamma \quad (2.2)$$

$$B_1 = \frac{c_{14}(1+x) - c_{15}v(0.5+x - \tanh[4(x+c_{16}+c_{17}v)])}{1 + e^{0.34(v-c_{18})}} \quad (2.3)$$

$$B_2 = (-d_1 + d_2 v_2) e^{-v_2} \quad (2.4)$$

$$x = (\theta - 40)/25. \quad (2.5)$$

The model is a function of wind speed ( $v$ ), wind direction (involved in  $\phi$ ), azimuth angle (also involved in  $\phi$ ), and incidence angle ( $\theta$ ), where  $\phi$  is the difference between wind direction and azimuth look angle, both measured from north. The  $c$  coefficients are all fixed values (listed in [19], [22]), and the  $a$ ,  $d$ ,  $s$ , and  $\gamma$  values are calculated from the stored  $c$  coefficients and the incidence angle. Given all these inputs, the model can then be used to estimate the wind for a backscatter measurement. The equations are described in detail in [19], [22], [23].

To retrieve the wind vector from CMOD5, the model  $\sigma_0$  output is compared with the observed normalized backscatter, and the wind direction and speed parameters (which together we call a wind vector) are adjusted until the error is minimized. This leads to multiple possible solutions, but by obtaining measurements of the same area from multiple different azimuth look angles, we narrow the possible solutions to the two to four most likely wind vectors, which are known as ambiguities [24]. Usually for ASCAT there are two ambiguities, differing in direction by about  $180^\circ$ . To obtain the final solution, an ambiguity selection process is applied to determine the most likely solution out of the possibilities. The current ambiguity selection process used for ASCAT products at ESA is known as 2D-VAR, which can be read about more in [25]. An example resultant UHR wind field can be seen in Figure 2.3.

Despite the overall success of the GMF and ambiguity selection process, there remain inaccuracies, whether due to excessive noise, limitations of the model, or limitations of the instrument. Chapter 3 describes the application of neural networks to this problem to observe their potential

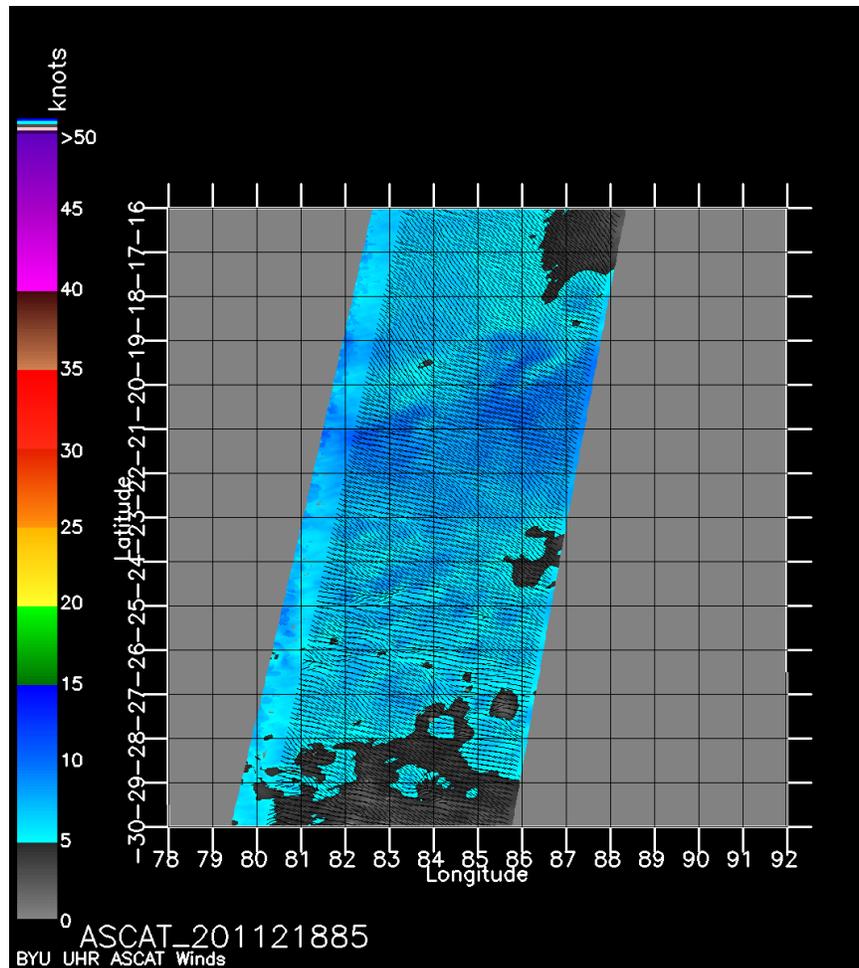


Figure 2.3: An example ASCAT UHR wind product in a region in the Indian Ocean, during day 7 of 2011. The small black quivers represent the wind direction, while the color denotes the wind speed.

in improving the ASCAT UHR accuracy. A well-trained neural network may be able to calculate wind given backscatter without the ambiguity of the GMF, and extracting information from that trained network could allow us to either improve the GMF or the ambiguity selection process.

## 2.2 SMAP Soil Moisture Retrieval

SMAP’s main objective, as stated in the mission handbook [7], is to “collect a 3-year data set that will be used to determine the moisture content of the upper soil and its frozen or thawed state, with global measurements every 3 days.” The specific goal for the moisture content is “to provide estimates of soil moisture in the top 5 cm of soil with an error of no greater than

0.04 cm<sup>3</sup> cm<sup>-3</sup> volumetric at 10 km spatial resolution and 3-day average intervals.” This excludes extreme areas, such as those with snow, frozen ground, or mountainous topography. The high-level measurement objectives are to: 1) have a 1000-km swath width at its orbit altitude of 685 km, 2) provide co-located L-band active radar measurements and passive radiometer measurements at an incidence angle near 40°, 3) provide radiometer measurements at spatial resolution of 40 km, and 4) provide dual-polarized radar measurements at spatial resolution of 3 km [7].

These goals are all met through the 6-meter deployable mesh reflector antenna used by both the radiometer and radar instruments, which share a single feed horn. SMAP is also the first spaceborne radiometer to fly a dedicated subsystem for detecting and correcting radio frequency interference (RFI) [7]. It is conically scanning and rotates at about 13.0 rpm, allowing it to maintain overlap of measurements in the along-track direction (see Figure 2.4). Both the radiometer and the radar receive signals at incidence angles between 35° and 50°. The radiometer specifically retrieves brightness temperature measurements at vertical and horizontal polarizations, as well as the third and fourth Stokes parameters, which are mapped to a 36-km resolution Equal-Area Scalable Earth grid (EASE-2) grid [26]. It operates in L-band, at 1.41 GHz, and has a 24 MHz bandwidth. This frequency, like the ASCAT instrument frequency, allows for results less affected by cloud cover and sunlight. The details of the radar are left to [7], [27].

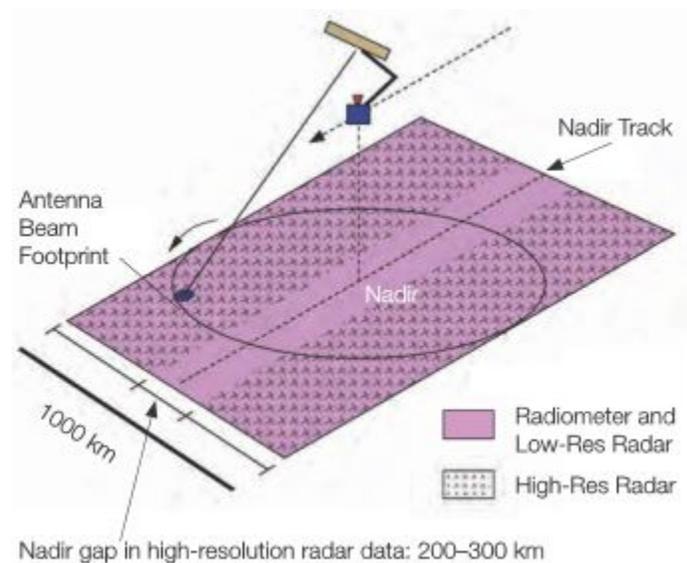


Figure 2.4: A visualization of the SMAP radiometer footprint as it looks on the earth, taken from [7].

In the rest of this section, I briefly detail the rSIR algorithm to enhance passive product resolution, followed by the SMAP passive soil moisture extraction algorithm. The algorithm descriptions are high-level, with some of the lower-level details left to documentation in [9]–[11] for rSIR, and [28], [29] for SMAP. The active soil moisture extraction algorithm, and method to combine the active and passive products for the original 9 km product, is described in [27], [30].

### 2.2.1 rSIR Algorithm

The rSIR algorithm takes advantage of irregular sampling theory and knowledge of the measurement response function (MRF) to construct an image of the SMAP brightness temperature measurements at a higher resolution than the baseline 36 km [9]–[11]. rSIR is a signal reconstruction algorithm which assumes that the original signal to be reconstructed in the image is bandlimited, the irregular sampling meets generalized Nyquist requirements, and the frequency response of the MRF is non-zero over the signal bandwidth. These requirements guarantee that, in the zero-noise case, rSIR can exactly estimate the original signal [9].

As a brief explanation, rSIR can be thought of as an iterative method to solve for the vector of actual surface brightness temperatures ( $\vec{a}$ ) in the equation,

$$\vec{T}_B = \mathbf{H}\vec{a}, \quad (2.6)$$

where  $\vec{T}_B$  is the observed brightness temperature measurements, and  $\mathbf{H}$  is the discrete samples of the MRF for each measurement [9]. rSIR results in an approximate maximum-entropy solution if the system is underdetermined, and a least-squares solution if it is overdetermined.

Since real brightness temperature measurements are not noise free, the full signal reconstruction can have significant noise effects. In order to provide regularization for this noise, the iterative solution of Equation 2.6 is stopped after a certain number of iterations. The goal of this regularization is to find a balance between signal reconstruction accuracy and noise enhancement [9], and can be varied for different applications.

Figure 2.5 shows an example brightness temperature image in a region of west Africa at the original 36 km resolution, then enhanced to 9 km and 3 km resolution using the rSIR algorithm.

The brightness temperature figures are zoomed in to help highlight the resolution enhancement provided by rSIR.

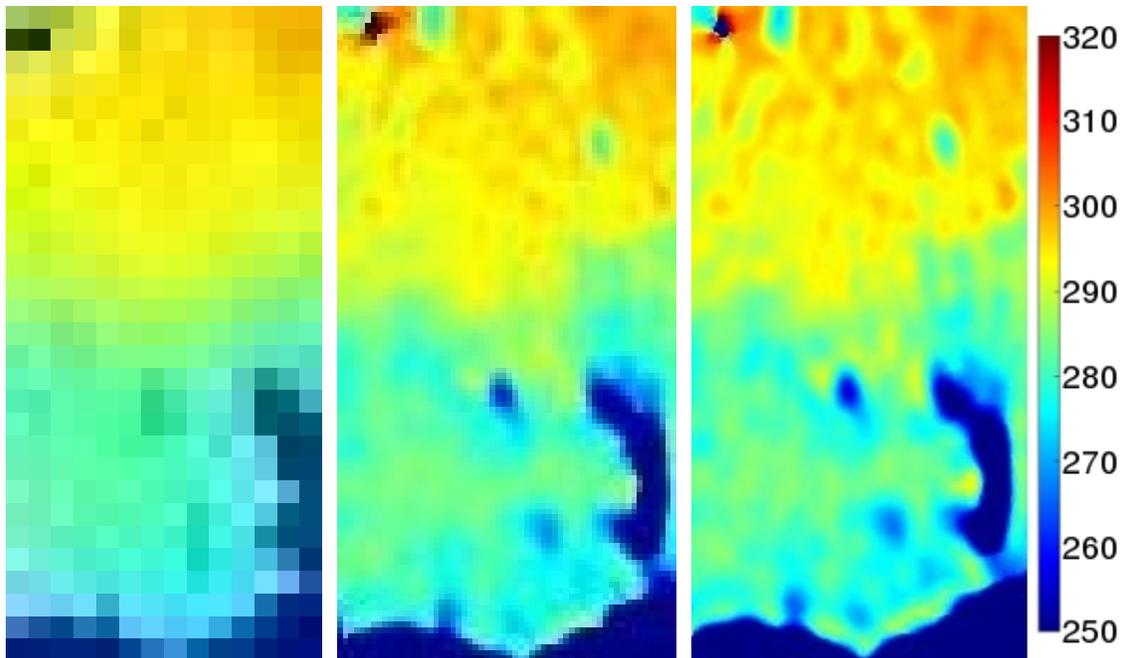


Figure 2.5: An example of the differences between resolutions for the SMAP brightness temperature product. The resolutions are (left) 36 km, (center) 9 km, and (right) 3 km. These are zoomed in on west America, and taken from day 300 of 2016. It can be seen that, as the resolution is increased, edges become more clear and finer scale patterns become more apparent. The color scale is in Kelvin.

### 2.2.2 Soil Moisture Extraction

The overall idea of the soil moisture retrieval algorithm is to isolate the thermal emission of the soil from the total signal received (see Figure 2.6), calculate the dielectric constant of the soil from that emission, then find the soil composition (including the soil moisture) which correspond with that dielectric constant.

The first step of this process is to gather the necessary data sets. Along with the brightness temperature, incidence angle, and retrieval quality data which come from the SMAP instrument, several ancillary data sets are needed: surface temperature, single scattering albedo, vegetation

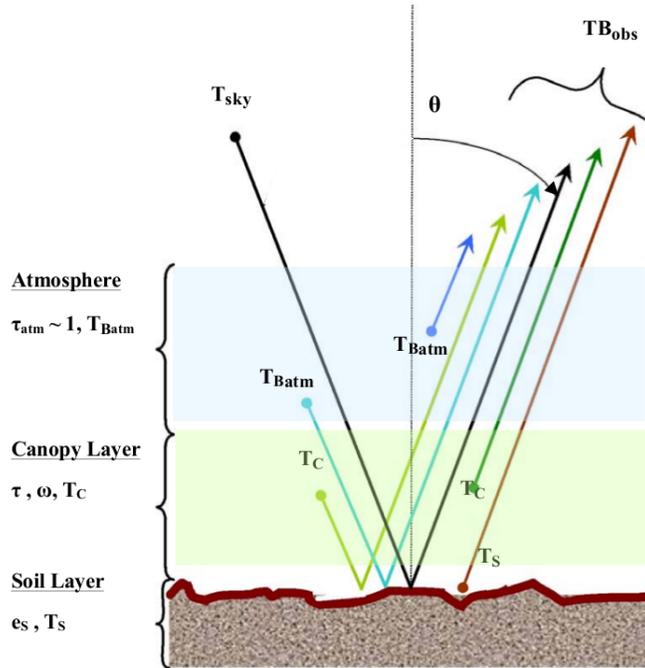


Figure 2.6: A visualization of the different emission sources contributing to the total received brightness temperature. In the soil moisture algorithm the effects of the soil emissivity are isolated, which shows up in the direct soil emission as well as the reflections off of the soil [28].

opacity, surface roughness, and soil texture (specifically clay fraction). Details about these ancillary data sets can be found in [28], [31]–[33]. For this description, we assume that all of these data sets have been mapped to the same resolution grid so the calculations can be performed on a pixel basis.

Once the necessary data is available, the NSIDC calculations begin with correction of the gridded brightness temperature for water bodies and unwanted emission sources. The goal of this is to have brightness temperature which represents only emissions over land in the target area. The antenna pattern correction (APC), and water body correction are described in [28], but not in sufficient detail to accurately recreate them here. However, there is a retrieval quality flag provided with the brightness temperature data which generally flags noisy or water-dominated pixels as unfit for soil moisture retrieval. For purposes here, I only compare algorithm performance in areas flagged as good quality.

For the soil moisture algorithm here, I use the uncorrected brightness temperature data. From the brightness temperature, the total emissivity ( $e^{\text{tot}}$ ) can be extracted. Both the brightness

temperature ( $T_B$ ) and surface temperature ( $T$ ) are reported in Kelvin, with the emissivity then calculated as

$$e^{\text{tot}} = \frac{T_B}{T}. \quad (2.7)$$

Next, the emissivity of the surface alone is computed by removing the contribution of canopy which covers or surrounds the target soil area. This is done according to the equation

$$e^{\text{surf}} = \frac{e^{\text{tot}} - 1 + \gamma^2 + \omega - \omega\gamma}{\gamma^2 + \omega\gamma - \omega\gamma^2}, \quad (2.8)$$

where

$$\gamma = e^{-\tau \sec(\theta)}. \quad (2.9)$$

The target surface emissivity is shown as  $e^{\text{surf}}$ ,  $\omega$  is the scattering albedo,  $\tau$  is the vegetation opacity, and  $\theta$  is the incidence angle. In the downloaded NSIDC datasets, the vegetation opacity maps already have the  $\sec(\theta)$  term incorporated into the reported value [29].

Next, the effects of surface roughness are removed to obtain the smooth surface emissivity according to the equation

$$e^{\text{smooth}} = 1 - (1 - e^{\text{surf}})e^{h \cos^2(\theta)}. \quad (2.10)$$

Here  $h$  is the surface roughness metric reported in the NSIDC dataset.

At this point in the algorithm, several possible soil moisture values (evenly spaced values in the range 0.02 to 0.5  $\text{cm}^3 \text{cm}^{-3}$  for 9 km and 36 km resolution, and 0.02 to 0.6  $\text{cm}^3 \text{cm}^{-3}$  for 3 km) are considered, and their corresponding smooth surface emissivities are calculated using a soil dielectric mixing model followed by Fresnel's equations. The dielectric mixing model used is known as the mineralogy-based soil dielectric model (MBSDM), described in detail and validated in [34]. MBSDM takes the possible soil moisture values as well as the local soil clay fraction, and approximates the corresponding dielectric constants. Next, those dielectric constants are passed into Fresnel's equations, which are written as

$$e_H(\theta) = 1 - \left| \frac{\cos\theta - \sqrt{\epsilon - \sin^2\theta}}{\cos\theta + \sqrt{\epsilon - \sin^2\theta}} \right|^2 \quad (2.11)$$

$$e_V(\theta) = 1 - \left| \frac{\epsilon \cos \theta - \sqrt{\epsilon - \sin^2 \theta}}{\epsilon \cos \theta + \sqrt{\epsilon - \sin^2 \theta}} \right|^2. \quad (2.12)$$

In most practical applications, the vertically-polarized brightness temperature is used for soil moisture retrieval, as well as the corresponding equation for  $e_V$ .

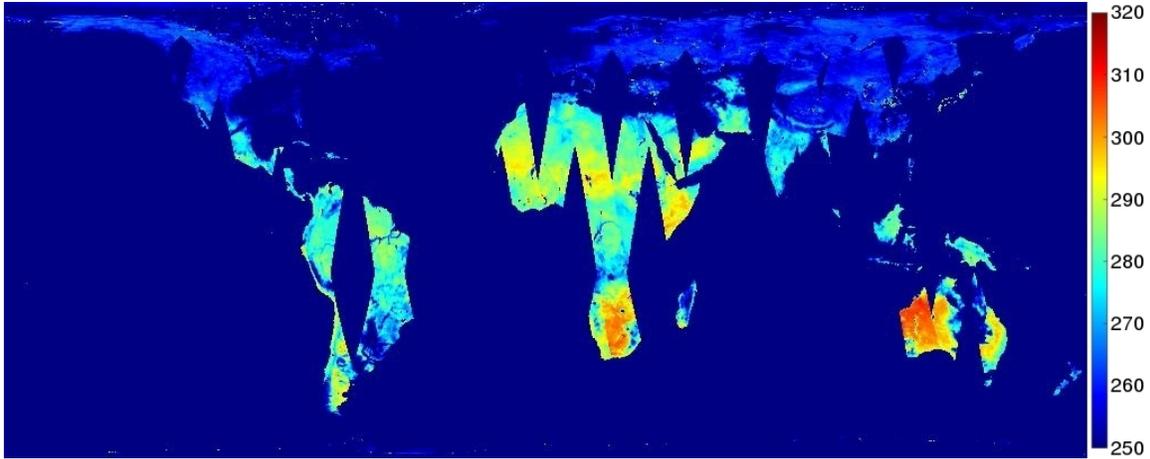
Finally, the soil moisture value which leads to the closest emissivity,  $e_V$ , to the one calculated directly from the received brightness temperature,  $e^{\text{smooth}}$ , is the reported soil moisture value for the relevant pixel. These calculations are repeated for each pixel, creating a full soil moisture map. An example of a global brightness temperature map and its resultant soil moisture map can be seen in Figure 2.7.

### 2.2.3 Soil Moisture Algorithm Limitations

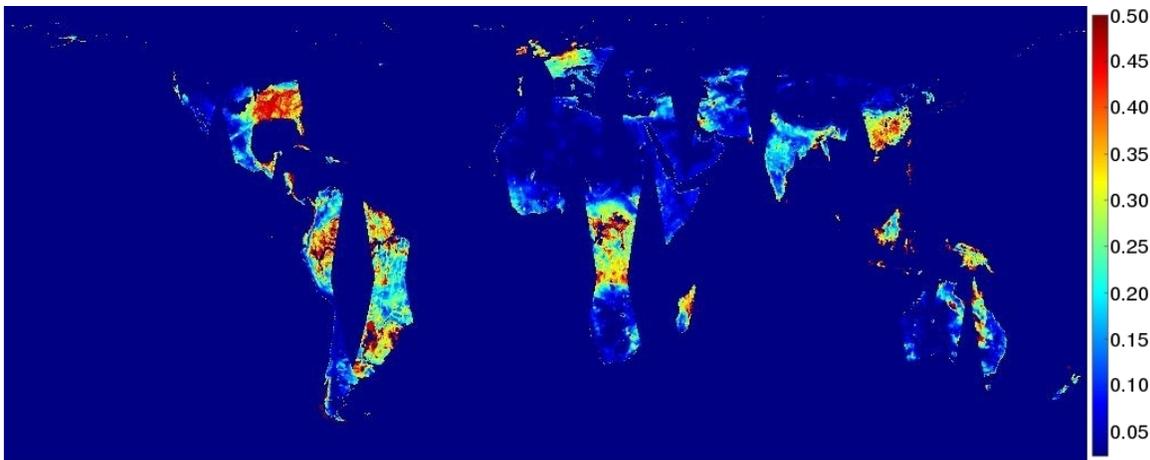
While there are always dark diamonds in the brightness temperature and soil moisture images which signify areas of no satellite coverage (see Figure 2.7), it can also be seen that there are several dark areas over land where brightness temperature was received, but soil moisture was not extracted. This occurs especially in the far north regions of the globe, and highlights the fact that there are several factors which result in poor performance of the retrieval algorithm. These factors include brightness temperature retrieval quality, soil freeze state, land cover, etc. For example, if the target pixel has excessive noise in its measurements, a nearby water body, frozen ground, or too dense of foliage covering the ground, the soil moisture accuracy is not trusted and is thus not reported. In general, however, consistent, reliable retrieval can be obtained between 45° North and 45° South.

## 2.3 Neural Networks

A neural network is composed of layers of nodes and connecting paths (visualized in Figure 2.8) which are trained to reproduce some desired calculation or classification. It accomplishes the calculations or classifications through a series of additions and multiplications, with the values modified by activation functions at several points to introduce non-linearity. Though the individual calculations performed in neural networks are simple, the layers and volume of calculations gen-



(a) Vertically-polarized brightness temperature (K).



(b) Soil moisture ( $\text{cm}^3 \text{cm}^{-3}$ ).

Figure 2.7: An example of a global map of brightness temperature (top) and soil moisture (bottom) for day 2 of 2016. The brightness temperature is reported in Kelvin, and the soil moisture is reported in  $\text{cm}^3 \text{cm}^{-3}$ . Observing the maps and color scales, we can visually validate the soil moisture map, with areas like northern Africa appearing extremely dry, whereas the Amazon region appears very moist.

erally convolute the underlying model, resulting in the black-box effect mentioned in Chapter 1. This section describes the method which I use to extract information from a neural network model in order to attempt to represent the model in a more transparent form.

### 2.3.1 Information Extraction

To confront the black-box nature of neural networks, much research has been done to try and extract information from a trained neural network and represent it in the form of logic-based

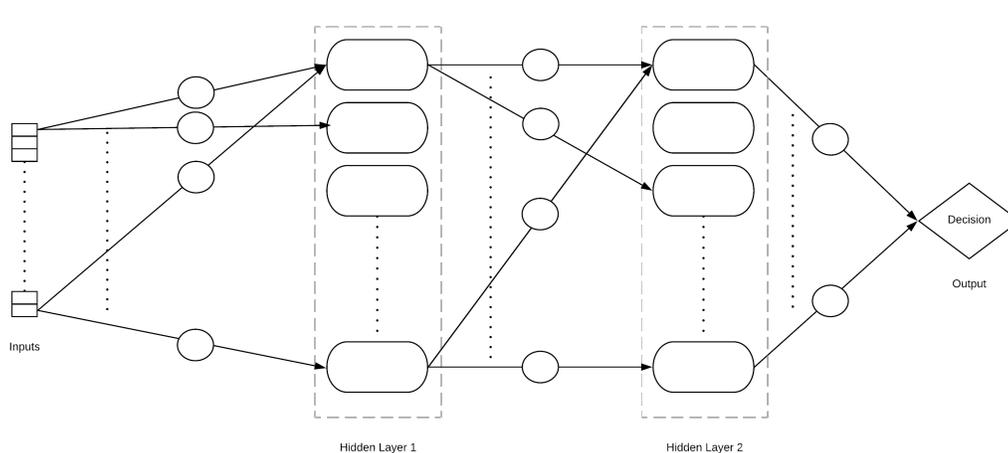


Figure 2.8: A visual representation of the feed forward neural networks described in this paper. The networks may have different numbers of input nodes and hidden layers, but this figure provides a generalized representation. The small boxes at the left of the image represent inputs, the rounded rectangles represent nodes in a hidden layer, the diamond represents the output, and the lines with circles represent weighted paths connecting the nodes between layers. In a full representation, each node in one layer has a weighted connection to each node in the next layer, but dotted lines are used here to represent the paths, weights, and nodes which are removed to reduce clutter. Not shown is the mapping which is applied to the input values before use in the network, as well as the remapping which is applied to the output. These are further described in section 2.3.1.

decision rules, more recognizable mathematical models, etc. Often these information extraction algorithms are specific to networks with certain architectures, training methods, or complexity levels. In [12], Hiroshi Tsukimoto develops an algorithm to extract decision rules from feed forward neural networks with any degree of complexity, any architecture, and any training method. The goal of the algorithm is to create a kind of “look-up table” or set of logic rules from which we could quickly determine the range of the output by simply looking at the inputs. Here I summarize the algorithm by first describing the underlying mathematical model of a neural network, then introducing the concepts of Tsukimoto’s algorithm in the discrete domain, and finally extending those concepts to the continuous domain. I also give two application examples to help illustrate the principles discussed.

## Underlying Mathematical Model

Here, I derive the equations to calculate the output of a neural network. The equations are specifically for a network with two hidden layers, but the ideas can easily expand to more or less layers.

Before any of the main network calculations, the input values are mapped from the range of each input to a normalized input range, which here is  $[-1, 1]$ . This is done as

$$\text{map}(n_k) = \frac{2(n_k - \min_k)}{\max_k - \min_k} - 1, \quad (2.13)$$

where  $n_k$  is the pre-mapping  $k^{\text{th}}$  input, and  $\min_k$  and  $\max_k$  define the approximate range of the pre-mapped values of the  $k^{\text{th}}$  input (defined during network training). Now, with the mapped inputs, we proceed into the hidden layers.

The output  $p_j^k$  of the  $j^{\text{th}}$  node in network layer  $k$  is the sum of the outputs of each node in the  $k - 1$  layer multiplied by the weight which connects it to the node, then passed through an activation function  $f(x)$ . This is shown as

$$p_j^k = f\left(b_j^k + \sum_{i=1}^N w_{ji}^k \cdot p_i^{k-1}\right), \quad (2.14)$$

with

$$f(x) = \tanh(x), \quad (2.15)$$

where  $w_{ji}^k$  is the weight which connects the  $i^{\text{th}}$  node of the  $k - 1$  layer to the  $j^{\text{th}}$  node of the  $k$  layer,  $b_j^k$  is a bias term for the current layer,  $p_i^{k-1}$  is the output of the  $i^{\text{th}}$  node in the  $k - 1$  layer, and  $N$  is the total number of nodes in the  $k - 1$  layer. Note the  $k$  and  $k - 1$  in the equation are superscripts, not exponents.

While Equation 2.14 shows the equation for just one node in a general hidden layer, the format of the equation simply repeats as we add layers and nodes. For example, the value of any node in the second hidden layer can be shown as

$$p_j^2 = f\left(b_j^2 + \sum_{i=1}^{N_2} [w_{ji}^2 \cdot f(b_i^1 + \sum_{h=1}^{N_1} w_{ih}^1 \cdot p_h^1)]\right), \quad (2.16)$$

where  $p_h^1$  are the mapped inputs to the system,  $N_1$  is the number of inputs to the system,  $N_2$  is the number of nodes in the first hidden layer, and the other terms are as previously defined.

After the network calculations have finished and there is a value at the output node, the wind speed or wind direction is found by mapping the output value from the output node range  $[-1,1]$  to the actual output range, which is defined during training ( $[0, 360]$  for wind direction, and approximately  $[0, 40]$  for wind speed). I write this step as  $\text{remap}(\cdot)$  as it, in essence, reverses the mapping performed in the first step. This calculation is done as

$$\text{remap}(n_o) = \frac{(\max_o - \min_o)(n_o + 1)}{2} + \min_o, \quad (2.17)$$

where  $n_o$  is the output node value, and  $\min_o$  and  $\max_o$  define the approximate range of the real output value (wind speed or wind direction).

This leads to our final equation for networks with two hidden layers, described by

$$\text{output} = \text{remap} \left( b_1^3 + \sum_{j=1}^{N_3} \left[ w_{1j}^3 \cdot \tanh \left( b_j^2 + \sum_{i=1}^{N_2} \left[ w_{ji}^2 \cdot \tanh \left( b_i^1 + \sum_{h=1}^{N_1} w_{ih}^1 \cdot p_h^1 \right) \right] \right) \right] \right), \quad (2.18)$$

where  $N_3$  is the number of nodes in the second hidden layer, and the rest of the variables are as previously defined. Again, note the superscripts are not exponents, but rather a layer indicator.

From here, the algorithm developed by Tsukimoto takes advantage of the nature of the activation functions used, which “squash” the input to a bounded output range [12]. This bounded output can be thought of like on/off logic rules. This idea allows us to extract logic rules which represent the decisions made in a neural network that lead to the activation of each node in the hidden layers, and in turn lead to different outputs. Although the individual weights may be different each time the network is trained, ideally the network always converges to a similar underlying form, given it has the same architecture and training data. It is this underlying form which I am trying to extract.

## Discrete Domain

The first idea to understand in Tsukimoto’s method is the approximation of the nodes of a network [12]. For this explanation we use a different activation function than mentioned previously,

the logistic sigmoid function, which is defined as

$$S(x) = \frac{1}{1 + e^{-x}}. \quad (2.19)$$

This function behaves similarly to the hyperbolic tangent visualized in Figure 3.1, but is bounded by 0 on the lower end rather than -1. Approximation simply forces the output of a node to the nearest bound, in this case resulting in either a 0 or a 1. With this idea, it can be seen that when the pre-activation value of a node (the input to the activation function) is less than 0, the node is approximated to 0. When this value is greater than or equal to 0, the node is approximated to 1. A node approximated to the upper bound is called active, while a node approximated to the lower bound is called inactive.

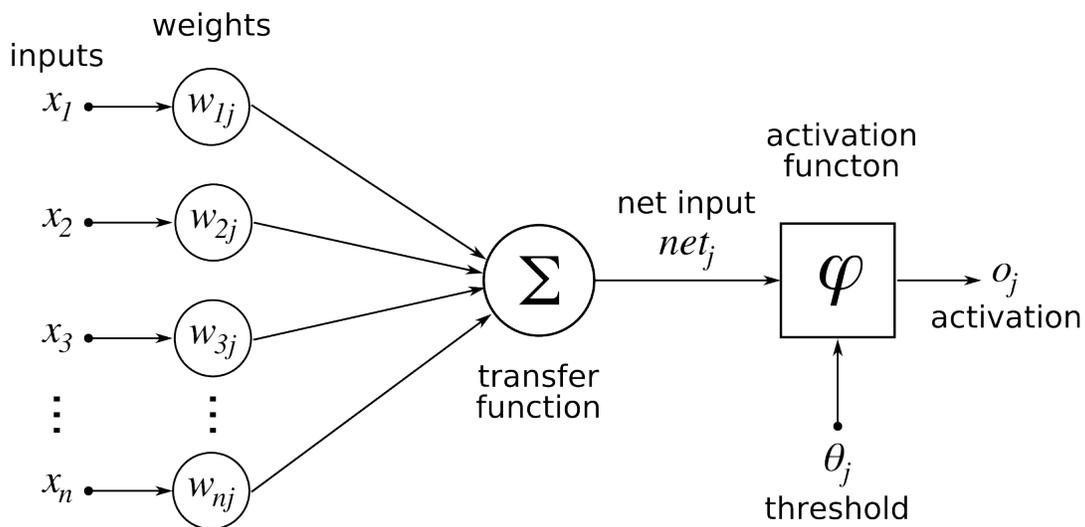


Figure 2.9: An expanded visualization of the calculations performed at one node in a hidden layer. The weighted inputs are summed, then a threshold/bias term is added to the sum, and an activation function is applied to that result.

The second idea to understand is the existence of a rule. Given inputs  $\{x_1, x_2, \dots, x_n\}$  to a node (each with domain  $\{0,1\}$ ), corresponding weights  $\{w_1, w_2, \dots, w_n\}$ , and a bias term  $\theta$  as in

Figure 2.9, we create the set of equations  $\{f_1, f_2, \dots, f_{2^n}\}$ :

$$\begin{aligned}
 f_1 &= S(w_1 + \dots + w_n + \theta) \\
 f_2 &= S(w_1 + \dots + w_{n-1} + \theta) \\
 &\dots \\
 f_{2^{n-1}} &= S(w_1 + \theta) \\
 f_{2^{n-1}+1} &= S(w_2 + \dots + w_n + \theta) \\
 &\dots \\
 f_{2^n-1} &= S(w_n + \theta) \\
 f_{2^n} &= S(\theta),
 \end{aligned} \tag{2.20}$$

where  $S(\cdot)$  is our activation function [12]. These equations represent the node's value for every possible combination of the weights which lead to that node, which follows from every combination of binary inputs. Now, the existence condition can be defined by saying a rule  $x_m$  exists if all  $f_i$  which involve the relevant  $w_m$  can be approximated to 1. For example, the rule  $x_1$  exists (meaning the overall output is active if input  $x_1$  is active) if all  $f_i$  involving  $w_1$  are approximated to 1, which might happen if  $w_1$  is much larger than the other weights. This example can also be expressed as

$$\min(f_i) \geq 0.5, (1 \leq i \leq 2^{n-1}). \tag{2.21}$$

For a higher order rule,  $x_2x_3x_4$  for example (meaning the output is 1 if  $x_2, x_3,$  and  $x_4$  are all 1), the rule exists if all  $f_i$  are approximated to 1 which involve  $w_2, w_3,$  and  $w_4$  (all three must be active).

Now, with the ideas of approximation and existence, the algorithm can be described simply: check the existence of each rule, starting with lower order terms and continuing up to a predetermined order limit (networks with fewer nodes may be able to consider all possible orders). The final decision rule representation of the neural network is the logical OR of all the existing terms found.

## Extension to Continuous Domain

In the continuous domain, Tsukimoto's algorithm is less developed. He describes a method to create simple function expressions for the neural network. However, for our application (further described later in Chapter 3), we are not necessarily in search of a direct function, but rather generalized logic rules which can help improve our scientific understanding. Thus I build upon the discrete principles developed by Tsukimoto and adjust the algorithm to fit this continuous application.

To find decision rules for the continuous domain, the input layer is momentarily ignored, and the first hidden layer nodes are treated as discrete inputs. From these, logic rules can then be developed according to Tsukimoto's discrete algorithm. Once there is an understanding of which nodes in the first hidden layer need to be active to lead to certain results, the actual inputs are analyzed to discover which circumstances lead to the activation of those first hidden layer nodes. Since the inputs are continuous, the logic rules are based on intervals which describe the input circumstances. For example, by observing the behavior of a node  $N$ , we may notice that the node is consistently active if one of its inputs  $A$  falls between 0 and 10. The logic rule can thus be defined as  $N = (0 < A < 10)$ , where  $N$  is 1 if the statement is true, and 0 if the statement is not true. Doing this, we develop a set of interval-based rules which, it is hoped, helps in understanding the behavior of the neural network.

## Examples

The first example comes from [12], where it is used to motivate the use of Tsukimoto's algorithm. The example describes a network trained to classify mushrooms as edible or poisonous based on 22 attributes, trained with 4062 samples. The neural network trained had three hidden nodes and an accuracy of 100% on the sample set. After running the algorithm, the following rules were deduced:

$$\begin{aligned} \textit{edible} &: (\textit{gill-size:broad})((\textit{odor:almond}) \vee (\textit{odor:anise}) \vee (\textit{odor:none})) \\ \textit{poisonous} &: (\textit{gill-size:narrow}) \vee (\neg(\textit{odor:almond}) \neg(\textit{odor:anise}) \neg(\textit{odor:none})). \end{aligned}$$

These rules were 97% accurate on the same sample set mentioned before. Although there is a slight drop in accuracy compared to the full network, the rules provide us not only with a classification, but with a reason behind that classification. Seeing this reasoning helps us improve our scientific understanding of mushrooms. The goal of applying this algorithm to wind retrieval is to extract similar logic rules which tell us the underlying decisions of a wind model, and how the ambiguity is resolved.

The second example is meant to demonstrate the effectiveness of this algorithm on a simple known relationship. I built a data set which expressed the XOR relationship for 4 inputs (output is true when an odd number of inputs are true), then trained a neural network with a single hidden layer containing 5 nodes to identify this relationship. The goal of this example was to see if the rules extracted from the trained neural network using the algorithm match the known relationship in the training data. Essentially this proves that the approximations done are insignificant to the logic rules. The rules extracted, where  $I$  represents an input node,  $X$  represents a hidden layer node, and  $Y$  is the output, are as follows:

$$X_1 = I_1\bar{I}_2\bar{I}_3 \vee I_1\bar{I}_2I_4 \vee I_1\bar{I}_3I_4 \vee \bar{I}_2\bar{I}_3I_4$$

$$X_2 = I_1I_2\bar{I}_3 \vee I_1I_2\bar{I}_4 \vee I_1\bar{I}_3\bar{I}_4 \vee I_2\bar{I}_3\bar{I}_4$$

$$X_3 = I_1\bar{I}_3$$

$$X_4 = I_1I_2 \vee I_1\bar{I}_3 \vee I_1\bar{I}_4 \vee I_2\bar{I}_3 \vee I_2\bar{I}_4 \vee \bar{I}_3\bar{I}_4$$

$$X_5 = \bar{I}_1I_2 \vee \bar{I}_1\bar{I}_4 \vee I_2\bar{I}_3 \vee I_2\bar{I}_4 \vee \bar{I}_3\bar{I}_4$$

$$Y = X_1X_2\bar{X}_3 \vee X_1X_2\bar{X}_4 \vee X_1X_2X_5 \vee X_1\bar{X}_3\bar{X}_4 \vee X_1\bar{X}_3X_5 \vee X_1\bar{X}_4X_5$$

$$\vee X_2\bar{X}_3\bar{X}_4 \vee X_2\bar{X}_3X_5 \vee X_2\bar{X}_4X_5 \vee \bar{X}_3\bar{X}_4X_5$$

Ideally, from the algorithm described earlier in this section, certain terms dominate the expressions, as is with  $X_3$ . This allows us to reduce these expressions to a simpler form with fewer terms. However, these rules were difficult to simplify, most likely because the network weights were all similar, which meant there were not many dominating terms. To avoid complex logistic simplification of these extracted rules, I reproduced the logic rules in code and compared

the output to the output of a direct XOR of the inputs. The result agreed with the direct XOR with 100% accuracy, meaning the simplification of these logic equations is equivalent to an XOR. This shows the effectiveness of neural networks in learning relationships and patterns, and the success of the algorithm in extracting the learned relationship from the trained neural network, at least for a discrete case.

## CHAPTER 3. NEURAL NETWORKS APPLIED TO ASCAT UHR WIND RETRIEVAL

This chapter discusses the application of neural networks to ASCAT UHR wind retrieval. I start by introducing the methods used to create and test the neural networks. I then show results of these tests, and attempt to extract a logic-rule representation of selected networks.

### 3.1 Network Selection and Validation

#### 3.1.1 Theory

While there is ambiguity in current scatterometer wind retrieval models, neural networks may present the possibility to produce a single unambiguous solution given the same inputs as those to a GMF. Here I reinforce this idea from a theoretical standpoint.

Previous research has established the fact that, given a sufficient number of hidden nodes, a feed-forward neural network with a single hidden layer can approximate any Borel measurable function to any desired degree of accuracy, which is proven theoretically and discussed in depth in [35]. This fact has helped drive the increased use of neural networks and machine learning in all kinds of calculation and classification applications. Adding to the theoretical proof cited, I present here a mathematical and visual explanation of the idea. To start, we look at one of the key features of neural networks, the activation function.

The activation function is applied to the value of a neural network at each hidden layer node throughout the calculations. One common function is the hyperbolic tangent, which is visualized in Figure 3.1. The figure shows the general shape of the hyperbolic tangent, as well as the effect of weighting its input. It can be seen that, as the weight increases, the hyperbolic tangent approaches a step-function shape.

Now, if a second heavily-weighted hyperbolic tangent with a slight offset in the input is subtracted from the first, it creates a boxcar-like function which is “on” for a desired input region,

and “off” everywhere else, as in Figure 3.2. This boxcar function can easily be manipulated by modifying the offset, and by adding a weight to the output of the hyperbolic tangents in addition to the input. This idea of combining step functions to create a boxcar function can also be readily extended to more dimensions.

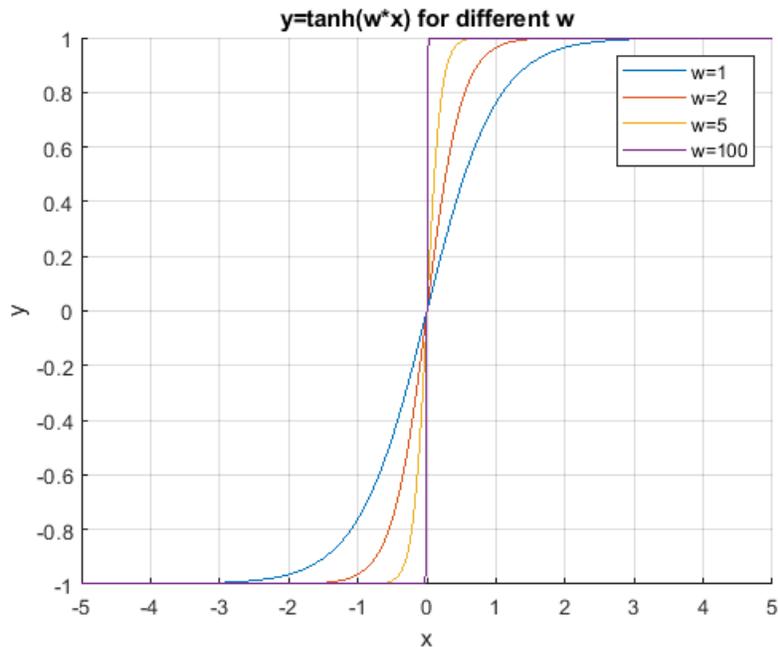


Figure 3.1: A plot of a common activation function, which introduce non-linearity to neural networks. The function shown is the hyperbolic tangent function, and is used to “squash” the input to a -1 to 1 output range.

Next, with the idea of a modifiable boxcar function, imagine breaking any function up into several disjoint sections and fitting a separate box to each section. As the number of sections increases, the network’s fit to the function is improved and the overall error decreases. This idea is visualized in Figure 3.3. In the network, the number of sections is increased by simply increasing the number of nodes used.

Thus in summary, in any dimension, several nodes can be used to create a boxcar-like shape which can then be fit to an individual segment of a target function, which can be any calculable relationship. Then, by increasing the number of boxes and decreasing the size of the segments which each box represents, the error can be reduced to any desired degree of accuracy. Interestingly, this

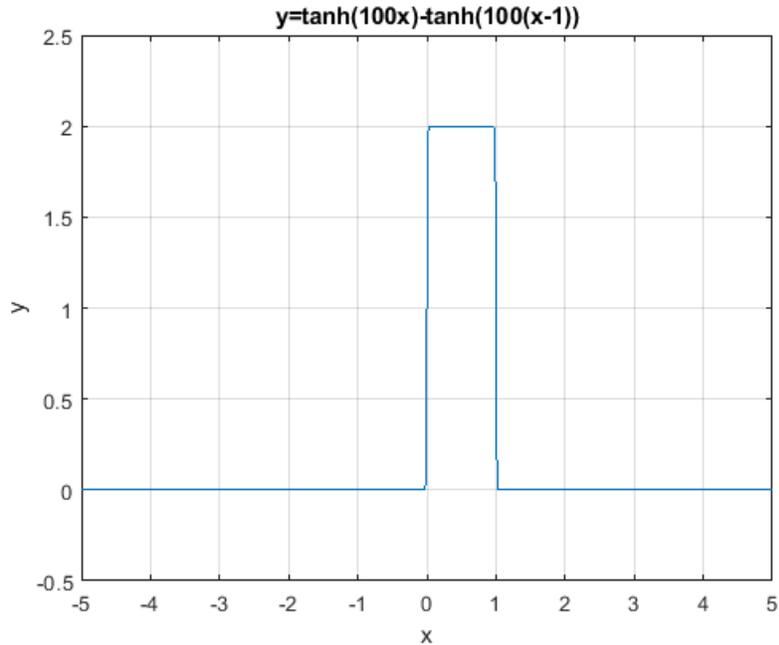


Figure 3.2: A boxcar function created by subtracting two equally-scaled hyperbolic tangent functions, with an offset of one in between.

concept shows that even the input-output behavior of a more complex non-feed forward network model can be represented in feed forward form.

In real applications, the linear portion of the activation function as it transitions between the two bounds, as well as alternative activation functions are also utilized to trim the number of nodes necessary to reach a desired degree of accuracy. However, this simplified visual proof, along with the theoretical approach in [35], illustrates the potential which neural networks provide to the modelling of any unknown relationship. These ideas suggest that, given a proper network architecture, a neural networks has the ability to model patterns and relationships which could help resolve the wind ambiguities that are left from current models.

For the network model to begin with a generic set of weights and offsets, and find the ideal model, the network is trained. In training, a set of inputs and corresponding outputs are presented to the network, and the weights and offsets are adjusted in an iterative process to best “learn” the desired calculations or relationship. If the proper architecture has been selected to begin with, this learning should iteratively approach the ideal model which can achieve the desired degree of accuracy. The training data and process used for the networks in this chapter is discussed next.

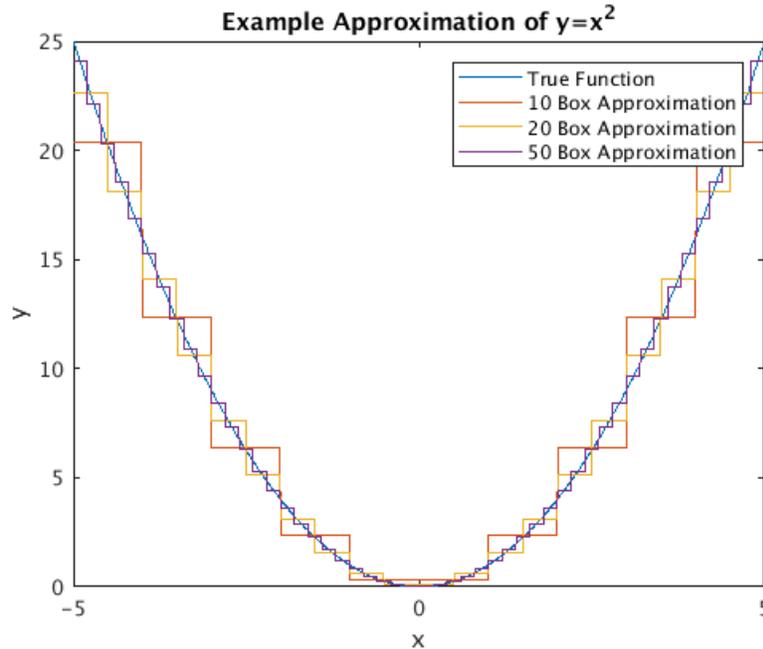


Figure 3.3: An example of approximating the function  $y = x^2$  using several box functions as described in Figure 3.2. It can be seen that, as the number of boxes used to approximate the function increases, the accuracy improves.

### 3.1.2 Training

Selecting an accurate, diverse training set of data with the proper inputs and outputs is essential to an accurate resultant network model. While neural networks are often able to quickly and accurately reproduce the desired output behavior of the training data, they may do so by learning features and relationships different than the intended model. For example, a neural network could use the latitude and longitude location of a pixel to improve its wind retrieval accuracy, while the actual backscatter-to-wind model should be independent of location. Or, if a training data set over-represents lower wind speeds, the neural network may train specifically to features of low wind speeds in order to minimize training error, making it more unpredictable when applied to higher wind speeds.

Examples of wind fields from the selected training set for this application can be seen in Figure 3.4. Each region image is  $14^\circ$  latitude by  $14^\circ$  longitude, and was selected in order to represent different wind speeds, directions, and/or patterns. The goal is to develop a diverse training set which produces a network that is accurate across all wind-vector possibilities, not just

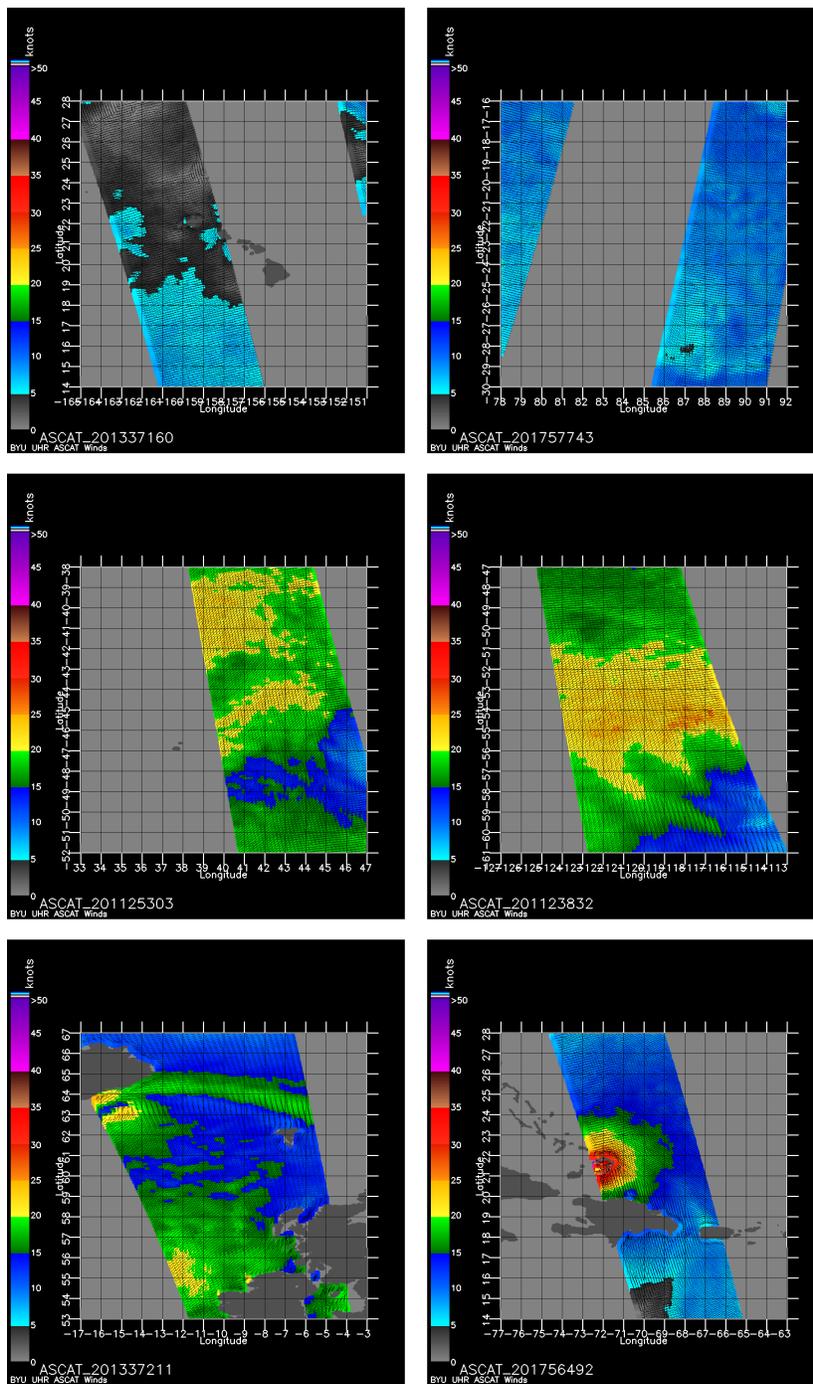


Figure 3.4: Example images of wind fields used to train the neural networks. The top two images represent low wind speeds, the middle two represent medium-to-high wind speeds, and the bottom two represent different interesting directional patterns. Together these images help to train the network with a wide variety of wind patterns so that it can provide accurate results for a wide range of situations after training.

specific to a subset of wind vector possibilities. As is the case with the CMOD5 model, the only information provided to the network comes from backscatter measurements, and the corresponding azimuth and incidence angles. The data used as the desired output in the training process is the European Centre for Medium-Range Weather Forecast (ECMWF) Numerical Weather Prediction (NWP) wind data from those regions [36], but interpolated to the higher UHR resolution.

During the training process, all network weights are initialized according to the Nguyen-Widrow method [37], then are updated after each iteration through the training set according to the Levenberg-Marquardt method [38]. The Nguyen-Widrow initialization provides some randomization of initial weights, helping to eliminate any hard-coded bias, but ideally the network converges to a similar solution each time it is recreated, independent of initial weights.

### 3.1.3 Network Testing

As previously mentioned, the framework for all neural networks used in this thesis is a feed forward, fully connected network, with wind retrieval done on a per pixel basis. Within the feed forward framework, the number of nodes and layers is altered to find an architecture that is well-tuned for the ASCAT sensor and its UHR winds. In general, increasing the number of inputs, nodes, and/or layers provides potential for increased accuracy, but may also result in over-training or the network learning undesirable features specific to the training set, thus worsening performance outside of training. Reductions in number of inputs, nodes, and/or layers often reduce training time and simplify the extraction of information from the network, but give less potential to learn the details of the desired relationship.

To determine which alterations result in significant improvements and are worth incorporating, I observe wind vector accuracy by calculating RMSE for the speed and direction components separately. The accuracy results are based on the use of ECMWF NWP interpolated models as the standard. One of the difficulties with scatterometer-based ocean wind retrieval has always been the lack of reliable true wind measurements with sufficient spatial and temporal resolution. The ECMWF NWP models are generally accepted as a reference data set, though at a coarse resolution, and can be read about more in [36]. Data used in testing here comes from two different days in each of three  $14^\circ$  latitude by  $14^\circ$  longitude locations: one centered on the Hawaiian islands, one near the southeast coast of Greenland, and the third in the middle of the Indian ocean. The

comparison results include only pixels that contain valid data in all three data sets involved (neural network, current GMF, and ECMWF NWP interpolated), with the assumption that all have been mapped to the same grid. These tests are certainly not exhaustive in testing architectures, nor in testing all wind patterns, but illustrate the ideas discussed in this thesis.

Along with changes in accuracy as we adjust network parameters, there is also a change in training time and time taken to perform the calculations when used post-processing. It was found that alterations to the architecture made the largest difference in training time for the networks, with little effect on calculation time in post-training usage. Since this post-training calculation is where the network is used most of the time, I am not concerned with the changes to training time and thus it is not discussed further.

I first search for a network tuned to only the wind speed retrieval. Wind speed presents the most simple relationship of the two components of wind vector retrieval and illustrates the architecture selection process well. After wind speed, I move to the more complex wind direction retrieval problem. For both wind speed and direction, I test networks with two different input styles. One input style only pays attention to relevant information for the target pixel, mirroring the CMOD5 model without ambiguity selection. The second input style takes into account information from the eight neighboring pixels to the target pixel, as well as the information from the target pixel itself. This is similar to the idea of ambiguity selection in the current model process, where the pixels neighboring the pixel of interest for wind retrieval are taken into account in order to smooth out obvious inconsistencies.

## **Wind Speed Networks**

In this thesis I first consider point-wise retrieval, meaning only measurements in one wind vector cell or pixel are used to retrieve wind for that pixel, independent of nearby cells. Later, a field-wise approach is considered. This point-wise approach for wind speed leads to neural networks with nine inputs: the fore, mid, and aft  $\sigma_0$  values; the three corresponding incidence angles; and the three corresponding azimuth angles. The results for several different architectures with these inputs are seen in Table 3.1. The simplest wind speed network which gives sufficient accuracy is selected for further exploration.

My first observation is that adding a second hidden layer is not worth the additional complexity. Although there are occasional accuracy improvements with adding a second hidden layer, the improvements are not significant or consistent enough to justify the complexity added. Next, among the single layer networks, it is seen that the network with six nodes is consistently among the most accurate performers, and it is thus chosen for further examination. It is not only fairly consistent across the six test regions here, but provides similar results each time the network is recreated. This makes it a good subject for the information extraction algorithm. Figures 3.5 and 3.6 show some comparison maps between the ECMWF model winds, CMOD5 with ambiguity selection, and the six node single layer network output. It can be seen that both the CMOD5 UHR winds and the neural network reveal some similar fine scale wind speed features, but may miss some of the larger scale features from the ECMWF winds. Overall, the neural network performance appears to match current models well.

Table 3.1: RMS error for several network architectures with nine inputs in each of the six test regions. The CMOD5 error is also included for comparison. The RMSE is reported in m/s. The name of each row describes the architecture, where AL\_B(\_C) describe first the number of layers (A), then the number of nodes in the first hidden layer (B), and finally the number of nodes in the second hidden layer (C), if it exists. Note that, due to some randomization in the initialization of these networks, these results may vary if the networks are recreated, trained, and tested.

WIND SPEED RMSE (m/s)	Region					
	Hawaii 1	Hawaii 2	Indian 1	Indian 2	Greenland 1	Greenland 2
<b>CMOD5</b>	0.859	0.797	1.00	1.42	2.14	1.11
<b>1L_5</b>	0.681	1.05	0.778	1.99	1.73	2.85
<b>2L_5_5</b>	0.848	1.99	0.650	2.38	1.36	2.17
<b>1L_6</b>	0.559	1.68	0.755	2.12	1.67	2.10
<b>2L_6_6</b>	0.587	1.31	0.694	1.46	1.34	2.11
<b>1L_9</b>	0.604	3.01	0.733	1.17	1.35	3.19
<b>2L_9_9</b>	0.500	2.92	0.652	2.30	1.19	2.84
<b>1L_12</b>	0.532	3.88	0.670	1.35	1.27	2.94
<b>2L_12_12</b>	0.471	6.79	0.587	2.17	1.08	2.57
<b>2L_5_6</b>	0.618	1.61	0.698	1.89	1.33	1.69
<b>2L_6_5</b>	0.580	2.10	0.663	2.60	1.33	3.89
<b>2L_7_6</b>	0.521	1.55	0.704	2.12	1.22	2.36
<b>2L_6_7</b>	0.540	2.49	0.665	2.13	1.25	8.29
<b>2L_7_7</b>	0.472	2.96	0.626	1.93	1.19	2.65

For the wind speed networks with more inputs, I started with the same nine inputs as the simple networks, then included the three backscatter measurements from each of the eight neighboring pixels (see Figure 3.7), resulting in 33 total inputs. Results from several of these networks are found in Table 3.2. While all results were generally within a few meters per second of the target wind speeds, the performance is consistently significantly worse than the networks in Table 3.1, which use fewer inputs. This suggests that for wind speed the increased number of inputs leads to over-training, meaning the networks learned features specific to the training data set, which worsened their performance on the different cases in validation. When more limited information is provided, the networks are forced to generalize their models in training, which transfers into better performance with the validation data set.

Table 3.2: RMS error for several network architectures with 33 inputs in each of the six test regions. The CMOD5 error is also included for comparison. The RMSE is reported in m/s. The name of each row describes the architecture, where AL\_B(\_C) describe first the number of layers (A), then the number of nodes in the first hidden layer (B), and finally the number of nodes in the second hidden layer (C), if it exists.

<b>WIND SPEED</b> <b>RMSE</b> (m/s)	<b>Region</b>					
	<b>Hawaii</b> <b>1</b>	<b>Hawaii</b> <b>2</b>	<b>Indian</b> <b>1</b>	<b>Indian</b> <b>2</b>	<b>Greenland</b> <b>1</b>	<b>Greenland</b> <b>2</b>
<b>CMOD5</b>	0.859	0.797	1.00	1.42	2.14	1.11
<b>1L_10</b>	3.71	3.44	2.86	3.69	8.14	6.07
<b>1L_15</b>	3.71	6.23	3.12	3.16	7.70	6.79
<b>2L_15_15</b>	3.743	6.27	2.75	4.17	8.77	6.86
<b>1L_20</b>	3.84	6.39	3.06	3.71	9.32	10.69
<b>2L_20_20</b>	3.86	5.46	2.83	3.86	9.20	31.78
<b>1L_25</b>	4.14	4.90	3.02	3.73	9.23	7.87
<b>2L_25_25</b>	3.76	6.60	2.49	3.38	8.11	10.19
<b>1L_30</b>	3.91	6.67	2.81	3.38	8.00	6.02
<b>2L_30_30</b>	3.65	14.8	2.59	4.34	12.73	11.85

### Wind Direction Network

Similar to the wind speed networks, I first tried to develop a more simple, nine input network to model the wind direction. This led to the results for several tested architectures in Table 3.3. My first observation from these results is that certain regions resulted in high error for most,

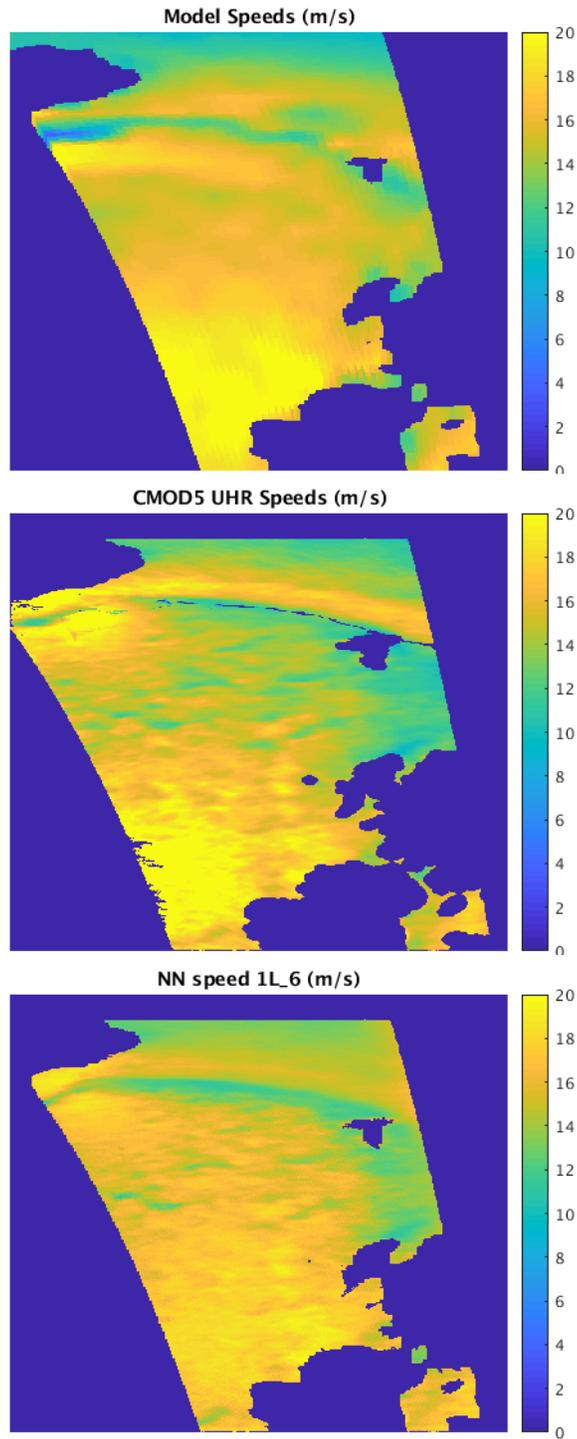


Figure 3.5: Maps of wind speed from a region near the south east coast of Greenland. The top image is the ECMWF NWP interpolated winds used as truth data, the middle comes from the CMOD5 UHR wind retrieval process, and the bottom is from the six node neural network.

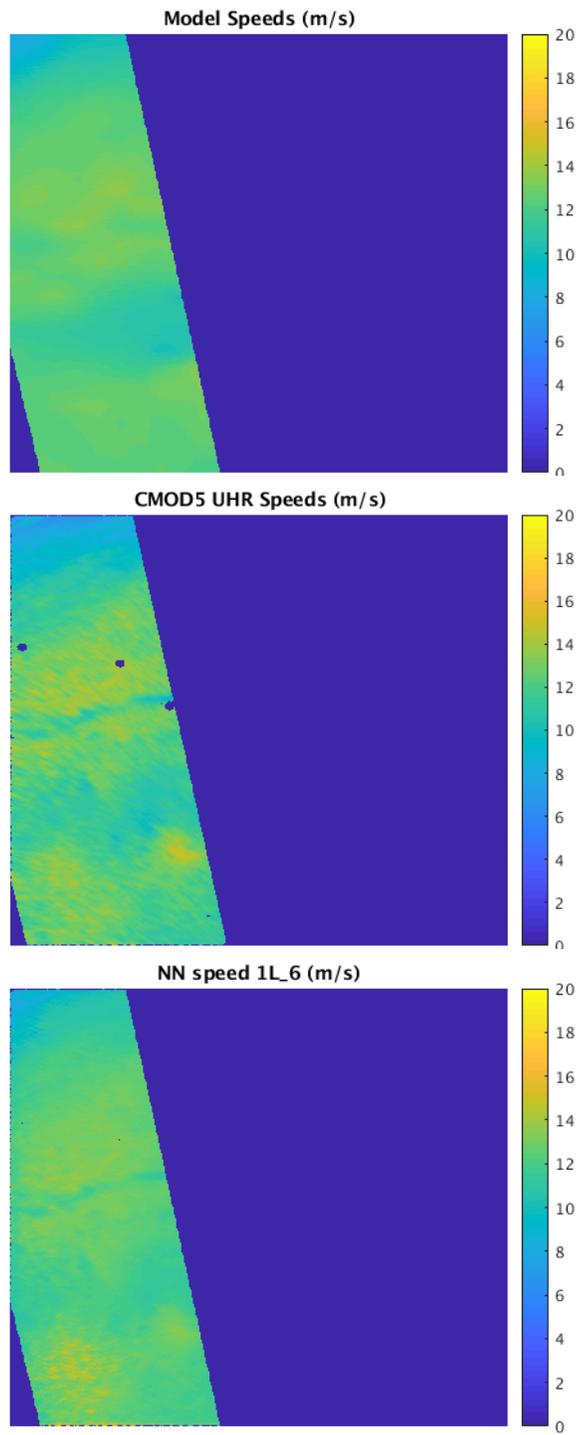


Figure 3.6: Maps of wind speed from a region near Hawaii. The top image is the ECMWF NWP interpolated winds used as truth data, the middle comes from the CMOD5 UHR wind retrieval process, and the bottom is from the six node neural network.

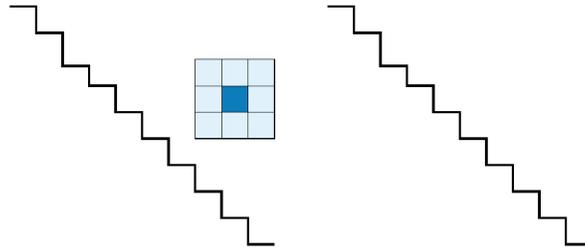


Figure 3.7: A visualization of the pixels taken into account for the 33 and 34 input neural networks. The darker blue is the target pixel, and the light blue are considered the neighboring pixels.

if not all of the networks. This is particularly apparent in the second Indian Ocean region, and the second Greenland region. This suggests that neural networks may have trouble resolving the same ambiguity that exists in the CMOD5 process. It also might suggest that there were patterns in those regions in the validation data set which may not have been well-represented in the training data. Overall, however, the network with a single hidden layer and 20 nodes in the layer had the best performance, and had similar accuracy to the CMOD5 process (including ambiguity selection) in four out of the six regions.

The performance of several increased input wind direction networks can be seen in Table 3.4. Along with the nine target pixel input values and the 24 neighboring pixel values as with wind speed networks (see Figure 3.7), I also included the wind speed value for the target pixel, resulting in 34 total inputs. The overall most consistent and accurate wind speed network found was the two layer network with 20 nodes in each layer. The RMSE values for this network are slightly better than the simple networks selected from Table 3.3, suggesting that the extra input information is useful in extracting an accurate model and resolving directional ambiguity. However, even with the increase to 34 inputs, the neural network model performance is significantly worse than the currently used CMOD5 process in the same trouble regions mentioned earlier. Figures 3.8 and 3.9 show some comparison maps between the ECMWF model winds, CMOD5 with ambiguity selection, and the neural network results (for both the 9 input and 34 input cases). Figure 3.8 shows an example where the neural networks performed well, and Figure 3.9 shows a poor performance example. It can be seen that the CMOD5 UHR winds and the neural network are similar in general,

Table 3.3: RMS error for several network architectures with the 9 inputs defined in the architecture section. The CMOD5 error is also included for comparison. The RMSE is reported in degrees.

The name of each row describes the architecture, where AL\_B(\_C) describe first the number of layers (A), then the number of nodes in the first hidden layer (B), and finally the number of nodes in the second hidden layer (C), if it exists.

WIND DIRECTION RMSE (degrees)	Region					
	Hawaii 1	Hawaii 2	Indian 1	Indian 2	Greenland 1	Greenland 2
<b>CMOD5</b>	5.46	14.69	9.37	15.75	21.20	20.11
<b>1L_3</b>	16.67	14.41	17.38	68.61	28.84	56.70
<b>2L_3_3</b>	25.90	26.27	20.15	75.81	27.90	57.71
<b>1L_6</b>	8.96	22.49	6.68	67.04	21.19	48.26
<b>2L_6_6</b>	8.26	43.99	6.28	76.88	16.94	67.63
<b>1L_20</b>	4.14	34.86	5.73	66.15	11.51	47.61
<b>2L_20_20</b>	3.35	65.24	5.24	62.26	7.08	57.35
<b>3L_20_20_20</b>	6.23	17.74	5.56	67.2	12.71	72.57
<b>1L_30</b>	3.64	38.92	5.68	72.22	8.63	76.88
<b>2L_30_30</b>	3.00	63.07	5.09	75.98	6.32	66.73
<b>1L_40</b>	4.18	114.3	5.33	74.73	10.07	96.67
<b>2L_40_40</b>	2.914	63.09	4.96	71.67	6.19	79.85

but the neural networks have several areas where the direction error is around  $180^\circ$ . This supports the idea that the networks struggle with the same ambiguity addressed in the CMOD5 process.

### 3.1.4 Analysis

The networks tested here demonstrate the potential to produce similarly accurate results to those of the CMOD5 process, without the need for an explicit ambiguity selection step. However, none of them do so consistently across all test data sets. Occasional high error is observed especially in the wind direction neural networks when they are presented with certain new wind direction patterns, excessive noise, etc., giving results significantly worse than the CMOD5 process. A larger training set of data and different neural network frameworks offer the potential to improve this performance, but the results here suggest that feed forward neural networks struggle with radar measurement noise and wind retrieval ambiguity. For a neural network, this noise and ambiguity in training can lead to undesired model features which result in the unpredictable behavior that we see in several of the test regions in Tables 3.1, 3.2, 3.3, and 3.4. This does not mean

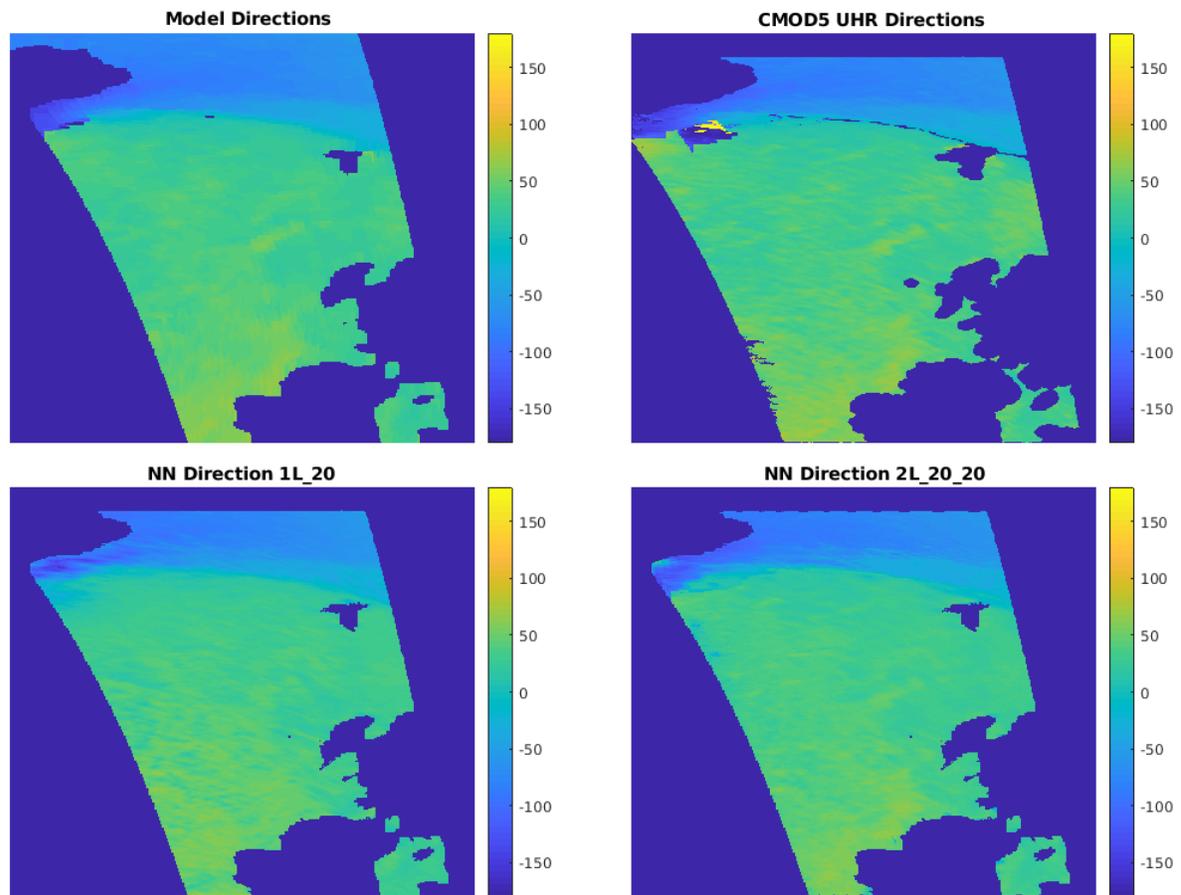


Figure 3.8: A comparison of results from the ECMWF NWP winds (top left), the CMOD5 process (top right), the most accurate 9 input network (bottom left), and the most accurate 34 input network (bottom right). The RMSE values for these networks are found in the Greenland 1 column of Table 3.3 for the 9 input 1L\_20 network, and Table 3.4 for the 34 input 2L\_20\_20 network. All values are in degrees from north. It can be seen that both networks perform very well in this test example.

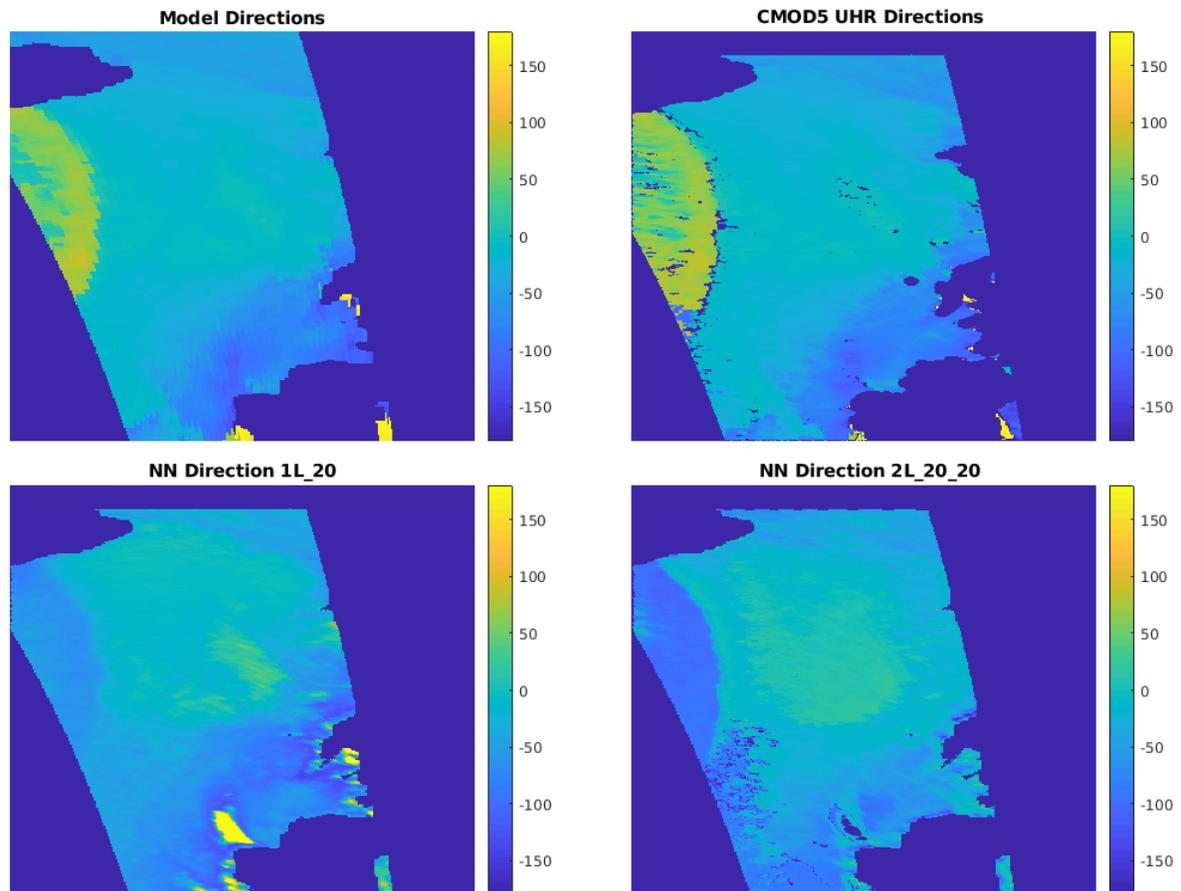


Figure 3.9: A comparison of results from the ECMWF NWP winds (top left), the CMOD5 process (top right), the most accurate 9 input network (bottom left), and the most accurate 34 input network (bottom right). The RMSE values for these networks are found in the Greenland 2 column of Table 3.3 for the 9 input 1L\_20 network, and Table 3.4 for the 34 input 2L\_20\_20 network. All values are in degrees from north. It can be seen that both networks have areas where their wind direction retrieval performed poorly.

Table 3.4: RMS error for several network architectures with the 34 inputs defined in the architecture section. The CMOD5 error is also included for comparison. The RMSE is reported in degrees. The name of each row describes the architecture, where AL\_B(\_C) describe first the number of layers (A), then the number of nodes in the first hidden layer (B), and finally the number of nodes in the second hidden layer (C), if it exists.

WIND DIRECTION RMSE (degrees)	Region					
	Hawaii 1	Hawaii 2	Indian 1	Indian 2	Greenland 1	Greenland 2
<b>CMOD5</b>	5.46	14.69	9.37	15.75	21.20	20.11
<b>1L_10</b>	6.03	33.99	6.65	68.59	16.35	74.05
<b>1L_15</b>	4.88	20.11	5.55	70.58	12.91	61.69
<b>2L_15_15</b>	3.16	31.86	5.07	61.79	7.40	61.17
<b>1L_20</b>	4.27	20.05	5.50	75.06	11.85	59.05
<b>2L_20_20</b>	3.89	16.32	5.37	65.04	10.48	58.15
<b>1L_25</b>	4.22	40.92	5.37	68.93	11.31	61.95
<b>2L_25_25</b>	2.85	33.26	4.75	70.77	6.21	64.20
<b>1L_30</b>	4.54	51.25	5.21	60.37	10.34	64.16
<b>2L_30_30</b>	2.69	34.39	4.56	70.66	4.99	62.14

neural networks cannot handle this noise or ambiguity, but may need more input information, more training examples, and/or different architectures than the feed forward networks used here in order to find an overall accurate model.

### 3.2 Extraction of the Neural Network Transfer Function

While it is difficult to find feed forward neural networks which had consistently accurate performance across all test regions, they still show potential to resolve some wind retrieval ambiguity without an explicit ambiguity selection process. In this section, I apply the information extraction algorithm described in Chapter 2 in an attempt to discover what decisions and calculations the neural networks used to resolve these ambiguities.

I first attempted information extraction from the nine input, six hidden node wind speed neural network, attempting to represent each node logically, then combine the rules for the final logic representation. The interval-based logic rules used in the final approximation of the neural

network are as follows, where  $X_i$  is the  $i^{th}$  node of the hidden layer:

$$X_1 = (f\_azi > 0)(f\_inc > 46) \left[ (m\_sig < 1.125 f\_sig + 8.875)(m\_sig < -17) \right. \\ \left. \vee (a\_sig > 0.9091 m\_sig - 5.6364)(m\_sig < -18.5) \right. \\ \left. \vee (f\_sig < -11)(f\_sig > 4.9 m\_sig - 3.1 a\_sig - 19.4) \right. \\ \left. (a\_sig > -16)(m\_sig < 0.7143 f\_sig + 1.8571) \right] \quad (3.1)$$

$$X_2 = (m\_sig < 2.2 f\_sig + 16) \quad (3.2)$$

$$X_3 = (f\_azi > 0) \left[ (f\_sig < -5.5 m\_sig - 2 a\_sig - 170) \right. \\ \left. \vee (a\_sig > -16)(a\_sig > 1.6667 f\_sig + 1) \right. \\ \left. \vee (a\_sig < -0.3333 f\_sig - 30) \vee (m\_sig < -19) \right. \\ \left. \vee (m\_sig < 1.25 f\_sig + 7.5)(a\_sig < f\_sig - 5) \right] \quad (3.3)$$

$$X_4 = (a\_azi > 32) \quad (3.4)$$

$$X_5 = (f\_azi > 0) \left[ (a\_sig < -f\_sig - 45) \right. \\ \left. \vee (a\_sig < 3 m\_sig - 2 f\_sig - 16) \vee (a\_sig < -18.5) \right. \\ \left. \vee (f\_sig > -8 m\_sig + 7 a\_sig + 2) \vee (m\_sig > 0.6923 f\_sig + 1.5385) \vee (m\_sig < -19) \right] \quad (3.5)$$

$$X_6 = (a\_azi > 32). \quad (3.6)$$

When the logic rule evaluates to true, the node value equals one, and negative one when false. Each of these rules individually was about 90% accurate to its respective node's activation in the actual neural network. This means that if we force the node output values to the -1 or 1 bound, the logic rules estimate the correct bound in 90% of the test cases. The actual output values of one of the hidden nodes and the logic rule based output values of the node are seen in Figures 3.10 and 3.11 respectively, plotted against different input values for several test cases.

The pre-mapping output of the network is then calculated as

$$Y = 0.0289X_1 + 0.1384X_2 + 0.1118X_3 + 1.7093X_4 - 0.0894X_5 - 1.6746X_6 - 0.454, \quad (3.7)$$

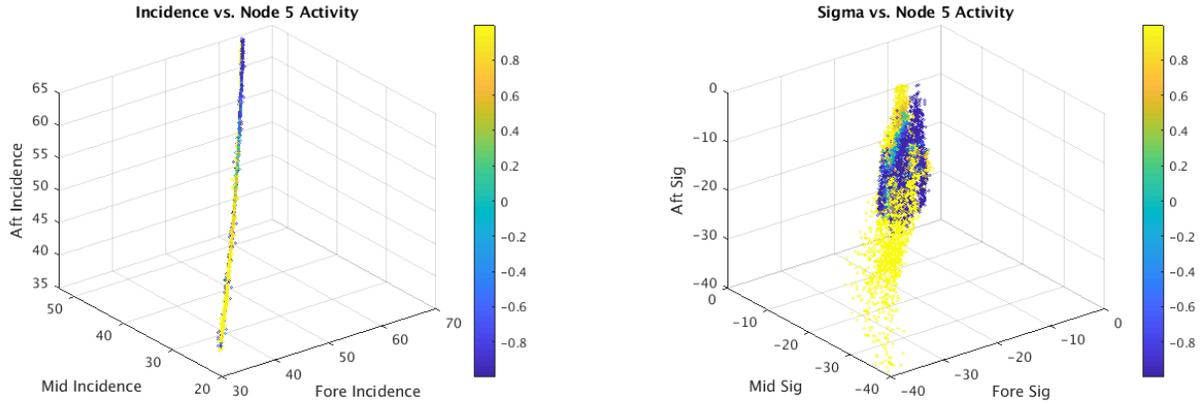


Figure 3.10: An example of the activation (displayed as the color of each point) of node 5 in the hidden layer based on different inputs. The left figure shows the activation versus incidence angle values, and the right shows activation versus backscatter values. For the approximation step, all positive values are replaced with 1 (active), and all negative are replaced with -1 (inactive).

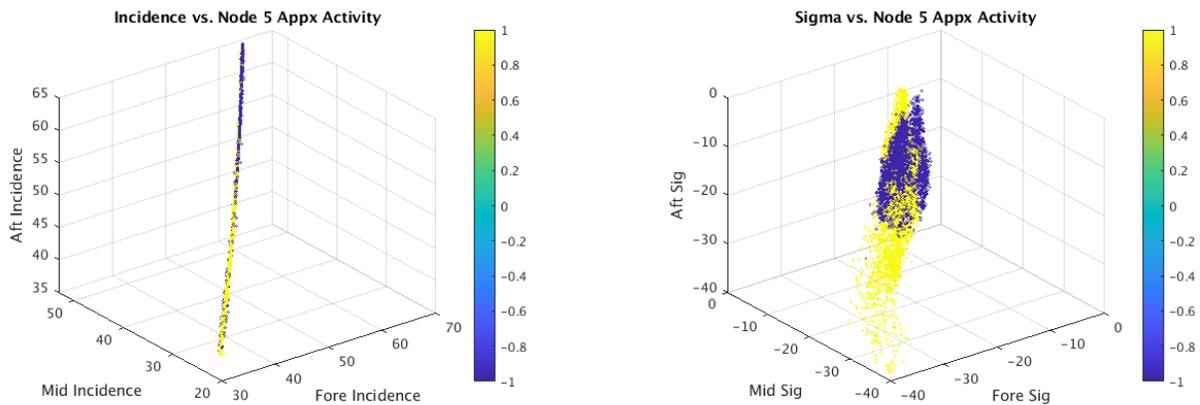


Figure 3.11: An example of the approximate activation (displayed as the color of each point) of node 5, which is the logic rule based version of Figure 3.10. Comparing this figure with Figure 3.10 we see that the regions where the node is activated by the inputs are very similar overall.

after which the output  $Y$  is mapped to the wind speed range. The mapping is calculated as

$$\text{OutSpeed} = \frac{(40.0197 - 2.0898)(Y + 1)}{2} + 2.0898. \quad (3.8)$$

Unfortunately, though logic rules are found to be able to represent an individual hidden node, the accuracy drops significantly when trying to combine these logic rules to describe the overall performance of the system. This is because the small inaccuracies of the logic rules for

individual nodes compound in the combined expression to result in larger overall inaccuracies in predicting the network output. Further, relatively few simplifications can be made to the combined rules for the network, and thus the resultant expression is very complex and doesn't add much improved transparency to the neural network representation.

I followed a similar process for the wind direction network, but the direction problem proved even more complex than the speed calculation and I quickly came to the same conclusion as with speed: for a neural network with enough complexity to accurately model the wind-retrieval problem, the extracted logic rules become too convoluted to provide much insight. A different approach is needed to extract the information in a more apparent form than the logic rule representation for the wind retrieval problem.

## **CHAPTER 4. NEURAL NETWORKS IN SMAP SOIL MOISTURE RETRIEVAL**

This chapter discusses resolution enhancement using rSIR of SMAP brightness temperature data, and the extracted soil moisture data. It also explores the application of neural networks as an alternate method to perform the soil moisture calculations. I first examine the effects of rSIR resolution enhancement on the soil moisture accuracy using the theory-based soil moisture extraction method. I then discuss the application of feed forward neural networks to the extraction, evaluate their performance, and compare them with the theory-based model at high and low resolutions.

### **4.1 rSIR Enhanced Resolution Soil Moisture**

As discussed in Chapter 2, the rSIR algorithm provides opportunity to enhance the resolution of brightness temperature images from SMAP data, and thus the resolution of the derived soil moisture product. Example images from the results of this rSIR resolution enhanced soil moisture can be seen in Figures 4.1 and 4.2, which show the 36 km, 9km and 3 km soil moisture for parts of eastern Africa and central South America, respectively. A visual inspection of these images shows that the resolution enhancement can reveal fine scale features and details that are over-smoothed or hidden in lower resolution images. Although the higher resolution features of the enhanced resolution soil moisture images are visibly apparent, the accuracy requires comparison with a validated data set, which is the goal of the rest of this section.

#### **4.1.1 Validation Method**

For validation of the rSIR-enhanced soil moisture data, I compare rSIR enhanced results with previously validated data products. Although there is currently no collection of “true” soil moisture data with significant coverage at all of the resolutions tested here, NSIDC produces different, accurate soil moisture products at each of the resolutions which are used for validation.

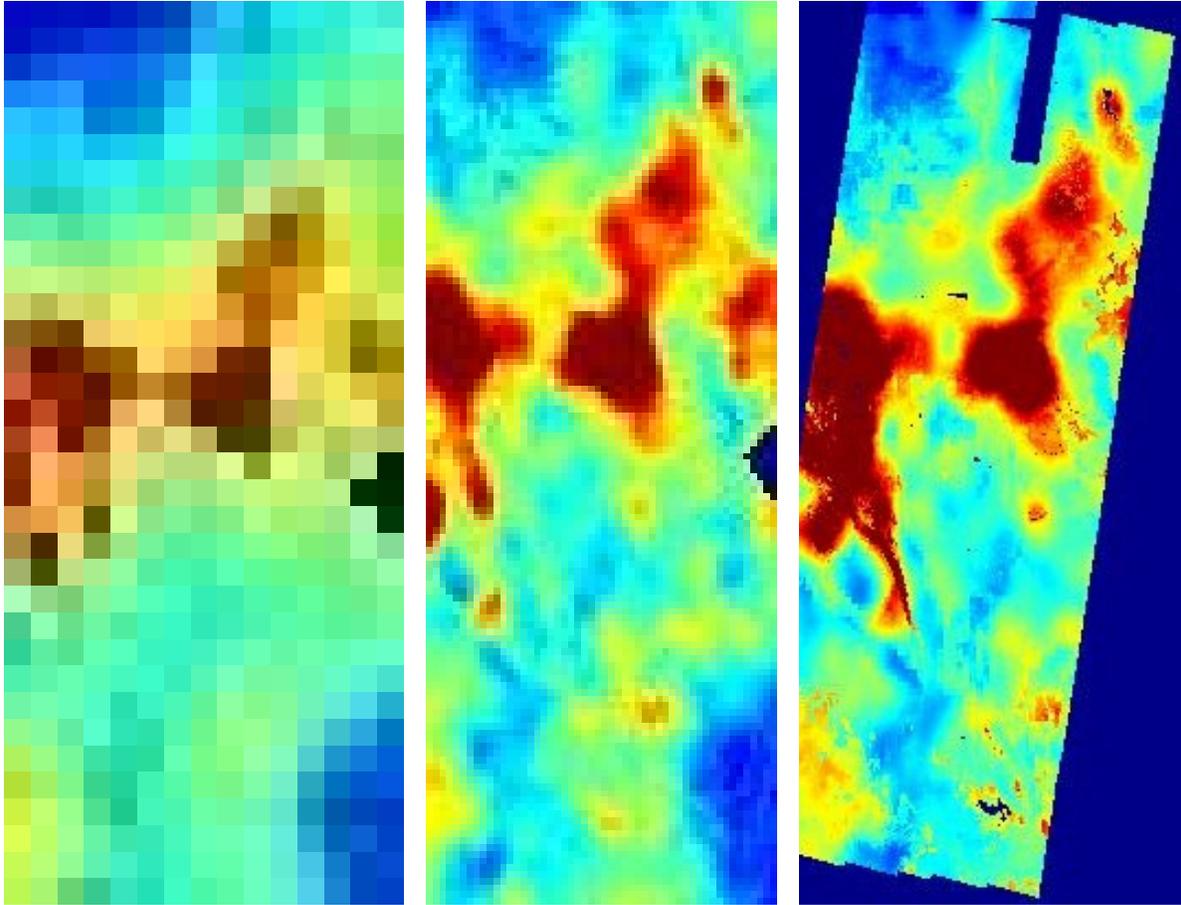


Figure 4.1: Soil moisture images from eastern Africa at 36 km resolution (left), 9 km resolution (center), and 3 km resolution (right), from day 300 in 2016. This soil moisture is extracted from rSIR enhanced brightness temperature images. The color scale for 36 km and 9 km can be seen in Figure 4.5, and Figure 4.6 for 3 km.

For the 36 km and 9 km resolution, I use SMAP radiometer-only soil moisture products. These products have both been validated to within  $0.04 \text{ cm}^3 \text{ cm}^{-3}$  of in situ measurements [28]. For the 3 km resolution, I use a recently released product which combines the SMAP radiometer and Sentinel radar data [39], [40]. This data has been validated to within  $0.05 \text{ cm}^3 \text{ cm}^{-3}$  of in situ measurements [41].

To produce the Sentinel-SMAP combined product, the Sentinel radar data is used as a direct replacement for the SMAP radar in the active-passive merged soil moisture algorithm [30]. Sentinel is a C-band radar and has several differences from the originally used SMAP radar, but one key advantage is that Sentinel provides backscatter data at a higher resolution than the SMAP radar. This allows for the production of a merged soil moisture product at 3 km, as well as the

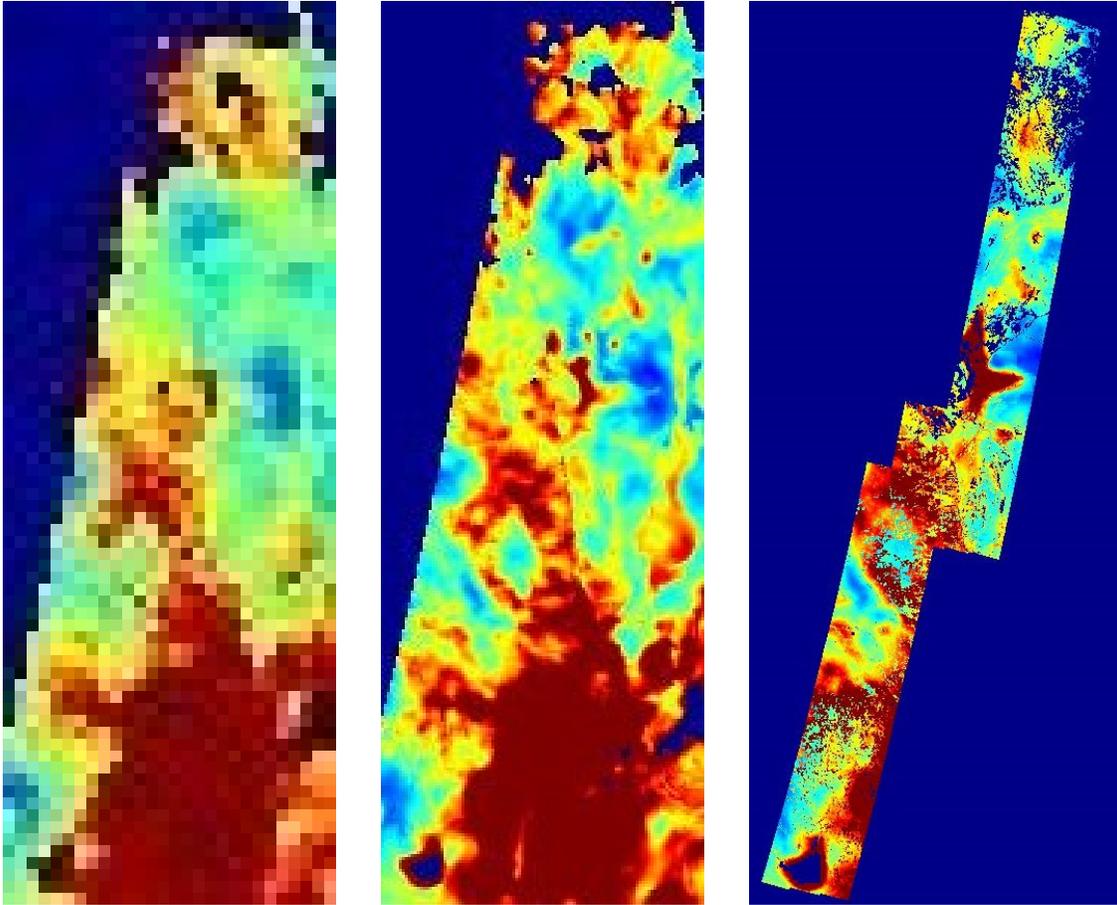


Figure 4.2: Soil moisture images from central South America at 36 km resolution (left), 9 km resolution (center), and 3 km resolution (right), from day 301 in 2016. This soil moisture is extracted from rSIR enhanced brightness temperature images. The color scale for 36 km and 9 km can be seen in Figure 4.5, and Figure 4.6 for 3 km.

originally-intended 9 km. However, due to the narrow swath width of the Sentinel system, there are relatively few SMAP and Sentinel measurements which correspond in time and location closely enough to give accurate results after soil moisture extraction. Thus the SMAP/Sentinel combined soil moisture product comes in small 30-second scenes (as in the 3 km images in Figures 4.1 and 4.2). I use four months worth of data, spread throughout the year, in order to include seasonal variability in comparisons to validate the enhanced resolution soil moisture product.

### 4.1.2 Validation Results and Analysis

A summary of the results from validation testing of the rSIR-enhanced soil moisture product (computed using the theoretical model presented in Chapter 2) is presented in Table 4.1 and Table 4.2, which contain the total RMS error and mean error for four separate months of soil moisture data when compared to the NSIDC data product. The base level of error introduced simply by inaccuracy in my model can be seen in the 36 km resolution column of the tables. This error is mainly due to corrections applied in NSIDC brightness temperature and soil moisture data, which are not reproduced in my version of the soil moisture extraction algorithm due to lack of information available in documentation. These corrections are discussed more in Chapter 2, and in [28].

Table 4.1: RMS error for soil moisture produced at several resolutions when compared to validated NSIDC products. The 36 km column shows the performance of my algorithm implementation at the default resolution, where the error is mainly due to algorithm differences. The table shows that, as we increase resolution, the RMSE increases as well. This shows the effect of the higher noise levels in the rSIR enhanced brightness temperature as it translates to the extracted soil moisture product.

<b>Soil Moisture RMSE</b>	<b>36 km Resolution</b>	<b>9 km Resolution</b>	<b>3 km Resolution</b>
<b>January</b>	0.0130	0.0208	0.0436
<b>April</b>	0.0123	0.0200	0.0452
<b>July</b>	0.0123	0.0202	0.0450
<b>October</b>	0.0120	0.0195	0.0415

As mentioned previously, the 36 km RMSE values in Table 4.1 and Table 4.2 can be used as a reference as the resolution is enhanced to 9 km and 3 km using the rSIR algorithm. First, observing the mean error table shows that the tendency of my version of the soil moisture extraction algorithm is to overestimate soil moisture compared to the NSIDC product. While the overestimate is small on average at all resolution levels, there is an increased overestimation at higher resolutions than lower.

Next, observing the RMSE table, it is seen that the RMSE level increases as the resolution is enhanced. There is also an increased level of uncertainty in the validation of the 3 km product which is introduced simply because of the use of the SMAP-Sentinel data as truth data. As

Table 4.2: Mean error for soil moisture produced at several resolutions when compared to validated NSIDC products. This number varies from the RMSE in Table 4.1 in that it includes negative errors. Thus a positive mean error suggests that my algorithm implementation tends to overestimate, and a negative mean error suggests underestimation. The table shows that my algorithm implementation has a slight tendency to overestimate on average, but has decent performance at all resolutions.

<b>Soil Moisture Mean Error</b>	<b>36 km Resolution</b>	<b>9 km Resolution</b>	<b>3 km Resolution</b>
<b>January</b>	0.0010	0.0023	0.0074
<b>April</b>	0.0014	0.0028	0.0083
<b>July</b>	0.0019	0.0034	0.0082
<b>October</b>	0.0012	0.0025	0.0064

mentioned in Section 4.1.1, the SMAP passive product has been verified to within  $0.04 \text{ cm}^3 \text{ cm}^{-3}$ , whereas the SMAP-Sentinel combined product has only been verified to within  $0.05 \text{ cm}^3 \text{ cm}^{-3}$ . The 3 km products also have a larger output range of 0.02 to 0.6, whereas the 9 km and 36 km work from 0.02 to 0.5. However, despite the differences mentioned which contribute to the RMSE, the overall performance at all three resolution levels is still relatively accurate. While there is certainly an increase in overall noise and uncertainty in the soil moisture product as we increase resolution, the RMSE performance is good enough to suggest that features and patterns revealed as we improve resolution are accurate.

## 4.2 Network Selection and Validation

As seen in the previous section, the noise introduced in brightness temperature resolution enhancement using rSIR results in a small but noticeable drop in accuracy of the derived soil moisture. This section looks at the effect of this increased noise on a trained neural network. I use the 36 km data to develop a neural network model which estimates soil moisture, then apply that model to the enhanced resolution 3 km data. This section describes the network training and selection process, then tests several networks at the 36 km and 3 km resolutions.

### 4.2.1 Training

For the neural networks tested here, the training set of data consists of pixels from brightness temperature and ancillary data maps as input, and soil moisture maps as output. Soil moisture maps for a few of the training days can be seen in Figures 4.3 and 4.4, which also show how data from the ascending and descending passes of the satellite are separated for this training process. The specific input maps needed are the same as those necessary for the theory-based soil moisture extraction algorithm (see Chapter 2): uncorrected brightness temperature (Kelvin), surface temperature (Kelvin), vegetation opacity, incidence angle (degrees), single scattering albedo, surface roughness, clay fraction, and water body fraction. Each map has near-global coverage at the default 36 km resolution, and represents a single day in 2016. The days chosen are spread throughout the year in order to represent wet and dry seasons of regions throughout the globe. All maps are drawn from the data sets on the National Snow and Ice Data Center (NSIDC) SMAP website.

We also take into account other parameters such as measurement quality, freeze-thaw fraction, and land cover type, then eliminate pixels with poor quality or undesirable characteristics from the training set. The target outputs are the 36 km volumetric soil moisture values ( $\text{cm}^3 \text{cm}^{-3}$ ) corresponding to the days involved. All data is passed in on a per pixel basis.

As is the case with the wind retrieval networks in Chapter 3, all network weights are initialized according to the Nguyen-Widrow method [37], then are updated after each iteration through the training set according to the Levenberg-Marquardt method [38].

### 4.2.2 Network Testing

I tested several networks with variations in number of nodes and layers in comparison with the theory-based soil moisture model at 36 km resolution. Again, as is with the wind retrieval networks, all networks are fully connected feed forward networks. There are eight inputs, which are the same eight inputs listed in the training section, and one output, the volumetric soil moisture measurement.

In Table 4.3, I show RMSE values for several of the different network variations that I tested. The values represent the overall root mean squared error over the entire globe for multiple

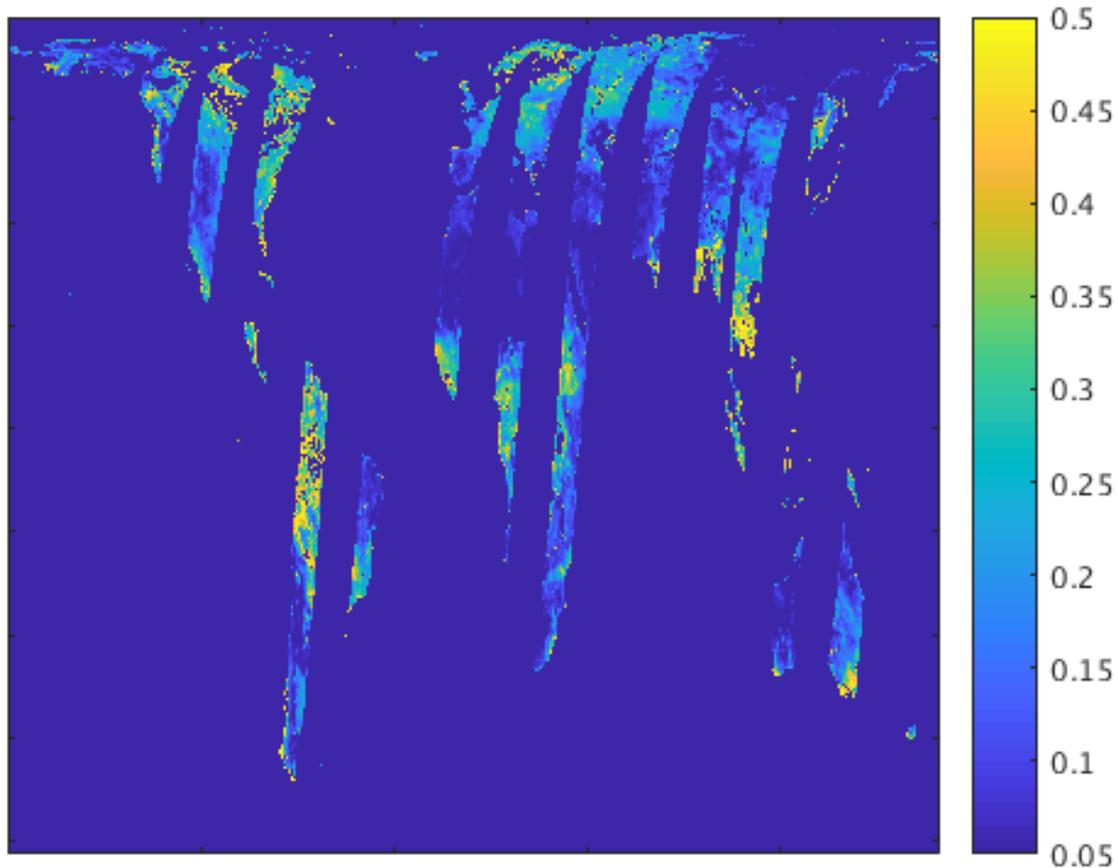


Figure 4.3: Example soil moisture data used as the target outputs in the neural network training process. This map shows ascending passes from a day in January 2016, downloaded from the SMAP NSIDC website.

days in the months listed in each column. All network output values are forced to the 0.02 to 0.5 bounds.

Although the RMSE values for the 36 km data show very small improvements as I increase the number of nodes in the network, this improvement stopped at an RMSE value of about 0.0116, which was achieved with a network that had 40 total hidden layer nodes (the 2L\_20\_20 network). This performance was only 0.0004 better than the smallest network shown though, which had only 5 hidden nodes. This suggests that it requires relatively few hidden nodes to accurately extract the underlying soil moisture model, and that the remaining error is due largely to the uncertainty introduced by noise. Also, the RMSE values in Table 4.3 are approximately equal to, and even slightly better than the values from the theory-based model in Table 4.1. The slight improvement

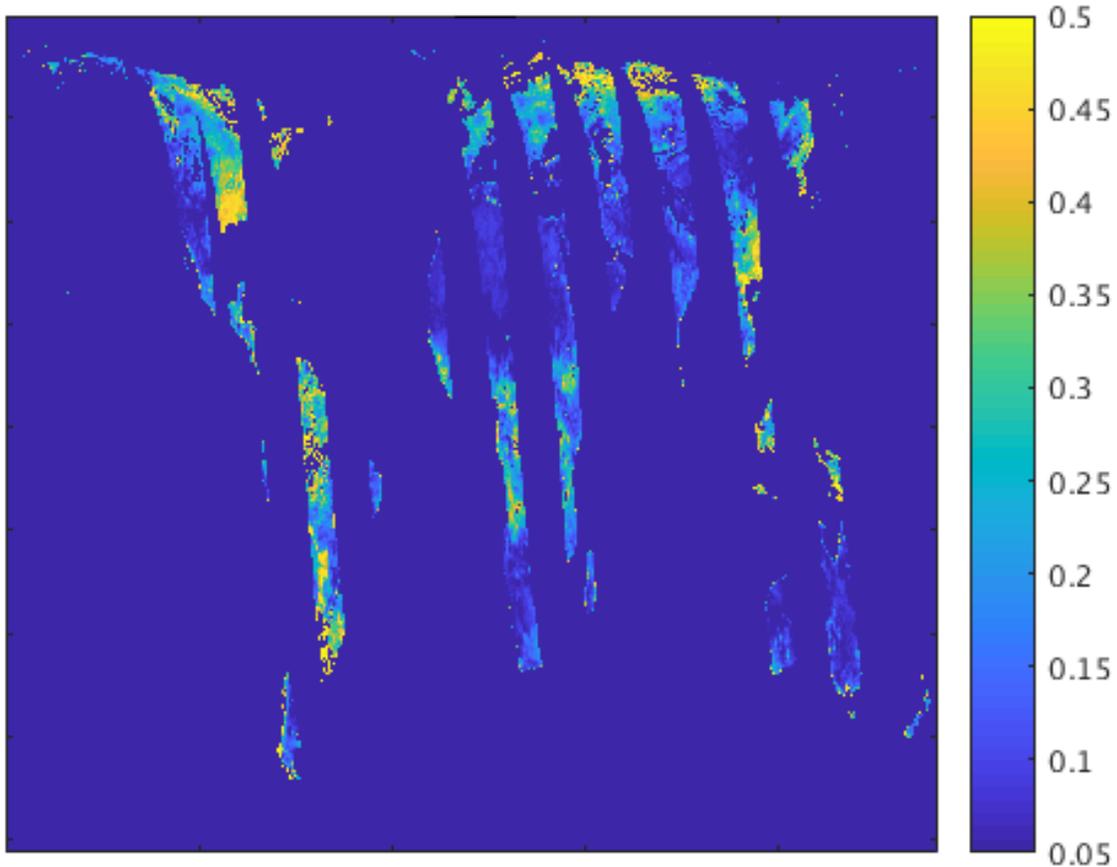


Figure 4.4: Example soil moisture data used as the target outputs in the neural network training process. This map shows descending passes from a day in April 2016, downloaded from the SMAP NSIDC website.

may be due to the neural networks learning a method to better account for the corrections that need to be made to the brightness temperature values. Figure 4.5 shows an example comparing soil moisture extracted by both methods at 36 km resolution.

Next, I tested the same networks at the enhanced 3 km resolution, with results listed in Table 4.4. The values again represent the overall RMSE for the entire globe for multiple days in the months listed in each column. It is seen that the results reflect the increased noise level and resultant lower overall accuracy of the higher resolution product. None of the networks tested were able to perform better than 0.0403 overall RMSE at 3 km resolution, as opposed to the 0.0116 level at 36 km. Also, while there is more variation in error between months than in previous tables, the error for each month in Table 4.4 is very similar between network architectures, which was

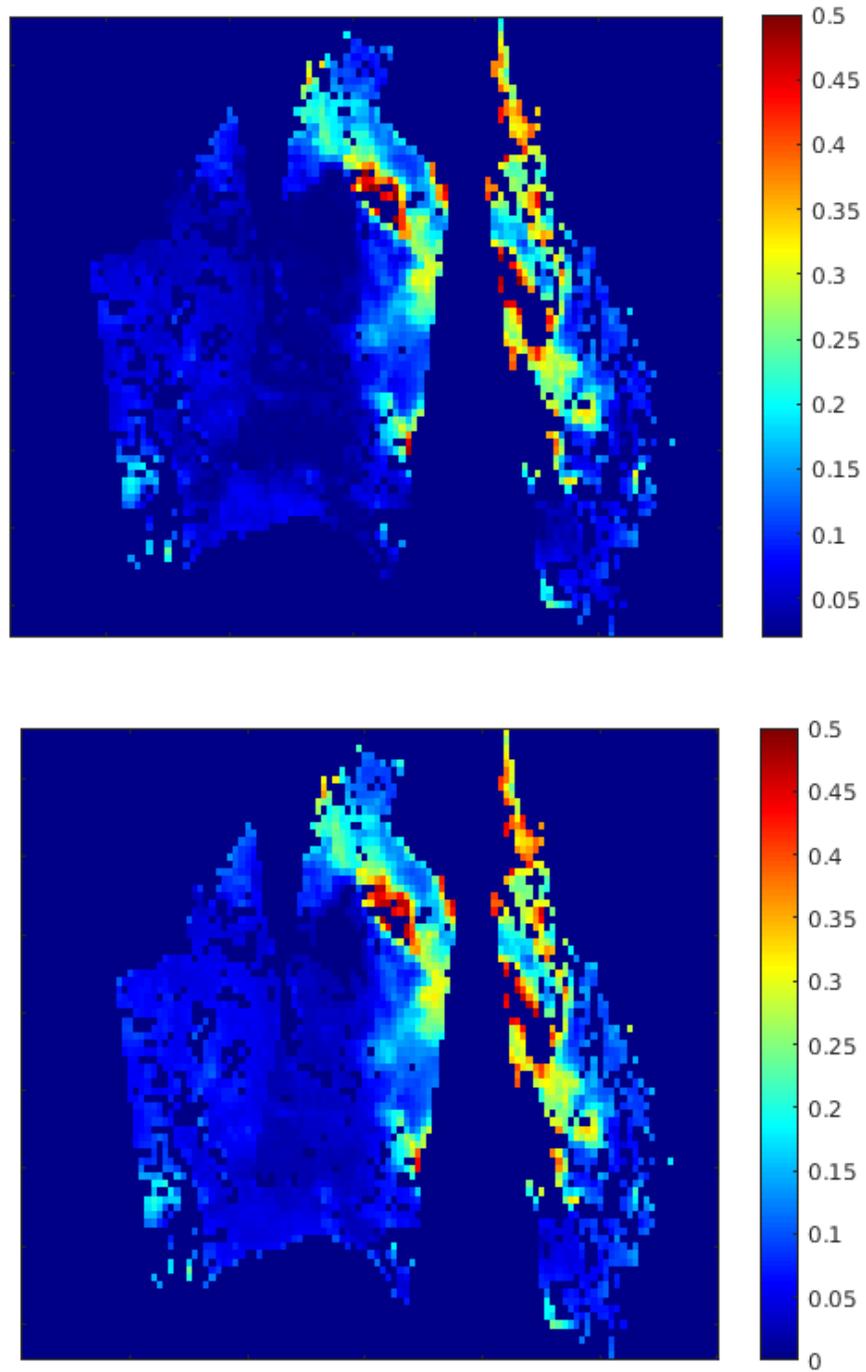


Figure 4.5: Example 36 km resolution soil moisture maps from day 2 of 2016. The top map is from the reference NSIDC SMAP passive soil moisture data set, and the bottom is calculated using the 1L\_15 neural network.

Table 4.3: RMS error for neural network soil moisture retrieval on 36 km brightness temperature data at several resolutions when compared to validated NSIDC products. The columns representing months take into account multiple days from the specified month. The Overall column represents all tested days.

<b>Soil Moisture 36 km RMSE</b>	<b>January</b>	<b>April</b>	<b>July</b>	<b>October</b>	<b>Overall</b>
<b>1L_5</b>	0.0123	0.0121	0.0124	0.0113	0.0120
<b>2L_5_5</b>	0.0119	0.0119	0.0121	0.0112	0.0118
<b>1L_10</b>	0.0118	0.0119	0.0121	0.0111	0.0117
<b>2L_10_10</b>	0.0117	0.0118	0.0121	0.0111	0.0117
<b>1L_15</b>	0.0118	0.0118	0.0117	0.0112	0.0118
<b>2L_15_15</b>	0.0116	0.0117	0.0121	0.0111	0.0117
<b>3L_15_15_15</b>	0.0117	0.0118	0.0120	0.0110	0.0116
<b>1L_20</b>	0.0118	0.0118	0.0121	0.0111	0.0117
<b>2L_20_20</b>	0.0117	0.0117	0.0120	0.0110	0.0116
<b>3L_20_20_20</b>	0.0117	0.0118	0.0121	0.0111	0.0117
<b>1L_25</b>	0.0117	0.0118	0.0121	0.0111	0.0117
<b>2L_25_25</b>	0.0117	0.0117	0.0120	0.0110	0.0116

also seen at 36 km. This suggests again that few nodes are required to model the soil moisture extraction, and that the training set of data is diverse enough to create well-generalized networks that transfer well to 3 km data.

When comparing the values in Table 4.4 with those in Table 4.1 we see that, similar to the comparison at 36 km resolution, the neural networks have slightly better overall accuracy than the theoretical approach. Figure 4.6 shows an example comparing soil moisture extracted by both methods at 3 km resolution. However, there is more variation in the neural network error than the theoretical approach. This highlights the unpredictability of neural networks when presented with noisy measurements, though in this case the unpredictability due to noise had much less effect on average than in the wind retrieval case presented in Chapter 3.

### 4.2.3 Analysis

The results in this chapter reveal the potential that neural networks have to accurately extract soil moisture at several resolutions and improve soil moisture data products. Neural networks used in conjunction with the rSIR algorithm provide a possible means to enhance the resolution of SMAP radiometer brightness temperature measurements, and the derived soil moisture product.

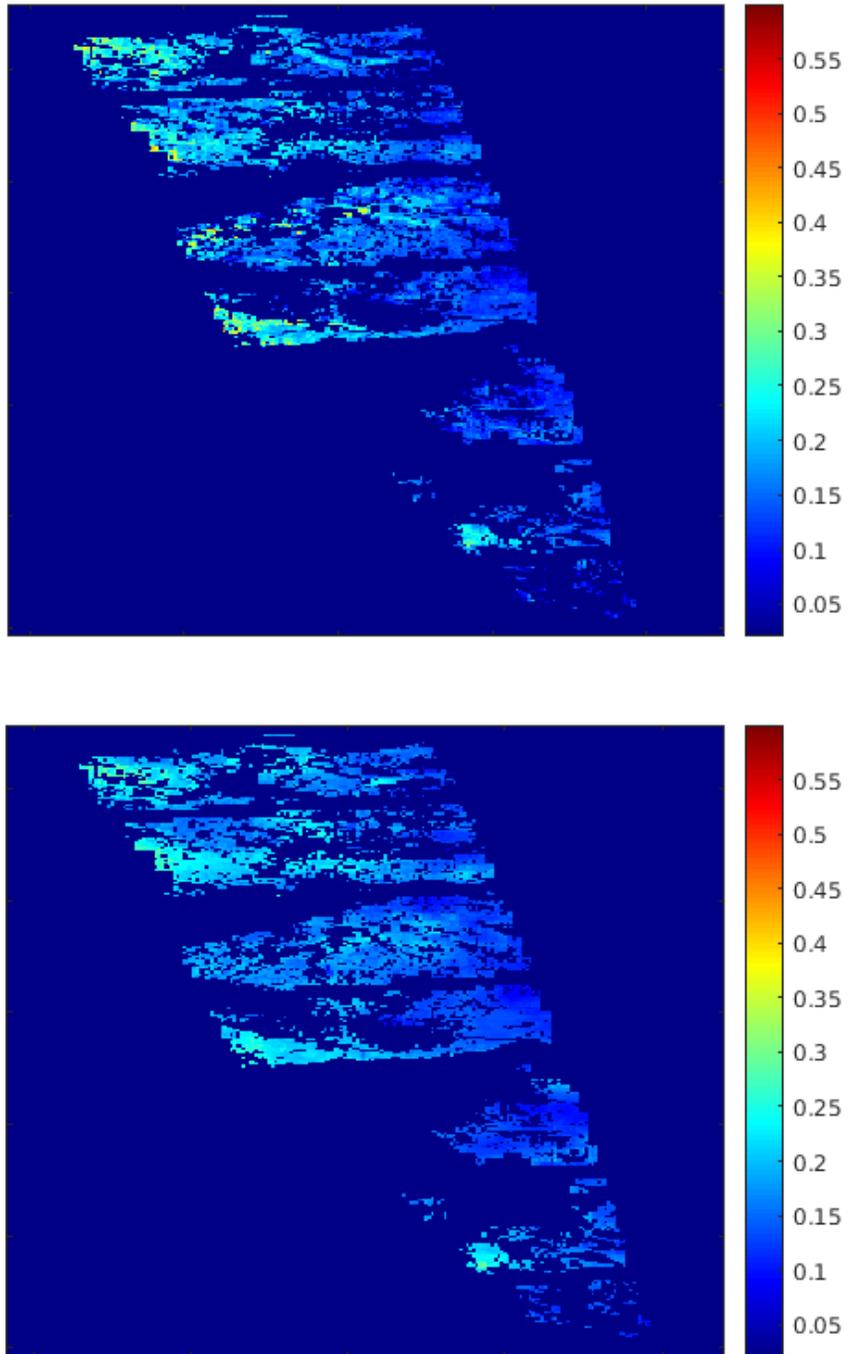


Figure 4.6: Example 3 km resolution soil moisture maps from day 2 of 2016. The top map is from the reference NSIDC Sentinel-SMAP soil moisture data set, and the bottom is calculated using the 1L\_15 neural network.

Table 4.4: RMS error for neural network soil moisture retrieval on 3 km brightness temperature data at several resolutions when compared to validated NSIDC products. The columns representing months take into account multiple days from the specified month. The Overall column represents all tested days.

<b>Soil Moisture 3 km RMSE</b>	<b>January</b>	<b>April</b>	<b>July</b>	<b>October</b>	<b>Overall</b>
<b>1L_5</b>	0.0367	0.0467	0.0417	0.0383	0.0404
<b>2L_5_5</b>	0.0369	0.0469	0.0415	0.0384	0.0404
<b>1L_10</b>	0.0367	0.0469	0.0415	0.0383	0.0403
<b>2L_10_10</b>	0.0366	0.0470	0.0415	0.0385	0.0404
<b>1L_15</b>	0.0365	0.0467	0.0414	0.0383	0.0403
<b>2L_15_15</b>	0.0368	0.0473	0.0415	0.0384	0.0405
<b>3L_15_15_15</b>	0.0368	0.0475	0.0416	0.0383	0.0405
<b>1L_20</b>	0.0368	0.0472	0.0416	0.0384	0.0405
<b>2L_20_20</b>	0.0367	0.0479	0.0415	0.0385	0.0405
<b>3L_20_20_20</b>	0.0366	0.0474	0.0415	0.0384	0.0405
<b>1L_25</b>	0.0367	0.0470	0.0415	0.0383	0.0404
<b>2L_25_25</b>	0.0367	0.0468	0.0414	0.0386	0.0405

The increased noise that accompanies the enhanced resolution is apparent in the RMSE results from the neural network based soil moisture, similar to the results of the theoretical model. The overall soil moisture accuracy for both the neural networks and the theoretical model increased from about 0.01 RMSE at 36 km resolution to 0.04 RMSE at 3 km. Still, for many applications this increase in error may be acceptable, given the additional features visible at higher resolutions.

The overall accuracy of the neural networks is similar, but slightly better than that of the theoretical model in these comparisons. However, this may change as the necessary brightness temperature corrections are applied. The corrected brightness temperature would likely improve the accuracy of trained neural networks, as well as the theoretical model in extracting soil moisture. Further testing is required to determine the effectiveness of both models with proper corrections in place.

## CHAPTER 5. CONCLUSION

### 5.1 Conclusion

This thesis explores the improvement of enhanced resolution data products in scatterometry and radiometry through the application of feed forward neural networks. This work specifically focuses on wind retrieval for the ASCAT instrument, and passive soil moisture retrieval for the SMAP instrument. Resolution enhancement of these products offers the possibility to perform climate and weather studies at finer scales and in areas that were previously not possible. However, the finer scale comes at the cost of increased noise and uncertainty.

The enhanced resolution ASCAT wind product, known as UHR winds, is well-validated, but inherent ambiguity in the radar backscatter-to-wind model couples with the increased noise at higher resolution and leads to inaccuracies in the product. While there are methods to resolve the ambiguity and smooth out inaccuracies, feed forward neural networks provide a possible method to extract wind speed and direction without the multiple ambiguities that result from current models.

Similarly, SMAP passive soil moisture products have been the subject of resolution enhancement studies since the SMAP radar failed, which originally contributed data used to produce a higher resolution soil moisture product. In this thesis, I validate the use of the rSIR technique to enhance SMAP brightness temperature data and the extracted soil moisture. However, as mentioned previously the enhanced resolution also comes with enhanced noise, which affects the accuracy of the soil moisture product. To address this, I examine the application of neural networks to the soil moisture problem, and compare their ability to handle the enhanced noise versus the traditional soil moisture extraction algorithm.

In the case of ASCAT UHR winds, I found that the feed forward neural networks considered are unable to outperform current models, and often perform significantly worse. For wind speed, I was able to find a neural network architecture which produced similar results to CMOD5, staying within about one or two meters per second of the reference data set. However, the net-

works for wind direction perform significantly worse than CMOD5 on average, with the best network tested having an overall RMSE of about 40 degrees, compared to the CMOD5 which had an RMSE of about 15 degrees. This is most likely due to the same ambiguity issue which occurs in the traditional approach. While in training the neural network finds features which help it to produce accurate results, but those patterns do not always generalize well and thus the validation testing performance is often much worse than the training.

For enhanced resolution SMAP passive soil moisture data, I explore the potential of the rSIR algorithm in providing a 3 km resolution product without the need for a radar contribution. I compare soil moisture extraction with the traditional theoretical approach to a neural network based approach. The methods perform similarly at both the baseline 36 km resolution and the enhanced 3 km resolution, with the neural networks often slightly outperforming the theoretical model. At 36 km resolution, the theoretical model is within about  $0.015 \text{ cm}^3 \text{ cm}^{-3}$  of a validated soil moisture product, where the neural networks are all within about  $0.012 \text{ cm}^3 \text{ cm}^{-3}$ . At the 3 km resolution, the traditional method is within about  $0.045 \text{ cm}^3 \text{ cm}^{-3}$ , while the neural network models are within  $0.041 \text{ cm}^3 \text{ cm}^{-3}$ .

Overall, feed forward networks provide a valid possible method to model unknown relationships with accuracy equal to and sometimes better than theoretical models. However, their unpredictability can lead to inaccuracy in applications where there is a significant noise component, or where inherent ambiguity exists. Theory based models for the soil moisture extraction and wind retrieval problems, despite their occasional inaccuracy, perform much more consistently. In the case of wind retrieval, the overall accuracy of the theoretical model was much higher, as the neural networks struggled with resolving wind ambiguities. For soil moisture, the neural networks generally gave slightly better RMSE accuracy than the theoretical model, but had higher error variance as noise increased.

## 5.2 Contributions

This thesis makes the following contributions to scatterometry and radiometry:

- Comparison of the conventional CMOD5 algorithm for ASCAT UHR wind retrieval compared to feed forward network based models.

The neural networks tested in this thesis struggled to resolve the same ambiguity problem which is found using the more theoretical CMOD5 approach. While the currently implemented ambiguity selection techniques used with the CMOD5 process have certain limitations which can lead to inaccurate wind vector solutions, they provide generally accurate, predictable behavior. The feed forward networks tested in Chapter 3 found relationships which help resolve ambiguities within the training data, but the resultant models perform inconsistently outside of the training set. This motivates the continued use of the well-generalized CMOD5 model and ambiguity selection process.

- Validation of the current soil moisture extraction algorithm for SMAP brightness temperature measurements compared to feed forward network based models.

The theory-based SMAP soil moisture extraction algorithm gives consistent, accurate results, even as the product resolution was increased and more noise was involved. While the neural networks have slightly more accurate performance on average than a theoretical model, the measurement noise increases the variance in neural network error. The theory-based soil moisture extraction method gives similar accuracy at all resolutions to the neural network models, with more consistent results. The corrections to the theoretical model noted in Chapter 2 could improve its accuracy at all resolutions.

- Application and validation of the rSIR method to enhance the resolution of SMAP brightness temperature data and its derived soil moisture data.

The higher resolution soil moisture product extracted from rSIR-enhanced SMAP brightness temperature data is found to give similar results to other validated high resolution soil moisture products. Brightness temperature correction techniques accounting for water body impact and other retrieval quality problems could further improve the accuracy. This is significant, as the default resolution of the passive data is 36 km, which hides several finer scale features. Generally, to achieve an accurate soil moisture product at a finer scale than 36 km or 9 km resolution, combination with radar data is required. This thesis validated an accurate 3 km soil moisture product based only on passive SMAP brightness temperature data.

- Thorough examination of the performance of feed forward neural networks in enhanced resolution ASCAT wind and SMAP passive soil moisture products.

Machine learning is becoming increasingly popular in a wide variety of calculation and classification applications. This thesis explored the potential of feed forward neural networks in scatterometry-based wind retrieval and radiometry-based soil moisture retrieval at enhanced resolutions. While feed forward networks show the potential to accurately learn some relationships and calculations, they also often behave poorly in the presence of noise. For wind retrieval, this unpredictability coupled with inherent ambiguity in the wind retrieval process and gave results significantly worse on average than the theory-based process. For soil moisture, neural networks provide slightly better accuracy on average than the theoretical model, but with more variation in the error.

### **5.3 Future Work**

There are several areas examined in this paper which could be extended in future work. Suggestions for work in some of these areas include:

- Vary the neural network frameworks and training methods used to model wind retrieval and soil moisture extraction algorithms.

This thesis focused exclusively on feed forward networks, but as mentioned in Chapter 1, there are several other styles of network. A different framework may be able to more accurately model the wind and soil moisture calculations, and perform more accurately and predictably in the presence of higher noise. Along with this possible change, variations to the amount and variety of data used in training the networks could be tested. Neural network performance is heavily influenced by the training data provided, and thus performance improvements may be seen simply by altering this data set.

- Apply neural networks to different scatterometry and radiometry derived data products, and different aspects of wind and soil moisture retrieval.

The derived data products from scatterometry and radiometry extend well beyond wind and soil moisture. Though the networks in this thesis were applied with variable success, there

is still enormous potential in machine learning to improve this performance, and apply it to a wide variety of other products, like iceberg tracking or freeze/thaw state. In addition, rather than modeling the full wind retrieval or soil moisture extraction process with a single network, machine learning could be applied to smaller portions of the process like flagging pixels of concern, correcting for contaminating water bodies, or performing ambiguity selection.

- Improve the theory-based soil moisture extraction algorithm with brightness temperature correction, and further validate the rSIR algorithm on SMAP passive data products.

Comparisons in my validation of the enhanced resolution passive soil moisture data using the rSIR algorithm were limited to pixels flagged for good retrieval. This is because I was unable to recreate certain brightness temperature and soil moisture corrections used in the comparison data set. These corrections would enable soil moisture extraction in the areas previously ignored. They would also improve retrieval accuracy in the included pixels in both a theory based approach, as well as trained neural networks. Recreating these corrections would allow for further exploration of the potential that rSIR has to provide high resolution passive data products.

## REFERENCES

- [1] R. D. Lindsley, J. R. Blodgett, and D. G. Long, "Analysis and validation of high-resolution wind from ASCAT," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 5699–5711, 2016. ii, 3, 9
- [2] F. Ulaby and D. G. Long, *Microwave Radar and Radiometric Remote Sensing*. University of Michigan Press, 2013. 1, 3, 4, 5, 9
- [3] A. Tch, "The mostly complete chart of neural networks, explained," 2017. [Online]. Available: <https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464> 2
- [4] Stellen, "7 breakthroughs in internet technology that will change the world," 2018. [Online]. Available: <https://www.stelleninfotech.com/blog/top-7-breakthroughs-in-internet-technologies-that-will-change-the-world> 2
- [5] T. Mai, "Meteorological operational satellite A (MetOp-A)," 2017. [Online]. Available: <https://www.nasa.gov/directorates/heo/scan/services/missions/earth/MetOpA.html> 4
- [6] NDMC, "United states drought monitor," 2019. [Online]. Available: <https://droughtmonitor.unl.edu> 5
- [7] SMAP Mission Team, *SMAP Handbook: Soil Moisture Active Passage*, National Aeronautics and Space Administration, 2013. 5, 6, 12, 13
- [8] Jet Propulsion Laboratory, "SMAP: Soil moisture active passive." [Online]. Available: <https://smap.jpl.nasa.gov/observatory/instrument/> 6
- [9] D. G. Long, M. J. Brodzik, and M. Hardman, "Enhanced-resolution SMAP soil moisture using image reconstruction," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017, pp. 2499–2502. 7, 14
- [10] D. G. Long and M. J. Brodzik, "Optimum image formation for spaceborne microwave radiometer products," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 2763–2779, 2016. 7, 14
- [11] D. G. Long and D. L. Daum, "Spatial resolution enhancement of SSM/I data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 2, pp. 407–417, 1998. 7, 14
- [12] H. Tsukimoto, "Extracting rules from trained neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 2, March 2000. 7, 20, 22, 24, 25

- [13] ESA, “About ASCAT,” 2017. [Online]. Available: [http://www.esa.int/Our\\_Activities/Observing\\_the\\_Earth/The\\_Living\\_Planet\\_Programme/Meteorological\\_missions/MetOp/About\\_ASCAT](http://www.esa.int/Our_Activities/Observing_the_Earth/The_Living_Planet_Programme/Meteorological_missions/MetOp/About_ASCAT) 9
- [14] D. G. Long, “Polar applications of spaceborne scatterometers,” *IEEE Journal of Selected Topics in Applied Earth Observations*, vol. 10, no. 3, pp. 2307–2320, 2017. 9
- [15] R. D. Lindsley and D. G. Long, “Enhanced-resolution reconstruction of ASCAT backscatter measurements,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 2589–2601, 2016. 9
- [16] EUMETSAT, “ASCAT,” 2017. [Online]. Available: <https://www.eumetsat.int/website/home/Satellites/CurrentSatellites/Metop/MetopDesign/ASCAT/index.html> 9
- [17] D. B. Lindell and D. G. Long, “High-resolution soil moisture retrieval with ASCAT,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 7, pp. 972–976, 2016. 9
- [18] ———, “Multiyear arctic ice classification using ASCAT and SSMIS,” *Remote Sensing*, vol. 8, no. 204, 2016. 9
- [19] H. Hersbach, A. Stoffelen, and S. de Haan, “An improved C-band scatterometer ocean geophysical model function: CMOD5,” *Journal of Geophysical Research*, vol. 112, 2007. 9, 11
- [20] R. D. Lindsley, C. Anderson, J. Figa-Saldana, and D. G. Long, “A parameterized ASCAT measurement spatial response function,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, 8 2016. 10
- [21] G. D. Chiara and S. Abdalla, “Active techniques for wind and wave observations: Scatterometer, altimeter (& SAR),” 2014. [Online]. Available: <https://www.ecmwf.int/sites/default/files/elibrary/2014/14672-active-techniques-wind-and-wave-observations-scatterometer-altimeter-sar.pdf> 10
- [22] H. Hersbach, “CMOD5.N: A C-band geophysical model function for equivalent neutral wind,” ECMWF, Tech. Rep., 04 2008. 11
- [23] Ocean and Sea Ice SAF Team, “Algorithm theoretical basis document for the OSI SAF wind products,” Royal Netherlands Meteorological Institute, Tech. Rep., 02 2017. 11
- [24] NOAA, “Wind algorithm,” 1 2014. [Online]. Available: [https://www.ospo.noaa.gov/Products/atmosphere/ascats/wind\\_algorithm.html](https://www.ospo.noaa.gov/Products/atmosphere/ascats/wind_algorithm.html) 11
- [25] A. Stoffelen, S. de Haan, Y. Quilfen, and H. Schyberg, “ERS scatterometer ambiguity removal scheme comparison,” Royal Netherlands Meteorological Institute, Tech. Rep., 2000. 11
- [26] M. J. Brodzik, B. Billingsley, T. Haran, B. Raup, and M. H. Savoie, “EASE-grid 2.0: Incremental but significant improvements for Earth-gridded data sets,” *ISPRS International Journal of Geo-Information*, vol. 1, no. 1, pp. 32–45, 2012. 13

- [27] J. Piepmeier, P. Mohammed, G. D. Amici, E. Kim, J. Peng, C. Ruf, M. Hanna, S. Yueh, and D. Entekhabi, “Algorithm theoretical basis document SMAP L2 & L3 radar soil moisture (active) data products,” Jet Propulsion Laboratory, Tech. Rep., 12 2014. 13, 14
- [28] P. O’Neill, R. Bindlish, S. Chan, E. Njoku, and T. Jackson, “Algorithm theoretical basis document level 2 & 3 soil moisture (passive) data products,” Jet Propulsion Laboratory, Tech. Rep., 06 2018. 14, 16, 48, 50
- [29] S. Chan, “Level 2 passive soil moisture product specification document,” Jet Propulsion Laboratory, Tech. Rep., 11 2013. 14, 17
- [30] D. Entekhabi, N. Das, E. Njoku, S. Yueh, J. Johnson, and J. Shi, “Algorithm theoretical basis document L2 & L3 radar/radiometer soil moisture (active/passive) data products,” Jet Propulsion Laboratory, Tech. Rep., 12 2014. 14, 48
- [31] SMAP Algorithm Development Team and SMAP Science Team, “Ancillary data report: Surface temperature,” Jet Propulsion Laboratory, Tech. Rep., 3 2013. 16
- [32] R. Bindlish, S. Chan, R. Hunt, T. Jackson, and J. Kimball, “Ancillary data report: Vegetation water content,” Jet Propulsion Laboratory, Tech. Rep., 1 2013. 16
- [33] N. Das, “Ancillary data report: Soil attributes,” Jet Propulsion Laboratory, Tech. Rep., 1 2013. 16
- [34] V. L. Mironov, L. G. Kosolapova, and S. V. Fomin, “Physically and mineralogically based spectroscopic dielectric model for moist soils,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2059–2070, 7 2009. 17
- [35] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, pp. 359–366, 1989. 28, 30
- [36] Ocean and Sea Ice SAF Team, “ASCAT wind product user manual,” Royal Netherlands Meteorological Institute, Tech. Rep., 03 2016. 33
- [37] D. Nguyen and B. Widrow, “Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights,” *1990 IJCNN International Joint Conference on Neural Networks*, 1990. 33, 52
- [38] M. T. Hagan and M. B. Menhaj, “Training feedforward networks with the Marquardt algorithm,” *IEEE Transactions on Neural Networks*, vol. 5, no. 6, 11 1994. 33, 52
- [39] N. N. Das, D. Entekhabi, S. Kim, T. Jagdhuber, S. Dunbar, S. Yueh, and A. Colliander, “High-resolution enhanced product based on SMAP active-passive approach using Sentinel 1 data and its applications,” in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2017, pp. 2493–2494. 48
- [40] N. N. Das, D. Entekhabi, S. Kim, T. Jagdhuber, S. Dunbar, S. Yueh, P. E. O’Neill, A. Colliander, J. Walker, and T. J. Jackson, “High resolution soil moisture product based on SMAP active-passive approach using Copernicus Sentinel 1 data,” in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, July 2018, pp. 3768–3770. 48

- [41] N. N. Das, D. Entekhabi, S. Kim, T. Jagdhuber, S. Dunbar, S. Yueh, and A. Colliander, “High-resolution enhanced product based on SMAP active-passive approach using Sentinel 1A and 1B SAR data,” in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2017, pp. 2543–2545. 48