DEVELOPMENT OF AN ADVANCED TECHNIQUE FOR MAPPING AND MONITORING SEA AND LAKE ICE FOR THE FUTURE GOES-R ADVANCED BASELINE IMAGER (ABI)

By

Rouzbeh Nazari

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2010

This manuscript has been read and accepted for the
Graduate Faculty in Engineering in satisfaction of the
dissertation requirement for the degree of Doctor of Philosophy.

Date	Prof. Reza Khanbilvardi NOAA Chair Professor of Civil Engineering Chair of Examining Committee
Date	Dean Mumtaz K. Kassir Executive Officer

Supervisory Committee

Prof. Shayesteh Mahani

Prof. Hansong Tang

Prof. Nir Y. Krakauer

Prof. Fred Moshary

THE CITY UNIVERSITY OF NEW YORK

Abstract

DEVELOPMENT OF AN ADVANCED TECHNIQUE FOR MAPPING AND MONITORING SEA AND LAKE ICE FOR THE FUTURE GOES-R ADVANCED BASELINE IMAGER (ABI)

By

Rouzbeh Nazari

Adviser: Professor Reza Khanbilvardi

Information on ice cover extent, distribution, concentration, ice surface temperature and other physical parameters of the ice pack is needed in numerical weather prediction, ship navigation, water management, and regional/global climate change impact assessment. Ice cover is also a sensitive indicator of climate variations. Ability of satellites to provide global observations at high temporal frequency has made them the primary tool for the ice cover monitoring. The main objective of this research is to explore the potentials of mapping ice cover with the future GOES-R ABI and to develop an automated ice-mapping algorithm which would make maximum use of ABI's improved observing capabilities and its frequent observation will provide an extreme enhanced temporal variability of scene response to improve image classification. Data collected by SEVIRI instrument onboard of the Meteosat Second Generation (MSG) satellite have been used as a prototype. The Northern region of the Caspian Sea has been selected for algorithm development and calibration. The approach used in the algorithm development includes daily cloud-clear image compositing as well as pixel-by-pixel image classification using spectral criteria. Available spectral channels (reflectance and temperature) have been tested and used in a statistical-based approach to

accurately discriminate between water, cloud and ice pixels. To assess the accuracy of ice mapping algorithm and maps, the produced ice maps over Caspian Sea derived with the automated algorithm were compared with interactive snow/ice charts produced with NOAA Interactive Multi-sensor Snow and Ice Mapping System (IMS). The results are promising and an additional screening is undergoing further research in order to reduce some remaining confusions related to water properties and presence of fractional ice.

Acknowledgements

It is gives me a great pleasure to thank the many people who made this thesis possible.

First and foremost I offer my sincerest gratitude to my PhD supervisor, Professor Reza Khanbilvardi, who has supported me throughout my thesis and graduate education. He has been a true role model to me with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my PhD degree to his encouragement and effort and without him this thesis, too, would not have been completed. One simply could not wish for a better supervisor.

I wish to express my thanks to the members of my PhD committee Dr. Shayesteh Mahani, Dr. Hansong Tang, Dr. Nir Y. Krakauer, Dr. Fred Moshary for their time and attention.

I am grateful to many people who have helped me all these years, administrative personal of NOAA CREST and the Department of Civil Engineering. Dr. Hosni Ghedira, Dr. Marouane Temimi, Dr. Shakila Merchant, Mr. Yevgeniy Leykin, Ms. Linda Smith, Ms. Beverly Wynn, Ms. Nitta Monteiro, Ms. Sanchia Peterson, Ms. Carla Cotta, deserve special mention.

I owe my deepest gratitude to my colleagues Dr. Tarendra Lakhankar, Dr. David Gollands, Dr. Yajaira Mejia, Mehrdad Kheiripour, Gerarda Shields, Ms. Cecelia Hernandez, Ms. Vanessa Clark at the City University of New York for all the discussions, cooperation and for the wonderful times we have shared.

I would also like to thank my best friends in life; Dr. Saeid who has encouraged me in the pursuit of higher education, Dr. Sean for his love and friendship and all my other great friends that made the journey of life pleasant.

I am heartily thankful to my entire extended family for providing a loving environment for me. My brothers; Dr. Ramin Nazari for being the most significant character in my life and Dr. Reza Nazari for all his support and opening his home to me, my dear sister Roza, my lovely sister in laws Leeda joon and Tannaz joon for their love and my nephews, Amir, Ali and Ryan joon for being so cute, my uncles and aunts for all the fun memories and my lovely grand parents that I am sure are watching over me all the time.

I am short of words, to express my loving gratitude to my lovely Maryam joon for her patience, kindness, understanding and support.

Lastly, and most importantly, I wish to thank, my parents, whom brought me to life, raised me, supported me, taught me, and loved me.

To my Mom who is the reason behind my graduate education, I dedicate this thesis.

Table of Contents

ABSTRACT IV

ACKNOWLEDGEMENTS VI

TABLE OF CONTENTS VII

LIST OF FIGURES X

LIST OF TABLES XIII

1 INTRODUCTION 1

1.1 BACKGROUND 1

1.2 THESIS OBJECTIVES 4

2 LITERATURE REVIEW 7

2.1 SEA ICE REMOTE SENSING 7

2.1.1 Visible 8

2.1.2 Infrared 10

2.1.3 Microwave 11

2.1.4 Passive Microwave 12

2.1.5 Active Microwave 14

3 METHODALOGY 17

3.1 NORMALIZED DIFFERENCE SEA ICE INDEX (NDSI) 17

3.2 SEA SURFACE TEMPERATURE (SST) 18

3.3 BIDIRECTIONAL REFLECTANCE DISTRIBUTION FUNCTION (BRDF) 21

3.3.1 Physical BRDF vs. Semi-Empirical BRDF 21

3.3.2 BRDF Model in Detecting Vegetation 22

3.3.3 Kernel-based BRDF (semi-empirical) 22

3.3.4 Different BRDF Kernels 23

3.3.5 BRDF on Polar Orbiting Satellites 23

3.3.6 BRDF on Geostationary Satellites (MSG) 24

3.4 INTRODUCTION TO NEURAL NETWORK SYSTEM 24
3.5 INTRODUCTION TO FUZZY LOGIC METHOD 27
3.6 STEPWISE MULTIPLE LINEAR REGRESSION ANALYSIS 27

4 DATA ACQUISITION AND STUDY AREA 31

4.1 THE MSG SEVIRI INSTRUMENT 31
4.2 STUDY AREA 33
4.2.1 Caspian Sea 33

5 ALGORITHEM DEVELOPMENT AND RESULTS 37

5.1 BRDF MODEL DATA COLLECTION 40
5.2 SEA SURFACE TEMPERATURE (SST) 60
5.3 NORMALIZED DIFFERENCE SEA ICE INDEX 67
5.4 DEVELOPMENT OF DYNAMIC THRESHOLD 68
5.5 STUDY OF CHANNELS CORRELATIONS 78
5.6 SEA ICE MAPPING MODEL 81
5.7 DEVELOPMENT OF ARTIFICIAL NEURAL NETWORK (ANN) 85
5.7.1 Methodology 86
5.7.2 Results 88

6 CONCLUSIONS AND FUTURE WORK 97

6.1 CONCLUSIONS 97

7 SUGGESTED FUTURE WORK 104

8 APPENDICES 105

8.1 APPENDIX A 105
8.1.1 Stepwise Multiple Regression Analysis 105
8.2 APPENDIX B 116
8.2.1 Dynamic Threshold Analysis 116
8.3 APPENDIX C 124

8.3.1 Channels Correlations 124

8.4 APPENDIX D 126

8.4.1 MATLAB CODES 126

9 PUBLICATIONS 153

10 REFERENCES 155

List of Figures

Figure 1: Typical Neural Network (3 Input layer, 2 Hidden Layer, 3 Output layer) 26

Figure 2: The MSG SEVIRI Instrument Radiometric Performance Characteristics 33

Figure 3: Caspian Sea map (Left), MODIS Terra image of Caspian Sea (Right) 35

Figure 4: Ice loads on offshore platforms, Van Leeuwen 35

Figure 5: Reflectance of the visible and near infrared channels 38

Figure 6: Hourly variation of reflectance for ice, water and clouds 39

Figure 7: Selected pixels for BRDF model 41

Figure 8 Top- Daily change in SOL, Figure 9 Bottom- Monthly variation of SOL 42

Figure 10 Top- Daily change in SAL, Figure 11 Bottom- Monthly variation of SAL 43

Figure 12, Top-Daily change in RAZ, Figure 13 Bottom-Monthly changes of RAZ 44

Figure 14: Top left- Daily variation of HRV reflection, Top right- Daily variation of R01, Bottom left- Daily variation of R02, Bottom Right- Daily variation of NIR 45

Figure 15: Effect of Sun Irradiance and Sensor Viewing Angles on the Reflected Radiance 46

Figure 16: Stepwise model processing system 48

Figure 17: Pixel Selection and Validation and testing process 49

Figure 18: Simulated vs. Observed HRV reflection values 51

Figure 19: Simulated and Observed Values of HRV vs. Solar Zenith Angles 51

Figure 20: Simulated and Observed Values of HRV vs. Satellite Zenith Angles 52

Figure 21-Left- HRV-1 model, training results, Right- HRV-1 model, Validation results 54

Figure 22: Observe (bottom) and simulated (up) HRV Reflectance (0.6 – 0.9μm), January 23 55

Figure 23: Observed vs. simulated HRV reflectance for pixel [360, 100], [320, 260], [310, 330], [150, 180], [290, 290] and [270, 360] on January 23, 2007 56

Figure 24: Reflectance of four optical channels; emphasizing the importance of Near Infrared channel in Cloud detection 58

Figure 25: Reflectivity of Ice/snow versus Cloud in 0.6, 0.8 and 1.6 μ 59

Figure 26: Absorption of ice (dashed) and water (solid) in different spectral regions 59

Figure 27: First simulation using the proposed SST model 61
Figure 28: Transect of temperature in various regions for comparison of calculated SST with the IR10.8 and 12.0 62
Figure 29: Transect of temperature in various regions for comparison of calculated SST with the IR10.8 and 12.0 63
Figure 30: New SST Model (bottom right) shows consistent ice presence for data taken during winter of 2007 of the Caspian Sea 65
Figure 31: Comparison of new SST model to the classified ice map 66
Figure 32: The Normalized Difference Sea Ice Index (NDSI) obtained from the visible and near infrared channels of MSG SEVIRI 67
Figure 33: Distribution functions of water (left) and ice (right) reflectance for Feb 28, 2009 69
Figure 34: Flowchart of the operation for threshold development 70
Figure 35: Classified images of Caspian Sea mid winter, Feb 28, 2007, with Daily Constant threshold, Ice=White, Water=Blue, Cloud=Gray, Land=Green 71
Figure 36: Hourly variation of reflectance for ice, water and clouds 72
Figure 37: (Left) Threshold values for HRV, R01 and R02 Reflectance during local acquisition time 8:45 to 15:15, (Right) The dynamic threshold (Red) in R01 (0.6μm) classifying Ice and Water Pixels 76
Figure 38: Classified images of Caspian Sea mid winter, Feb 28, 2007, with Dynamic threshold, Ice=White, Water=Blue, Cloud=Gray, Land=Green 77
Figure 39: Scatter Plot (left) of near infrared (band 1.6) vs. visible (0.8), HRV image (right) shows clouds (black oval), Ice (red oval) and Water (white oval) 79
Figure 40: Scatter Plot (left) of NDSI vs. near infrared (band 1.6) HRV image (right) shows clouds (black oval), Ice (red oval) and Water (white oval) 79
Figure 41: Scatter Plot (left) of near infrared (band 1.6) vs. thermal infrared (band 10.8) HRV image (right) shows clouds (black oval), ice cloud (blue), Ice (red oval) and Water (white oval) 80
Figure 42: Flow Chart of the Operation 83
Figure 43: Threshold based ice classification model 84
Figure 44: Flowchart of ANN based Sea Ice Mapping from Geostationary Satellites 88

Figure 45: Reflectance in HRV and optical channels 89

Figure 46: Ice Pixels Observed and simulated reflectance 90

Figure 47: Water Pixels Observed and simulated reflectance 91

Figure 48: Neural Network base hourly ice map with cloud (Ice=light blue, water=blue, cloud= yellow, land=red) 92

Figure 49: Neural Network base Daily Ice map with cloud 93

Figure 50: Neural Network (NN) base multi day image composited cloud-free ice map 94

Figure 51: Neural Network ice concentration mp 95

Figure 52: IMS Map and models generated sea ice maps of February 28th, 2008 of Caspian Sea 99

Figure 53: Models quality control station 99

Figure 54: Model quality control station observed a sudden reflectance change 100

Figure 55: Neural Network simulation of Ice and water pixel 101

List of Tables

Table 1: Microwave wavelengths and frequencies 11

Table 2: Surface temperature algorithms 20

Table 3: Sea surface emissivities obtained from the methodology of Masuda et al. (1988) 20

Table 4 Sample of selected coefficients (b_j) for HRV models 47

Table 5 Statistical results of the Stepwise Multiple Linear Regression calibration for R01 (0.6μm) 50

Table 6 Results of split-sample validation of the HRV models 53

Table 7 Results of split-sample validations of HRV models 53

Table 8: Estimated means (μ) and standard errors (σ) for individual acquisition times (AT) obtained from population distribution frequency plots of ice (normal distribution) and water data (lognormal distribution) 74

Table 9: Threshold values for HRV, R01 and R02 Reflectance channels 75

Table 10: Simulation performances and RMSE values 91

1 INTRODUCTION

1.1 Background

Sea ice forms at the ocean and surface once the surface temperature drops under the freezing point during the cold months. In most mid-latitude regions, sea ice melts during the warm months. However some sea ice remains all year in high latitude and Polar Regions. The freezing point highly depends on the salinity of the water. Salty ocean water freezes at about -1.8 °C (28.8 °F) and fresh water at 0°C (32°F). The entire formation process of sea ice happens in the water opposed to icebergs, glaciers, ice sheets, and ice shelves which mostly originate and grow on land. Most of sea ice forms in both the Arctic and Antarctic regions. In the Northern Hemisphere, sea ice is formed as far south as Bohai Bay, China (approximately 38 degrees north latitude). However, in the Southern Hemisphere, sea ice only develops around Antarctica, occurring as far south as 55 degrees latitude. About 15 percent of the world's oceans are covered by sea ice during some part of the year. Sea Ice plays a major role in local and global climates. Its variation can lead the climate from ice age to global warming by changing the albedo, ocean circulation and ocean heat transfer rate. Alongside of the global climate impacts, sea ice also affects large regions of economic, environmental, social and transportation importance. Such conditions require the government to provide continuous coverage of the ice extent and dynamic. It should be mentioned that spatial and temporal information on sea ice has various end users such as local residents of the ice covered regions to transportation companies, mariners, army and etc. Sea ice monitoring from weather and coastal stations, buoy and ships has history of more than 100 years. However, in the past three decades, the use of satellite data has

gradually developed to the point that today remote sensing based techniques are the main tool in sea ice observation and monitoring.

Polar orbiting satellites data were extensively being used in sea ice related researches and in development of the existing operational tools. Visible channels onboard of polar orbiting satellites are capable of visualization of the sea ice extent and location under cloud free condition. Passive microwave and scatterometer data offer estimates of the large-scale ice drift while Synthetic Aperture Radar (SAR) can be used in order to monitor the ice formation, growth and extent of thick ice even in the presence of cloud.

Polar orbiting satellites orbit the earth longitudinally passing the poles at an altitude of about 850 km (530 miles). With each orbit the satellite scans an area in the west of the previous pass. Since these satellites are in a low orbit and look directly down, their images have high spatial resolution. However, their temporal resolution is lower than geostationary satellites and they capture only a few images per location per day. Consequently, in cloudy days all visible and infrared channels of the polar orbiting satellites would not be able to produce any data and because of the low time resolution they have less chance of catching cloud free sequences. On the other hand, geostationary satellites rotate with the earth in a west-to-east direction directly over the equator at an altitude of 35,800 km (22,300 miles). Since the satellite rotates at the same speed and in the same direction as the earth, it always has the same view of the earth's surface. These satellites are in a higher orbit than polar orbiting satellites. Therefore, each image covers a much greater area allowing the satellites a full-disc view of the Earth but lower spatial resolution compared to polar orbiting satellites. Geostationary satellites circle the Earth in geosynchronous orbit, which means they orbit the equatorial plane of the Earth at a speed matching the Earth's rotation. This allows them to

continuously cover over one position on the surface. GOES satellites managed by the U.S. National Oceanic and Atmospheric Administration (NOAA) provide the kind of continuous monitoring necessary for intensive data analysis which makes the observation and monitoring in cloudy regions possible with higher chance of getting cloud free scenes. GOES satellites mainly designed for imagery are also used to estimate rainfall during thunderstorms, hurricanes for flash flood warnings and to estimate snowfall accumulations and overall extent of snow cover.

GOES satellite sensors also currently detect ice fields and map the movements of sea and lake ice in mid-latitude regions currently. However, because of limited field of view of the geostationary satellites to mid-latitudes and the position of such satellite on the equator, GEOS are unable to view and detect ice in the poles. The United States has two geostationary satellites called, Geostationary Operational Environmental Satellites (GOES). The National Oceanic and Atmospheric Administration (NOAA) is preparing to launch a new series of the next generation of the GOES satellites with advanced sensors in order to respond to the growing need for data. The Advanced Baseline Imager (ABI) will be the next generation of geostationary imager on board of GEOS-R expected to be launched in 2015. The ABI will include a number of improvements over the existing imager besides the additional bands. The ABI will improve the spatial resolution from nominally 4 to 2 km for the infrared bands, as well as increase of the coverage rate.

This imager will have between eight to twelve spectral bands. The minimum eight bands are similar to the five bands on the current GOES-8/11 Imagers, plus a snow and Ice /cloud discriminating 1.6μm band. In the presence of sunlight, the 1.6μm band can be used for cloud/snow/ice discrimination which represents the main motivation of this project. With the

improvement of sensors capabilities and added new channels, sun angle illumination and reflectance variation in visible channels will remain a problem in data processing. Overcoming this problem requires the development of an angle correction tool for derivation of steady and robust reflectance values for each channel in order to accurately classify the target pixels.

The most well-known and widely used angle correction tool is the Bi-directional Reflectance Distribution Function (BRDF). This is an angle correction tool for viewing and describing top of the atmosphere and land surface properties. The BRDF of a given surface depends on the incident and reflected viewing directions and also on properties of the surface itself. There are different types of BRDFs that can be used depending on the observed scene. One of the most common BRDF models is the Kernel-based Semi-empirical.

This model consists of three components: isotropic, volumetric and geometric scattering. They all have linear properties and use few model parameters derived from physically-based BRDF models. Each satellite uses a different type of kernel model and therefore, there is a need for development of a new BRDF model for this research. The details of procedure are presented in the next chapters.

1.2 Thesis Objectives

The main objective of this study is focused on developing an appropriate algorithm for sea ice mapping from Meteosat Second Generation (MSG) SEVIRI data. In order to produce sea ice maps, an algorithm based on spectral bands value and Neural Network will be developed, optimized and validated. METEOSAT SEVIRI images over Caspian Sea at 30 minutes interval have been routinely collected since December 2006 and will be used for this project.

This research has no intention to develop the best sea ice product however, the developed model, is one of the most efficient sea ice mapping tool. The aim of this project is to explore the potentials of the future GOES-R Advanced Baseline Imager (ABI) in sea ice detection, mapping and monitoring.

The final product of this project will be hourly, daily and multi day sea ice maps derived from daily composited images from the U.S. geostationary satellite by implying the ABI's new added bands such as 1.6μm with 30min resolution. European Meteosat Second Generation (MSG) SEVIRI has been used as prototype. Data from this instrument has been collected and processed for this project and used in the development and validation of the technique which will be adapted for the future GOES-R. The algorithm will be tested over seas and large lakes within the METEOSAT domain which receive substantial seasonal sea ice cover. This includes Caspian Sea and Baltic Sea.

The ice identification and mapping algorithm for this research has been developed and tested over the Caspian Sea. The choice of the study areas for this work was driven mainly by the availability of data and the unique location and property of Caspian Sea which is located in the METEOSAT domain.

The final product of this project will be:

- Daily Ice Map with Cloud
- Daily Cloud-free ice map (multi-day image composited approach)

If a pixel remains unclassified after all these tests, it will be either tagged as cloudy pixel or assumed to have the class of the previous day (water/ice). Quality control flags for the pixels

classified with the multi-day approach will be provided based on the number of days used to make the final decision.

In order to achieve the above mentioned objectives in generating the daily and multi-day cloud free ice maps, there is a need for development of robust angle correction models. In this research the three following angle correction models has been developed and used:

- Stepwise Linear Regression Model
- Threshold Base Model
- Neural Network Model
- Ice Concentration Model.

METEOSAT second Generation (MSG) winter 2006 and 2007 data has been collected and processed for this purpose. In general these models use the combination of satellite and solar angle to simulate the reflectance value in the visible channels. The model would correct the data for the high reflectance in early morning and late afternoon when there is a large solar angle and makes the mapping of cloudy days possible. The methodology in developing such models and the results are presented in the following chapters.

2 LITERATURE REVIEW

2.1 Sea Ice Remote Sensing

In broad-spectrum, the acquisition of information of an object or phenomenon, that is not in direct physical or close contact by the use of either recording or real-time sensing devices on board of aircraft, spacecraft or satellite is called remote sensing. Nowadays remote sensing is vastly being used in earth observations and weather forecasts from sea ice mapping to tropical storm predictions.

All substances with a temperature greater than absolute zero (-273 degrees Celsius, or -459 degrees Fahrenheit) emit some form of electromagnetic radiations. Some satellite sensors depend on their application to detect different parts of a spectrum such as visible light reflected from the earth's surface or atmosphere or earth emitted radiation.

Electromagnetic (EM) radiation in the Visible (VIS) 4 to 7μm, Near Infrared (NIR) 0.75-1.4μm, Thermal Infrared (TIR) 5.6 μm to 1cm or microwave bands 1 mm to 1 m of the electromagnetic spectrum are being used in every remote sensing techniques in order to observe the Earth's surface and Top of Atmosphere (TOA) properties.

Sea ice can be observed in the visible, infrared and microwave regions of the electromagnetic spectrum. There are advantages and disadvantages to each type. However, none of the spectral regions allow scientists to optimally view sea ice in all conditions (Nuttall, M., 2005). These observations are being done based on the three following principal measurement techniques applicable to ocean and sea ice:

- Visible and near infrared remote sensing that measures the earth surface reflectance of incoming solar radiation
- Thermal infrared and passive microwave remote sensing that measures the surface thermal radiation
- Active microwave that measures the return signal from an active source and surface backscattered radiation

In the following sections each spectral regions and their ability to detect sea ice will be described in more detail.

2.1.1 Visible

Electromagnetic spectrum with wavelength of 380 to 750 is detectable or visible to human eyes and is thus called visible light. The source of all lights and radiation on the surface of the earth is sun and is reflected off objects around us make it possible for us to see them.

The basic parameter observed by visible light (VIS) is the diffuse reflection percentage or coefficient which is called albedo. The unitless albedo is measured as a value between 0 for dark objects with 100% absorption of the sunlight and 1 for white objects with the total of 100% diffuse reflection of the sunlight.

Sea ice reflects sun radiation in a higher rate causing a much higher albedo compared to other earth surfaces such as, the surrounding land and ocean. Ocean surface reflects 6 percent of the incoming solar radiation and absorbs the rest which makes the albedo of approximately 0.06, while sea ice reflects 50 to 70 percent of the incoming energy with albedo of approximately 0.5 to 0.7. This difference in albedo makes it easy to detect sea ice from visible remote sensing instruments.

As mentioned there are limitations involved in the employment of each spectral bands in sea ice observations from space. The following paragraphs include some of the limitations in using reflected visible light.

Visible data can only be collected in daytime since these sensors measure reflected radiation from the sun. With taking in to consideration that sea ice happens in high latitude regions that have shorter daytimes and darkness prevails, inability to have observation at night is a problem of visible data in sea ice monitoring,.

Visible electromagnetic waves do not penetrate clouds and a cloudy sky prevents satellites from viewing visible light reflected from sea ice. Therefore, observation of the Earth's surface requires cloud free condition which is a rather sever limitation since sea ice exist in the regions that tend to be cloudy most of the time.

The following satellites and sensors are a selection of those that observe visible radiation and often used to observe and map sea ice.

- The Defense Meteorological Satellite Program (DMSP) Operational Linescan System (OLS)
- The National Oceanic and Atmospheric Administration (NOAA) Advanced Very High Resolution Radiometer (AVHRR)
- The National Aeronautics and Space Administration (NASA) Moderate Resolution Imaging Spectroradiometer (MODIS)

2.1.2 Infrared

Electromagnetic radiation with a wavelength of about 0.75 μm to 1mm which is between visible and microwave radiation is called infrared radiation. Any objects on the surface of the planet emit infrared radiation with a specific wavelength which can be detected with satellite sensors. For example, human body with the average temperature of 37 degree Celsius can radiate at approximate wavelength of 10 micrometers. Like any other object sea ice along with its surrounding environment such as ocean and land emit infrared radiation. However, the emission rate of any of these objects depends on their temperature and since there is a distinct temperature difference between sea ice and ocean water, emitted infrared radiation is distinguishable. This quality makes the infrared band a useful discrimination tool in sea ice observation. Sea ice surrounding ocean temperature is above the freezing point where sea ice temperature is well below 2 degree Celsius. Sea Ice and cloud have very similar reflection patterns in the visible spectral channels, 0.6-0.8 μm, which makes the classification process very difficult. While in near infrared spectral channels sea ice shows a very low reflectance values (approximately below 0.1) which can be used in sea ice and cloud discrimination. Like visible radiation, there are some limitations involved in measuring infrared radiation from space. Infrared radiation can not penetrate cloud since its wavelength is smaller than cloud droplets diameter. Clouds emit and reflect infrared radiation and also block the reflected infrared radiation from the sea ice surface making satellites unable to detect sea ice. Sea ice melts during the summer and the surface temperature increases to the freezing point. Melting sea ice as well as freezing and shallow sea ice is difficult to distinguish from surrounding ocean that is also near the freezing point. The following satellites and sensors observe infrared radiation:

- The Defense Meteorological Satellite Program (DMSP) Operational Linescan System (OLS)
- The National Oceanic and Atmospheric Administration (NOAA) Advanced Very High Resolution Radiometer (AVHRR)
- The National Aeronautics and Space Administration (NASA) Moderate Resolution Imaging Spectroradiometer (MODIS)

2.1.3 Microwave

The microwave region of the electromagnetic spectrum has frequency in the range of 0.3 to 300 GHz and sub-divided in the various band designations (Table 1). The greatest advantage of this region of the spectrum is the possibility to observe earth surface in all weather conditions, which is not possible in visible or infrared region. The microwave remote sensing is classified as passive and active microwave remote sensing, having distinct advantages over each other.

Table 1: Microwave wavelengths and frequencies

Band Designations	Wavelength (cm)	Frequency (GHz)
K_a	0.75 - 1.18	40.0 – 26.5
K	1.19 – 1.67	26.5 – 18.0
K_u	1.67 – 2.4	18.0 – 12.5
X	2.4 – 3.8	12.5 – 8.0
C	3.9 – 7.5	8.0 – 4.0
S	7.5 – 15.0	4.0 – 2.0
L	15.0 – 30.0	2.0 – 1.0
P	30.0 – 100	1.0 – 0.3

2.1.4 Passive Microwave

The portion of electromagnetic spectrum with wavelengths ranging from 1 mm to 1 m and frequencies between 300 MHz and 300 GHz are so called microwave radiation. On the surface of the planet earth objects naturally emit infrared radiation in various rates. Objects also emit some relatively low energy level microwave radiation, which when detected with satellite sensors is called passive microwave. The amount of received passive microwave energy in satellite's field of view is related to the temperature and moisture properties of the emitting object or surface. Passive microwave measures a parameter called Brightness Temperature (BT). Brightness temperature is calculated by the following equation:

Brightness Temperature (BT) = Surface Temperature (ST) × emissivity (e)

Sea ice and water have large variation of emissivity in the most common microwave channels with frequencies of about 6 to 90 GHZ. In comparison with sea ice, clouds do not emit much microwave radiation. Microwaves with wavelengths larger than 0.3 cm surpass the cloud droplet diameter size and makes them able to penetrate clouds which can be used to detect sea ice during the day and night regardless of cloud coverage.

The amount of microwave radiation emitted from an object does not strongly depend on the temperature of an object as it was in infrared; instead is related to the object's physical properties such as atomic composition and crystalline structure. The crystalline structure of ice typically emits more microwave energy than the liquid water in the ocean (Nuttall, M., 2005) and therefore, sea ice can easily be distinguished from ocean with passive microwaves sensors.

Like any other spectral bands there are advantages and disadvantages in applying passive microwave in sea ice observation which will be stated in the following paragraphs.

Microwaves at wavelengths longer than 0.3 cm have the ability to penetrate clouds and detect sea ice. Since microwave radiations do not depend on solar radiation, obtaining the regular day and night observations is feasible regardless of the presence of cloud.

Passive microwave sensors provide nearly complete images of all sea ice-covered regions every day, because of their ability to detect sea ice through clouds during the day and night. These sensors have provided the most complete and long-term observations of sea ice, allowing scientists to detect notable changes in Arctic sea ice.

One major problem with passive microwave is that the collected data in each pixel has quite low energy level and as a result of that the radiation requires collection over a large region with footprints of typically 10-60 km, depending on the frequency, which will cause in the loss of sea ice details.

The following satellites and sensors have observed passive microwave radiation:

- The National Oceanic and Atmospheric Administration (NOAA) Electrically Scanning Microwave Radiometer (ESMR)
- The National Aeronautics and Space Administration (NASA) Scanning Multichannel Microwave Radiometer (SMMR)
- The Defense Meteorological Satellite Program (DMSP) Special Sensor Microwave/Imager (SSM/I)
- The National Aeronautics and Space Administration (NASA) Advanced Microwave Scanning Radiometer–Earth Observing System (AMSR-E).

2.1.5 Active Microwave

The portion of electromagnetic spectrum which has wavelengths ranging from 1 to 30cm and 1 to 30GHZ is called active microwave. Active microwave systems provide their own radiation which is transmitted to the earth surface and make them independent of solar radiation. Therefore, active microwave sensors can operate day or night making the 24 hours monitoring possible. The sensor measures the microwave radiation reflected off the surface returning to the satellite and collected by antennae. The amount and characteristics of the returning radiation highly depends on the physical properties of the object. Microwaves that propagate through clouds and rain with limited attenuation provide information and observation in cloud covered regions. This technology is also being used in order to track aircraft, ships, and speeding automobiles. Active microwave observations can be obtained by several types of instruments. Three types of active microwave sensors are used to detect sea ice and the most commons are; Synthetic Aperture Radars (SAR), Side-Looking Radars (SLR) and scatterometers. The SAR instruments provide high-resolution images, with pixel size down to about 10 m, while the SLR provides medium-resolution images with resolution of 1–2 km. Scatterometer data have coarser resolution varying from 10 km to 50 km.

a) Imaging Radar

Imaging radar is similar to a photograph taken by a camera but the image is of radar waves not visible light. Sea ice typically reflects more of the radar energy emitted by the sensor than the surrounding ocean which makes it easy to distinguish between the two. However, the amount and character of reflected energy are functions of the physical properties of the sea ice, which can be quite complex, thus it can be difficult to interpret radar images of sea ice. Although, thicker multiyear ice is readily distinguishable from younger thinner ice

because radar energy bounces back to the sensor from the bubbles in the ice left when the brine drains. This feature makes synthetic aperture radar (SAR) an especially useful tool for measuring the extent of thick versus thin sea ice.

SAR is a special type of imaging radar that involves advanced technology and complex data processing to obtain detailed images of sea ice. The RADARSAT mission managed by the Canadian Space Agency is the primary SAR mission today. SAR instruments can even detect small leads in sea ice. This fine resolution allows them to analyze sea ice and help route ships through ice-covered regions. SAR imagery is particularly valuable for operational ice centers.

b) Non-imaging radar

Non-imaging radars are also called scatterometer. These sensors measure the amount of reflected energy or backscatter from the earth's surface. It cannot obtain the same detail as a SAR sensor but it does provide a complete daily data on sea ice day and at night, through cloud cover. Images from non-imaging radar have about the same level of detail as passive microwave imagery. The SeaWinds sensor aboard NASA's Quick Scatterometer (QuikSCAT) satellite provides daily global views of ocean winds and sea ice.

c) Altimetry

This sensor sends a pulse of radar energy toward the earth and measures the time it takes to return to the sensor. The pulse's round-trip time determines how far the satellite is from the reflecting surface. With a known reference, this information is used to measure the altitude of various features at the earth's surface. With enough precision a radar altimeter can determine the height of the sea ice surface above sea level which scientists use to calculate

the total thickness of the sea ice. Early satellites with radar altimeters were not in orbits that adequately covered the poles and did not collect substantial sea ice data. Another type of altimeter called laser altimeter sends pulses of visible light toward the earth.

3 METHODALOGY

3.1 Normalized Difference Sea Ice Index (NDSI)

The Normalized Difference Sea Ice Index (NDSI) in simple wording is a numerical indicator or index value that analyzes remote sensing measurements such as reflection values in order to assess the existence of sea ice in a given pixel.

The Normalized Difference Sea Ice Index (NDSI) is the modified version of Normalized Difference Snow Index (NDSI) for sea ice applications to distinguish sea ice from Open Ocean based on reflective characteristics.

The NDSI is also useful for the identification of sea ice and most cumulus clouds and is a measure of the relative magnitude of the difference characteristic reflectance between the visible and short-wave IR reflectance of sea Ice. NDSI, which is insensitive to a wide range of illumination conditions is partially normalized for atmospheric effects and it does not depend on reflectance in a single band. The NDSI is equivalent to the normalized-difference vegetation index (NDVI) (Tucker, 1979 and 1986; Townshend and Tucker, 1984).

NDSI values have been found to be 0.4 in pixels with approximately 50% or more snow coverage (Rosenthal and Dozier, 1996). A very similar concept has been applied in the current sea ice algorithms since it is primarily modified from snow index. Sea ice and snow behave very similar in the channels used in NDSI.

Shallow and new forming sea ice has a low NDSI but NDSI increases as it is completely formed into pure sea ice. Thus, sea ice in mixed pixels has an NDSI that is less than what

normally is for pure sea ice. Pure sea ice can be distinguished by its high NDSI value. METOSAT-8 bands 1 (0.6 μm) and 3 (1.6 μ m) have been used to calculate the NDSI.

$$NDSI = (band\ 1 - band\ 3)/ (band\ 1 + band\ 3)$$

One of the classification threshold for a pixel to be mapped as sea ice is that the NDSI value to be 0.4 or more. Reflectance values are between 0-100 percent and the NDSI ratio should be within -1.0 to +1.0.

3.2 Sea Surface Temperature (SST)

Sea surface temperature is the satellite infrared radiometer's indirect measure of the top layer, approximately 10cm of ocean, which is also called skin temperature because infrared radiation is emitted from the outer layer.

The SST measurement via satellite is being done by sensing the ocean emitted radiation in two or more infrared or any other channels that could be related to SST.

In order to retrieve sea surface temperature from METOSAT-8 Spinning Enhanced Visible and Infrared Imager (SEVIRI) data, there is a need to apply one of the three existing SST estimation algorithms (M. ROMAGUERA et al, 2005).

The three algorithms structure and errors of them are shown in Table 2. The first two algorithms are based on split window technique and the last one is based on a nonlinear sea surface temperature (NLSST) algorithm (Walton 1988).

The first one is a general algorithm for surface temperature (ST) estimation given by Sobrino and Romaguera (2004) for SEVIRI sensor. This data is presented in the Table as LST also known as land surface temperature, where ST is the surface temperature (K), $T_{IR10.8}$ and

$T_{IR12.0}$ are the at-sensor brightness temperatures of the IR10.8 and IR12.0 SEVIRI thermal channels (K), ε is the mean effective emissivity

$$\varepsilon = (\varepsilon \text{ IR10.8} + \varepsilon \text{ IR12.0})/2$$

Δε is the emissivity difference

$$\Delta\varepsilon = \varepsilon \text{ IR10.8} - \varepsilon \text{ IR12.0}$$

W is the total atmospheric water vapor (g/cm^2) and θ is the satellite zenith observation angle. The second algorithm shown in Table 2 (SST) is based on the Sobrino and Romaguera (2004) procedure using the values of sea surface emissivity obtained from Masuda et al. 1988 (See Table 3). Finally, in the NLSST algorithm; Tsurf is a priori SST estimate and Secθ is the secant of the observation angle. The coefficients of the equation in this algorithm are obtained by regression and the value of Tsurf is the temperature of the lower layer of the radiosondes.

The second algorithm, SST model, has fitted this research very well and was chosen based on the availability of data and performance of the algorithm. As it could be seen in Table 2, SST has the least error and Root Mean Square Error (RMSE) which makes it a better method in comparison to the other algorithms.

	Algorithm	Minimization error (K)	Rms error (K)
LST	$ST = T_{IR10.8} + [3.17 - 0.64\cos\theta]\cdot(T_{IR10.8} - T_{IR12.0}) + \left[-0.05 + \frac{0.157}{\cos\theta}\right](T_{IR10.8} - T_{IR12.0})^2 + + \left[65 - \frac{4}{\cos^2\theta}\right](1-\varepsilon) + \left[-11.8 + \frac{5.1}{\cos\theta}\right] W(1-\varepsilon) + \left[-180 + \frac{24}{\cos\theta}\right]\Delta\varepsilon + [-4 + 34\cos\theta] W\Delta\varepsilon - 0.6$	1.5	1.7
SST	$SST = T_{IR10.8} + [0.99\cos\theta + 0.21](T_{IR10.8} - T_{IR12.0}) + + \left[\frac{0.364}{\cos\theta} + 0.15\right](T_{IR10.8} - T_{IR12.0})^2 + \left[\frac{0.327}{\cos^2\theta} + 0.11\right]$	0.7	0.8
NLSST	$SST = 0.971 T_{IR10.8} + 0.00954(T_{IR10.8} - T_{IR12.0}) T_{surf} + + 0.85(T_{IR10.8} - T_{IR12.0})[\sec\theta - 1] + 7.9$	0.8	0.8

Table 2: Surface temperature algorithms

Observation angle	$\varepsilon_{IR10.8}$	$\varepsilon_{IR12.0}$
0	0.9919	0.9897
10	0.9919	0.9896
20	0.9918	0.9895
30	0.9912	0.9887
40	0.9892	0.9860
50	0.9831	0.9775
60	0.9641	0.9526

Table 3: Sea surface emissivities obtained from the methodology of Masuda et al. (1988)

3.3 Bidirectional Reflectance Distribution Function (BRDF)

The BRDF is the Bi-directional Reflectance Distribution Function. It is a form of correction when viewing and describing land cover, cloud detection, and vegetation at different sun illumination and satellite viewing angles. There are different types of BRDF that can be derived depending on what is observed.

Several types of BRDF models have been formed and used; some of those are mentioned in the following:

1) Specular reflectance: mirror BRDF over flat land, is a perfect or mirror-like reflection of light off of a surface. Light comes in from one direction and is reflected in another direction. The angle of incidence equals the angle of reflection.

2) Water surface BRDF: sunlight reflectance over water

3) Volume scattering BRDF: leaf and vegetation reflectance, when light is reflected and scattered within the vegetation, measuring density and structure of the vegetation.

4) Gap-driven BRDF (forest): shadow driven reflectance.

It should be mentioned that the BRDF of a given surface depends on the incident (sun angle) and reflected (satellite angle) viewing directions and also on the properties of the surface itself.

3.3.1 Physical BRDF vs. Semi-Empirical BRDF

Physical BRDF models are related to the biophysical structure of what is being viewed. Semi-empirical kernel-based BRDF models are the relationship between BRDF parameters which are mathematically derived. There are different semi-empirical BRDF models which can be suited to different surface types. In detecting vegetation structure using a Kernel

based BRDF Model, a structural scattering index (SSI) and a relative structural scattering index (RSSI) were derived based on BRDF parameters. Semi-empirical BRDF models consist of three components of basic scattering, isotropic, volumetric, and geometric (F. Gao. *et all*, 2003).

3.3.2 BRDF Model in Detecting Vegetation

The BRDF model establishes a relationship between the bidirectional reflectance and spectral and structural features when dealing with vegetation.

There are two ways to relate vegetation structure of a land surface cover to its anisotropic reflectance pattern. One way is based on the reflectance ratios at different viewing or illumination geometries connected with structural information and bidirectional reflectance. The second way is to use an inversion method to retrieve physical parameters from a physical BRDF model. The inversion method requires some form of knowledge regarding the area being observed therefore it is not an effective method when researching a large area.

3.3.3 Kernel-based BRDF (semi-empirical)

Semi-empirical BRDF models consist of three components; isotropic, volumetric and geometric scattering. They have linear properties and use few model parameters, all being derived from physically-based BRDF models. The basic description is

$$BRDF = f_{iso} + f_{vol} * k_{vol}(\theta_i, \theta_v, \varphi) + f_{geo} * k_{geo}(\theta_i, \theta_v, \varphi)$$ (F. Gao. *et al*, 2003)

Where k_{vol} is a function view of the zenith θ_v and illumination zenith θ_i and relative azimuth φ describes the volume scattering from the canopy and k_{geo} describes the surface scattering from canopy.

Coefficients that are determined by a set of observed reflectances are given to each kernel. There are different kernels that have been developed which are suitable to a set of observed reflectances. Each satellite uses a different type of kernel.

3.3.4 Different BRDF Kernels

The following are the most used kernels BRDF models:

- Ross-thick: this model works best on large leaf area index (thick vegetation canopies) and volumetric scattering.
- Ross thin: works on smaller leaf area index and volumetric scattering.
- Li-Sparse: functions the best on bare soil and sparse vegetation and geometric scattering
- Li-Dense: this model is for dense vegetation canopies and geometric scattering.
- Roujean Model: best suited to a linear formula of a kernel-driven BRDF

3.3.5 BRDF on Polar Orbiting Satellites

MODIS uses Li-Sparse, Li-dense or both which treat canopy crowns as spheroids and have two non-linear parameters (b1, b2) (Baoxin Hu, 1996)

MODIS BRDF- land surfaces are modeled as the weighted sum of 3 kernels:

a) Isotropic surface

b) Ross-type

c) Li-type (b2=2.0)

On MODIS, the high leaf transmittance in the near infrared channels for vegetation results in high multiple scattering within the canopy. This feature is best described by volumetric scattering. For small leaf transmittance in the red band (VIS) can best be described by geometric scattering. Therefore volumetric and geometric scattering are the most sensitive to structure in vegetation. The total BRDF for the surface is then related to the top-of-atmosphere (TOA) BRDF (as seen by MODIS) by means of an atmospheric correction.

POLDER BRDF- 3 kernel combo known as Roujean model

1. Lambertian term

2. Ross-thick volume scattering

3. Simple parameter-free geometric optics

3.3.6 BRDF on Geostationary Satellites (MSG)

Measured top-of-atmosphere radiances are corrected for atmospheric effects, in order to become the top-of-canopy reflectance values. The top-of-canopy spectral reflectance values become inputs for the inversion of a linear kernel-driven BRDF model and it takes into account angular dependence of the reflectance factor. A narrow to broadband conversion is applied with linear regression formula (Bernhard Geiger, 2003).

3.4 Introduction to Neural Network System

Neural networks (NN) have been applied to a wide range of problems in remote sensing technology. They have been increasingly used since 1988 for the classification of remotely sensed images (Benediktsson *et al.* 1990; Paola, and Schowengerdt 1995; Low *et al.* 1999; Carpenter *et al.* 1999). Neural Network applications in remote sensing have rapidly

increased mainly due to its more accurate performance than other classification techniques. The other advantages of NN over conventional classification techniques are; the ability to handle data acquired at different levels of measurement precision and its fast processing time after training the network (Foody and Arora, 1997).

Multi-layer perceptron trained by back-propagation algorithm is the most common neural network used for image classification. This type of neural network has been successfully applied to image processing and has shown a great potential in the classification of different remotely sensed data. Furthermore, neural networks can provide an alternative for conventional statistical methods such as maximum likelihood or Bayesian methods. A useful review of the application of neural networks in remote sensing may be found in Benediktsson *et al.* 1990; Paola and Schowengerdt 1995. A typical neural network consists of an interconnected group of artificial neurons that processes information using a connectionist approach to computation is shown in Figure 1. The major advantages of the neural network method over traditional classification methods are (Ghedira *et al.* 2000):

- Easy adaptation to different types of data and input configuration. Moreover, neural networks can easily incorporate ancillary data which would be difficult or impossible with conventional techniques.
- The traditional parametric classification methods such as the Maximum Likelihood classifier make unreasonable assumptions about the statistic proprieties of the data even though, they are normally (or Gaussian) distributed for each ground cover class. However, this assumption is not always satisfied.

- Neural network use its complex configuration to find the best nonlinear function between the input and the output data without the constraint of linearity or pre-specified non-linearity which is required in regression analysis.
- Neural networks are able to assign more than one label of land cover classes to each pixel in an image. This propriety resolve the mixed pixel problem usually observed in image classification.

Given this list of advantages and the several unique capabilities of neural networks in remote sensing, it will be advantageous to test the ability of this tool for estimation and mapping sea ice from MSG SEVIRI images. In order to accomplish this task, several parameters of the neural network have been varied to extract the maximum of information from MSG SEVIRI data.

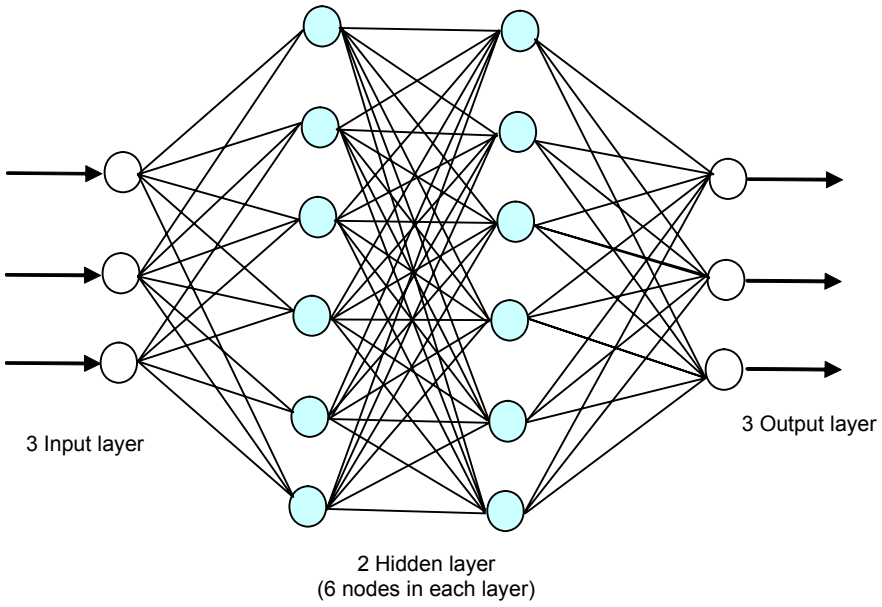


Figure 1: Typical Neural Network (3 Input layer, 2 Hidden Layer, 3 Output layer)

3.5 Introduction to Fuzzy Logic Method

The fuzzy set theory developed by Dr. Lotfi Zadeh in 1965 is a mathematical tool that addresses imprecise and vague data. In the classical fuzzy logic, an element is expressed in binary terms: 0 or 1, yes or no, black and white; in terms of Boolean algebra. A fuzzy set generalized the Boolean set, by allowing partial membership in a set, with values ranges from 0 to 1. The Fuzzy Logic (FL) technique based on fuzzy set theory has been applied in many areas where relationship between variable is not empirically defined. The elements of a fuzzy set have different degree of membership values. The membership value ranges between 0 and 1 depends upon partial or full membership.

A mathematical function defines the degree of an element's membership in a fuzzy set is called membership function. The fuzzy logic has been used in very wide the areas of applications; process control, management and decision making, operations research, economics and engineering. The advantage of fuzzy logic is in dealing with uncertainty and imprecision in a decision-making process, and thus offers a new approach for classifying remotely sensed images (Nedeljkovic, 2004).

3.6 Stepwise Multiple Linear Regression Analysis

Multiple linear regressions are extensions of simple linear regression treating more than one independent variable to form a relative predictive model of the independent variables by creating beta weights. Multiple regressions can establish a set of independent variables having a fraction of the variance in a dependent variable at a significant level (through a significance test of R^2). Curvilinear effects can be explored by adding independent variables as a power term in the model. Interaction effects can be tested by adding the cross-product

terms of independent variables. The effect of addition of an independent variable to the model can be tested based on two R^2's values as a test of significance. Hierarchical regression can be used to compare the variance in the dependent due to one or a set of new independent variables over others. The estimates of b coefficients can be used to construct a prediction equation on a variable for further analysis (Gilfilian and Page 1986 ; Khoshgoftaar et al. 1995).

The stepwise regression method is used to generate the multivariable model. The data fitting in stepwise regression is tested by multiple correlations and an overall test of significance. Multiple correlations are actually values of r^2 for the observed values versus predicted values. The test of significance is done by:

- a standardized regression coefficient (*b* if all variables are standardized)
- *t* value, and
- *p* value associated with that *t* value.

The standardized coefficient equates the value of *r* between the variable of interest and the residuals from the regression. The general multivariable model is expressed as:

$$y = ß_0 + ß_1x_1 + ß_2x_2 + ß_3x_3 + ß_4x_4 + ß_5x_5 + ß_6x_6 + ß_7x_7 + \ldots\ldots + c \qquad (6.6)$$

The ß's are the regression coefficients. These express the amount to which the dependent variable *y* changes with the change in 1 unit of the corresponding independent variable. The $ß_0$ is the constant intercept of the regression line at the y axis which represents the amount of the dependent *y* when all the independent variables are 0. The standardized versions of the *b* coefficients are expressed as the beta weights. The ratio of the beta coefficients determines the relative predictive power of the associated independent variables. The values of b

coefficients and the constant are used to generate a prediction equation. The determination coefficient R^2 associated with regression model is the percent of variance in the dependent variable explained collectively by all of the independent variables (Khoshgoftaar et al. 1995).

The basic procedures in stepwise regression analysis involve; (A) identifying an initial model, (B) iteratively altering the initial model by adding or dropping an independent variable in agreement with the "significant test criteria", and (C) terminating the search when stepping is no longer possible given the significant test criteria, or when a specified maximum number of steps has been reached (Khoshgoftaar et al. 1995). Three techniques are used in stepwise regression analysis:

1. Forward Selection: In this method an independent variable is selected which is responsible for higher variation in the dependent variable. If the second variable selected has a higher residual variation, then its regression coefficients are recalculated, until no variables "significantly" explain residual variation.

2. Backward Selection: In this method all variables in the model are selected and the least "significant" is dropped one at a time until only "significant" variables are left.

3. Mixture Selection: In this method, the forward selection is used. However, variables with no "significant" are dropped after introduction of new variables.

The significance test is performed using the *p*-values to compare the effect of different variables on regression analysis. It is necessary to run a stepwise procedure a number of times using random selection of data to find meaningful patterns.

In this study, a stepwise regression analysis has been used by choosing band reflectance value as dependent variables to satellite, solar and azimuth angles.

$$R = c + ß_1 * ARZ + ß_2 * SAT + ß_3 * SOL$$

The model coefficients have been estimated based on several runs (100) of stepwise regression analysis with different datasets. Where *R* is reflectance in a given channel, SAT is satellite viewing angle, SOL is sun angle and ARZ is the azimuth angle. Further, an effort has been made to estimate the values of these variables based on different input parameters.

4 Data Acquisition and Study Area

4.1 The MSG SEVIRI Instrument

Data collected by SEVIRI instrument onboard of Meteosat Second Generation (MSG) satellite have been used as a prototype. The Northern region of the Caspian Sea has been selected for algorithm development and calibration. The approach used in the algorithm development includes daily cloud-clear image compositing as well as pixel-by-pixel image classification using spectral criteria. All available spectral channels (reflectance and temperature) should be tested in order to be used in a statistical-based approach to accurately classify cloud, water and ice. The reflectance of thick ice cover is very high in the visible and drops substantially in the shortwave and middle-infrared. This specific spectral signature provides the physical basis for ice identification from space. It will be primarily used in the new ice detection algorithm for GOES-R ABI. Clouds present the major factor hampering ice identification and mapping. In comparison to polar orbiting satellite data, availability of frequent observations from geostationary satellites increases the chance to obtain a cloud clear view during the day which helps to reduce cloud gaps in the ice map. The next step of the work will be the bidirectional reflectance distribution function (BRDF) correction which reduces the angles effect on our classification. The Stepwise Multiple Linear Regression (SMLR) analysis is performed on the angles and the trigonometric and geometric combinations of them in order to derive the best equation to simulate the values of the channels and reduce the angular effect.

The Imaging Radiometer SEVIRI has a permanent imaging capability of the Earth using 12 spectral channels with a time resolution of 15 minutes and the spatial resolution of 3 km for standard channels and 1 km for the High Resolution Visible (HRV) channel.

The MSG SEVIRI 12 spectral channels have the following functions:

- Two visible channels VIS 0.6 and VIS 0.8 are used to provide cloud and land surface imagery during daytime. These wavelengths help in the discrimination of vegetated surfaces from clouds at different periods of the year and in the determination of the vegetation index and aerosol loads.
- The NIR1.6 channel is used to discriminate clouds from ice and water. In combination with the two visible channels (VIS0.6 and VIS0.8) it improves the observation of ice, water and sea ice index.
- The HRV Channel is implemented in support of the now-casting.
- IR6.2 and IR7.3 channels are used to determine the water vapor distribution in two distinct layers of the atmosphere. They are also used in combination with the long wave IR window channels for temperature determination of thin clouds which can appear warmer than they are because of earth background and wind determination in cloud free areas.
- The four channels IR3.8, IR8.7, IR10.8 and IR12.0 provide continuous cloud observation along with a temperature estimate of clouds, land and sea surfaces. IR3.8 is especially used at night to detect fog and very low clouds. This channel is less useful during daytime as the sun illumination is not negligible at this wavelength.

- Channels IR9.7 and IR13.4 are used to meet the Air Mass Analysis mission and improve the Basic Multi-Spectral Imaging, Cloud Motion Vectors and Surface Parameter Performances.
- Channel IR 9.7 also belongs to the ozone absorption band and is used for the monitoring of the upper atmosphere, mainly tropopause features and stratospheric winds. It may also monitor the total ozone content.
- IR13.4 lies in the CO2 absorption band and is intended to being used for cirrus discrimination, cloud top pressure evaluation, a cloud track and wind height assignment.

The MSG SEVIRI radiometric performance capabilities and characteristics can be seen in the Figure 2.

Channel (μm)	HRV	0.6	0.8	1.6	3.9	6.2	7.3	8.7	9.7	10.8	12.0	13.4
Noise	0.52	0.39	0.36	0.08	0.24	0.40	048	0.15	0.24	0.13	0.21	0.29
Spec.	1.07	0.53	0.49	0.25	0.35	0.75	0.75	0.28	1.50	0.25	0.37	1.80

Figure 2: The MSG SEVIRI Instrument Radiometric Performance Characteristics

4.2 STUDY AREA

4.2.1 Caspian Sea

The Caspian Sea is the world's largest lake and because of its surface area of 371,000 square kilometers (143,244 sq mi) and a volume of 78,200 cubic kilometers (18,761 cu mi) has been called and classified as sea. It is a closed body of water which has no outflows. It is located between the northern Iran and has a shared coastline with Azerbaijan, Kazakhstan, Russia, and Turkmenistan. The maximum depth of Caspian Sea reaches to about 1025 meters (3,363

ft). It was perceived as a sea by its ancient coastal inhabitants because it is salty and seemed extremely large and boundless. In fact, it is the remainder of an ancient ocean and has a salinity of approximately 1.2% (about a third the salinity of most seawater). The Northern Caspian Sea is very shallow and accounts for less than one percent of the total water volume with an average depth of only five to six meters. The sea bed noticeably drops off towards the Middle Caspian where the average depth is 190 meters. The Southern Caspian is the deepest part of the sea. The Middle and Southern Caspian account for 33 percent and 66 percent of the total water volume, respectively. The northern portion of the Caspian Sea typically freezes in the cold months of winter. It has been observed that in the coldest winters, ice forms in the south as well. Along with the Volga River being the largest over 130 rivers pour into the Caspian Sea. There are some small islands in Caspian Sea which are primarily located in the North and have a total land area of approximately 2000 square kilometers. The biomes located at the north and the east is characterized by cold, continental deserts. Conversely, the climate to the southwest and south are generally warm with uneven elevation due to a mix of highlands and mountain ranges. The drastic changes in climate alongside the Caspian have led to a great deal of biodiversity in the region. Despite the fact that the water temperature in the Caspian Sea may rise up to +30°C in summer time the northern Caspian Sea is exposed to ice from about early December to late March (with ice thickness of up to 2 m). Ice can cause severe loads on structures (such as the platform at the Kashagan East exploration well, (**Error! Reference source not found.**). (Pieter H.A.J.M. van Gelder *et al.* 2004).

Figure 3: Caspian Sea map (Left), MODIS Terra image of Caspian Sea (Right)

Figure 4: Ice loads on offshore platforms, Van Leeuwen

Studies of ice in the Caspian Sea began in 1922 when systematic observations from coastal and island hydro meteorological stations were organized. Regular observations of the ice using aircraft began in 1930. The main problem was to obtain information about ice conditions in individual areas of the sea. It became possible to obtain a general pattern of the distribution and the drift of ice in the sea initially with the establishment of ice radio watches (1953) and later (1957) from long-term automatic radio-meteorological stations (Klevtsova, 1960). The application of space research methods provided new possibilities for studying the state and the dynamics of sea ice cover. In 1980 images of the ice cover of the northern Caspian Sea obtained in the period January-April by the Meteor 2 satellite were processed and used as the basis for ice charts. In order to interpret and decode these properly, synchronous and quasi-synchronous aerial surveys were carried out. Study of the ice charts obtained from Meteor satellite images and enhanced aerial survey data gives a detailed picture of the dynamics of ice condition in the vast waters of the northern Caspian Sea. Meteor satellite data may be used to inform national economic organizations about all changes in the position of the edges of the ice, the boundaries of shore ice, close-pack ice and open water in between aerial surveys (except on cloudy days), (Bukharitsyn,1983). During the recent years arsenal of researchers was amplified by new equipment with higher time and spatial resolution (radar-tracking, laser, infra-red, microwave and etc). The experience of research and mapping of the sea ice cover reveals the advantages of using satellite images rather than old style methods.

5 ALGORITHEM DEVELOPMENT AND RESULTS

Ice detection approach consists of daily cloud-clear image compositing and pixel-by-pixel image classification using spectral criteria. Additional testing should also be performed for pixels having similar spectral reflectance as ice. Such testing ensures a temporal stability of reflectance and temperature since a large variation of the scene temperature and/or reflectance is indicative of cloud contamination to end with a final testing of classification results against ice cover climatology and sea surface temperature. In order to accurately classify cloud and ice in the image classification algorithm, satellite observations in all available spectral channels in the visible, near infrared, shortwave, middle and far-infrared spectral bands were used. Additionally, derived daily ice maps have been composited on a multi day basis to derive a daily ice map with reduced cloud contamination. The developed ice detection and mapping algorithm have been applied to MSG SEVIRI data and have been tested over the Caspian Sea. Evaluation of the accuracy of mapping ice cover with MSG have been assessed using interactive maps of snow and ice cover produced within NOAA Interactive Multisensor Snow and Ice Mapping System (IMS). In the following, the spectral capability of each channel will be examined to explore their potentials in ice mapping. One of the primary steps of the work is the projection of the obtained images which is done in MATLAB. Figure 5 shows the image of the study area (Caspian Sea) in visible and near infrared. This Figure notes of some of the characteristics of HRV, VIS 0.6, VIS 0.8 and NIR 1.6 channels. One of them is different backscattering of cloud, ice and water in these bands.

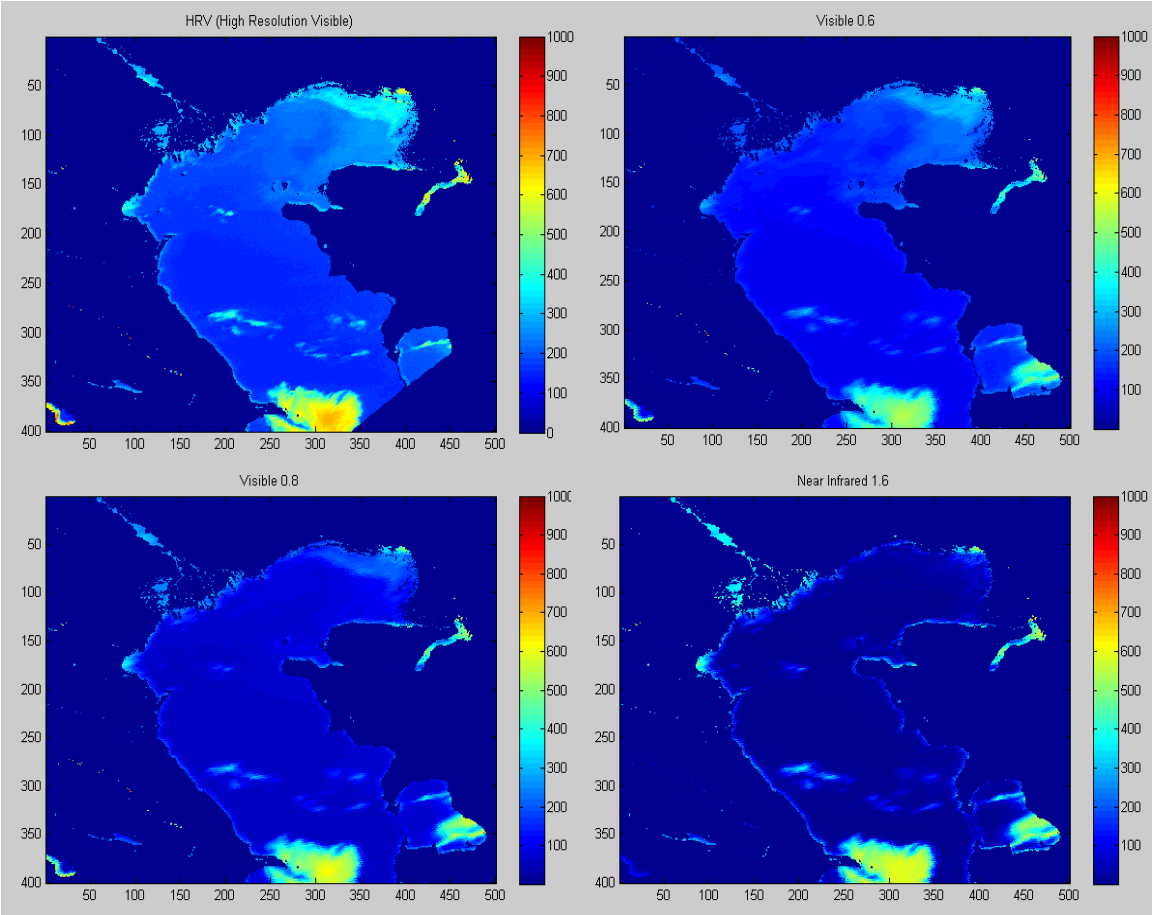


Figure 5: Reflectance of the visible and near infrared channels

This can be further explored in Figure 6 which shows the hourly variation of reflectance for one ice and three water pixels in SEVIRI optical channels: HRV (High Resolution Visible: 0.6-0.9 µm), R01 (0.6 µm), R02 (0.8 µm), and R03 (1.6 µm). The water pixels were selected at three different latitudes to illustrate the effect of viewing geometry on reflectance. This data were collected on January 23rd, 2007 under an exceptional clear sky condition. The x axis represents the number of daylight views (with 30 minutes intervals) and the y axis represents the reflectance in percentage. These graphs detect a presence of clouds over one water-pixel at the end of the day. The R02 channel shows the lowest spatial variability of water reflectance and the best separablity between water and ice but there are some types of clouds and even low level moisture which can increase the reflection and could easily be

confused with ice. On other hand, the near-infrared channel (R03) gives the best separability of cloud and ice/water but inadequate discrimination between water and ice. This channel also shows the lowest sensitivity of water pixels to latitude, viewing geometry and water properties.

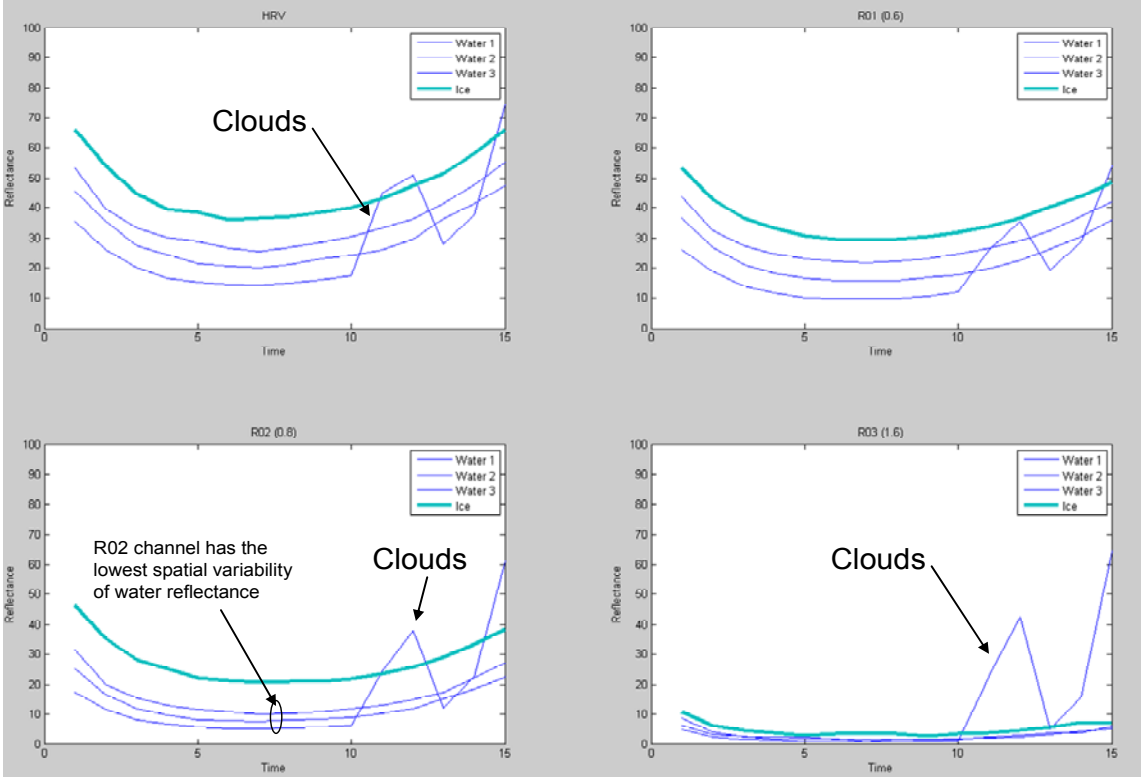


Figure 6: Hourly variation of reflectance for ice, water and clouds

The other noticeable fact about Figure 6 is the effect of viewing angle geometry which creates the curve shape to the reflectance versus time graph with the highest reflectance at early morning (sunrise) and late afternoon (sunset) and the lowest reflectance at mid day (noon). This effect has to be further investigated for construction of the BRDF model.

5.1 BRDF Model Data Collection

Metosat SEVERI data of the winter 2007 was used to form two sample collections of cloud-free pixels. This process was performed manually by comparison of the images in different channels and movement of the clouds in RGB images. In this process the collected MSG SEVIRI data was process in MATLAB and the images were made and for the RGB application all the images had to be compiled in order to make the movies. Each pixel was looked at HRV, R01, R02, R03 and RGB movies to be selected as cloud free. The result of this laborious and lengthy work was two sample collections of 12000 cloud free water pixels and 5000 cloud free ice pixels which covered each and every possible time and longitude and latitude of the study area. The collected samples have been studied for the effect of the solar and satellite viewing angle which are significant factors in geostationary satellites. Figure 7 shows the even distribution of the selected pixels of the two sample collections over the study area (Caspian Sea) which has formed the actual shape of Caspian Sea. As it can be seen in the figure, ice forms mostly in northern parts of Caspian Sea.

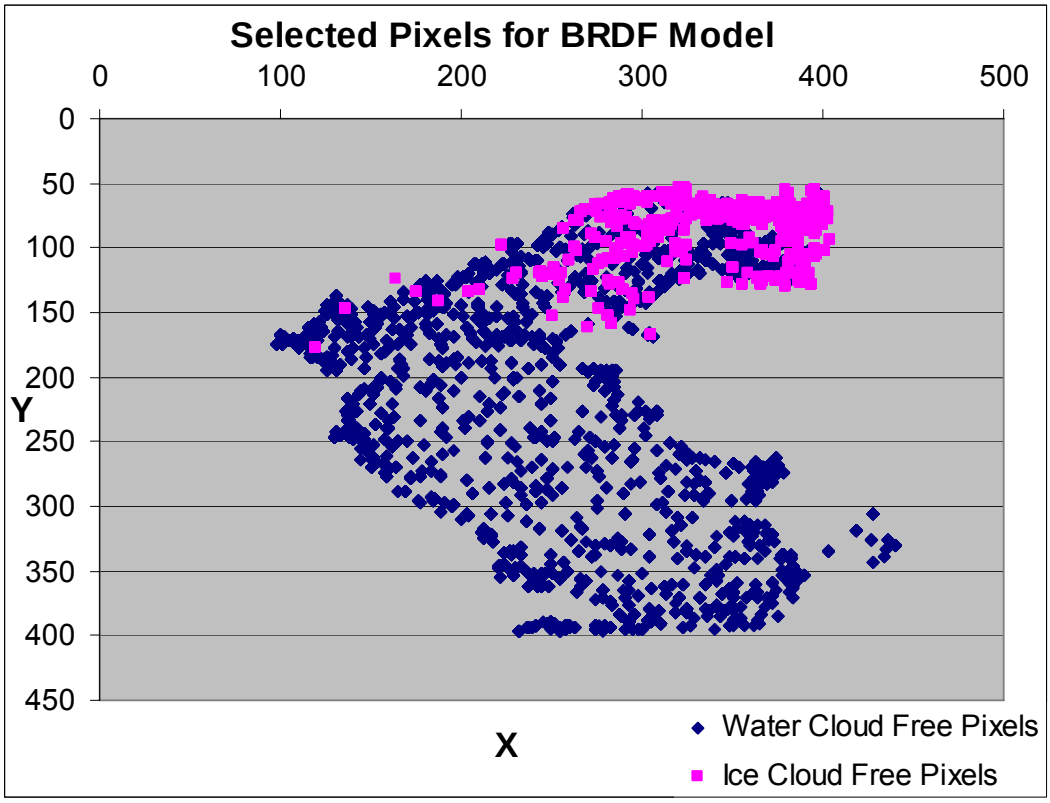
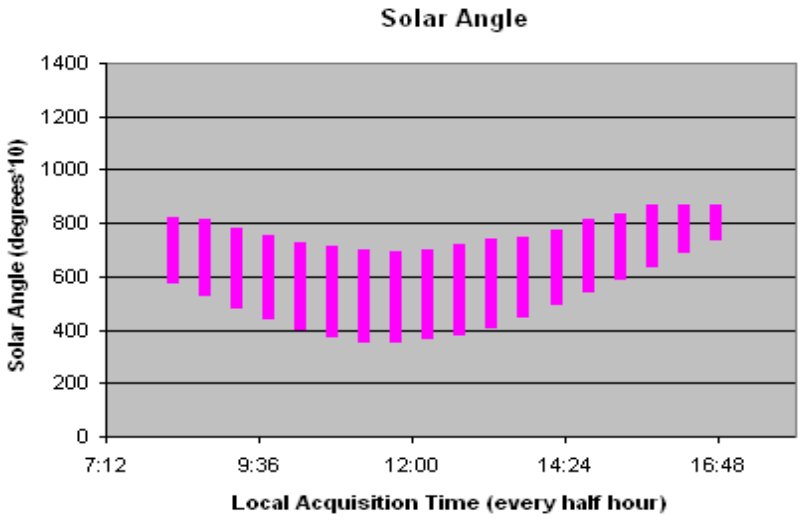


Figure 7: Selected pixels for BRDF model

In the next step the collected data has been tested for detection of any relation between the location of the pixels, time of the day and even the seasonal changes with the changes in the satellite viewing angle and the solar zenith angle. The results have been investigated for possible relationship between the angles and the reflection in each channel. Figure 8 shows the daily variation of solar zenith angle and Figure 9 shows the seasonal changes of the solar angle. Figure 9 starts at the first day of the year (Julian year) and as spring approaches longer days are observed which means more daily observations and more daily variation of the angle.



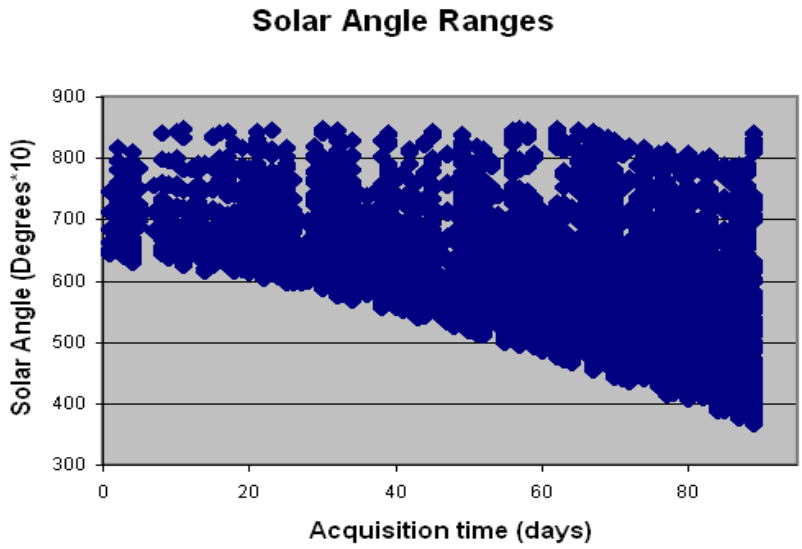
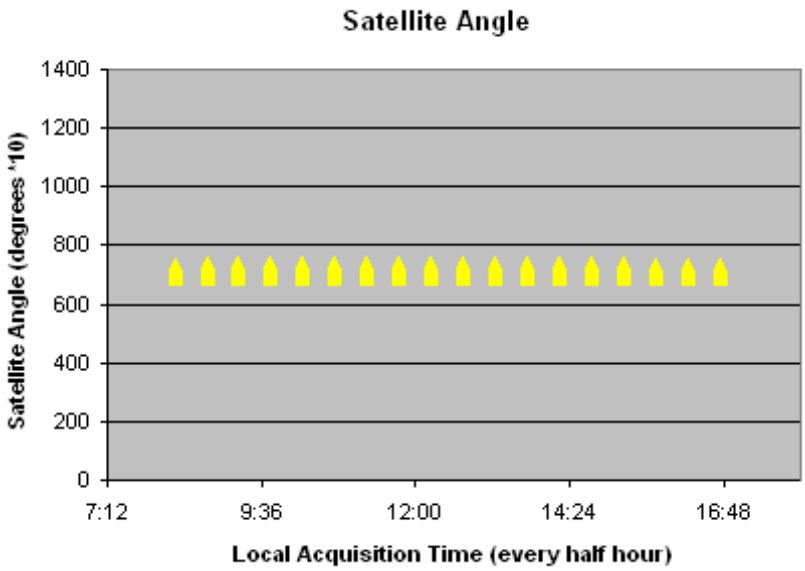


Figure 8 Top- Daily change in SOL, Figure 9 Bottom- Monthly variation of SOL

As we expected, satellite angle does not vary neither daily nor seasonally for a given pixel, because of the fact that MSG SEVIRI is a geostationary satellite, as it could be seen in Figure 10 and Figure 11. However, we do have a short range of variation of the satellite viewing angle because it is dependent on pixel's location. Given the satellite location, this is over the equator, satellite viewing angle increases with latitude. This has been taken into account while developing the BRDF and classification model.



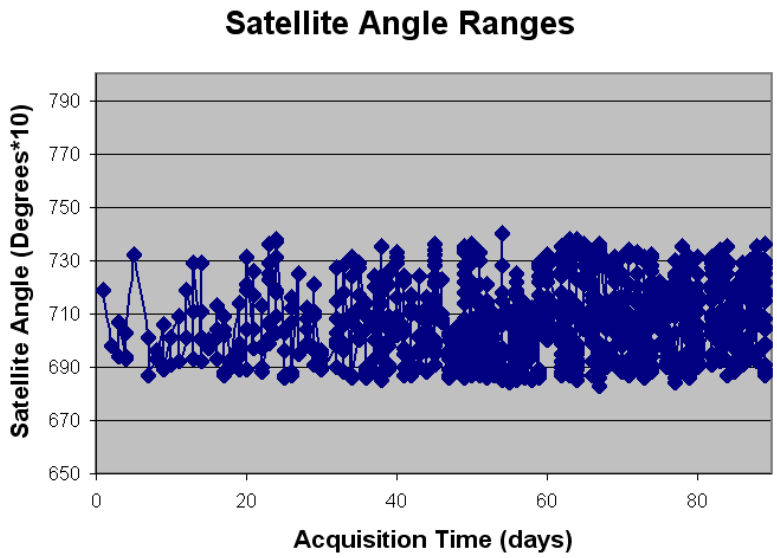


Figure 10 Top- Daily change in SAL, Figure 11 Bottom- Monthly variation of SAL

The next parameter that could be obtain from our data set is the solar- satellite azimuth angle or as commonly called relative azimuth angle which is defined as the absolute difference between the satellite azimuth angle and the solar azimuth angle. The relative azimuth angle is highly dependent on the variation of the solar and satellite angle and since in our case satellite angle for each pixel is constant, the variation of this parameter is caused by the change of the solar angle.

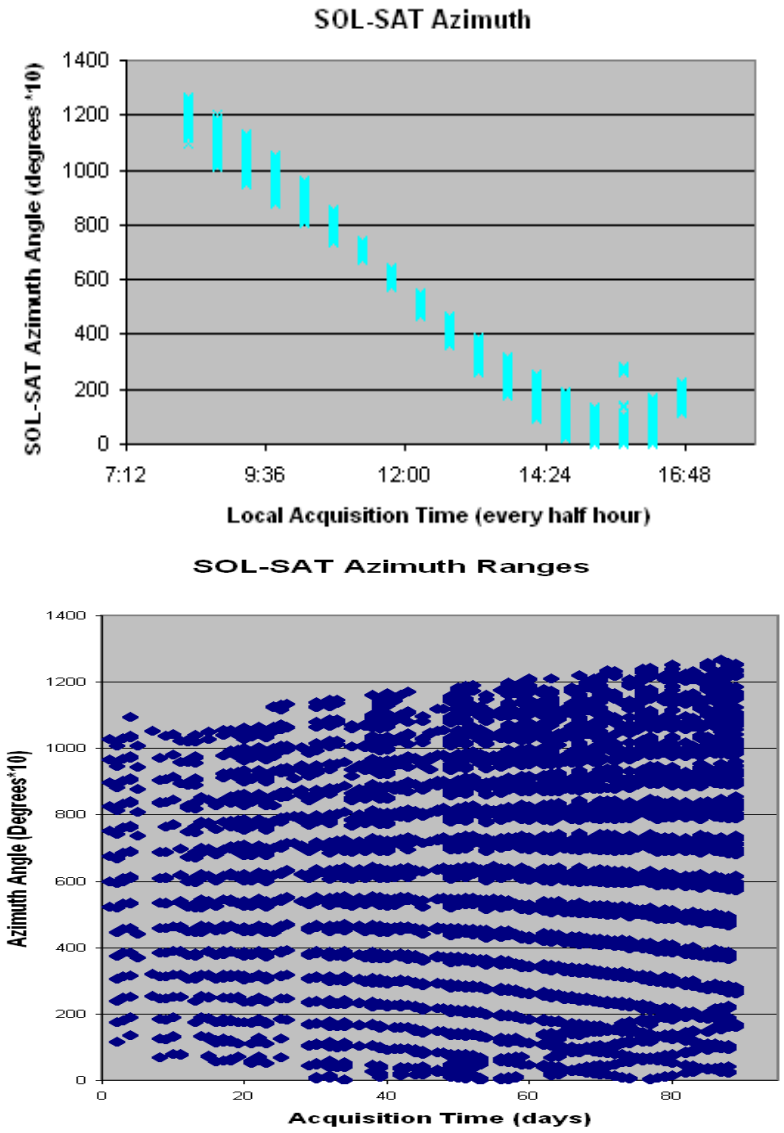


Figure 12, Top-Daily change in RAZ, Figure 13 Bottom-Monthly changes of RAZ

After studying the angle variations it is time to look at the channels data and try to understand and detect any possible patterns and correlations. Figure 14 shows the reflectance variation in 4 spectral bands of MSG SEVIRI. As it can be seen the High Resolution Visible (HRV) band (0.6-0.9) and R01 have the highest response to solar angle variations. The reflection response of R02 band (0.8) is rather moderate. In the Near Infrared (NIR) band

(1.6) there is almost no response to the variation of solar angle which makes it a rather stable band for the classification process.

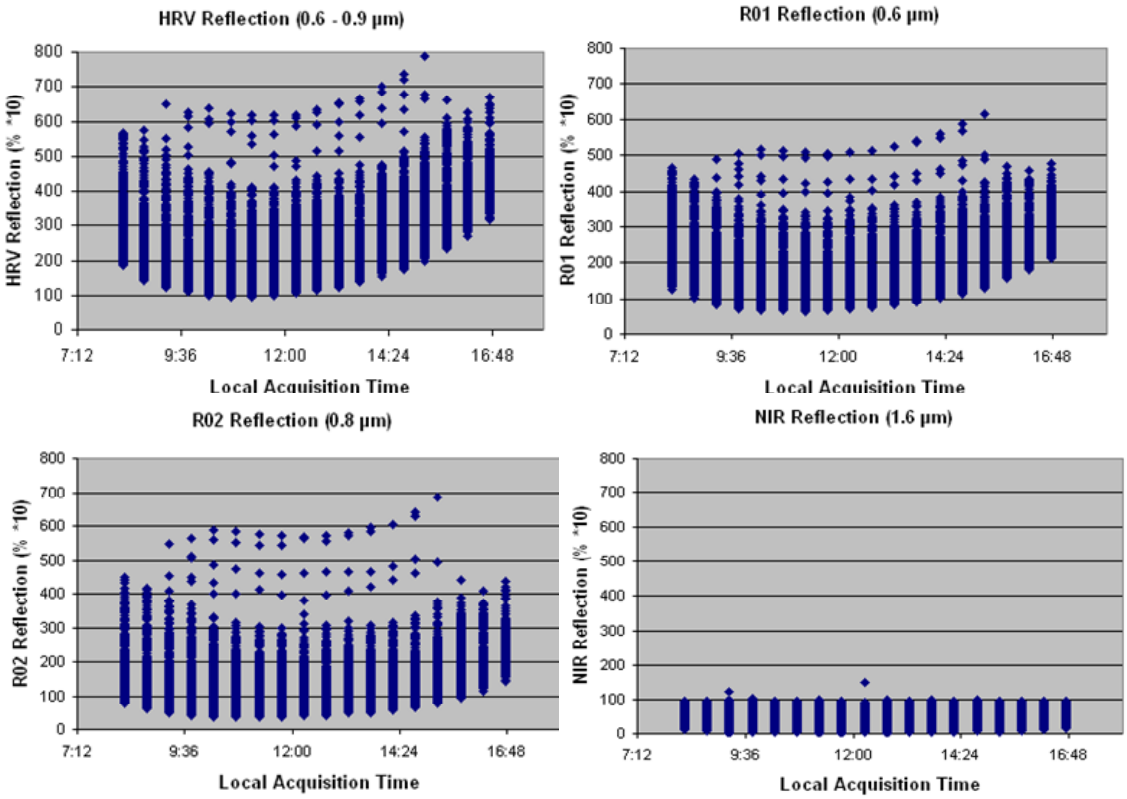


Figure 14: Top left- Daily variation of HRV reflection, Top right- Daily variation of R01, Bottom left- Daily variation of R02, Bottom Right- Daily variation of NIR

The study of water pixels reflectance in this channel will enhance any possible patterns. This phenomenon can be further understood when all the reflectance are being projected together. Figure 15 illustrates the daily variation of solar, satellite and azimuth angel along with the reflectance of water pixel at the same acquisition time. It also depicts the sensitivity of each channel to the angular changes during the day.

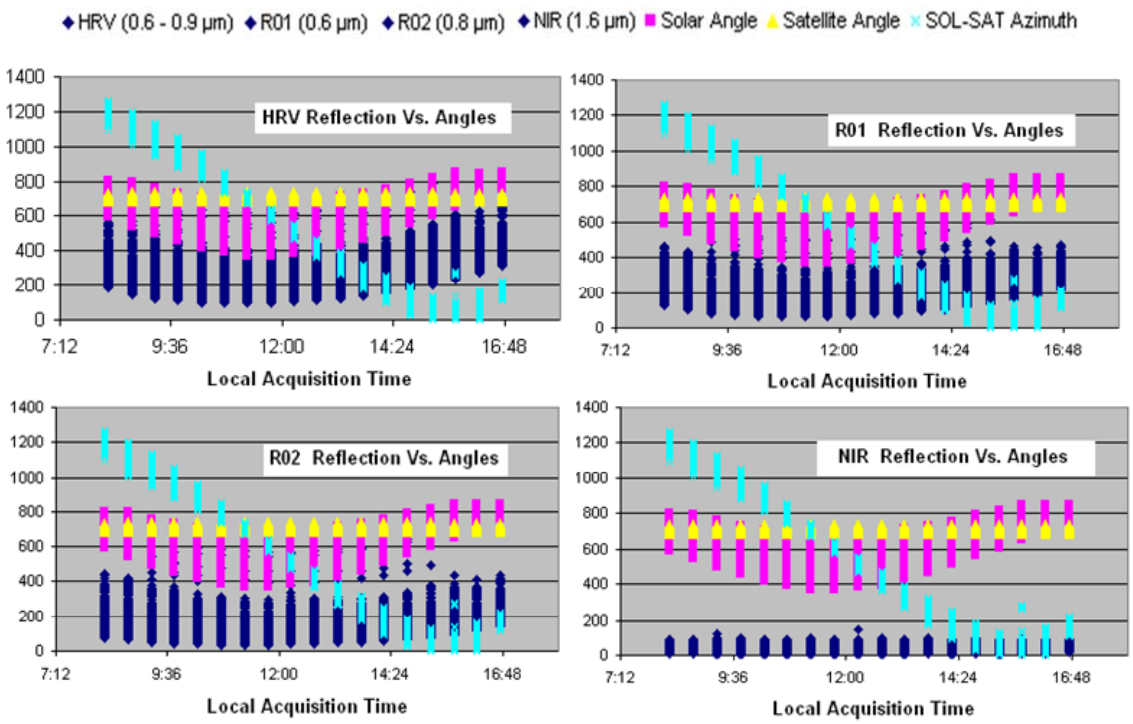


Figure 15: Effect of Sun Irradiance and Sensor Viewing Angles on the Reflected Radiance

In order to analyze the data, Stepwise Multiple Linear Regression (SMLR) method was used to search for possible correlations. In this method, 50 possible geometric combinations of angles were formed and used. Table 4 is presenting some of the angles combinations for creation and formulation of BRDF model. As it can be seen in this Table some of the geometric values of the angle show a significant correlation to the observed reflection values which are detected through the stepwise model.

Table 4 Sample of selected coefficients (b_j) for HRV models

Set	B [a]	C [b]	D [c]	Cos B	Cos C	Cos D	Sin B	Sin C	Sin D	1/CosB	1/CosC	1/Cos D
HRV-1	4071.44	0	59557.66	3593.02	81039.14	38416.11	-1414.41	0	-5436.97	0	0	0
HRV-2	3114.09	12580993	0	2823.13	10945638	19048.18	-1056.49	0	0	0	0	9.41
HRV-3	1819.51	12925051	23816.74	0	11232847	30688.29	0	0	0	0	0	9131.83
HRV-4	-339.18	61906845	0	2232.47	53688659	141509	-790.92	0	0	0	0	0
HRV-5	-24494.9	-1.2E+09	-144551	3443.57	-1E+09	-33153.7	-1298.11	0	-10603.1	0	0	0
HRV-6	2393.51	9714654	-234750	2671.11	8746670	97613.75	-820.50	0	-20520.9	0	0	0
HRV-7	5894.94	-5091239	-267728	5062.12	-3931639	101381.6	-3484.21	0	-20849.3	0	0	0
HRV-8	2573.59	195423.3	-245670	3018.73	530922.7	101816.2	-956.62	0	-21121	0	0	0

1/Sin B	1/Sin C	1/Sin D	B*C	B*D	C*D	B/C	B/D	C/D	C/B	D/B	D/C	1/B
0	0	0	0	-2084.04	0	0	-444.27	2943.14	0	0	-45392.6	0
0	0	15989.7	0	-1630.39	4410.19	0	0	-10335.5	0	0	15854.37	0
0	0	21642.85	0	-1240.78	0	0	246.57	-6671.38	0	0	6262.07	0
0	0	0	0	0	68664.99	0	0	6692.76	-2.03	2.15	19534.24	0
0	0	0	13323.44	-1894.22	67171.8	15797.63	-281.57	0	0	0	48280.56	0
0	0	-34106.6	0	-1622.02	203611.3	0	0	32387.48	0	0	60719.19	0

1/C	1/D	B/ CosB	B/ CosC	B/ Cos D	C/ Cos B	C/ CosC	C/ CosD	D/ CosB	D/ CosC	D/ CosD	CosB * CosC	CosB * CosD	CosC * CosD
0	0	0	354.92	25.69	0	0	0	0	1782.62	0	-4096.9	2214.95	-48327.34
3406224.47	0	0	297.65	0	0	0	0	0	0	0	-2675.05	2736.74	0
3519910.24	0	0	0	-17.31	0	0	0	0	0	0	0	2186.34	0
18369755.36	0	0	376.59	0	0	0	0	0	0	0	-4276.97	0	-56477.79
-301834917.1	0	0	0	12.38	0	0	0	0	0	0	-3817.66	2226.8	-39508.40
2650639.39	0	0	366.25	0	0	0	0	0	0	0	-4332.81	2294.06	-159227.03

CosB*CosC *CosD	Sin2B	Sin2C	Sin2D	Cos2B	Cos2C	Cos2D	1/ Sin2B	1/Sin 2C	1/Sin2D	1/ Cos2B	1/ Cos2C	1/ Cos2D
0	-378.42	0	0	-350.62	0	0	0	0	0	0	0	0
-1371.68	-391.45	0	0	-261.47	0	0	0	0	0	0	0	0
0	-201.44	0	0	19.31	0	0	0	0	-18287.7	0	0	0
0	0	0	-13628.5	-149.00	-588142	0	0	0	0	0	0	0
0	-389.85	0	9454.46	-322.72	-1.1E+07	0	0	908990.8	0	0	0	0
0	-295.87	0	0	-187.05	0	0	0	0	0	0	0	0
0	-345.87	0	0	-631.26	-752951	0	0	36135.18	0	0	0	0

[a] B stands for azimuth angle [b] C stands for satellite zenith angle [c] D stands for solar zenith angle

The stepwise model (Figure 16) determines the significance of the parameters to the value of the variable being predicted. Any contribution of the parameter has been recorded and that

parameter has been selected from the pool for the model. These significant parameters have deployed in the formation of the model.

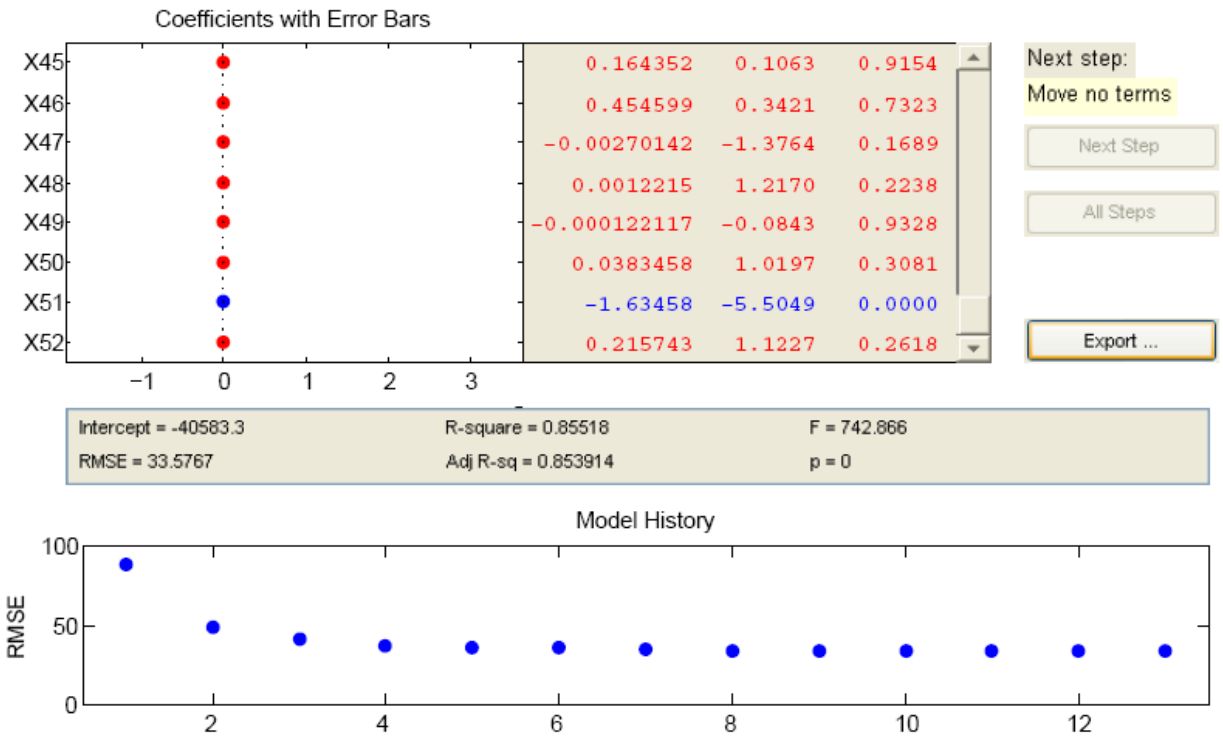


Figure 16: Stepwise model processing system

Two types of split sample validation were applied (Figure 17) for the estimated models: (i) split sample validation with a test set size of one third of the training set, and (ii) split sample validation with a test set size of the remaining samples that were not used for model calibration. In the latter method, the validation data were divided into testing sets of 1000, 2000, 3000, and 4000 sample sizes to observe the statistical indicators (RMSE and R^2) that changes with the altering validation set size for an individual calibration model. Figure 17 shows the pixel selection, validation and testing process.

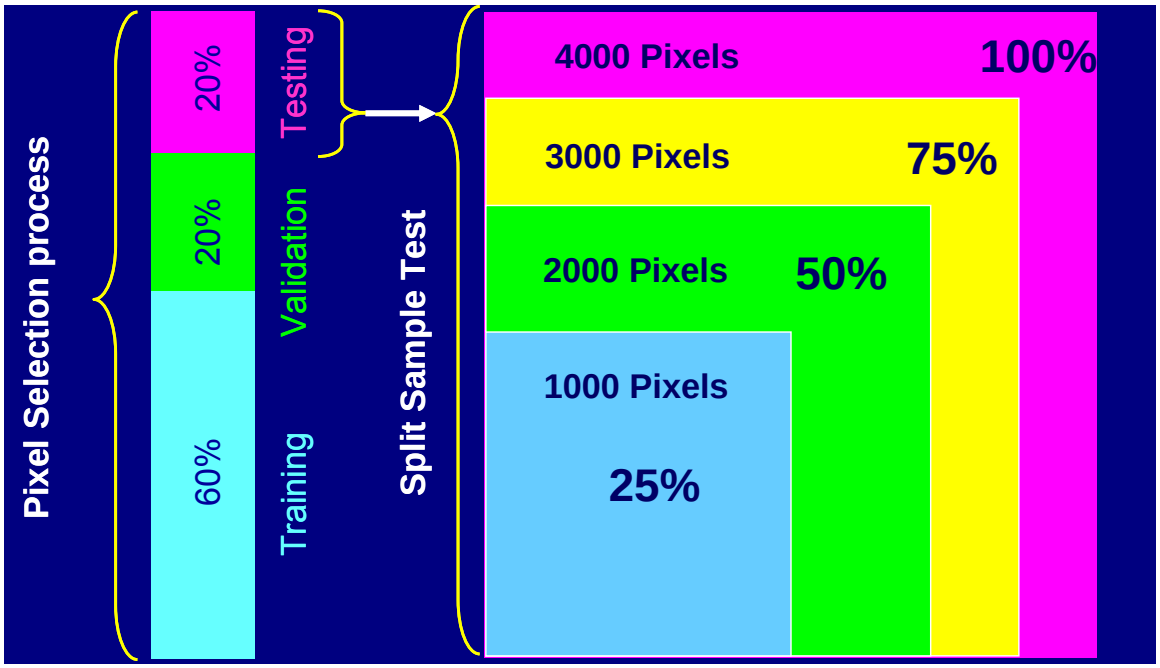


Figure 17: Pixel Selection and Validation and testing process

Both validation methods were applied on eight training data sets (HRV-1 to HRV-8). A postulate of using completely different and independent test data for validation and calibration was met (Duckworth, 1998; Snee, 1977).

For this process the data was first randomized and five sample sizes of 3000, 3000, 3000, 5000 and 8000 were selected and tested for each channel to derive the coefficient values and check the consistency of the model in each trial. The stepwise model has been used in order to check the correlation between the reflection values of that channel with the angles' combinations. All the statistic values were recorded in each round and the test was performed for other sample sizes. The combinations with highest coefficient values were selected and formed a BRDF model.

Table 5 Statistical results of the Stepwise Multiple Linear Regression calibration for R01

Set	Training Size	Train. Range	Test Size[a] (th)	Test Range (th)	Train. RMSE	Test RMSE (th)	Train. R^2	Test R^2 (th)
R01-1	3000	2-3001	1000	4002-5001	4.07649	13.62153	0.7030	-2.1124
R01-2	3000	3002-6001	1000	1002-2001	4.23452	6.24764	0.7046	0.3475
R01-3	3000	6002-9001	1000	9002-10001	4.12608	245.92226	0.7083	-1045.1004
R01-4	3000	8790-11789	1000	7002-8001	3.99532	36.48907	0.7126	-20.0941
R01-5	5000	2-5001	1666	9002-10667	4.06957	45.97084	0.7150	-37.0779
R01-6	5000	5002-10001	1666	2002-3667	4.05875	458.26447	0.7170	-3622.5493
R01-7	8000	2-8001	2666	8002-10667	4.10478	3.89606	0.7131	0.7231
R01-8	8000	3790-11789	2666	2-2667	4.05925	3.97338	0.7155	0.7200

Regression analysis represents the best results when: (i) a square multiple regressions coefficient is close to 1 and/or -1 and (ii) the root mean square error (RMSE) reaches the zero value. The square multiple regression coefficients and the root mean square error (RMSE) are given by: $R^2 = 1 - [\sum (y_j - y_{j,comp})^2 / \sum (y_j - y_{mean})^2]$

$$RMSE = \sqrt{S^2} = \sqrt{SSR / (N-p)} = \sqrt{\sum (y_j - y_{j,comp})^2 / (N - p)}$$

The new BRDF model was tested with the same procedure using the same samples for validation of the new model. The new model was used to simulate the reflection in each channel the statistic values was compared. Simulated values were plotted against observed values presented in Figure 18 which indicates that, there is a correlation of %76 which is satisfactory at this point.

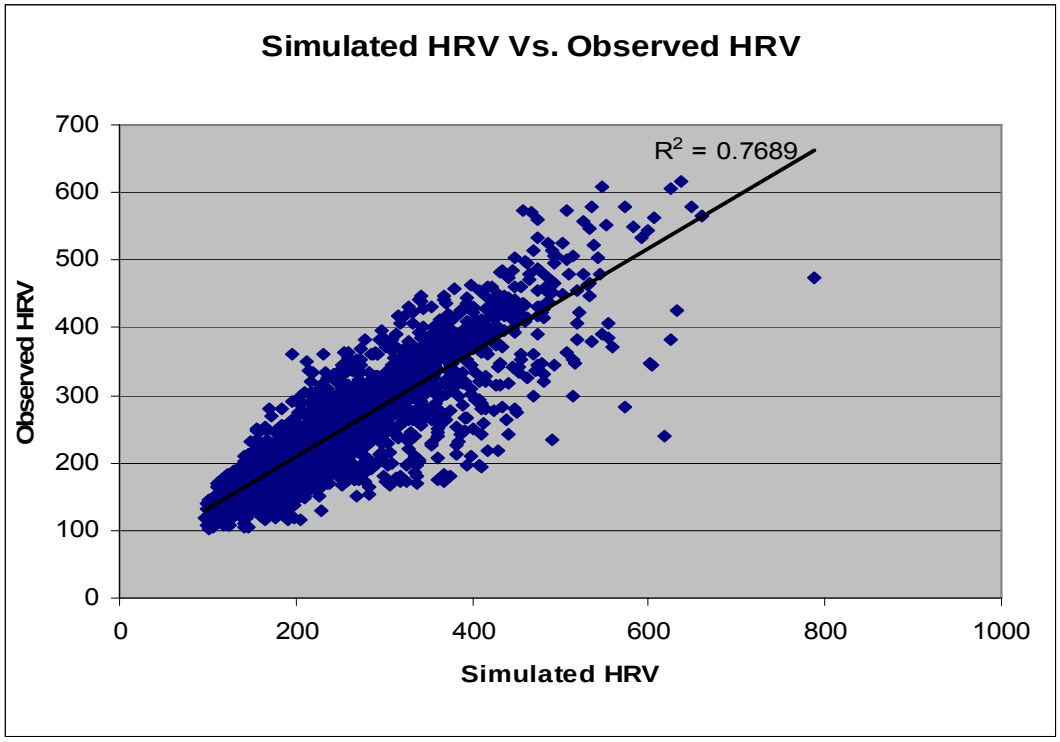


Figure 18: Simulated vs. Observed HRV reflection values

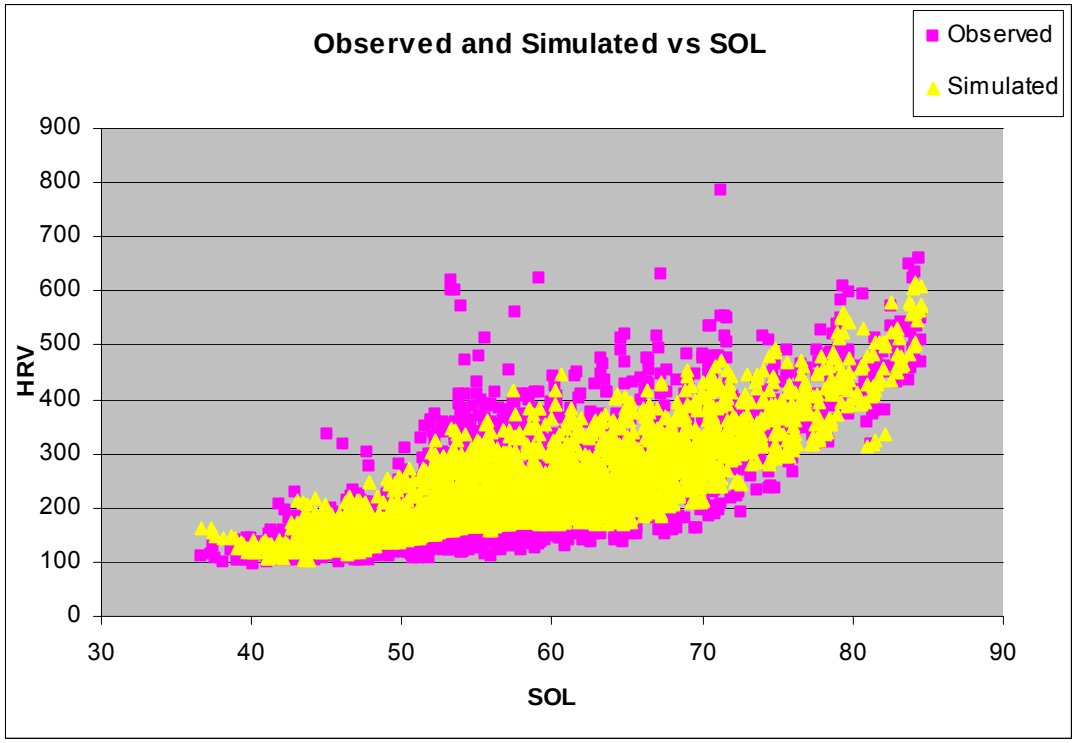


Figure 19: Simulated and Observed Values of HRV vs. Solar Zenith Angles

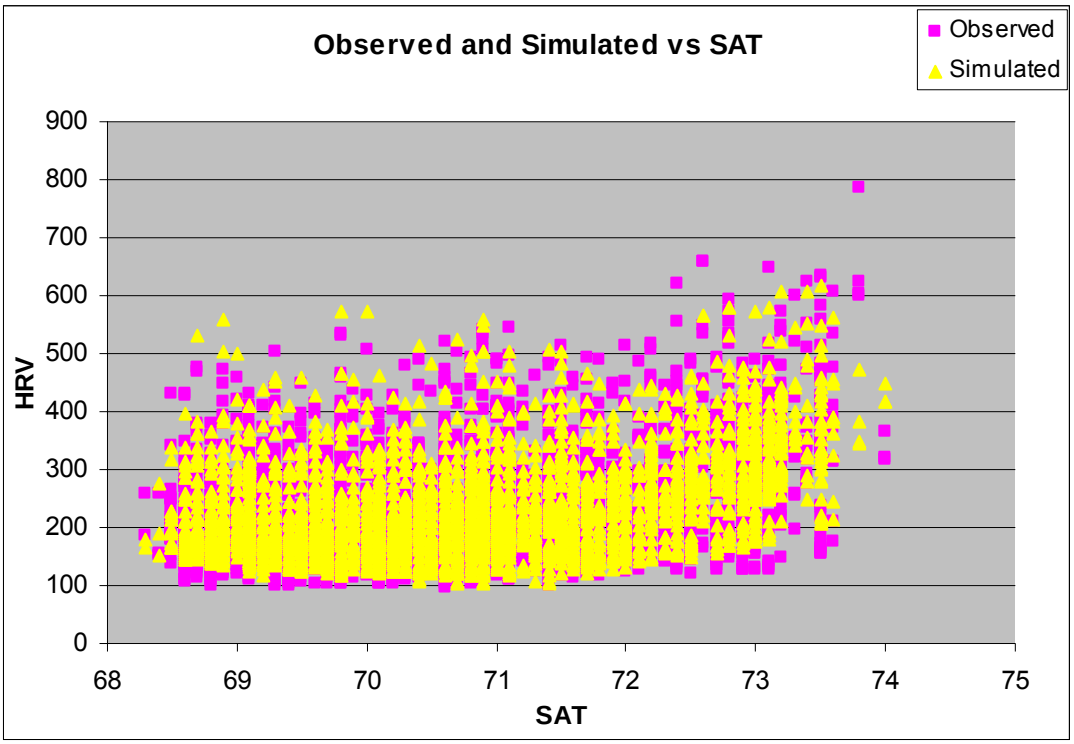


Figure 20: Simulated and Observed Values of HRV vs. Satellite Zenith Angles

Figure 18,Figure 19 andFigure 20 are presenting the actual point data and show the correlations between simulated and observed values in High Resolution Visible (HRV) and the effects of the angels.

After running the Stepwise Multiple Linear Regression (SMLR) with the selected angle combinations to be included in the linear model, split sample validation with test set sizes have been examined in order to find the best possible model. In this method one third of the data was chosen to serve as the calibration and training data and the rest of the data was used for the validation process. As it can be observed from the results in Table 6, HRV-7 model has the lowest root mean square error (4.6412 %) than the other test sets. While the validation test presents the HRV-7 model as the best predictor for reflectance values, the calibration test shows the HRV-7 model as the sixth best fit out of eight. Therefore, the

HRV-1 model is considered as the most reliable as it has the best calibration results as well as acceptable validation results (RMSE of 5.0889 %). Training sets HRV-3, HRV-4 and HRV-5 with extremely high values of validation RMSE do not predict the reflectance values well enough and can not be used for BRDF model.

Table 6 Results of split-sample validation of the HRV models

Training Set	Training Range	Training Size	Test Range	Test Size [a]	RMSE (th)
HRV-1(th)	2-3001	3000	4002-5001	1000	50.889
HRV-2(th)	3002-6001	3000	1002-2001	1000	50.515
HRV-3(th)	6002-9001	3000	9002-10001	1000	4850.193
HRV-4(th)	8787-11786	3000	7002-8001	1000	4589.018
HRV-5(th)	2-5001	5000	9002-10667	1666	24334.975
HRV-6(th)	5002-10001	5000	2002-3667	1666	50.449
HRV-7(th)	2-8001	8000	8002-10667	2666	46.412
HRV-8(th)	3787-11786	8000	2-2667	2666	46.461

[a] Test size equals to one third of the training size

Table 7 Results of split-sample validations of HRV models

Training Set	Training Range	Training Size	Test Range	Test Size[b]	Average RMSEv
HRV-1	2-3001	3000	3002-11001	8 x 1000	48.209
HRV-2	3002-6001	3000	2-3001, 6002-11001	8 x 1000	50.395
HRV-3	6002-9001	3000	2-6001, 9002-11001	8 x 1000	4844.328
HRV-4	8787-11786	3000	2-8001	8 x 1000	4587.565
HRV-5	2-5001	5000	5002-11001	6 x 1000	24534.279
HRV-6	5002-10001	5000	2-5001, 10002-11001	6 x 1000	51.263
HRV-7	2-8001	8000	8002-11002	3 x 1000	46.815
HRV-8	3787-11786	8000	2-3001	3 x 1000	46.464

[b] Test Size varies according to the training size. Testing was processed on each 1000 samples that were not used for training set.

The results from split sample validation with test set sizes of one third of the training sets are confirmed by the results from split-sample validations with test set sizes of the remaining samples (Table 7). The comparison of the results of the two split sample validation shows the significance of the test sample size. The larger the test sample size for validation, the lower the root mean square error. This rule does not apply on individual test sets because the

RMSE value also depends on the training samples that influence greatly the value of the statistical indicators. Figure 21 illustrates the distribution of the estimated reflectance against the measured HRV. The R^2 and RMSE indicate better fit in figure 20 due to larger size of the test set. Benefits of split-sample validation with test set sizes of the remaining samples lies in its capability of providing comparison of the validation tests.

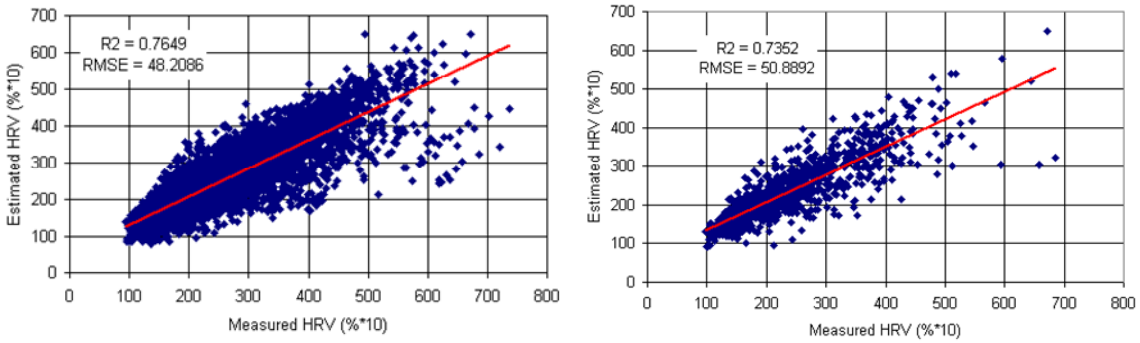


Figure 21-Left- HRV-1 model, training results, Right- HRV-1 model, Validation results

The result of HRV-1 model, the model with the best calibration and validation results, was selected to simulate the distribution of the water reflectance in HRV channel over the Caspian Sea (Figure 22). The equation used for the value prediction and the simulation model are available in the Appendix. The simulated values of HRV are close to the trend and magnitude of the reflectance obtained from the satellite sensor (Figure 23) and therefore can be applied for the prediction of water BRDF model for HRV. The advantage of the model to the real data is that in the presence of cloud, where satellite would not be able to function, our model still can generate data and in addition to that, can be used as a cloud detection tool. As Figure 23 shows, the model follows the normal daily trend of the satellite observation and where clouds start appearing in the scene the reflectance value of the satellite data changes, as can been seen as spike in the figure, but the model follows the normal pattern without being effected by cloud. This capability of the model in cloud detection is going to be used in the future the classification model. As it can be seen in

figures 22 and 23, the stepwise method simulates the reflectance value of the channels. The figure 23 compares the observed and simulated HRV. The simulated curve has a normal trend of HRV reflectance values but the observed values show a different trend. The lower left graph in Figure 23 shows a spike in the reflectance on acquisition time 10 which is indicative of presence of cloud and blockage of the signal at that time by a moving cloud. This feature of the stepwise model makes it a powerful and unique cloud detection device which will be extremely valuable in the process of formation of our classification model. There are separate stepwise models built for all other reflection and near infrared channels that are presented in the appendix.

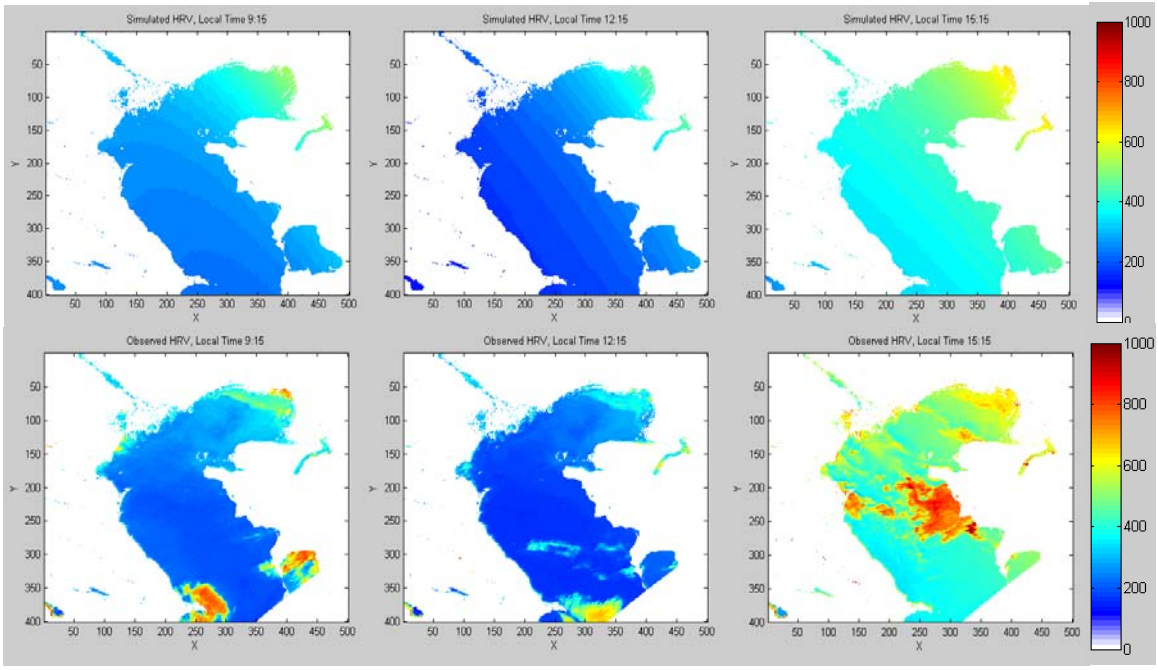


Figure 22: Observe (bottom) and simulated (up) HRV Reflectance (0.6 – 0.9µm), January 23

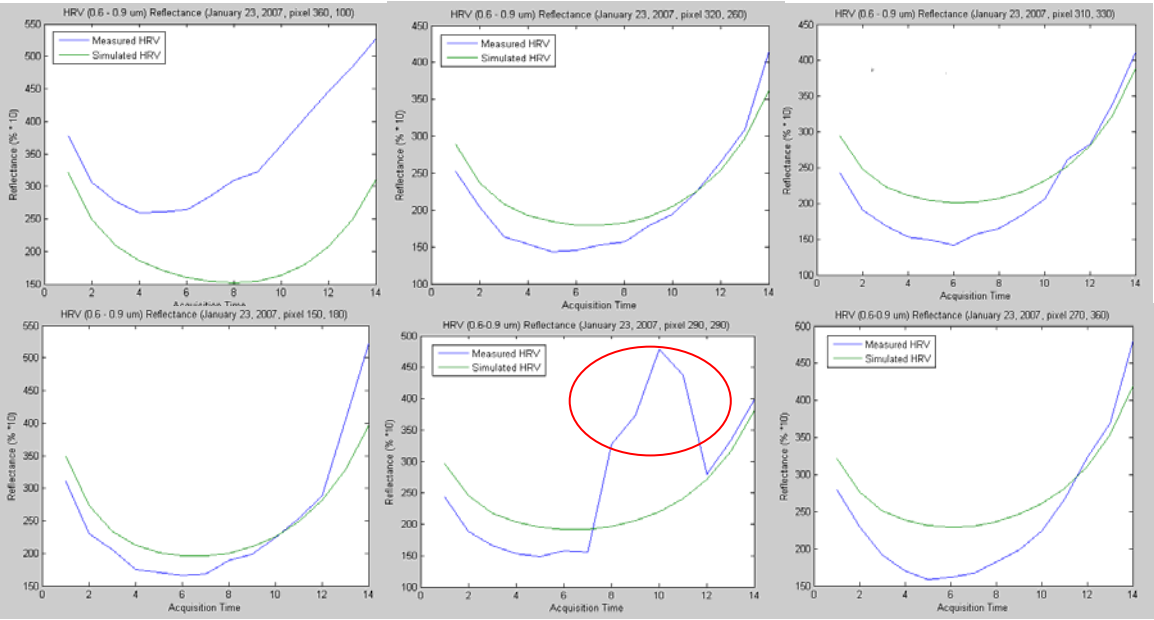


Figure 23: Observed vs. simulated HRV reflectance for pixel [360, 100], [320, 260], [310, 330], [150, 180], [290, 290] and [270, 360] on January 23, 2007

The stepwise model can simulate the reflectance value of any given channel in presence of cloud and even in the case of complete overcast. When compared to the observed images, stepwise model has the ability to highlight the cloud and identify the cloud contaminated pixels. In the pixel scale a spike in the reflectance values is an indicative of a rapid change the presence of cloud and the blockage of the signal at that time by a moving cloud. This feature of the stepwise model makes it a powerful and unique cloud detection tool which will be extremely valuable in any classification model.

Furthermore the stepwise model is a necessary tool in studying and formulating the BRDF model and any classification and mapping models. This model is an effective tool in the representation of the angular behavior of the existing conditions at any time and given location. It is necessary to see the angular variation and understand the high and lows of the reflectance values before setting up any threshold for cloud, water and ice or any other target

features. The other application of stepwise model is in elimination of the confusion in case of unclassified pixels after the first round of classification process. In this case the stepwise model can offer a close range of expected values to set up a new threshold or explain the extreme values by looking at the stepwise generated maps.

This approach will be used to retrieve the simulation equation for all four reflectance channels for both water and ice pixels. The Bidirectional Reflectance Distribution Functions (BRDF) will be included in the correction of normalized difference snow index (NDSI) and defining the Ice Surface Temperature (IST) and Sea Surface Temperature (SST) threshold in order to improve the cloud, water and ice classification.

In addition to the above mentioned methods, the cloud discrimination potentials of the near-infrared channel can also be seen in Figure 24. This figure shows the reflectance of the four optical channels at 11:45 am local time. The other two visible channels (HRV and R01) show a moderate separability between ice and water and higher confusion between ice and clouds. The comparison between the four optical channels show the importance and the distinctiveness of the near-infrared channel (R03) in cloud filtering where a simple static threshold can be used to undoubtedly separate between clouds and ice. In addition this effect can clearly be recognized by studying the characteristics of these channels.

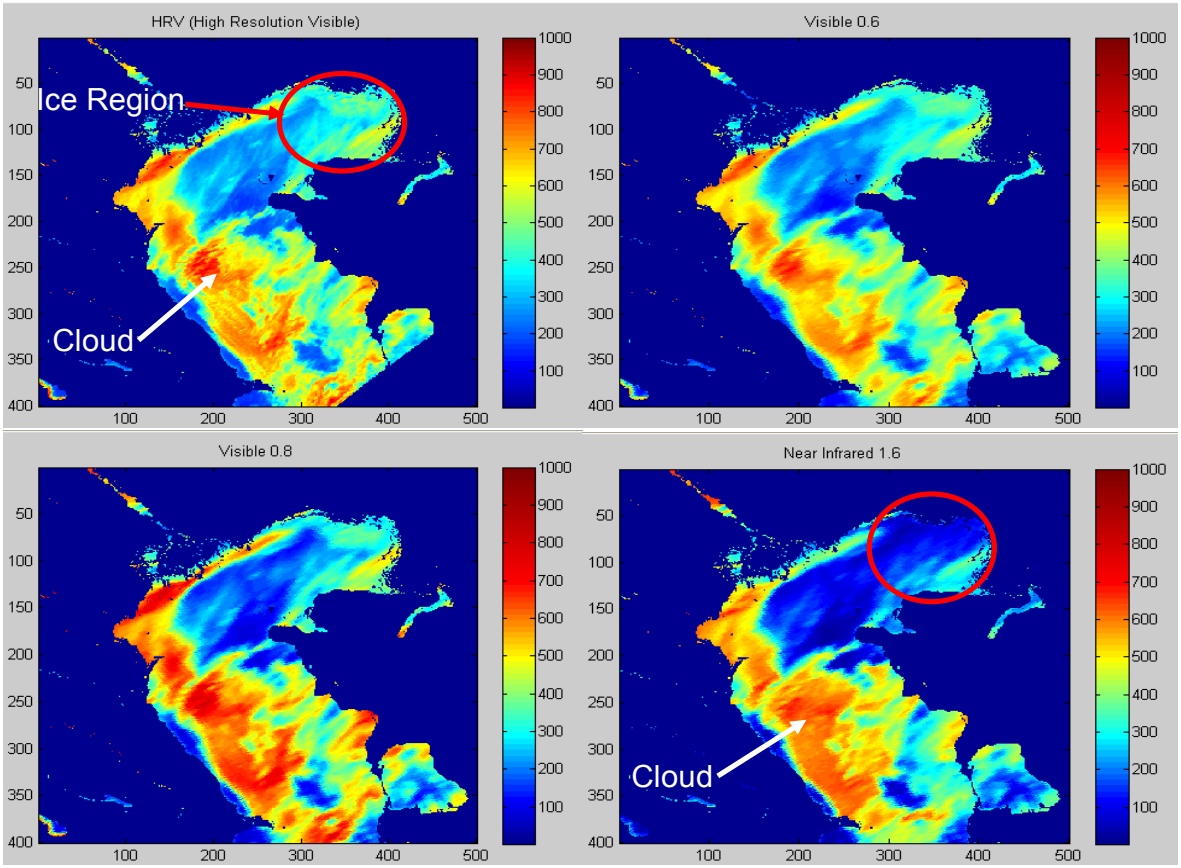


Figure 24: Reflectance of four optical channels; emphasizing the importance of Near Infrared channel in Cloud detection

As Figure 25 shows the characteristics of these channels; cloud and ice have very similar reflection at HRV and 0.6 which makes the process of classification very difficult since the cloud thickness is constantly changing and does not define a consistent reflectance. However, there is a complete deferent case in near infrared channel (1.6) where ice and cloud offer considerably distinguishable reflectance. Ice has less than 15% reflectance versus cloud with more than 75% reflectance in near infrared channel.

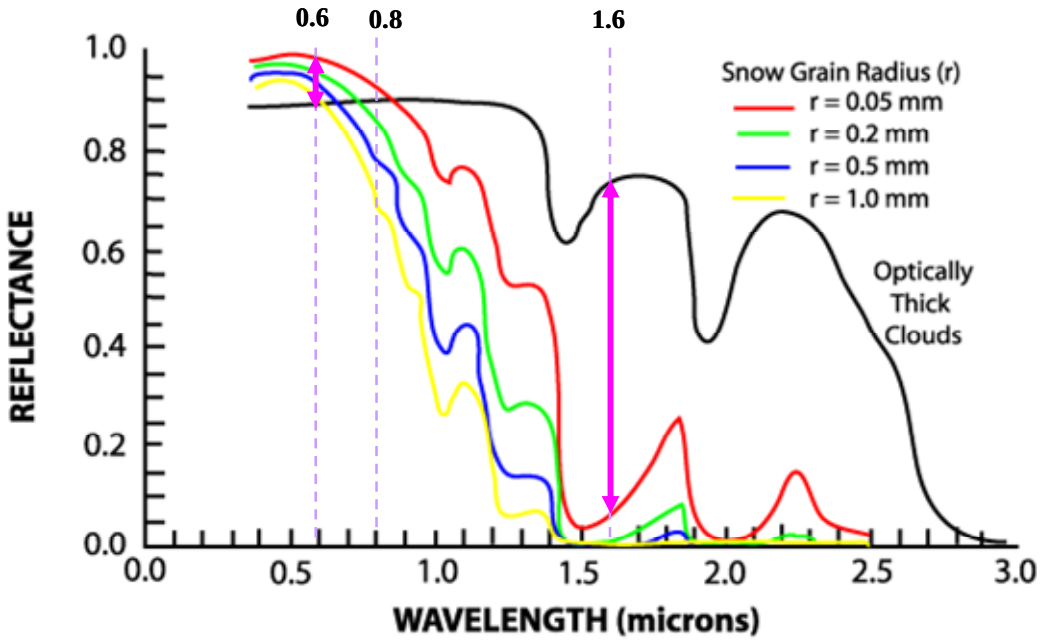


Figure 25: Reflectivity of Ice/snow versus Cloud in 0.6, 0.8 and 1.6 μ

Also a further observation of the absorption imaginary refraction index of light in ice and water shows that ice has relatively higher refraction than water which could be used as another classification parameter if a threshold could be established. Figure 26 indicates different absorption of ice and water clouds for 1.6μm channel which is marked by arrows.

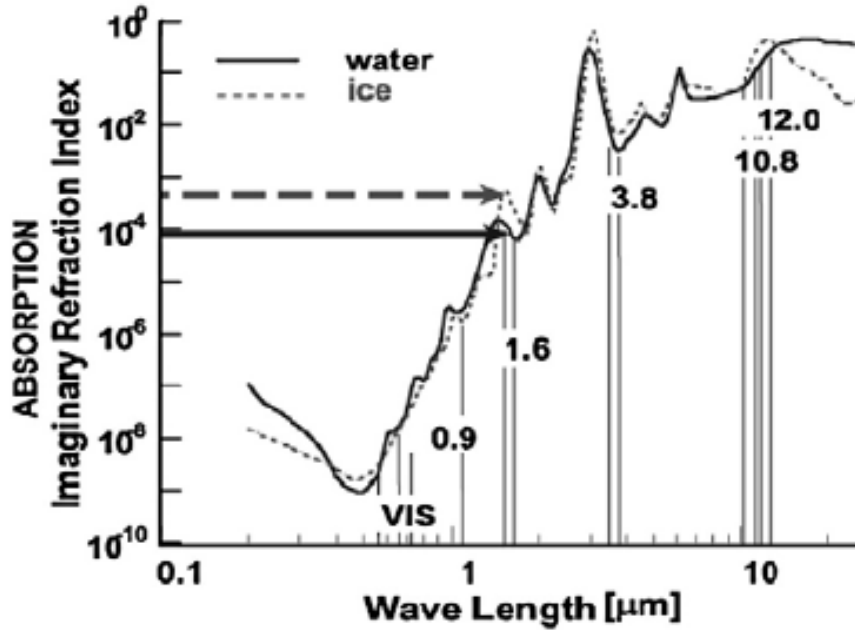


Figure 26: Absorption of ice (dashed) and water (solid) in different spectral regions

Taking all the above elements and factors that we have discovered by studying the data and the electromagnetic properties of these channels with a variety of wavelengths, we are going to develop the next steps which will be finding a correlation and formation of a possible model that could connect and confirm all the factors and parameters that we have so far discovered and eventually shape the classification model.

5.2 Sea Surface Temperature (SST)

The first assessment of Sea Surface Temperature (SST) starts with comparison of 2 IR bands (10.8 and 12.0μm) and a brightness temperature band of 3.9μm to the SST map generated by the proposed algorithm by Romaguera. The Northern Caspian Sea is expected to show low values of SST due to the presence of ice, below 273 degree Kelvin, for the selected time period in February of 2007 which is mid winter for this region. However, the projected SST map based on Romaguera's algorithm indicates the warmer than expected temperature values well above the freezing point which is inconsistent with the actual sea surface temperature for this region at the chosen time. Figure 27 illustrates the inconsistency in SST values, when the existing SST models have been applied to a non-temperate region such as Caspian Sea.

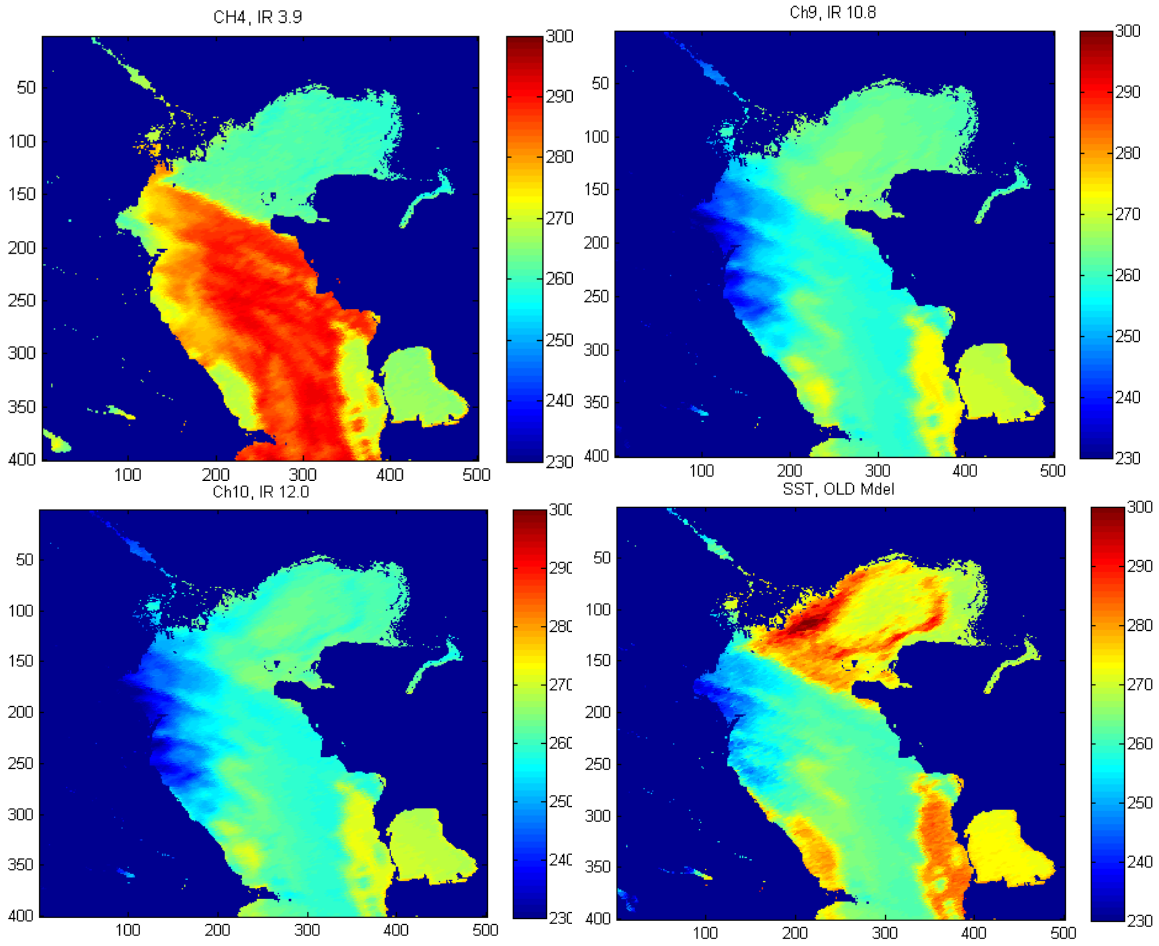


Figure 27: First simulation using the proposed SST model

Based on the projected graphs, Romaguera's SST model does not produce optimal results. Especially given that this area of water is surrounded by ice, high values of SST are almost impossible to observe. Further investigation has shown that Romaguera has developed and tested his algorithm over warm water bodies of the low latitudes of tropics. His research team collected the in situ data from a field campaign over the Mediterranean Sea in May 8th, 2004. Estimation of radiometric temperatures was compared to satellite data compiled from SEVIRI to produce his algorithm. Moreover the studies of his model in different parts of Caspian Sea show how this model behave in high latitudes and in the areas with low water temperature. The following figure illustrates transects of the 2 IR channels and Romaguera's

SST model. As it can be seen, the SST model shows extremely high values in the coldest part and this difference increases with latitude. The other important information that can be extracted out of the presented graph is the effect of discrepancy between the 2 IR bands. The difference between the 2 IR bands increases the values of SST exponentially.

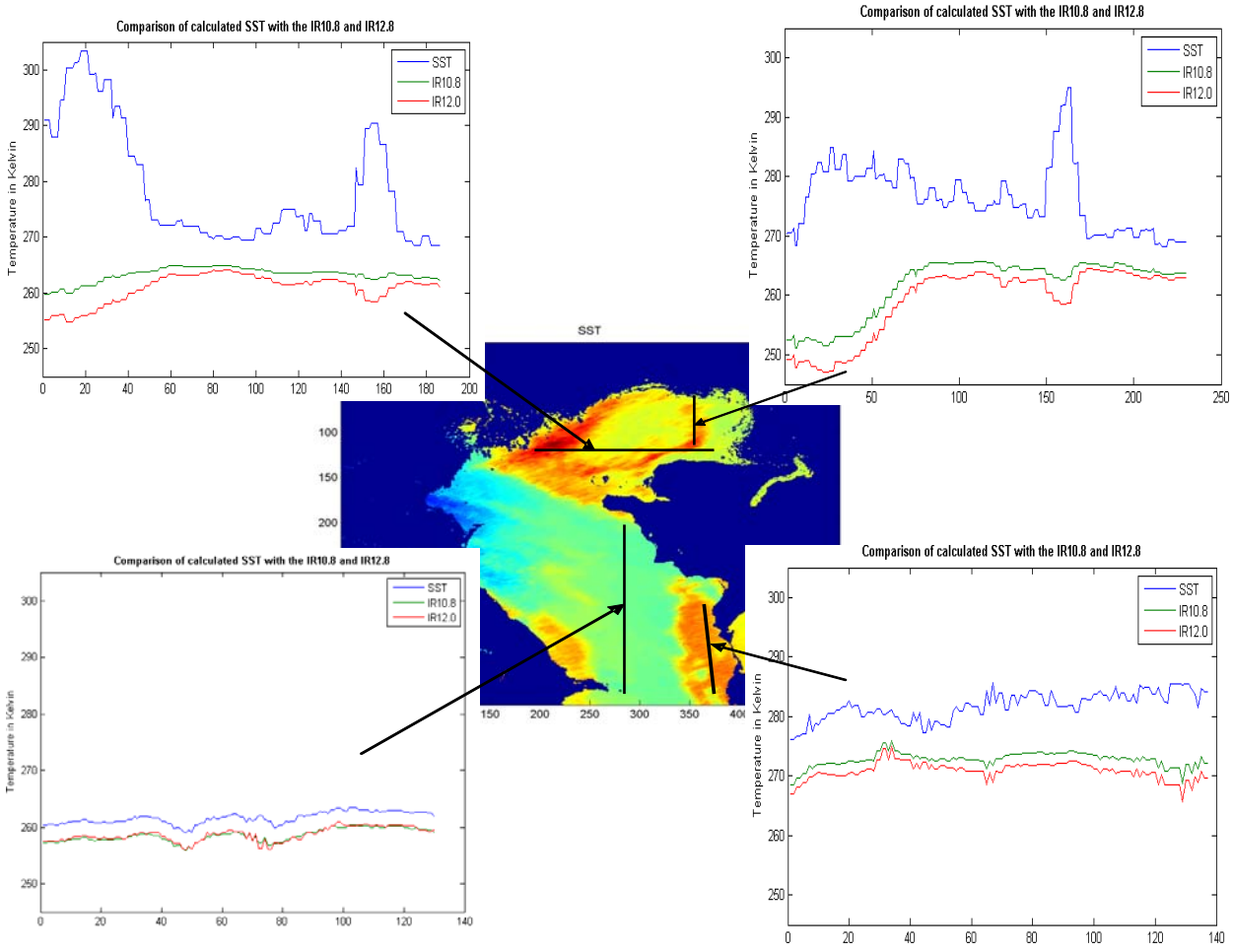


Figure 28: Transect of temperature in various regions for comparison of calculated SST with the IR10.8 and 12.0

In the above figure the upper left graph shows how the SST values spike with a few degrees of difference between the 2 IR channels. The best SST values are obtained when there is minimal dissimilarity between the 2 channels. In order to further investigate this model multiple scenarios have been designed to study the behavior of model. Figure 29 shows that

the best predicted SST comes when there is an almost zero difference between the 2 channels value. Conversely, SST increases as this disparity between channels grow.

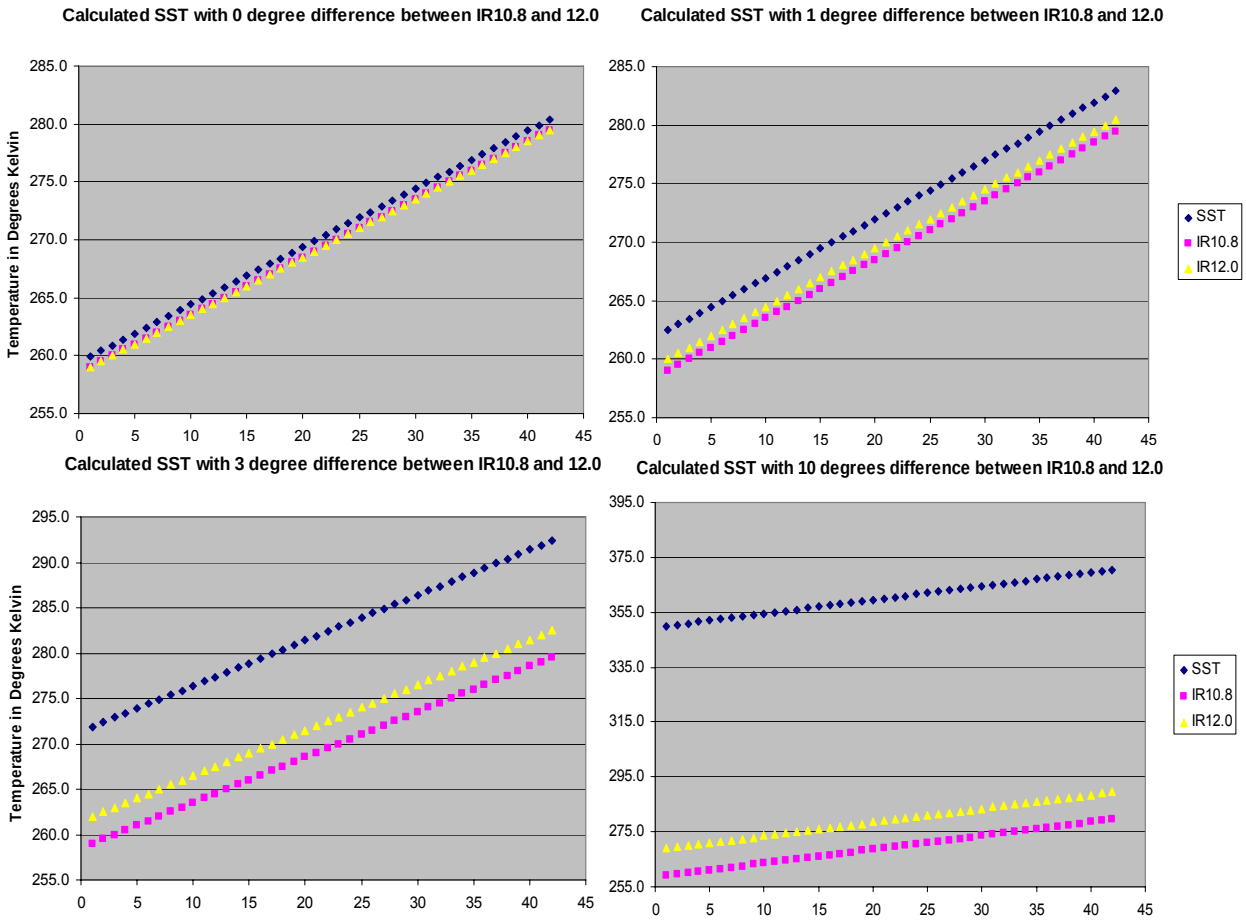


Figure 29: Transect of temperature in various regions for comparison of calculated SST with the IR10.8 and 12.0

This algorithm uses the infrared channels of 10.8 and 12.0 μm in following equation:

$$SST = T_{IR10.8} + [0.99\cos\theta + 0.21\theta]*(T_{IR10.8} - T_{IR12.0}) + [(0.364/\cos\theta) + 0.15]*(T_{IR10.8} - T_{IR12.0})^2 + [(0.327/\cos^2\theta) + 0.11]$$

In this model θ is the satellite zenith observation angle and measurements are assumed to be taken during cloud free days. Validation of the proposed algorithm was carried out by means of a field campaign off the hot water of coast of Valencia, Spain. This environment is very

different from coastal waters in colder regions. Therefore, we proposed a new algorithm that estimates SST as an average of temperatures from the IR 10.8 and IR 12.0 bands.

The following SST equation is an improvement over the existing model since it can accurately simulate the presence of ice in cold regions.

$\mathbf{SST_{RN}}$ = (T(IR) 10.8+T(IR)12.0)/2

In this approach, satellite zenith angles up to 70-75 degrees gives minimal changes in SST evaluation and thus was not included in the algorithm. The new model offers values that are consistent with temperature ranges over different regions of the Caspian. Figure 30 illustrates the new model (bottom right) that has been developed based on the average value of IR10.8 and IR12.0. It offers SST values which are consistent with sea ice in the Northern Caspian Sea and is validated by comparing it with both the IR channels and reflectance values. Thus, it could be stated that for cold water bodies with low presence of water vapor, this method will produce a reasonable indicator of SST for classification purposes. So far, the results are very promising in giving credence to the new proposed algorithm.

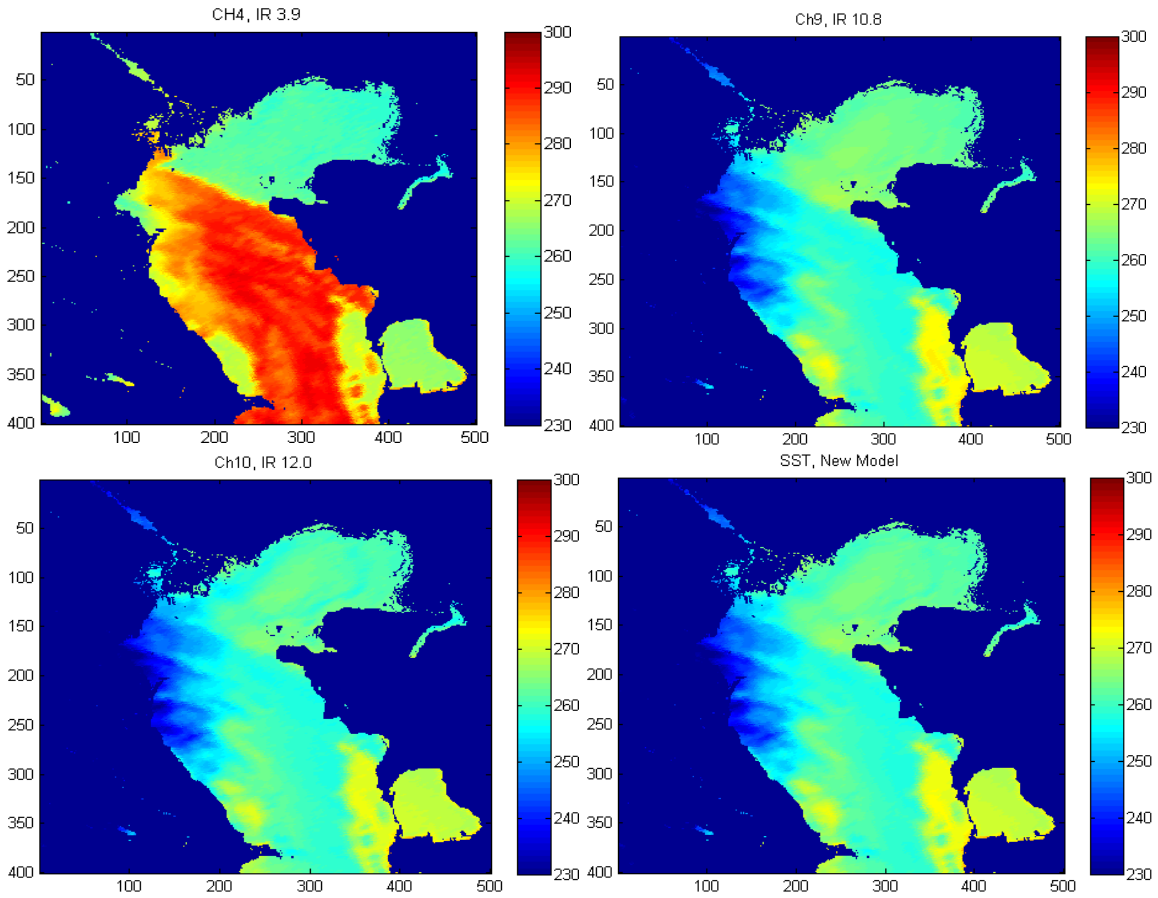


Figure 30: New SST Model (bottom right) shows consistent ice presence for data taken during winter of 2007 of the Caspian Sea

The new proposed model has been further tested for verification of its applicability to determine SST for cold water bodies containing ice. The Normalized Difference Sea Ice Index (NDSI) parameter has been used to confirm the presence of ice in the study model. So far, the results are very promising in giving credence to the new proposed algorithm and our own sea ice classified maps validate the accuracy of the new SST model (See Figure 31).

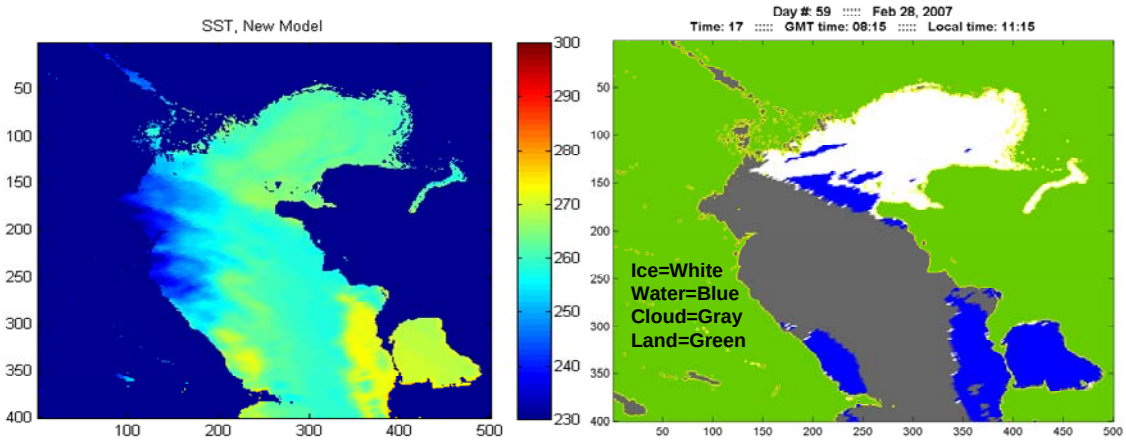


Figure 31: Comparison of new SST model to the classified ice map

Investigating Romaguera's model shows that he has developed and tested his algorithm over warm water bodies of the low latitudes of Tropics. His SST model shows high sensitivity to cold water and high latitudes as well as discrepancies between the 2 thermal IR channels of IR10.8 µm and IR12.0 µm since the model is primarily base on these channels. Extremely high values of SST in the very cold and icy waters of Northern Caspian Sea are the proof of such sensitivity and lack of performance of this model. On the other hand, the new proposed model has a realistic performance record in the cold water bodies. Particularly in the low presence of water vapor, the new model produces reasonable SST values.

5.3 Normalized Difference Sea Ice Index

The Normalized Difference Sea Ice Index (NDSI) parameter has been used to confirm the presence of ice in the study model.

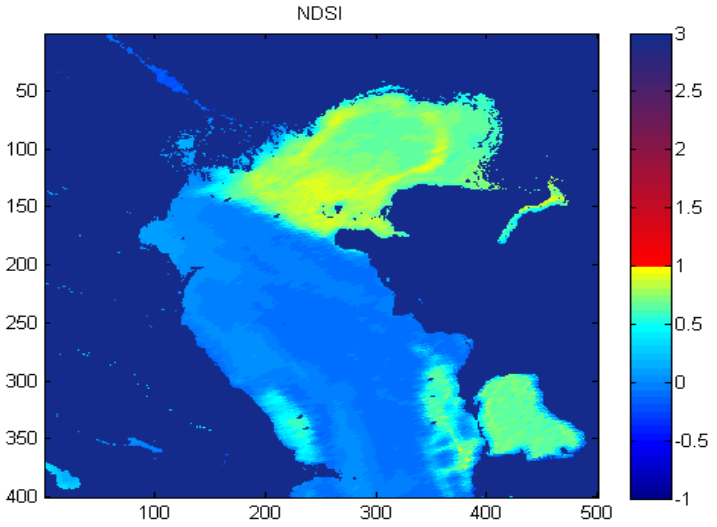


Figure 32: The Normalized Difference Sea Ice Index (NDSI) obtained from the visible and near infrared channels of MSG SEVIRI

NDSI is calculated from the combination of visible (VIS) and near infrared (NIR) band, thru the following equation:

$$NDSI = (VIS - NIR)/ (VIS + NIR)$$

MSG SEVIRI bands 1 (0.6 μm) and 3 (1.6 μ m) have been used to calculate the NDSI.

$$NDSI = (0.6\ \mu m - 1.6\ \mu m)/ (0.6\ \mu m + 1.6\ \mu m)$$

NDSI in mixed pixels is significantly less than a fully ice covered pixels. In general if a pixel has 50% or more ice, NDSI value will be more than 0.4. NDSI values range from -1.0 to +1.0, where -1.0 is a pixel with no ice and 1 for a fully sea ice covered pixel.

5.4 Development of Dynamic Threshold

Our aim is the development of ice-mapping algorithm for the processing of data from the next generation of NOAA Geostationary Operational Environment Satellite, R Series (GOES-R) that is scheduled for launch in approximately 2014. The radiation parameters are planned to be derived from the Advanced Baseline Imager (ABI) observations. A preliminary version of the ice-mapping algorithm has been developed for data in the Northern region of the Caspian Sea from the Spinning Enhanced Visible and Infrared Imager (SEVIRI) onboard the METEOSAT SECOND GENERATION (MSG) satellite surrogating for ABI instrument. In particular, we are interested in identification and mapping of ice cover that requires being distinguished from atmospheric cloud appearance and sea water surfaces. Threshold reflectance values for distinguishing these three different viewed matters are introduced. The cloud removals have been primarily done within the first round of classification. However the confusion will remain in classification of sea ice and water. When a general daily threshold is being used the classification process ends with a substantial number of unclassified pixels and mainly the confusion is in ice/water pixels. Therefore there is a need for the development of a dynamic threshold which takes in effect all the changes in terms of illumination of the target pixel.

The dynamic threshold will be a very useful tool in the classification process to minimize any influence from the Satellite Viewing angel or Solar Angel (SOL). The SEVIRI products need to be corrected from uncertainties produced from angular sources (viewing and solar) with the aid of Bidirectional Reflectance Distribution Function (BRDF) Models. For this purpose, cloud free pixels have been collected through the classification process after applying the cloud filter. Data is available in 14 acquisition times, starting from 8:45 to

15:15 local time with 30 minute intervals. Obtained data has been sorted into two classes of water and ice. In order to develop the threshold, we analyzed how often particular reflectance values of water or ice occur within certain acquisition time. For each acquisition period, the population distribution frequency plots for the ice and water data has been separately obtained. The ice data is best fitted to the normal distribution and passes the normal distribution tests and water data are best fitted to the lognormal distribution as it could be seen in the Figure 33. By overlapping the normal distribution curve (ice data) and lognormal distribution curve (water data), it is possible to examine the range of reflectance values for each class. As it could be seen in the following Figure, water has reflectance range of 0 to 33% and ice reflectance range goes from 15 to 60%. In order to define a threshold to classify ice and water we have to develop a series of tests to accurately separate the two classes. Additionally, it is well understood that most confusion is generated from pixels on the marginal range. As it could be seen in the Figure 33 the area that is covered by the two curves are the disputed territory and there is a need for a reasonable threshold that can make the classification possible.

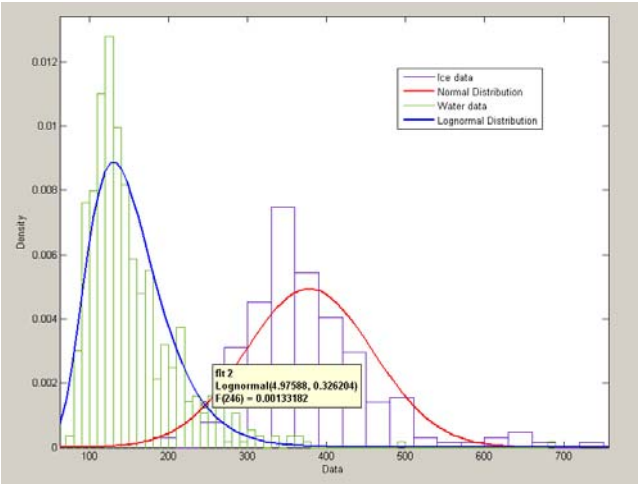


Figure 33: Distribution functions of water (left) and ice (right) reflectance for Feb 28, 2009

By looking at the above figure, it could be concluded that if we assume the cross point of the fitting distribution functions is the threshold value between ice and water pixels, there will be handful of pixels that are being miss classified. Therefore, there is a need for a development of an approach that accurately identifies the threshold. From the statistical parameters of the distribution fits (estimated mean and standard deviation), it can be calculated a probability that a pixel at certain acquisition time will be either ice or water.

Probability that pixel is water: $P_W = 1 - \text{LOGNORMDIST}(r, \mu, \sigma)$

Probability that pixel is ice: $P_I = \text{NORMDIST}(r, \mu, \sigma, 1)$

Where in the above mentioned equations, reflectance, estimated mean value and standard deviation are represented by r, μ, σ. The threshold value was obtained from comparison of the two probabilities:

$$\text{THRESHOLD} = \text{IF}(P_W > P_I, \text{“WATER”}, \text{“ICE”})$$

The following figure illustrates the steps in which each threshold has been developed.

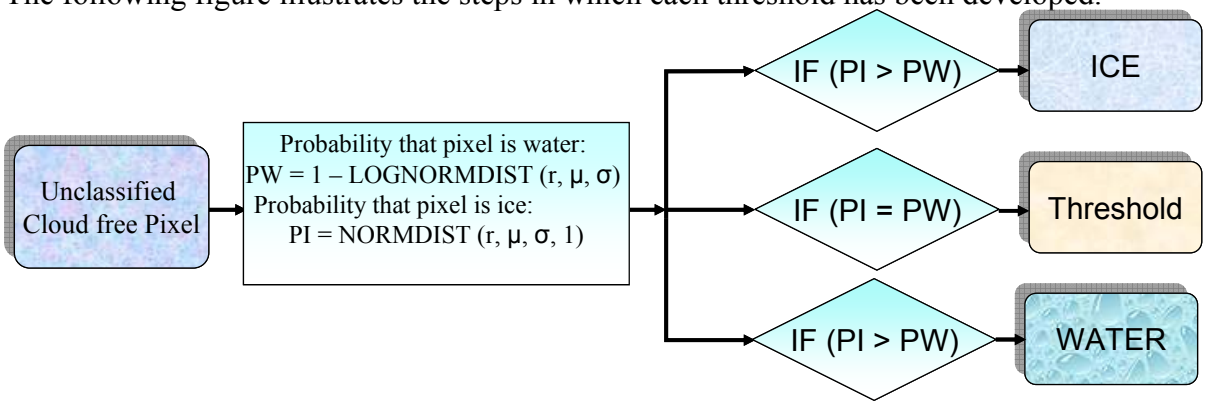


Figure 34: Flowchart of the operation for threshold development

As it could be seen, unclassified pixels enter the model and in the first step, their probability is calculated and then, they have to satisfy the conditions of the flowchart to be placed in a category (ice or water) with turning point recorded as the threshold.

This process was performed for three visible channels of High Resolution Visible (HRV 0.6-0.9μ), R01 (0.6μ) and R02 (0.8 μ) and the constant daily threshold values were obtained. The constant daily threshold was developed from the pool of manually inspected ice and water reflectance values of one day of data and later was applied to the classification model.

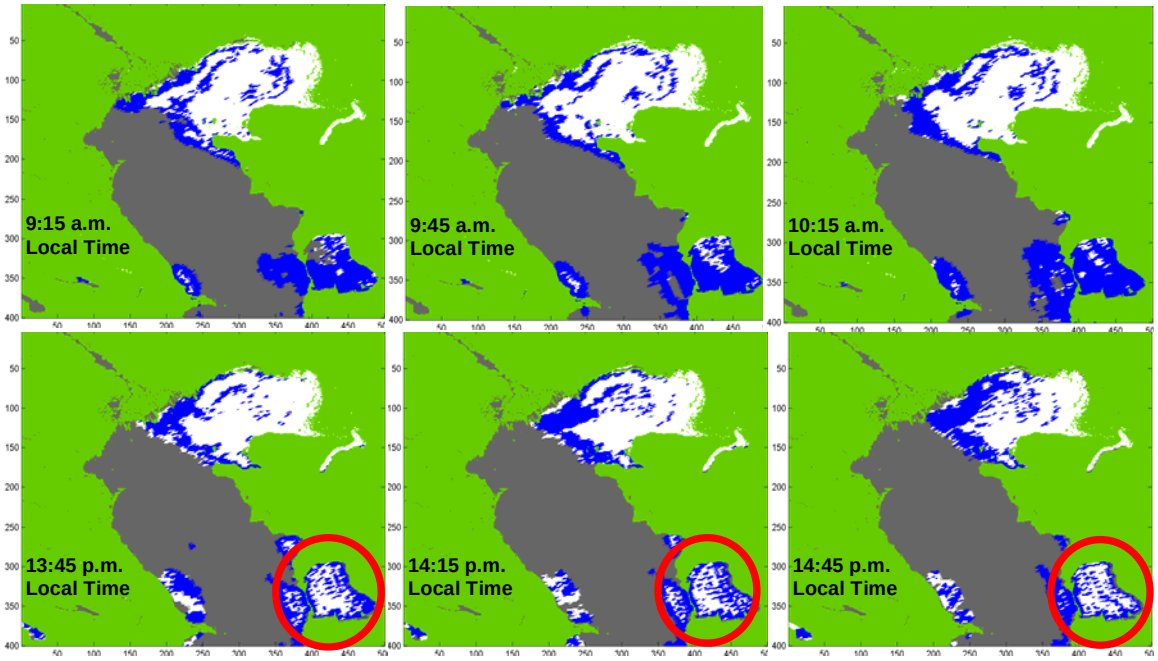


Figure 35: Classified images of Caspian Sea mid winter, Feb 28, 2007, with Daily Constant threshold, Ice=White, Water=Blue, Cloud=Gray, Land=Green

Figure 35 illustrates the classified images after the constant daily threshold has been applied which has a reasonable performance. It generates clean images with very few misclassified pixels in midday (around noon of local time) but when it comes to the early morning and late afternoon as the solar illumination angle increases, the number of misclassified pixels increases. The further investigation is required to be done on the properties of channels and their daily variations to understand the fluctuation. The graphs presented in Figure 36 show the hourly variation of reflectance for one ice and three water pixels in SEVIRI optical channels; HRV (High Resolution Visible: 0.6-0.9 μm), R01 (0.6 μm), R02 (0.8 μm), and the Near Infrared (NIR) channel R03 (1.6 μm). The water pixels were selected at three different latitudes to illustrate the effect of viewing geometry on reflectance. This data were collected

under a partially clear sky condition manually with visual inspection on Feb 3rd 2007. The x axis represents the number of daytime acquisitions (with 30 minutes intervals) and the y axis represents the reflectance in percentage. These graphs detect a presence of clouds over one water-pixel at the end of the day. The R01 channel showed a good separation between ice and water and R02 has the lowest spatial variability of water reflectance and the best separablity between water and ice, however, some types of cloud can easily be confused with ice. On the other hand, the near-infrared channel (R03) gives the best separability of clouds and ice/water but an inadequate discrimination between water and ice. This channel also shows the lowest sensitivity of water pixels to latitude, viewing geometry and water properties.

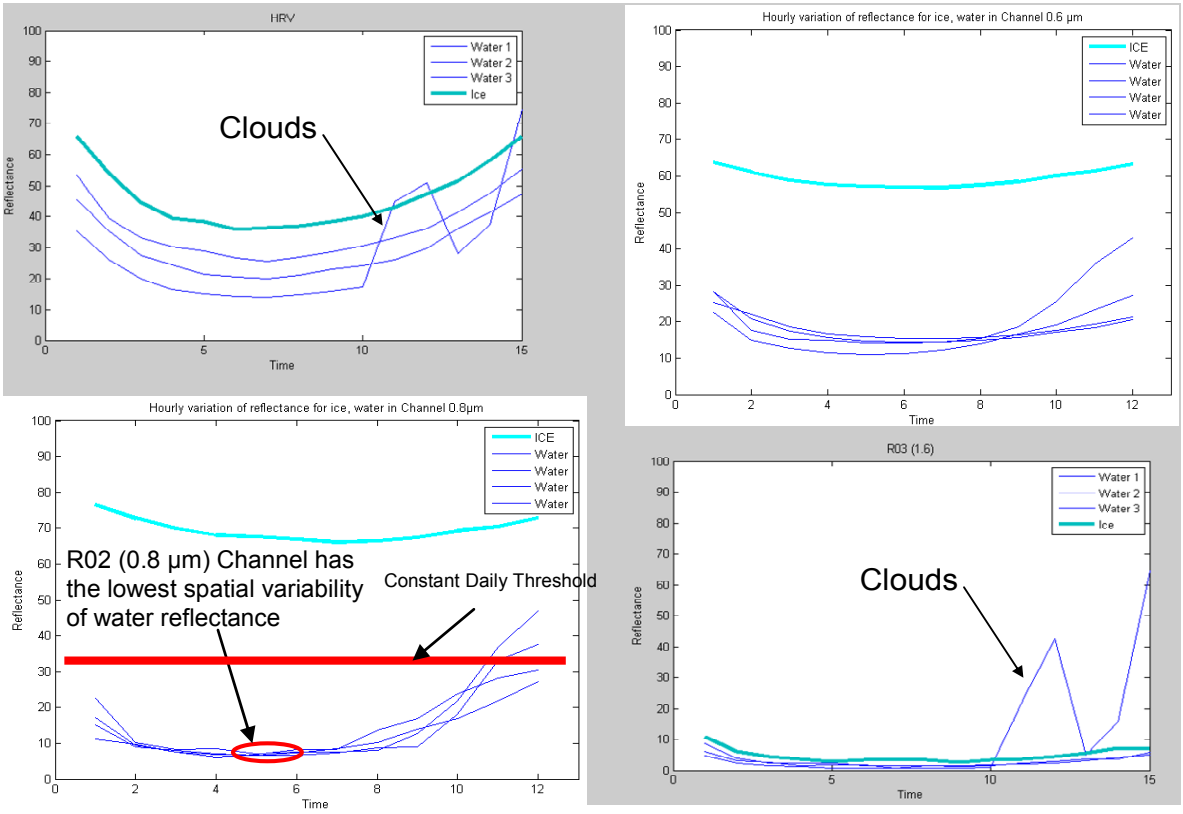


Figure 36: Hourly variation of reflectance for ice, water and clouds

The other noticeable fact about Figure 6 is the effect of viewing angle geometry that creates the curve shape of the reflectance vs. time graph with the highest reflectance at early morning (sunrise) and late afternoon (sunset), and lowest reflectance at mid day (noon). This variation is the primary source of the misclassified pixels in early morning and late afternoon in the classified images after applying the constant daily threshold. As it can be seen in Figure 36 the red line shows the position of the constant threshold. Hence, all the pixels above the line are been classified as ice and the pixels below that are water. However, because of the daily variation of solar angle, the water pixels that cross the line are reflecting higher and in such situations and being classified as ice. To resolve this problem, there is a need for a threshold that could consider all the fluctuations and curve through them. Consequently, instead of having one daily constant threshold we need a dynamic threshold that could be applied to all acquisition times. In the next step instead of using the pool of daily ice and water, we are going to use the data available for every single acquisition time.

In Table 8, estimated mean and standard deviation values for water and ice pixel distributions are shown. These values were obtained from distribution fitting tool of Mathlab software. The probabilities of water and ice appearance were used to calculate the threshold values that are summarized in Table 9. The thresholds are illustrated graphically in Appendix A.

Table 8: Estimated means (μ) and standard errors (σ) for individual acquisition times (AT) obtained from population distribution frequency plots of ice (normal distribution) and water data (lognormal distribution)

AT for HRV	μ_{water}	σ_{water}	μ_{ice}	σ_{ice}	Probability of Water	Probability of Ice
12	5.54612	0.265713	498.596	71.255	0.060367988	0.058656782
13	5.37911	0.295547	464.709	73.2454	0.055804977	0.05402173
14	5.25495	0.314681	447.018	88.213	0.061689533	0.061545676
15	5.17933	0.317216	427.717	90.5096	0.063678282	0.061334087
16	5.13163	0.312326	419.534	93.234	0.061579051	0.059267284
17	5.12046	0.31151	415.269	95.872	0.063951004	0.063546079
18	5.14256	0.30692	421.679	100.045	0.067154428	0.06601918
19	5.19847	0.302623	425.857	98.9345	0.073032342	0.071573652
20	5.25995	0.292207	438.584	96.6874	0.070380804	0.070147872
21	5.32966	0.280398	458.584	96.3945	0.067207218	0.066817615
22	5.41504	0.259169	483.858	93.2881	0.059043641	0.057715812
23	5.51184	0.246293	512.368	93.0338	0.057559896	0.05659385
24	5.62546	0.242143	551.184	95.0749	0.061490619	0.059545217
25	5.74055	0.222573	589.201	89.525	0.054101404	0.053618809
All AT	5.32213	0.341784	463.101	104.536	0.091302273	0.090087285
AT for R01	μ_{water}	σ_{water}	μ_{ice}	σ_{ice}	Probability of Water	Probability of Ice
12	5.27172	0.299067	398.584	48.3189	0.050550746	0.047682365
13	5.09492	0.334514	372.996	52.0239	0.04777672	0.045375125
14	4.96722	0.358171	357.456	63.4799	0.054499563	0.053264667
15	4.87951	0.366864	342.547	66.5286	0.055608397	0.054630703
16	4.82217	0.358421	336.101	70.8592	0.054024077	0.052149274
17	4.80413	0.360139	331.783	72.2056	0.056379748	0.054410138
18	4.81868	0.354139	335.679	74.2029	0.056512994	0.054867549
19	4.87224	0.348469	337.55	71.956	0.059295638	0.058891136
20	4.92516	0.333097	345.149	69.6357	0.055735979	0.05522769
21	4.99663	0.323761	357.399	69.4412	0.056624659	0.055937706
22	5.0753	0.295946	374.182	66.4422	0.047857223	0.045665807
23	5.17438	0.281579	391.159	63.2616	0.045948526	0.045141879
24	5.28616	0.270705	414.91	61.1931	0.046893945	0.044747533
25	5.3995	0.2466	439.089	58.1936	0.042940079	0.0411834
All AT	5.00435	0.375608	363.386	73.4701	0.074951807	0.071921505
AT for R02	μ_{water}	σ_{water}	μ_{ice}	σ_{ice}	Probability of Water	Probability of Ice
12	4.93432	0.392729	372.348	74.7791	0.059923369	0.059867212
13	4.73097	0.439234	348.229	81.051	0.061840853	0.061165365
14	4.58361	0.474742	333.956	97.2679	0.076298335	0.073647529
15	4.48446	0.484813	317.761	100.928	0.080267422	0.078609981
16	4.41291	0.475564	309.549	106.415	0.081872495	0.079960457
17	4.374	0.472167	302.317	107.868	0.084352236	0.081730202
18	4.37387	0.464312	305.741	109.469	0.080765134	0.080096166
19	4.43088	0.462627	302.098	106.33	0.088224277	0.086189281
20	4.47877	0.439432	304.369	104.567	0.087356412	0.083695402
21	4.54809	0.427661	311.371	104.627	0.086844778	0.086796641
22	4.6263	0.39181	323.855	100.69	0.076066839	0.075128613
23	4.72848	0.372391	335.781	95.3074	0.071839734	0.069820987
24	4.84849	0.355601	356.234	88.7921	0.062637507	0.062477544
25	4.97588	0.326204	377.804	80.8283	0.052287349	0.051480356
All AT	4.58445	0.474436	325.83	102.666	0.088235136	0.086600621

All three channels have similar trends. The reflectance values are the highest in the beginning and at the end of the day and the lowest values are reached at the local noon. The highest values differ from the lowest reflectance approximately by 10% (Table 9). Daily threshold for distinguishing ice from water in HRV has range of 26.9-38.7% with average daily reflectance of 32.3% and ranges between 21.6-31.8% with average daily reflectance of 25.6% in R01 (0.6μm) and in R02 (0.8 μm) the reflectance ranges from 15.2- 25.6% with average daily reflectance of 18.6%.

Table 9: Threshold values for HRV, R01 and R02 Reflectance channels

Acquisition Time	Local Acquisition Time	Treshold HRV (%)	Treshold R01 (%)	Treshold R02 (%)
12	8:45	38.7	31.8	25.6
13	9:15	34.7	28.5	22.3
14	9:45	31.1	25.5	19.3
15	10:15	28.8	23.6	17.5
16	10:45	27.4	22.1	16
17	11:15	26.9	21.6	15.2
18	11:45	27.1	21.7	15.2
19	12:15	28.1	22.5	15.7
20	12:45	29.6	23.4	16
21	13:15	31.4	24.7	16.9
22	13:45	33.7	26.2	17.9
23	14:15	36.5	28.4	19.5
24	14:45	40.3	31.1	22
25	15:15	44.5	33.8	24.6
All (12-25)	8:45 - 15:15	32.3	25.6	18.6

As it could be seen in Figure 37 the developed dynamic thresholds are curving as time changes and solar illumination angle varies. Unlike the constant daily threshold, dynamic threshold realizes ice and water in different time of the day even with the broad range of reflectance in these 2 classes.

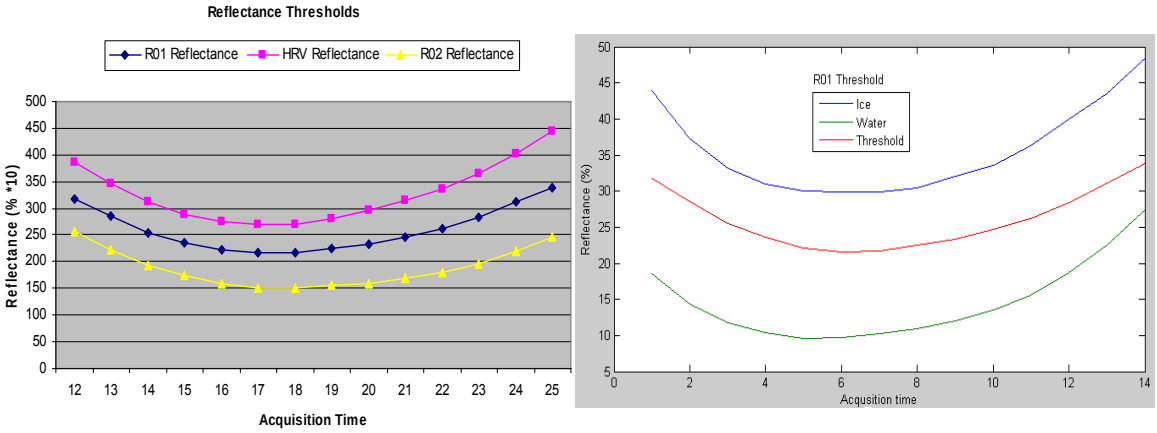


Figure 37: (Left) Threshold values for HRV, R01 and R02 Reflectance during local acquisition time 8:45 to 15:15, (Right) The dynamic threshold (Red) in R01 (0.6µm) classifying Ice and Water Pixels

The right graph in Figure 37 depicts the function of the dynamic threshold in the classification process. As the selected ice and water pixels' reflectance fluctuates during the day, the dynamic threshold in walking the fine line in between is able to distinguish water from ice throughout the day. The dynamic threshold can not be easily crossed in early morning and late afternoon when water reflectance is higher. Later, the dynamic threshold was applied to the classification model for operational tests. As it could be seen in Figure 39, the number of misclassified pixels has substantially decreased and the produced images are clean. In addition, the dynamic threshold added certainty to the ice classified pixels and has helped the model to generate a coarser ice and water pixels. In comparison between Figure 35 (classified images with constant daily threshold) and Figure 38 (classified images with dynamic threshold), it is clear that there is an application and functionality of the dynamic threshold.

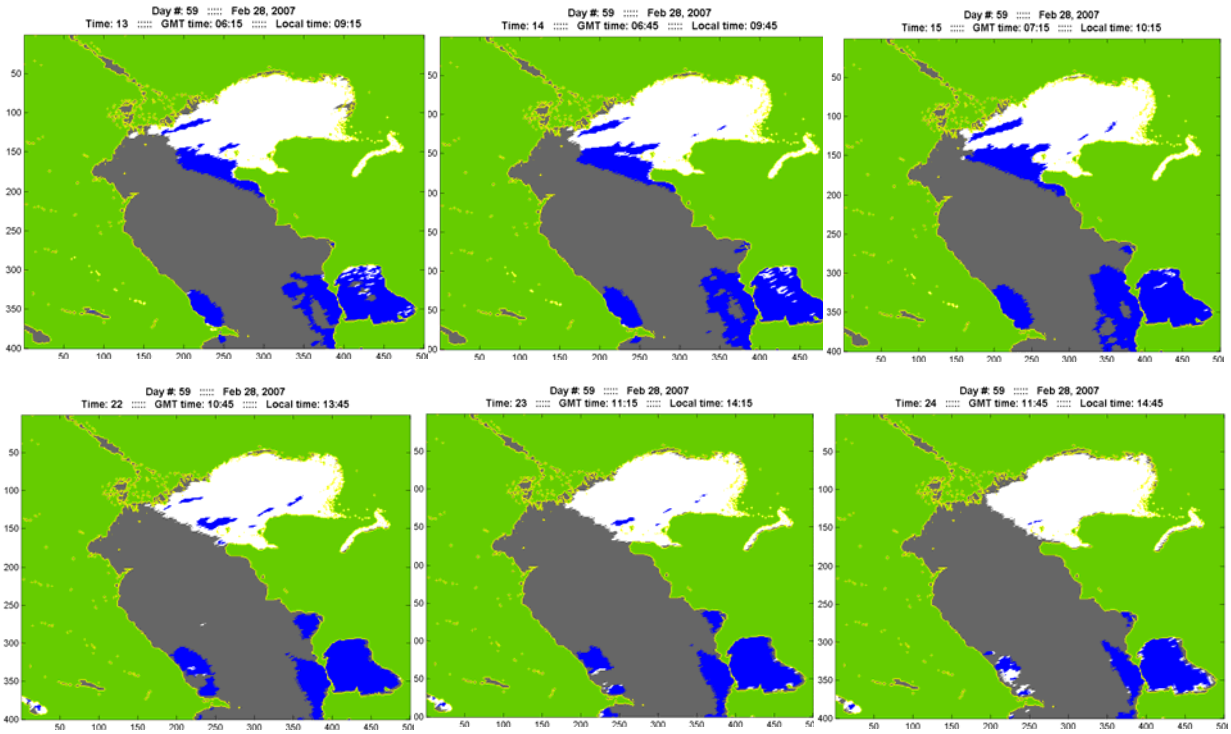


Figure 38: Classified images of Caspian Sea mid winter, Feb 28, 2007, with Dynamic threshold, Ice=White, Water=Blue, Cloud=Gray, Land=Green

5.5 Study of Channels Correlations

In the previous chapters the daily and seasonal variations and other properties of each channel has been individually studied. The optical channels have been investigated for the angular variations of the solar, satellite and zenith angle. It has been established that the High Resolution Visible (HRV 0.6-0.9) and R01 visible (VIS 0.6) have the highest sensitivity and near infrared channel (NIR 1.6) has the least sensitivity to the variations of solar angle. The results of that finding led to the formation of the BRDF model and the establishment of the daily minimum and maximum values for each spectral band. In the following correlations of the channels, derived parameters and indices will be investigated for the formation of the classification model. In addition to the parameters and values there is a need for a set of criteria and boundaries for each class. When two of these parameters are plotted against each other, clusters or pattern will be recognized. After the recognition, these patterns could reveal the location of these pixels. If a group of the pixels have high reflectance values in VIS 0.6 and NIR 1.6 bands, as it can be seen in the Figure 40, then they can be classified as cloud. Figure 39 indicates that the pixels that have low reflectance in both VIS 0.6 and NIR 1.6 belong to the lower part of Caspian Sea (lower right graph) where, there is no ice or cloud and is water. The green arrow in the figure points to the location of these pixels on the Caspian Sea image. The scatter plot in the Figure 39 further illustrates that ice and cloud can be separated by their unique act in the combination of these two bands. As it could be seen ice has high reflectance in VIS 0.6 and low reflectance in NIR 1.6 which separates it from cloud. The red arrow in the Figure points to the location of ice on the image.

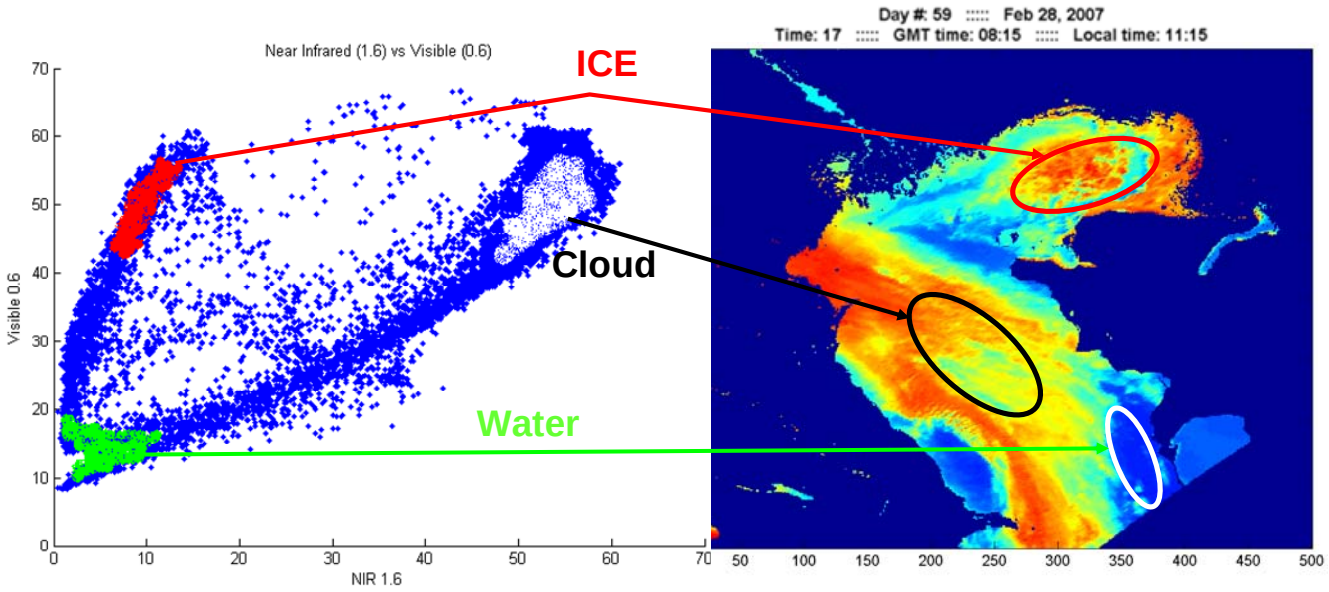


Figure 39: Scatter Plot (left) of near infrared (band 1.6) vs. visible (0.8), HRV image (right) shows clouds (black oval), Ice (red oval) and Water (white oval)

When other key parameters like NDSI and NIR 1.6 are plotted some other pattern are being revealed. Figure 40 shows, ice has high values of NDSI and low values of NIR 1.6 and cloud has very high value of NIR 1.6 and low value of NDSI.

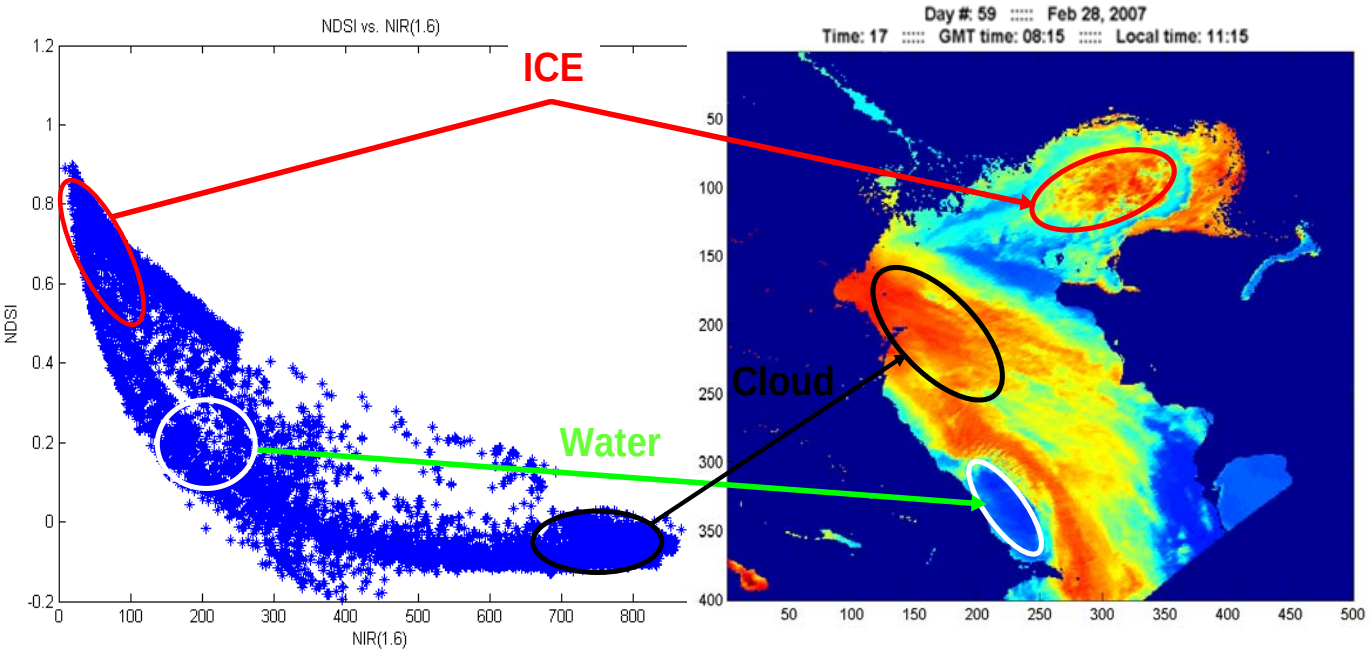


Figure 40: Scatter Plot (left) of NDSI vs. near infrared (band 1.6) HRV image (right) shows clouds (black oval), Ice (red oval) and Water (white oval)

And as we expected water has NDSI of 0-0.4 with low reflectance in NIR1.6. White ovals in Figure 40 highlight the location of water in pixels in both scatter plot and Caspian Sea image.

The further study of scatter plots reveals some other important information that is essential in the classification model building. Figure 41 shows the capabilities of the scatter plot in the recognition of an extra class. Blue oval in Figure 41 shows a group of pixels with high reflectance in NIR 1.6 band and extremely low temperature. These properties indicate the presence of ice cloud, which can easily confuse the model if ignored.

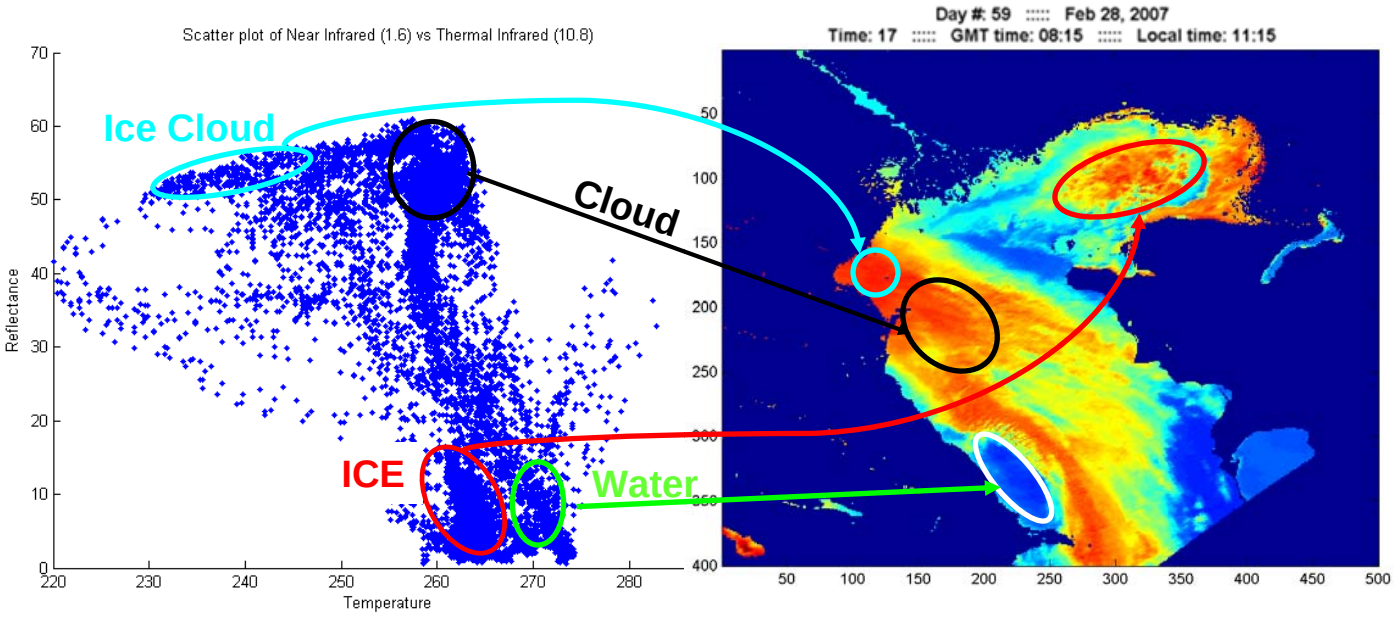


Figure 41: Scatter Plot (left) of near infrared (band 1.6) vs. thermal infrared (band 10.8) HRV image (right) shows clouds (black oval), ice cloud (blue), Ice (red oval) and Water (white oval)

The study of the clusters in the scatter plots will lead to the discovery of the common properties of them and the establishment of the thresholds that we need for classifications.

5.6 Sea Ice Mapping Model

Sea ice is differentiated from water and clouds by thermal and reflectance parameters. According to the National Snow and Ice Data Center (NSIDC), pixels collected from images using MODIS/Terra satellite are classified as sea ice if the ice surface temperature is less than or equal to 271.5 Kelvin. Moreover, sea ice reflectance values are evaluated using the visible and near-infrared regions of the spectrum to determine windows for ice detection. Based on our collected parameters and factors a pixel is characterized as sea ice if it meets the following criteria:

- NDSI value >= 0.4
- SST value =< 271.5
- Band 1 has r > %25
- Band 2 has r > %18

According to NSIDC, reflectance values are represented as percentages and should not exceed 100% for ice detection. Normalized Difference Sea Ice Index (NDSI) is useful for the detection of high reflectance. The NDSI is especially useful because it is insensitive to a range of illumination conditions and atmospheric effects. NDSI values not only validate the presence of sea ice, it is also used in the classification threshold values that distinguish ice from cloud or ice from water pixels.

Thermal inputs to identify ice have been proven to give satisfactory results. The analysis of SST data values from channels 9 and 10 of the SEVIRI sensor has already been proven to give fairly accurate results. For image classification, we applied threshold tests that identify cloud and ice pixels using specific thermal and spectral band ranges that are applicable to MSG satellite data. The extended criteria are listed below.

- Channel 2 (0.6μm) is combined with Channel 4 (1.6 μm) for NDSI determination.

  NDSI >= 0.3 gives ice pixels.

- SST is determined with a combination of the satellite viewing angle and brightness temperatures of channels 9 (IR 10.8 μm) and 10(IR 12.0 μm). If SST<= 271.5K, then ice is present.
- Channel 2 gives reliable reflectivity responses to ice and water. Ch 2 >= 264 has been found to represent ice pixels. Several pixels have been examined to establish these threshold relationships.
- The converse of all the above equal water pixels.
- Cloud determination is extremely important in classification as it can give misleading data on the ice extent. Cloud has wide range of properties, they could be very cold and thus, their pixels have considerably lower thermal values. Furthermore, reflectivity values are also high in the near infrared band of the spectrum. SST <= 240K and Channel 4 > 200 is recognized as cloud pixels. The opposite can suggest cloud-free pixels.

The High Resolution Visible Channel (HRV) or Channel 1 has a ground resolution of 1km which allows for detection of land pixels from satellite data.

Figure 42 illustrates the flowchart of the sea ice classification and the detail of the classification steps are presented in this Figure. NSDI and SST values are calculated simultaneously. Dynamic threshold box is calculated previously through the appropriate algorithm which is presented in Figure 34. The outputs of this model are four classes of land, water, ice and cloud. This model is only based on the collected threshold values and since it only takes 55sec to generate the hourly ice map, it is one of the fastest and most efficient sea ice mapping tools. This model could be the primary mapping tool for the satellite which could generate the maps in the shortest time.

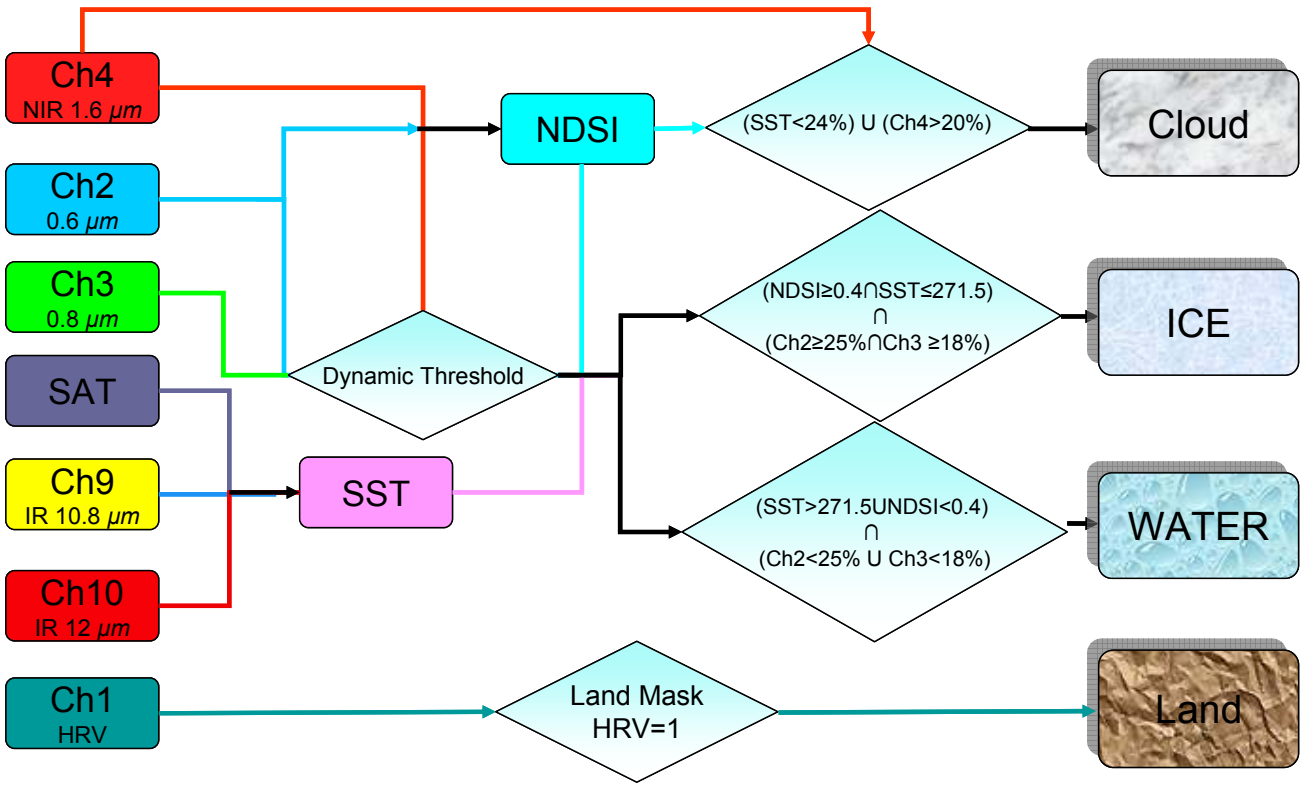


Figure 42: Flow Chart of the Operation

As Figure 42 indicates all the available sea ice bands on board of MSG SEVIRI are being deployed in the model. The model generates four layers (ice, water, cloud and land) which are mounted on each other to produce the sea ice map with the desired temporal resolution. Depends on the application this model is capable of producing hourly and daily and multi-day image composited sea ice maps. The post classification and mapping process is depicted in Figure 43. Each class has been color coded according to the property of that class. Land has been given the color green, water is blue, ice is white and cloud has been coded by gray color.

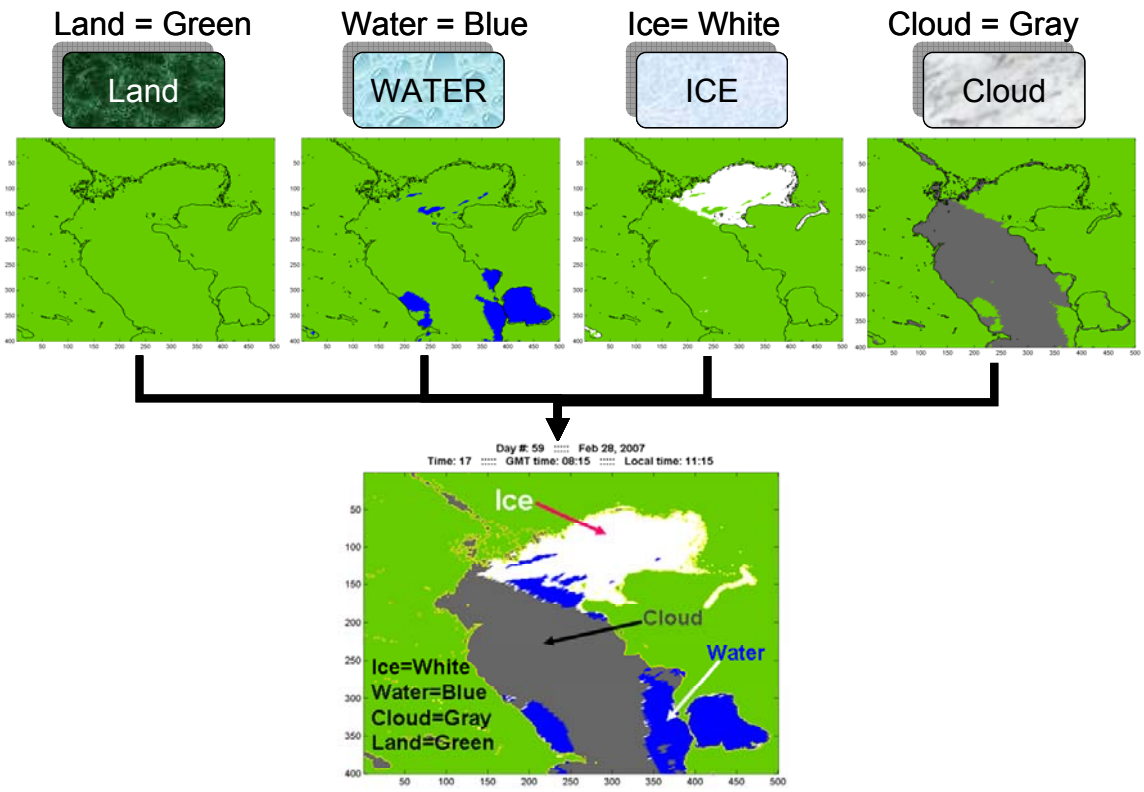


Figure 43: Threshold based ice classification model

The developed algorithm for this part is very efficient and easy to work with and could be uploaded in any systems. In order to generate the hourly ice map, the model does not need any extra data files to work. As soon as the satellite image enters the sea ice could be mapped. In order to make the daily ice map, one full day of data is required and for the multi day image composited maps with no cloud, two to three days of data is needed to collect enough cloud free observation to make the map. The model has been very successful so far and the quality of the produced maps is well above the existing maps. The other notable point about this model is that it does not have any fixed values or coefficient that has to be change if the model is being used over a different region. The comparison of the sea ice maps produced thru this model is compared to other models and existing maps in the conclusions chapter.

5.7 Development of Artificial Neural Network (ANN)

This part of the study focuses on the development of Artificial Neural Network (ANN) model for the future GOES-R mission. The ultimate objective of this research is to explore the potentials of the future GOES-R ABI in mapping sea ice and to develop an automated ice-mapping algorithm. Therefore, an empirical approach has been used to determine the atmospheric and angular correction. In order to characterize bidirectional properties of the top of the atmosphere reflectance, a linear combination of functions depending on observation geometry angles (solar-satellite relative azimuth along with solar and satellite zenith angles) is applied. This simplified approach can not adequately represent bidirectional effects for all possible geometries. However, observation geometries involved in ice identification and mapping from geostationary satellites are limited to high solar zenith and satellite zenith angles, generally over 50 degrees. Thus for this particular application, the use of an empirical approach may be appropriate. This study is a first attempt to apply a Bidirectional Reflectance Distribution Function (BRDF) model for ice cover mapping. Moreover, an ANN technique has been also utilized for model calibration and application. ANN has been widely utilized in remote sensing applications (Benediktsson et al. 1990; Schowengerdt 1995).

Multi-layer perceptron trained by back propagation algorithm is the most common NN used for image classification. This type of NN has been successfully applied to image processing and has shown a great potential in the classification of different types of remotely sensed data. In contrast to traditional techniques such as regression analysis, NN uses its complex configuration to find the best nonlinear function between the input and the output data

without any constraint of linearity or pre-specified non-linearity (Ghedira et al. 2007; Ghedira et al. 2005).

5.7.1 Methodology

Data collected by SEVIRI instrument onboard of Meteosat Second Generation (MSG) satellite have been used as a prototype. Four channels were utilized: HRV (High resolution Visible: 0.6-0.9 μm), R01 (0.6 μm), R02 (0.8 μm) and R03 (1.6 μm). The northern region of the Caspian Sea has been selected for algorithm development and validation. The surface area of the Caspian Sea is about 371,000 square kilometers. Ice conditions dominate during the winter in the northern region of the Caspian Sea. It is crucial to monitor ice extent and condition since the Caspian Sea is an oil rich region (Kouraev 2004). Similar to snow, the reflectance of thick ice cover is very high in the visible and drops substantially in the shortwave- and middle-infrared. This specific spectral signature provides the physical basis for ice identification from space. It will be primarily used in the new ice detection algorithm for GOES-R ABI. Clouds present the major factor hampering ice identification and mapping. As compared to polar orbiting satellite data, availability of frequent observations from geostationary satellites increases the chance to obtain a cloud clear view during a day and thus helps to reduce cloud gaps in the ice map. The proposed approach is based on combination of a BRDF model and an ANN technique. The output of this combination is a simulated value of reflectance. Reflectance values of ice and water were estimated in this study. These reflectances are supposed to be generated by cloud free ice and water pixels, respectively. However, two necessary steps were performed before simulating ice and water reflectances as shown in flowchart of figure 44. Firstly, a sample of pixels of the Caspian Sea was manually selected. This sample was used for neural network training and model

validation. Additional details about the neural network structure will be provided in the following sections. The selected pixels were completely covered by ice or water. Several clear-sky MODIS images have also been used to improve the identification of fully covered ice pixels. A sample of more than 12,000 pixels was selected for both water and ice.

Secondly, an automated approach for cloud detection was performed using a simple threshold method. The channel R03 (1.6 μm) was used to automatically detect cloudy pixels.

In this work, the BRDF model describes an existing relationship between three angles i.e. satellite, solar and azimuthal angles and observed reflectance. Trigonometric functions such as the sine and cosine of these angles were also utilized in the BRDF model formula. The total number of inputs which are the three angles, their sine and cosine is 9.

$$R_{obs} = f(ARZ; SOL; SAT; \cos(ARZ); \cos(SOL); \cos(SAT); \sin(ARZ); \sin(SOL); \sin(SAT)) \quad (1)$$

Where SAT is the satellite angle; ARZ is the azimuthal angle and SOL is the solar angle and R_{obs} is the observed reflectance. The main assumption of this work is that simulated reflectances do not depend on the ice or water features i.e. roughness, thickness or quality. Therefore, assuming these characteristics remain constant during a day, the observed reflectances are exclusively affected by the three angles variations: Azimuthal, Solar, Satellite. The ANN has been used, on the other hand, to model the relationship between angles as inputs and reflectances as outputs. In a multi-layer neural network, the nodes are organized into layers where each node transforms the inputs received from other nodes. The input layer serves as an entry for the vector of data presented to the network (Azimuthal, Solar, Satellite and their sines and cosines). The output layer serves to produce the neural network decision (simulated reflectance) for the pixel presented at the input layer. All layers

between the input and output layers are referred to hidden layers. The best neural network architecture can only be determined experimentally for each particular problem. The number of hidden nodes should be large enough to ensure a sufficient number of degrees of freedom for the network function and simultaneously small enough to keep sufficient generalization ability to the network. Several tests were conducted to determine the optimal number of hidden nodes. After several tests, a configuration of one hidden layer with 20 hidden nodes was selected.

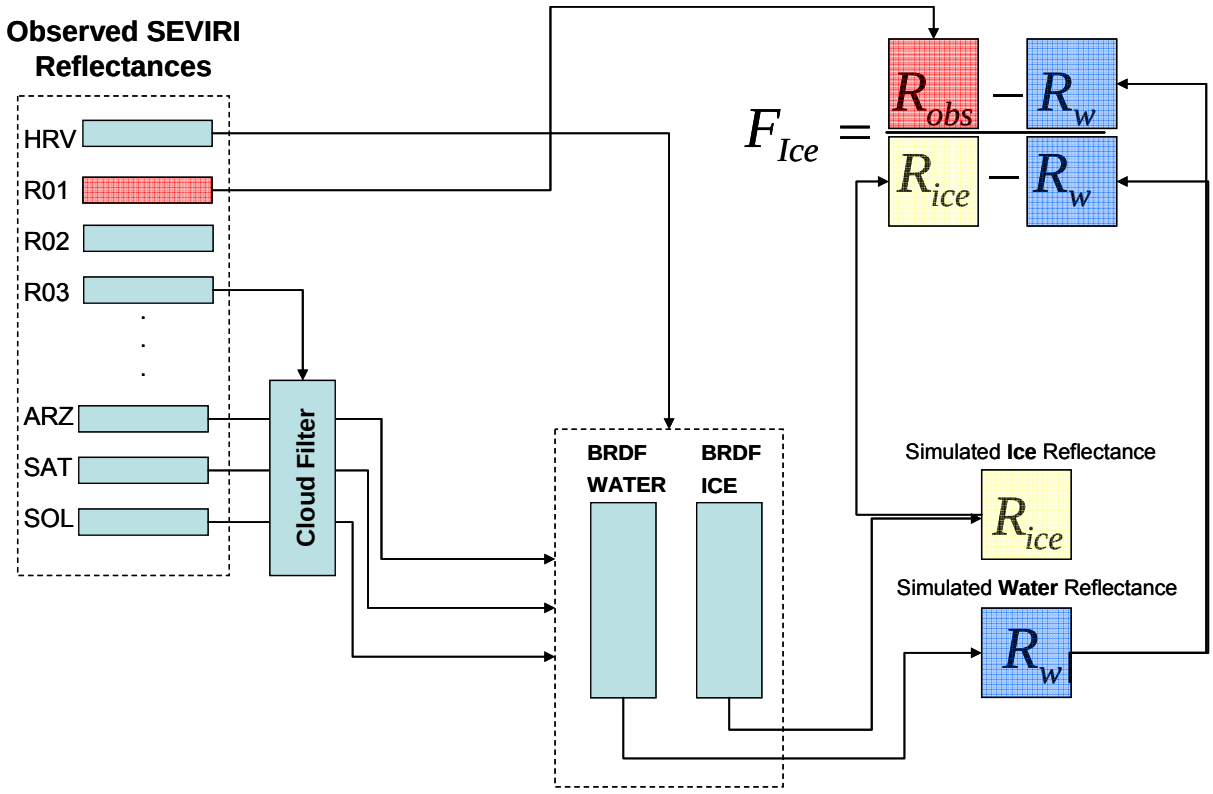


Figure 44: Flowchart of ANN based Sea Ice Mapping from Geostationary Satellites

5.7.2 Results

The cloud discrimination potentials of the near infrared channel can also be seen in Figure 45. This figure shows the reflectance of the four optical channels at 11:45 am local time. These data were collected on January 23rd, 2007. As it was discussed above, simulations

were carried out, according to the flowchart of Figure 44. Cloudy pixels were detected and eliminated. Then, the neural network was trained. The primary goal of this training step is the estimation of the weights connecting the nine input-layer nodes (angles + trigonometric transformations) to the 20 hidden nodes, and then the ones connecting hidden nodes with the output layer containing the observed reflectance.

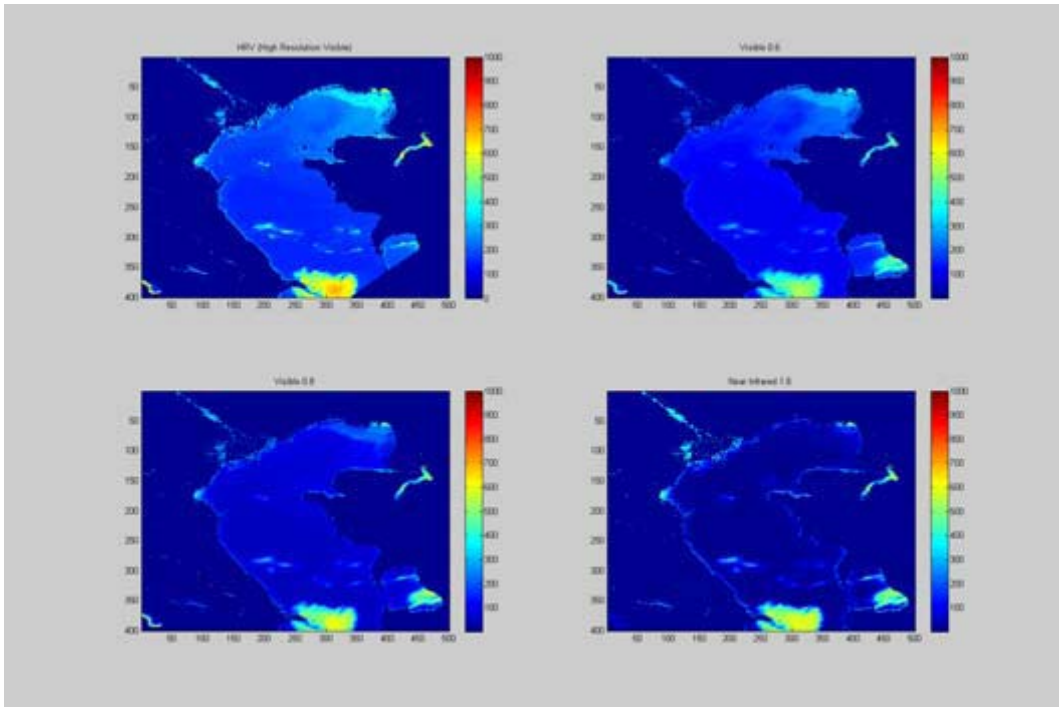


Figure 45: Reflectance in HRV and optical channels

Two separate neural networks with same structure have been trained to simulate ice and water reflectance. Figures 46 and 47 show the simulated reflectance for ice and water, respectively. Firstly, these results illustrate a satisfactory agreement between simulated and observed reflectances. The root mean squared errors (RMSE) for both ice and water summarized in table 14 are non significant. However, RMSE values were systematically higher when ice reflectances are simulated. This can be explained by the fact that ice reflectance are highly affected by ice features such as roughness, thickness and presence of

fractional ice. RMSE obtained with simulated water reflectances can be generated by the variation of water reflectance due to the atmospheric effect as well as to water properties such as high concentration in sediments, presence of river deltaic deposits and presence of fractional ice. In future work, the simulated reflectances will be used to retrieve these features. Overall, the simulated performances are acceptable and very encouraging. This implies that a combination of BRDF model and ANN allows simulating ice and water reflectances.

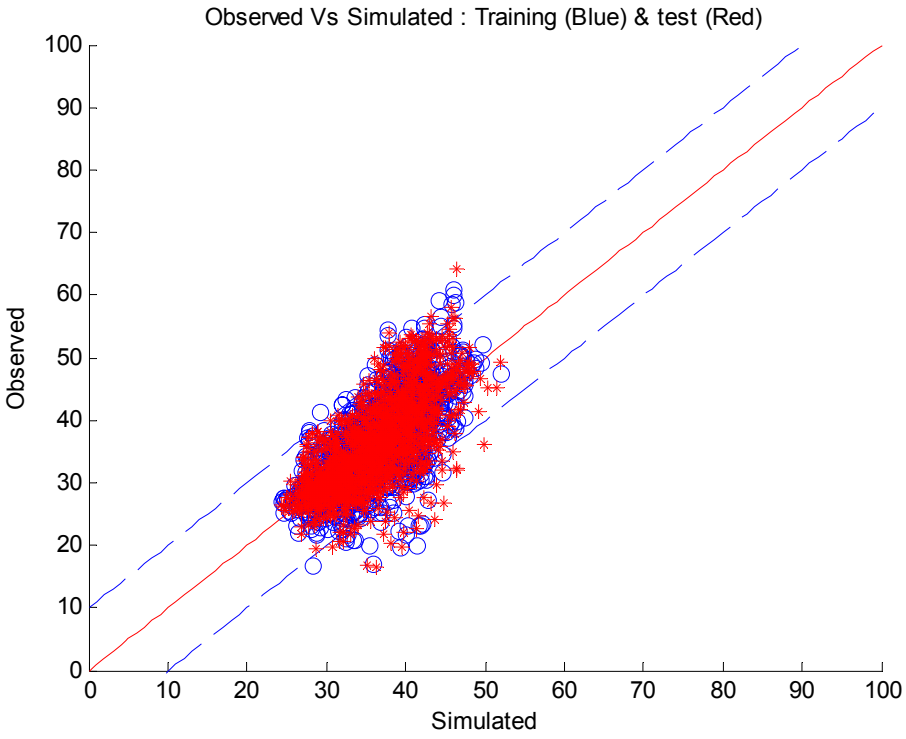


Figure 46: Ice Pixels Observed and simulated reflectance

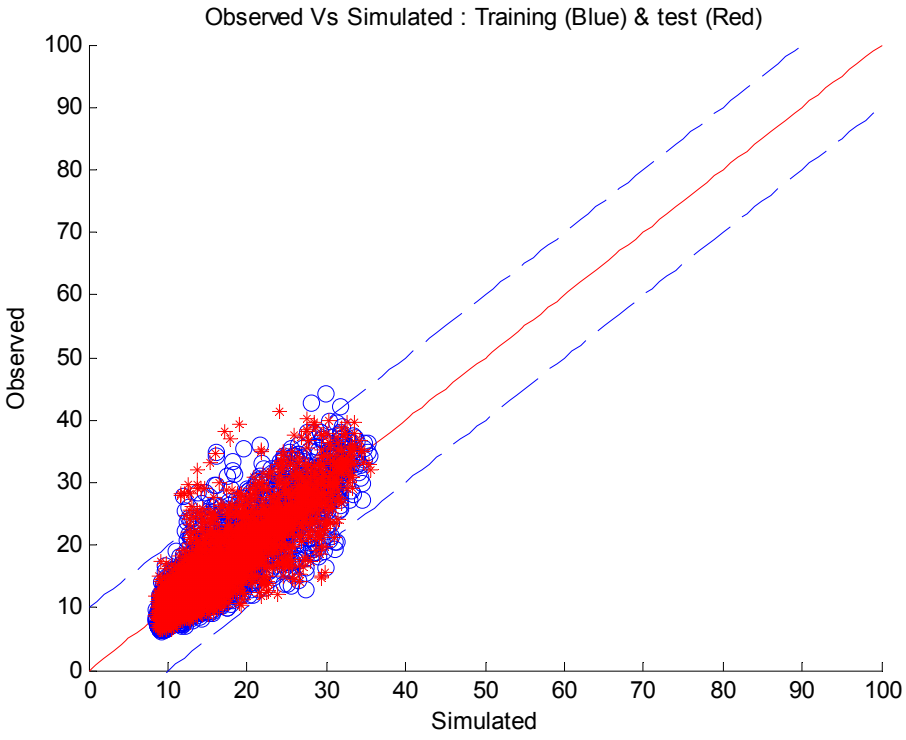


Figure 47: Water Pixels Observed and simulated reflectance

The tabulated result of the RMSE of ice and water is presented in Table 10.

Table 10: Simulation performances and RMSE values

	RMSE	
	ICE	Water
Training	5.09%	3.38%
Validation	5.35%	3.63%
Testing	5.22%	3.51%

After running the model for the Caspian Sea region, the following sea ice maps have been generated which confirms the accuracy of the method. Figure 48 show the classified images, were ice, water, cloud and land has been color coded and forms the hourly ice map.

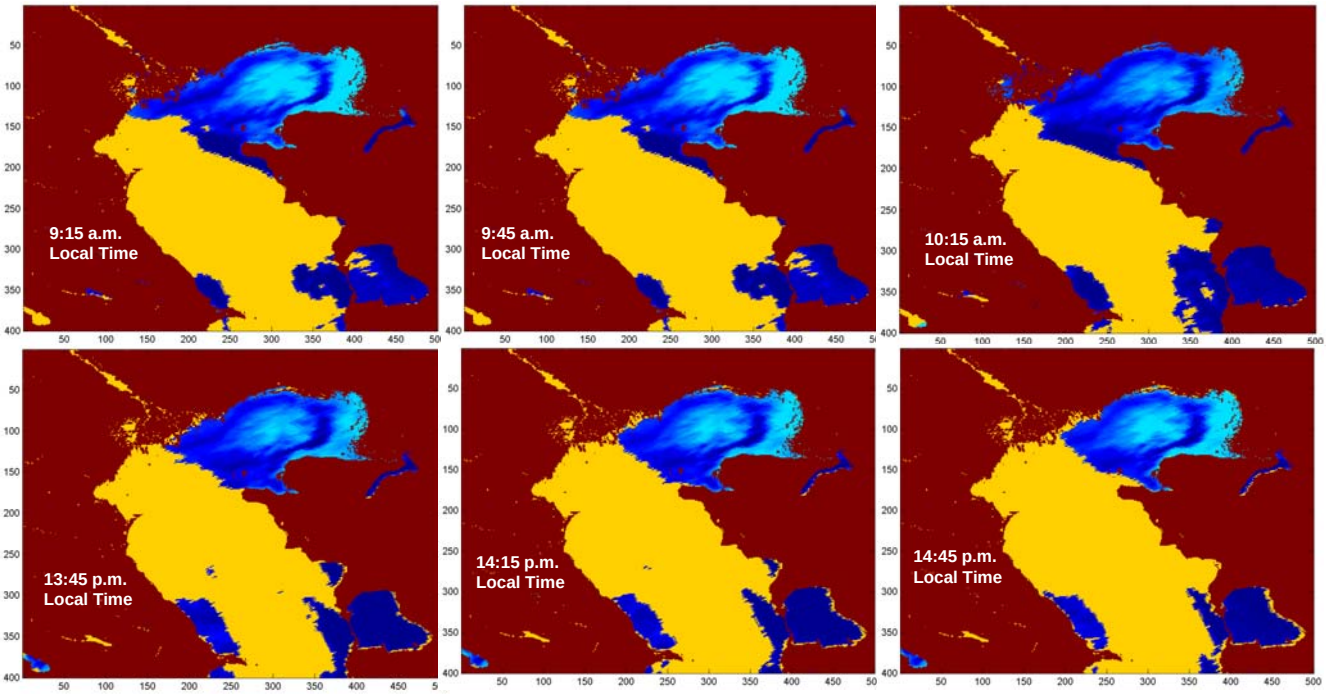


Figure 48: Neural Network base hourly ice map with cloud (Ice=light blue, water=blue, cloud= yellow, land=red)

After collecting all the hourly images the model could go through the pixels and look for cloud free conditions. At the end of the day, the collection of all cloud free pixel will be put together to form the daily ice map. The daily ice map generated has all the four classes of ice, water, cloud and land which in other word can be called daily ice map with cloud. Figure 49 shows the daily ice map with cloud coverage. It can be seen that the number of cloudy pixels have been significantly reduced due to the fact that some of these pixels have been cloud free at least once during the day, thus they are being classified based on the presented properties of that cloud free condition.

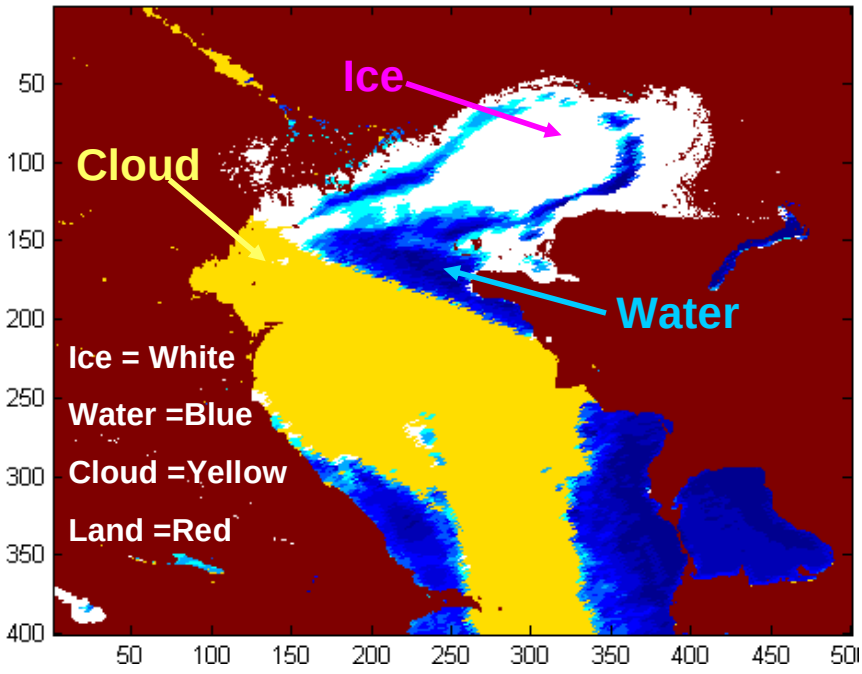


Figure 49: Neural Network base Daily Ice map with cloud

Because of the high presence of convective clouds in Caspian Sea region, having daily cloud free maps solely based on one day of data is very difficult and almost impossible. Moreover, it is rare to have a clear sky condition over the Caspian Sea for all pixels during a winter day, with some rare exceptions. Figure 50 shows the multi day image composited sea ice map of Caspian Sea which represents the three classes of ice, water and land. These cloud free images are the collection of the cloud free pixels of the day before and after whichever is closer to that time and date. Indeed, by processing all reflectance pixels collected over the Caspian Sea between December 2006 and February 2007, the average time between two clear sky conditions is 2.85 days. However, the use of night brightness temperature images has reduced the time gap to 2.25 days and decreased the number of unclassified pixels (cloudy pixels) by 10 to 30% for most days.

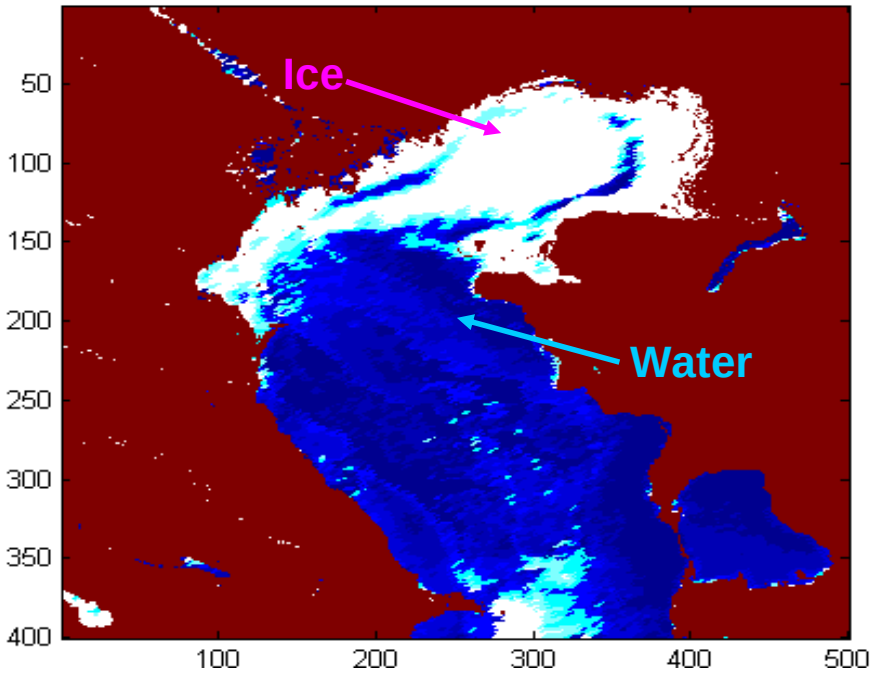


Figure 50: Neural Network (NN) base multi day image composited cloud-free ice map

Other sources of confusion of water reflectance could be related to water property factors such as high concentration in sediments, presence of river deltaic deposits and presence of fractional ice. These sources of confusion are not introduced by the viewing geometry effects and cannot be corrected by a regular bidirectional reflectance distribution function model (BRDF). Additionally, the random and dynamic aspects of these sources of confusion make their parameterization and introduction into a modified BRDF model difficult. Therefore there is a need for the development of a method that could determine the ice concentration and detect the fractional ice. Satellite observations in the visible and infrared spectral bands have been used for ice mapping (Kwok et al. 1995). A widely used approach to the atmospheric correction of satellite observations over water consists in the use of physically-based models which explicitly account for the Rayleigh and aerosol scattering as well as water vapor, ozone and other atmospheric gases absorption. In this study, in order to accurately detect sea ice, it is essential to calculate the Ice Fraction (Fi). With the purpose of

doing so, other parameters such as the observed reflectance (Robs), the water reflectance (Rw) and the ice reflectance (Ri) are also need which has to be computed thru the neural network model. After obtaining the key parameters, the following equation has been derived to algebraically compute Ice Fraction (IF).

$$F_i = \frac{R_{obs} - R_w}{R_i - R_w}$$

The result of this calculation will lead to the determination of ice fraction or concentration which is a very valuable parameter for climate studies and navigation purposes. Ice concentration values have been map and presented in Figure 51.

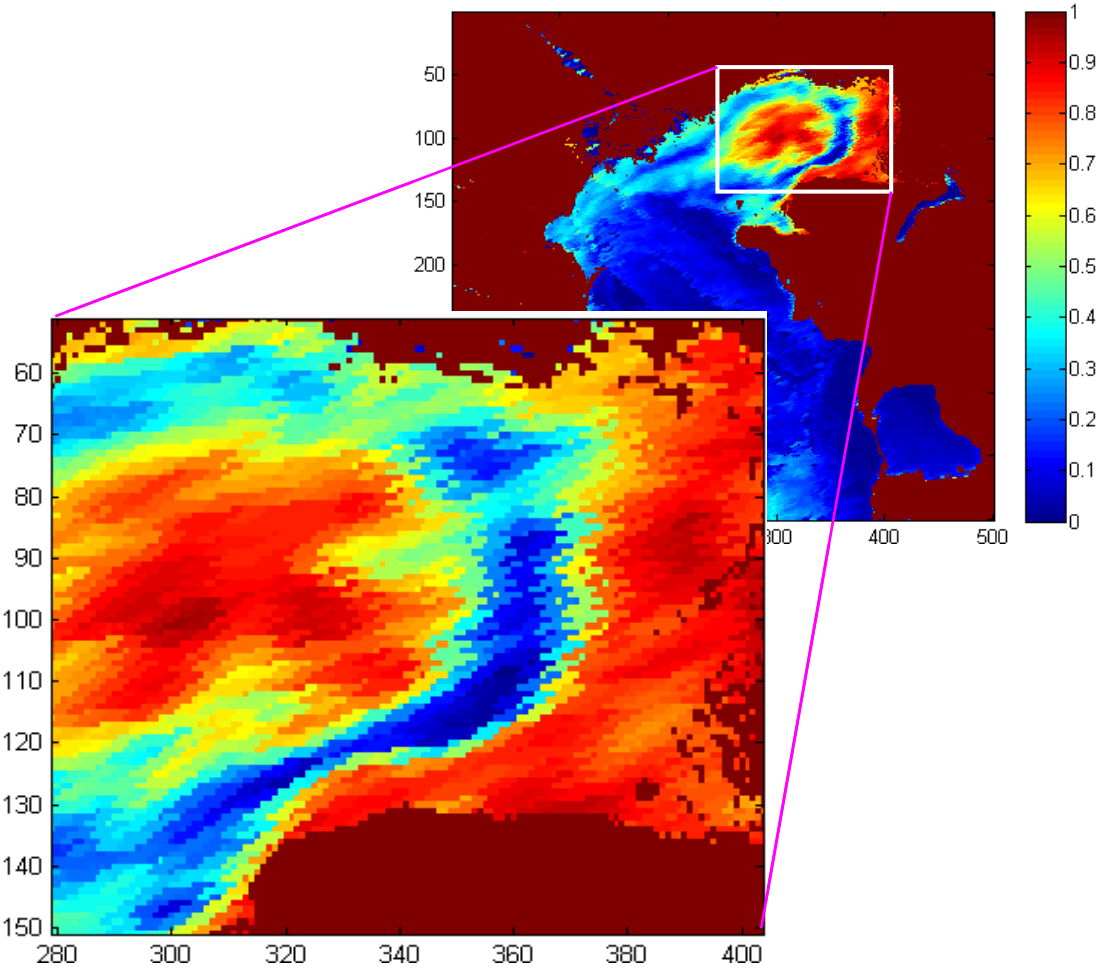


Figure 51: Neural Network ice concentration mp

Figure 51 illustrates this neural network based ice concentration map which has used the above mentioned formula. The Figure indicates that each pixel has a numerical value between 0-1 with 0 being no ice and concentration increases toward 1. As it can be seen in the Figure 51 the highest ice concentration can be found in the northeastern region of Caspian Sea.

Ice concentration could define how dense the ice coverage is in a target area. This parameter has helped with the sea ice classification model and since it can generate numerical values, the ice concentration of different regions can be compared and assessed for the study of the ice thickness.

6 CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

Other sources of confusion are the variation of water reflectance due to factors related to water properties such as high concentration in sediments, presence of river deltaic deposits. These sources of confusion are not introduced by the viewing geometry effects and cannot be corrected by bidirectional reflectance distribution function model (BRDF). Additionally, the random and dynamic aspects of these sources of confusions make their parameterization and introduction into a modified BRDF model difficult. Fractional ice has been added as an additional class to the final product. However, the lack of truth data and in situ information increase the risk of confusion of fractional ice with other water properties. The temperature difference between the water and the land surface create a convective condition over the Caspian Sea that generate a frequent and thick cloud coverage which makes the production of timely classification of ice and increase of temporal resolution of the final product difficult. Indeed, by processing all reflectance pixels collected over the Caspian Sea between December 2006 and February 2007, the average time between two clear sky conditions is 2.85 days. However, the use of night brightness temperature images has reduced the time gap to 2.25 days and decreased the number of unclassified pixels (cloudy pixels) by 10 to 30% for most days. Other logic-based tests can be applied to classify cloudy pixels to either ice or water using previous classifications and simulations of sea ice dynamics. In such case, quality control flags for pixels classified with the multi-day are provided based on the number of days used to make decision. In this research, the SEVIRI instrument onboard Meteosat Second Generation (MSG) satellite was used as a prototype for the future GOES-R

ABI. The rate of observations from SEVIRI (one image per 15 minutes) is the same as for GOES-R ABI. A neural-network-based model has been used to simulate water and ice reflectances over the Caspian Sea. Pixel geometry defined by the three solar, azimuthal and satellite angles were the primary input to the model. Trigonometric functions such as sine and cosine have also been considered. A fine tuning exercise allowed us to select an optimal structure of the used neural network. The developed ice detection and mapping algorithm have been applied to MSG SEVIRI data and have been tested over the Caspian Sea. The algorithm has been assessed using observed reflectance values. The obtained acceptable results have shown that a neural network based BRDF model has an interesting potential for ice mapping and monitoring from geostationary platforms. The main objectives of this research have been met through the production of the promised products. The developed algorithms are capable of generating the following products:

- Hourly Ice Map
- Daily Ice Map with Cloud
- Daily Cloud-free ice map (multi-day image composited approach)

Comparison of the final products with the Interactive Multisensor Snow and Ice Mapping System (IMS) maps show that the models have high potentials in sea ice mapping and the produced map show better resolution and have coarse pixels.

Figure 52 compares the model generated sea ice map with the IMS maps. Models sea ice map has a better accuracy than the IMS map and also a coarser pixel representation.

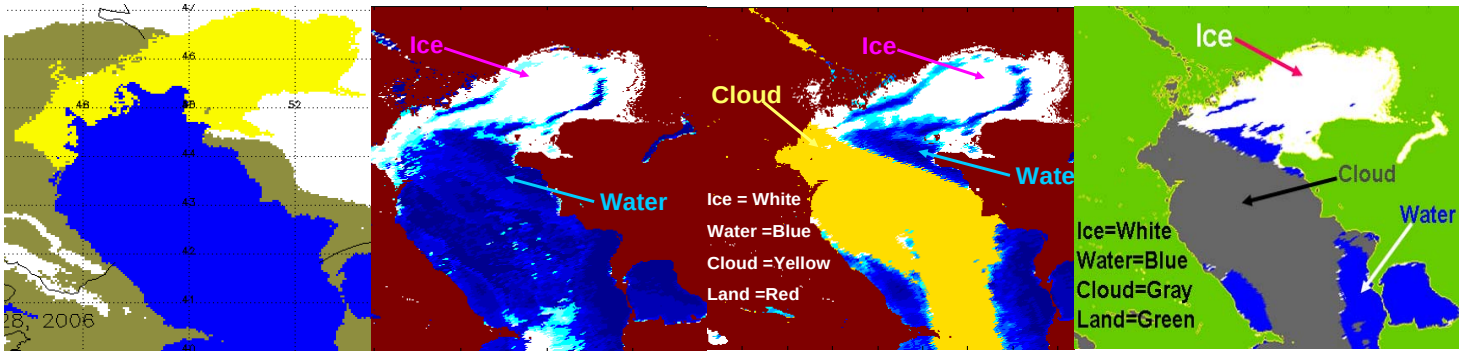


Figure 52: IMS Map and models generated sea ice maps of February 28th, 2008 of Caspian Sea

In addition to the models, in order to check the models' results, a model quality control station was built to continuously check the pixels value and the mapping trend at any given time, location for possible errors.

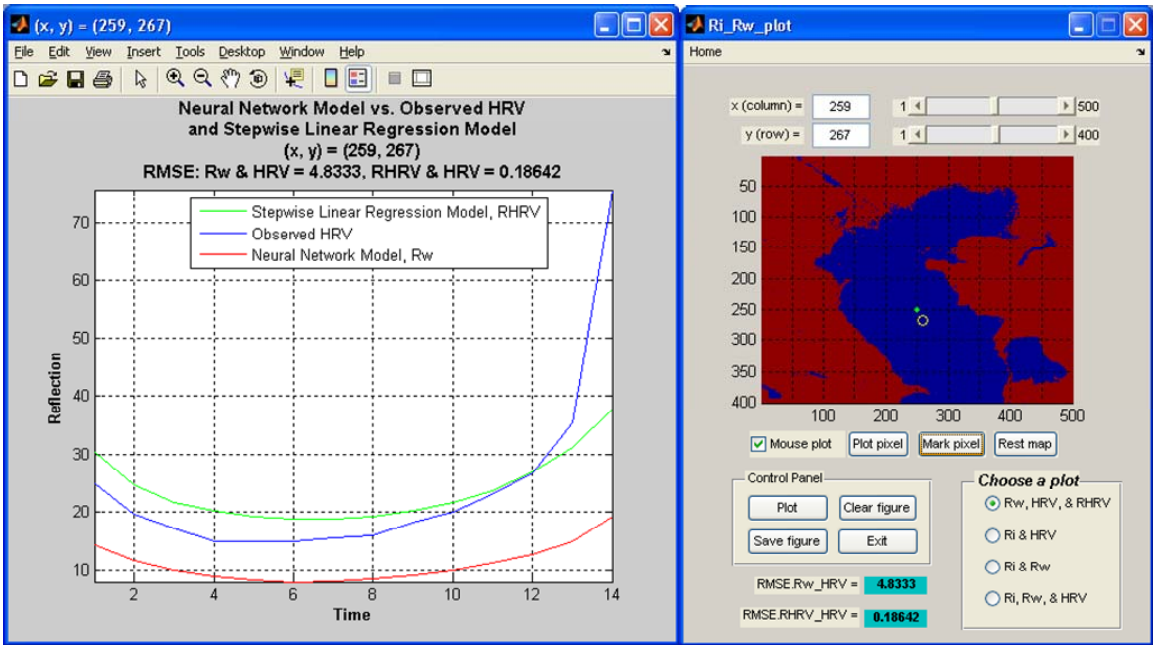


Figure 53: Models quality control station

As it could be seen in the Figure 53, the results of stepwise linear regression model and neural network model are being compared with observed band value and RMSE of each

model is recorded. The quality control station allows the user to check the accuracy of the simulated models at anytime and could report the miss classifications, check for errors and keep track clouds in the system. This feature could be seen in Figure 54. This Figure illustrates the daily pattern of the pixel value for a given day. The models are showing the normal pattern of the reflectance changes but the observed value shows a spike which means that an object with high reflectance is being detected for this given time and location and since the terrain will not significantly change in that time span that informs us of the presence of cloud at that acquisition.

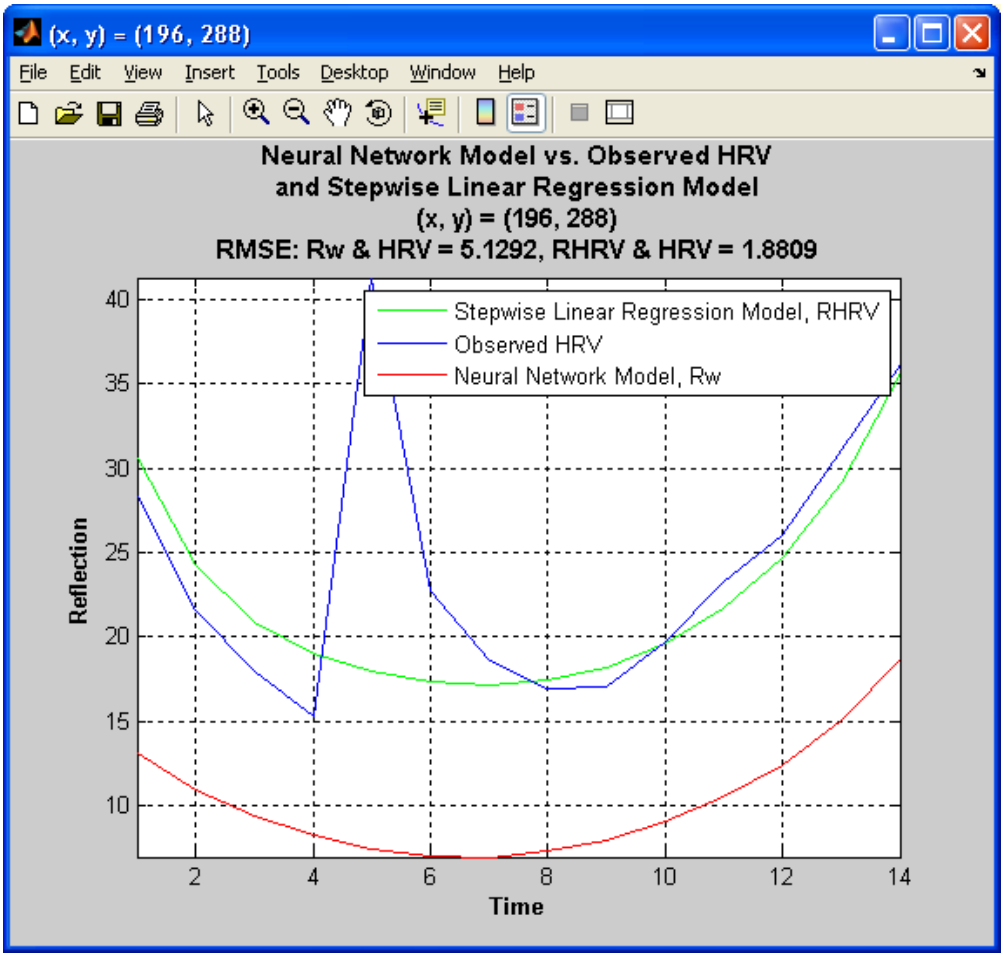


Figure 54: Model quality control station observed a sudden reflectance change

The control station also allows for other comparisons such as the comparison Neural Network (NN) simulated ice and water pixel of at any given time and location. Figure 55 shows the daily variation of these two classes.

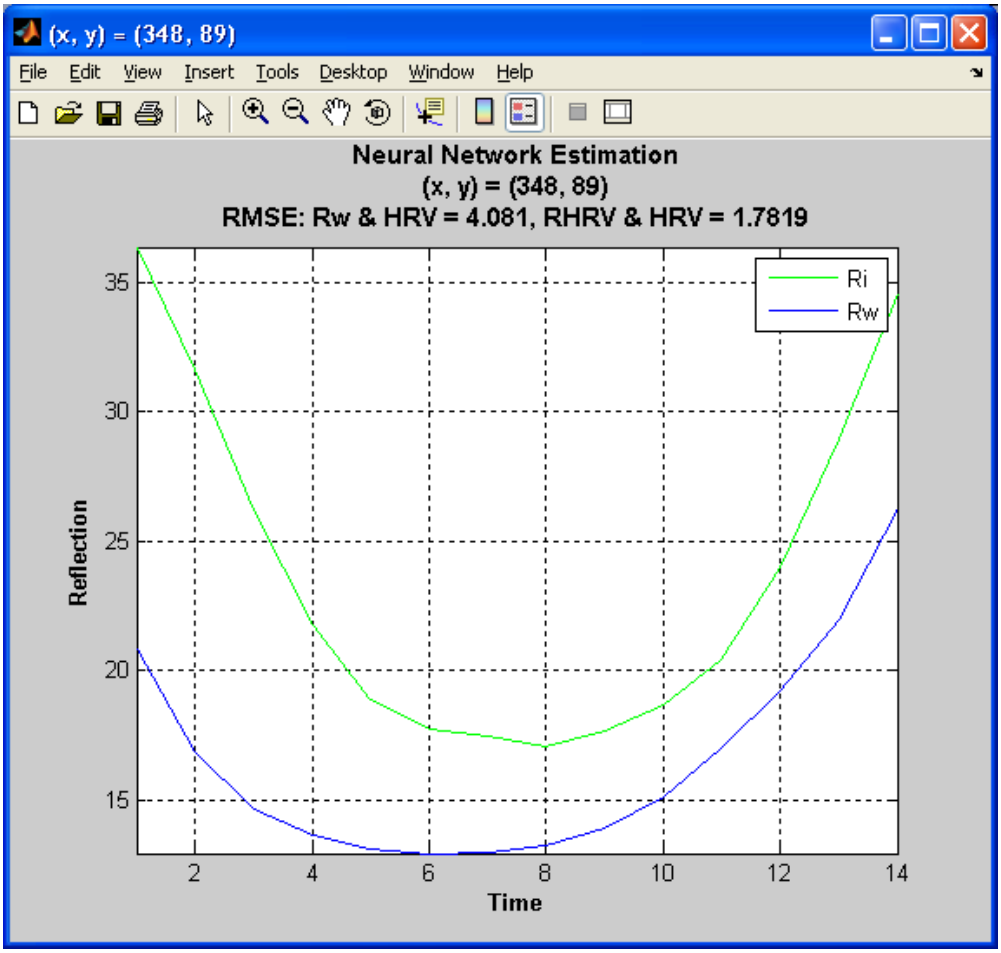


Figure 55: Neural Network simulation of Ice and water pixel

As Figure 55 shows, the reflectance of ice during the day is much higher than reflectance of water and this is being confirmed with the model and the figure which follows the same path. Ri (ice reflectance) has a higher average reflectance than Rw (water reflectance) during the day and also the effect of high solar angel in sunrise and sunset is enhanced on ice due to the reflective nature of ice.

During the course of observation it was noticed that the reflectance values are the highest in the beginning and the end of the day and the lowest values are reached at local noon. As the Figure 55 indicates, the highest reflectance values differ from the lowest by 10% approximately. This variation due to solar illumination angel is the primary reason for the failure of the constant daily threshold. Consequently instead of having one constant daily threshold, the dynamic threshold was developed and applied. Dynamic threshold is able to distinguish ice and water throughout the day even with the broad range of reflectance in these two classes and the number of misclassified pixels has substantially decreased and the produced images are clean. In addition the dynamic threshold added certainty to the classified pixels and has helped the classification model to generate a coarser ice and water pixels. In conclusion, it can be mentioned that the dynamic threshold is a very useful tool in the mapping and classification process to minimize any influence from the Satellite Viewing Angel (SAT) or Solar Angel (SOL).

The developed parameters have been tested and validated. These parameters form the sea ice algorithms that are producing the objected maps. The threshold base model and the stepwise model, threshold based classification model and the neural network model are capable of delivering the products with high accuracy. Alongside of these models, the ice concentration model can produce pixel values which can point to the existing ice content of the pixel.

One of the principle motivations of this project is to contribute to the ongoing GOES-R satellite research and preparation which is scheduled to lunch in 2015. The sea ice algorithm will produce the first sea ice map from a U.S. optical geostationary satellite and is the pioneer of sea ice mapping and monitoring from GEOS-R. The sea ice algorithm will open a new era in the capabilities to mapping systems, prevention of costly hazards and climate

studies. Furthermore, high temporal resolution sea ice maps have important applications in the navigation and marine transport, fishing and the economy of the ice covered regions. Sea ice maps and information is beneficial to industry, governments, citizens, and society in a number of in many ways. Sea ice is as an important contributor to the better prediction of weather and climate and having access to long term measurements and archive of imagery and sea ice maps and products has a wide range of applications for large-scale global change climate studies.

7 SUGGESTED FUTURE WORK

- Correction of satellite-observed reflectance for atmospheric effects and sea surface reflectance anisotropy.
- BRDF models development and validation for different date and time.
- Assessment of the effect of variability in water properties (higher reflectance due to sediments, river deltaic deposits, shallow water, etc.).
- Determination of the spectral reflectances for different types of sea ice to add other classes to the classification model
- Testing and validating the developed technique over seas and large lakes in Europe that are getting seasonal ice cover (Baltic Sea, Gulf of Finland)
- Preparation of the technical documentation for all developed algorithms
- Preparation of the developed software for operational implementation at NESDIS

8 APPENDICES

8.1 APPENDIX A

8.1.1 Stepwise Multiple Regression Analysis

a) Channel 2- (0.6)

Table 1 Statistical results of the Stepwise Multiple Linear Regression calibration for channel 2 (R01)

Set	Training Size	Train. Range	RMSE (Train.)	Intercept	Rsq	AdjRsq	Fstat	Coefficient #
R01-1	3000	2-3001	40.7649	805592000.00	0.702992	0.701399	441.28	16
R01-2	3000	3002-6001	42.3452	-12327700.00	0.704625	0.702842	395.07	18
R01-3	3000	6002-9001	41.2608	1137430000.00	0.70833	0.706372	361.73	20
R01-4	3000	8790-11789	39.9532	815141000.00	0.712586	0.711045	462.235	16
R01-5	5000	2-5001	40.6957	2297970000.00	0.715039	0.713664	520.147	24
R01-6	5000	5002-10001	40.5875	1058360000.00	0.71696	0.715709	573.049	22
R01-7	8000	2-8001	41.0478	7045010.00	0.713082	0.712291	901.155	22
R01-8	8000	3790-11789	40.5925	4889570.00	0.715535	0.714715	872.288	23

Table 2 Coefficients (b_j) for R01 models estimated by the Stepwise Multiple Linear Regression calibration

Set	B [a]	C [b]	D [c]	Cos B	Cos C	Cos D	Sin B	Sin C	Sin D	1/ CosB	1/ CosC
R01-1	-2438.8	-469283634.0	15641.8	0	-404760232.6	7105.7	0.0	0	0	0	0
R01-2	2247.6	7078676.9	-577.6	2160.3	6180483.3	0.0	-617.6	0	0	0	0
R01-3	2046.3	-660527990.9	45958.6	2003.0	-569896846.7	49293.5	-611.8	0	0	0	0
R01-4	-1348.1	-474334908.2	9949.0	702.1	-409172337.4	6897.8	0.0	0	0	0	0
R01-5	2612.6	-1331062280.8	-1745790.3	2622.8	-1148408130.6	-2115074.3	-904.6	0	-659010.1	0	0
R01-6	-178595.5	-614868291.6	0	2213.4	-530327276.2	66162.4	-682.3	0	-4615.8	0	0
R01-7	2541.0	-5200608.7	-233092.2	2614.9	-4062172.9	93974.7	-860.0	0	-19234.2	0	0
R01-8	-22157.5	-3646639.7	-191394.2	2421.5	-2779127.5	85321.2	-770.0	0	-17451.8	0	0

1/CosD	1/ Sin B	1/ Sin C	1/Sin D	B*C	B*D	C*D	B/C	B/D	C/D	C/B	D/B	D/C	1/B
0	0	0	14620.5	0	0	-9901.9	2298.6	0	-11152.6	0	0	0	0
593.9	0	0	0	0	-1385.2	0	0	0	0	0	0	0	0
0	0	0	11909.9	0	-1366.2	-2515.2	0	0	-5625.2	0	0	0	0
0	0	0	10716.2	0	0	-4500.4	0	0	-7953.1	0	0	0	0
-3201750.0	0	0	0	0	-1447.6	52527.4	0	-91.7	0	0	0	33479.3	0
0	0	0	0	84315.0	-1345.2	41421.3	101908.9	0	4289.5	0	0	0	0
0	0	0	-34289.2	0	-1425.2	201034.2	0	-68.0	32101.9	0	0	56170.3	0
0	0	0	-29396.4	11663.0	-1319.7	172170.7	13388.3	0	28084.3	0	0	44515.1	0

1/C	1/D	B/ CosB	B/ CosC	B/ Cos D	C/ Cos B	C/ CosC	C/ CosD	D/ CosB	D/ CosC	D/ CosD	CosB * CosC	CosB * CosD	CosC * CosD
-120361003.7	0	0	194.2	0	0	0	162.9	0	0	0	0	0	0
1908227.6	0	0	285.7	0	0	0	-451.8	0	640.8	0	-2590.1	2353.5	0
-172250386.0	0	0	272.4	0	0	0	0	0	0	0	-3162.6	1953.8	0
-122337711.5	0	0	340.9	0	0	0	0	0	0	0	-3807.7	0	0
-347450825.8	0	0	293.5	0	0	0	0	0	0	2038296	-3287.8	1949.2	-34067.6
-160066628.1	0	0	-1683.8	0	0	0	0	0	0	0	-3462.2	1925.8	-42338.6
0	0	0	292.3	0	0	0	0	0	0	0	-3324.1	1946.6	-160988.6
0	0	0	0	0	0	0	0	0	0	0	-3327.5	2165.0	-138847.0

CosB*CosC *CosD	Sin2B	Sin2C	Sin2D	Cos2B	Cos2C	Cos2D	1/ Sin2B	1/Sin 2C	1/ Sin2D	1/ Cos2B	1/ Cos2C	1/ Cos2D
0	0	0	0	19.3	-5286463.2	0	0	410351.3	-419.9	0	0	0
-1213.6	-318.4	0	0	-157.6	0	-515.8	0	0	0	0	0	0
0	-254.5	0	-3767.6	-149.6	-5978752.7	0	0	517259.6	0	0	0	0
0	120.7	0	0.0	48.4	-4983598.9	0	0	398914.5	0	0	0	0
0	-312.8	0	235325.4	-229.1	-11865867.8	0	0	1032359.9	0	0	0	0
0	-256.8	0	-3499.0	-163.7	-5745124.7	0	0	493199.7	0	0	0	0
0	-316.4	0	0.0	-219.2	-755520.5	0	0	35544.2	0	0	0	0
-833.2	-287.9	0	0.0	-190.7	-538101.9	0	0	26626.0	0	0	0	0

[a] B stands for azimuth angle [b]C stands for satellite zenith angle [c] D stands for solar zenith angle

Table 3 Results of split-sample validation of the R01 models with test set sizes of one third of the training sets

Training Set	Training Range	Training Size	Test Range	Test Size [a]	R^2	RMSE (th)
R01-1	2-3001	3000	4002-5001	1000	-2.112	136.215
R01-2	3002-6001	3000	1002-2001	1000	0.348	62.476
R01-3	6002-9001	3000	9002-10001	1000	-1045.100	2459.223
R01-4	8787-11786	3000	7002-8001	1000	-20.094	364.891
R01-5	2-5001	5000	9002-10667	1666	-37.078	459.708
R01-6	5002-10001	5000	2002-3667	1666	-3622.549	4582.645
R01-7	2-8001	8000	8002-10667	2666	0.723	38.961
R01-8	3787-11786	8000	2-2667	2666	0.720	39.734

[a] Test size equals to one third of the training size

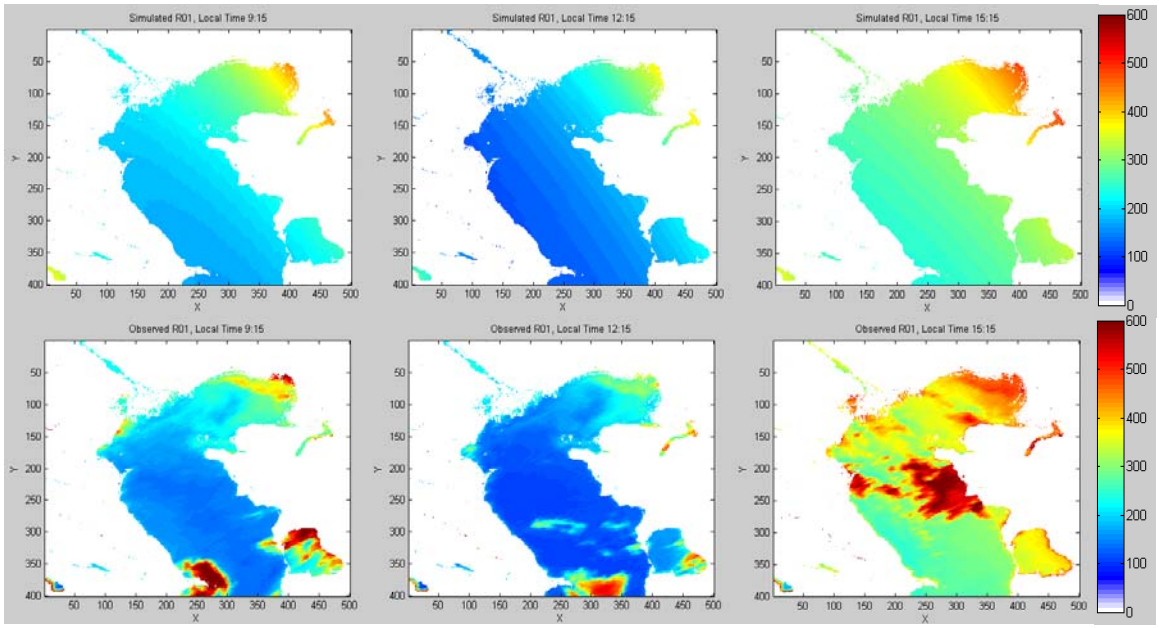


Figure 1 Observed (Bottom) and simulated (Up) R01 Reflectance (0.6 μm) at the local acquisition times of 8:45, 12:15 and 15:15, on January 23, 2007

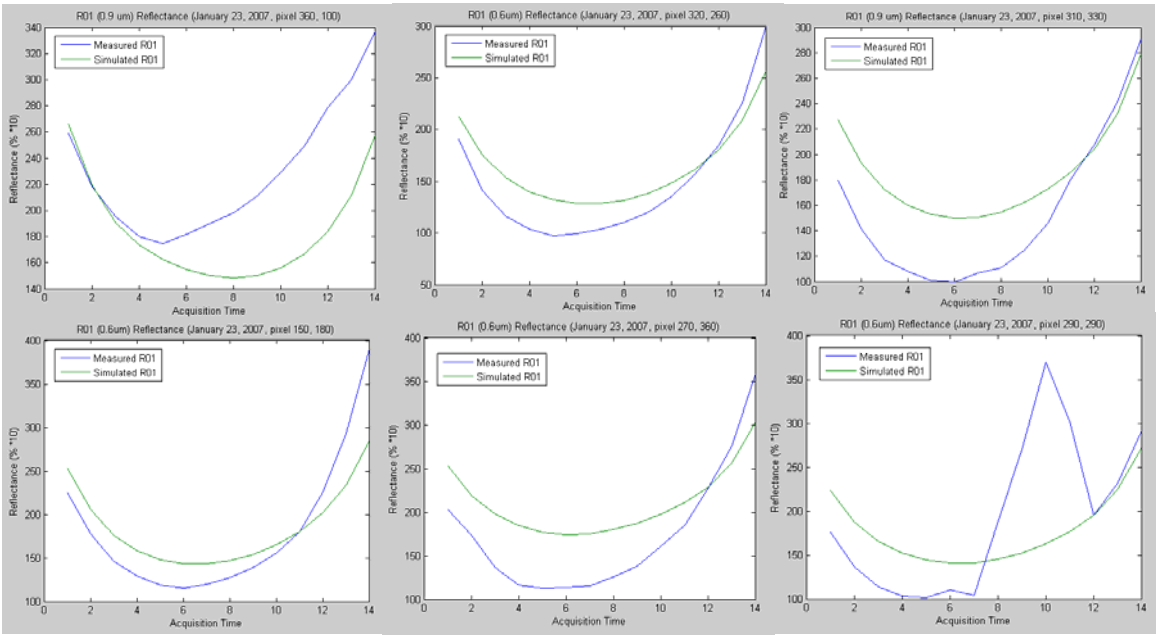


Figure 2 Observed vs. simulated R01 reflectance for pixel [360, 100], [320, 260], [310, 330], [150, 180], [270, 360] and [290, 290] on January 23, 2007

R01-8 =
(-22157.5211798621)*ARZ + (-3646639.68742517)* SAT + (-191394.158777069)*SOL + ...
(2421.4906296704)*cos (ARZ) + (-2779127.48564793)*cos (SAT) + ...
(85321.2237197447)*cos (SOL) + (-770.002926702377) * sin (ARZ) + ...

```
(-17451.7706028269) * sin (SOL) + (-29396.4474204779)*(1./sin (SOL)) + ...
(11663.0322436565) * ARZ .* SAT + (-1319.67538571718) * ARZ .* SOL + ...
(172170.659950775) *SAT.*SOL + (13388.2993375045) * ARZ./ SAT + ...
(28084.2685358272)*SAT./SOL + (44515.1024103154) * SOL ./ SAT +  ...
(-3327.54863644377)*cos (ARZ).* cos (SAT)+(2164.98713797329)*cos(ARZ).*cos
   (SOL)+...
(-138846.9568119)*cos (SAT).*cos (SOL) + ...
(-833.186144929572)* cos (ARZ) .*cos (SAT).*cos (SOL)  + ...
(-287.949051578011) *sin (2*ARZ) + (-190.700447957509)*cos (2*ARZ) + ...
(-538101.864547385)*cos (2*SAT) + ...
(26626.0061623959)*1./sin(2*SAT)+ (4889570);
```

b) Results - Channel 3

Table 4 Statistical results of the Stepwise Multiple Linear Regression calibration for channel 3 (R02)

Set	Training Size	Train. Range	RMSE (Train.)	Intercept	Rsq	AdjRsq	Fstat	Coefficient #
R02-1	3000	2-3001	46.1307	2.95E+07	0.550096	0.547683	227.956	16
R02-2	3000	3002-6001	50.6438	3.63E+09	0.513923	0.512297	316.023	10
R02-3	3000	6002-9001	49.4731	2.79E+08	0.502758	0.500927	274.65	11
R02-4	3000	8790-11789	44.1459	1.07E+09	0.572217	0.569345	199.24	20
R02-5	5000	2-5001	48.6284	3.95E+07	0.528485	0.527351	465.796	12
R02-6	5000	5002-10001	48.1948	2.99E+07	0.520241	0.5187	337.716	16
R02-7	8000	2-8001	47.7217	2.66E+07	0.549392	0.547753	335.077	29
R02-8	8000	3790-11789	47.0588	2.29E+07	0.545036	0.543839	455.117	21

Table 5 Coefficients (b_j) for R02 models estimated by the Stepwise Multiple Linear Regression calibration

Set	B [a]	C [b]	D [c]	Cos B	Cos C	Cos D	Sin B	Sin C	Sin D	1/ CosB
R02-1	0	-21229453.5	22818.6	0	-17802465.5	0	0	0	-8332.3	0
R02-2	0	-2110606633.0	-52440.4	0	-1820803349.1	17892.0	0	0	0	0
R02-3	0	-187852663.2	-27212.3	0	-172803170.9	0	0	0	0	0
R02-4	1767.8	-625017350.2	0	2396.0	-538976811.5	0	-820.2	0.0	37837.3	0
R02-5	-70.5	-35404608.3	0	0	-31512804.3	11861.8	0	15227719.4	0	0
R02-6	-1326.8	-26913667.4	-103046.0	0	-23853991.4	12942.8	309.2	11637995.2	0	0
R02-7	-171533.0	-24714980.8	-325519.6	2415.4	-21414498.5	222121.7	-788.7	10772340.3	48210.2	0
R02-8	2152.5	-20728624.0	-86673.5	2237.9	-18307387.1	51119.0	-701.8	9021205.6	-8515.6	0

1/ CosC	1/CosD	1/ Sin B	1/ Sin C	1/Sin D	B*C	B*D	C*D	B/C	B/ D	C/D	C/ B	D/ B	D/C
66683.2	-322.9	0	0	0	-1275.6	0	0	1048.6	0	0	0	0	-42868.7
0	24.4	0	0	0	0	0	43280.7	0	0	0	0	0	0
226137.3	51505.0	0	0	0	0	0	24813.4	0	0	0	0	0	0
0	0	0	0	13600.4	0	-1214.3	0	0	0	-5462.8	0	0	1310.9
74857.8	0	0	0	13324.8	0	0	0	0	0	-9355.4	0	0	10803.0
57807.2	20.8	0	0	0	0	0	58372.2	0	0	0.0	0	0	39016.5
55614.4	510633.0	0	0	-71098.8	81612.6	-1333.9	334065.8	97749.2	0	52767.5	0	0	98335.1
45511.5	0	0	0	0	0	-1331.4	74665.3	0	0	4473.4	0	0	46196.6

1/B	1/C	1/D	B/CosB	B/CosC	B/CosD	C/CosB	C/CosC	C/CosD	D/CosB	D/CosC	D/CosD	CosB * CosC	CosB * CosD	CosC * CosD
0	0	0	0	244.2	0	0	0	259.7	0	1810.9	0	0	0	-53894.7
0	-548860347.7	0	0	0	0	0	0	0	0	0	0	0	0	-51125.4
0	0	0	0	0	0	0	0	0	0	0	-32790.3	0	0	-28657.0
0	-160023432.3	0	0	381.7	0	0	0	0	0	0	0	-3424.7	2345.9	0
0	0	0	0	28.4	0	0	0	5.3	0	0	0	0.0	0	0
0	0	0	0	233.4	0	0	0	0	0	0	0	-2643.5	0	-37411.3
0	0	0	0	-1699.9	0	0	0	355.4	0	0	-325368.6	-2779.0	2324.4	-268080.2
0	0	0	0	305.3	0	0	0	0	0	0	0.0	-2808.1	2367.7	-44936.7

CosB*CosC *CosD	Sin2B	Sin2C	Sin2D	Cos2B	Cos2C	Cos2D	1/Sin2B	1/Sin 2C	1/Sin2D	1/Cos2B	1/Cos2C	1/Cos2D
0	0	0	4679.8	12.5	-3029599.3	0	0	0.0	0	0	0.0	0
0	0	0	0	0	-19898746.3	0	0	1684487.9	0	0	0.0	0
0	0	0	5313.9	0	-15257274.3	0	0	0.0	0	0	182634 7.7	0
-1704.9	-256.8	0	0	-204.0	-7216251.8	8689.3	0	555659.0	0	0	0	0
0	0	0	0	11.7	0	0	0	0	0	0	0	0
0	121.5	0	0	111.7	0	0	0	0	0	0	0	0
-1074.3	-331.0	0	0	-214.1	0	0	0	0	0	0	0	0
-1320.6	-305.1	0	0	-185.6	0	0	0	0	0	0	0	0

[a] B stands for azimuth angle [b] C stands for satellite zenith angle [c] D stands for solar zenith angle

Table 6 Results of split-sample validation of the R02 models with test set sizes of one third of the training sets

Training Set	Training Range	Training Size	Test Range	Test Size [a]	R^2	RMSE (th)
R02-1	2-3001	3000	4002-5001	1000	0.409	55.885
R02-2	3002-6001	3000	1002-2001	1000	-1100.082	2366.668
R02-3	6002-9001	3000	9002-10001	1000	-53.447	498.471
R02-4	8787-11786	3000	7002-8001	1000	-426.400	1529.921
R02-5	2-5001	5000	9002-10667	1666	0.399	50.996
R02-6	5002-10001	5000	2002-3667	1666	0.470	51.078
R02-7	2-8001	8000	8002-10667	2666	0.207	58.834
R02-8	3787-11786	8000	2-2667	2666	0.554934522	46.167319

[a] Test size equals to one third of the training size

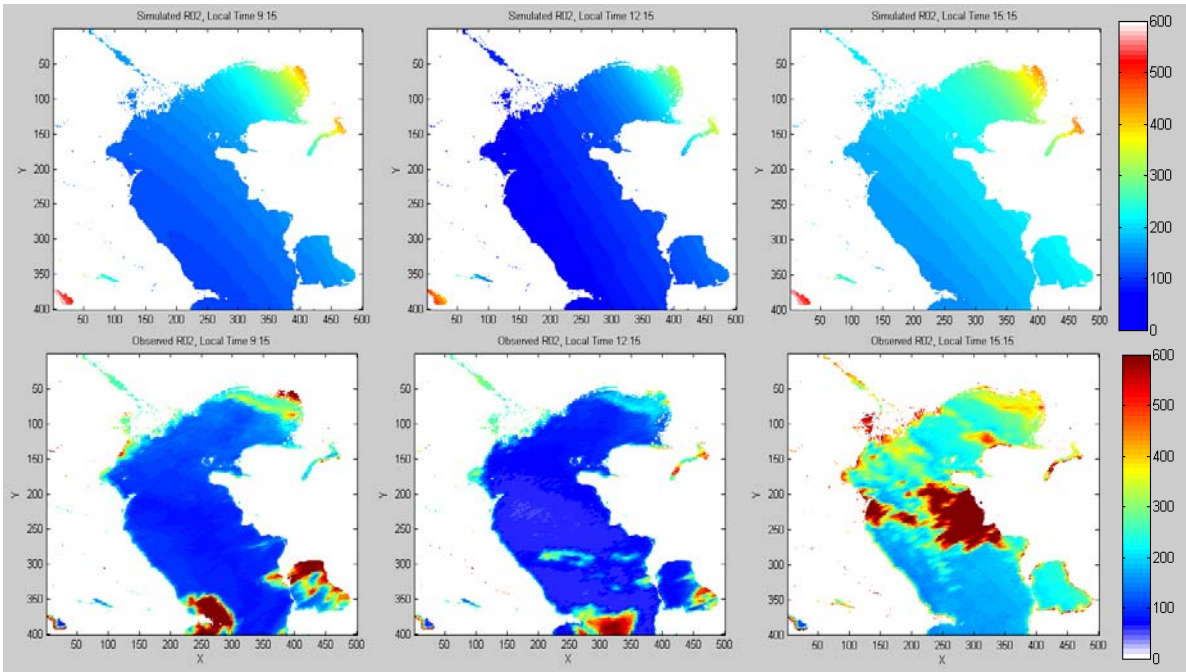


Figure 3 Observed (right) and simulated (left) R02 Reflectance (0.8 µm) at the local acquisition times of 8:45, 12:15 and 15:15, on January 23, 2007

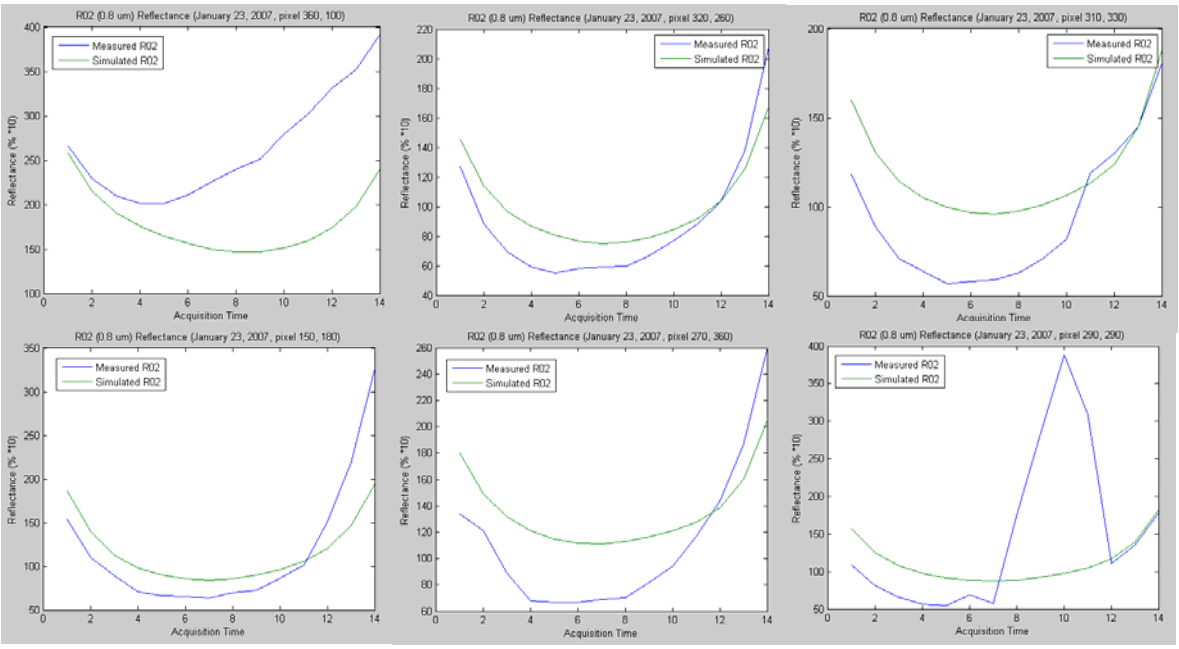


Figure 4 Observed vs. simulated R02 reflectance for pixel [360, 100], [320, 260], [310, 330], [150, 180] and [270, 360] and [290, 290] on January 23, 2007

R02-8 =

(2152.4680344739)*ARZ +(-20728623.9698639)* SAT + (-86673.4614278628)*SOL + ...

(2237.86072162681)*cos (ARZ) +(-18307387.0861547)*cos (SAT) + ...
(51118.9847558398)*cos (SOL) + (-701.804700097002) * sin (ARZ) + ...
(9021205.58144072) * sin (SAT) + (-8515.57276191785) * sin (SOL) + ...
(45511.4849349455)*(1./cos (SAT)) + (-1331.37173949849) * ARZ .* SOL + ...
(74665.312862983) *SAT.*SOL + (4473.41790094813)*SAT./SOL + ...
(46196.5599435315) * SOL ./SAT + (305.315312928979) *(ARZ./ cos(SAT)) + ...
(-2808.07917483622)*cos (ARZ).*cos (SAT) + (2367.6963137188)*cos (ARZ).*cos (SOL)
+ ...
(-44936.7046161723)*cos (SAT).*cos (SOL) + (-1320.64898016031)*cos (ARZ) .* cos
(SAT).*cos (SOL) + ...
(-305.100899306909) *sin (2*ARZ) + (-185.609236967029)*cos (2*ARZ) + (22914000);

c) Results - Channel 4

Table 7 Statistical results of the Stepwise Multiple Linear Regression calibration for channel 3 (R02)

Set	Training Size	Train. Range	RMSE (Train.)	Intercept	Rsq	AdjRsq	Fstat	Coefficient #
R03-1	3000	2-3001	20.7296	-226.845	0.155576	0.153318	68.8826	8
R03-2	3000	3002-6001	20.773	-0.903981	0.138983	0.137545	96.657	5
R03-3	3000	6002-9001	20.6185	1.78861	0.142357	0.141498	165.765	3
R03-4	3000	8790-11789	20.3711	8.28736	0.112311	0.111422	126.352	3
R03-5	5000	2-5001	20.6293	-275.536	0.152068	0.150368	89.4726	10
R03-6	5000	5002-10001	20.6368	-44.6167	0.137561	0.136698	159.311	5
R03-7	8000	2-8001	20.7127	-21.9605	0.144159	0.143731	336.672	4
R03-8	8000	3790-11789	20.5479	-349.611	0.135863	0.134889	139.58	9

Table 8 Coefficients (b_j) for R02 models estimated by the Stepwise Multiple Linear Regression calibration

Set	B [a]	C [b]	D [c]	Cos B	Cos C	Cos D	Sin B	Sin C	Sin D	1/ CosB	1/ CosC	1/ CosD	1/ Sin B	1/ Sin C
R03-1	0	0	122.45	0	0	0	0	0	0	0	0	0	0	0
R03-2	0	0	22.67	0	0	0	19.06	0	0	0	0	0	0	0
R03-3	0	0	24.63	0	0	0	0	0	0	0	0	0	0	0
R03-4	0	0	19.32	0	0	0	0	0	0	0	0	0	0	0
R03-5	0	0	166.81	0	0	0	-40.09	0	0	0	0	0	0	0
R03-6	0	0	51.36	0	0	0	0	0	0	0	0	0	0	0
R03-7	0	0	39.32	0	0	0	0	0	0	0	0	0	0	0
R03-8	0	0	234.54	0	0	0	0	0	0	0	0	0	0	0

1/ Sin D	B*C	B*D	C*D	B/C	B/D	C/D	C/B	D/B	D/C	1/B	1/C	1/D	B/CosB	B/CosC
0	0	0	0	0	20.03	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	77.85	-34.54	0	0	0	0	0	0	0	0
0	0	0	0	0	18.19	0	0	0	0	0	0	0	0	0
0	0	0	0	0	11.22	0	0	0	0	0	0	0	0	0
0	0	-13.10	0	0	88.79	0	0	0	0	0	0	0	0	0

B/Cos D	C/Cos B	C/CosC	C/CosD	D/CosB	D/CosC	D/CosD	CosB * CosC	CosB * CosD	CosC * CosD
0	0	0	2.96	0	12.49	0	0	129.32	399.15
0	0	0	2.00	0	0	0	0	-19.59	0
0	0	0	0	0	0	0	0	0	0
0	0	0	2.63	0	0	0	0	0	0
-4.79	0	0	3.57	0	15.63	0	0	174.77	337.28
0	0	0	2.12	0	0	0	0	14.86	0
0	0	0	2.16	0	0	0	0	0	0
-4.50	0	0	4.14	0	0	0	0	158.11	0

CosB*CosC *CosD	Sin2B	Sin2C	Sin2D	Cos2B	Cos2C	Cos2D	1/ Sin2B	1/ Sin 2C	1/ Sin2D	1/ Cos2B	1/ Cos2C	1/ Cos2D
-356.95	0	0	0	9.16	0	0	0	0	0	0	0	0
0	0	0	0	11.68	0	0	0	0	0	0	0	0
0	-4.50	0	0	0	0	0	0	0	6.89	0	0	0
0	-2.84	0	0	0	0	0	0	0	0	0	0	0
-192.67	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	8.12	0	0	0	0	0	0	0	0
0	0	0	0	6.44	0	0	0	0	0	0	0	0
0	-13.73	0	42.15	6.80	0	0	0	0	0	0	0	0

[a] B stands for azimuth angle [b] C stands for satellite zenith angle [c] D stands for solar zenith angle

Table 9 Results of split-sample validation of the R02 models with test set sizes of one third of the training sets

Training Set	Training Range	Training Size	Test Range	Test Size [a]	R^2	RMSE (th)
R03-1	2-3001	3000	4002-5001	1000	0.125	20.714
R03-2	3002-6001	3000	1002-2001	1000	0.144	21.297
R03-3	6002-9001	3000	9002-10001	1000	0.098	20.418
R03-4	8787-11786	3000	7002-8001	1000	0.161	20.252
R03-5	2-5001	5000	9002-10667	1666	0.117	20.131
R03-6	5002-10001	5000	2002-3667	1666	0.140	20.567
R03-7	2-8001	8000	8002-10667	2666	0.113	20.495
R03-8	3787-11786	8000	2-2667	2666	0.142	21.015

[a] Test size equals to one third of the training size

R03-3 =
(24.6388747996745)*ARZ + (-4.49788285206644) *sin (2*ARZ) + ...
(6.89291960413664)*1./sin (2*SOL) + (1.78861);

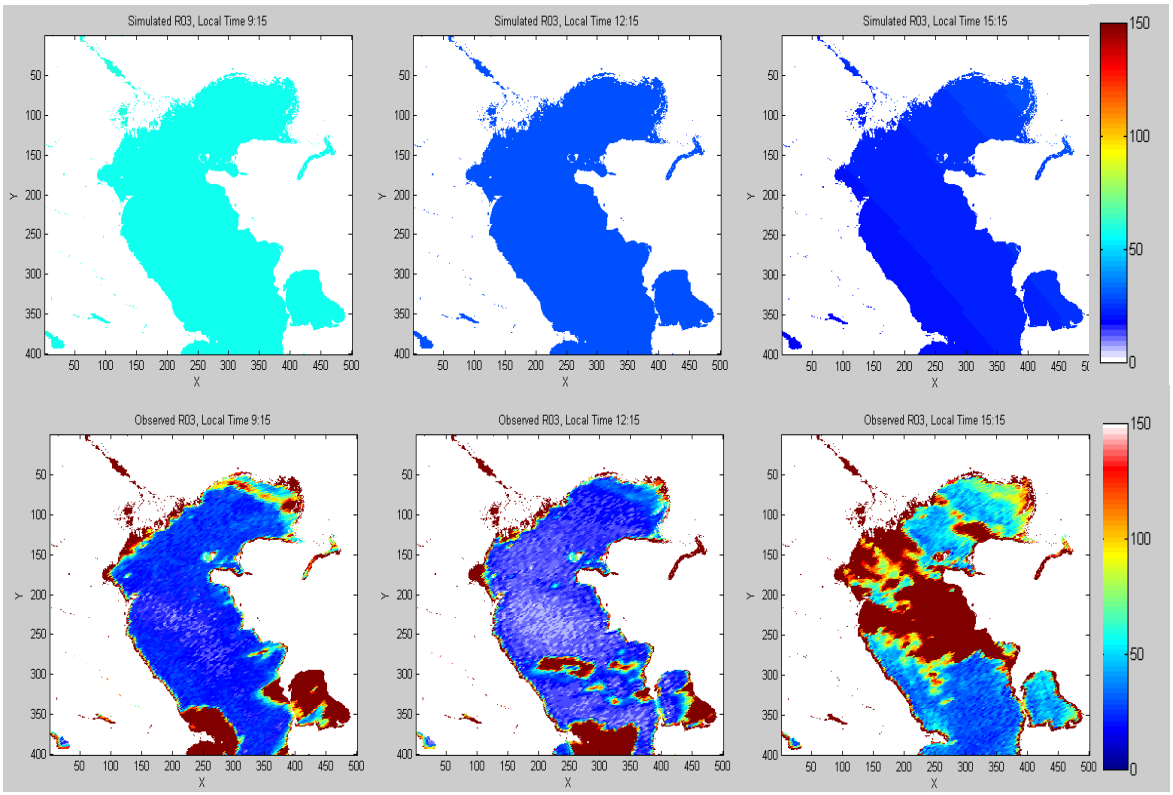


Figure 5 Observed (right) and simulated (left) R03 Reflectance (1.6 μm) at the local acquisition times of 8:45, 12:15 and 15:15, on January 23, 2007

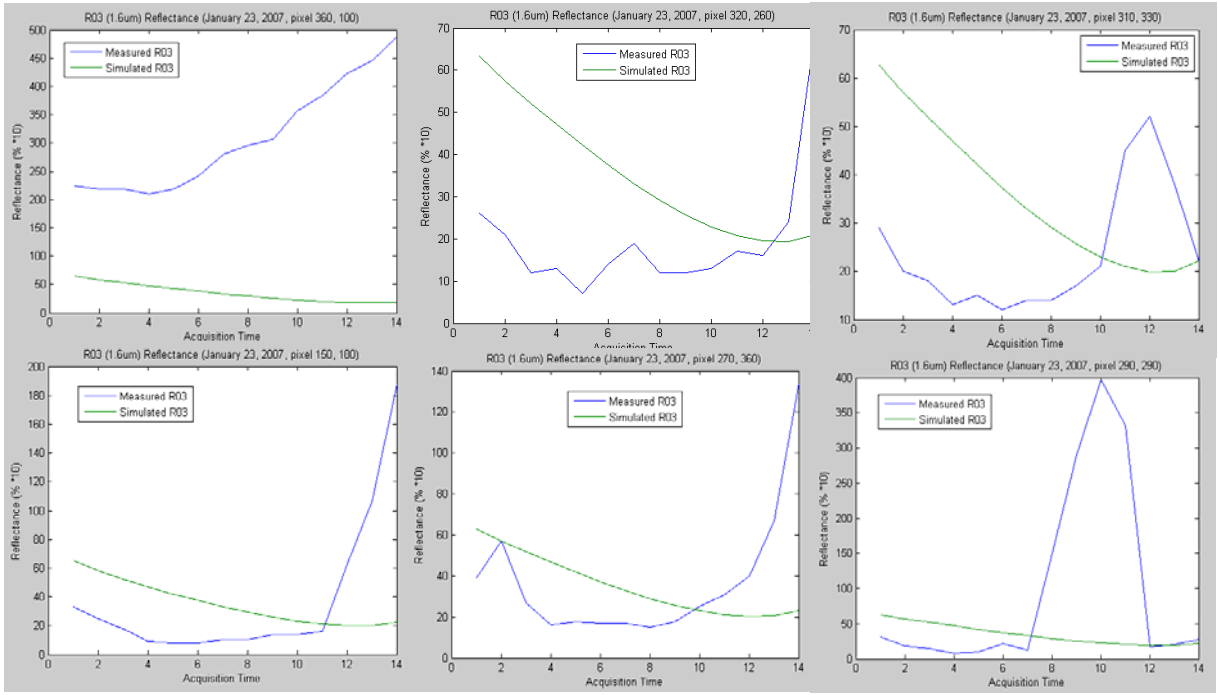
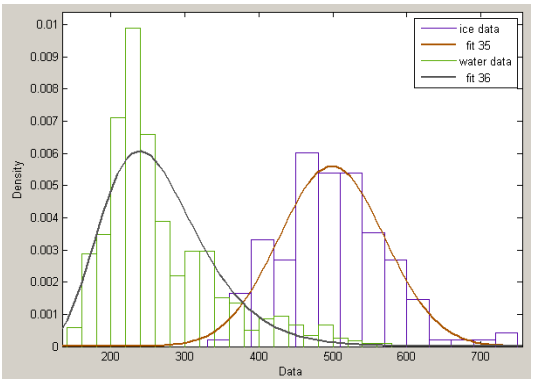


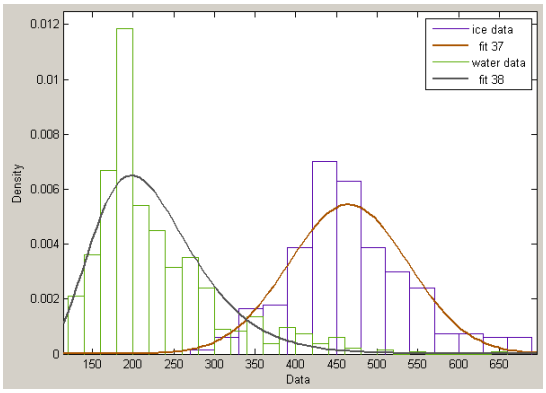
Figure 6 Observed vs. simulated R03 reflectance for pixel [360, 100], [320, 260], [310, 330], [150, 180], [270, 360] and [290, 290] on January 23, 2007

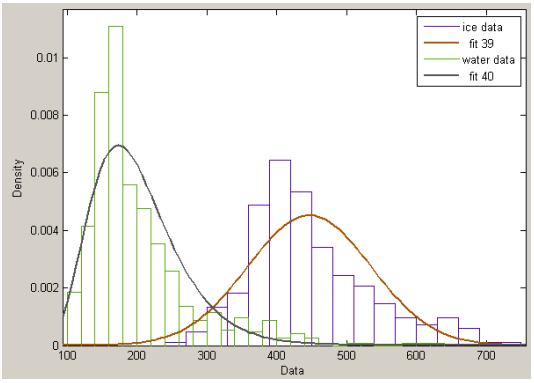
8.2 APPENDIX B

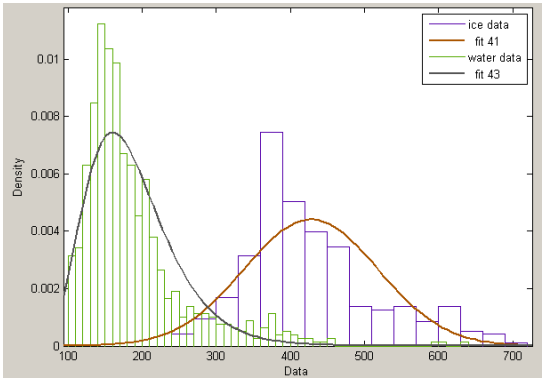
8.2.1 Dynamic Threshold Analysis

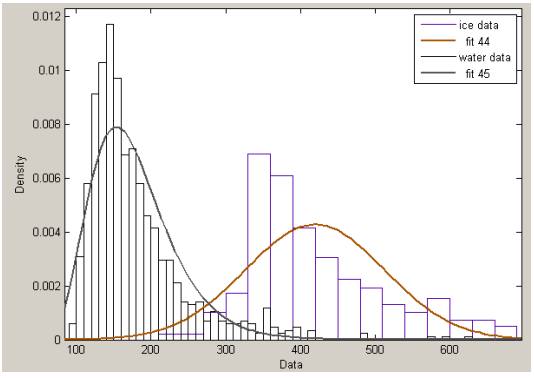
Figure A1. Distribution functions of water (left) and ice (right) reflectance in HRV channel for acquisition time 12 – 19 (from left to right)

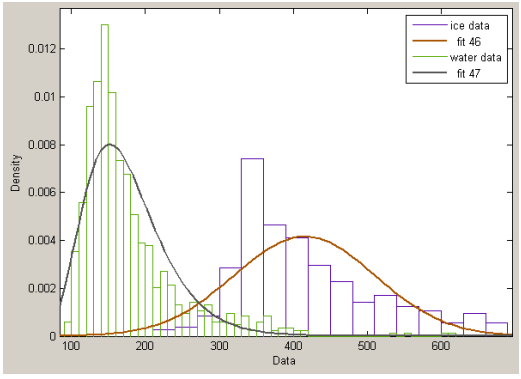


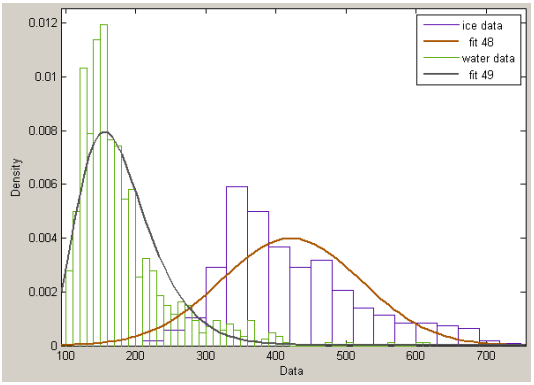












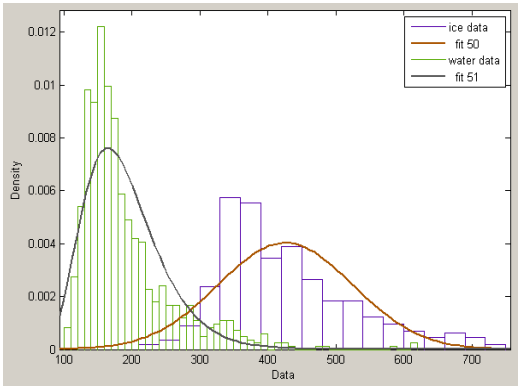
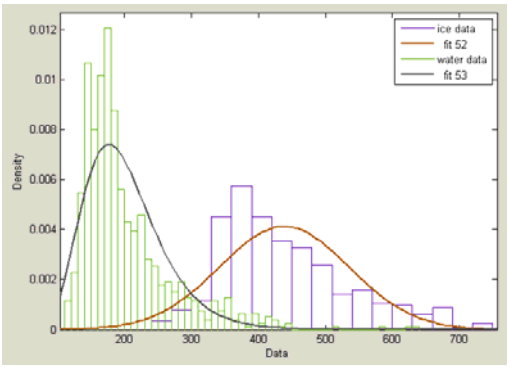
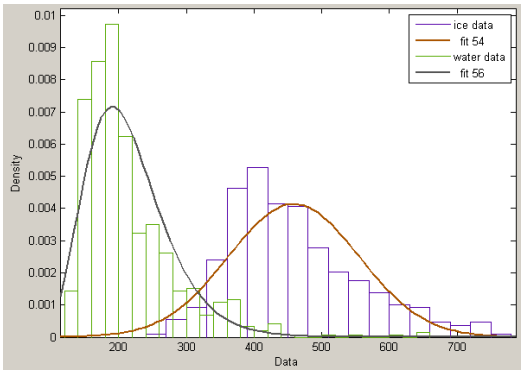
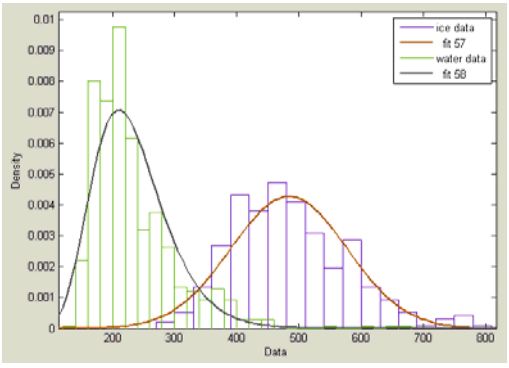
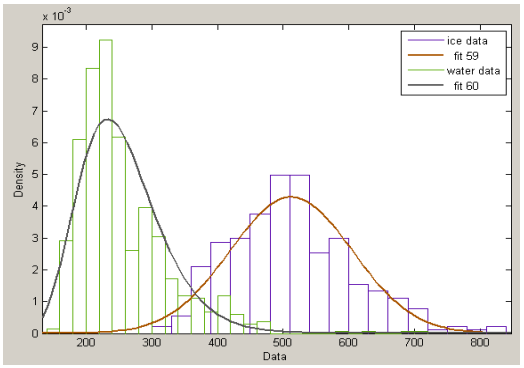


Figure A2. Distribution functions of water (left) and ice (right) reflectance in HRV channel for acquisition time 20 – 25 (from left to right)









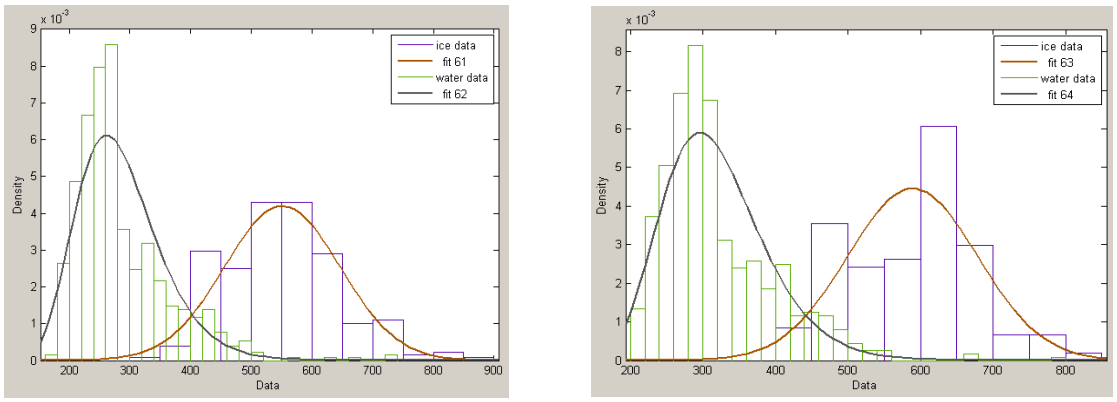
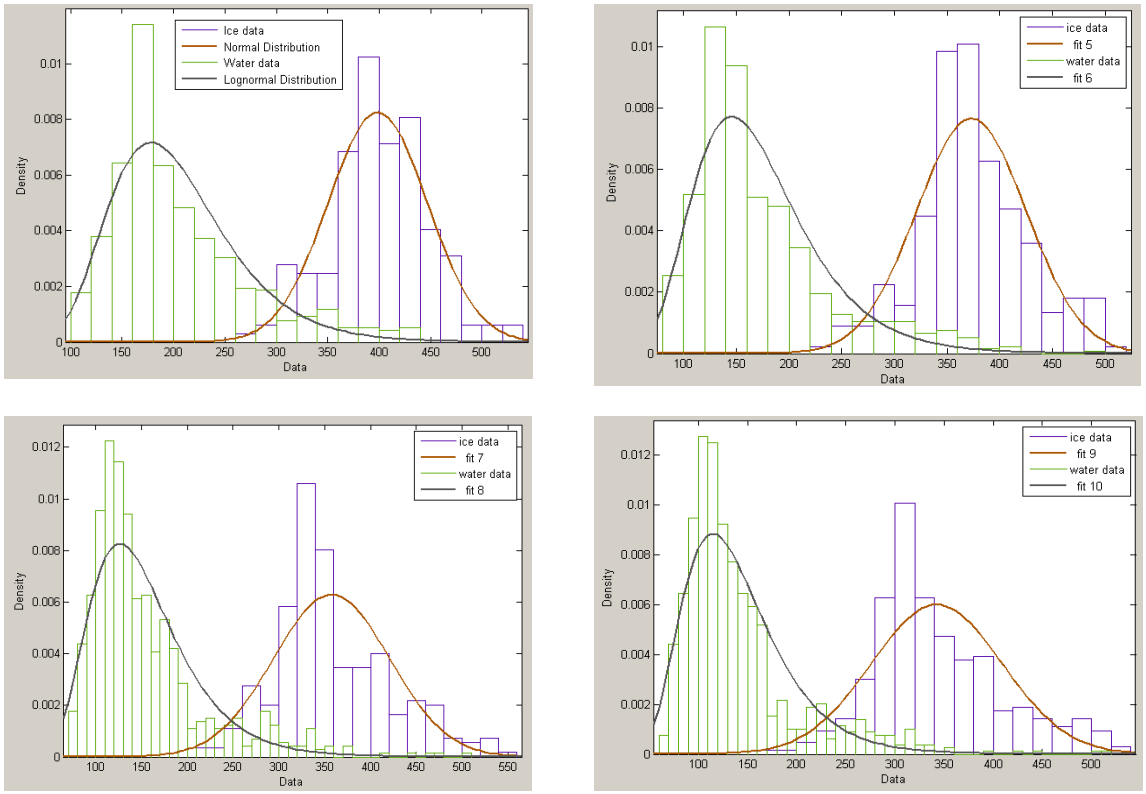


Figure A3. Distribution functions of water (left) and ice (right) reflectance in R01 channel for acquisition time 12 – 19 (from left to right)



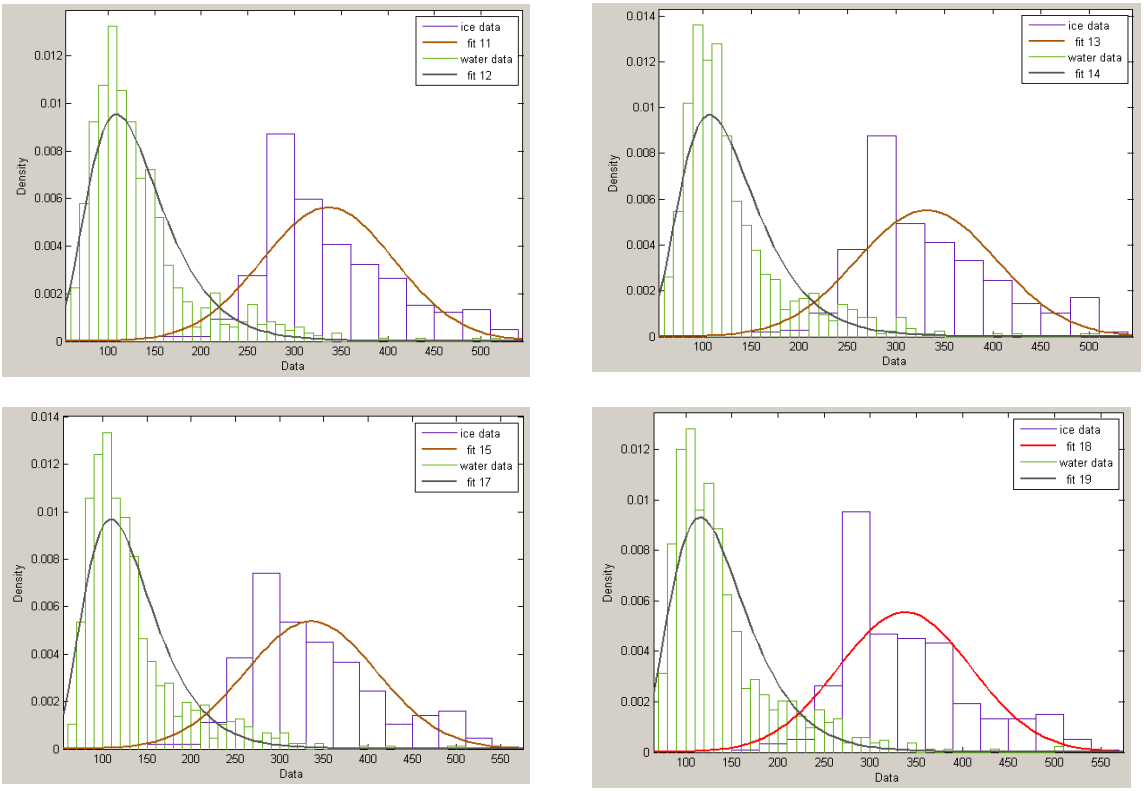
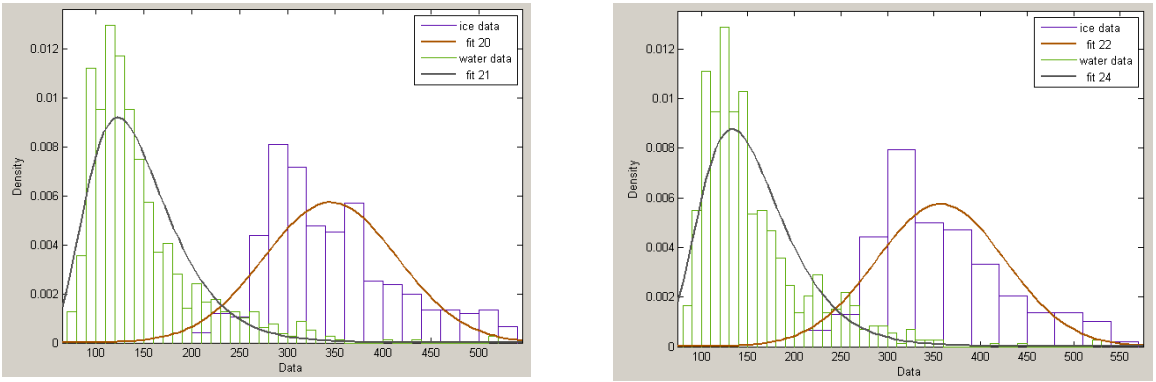


Figure A4. Distribution functions of water (left) and ice (right) reflectance in R01 channel for acquisition time 20 – 25 (from left to right)



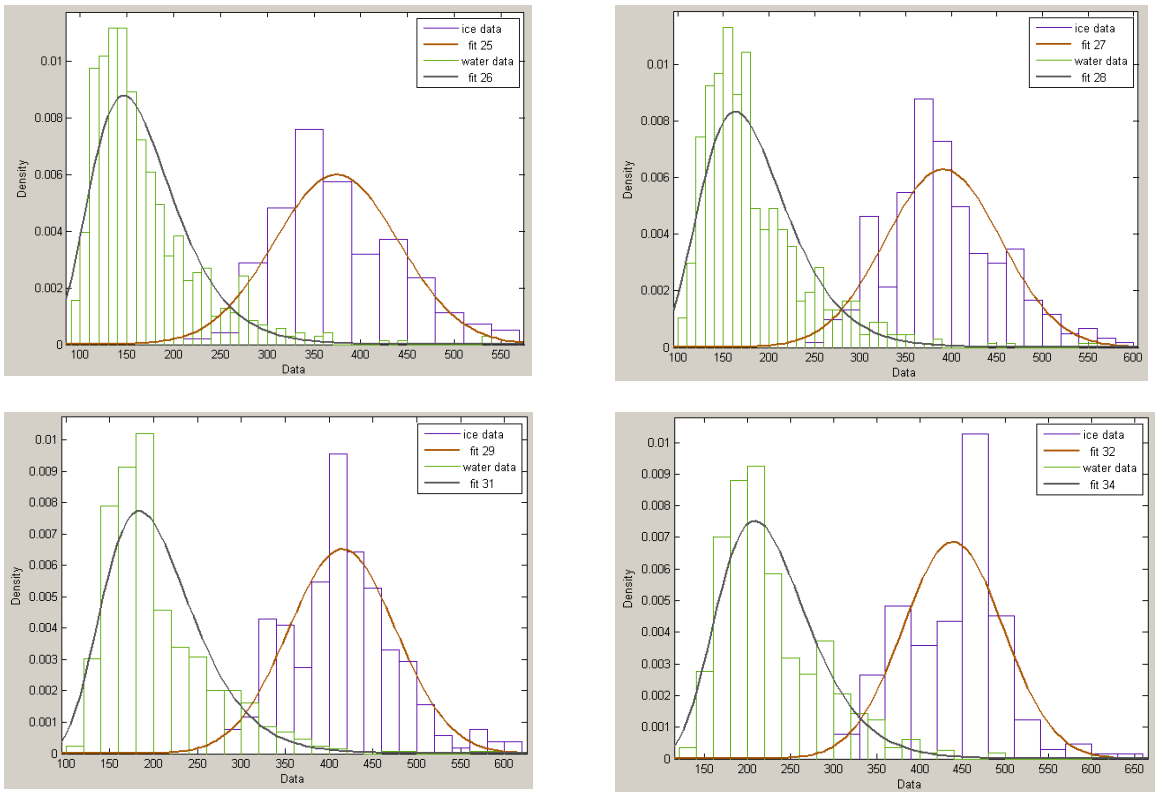
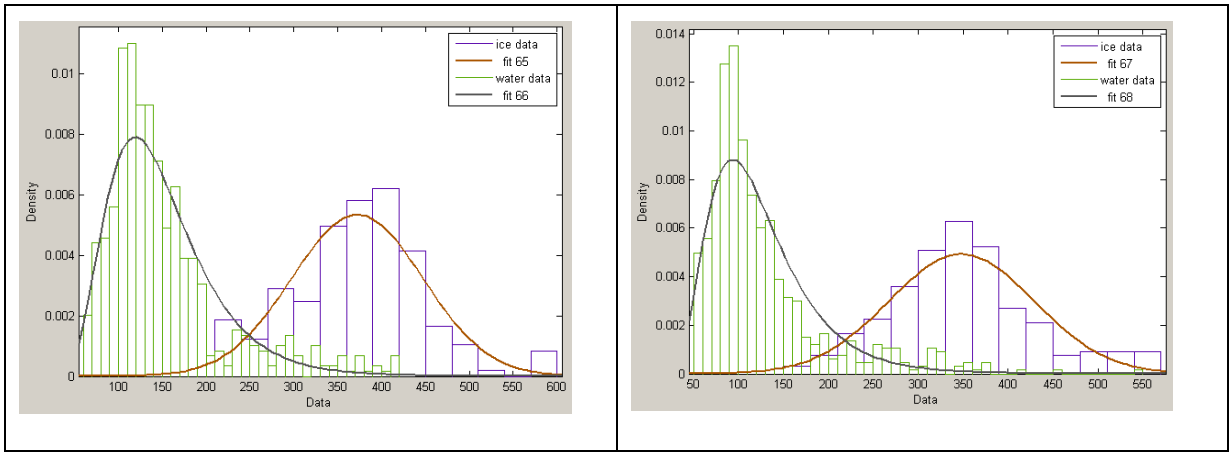


Figure A5. Distribution functions of water (left) and ice (right) reflectance in R02 channel for acquisition time 12 – 19 (from left to right)



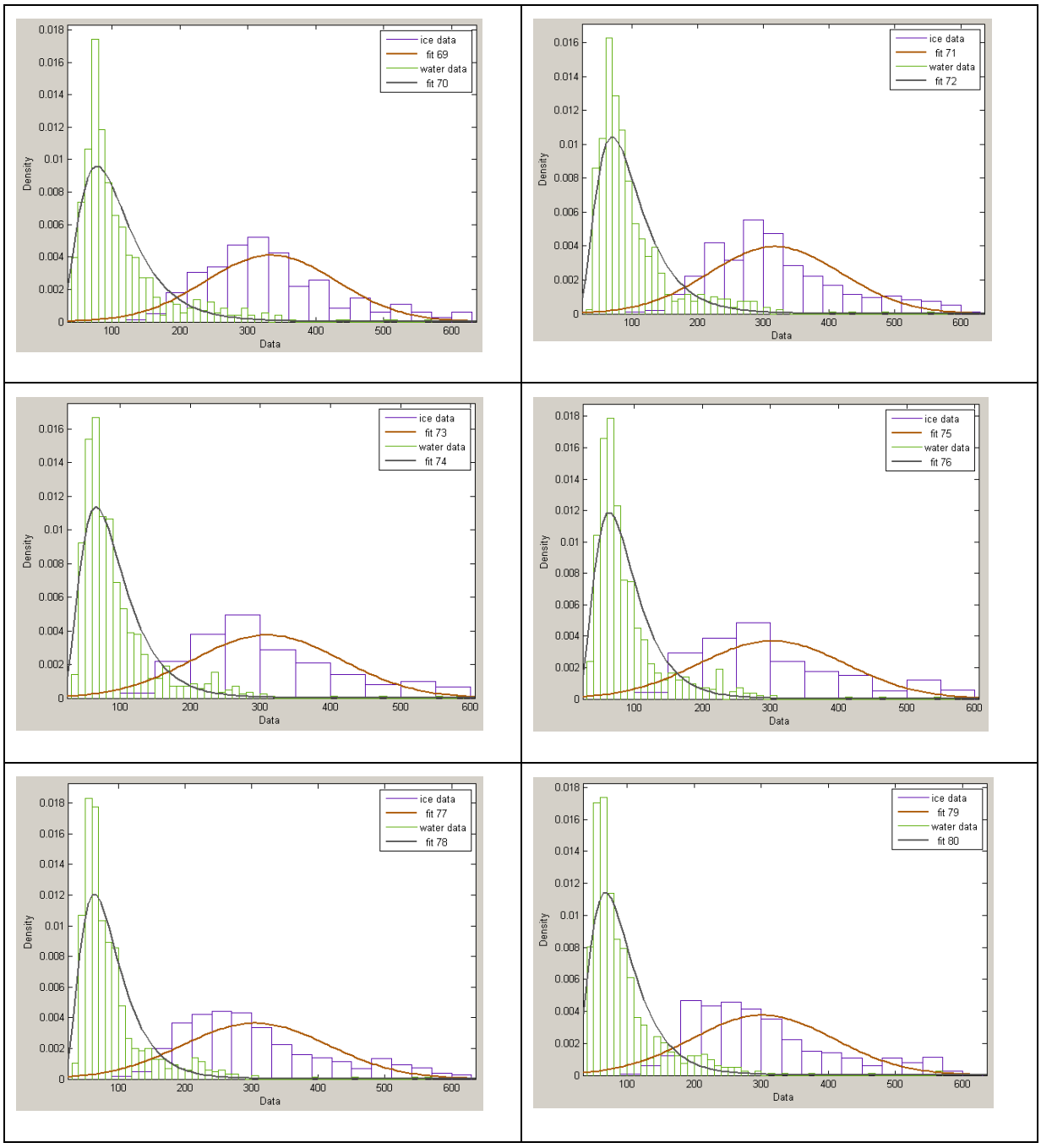
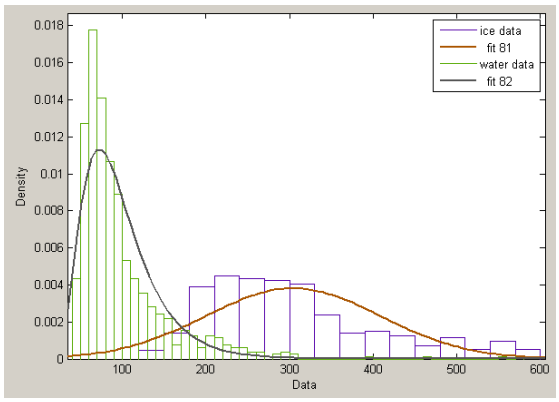
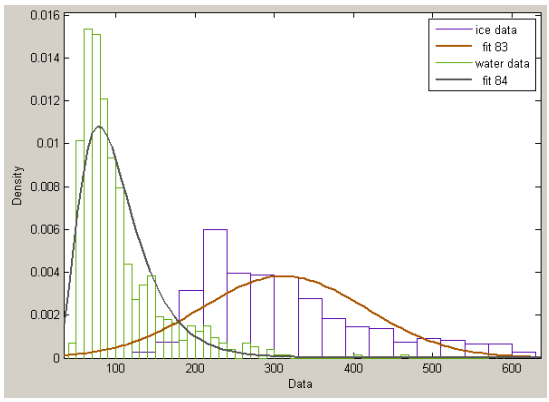
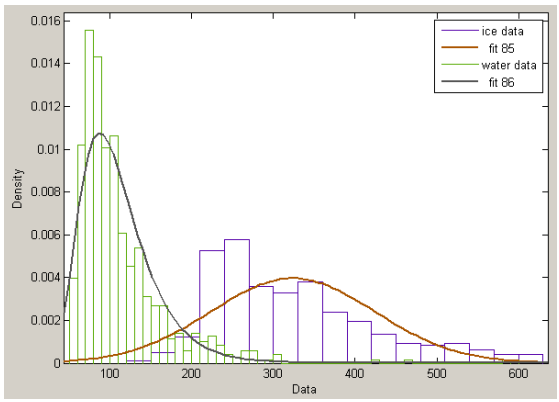
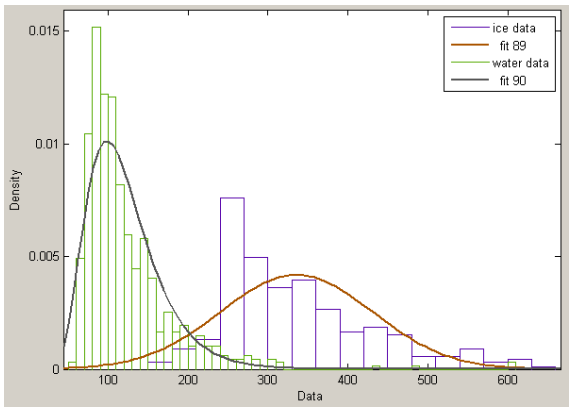


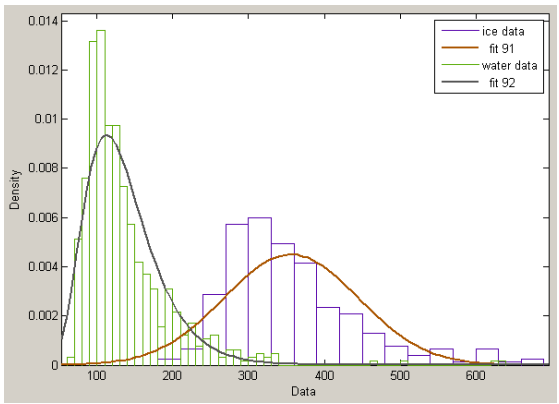
Figure A6. Distribution functions of water (left) and ice (right) reflectance in R02 channel for acquisition time 20 – 25 (from left to right)


ice data
fit 81
water data
fit 82
Density
Data


ice data
fit 83
water data
fit 84
Density
Data


ice data
fit 85
water data
fit 86
Density
Data


ice data
fit 89
water data
fit 90
Density
Data


ice data
fit 91
water data
fit 92
Density
Data

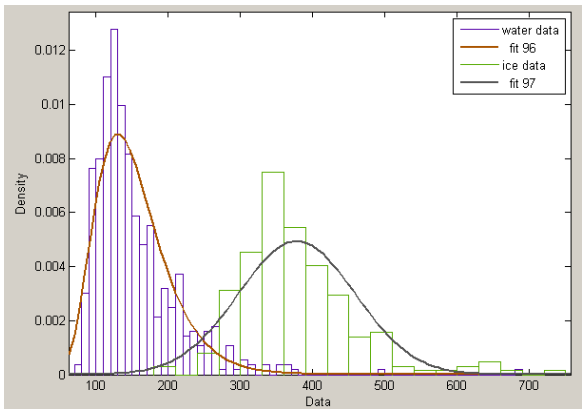
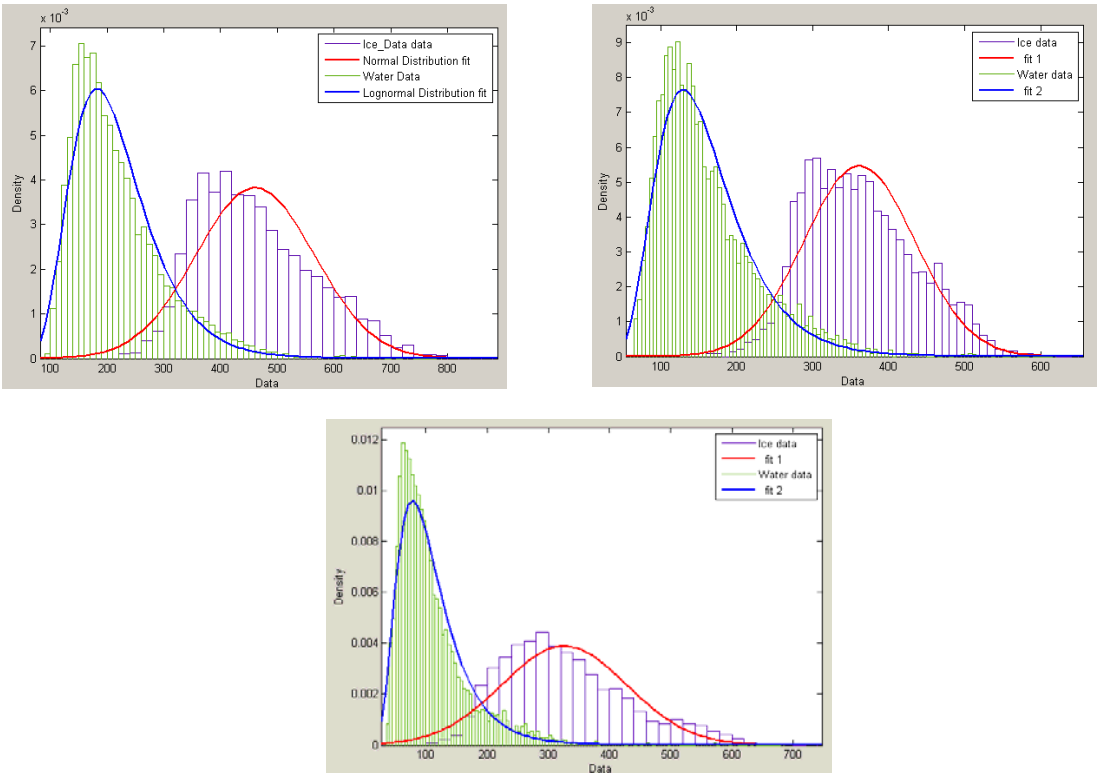
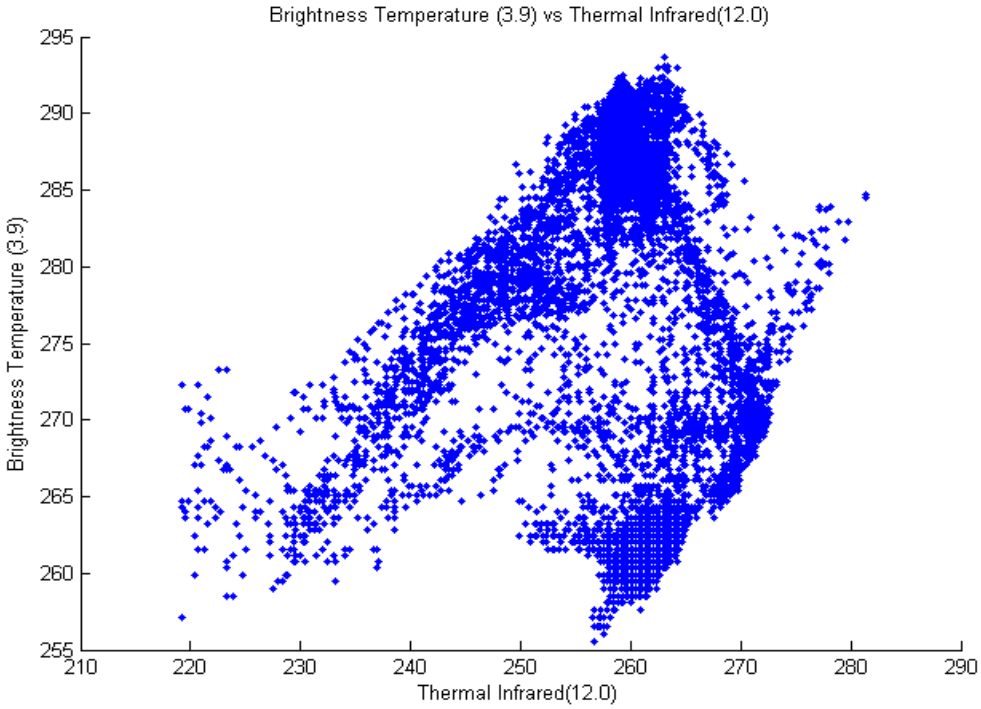

water data
fit 96
ice data
fit 97
Density
Data

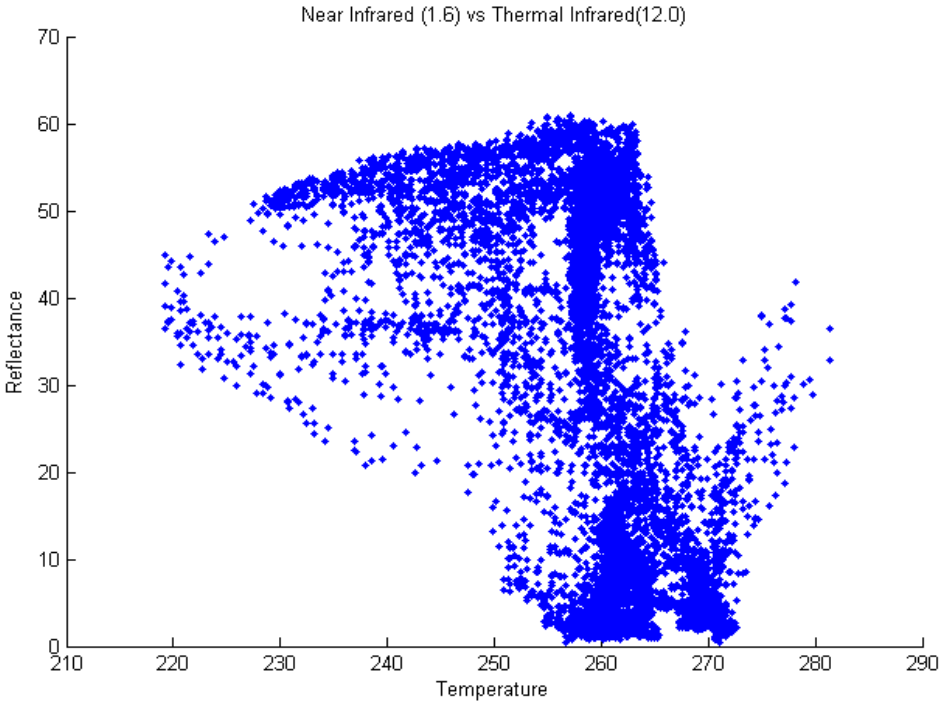
Figure A7. Distribution functions of water and ice reflectance for all acquisition times in HRV, R01, and R02 bands.

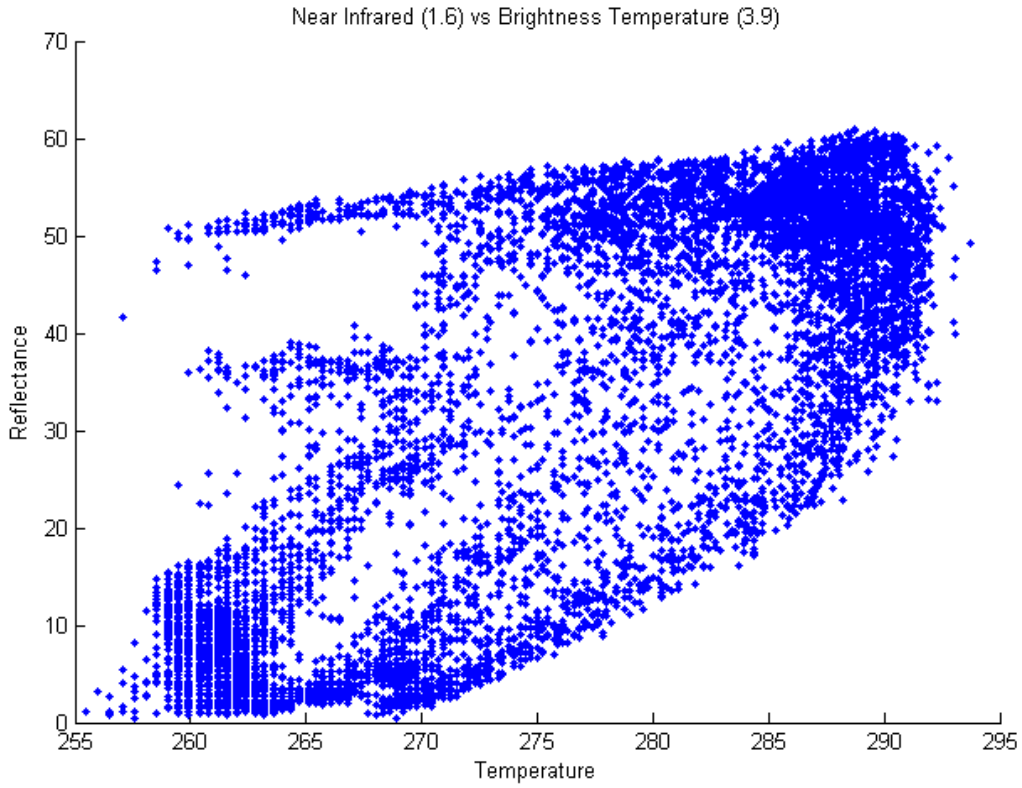


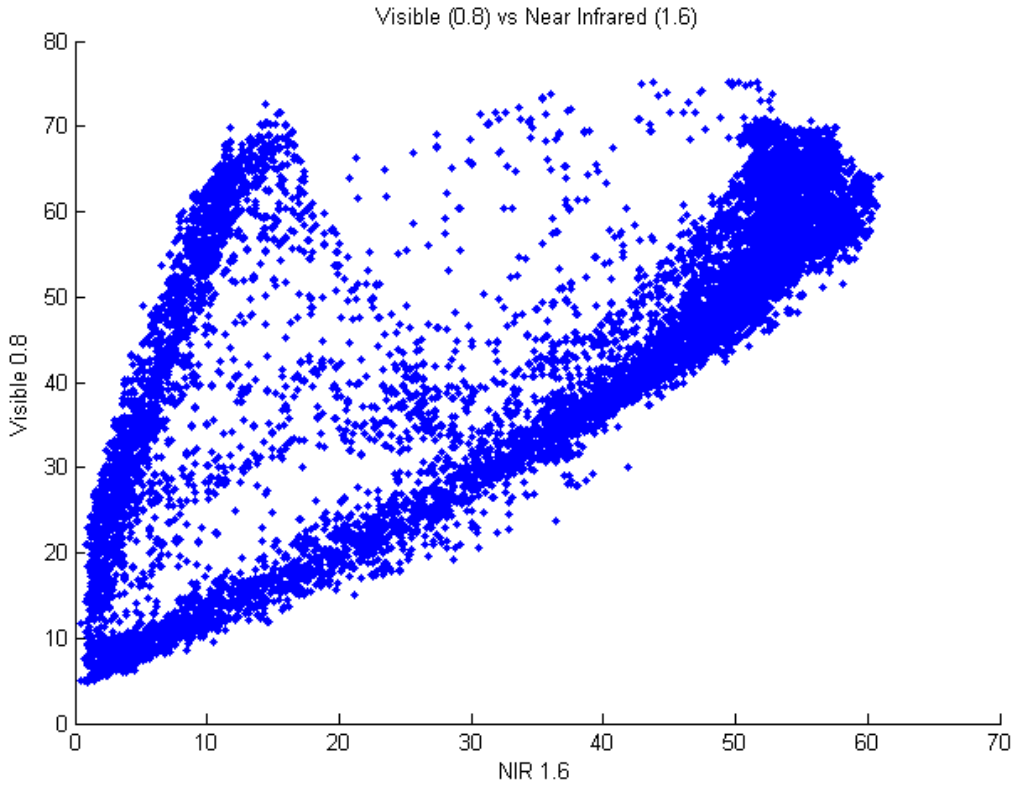
8.3 APPENDIX C

8.3.1 Channels Correlations






Near Infrared (1.6) vs Brightness Temperature (3.9)
Reflectance
Temperature
0
10
20
30
40
50
60
70
255
260
265
270
275
280
285
290
295


Visible (0.8) vs Near Infrared (1.6)
Visible 0.8
NIR 1.6
0
10
20
30
40
50
60
70
80

8.4 APPENDIX D

8.4.1 MATLAB CODES

a) BRDF Matrix Builder:

```
Load 'Matrix_BRDF_New'
%Matrix_BRDF = file;
Observation =  size(Matrix_BRDF_New,1)+ 1; %input('Observation # = ?');    %1-200
%Observation = size(Matrix_BRDF,1)+1;
%Observation = input(Matrix_BRDF,1)+1;
%Observation = size(BRDF_Matrix,1)+1;

column(1) =  input('date = ?');     %date
column(2) =  input('x = ?');   %x
column(3) =  input('y = ?');   %y
column(4) =  input('acquisition time = ?');    %time

%_____________________________________________________________________________

column(5) =  Matrix013307_1To4 (column(2), column(3), column(4),1);%HRV
column(6) =  Matrix013307_1To4 (column(2), column(3), column(4),2);%R01
column(7) =  Matrix013307_1To4 (column(2), column(3), column(4),3);%R02
column(8) =  Matrix013307_1To4 (column(2), column(3), column(4),4);%R03
column(9) =  Matrix013307_RAZ_SOL(column(2), column(3), column(4),1);%ARZ
column(10)=  Matrix013307_RAZ_SOL(column(2), column(3), column(4),2);%SAT
column(11)=  Matrix013307_RAZ_SOL(column(2), column(3), column(4),3);%SOL


%Observation = size(BRDF_Matrix,1)+1;

for i = 1:11
Matrix_BRDF_New(Observation, i) = column (i) ;
end

save Matrix_BRDF_New Matrix_BRDF
```

b) SST and NDSI Model:

SETTING THE DIRECTORY & PREBUILTING A MATRIX FOR COLORMAP 126

LOADING DATA FROM FILES 127

VECTOR CONVERSION 127

SST 127

STEPS 127


```
clear all   % clear workspace

clc     % clear Command Window
clf     % clear figure

delete images\*

% All .mat files are in "Data" folder
% All graphs will be saved in "images" folder

Setting the directory & prebuilting a matrix for colormap
diri = [pwd, '\'];  % assign the current working directory and \ to diri
```

```
diri2 = [diri, 'images\'];

% ******************ICE FRACTION DETERMINATION*************************

Loading data from files ************************************
for d = 59:59
    % Select variables ***************************************************
    % Select only sun illuminated pixels *********************************

    load ([diri,'Data/Matrix01' num2str(d,'%0.2d'),'07_1To4.mat']);
%     Ch1 = eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:, 12:25, 1)']);
    Ch2 = eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:, 12:25, 2)']);
%     Ch3 = eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:, 12:25, 3)']);
    Ch4 = eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:, 12:25, 4)']);
    clear(['Matrix01' num2str(d,'%0.2d') '07_1To4']);

    load ([diri,'Data/Matrix01',num2str(d,'%0.2d'),'07_RAZ_SOL.mat']);
%     s5 = ['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:, 12:25, 1)'];
%     ARZ = pi*eval(s5)/1800; clear s5
    s6 = ['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:, 12:25, 2)'];
    SAT = pi*eval(s6)/1800; clear s6
%     s7 = ['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:, 12:25, 3)'];
%     SOL = pi*eval(s7)/1800; clear s7
    clear (['Matrix01' num2str(d,'%0.2d') '07_RAZ_SOL']);

    load ([diri,'Data/Matrix01' num2str(d,'%0.2d'),'07_T04_T10.mat']);
    Ch9 =eval(['Matrix01', num2str(d,'%0.2d'), '07_T04_T10(:,:,12:25,2)']);
    Ch10=eval(['Matrix01', num2str(d,'%0.2d'), '07_T04_T10(:,:,12:25,3)']);
    clear(['Matrix01' num2str(d,'%0.2d') '07_T04_T10']);

    n = size(Ch2);  % [400, 500, 14] = [n(1), n(2), n(3)]

Vector conversion ******************************************
%     Ch1 = reshape(Ch1, [n(1)*n(2), n(3), 1]);
    Ch2 = reshape(Ch2, [n(1)*n(2), n(3), 1]);
    Ch2(Ch2==1) = 3;
%     Ch3 = reshape(Ch3, [n(1)*n(2), n(3), 1]);
    Ch4 = reshape(Ch4, [n(1)*n(2), n(3), 1]);

%     ARZ = reshape(ARZ, [n(1)*n(2), n(3), 1]);
%     SOL = reshape(SOL, [n(1)*n(2), n(3), 1]);
    SAT = reshape(SAT, [n(1)*n(2), n(3), 1]);

    Ch9 = reshape(Ch9, [n(1)*n(2), n(3), 1]);
    Ch9 = Ch9/10;
    Ch10 = reshape(Ch10, [n(1)*n(2), n(3), 1]);
    Ch10 = Ch10/10;

    y1 = max(Ch9);
    y2 = find(y1>320);
    if ~isempty(y2)
        Ch9(:,y2) = nan;
    end
    Ice2 = (Ch2-Ch4)./(Ch2+Ch4);    % NDSI
    Ice2(Ice2==0.5) = 3;  % set 0 equal to 3 in Ice2, brown background
%     Ice2(Ice2==0.5) = NaN;  % set 0 equal to NaN in Ice2, deep blue background

SST ****************************************************
     SST = Ch9 + (0.99*cos(SAT) + 0.21) .* (Ch9-Ch10) + ...
         (0.364./cos(SAT)+0.15) .* (Ch9-Ch10).^2 + ...
         (0.327./(cos(SAT)).^2 + 0.11);

steps
    % Step 1 : cloud detection ***********************************************
    for m = 1:n(3) % n(3) = 14
        %   NDSI ********************************
        NDSI = reshape(Ice2(:,m),[n(1) n(2)]);

        % plotting graphs
%         subplot(1, 2, 1);
        h = imagesc(NDSI,[-1, 3]);
```

```
        title('NDSI')

        t = ['NDSI_map',num2str(d,'%.2d'),num2str(m+11,'%.2d')];
        % saving the graphs
        saveas(h, t, 'fig');
        % moving the graphs to the subfolder, images
        movefile([t, '.fig'], diri2);

%         print ('-dpng',t, '.png');
        saveas(h, t, 'png')
        movefile([t, '.png'], diri2);



        % SST ************************************************************
        SSTc = reshape(SST(:,m),[n(1) n(2)]);
        % plotting graphs
%         subplot(1, 2, 2);
        if isempty(min(SSTc(SSTc>200))) | isnan(max(max(SSTc)))
            h = imagesc(SSTc);
        else
            h = imagesc(SSTc,[min(SSTc(SSTc>200)), max(max(SSTc))]);
        end

        title('SST')

        t = ['SST_map',num2str(d,'%.2d'),num2str(m+11,'%.2d')];
        % saving the graphs
        saveas(h, t, 'fig');
        % moving the graphs to the subfolder, images
        movefile([t, '.fig'], diri2);

%         print ('-dpng',t, '.png');
        saveas(h, t, 'png')
        movefile([t, '.png'], diri2);

        clf     % clear figure
    end
end
close   % close figure
return
%*************************************************************************
```

c) Normalization of ICE and Water Reflectance in all Channels:

```
function NormR01(IceR,IceT,WaterR,WaterT)
%NORMR01    Create plot of datasets and fits
%   NORMR01(ICER,ICET,WATERR,WATERT)
%   Creates a plot, similar to the plot in the main distribution fitting
%   window, using the data that you provide as input.  You can
%   apply this function to the same data you used with dfittool
%   or with different data.  You may want to edit the function to
%   customize the code and this help message.
%
%   Number of datasets:  2
%   Number of fits:  2

% This function was automatically generated on 11-Aug-2008 10:20:47

% Data from dataset "IceR data":
%    Y = IceR
%    Frequency = IceT

% Data from dataset "WaterR data":
%    Y = WaterR
%    Frequency = WaterT

% Force all inputs to be column vectors
IceR = IceR(:);
IceT = IceT(:);
```

```
WaterR = WaterR(:);
WaterT = WaterT(:);

% Set up figure to receive datasets and fits
f_ = clf;
figure(f_);
set(f_,'Units','Pixels','Position',[441 243 680 475.45]);
legh_ = []; legt_ = {};   % handles and text for legend
ax_ = newplot;
set(ax_,'Box','on');
grid(ax_,'on');
hold on;

% --- Plot data originally in dataset "IceR data"
t_ = ~isnan(IceR) & ~isnan(IceT);
Data_ = IceR(t_);
Freq_ = IceT(t_);
[F_,X_] = ecdf(Data_,'Function','cdf'...
               ,'freq',Freq_...
              );  % compute empirical cdf
Bin_.rule = 1;
[C_,E_] = dfswitchyard('dfhistbins',Data_,[],Freq_,Bin_,F_,X_);
[N_,C_] = ecdfhist(F_,X_,'edges',E_); % empirical pdf from cdf
h_ = bar(C_,N_,'hist');
set(h_,'FaceColor','none','EdgeColor',[0.333333 0 0.666667],...
       'LineStyle','-', 'LineWidth',1);
xlabel('Data');
ylabel('Density')
legh_(end+1) = h_;
legt_{end+1} = 'IceR data';
% --- Plot data originally in dataset "WaterR data"
t_ = ~isnan(WaterR) & ~isnan(WaterT);
Data_ = WaterR(t_);
Freq_ = WaterT(t_);
[F_,X_] = ecdf(Data_,'Function','cdf'...
               ,'freq',Freq_...
              );  % compute empirical cdf
Bin_.rule = 1;
[C_,E_] = dfswitchyard('dfhistbins',Data_,[],Freq_,Bin_,F_,X_);
[N_,C_] = ecdfhist(F_,X_,'edges',E_); % empirical pdf from cdf
h_ = bar(C_,N_,'hist');
set(h_,'FaceColor','none','EdgeColor',[0.333333 0.666667 0],...
       'LineStyle','-', 'LineWidth',1);
xlabel('Data');
ylabel('Density')
legh_(end+1) = h_;
legt_{end+1} = 'WaterR data';

% Nudge axis limits beyond data limits
xlim_ = get(ax_,'XLim');
if all(isfinite(xlim_))
   xlim_ = xlim_ + [-1 1] * 0.01 * diff(xlim_);
   set(ax_,'XLim',xlim_)
end

x_ = linspace(xlim_(1),xlim_(2),100);

% --- Create fit "fit 1"

% Fit this distribution to get parameter values
t_ = ~isnan(IceR) & ~isnan(IceT);
Data_ = IceR(t_);
Freq_ = IceT(t_);
% To use parameter estimates from the original fit:
%     p_ = [ 365.5663311871, 74.16110421238];
pargs_ = cell(1,2);
[pargs_{:}] = normfit(Data_, 0.05, [], Freq_);
p_ = [pargs_{:}];
y_ = normpdf(x_,p_(1), p_(2));
h_ = plot(x_,y_,'Color',[1 0 0],...
          'LineStyle','-', 'LineWidth',2,...
```

```
          'Marker','none', 'MarkerSize',6);
legh_(end+1) = h_;
legt_{end+1} = 'fit 1';

% --- Create fit "fit 2"

% Fit this distribution to get parameter values
t_ = ~isnan(WaterR) & ~isnan(WaterT);
Data_ = WaterR(t_);
Freq_ = WaterT(t_);
% To use parameter estimates from the original fit:
%     p_ = [ 180.2315394019, 77.76066798109];
pargs_ = cell(1,2);
[pargs_{:}] = normfit(Data_, 0.05, [], Freq_);
p_ = [pargs_{:}];
y_ = normpdf(x_,p_(1), p_(2));
h_ = plot(x_,y_,'Color',[0 0 1],...
          'LineStyle','-', 'LineWidth',2,...
          'Marker','none', 'MarkerSize',6);
legh_(end+1) = h_;
legt_{end+1} = 'fit 2';

hold off;
leginfo_ = {'Orientation', 'vertical'};
h_ = legend(ax_,legh_,legt_,leginfo_{:}); % create and reposition legend
set(h_,'Units','normalized');
t_ = get(h_,'Position');
t_(1:2) = [0.703922,0.659851];
set(h_,'Interpreter','none','Position',t_);
```

d) Neural Network Water Training Model:

```
clear all;


load New_Inputs;
load New_Outputs;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%        Randomizing of training sets         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Inputs = New_Inputs(1:9,:);
Output = New_Outputs(2,:)/10; % 1: HRV; 2: R01; 3: R02

%Inputs = Inputs_water(1:9,:);
%Output = Outputs_water(1,:)/10;

In_Out = cat(1,Inputs,Output);
Rand_In_Out = In_Out(:,randperm(size(In_Out,2)));
Inputs = Rand_In_Out(1:9,:);
Output = Rand_In_Out(10,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Normalization of training sets        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Sample_size = size(Inputs,2);
set_size = fix(Sample_size/3)

[Norm_Inputs,PS_water_Inputs]= mapminmax(Inputs);
[Norm_Outputs,PS_water_Outputs]= mapminmax(Output);


%[Norm_Inputs,min_Inputs,max_Inputs] = premnmx(Inputs);
%[Norm_Outputs,min_Outputs,max_Outputs] = premnmx(Output);

Norm_Inputs_train = Norm_Inputs(:,1: set_size);
```

```
Norm_Inputs_val = Norm_Inputs(:,set_size+1: 2*set_size);
Norm_Inputs_test = Norm_Inputs(:,2*set_size+1:Sample_size);

Norm_Outputs_train = Norm_Outputs(:,1: set_size);
Norm_Outputs_val = Norm_Outputs(:,set_size+1: 2*set_size);
Norm_Outputs_test = Norm_Outputs(:,2*set_size+1:Sample_size);

val.P = Norm_Inputs_val; val.T = Norm_Outputs_val;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          NN Setting and training             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
net_water = newff(minmax(Norm_Inputs),[9 20 1],{'tansig'  'tansig' 'tansig'
'tansig'},'traingdx');

net_water.trainParam.goal = 0.0001;
net_water.trainParam.show = 20;
net_water.trainParam.epochs = 10000;
net_water.trainParam.max_fail = 200;

net_water.trainParam.lr = 0.01; % learning rate
net_water.trainParam.lr_inc = 1.01; % acceleration rate
net_water.trainParam.lr_dec = 0.9;  % stopping rate
net_water.trainParam.max_perf_inc = 1.02 % Rmax
net_water.trainParam.mc = 0.9; % momentum

[net_water,tr] = train(net_water,Norm_Inputs_train,Norm_Outputs_train,[],[],val);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         Simulation and assessment            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%[Results_train] = postmnmx(Norm_Results_train, min_Outputs,max_Outputs)';

Norm_Results_train = sim(net_water,Norm_Inputs_train);
[Results_train] = mapminmax('reverse', Norm_Results_train, PS_water_Outputs)';

Norm_Results_val = sim(net_water,Norm_Inputs_val);
[Results_val] = mapminmax('reverse', Norm_Results_val, PS_water_Outputs)';

Norm_Results_test = sim(net_water,Norm_Inputs_test);
[Results_test] = mapminmax('reverse', Norm_Results_test, PS_water_Outputs)';

Obs_sim_train(1,:) = Output(1: set_size);
Obs_sim_train(2,:) = Results_train(:,1);
Obs_sim_train(3,:) = Obs_sim_train(2,:)- Obs_sim_train(1,:);
Obs_sim_train(4,:) = Obs_sim_train(3,:) .* Obs_sim_train(3,:);

RMSE_train = sqrt(sum(Obs_sim_train(4,:))/set_size)

Obs_sim_val(1,:) = Output(set_size+1: 2*set_size);
Obs_sim_val(2,:) = Results_val(:,1);
Obs_sim_val(3,:) = Obs_sim_val(2,:)-Obs_sim_val(1,:);
Obs_sim_val(4,:) = Obs_sim_val(3,:) .* Obs_sim_val(3,:);

RMSE_val = sqrt(sum(Obs_sim_val(4,:))/set_size)

Obs_sim_test(1,:) = Output(2*set_size+1:Sample_size);
Obs_sim_test(2,:) = Results_test(:,1);
Obs_sim_test(3,:) = Obs_sim_test(2,:)-Obs_sim_test(1,:);
Obs_sim_test(4,:) = Obs_sim_test(3,:) .* Obs_sim_test(3,:);

RMSE_test = sqrt(sum(Obs_sim_test(4,:))/set_size)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

e) Neural Network Ice Training Model:

```
clear all;

load Outputs_Thick_Ice;
load Inputs_Thick_Ice;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%        Randomizing of training sets          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Inputs = Inputs_Thick_Ice(1:9,:);
Output = Outputs_Thick_Ice(3,:)/10; % 1: HRV; 2: R01; 3: R02

%Inputs = Inputs_Ice(1:9,:);
%Output = Outputs_Ice(1,:)/10;

In_Out = cat(1,Inputs,Output);
Rand_In_Out = In_Out(:,randperm(size(In_Out,2)));
Inputs = Rand_In_Out(1:9,:);
Output = Rand_In_Out(10,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Normalization of training sets         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Sample_size = size(Inputs,2);
set_size = fix(Sample_size/3)

[Norm_Inputs,PS_Ice_Inputs]= mapminmax(Inputs);
[Norm_Outputs,PS_Ice_Outputs]= mapminmax(Output);


%[Norm_Inputs,min_Inputs,max_Inputs] = premnmx(Inputs);
%[Norm_Outputs,min_Outputs,max_Outputs] = premnmx(Output);

Norm_Inputs_train = Norm_Inputs(:,1: set_size);
Norm_Inputs_val = Norm_Inputs(:,set_size+1: 2*set_size);
Norm_Inputs_test = Norm_Inputs(:,2*set_size+1:Sample_size);

Norm_Outputs_train = Norm_Outputs(:,1: set_size);
Norm_Outputs_val = Norm_Outputs(:,set_size+1: 2*set_size);
Norm_Outputs_test = Norm_Outputs(:,2*set_size+1:Sample_size);

val.P = Norm_Inputs_val; val.T = Norm_Outputs_val;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          NN Setting and training             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
net_ice = newff(minmax(Norm_Inputs_train),[9 20 1],{'tansig'  'tansig' 'tansig'
'tansig'},'traingdx');

net_ice.trainParam.goal = 0.0001;
net_ice.trainParam.show = 20;
net_ice.trainParam.epochs = 5000;
net_ice.trainParam.max_fail = 200;

net_ice.trainParam.lr = 0.01; % learning rate
net_ice.trainParam.lr_inc = 1.01; % acceleration rate
net_ice.trainParam.lr_dec = 0.9;  % stopping rate
net_ice.trainParam.max_perf_inc = 1.02 % Rmax
net_ice.trainParam.mc = 0.9; % momentum

[net_ice,tr] = train(net_ice,Norm_Inputs_train,Norm_Outputs_train,[],[],val);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         Simulation and assessment           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Norm_Results_train = sim(net_ice,Norm_Inputs_train);
%[Results_train] = postmnmx(Norm_Results_train, min_Outputs,max_Outputs)';
[Results_train] = mapminmax('reverse', Norm_Results_train, PS_Ice_Outputs)';

Norm_Results_val = sim(net_ice,Norm_Inputs_val);
[Results_val] = mapminmax('reverse', Norm_Results_val, PS_Ice_Outputs)';

Norm_Results_test = sim(net_ice,Norm_Inputs_test);
[Results_test] = mapminmax('reverse', Norm_Results_test, PS_Ice_Outputs)';


Obs_sim_train(1,:) = Output(1: set_size);
Obs_sim_train(2,:) = Results_train(:,1);
Obs_sim_train(3,:) = Obs_sim_train(2,:)- Obs_sim_train(1,:);
Obs_sim_train(4,:) = Obs_sim_train(3,:) .* Obs_sim_train(3,:);

RMSE_train = sqrt(sum(Obs_sim_train(4,:))/set_size)

Obs_sim_val(1,:) = Output(set_size+1: 2*set_size);
Obs_sim_val(2,:) = Results_val(:,1);
Obs_sim_val(3,:) = Obs_sim_val(2,:)-Obs_sim_val(1,:);
Obs_sim_val(4,:) = Obs_sim_val(3,:) .* Obs_sim_val(3,:);

RMSE_val = sqrt(sum(Obs_sim_val(4,:))/set_size)

Obs_sim_test(1,:) = Output(2*set_size+1:Sample_size);
Obs_sim_test(2,:) = Results_test(:,1);
Obs_sim_test(3,:) = Obs_sim_test(2,:)-Obs_sim_test(1,:);
Obs_sim_test(4,:) = Obs_sim_test(3,:) .* Obs_sim_test(3,:);

RMSE_test = sqrt(sum(Obs_sim_test(4,:))/set_size)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Neural Network Simulation:

```
clear all;
load Matrix016307_RAZ_SOL;

load PS_water_Outputs;
load PS_water_Inputs;

load PS_Ice_Outputs;
load PS_Ice_Inputs;

load net_ice;
load net_water;

% Input format (nine entries): [ARZ SAT SOL cos(ARZ) cos(SAT) cos(SOL) sin(ARZ) sin(SAT)
% sin(SOL)]

SOLSAT = Matrix016307_RAZ_SOL;

ARZ = pi*(SOLSAT(:,:,:,1))/1800;
SAT = pi*(SOLSAT(:,:,:,2))/1800;
SOL = pi*(SOLSAT(:,:,:,3))/1800;

COSARZ = cos (ARZ);
COSSAT = cos (SAT);
COSSOL = cos (SOL);

SINARZ = sin (ARZ);
SINSAT = sin (SAT);
SINSOL = sin (SOL);

x1 = 1;
y1 = 1;
```

```
t1 = 1;

     for t = 15:16


     for y = 101:350

        for x = 81:300



input(:,:) = [ARZ(x,y,t) SAT(x,y,t) SOL(x,y,t) COSARZ(x,y,t) COSSAT(x,y,t) COSSOL(x,y,t)
SINARZ(x,y,t) SINSAT(x,y,t) SINSOL(x,y,t)];

NORM_water_input = mapminmax('apply',input', PS_water_Inputs);
norm_sim_water = sim(net_water,NORM_water_input);
sim_water = mapminmax('reverse',norm_sim_water, PS_water_Outputs)';


NORM_ice_input = mapminmax('apply',input',  PS_Ice_Inputs);
norm_sim_ice = sim(net_ice,NORM_ice_input);
sim_ice = mapminmax('reverse',norm_sim_ice, PS_Ice_Outputs)';


SIMWAT(x-80,y-100,t-14) = sim_water;
SIMICE(x-80,y-100,t-14) = sim_ice;


        end

    end

end
```

f) Model Evaluator and Plotter:

```
function varargout = Ri_Rw_plot(varargin)
% Last Modified by GUIDE v2.5 19-Aug-2008 23:56:52

% Begin initialization code - DO NOT EDIT
gui_Singleton = 0;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Ri_Rw_plot_OpeningFcn, ...
                   'gui_OutputFcn',  @Ri_Rw_plot_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
return


% --- Executes just before Ri_Rw_plot is made visible.
function Ri_Rw_plot_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
% ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
mkdir images
clc
% load data
Dir = fullfile(pwd, 'Data\');
load([Dir, 'Ri.mat'])
load([Dir, 'Rw.mat'])
handles.Ri = Ri;
handles.Rw = Rw;
clear Ri Rw
% load ('..\data\Matrix012307_1To4')

load ('Data\HRV');
handles.HRV = HRV./10;
clear HRV

load('Data\Map_land.mat')
handles.Map_land = Map_land./10;
clear Map_land

% 8/12/2008
load('Data\RHRV.mat')
handles.RHRV = RHRV./10;
clear RHRV
% ------------------------------------------------

% initialization the first plot = Rw & HRV
handles.x = str2num(get(handles.x_edit, 'String'));
handles.y = str2num(get(handles.y_edit, 'String'));

handles.p1 = zeros(14, 1);
handles.p2 = zeros(14, 1);
handles.p1(:) = handles.HRV(handles.y, handles.x, :);
handles.p2(:) = handles.Rw(handles.y, handles.x, :);

handles.l1 = 'HRV';
handles.l2 = 'Rw';
% ------------------------------------------------

% initialize the editers and the sliders
set(handles.x_edit, 'String', '250');
set(handles.x_slider, 'Value', 250);
set(handles.y_edit, 'String', '250');
set(handles.y_slider, 'Value', 250);
% ------------------------------------------------

% axes: map area
axes(handles.axes1)
set(gca, 'NextPlot', 'add')
h = imagesc(handles.Map_land);
axis image tight
set(h, 'HitTest', 'off')
hold on

for i=50:50:450
    set(gca, 'NextPlot', 'add')
    h = line([i i], [1 400], 'color', 'black', 'LineStyle', ':');
    set(h, 'HitTest', 'off')
    hold on
end

for i=50:50:350
    set(gca, 'NextPlot', 'add')
    h = line([1 500], [i i], 'color', 'black', 'LineStyle', ':');
    set(h, 'HitTest', 'off')
    hold on
end

h = plot(250, 250, 'g.');
set(h, 'HitTest', 'off')
hold on
```

```
% --------------------------------------------------

set(handles.figure1, 'KeyPressFcn', @KeyFunction);
set(handles.x_edit, 'KeyPressFcn', @KeyFunction);
set(handles.y_edit, 'KeyPressFcn', @KeyFunction);
% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@


% Choose default command line output for Ri_Rw_plot
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Ri_Rw_plot wait for user response (see UIRESUME)
% uiwait(handles.figure1);
return

% --- Outputs from this function are returned to the command line.
function varargout = Ri_Rw_plot_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
return


function x_edit_Callback(hObject, eventdata, handles)
% ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
x_sliderValue = get(handles.x_edit, 'String');
x_sliderValue = str2num(x_sliderValue);
if (isempty(x_sliderValue) || x_sliderValue < 1 || x_sliderValue > 500)
    set(handles.x_slider, 'Value', 250);
    set(handles.x_edit, 'String', '250');
else
    set(handles.x_slider, 'Value', x_sliderValue);
end
% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

% Hints: get(hObject,'String') returns contents of x_edit as text
%        str2double(get(hObject,'String')) returns contents of x_edit as a double
return


% --- Executes during object creation, after setting all properties.
function x_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to x_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
return


function y_edit_Callback(hObject, eventdata, handles)
% ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
y_sliderValue = get(handles.y_edit, 'String');
y_sliderValue = str2num(y_sliderValue);
if (isempty(y_sliderValue) || y_sliderValue < 1 || y_sliderValue > 400)
    set(handles.y_slider, 'Value', 250);
    set(handles.y_edit, 'String', '250');
else
    set(handles.y_slider, 'Value', y_sliderValue);
end
```

```
% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

% Hints: get(hObject,'String') returns contents of y_edit as text
%        str2double(get(hObject,'String')) returns contents of y_edit as a double
return


% --- Executes during object creation, after setting all properties.
function y_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to y_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
return

% --- Executes on button press in plot_pushbutton.
function plot_pushbutton_Callback(hObject, eventdata, handles)

% ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
handles.x = str2num(get(handles.x_edit, 'String'));
handles.y = str2num(get(handles.y_edit, 'String'));

if get(handles.Rw_HRV_radiobutton, 'Value') == 1
    clear handles.p
    handles.p = zeros(14, 3);
    handles.p(:, 1) = handles.RHRV(handles.y, handles.x, :);
    handles.p(:, 2) = handles.HRV(handles.y, handles.x, :);
    handles.p(:, 3) = handles.Rw(handles.y, handles.x, :);

    handles.l1 = 'Stepwise Linear Regression Model, RHRV';
    handles.l2 = 'Observed HRV';
    handles.l3 = 'Neural Network Model, Rw';
    Title = ({'Neural Network Model vs. Observed HRV'; ...
        'and Stepwise Linear Regression Model'});

% -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
    handles.RMSE(1) = sqrt(sum((handles.Rw(handles.y, handles.x, :) - ...
        handles.HRV(handles.y, handles.x, :)))^2/1400);
    handles.RMSE(2) = sqrt(sum((handles.RHRV(handles.y, handles.x, :) - ...
        handles.HRV(handles.y, handles.x, :)))^2/140);

%     fprintf('RMSE.Rw_HRV = %f\nRMSE.RHRV_HRV = %f\n\n', ...
%         handles.RMSE(1), handles.RMSE(2))
    set(handles.RMSE_Rw_HRV, 'String', num2str(handles.RMSE(1)));
    set(handles.RMSE_RHRV_HRV, 'String', num2str(handles.RMSE(2)));
% -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-

elseif get(handles.Ri_HRV_radiobutton, 'Value') == 1
    clear handles.p
    handles.p = zeros(14, 2);
    handles.p(:, 1) = handles.Ri(handles.y, handles.x, :);
    handles.p(:, 2) = handles.HRV(handles.y, handles.x, :);

    handles.l1 = 'Ri';
    handles.l2 = 'HRV';
    Title = 'Neural Network Estimation and Observed HRV';

elseif get(handles.Ri_Rw_radiobutton, 'Value') == 1
    handles.p = zeros(14, 2);
    handles.p(:, 1) = handles.Ri(handles.y, handles.x, :);
    handles.p(:, 2) = handles.Rw(handles.y, handles.x, :);

    handles.l1 = 'Ri';
    handles.l2 = 'Rw';
    Title = 'Neural Network Estimation';
else
```

```
    handles.p = zeros(14, 3);
    handles.p(:, 1) = handles.Ri(handles.y, handles.x, :);
    handles.p(:, 2) = handles.HRV(handles.y, handles.x, :);
    handles.p(:, 3) = handles.Rw(handles.y, handles.x, :);

    handles.l1 = 'Ri';
    handles.l2 = 'HRV';
    handles.l3 = 'Rw';
    Title = 'Neural Network Estimation';
end

% -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-

[r, c] = size(handles.p);

if ~isempty(findobj('Tag', 'Ri_Rw'))
    handles.h1name = figure((findobj('Tag', 'Ri_Rw')));
    set(handles.h1name, 'name', ['(x, y) = (', ...
        num2str(handles.x), ', ',num2str(handles.y),')'], ...
        'NumberTitle', 'off');

    plot(1:14, handles.p(:, 1), 'g', 1:14, handles.p(:, 2), 'b');
    legend(handles.l1, handles.l2)
    grid on

    if c == 3
        hold on
        plot(1:14, handles.p(:, 3), 'r')
        legend(handles.l1, handles.l2, handles.l3)
        grid on
        hold off
    end
    axis tight

    xlabel('Time', 'FontSize', 10, 'FontWeight','bold')
    ylabel('Reflection', 'FontSize', 10, 'FontWeight','bold')
    hold off
    title([Title; {['(x, y) = (', ...
        num2str(handles.x), ', ',num2str(handles.y),')']}; ...
        ...
        {['RMSE: Rw & HRV = ', num2str(handles.RMSE(1)), ...
        ', RHRV & HRV = ', num2str(handles.RMSE(2))]}], ...
        ...
        'FontSize', 11, 'FontWeight','bold')
%     title([Title], 'FontSize', 11, 'FontWeight','bold')
else
    handles.h1name = figure('Tag', 'Ri_Rw', ...
        'Name', ['(x, y) = (', ...
        num2str(handles.x), ', ',num2str(handles.y),')'], ...
        'NumberTitle', 'off');
    plot(1:14, handles.p1, 'g', 1:14, handles.p2, 'b');
    grid on
    legend(handles.l1, handles.l2)

    if c == 3
        hold on
        plot(1:14, handles.p(:, 3), 'r')
        legend(handles.l1, handles.l2, handles.l3)
        grid on
        hold off
    end
    axis tight

    xlabel('Time', 'FontSize', 10, 'FontWeight','bold')
    ylabel('Reflection', 'FontSize', 10, 'FontWeight','bold')
    hold off
    title([Title; {['(x, y) = (', ...
        num2str(handles.x), ', ',num2str(handles.y),')']}; ...
        ...
        {['RMSE: Rw & HRV = ', num2str(handles.RMSE(1)), ...
        ', RHRV & HRV = ', num2str(handles.RMSE(2))]}], ...
```

```
        ...
        'FontSize', 11, 'FontWeight','bold')
%     title([Title], 'FontSize', 11, 'FontWeight','bold')
end
guidata(hObject, handles);
return
% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

% --- Executes on button press in exit_pushbutton.
function exit_pushbutton_Callback(hObject, eventdata, handles)
% ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Figures = findobj('Name', 'Ri_Rw_plot');
if ~isempty(Figures)
    for i = 1:length(Figures)
%         close(figure(findobj('Name', 'Ri_Rw_plot')))
        close(figure(Figures(i)))
    end
end
return
% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@


% --- Executes on button press in clf_pushbutton.
function clf_pushbutton_Callback(hObject, eventdata, handles)
% ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
if ~isempty(findobj('Tag', 'Ri_Rw'))
    close(figure(findobj('Tag', 'Ri_Rw')))
end
return
% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

% --- Executes on button press in savefigure_pushbutton.
function savefigure_pushbutton_Callback(hObject, eventdata, handles)
% ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
if ~isempty(findobj('Tag', 'Ri_Rw'))
    h = figure((findobj('Tag', 'Ri_Rw')));
    saveas(h, ['images\', num2str(handles.x), ', ', num2str(handles.y)], 'fig')
    saveas(h, ['images\', num2str(handles.x), ', ', num2str(handles.y)], 'png')

    warndlg('The figure has been saved in both .png and .fig formates to "images" folder',...
        'Figure saving confirmation');
end
return
% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@


% --------------------------------------------------------------------
function home_menu_Callback(hObject, eventdata, handles)
% hObject    handle to home_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on slider movement.
function y_slider_Callback(hObject, eventdata, handles)
y_sliderValue = get(handles.y_slider, 'Value');
set(handles.y_edit, 'String', num2str(round(y_sliderValue)));
guidata(hObject, handles);

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
return


% --- Executes during object creation, after setting all properties.
function y_slider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to y_slider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
```

```
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
return


% --- Executes on slider movement.
function x_slider_Callback(hObject, eventdata, handles)
% ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
x_sliderValue = get(handles.x_slider, 'Value');
set(handles.x_edit, 'String', num2str(round(x_sliderValue)));
guidata(hObject, handles);
% @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
return


% --- Executes during object creation, after setting all properties.
function x_slider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to x_slider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
return


% --- Executes on mouse press over axes background.
function axes1_ButtonDownFcn(hObject, eventdata, handles)

%
a = get(gca,'Currentpoint');
a1 = round(a(1, 1));
a2 = round(a(1, 2));


if (a1 < 1 || a1 > 500)
    set(handles.x_edit, 'String', '250');
    set(handles.x_slider, 'Value', 250);
    a1 = 250;
else
    set(handles.x_edit, 'String', num2str(a1));
    set(handles.x_slider, 'Value', a1);
end

if (a2 < 1 || a2 > 400)
    set(handles.y_edit, 'String', '250');
    set(handles.y_slider, 'Value', 250);
    a2 = 250;
else
    set(handles.y_edit, 'String', num2str(a2));
    set(handles.y_slider, 'Value', a2);
end

if (get(handles.mouse_checkbox, 'Value'))
    axes(handles.axes1)
    h = plot(a1, a2, 'g+');
    set(h, 'HitTest', 'off')
    hold on
    % call plot_pushbutton_Callback function
    plot_pushbutton_Callback(hObject, [], handles)
else
    axes(handles.axes1)
    h = plot(a1, a2, 'g.');
    set(h, 'HitTest', 'off')
    hold on
```

```
end
guidata(hObject, handles);
return


% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% % hObject    handle to axes1 (see GCBO)
% % eventdata  reserved - to be defined in a future version of MATLAB
% % handles    empty - handles not created until after all CreateFcns called

% % Hint: place code in OpeningFcn to populate axes1
return
%


function KeyFunction(src, evnt)
handles = guidata(src);
k = evnt.Key;
if strcmp(k, 'return')
    pause(0.01)
    hObject = handles.plot_pushbutton;
    plot_pushbutton_Callback(hObject, [], handles);
end
return
%


% --- Executes on button press in rest_pushbutton.
function rest_pushbutton_Callback(hObject, eventdata, handles)
%
set(handles.x_edit, 'String', '250');
set(handles.x_slider, 'Value', 250);
set(handles.y_edit, 'String', '250');
set(handles.y_slider, 'Value', 250);

% ------------------------------------------------
axes(handles.axes1)
set(gca, 'NextPlot', 'replacechildren')
h = imagesc(handles.Map_land);
axis image
set(h, 'HitTest', 'off')
hold on

for i=50:50:450
    set(gca, 'NextPlot', 'add')
    h = line([i i], [1 400], 'color', 'black', 'LineStyle', ':');
    set(h, 'HitTest', 'off')
    hold on
end

for i=50:50:350
    set(gca, 'NextPlot', 'add')
    h = line([1 500], [i i], 'color', 'black', 'LineStyle', ':');
    set(h, 'HitTest', 'off')
    hold on
end

h = plot(250, 250, 'g.');
set(h, 'HitTest', 'off')
hold on
guidata(hObject, handles);
return
%


% --- Executes on button press in mark_pixel_pushbutton.
function mark_pixel_pushbutton_Callback(hObject, eventdata, handles)
%
local_x = str2num(get(handles.x_edit, 'String'));
local_y = str2num(get(handles.y_edit, 'String'));
```

```
h = plot(local_x, local_y, 'yo');
set(h, 'HitTest', 'off')
hold on
% guidata(hObject, handles);
return
% --- Executes when selected object is changed in radio_buttongroup.
function radio_buttongroup_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in radio_buttongroup
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%      EventName: string 'SelectionChanged' (read only)
%      OldValue: handle of the previously selected object or empty if none was selected
%      NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)
return


% --- Executes on button press in plot_pixel_pushbutton.
function plot_pixel_pushbutton_Callback(hObject, eventdata, handles)
%
pixel_x = str2num(get(handles.x_edit, 'String'));
pixel_y = str2num(get(handles.y_edit, 'String'));
h = plot(pixel_x, pixel_y, 'cx');
set(h, 'HitTest', 'off')
hold on
% guidata(hObject, handles);
return


% --------------------------------------------------------------------
function exit_menuItem_Callback(hObject, eventdata, handles)
% hObject    handle to exit_menuItem (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

close
return


% --- Executes on button press in mouse_checkbox.
function mouse_checkbox_Callback(hObject, eventdata, handles)
% hObject    handle to mouse_checkbox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of mouse_checkbox
return
```

g) Stepwise Model Classification Code:

TIME & DATE INITIALS 142

SETTING UP THE DIRECTORY & PREBUILTING A MATRIX FOR COLORMAP 143

LOADING DATA FROM FILES 143


```
tic

cc  % cc.m
Time & date initials
time_local = ['08:45', '09:15', '09:45', '10:15', '10:45', ...
    '11:15', '11:45', '12:15', '12:45', '13:15', ...
    '13:45', '14:15', '14:45', '15:15'];
```

```
time_GMT = ['05:45', '06:15', '06:45', '07:15', '07:45', ...
    '08:15', '08:45', '09:15', '09:45', '10:15', ...
    '10:45', '11:15', '11:45', '12:15'];
Time = 12:25;

Mon_name = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', ...
    'Sep', 'Oct', 'Nov', 'Dec'];
Day_num = [31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365];
% Day_mod = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
Day_cal = [0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334];

Setting up the directory & prebuilting a matrix for color map
diri = [pwd, '\'];  % assign the current working directory and \ to diri
diri2 = [diri, 'images\'];

% colormap data
gr = [1, 1, 0; ...  % yellow
    1,0,0; ...      % red
    0,0,1; ...      % blue
    1,1,1; ...      % white
    0.400000005960465,0.400000005960465, 0.400000005960465; ...    % gray
    0.400000005960465,0.800000011920929,0]; % green
% [green, gray, white, blue, red, yellow] = [3 ==> -2]

% load Caspian Sea's coastline data
load([pwd, '\Caspian_coastline_logical.mat']);

% new conditions for ch2 for 14 different time
ch2C1 = [318 285 255 236 221 216 217 225 234 247 262 284 311 338];
ch2C = zeros(400, 500, 14);
for i=1:14
    ch2C(:,:,i) = ch2C1(i);
end
Loading data from files ************************************
d_start = 59;
d_end = 59;
for d =d_start:d_end

    % call function Sim_R_In_All_Channels with input d
    Sim_R_In_All_Channels(d);
    % load RHRV
    load ([diri, 'Data\RHRV01', num2str(d, '%0.2d')]);
    % assign RHRV to Ch2
    Ch2 = RHRV;

    load ([diri,'Data\Matrix01' num2str(d,'%0.2d'),'07_1To4.mat']);
    Ch1 = eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:, 12:25, 1)']);
%     Ch2 = eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:, 12:25, 2)']);
    Ch3 = eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:, 12:25, 3)']);
    Ch4 = eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:, 12:25, 4)']);
    clear(['Matrix01' num2str(d,'%0.2d') '07_1To4']);

    load ([diri,'Data\Matrix01',num2str(d,'%0.2d'),'07_RAZ_SOL.mat']);
    s6 = ['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:, 12:25, 2)'];
    SAT = pi*eval(s6)/1800; clear s6
    clear (['Matrix01' num2str(d,'%0.2d') '07_RAZ_SOL']);

    load ([diri,'Data\Matrix01' num2str(d,'%0.2d'),'07_T04_T10.mat']);
    Ch9 =eval(['Matrix01', num2str(d,'%0.2d'), '07_T04_T10(:,:,12:25,2)']);
    Ch10=eval(['Matrix01', num2str(d,'%0.2d'), '07_T04_T10(:,:,12:25,3)']);
    clear(['Matrix01' num2str(d,'%0.2d') '07_T04_T10']);

    n = size(Ch1);  % [400, 500, 14] = [n(1), n(2), n(3)]

    %% NDSI ************************************************
%     Ch2(Ch2==1) = 3;

    NDSI = (Ch2-Ch4)./(Ch2+Ch4);    % NDSI, Water < 0.4, 0.4 < Ice

    %% SST ************************************************
    Ch9 = Ch9/10;
```

```
    Ch10 = Ch10/10;

    for k = 1:14
        if max(max(Ch9(:, :, k)))>320
            Ch9(:, :, k) = NaN;
        end
    end

    SST = Ch9 + (0.99*cos(SAT) + 0.21) .* (Ch9-Ch10) + ...
        (0.364./cos(SAT)+0.15) .* (Ch9-Ch10).^2 + ...
        (0.327./(cos(SAT)).^2 + 0.11);  % Ice < 271.5

    %% Classification steps
        Graph = NaN(n(1), n(2), n(3));

        Land = (Ch1==1);
        Cloud = (SST<240 | Ch4>200);
        Water = ((SST>271.5 | NDSI<0.3) & Ch2<ch2C);
        Ice = ((NDSI>=0.3 & SST<=271.5) | Ch2>=ch2C);
        clear Ch1 Ch2 Ch3 Ch4 SAT Ch9 Ch10 NDSI SST

        Graph(Water) = 0;   % blue
        Graph(Ice) = 1;     % white
        Graph(Cloud) = 2;   % gray
        Graph(Land) = 3;    % green
        clear Water Ice Cloud Land

        Unclassified = isnan(Graph);
        Graph(Unclassified) = -1;   % red
        clear Unclassified

        Graph(Caspian_coastline_logical) = -2;  % yellow

       %% date and month
        % returns the specific month of the day
        Mon_num = find(Day_num>=d, 1, 'first' );
        % Mon = min(find(Day_num>=d));

        % returns the name of the month
        Mon = Mon_name((Mon_num-1)*3+1:Mon_num*3);

        % calculates the date of the month
        Date = d - Day_cal(Mon_num);
%         if Date == 0
%             Date = 1;
%         end

        %% graphing
        for i = 1:n(3)
            % plotting graphs
            h = imagesc(Graph(:, :, i));
            colormap(gr)

            title({['Day #: ', num2str(d,'%0.2d'), '   :::::   ', ...
                Mon, ' ', num2str(Date, '%.2d'), ', 2007']; ...
                ['Time: ', num2str(Time(i), '%.2d'), ...
                '   :::::   GMT time: ', time_GMT(((i-1)*5+1):i*5), ...
                '   :::::   Local time: ', time_local(((i-1)*5+1):i*5)]}, ...
                'FontSize', 12, 'FontWeight','bold')

            t = ['images\Icemap', num2str(d, '%.2d'), num2str(i,'%.3d')];

            saveas(h, t, 'fig');
            %         movefile([t, '.fig'], diri2);
            saveas(h, t, 'png')

            clf     % clear figure
        end
end
close   % close figure
```

```
toc

return
%************************************************************************
```

h) Stepwise Simulation Model:

```
% diri='C:\FTP\ICE Research\';
diri = [pwd, '\data\'];

figure, clf
% File ------------------------------------------------------------------

for date1 = 23 %1:89
    in1=['Matrix01', num2str(date1,'%.2d'),'07_'];
    mat=getfield(load([diri,in1,'RAZ_SOL.mat']),[in1,'RAZ_SOL']);
    mask = getfield(load([diri,in1,'1To4.mat']),[in1,'1To4']);

    ARZ1=mat(:,:,12:25,1);
    SAT1=mat(:,:,12:25,2);
    SOL1=mat(:,:,12:25,3);
    mk = mask(:,:,12:25,1);

    ARZ=deg2rad(ARZ1/10);
    SAT=deg2rad(SAT1/10);
    SOL=deg2rad(SOL1/10);

    RHRV = zeros(400, 500, 14);

%HRV-1=Ch1
%  RHRV = (4071.4352220296)*ARZ + (59557.6640426094)*SOL + ...
%          (3593.01455611974)*cos(ARZ) + (81039.1431121054)*cos(SAT) + ...
%          (38416.1065580807)*cos(SOL) + (-1414.41141777918)*sin(ARZ) + ...
%          (-5436.96787777452) *sin(SOL) + (-2084.03727537616) *ARZ.*SOL + ...
%          (-444.265672018324)*ARZ./SOL + (2943.14445729393) *SAT./SOL + ...
%          (-45392.6405378837)*SOL./SAT + (354.916236745397)*ARZ./cos(SAT) +...
%          (25.6850377479388)*ARZ./cos(SOL)+ (1782.62370466617)*SOL./cos(SAT) + ...
%          (-4096.90194239776)*cos(ARZ).*cos(SAT) + (2214.9521210289)*cos(ARZ).*cos(SOL) +...
%          (-48327.3391734095)*cos(SAT).*cos(SOL) + (-378.417582101337) *sin(2*ARZ) + ...
%          (-350.616126860197)*cos(2*ARZ) + (-69063.2);

%R01-8=Ch2
%     RHRV = (-22157.5211798621)*ARZ + (-3646639.68742517)* SAT + (-191394.158777069)*SOL +
...
%          (2421.4906296704)*cos (ARZ) + (-2779127.48564793)*cos (SAT) + ...
%          (85321.2237197447)*cos (SOL) + (-770.002926702377) * sin (ARZ) + ...
%          (-17451.7706028269) * sin (SOL) + (-29396.4474204779)*(1./sin (SOL)) + ...
%          (11663.0322436565) * ARZ .* SAT + (-1319.67538571718) * ARZ .* SOL + ...
%          (172170.659950775) *SAT.*SOL + (13388.2993375045) * ARZ./ SAT + ...
%          (28084.2685358272)*SAT./SOL + (44515.1024103154) * SOL ./ SAT +  ...
%          (-3327.54863644377)*cos (ARZ).* cos (SAT) + (2164.98713797329)*cos (ARZ).*cos
(SOL) +  ...
%          (-138846.9568119)*cos (SAT).*cos (SOL) + ...
%          (-833.186144929572)* cos (ARZ) .*cos (SAT).*cos (SOL)  + ...
%          (-287.949051578011) *sin (2*ARZ) + (-190.700447957509)*cos (2*ARZ) + ...
%          (-538101.864547385)*cos (2*SAT) + ...
%          (26626.0061623959)*1./sin(2*SAT)+ (4889570);

     %R02-8=Ch3
%      RHRV =(2152.4680344739)*ARZ +( -20728623.9698639)* SAT + (-86673.4614278628)*SOL + ...
%          (2237.86072162681)*cos (ARZ) +(-18307387.0861547)*cos (SAT) + ...
%          (51118.9847558398)*cos (SOL) + (-701.804700097002 ) * sin (ARZ) + ...
%          (9021205.58144072) * sin (SAT) + (-8515.57276191785) * sin (SOL) + ...
%          (45511.4849349455)*(1./cos (SAT)) + (-1331.37173949849) * ARZ .* SOL + ...
%          (74665.312862983) *SAT.*SOL + (4473.41790094813)*SAT./SOL + ...
%          (46196.5599435315) * SOL ./SAT + (305.315312928979) *(ARZ./ cos(SAT)) + ...
%          (-2808.07917483622)*cos (ARZ).*cos (SAT) + (2367.6963137188)*cos (ARZ).*cos (SOL)
+ ...
%          (-44936.7046161723)*cos (SAT).*cos (SOL) + (-1320.64898016031)*cos (ARZ) .* cos
(SAT).*cos (SOL) + ...
```

```
%          (-305.100899306909) *sin (2*ARZ) + (-185.609236967029)*cos (2*ARZ) + (22914000);


%   R03=Ch4
    RHRV = (24.6388747996745)*ARZ + (-4.49788285206644) *sin (2*ARZ) + ...
         (6.89291960413664)*1./sin (2*SOL) + (1.78861);


    RHRV(mk==1) = NaN;

%     clear ARZ SAT SOL mk jj1
    for j=1:14
        clf
        imagesc(RHRV(:,:,j));
        k=['images\RHRV01',num2str(date1,'%.2d'),num2str(j,'%0.2d'),'.png'];
        print ('-dpng',k)
        clear h k
    end
end
save(['RHRV01',num2str(date1,'%.2d'),'.mat'],'RHRV');

% Download data from Matrix012307_1To4
load('C:\Documents and Settings\magryc\Desktop\Rouzbeh\Data\Matrix012307_1To4.mat');

%GRAPHS OF SIMULATED AND OBSERVED REFLECTANCE

% v1(1,:)=Matrix012307_1To4 (360, 100, 12:25, 4);
% v1(2, :)= RHRV (360, 100, :);
% v1=v1';

% v1(1,:)=Matrix012307_1To4 (320, 260, 12:25, 4);
% v1(2, :)= RHRV (320, 260, :);
% v1=v1';
%
% v1(1,:)=Matrix012307_1To4 (310, 330, 12:25, 4);
% v1(2, :)= RHRV (310, 330, :);
% v1=v1';
%
% v1(1,:)=Matrix012307_1To4 (150, 180, 12:25, 4);
% v1(2, :)= RHRV (150, 180, :);
% v1=v1';

% v1(1,:)=Matrix012307_1To4 (270, 360, 12:25, 4);
% v1(2, :)= RHRV (270, 360, :);
% v1=v1';

v1(1,:)=Matrix012307_1To4 (290, 290, 12:25, 4);
v1(2, :)= RHRV (290, 290, :);
v1=v1';

plot(v1, 'DisplayName', 'v1', 'YDataSource', 'v1'); figure(gcf);
xlabel('Acquisition Time');
ylabel('Reflectance (% *10)');
title('R03 (1.6um) Reflectance (January 23, 2007, pixel 290, 290)');
legend('Measured R03', 'Simulated R03');


% IMAGES - MAPS OF REFLECTANCES - SIMULATED + OBSERVED

% sim = RHRV(:,:,1);
% obs = Matrix012307_1To4(:,:,12,1);
%
% subplot(1,2,2); imagesc(sim);
% colorbar; xlabel('X');
% ylabel('Y');
% title('Simulated HRV (0.6-0.9 um) Reflectance, 8:45 of Local Time');
%
% subplot(1,2,1); imagesc(obs);
% colorbar; xlabel('X');
% ylabel('Y');
% title('Measured HRV (0.6-0.9 um) Reflectance, 8:45 of Local Time');
```

i) Neural Network base Hourly Ice Concentration Model:

```
clear all

load PS_water_Outputs;
load PS_water_Inputs;
load PS_Ice_Outputs;
load PS_Ice_Inputs;
load net_ice;
load net_water;

%Ice4=nan(200000,90);

% ********************ICE FRACTION DETERMINATION*************************

for d=59:59

load (['Matrix01' num2str(d,'%0.2d'),'07_1To4.mat']);
load (['Matrix01',num2str(d,'%0.2d'),'07_RAZ_SOL.mat']);

% Select variables ****************************************************

    Ch1=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,1)']);
    Ch2=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,2)']);
    Ch3=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,3)']);
    Ch4=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,4)']);
    clear(['Matrix01' num2str(d,'%0.2d') '07_1To4']);
    s5=['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:,:,1)'];
    ARZ=pi*eval(s5)/1800; clear s5
    s6=['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:,:,2)'];
    SAT=pi*eval(s6)/1800; clear s6
    s7=['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:,:,3)'];
    SOL=pi*eval(s7)/1800; clear s7
    clear (['Matrix01' num2str(d,'%0.2d') '07_RAZ_SOL']);

% Select only sun illuminated pixels **********************************

        Ch1=Ch1(:,:,12:25);
        Ch2=Ch2(:,:,12:25);
        Ch3=Ch3(:,:,12:25);
        Ch4=Ch4(:,:,12:25);
        ARZ=ARZ(:,:,12:25);
        SOL=SOL(:,:,12:25);
        SAT=SAT(:,:,12:25);
        n=size(Ch1);

% Vector conversion ***************************************************

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch1(:,:,j); A(:,j)=A1(:); end, Ch1=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch2(:,:,j); A(:,j)=A1(:); end, Ch2=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch3(:,:,j); A(:,j)=A1(:); end, Ch3=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch4(:,:,j); A(:,j)=A1(:); end, Ch4=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=ARZ(:,:,j); A(:,j)=A1(:); end, ARZ=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=SOL(:,:,j); A(:,j)=A1(:); end, SOL=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=SAT(:,:,j); A(:,j)=A1(:); end, SAT=A;
        clear A A1 j
```

```
% Step 1 : cloud detection ***********************************************

        Ice3=nan(n(1)*n(2),n(3));

        % Filter for Ch1 ~= 1
        hCh1=Ch1(:,1)~=1;
        h1=find(hCh1~=1); Ice3(h1,:)=3; clear h1
        h=find(hCh1==1);
        [d]
        for i=1:size(h,1)
            %[i]
            cloudfree=find(Ch4(h(i),:)<300);
            cl2=find(Ch4(h(i),:)>=300);
            if ~isempty(cloudfree)
                % Input format (nine entries): [ARZ SAT SOL cos(ARZ)
                % cos(SAT) cos(SOL) sin(ARZ) sin(SAT) sin(SOL)]
                input = [ARZ(h(i),cloudfree)' SAT(h(i),cloudfree)' SOL(h(i),cloudfree)'
cos(ARZ(h(i),cloudfree))' cos(SAT(h(i),cloudfree))' cos(SOL(h(i),cloudfree))'
sin(ARZ(h(i),cloudfree))' sin(SAT(h(i),cloudfree))' sin(SOL(h(i),cloudfree))'];
                input=input';

                % water simulation
                NORM_water_input = mapminmax('apply',input, PS_water_Inputs);
                norm_sim_water = sim(net_water,NORM_water_input);
                sim_water = mapminmax('reverse',norm_sim_water, PS_water_Outputs)';
                Rw=sim_water;

                % ice simulation
                NORM_ice_input = mapminmax('apply',input,  PS_Ice_Inputs);
                norm_sim_ice = sim(net_ice,NORM_ice_input);
                sim_ice = mapminmax('reverse',norm_sim_ice, PS_Ice_Outputs)';
                Ri=sim_ice;

                Robs=(Ch2(h(i),cloudfree)/10)';
                %Robs=(reshape(Robs,[size(Robs,2) size(Robs,3)]))';

                hRob=find(Robs<Rw);
                hRobg=find(Robs>Ri);
                if ~isempty(hRob), Robs(hRob)=Rw(hRob); end
                if ~isempty(hRobg), Robs(hRobg)=Ri(hRobg); end

                Ice2=(Robs-Rw)./(Ri-Rw);
                Ice3(h(i),cloudfree)=Ice2;
                Ice3(h(i),cl2)=2;
            else
                Ice3(h(i),:)=2;
            end
        end

        for ii=1:n(3)
            Ice4=reshape(Ice3(:,ii),[n(1) n(2)]);
            figure, imagesc(Ice4);
            clear Ch* Ice2 R* h* input norm_sim_ice norm_sim_water
            clear sim_ice sim_water ARZ SAT cloudfree SOL

            t=['Icemap',num2str(d,'%.2d'),num2str(ii+11,'%.2d'),'.png'];
            print ('-dpng',t)
            movefile(t,'C:\FTP\ICE Research\Temimi\all\');
            close
        end
end

%************************************************************************
```

j) Neural Network base Daily Ice Concentration Model:

```
clear all

load PS_water_Outputs;
load PS_water_Inputs;
load PS_Ice_Outputs;
load PS_Ice_Inputs;
load net_ice;
load net_water;

Ice4=nan(200000,90);

% ********************ICE FRACTION DETERMINATION*************************

for d=82:90
    if d~=61 || d~=81


load (['Matrix01' num2str(d,'%0.2d'),'07_1To4.mat']);
load (['Matrix01',num2str(d,'%0.2d'),'07_RAZ_SOL.mat']);

% Select variables ****************************************************

    Ch1=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,1)']);
    Ch2=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,2)']);
    Ch3=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,3)']);
    Ch4=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,4)']);
    clear(['Matrix01' num2str(d,'%0.2d') '07_1To4']);
    s5=['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:,:,1)'];
    ARZ=pi*eval(s5)/1800; clear s5
    s6=['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:,:,2)'];
    SAT=pi*eval(s6)/1800; clear s6
    s7=['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:,:,3)'];
    SOL=pi*eval(s7)/1800; clear s7
    clear (['Matrix01' num2str(d,'%0.2d') '07_RAZ_SOL']);

% Select only sun illuminated pixels **********************************

        Ch1=Ch1(:,:,12:25);
        Ch2=Ch2(:,:,12:25);
        Ch3=Ch3(:,:,12:25);
        Ch4=Ch4(:,:,12:25);
        ARZ=ARZ(:,:,12:25);
        SOL=SOL(:,:,12:25);
        SAT=SAT(:,:,12:25);
        n=size(Ch1);

% Vector conversion ***************************************************

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch1(:,:,j); A(:,j)=A1(:); end, Ch1=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch2(:,:,j); A(:,j)=A1(:); end, Ch2=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch3(:,:,j); A(:,j)=A1(:); end, Ch3=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch4(:,:,j); A(:,j)=A1(:); end, Ch4=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=ARZ(:,:,j); A(:,j)=A1(:); end, ARZ=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=SOL(:,:,j); A(:,j)=A1(:); end, SOL=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=SAT(:,:,j); A(:,j)=A1(:); end, SAT=A;
```

```
        clear A A1 j


% Step 1 : cloud detection ********************************************

        Ice3=nan(n(1)*n(2),1);

        % Filter for Ch1 ~= 1
        hCh1=Ch1(:,1)~=1;
        h1=find(hCh1~=1); Ice3(h1)=3; clear h1
        h=find(hCh1==1);
        [d]
        for i=1:size(h,1)
            %[i]
            cloudfree=find(Ch4(h(i),:)<200);
            if ~isempty(cloudfree)
                % Input format (nine entries): [ARZ SAT SOL cos(ARZ)
                % cos(SAT) cos(SOL) sin(ARZ) sin(SAT) sin(SOL)]
                input = [ARZ(h(i),cloudfree)' SAT(h(i),cloudfree)' SOL(h(i),cloudfree)'
cos(ARZ(h(i),cloudfree))' cos(SAT(h(i),cloudfree))' cos(SOL(h(i),cloudfree))'
sin(ARZ(h(i),cloudfree))' sin(SAT(h(i),cloudfree))' sin(SOL(h(i),cloudfree))'];
                input=input';

                % water simulation
                NORM_water_input = mapminmax('apply',input, PS_water_Inputs);
                norm_sim_water = sim(net_water,NORM_water_input);
                sim_water = mapminmax('reverse',norm_sim_water, PS_water_Outputs)';
                Rw=sim_water;

                % ice simulation
                NORM_ice_input = mapminmax('apply',input,  PS_Ice_Inputs);
                norm_sim_ice = sim(net_ice,NORM_ice_input);
                sim_ice = mapminmax('reverse',norm_sim_ice, PS_Ice_Outputs)';
                Ri=sim_ice;

                Robs=(Ch2(h(i),cloudfree)/10)';
                %Robs=(reshape(Robs,[size(Robs,2) size(Robs,3)]))';

                hRob=find(Robs<Rw);
                hRobg=find(Robs>Ri);
                if ~isempty(hRob), Robs(hRob)=Rw(hRob); end
                if ~isempty(hRobg), Robs(hRobg)=Ri(hRobg); end

                Ice2=(Robs-Rw)./(Ri-Rw);
                Ice3(h(i))=mean(Ice2);
            else
                Ice3(h(i))=2;
            end
        end
        Ice3=reshape(Ice3,[n(1) n(2)]);
        figure, imagesc(Ice3);
        % Ice4(:,d)=Ice3(:);
        save(['C:\FTP\ICE Research\Temimi\all cloudfree 2nd
test\Icemap',num2str(d,'%.2d'),'.mat'], 'Ice3');
        clear Ice3 Ch* Ice2 R* h* input norm_sim_ice norm_sim_water
        clear sim_ice sim_water ARZ SAT cloudfree SOL

        t=['Icemap',num2str(d,'%.2d'),'.png'];
        print ('-dpng',t)
        movefile(t,'C:\FTP\ICE Research\Temimi\all cloudfree 2nd test\');
    end
end
% ***************************************Step 4: Validation step
```

k) Neural Network base Multi Daily Ice Concentration Model:

```
clear all

load PS_water_Outputs;
load PS_water_Inputs;
load PS_Ice_Outputs;
load PS_Ice_Inputs;
load net_ice;
load net_water;

Ice4=nan(200000,90);

% ******************ICE FRACTION DETERMINATION*************************

for d=82:90
    if d~=61 || d~=81


load (['Matrix01' num2str(d,'%0.2d'),'07_1To4.mat']);
load (['Matrix01',num2str(d,'%0.2d'),'07_RAZ_SOL.mat']);

% Select variables ****************************************************

    Ch1=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,1)']);
    Ch2=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,2)']);
    Ch3=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,3)']);
    Ch4=eval(['Matrix01', num2str(d,'%0.2d'), '07_1To4(:,:,:,4)']);
    clear(['Matrix01' num2str(d,'%0.2d') '07_1To4']);
    s5=['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:,:,1)'];
    ARZ=pi*eval(s5)/1800; clear s5
    s6=['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:,:,2)'];
    SAT=pi*eval(s6)/1800; clear s6
    s7=['Matrix01', num2str(d,'%0.2d'), '07_RAZ_SOL(:,:,:,3)'];
    SOL=pi*eval(s7)/1800; clear s7
    clear (['Matrix01' num2str(d,'%0.2d') '07_RAZ_SOL']);

% Select only sun illuminated pixels **********************************

        Ch1=Ch1(:,:,12:25);
        Ch2=Ch2(:,:,12:25);
        Ch3=Ch3(:,:,12:25);
        Ch4=Ch4(:,:,12:25);
        ARZ=ARZ(:,:,12:25);
        SOL=SOL(:,:,12:25);
        SAT=SAT(:,:,12:25);
        n=size(Ch1);

% Vector conversion ***************************************************

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch1(:,:,j); A(:,j)=A1(:); end, Ch1=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch2(:,:,j); A(:,j)=A1(:); end, Ch2=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch3(:,:,j); A(:,j)=A1(:); end, Ch3=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=Ch4(:,:,j); A(:,j)=A1(:); end, Ch4=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=ARZ(:,:,j); A(:,j)=A1(:); end, ARZ=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=SOL(:,:,j); A(:,j)=A1(:); end, SOL=A;

        A=nan(n(1)*n(2),n(3));
        for j=1:n(3), A1=SAT(:,:,j); A(:,j)=A1(:); end, SAT=A;
```

```
        clear A A1 j


% Step 1 : cloud detection ********************************************

        Ice3=nan(n(1)*n(2),1);

        % Filter for Ch1 ~= 1
        hCh1=Ch1(:,1)~=1;
        h1=find(hCh1~=1); Ice3(h1)=3; clear h1
        h=find(hCh1==1);
        [d]
        for i=1:size(h,1)
            %[i]
            cloudfree=find(Ch4(h(i),:)<200);
            if ~isempty(cloudfree)
                % Input format (nine entries): [ARZ SAT SOL cos(ARZ)
                % cos(SAT) cos(SOL) sin(ARZ) sin(SAT) sin(SOL)]
                input = [ARZ(h(i),cloudfree)' SAT(h(i),cloudfree)' SOL(h(i),cloudfree)'
cos(ARZ(h(i),cloudfree))' cos(SAT(h(i),cloudfree))' cos(SOL(h(i),cloudfree))'
sin(ARZ(h(i),cloudfree))' sin(SAT(h(i),cloudfree))' sin(SOL(h(i),cloudfree))'];
                input=input';

                % water simulation
                NORM_water_input = mapminmax('apply',input, PS_water_Inputs);
                norm_sim_water = sim(net_water,NORM_water_input);
                sim_water = mapminmax('reverse',norm_sim_water, PS_water_Outputs)';
                Rw=sim_water;

                % ice simulation
                NORM_ice_input = mapminmax('apply',input,  PS_Ice_Inputs);
                norm_sim_ice = sim(net_ice,NORM_ice_input);
                sim_ice = mapminmax('reverse',norm_sim_ice, PS_Ice_Outputs)';
                Ri=sim_ice;

                Robs=(Ch2(h(i),cloudfree)/10)';
                %Robs=(reshape(Robs,[size(Robs,2) size(Robs,3)]))';

                hRob=find(Robs<Rw);
                hRobg=find(Robs>Ri);
                if ~isempty(hRob), Robs(hRob)=Rw(hRob); end
                if ~isempty(hRobg), Robs(hRobg)=Ri(hRobg); end

                Ice2=(Robs-Rw)./(Ri-Rw);
                Ice3(h(i))=mean(Ice2);
            else
                Ice3(h(i))=2;
            end
        end
        Ice3=reshape(Ice3,[n(1) n(2)]);
        figure, imagesc(Ice3);
        % Ice4(:,d)=Ice3(:);
        save(['C:\FTP\ICE Research\Temimi\all cloudfree 2nd
test\Icemap',num2str(d,'%.2d'),'.mat'], 'Ice3');
        clear Ice3 Ch* Ice2 R* h* input norm_sim_ice norm_sim_water
        clear sim_ice sim_water ARZ SAT cloudfree SOL

        t=['Icemap',num2str(d,'%.2d'),'.png'];
        print ('-dpng',t)
        movefile(t,'C:\FTP\ICE Research\Temimi\all cloudfree 2nd test\');
    end
end
% ***************************************Step 4: Validation step
```

9 PUBLICATIONS

Rouzbeh Nazari and Reza Khanbilvardi, Application of Dynamic Threshold in Development of an Advanced Technique for Mapping and Monitoring Sea and Lake Ice, Working Paper

ROUZBEH NAZARI, VANESSA CLARK and REZA KHANBILVARDI, Assessment of Sea Surface Temperature (SST) and Normalized Difference Sea Ice Index (NDSI) Derived from MSG SEVIRI Satellite for Sea Ice Applications, International Journal of Remote Sensing, Submitted on 29-Nov-2009

Rouzbeh Nazari, Magdalena Rychtecka, Hosni Ghedira and Reza Khanbilvardi, Stepwise Linear Regression for mapping and monitoring sea and lake ice for the future GOES-R, International Journal of Terraspace Science and Engineering , Volumn I, Issue 2, June, 2009

R. Nazari, NOAA-CREST, New York, NY; and Hosni Ghedira, M. Temimi, P. Romanov, and R. KhanbilvardiDevelopment and validation of a BRDF model for ice mapping for the future GOES-R Advanced Baseline Imager (ABI) using Artificial Neural Network, The 88th Annual Meeting (20-24 January 2008) (New Orleans, LA)

Rouzbeh Nazari, Marouane Temimi, Hosni Ghedira and Reza Khanbilvardi, NOAA-CRESTNew York City, New York, An automated approach for sea ice mapping and ice fraction determination for the future GOES-R Advanced Baseline Imager (ABI), 2008 IEEE International Geoscience & Remote Sensing Symposium, July 6-11, 2008, Boston, Massachusetts, U.S.A

R. Nazari, S. Mahani, and R. Khanbilvardi, Climate Changes Interaction with Tropical Storm (Hurricane), Fourth Annual NOAA-CREST Symposium, February 23-25 2006, Puerto Rico, Mayaguez

R. Nazari, S. Mahani, and R. Khanbilvardi, Changes in Sea Surface Temperature and North Atlantic Hurricane Activities, AGU Joint Assembly, May 23-25 2006, Baltimore, Maryland

R. Nazari, S. Mahani, and R. Khanbilvardi, Impacts of Climate Change and Tropical Storms (Hurricanes) on Coastal regions, 27th Conference on Hurricanes and Tropical Meteorology, 2006, Monterey, California

Nazari, Rouzbeh and Eslamian, S. S., Management, Optimization and Simulation for an Optimum Distribution of Water in Kalamarz Multi-Reservoir System, Mianeh Basin. 6th International Conference on Hydro science and Engineering (ICHE-2004), Brisbane, Australia, May 30-June 2, 2004.

Nazari, R., Tabatabaei H., Abedi Koupai, J., A Mathematical & Management Model of Ground Water with Emphasis on Artificial Recharge for Damaneh Plain, The Second International Conference on Salt Water Intrusion and Coastal Aquifers, Mirada, Mexico, 2003

Nazari, R. and Eslamian, S. S. Hydrological Homogeneity Test of Catchments Using L-moments Diagram, International Conference on the Rational Use and Conservation of Water Resources in a Changing Environment, Yerevan, Armenia, 2003.

10 REFERENCES

Alhumaidi, S. M., Jones, L., Park, J.-D., and Ferguson, S. M. (1997). "A neural network algorithm for sea ice edge classification." IEEE Transactions on Geoscience and Remote Sensing, 35(4), 817-826.

Atkinson, P. M., and Tatnall, A. R. L. (1997). "Neural networks in remote sensing." International Journal of Remote Sensing, 18(4), 699-709.

Benediktsson, J., Swain, P., and Ersoy, O. K. (1990). "Neural network approaches versus statistical methods in classification of multi-source remote sensing data." IEEE Transactions on Geoscience and Remote Sensing, 28(4), 540-552.

Chiu, S. L. (1997). "An Efficient Method for Extracting Fuzzy Classification Rules from High Dimensional Data." Journal of Advanced Computational Intelligence, 1(1), 1-7.

Congalton, R. G. (1991). "A review of assessing the accuracy of classifications of remotely sensed data." Remote Sensing of Environment, 37, 35–46.

Foody, G. M., and Arora, M. K. (1997). "An evaluation of some factors affecting the accuracy of classification by an artificial neural network." International Journal of Remote Sensing, 18(4), 799-810.

Jensen, J. R. (2000). Remote Sensing of the Environment: An Earth Resource Perspective, Prentice Hall.

Kanellopoulos, I., and Wilkinson, G. (1997). "Strategies and best practice for neural network image classification." International Journal of Remote Sensing, 18(4), 711-725.

Lippman, R. (1987). "An introduction to computing with neural nets." IEEE ASSP Magazine, 4, 4-22.

Nedeljkovic, I. "Image classification based on fuzzy logic." The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 74-79.

Paola, J., and Schowengerdt, R. A. (1995). "A review and analysis of back-propagation neural networks for classification of remotely-sensed multi-spectral imagery." International Journal of Remote Sensing, 16(16), 3033-3058.

Richards, J. A. (1996). "Classifier performance and map accuracy." Remote Sensing Environment, 57, 161-166.

Robinson, N. (1966). Solar Radiation, Elsevier Publishing Compony, Amsterdam.

Takagi, T., and Sugeno, M. (1985). "Fuzzy identification of systems and its application to modeling and control." IEEE Transactions on Systems, Man and Cybernetics, SMC-15, 116-132.

Topouzelis, K., Karathanassi, V., Pavlakis, P., and Rokos, D. "Neural Networks for oil spill detection using ERS-SAR Data." IEEE Transactions on geoscience, Istanbul, Turkey, July 2004, 724-728.

Wang, Y., and Jamshidi, M. "Fuzzy logic applied in remote sensing image classification." IEEE International Conference on Systems, Man and Cybernetics 10-13 Oct. 2004, 6378-6382.

MASUDA, K., TAKASHIMA, T. and TAKAYAMA, Y., 1988, Emissivity of pure and sea waters for the model sea surface in the infrared window regions. Remote Sensing of Environment, 24, pp. 313–329.

PHYSICAL OCEANOGRAPHY DAAC (PO.DAAC) website. http://podaac www.jpl.nasa.gov.

SOBRINO, J.A. and ROMAGUERA, M., 2004, Land surface temperature retrieval from MSG/SEVIRI data. Remote Sensing of Environment, 92, pp. 247–254.

WALTON, C.C., 1988, Nonlinear multichannel algorithms for estimating sea surface temperature with AVHRR satellite data. Journal of Applied Meteorology, 27, pp. 115–127.

Assel, R.A., T.E. Croley II and K. Schneider, 1994: Normal daily temperatures and ice cover of the Laurentian Great Lakes of North America, Abstract only, Proceedings of the 51st Eastern Snow Conference, 15-16 June 1994, Dearborn, MI.

Barry, R.G., 1986: The sea ice data base, In: The Geophysics of Sea Ice, Untersteiner, N. (ed.), Plenum Press, NY, pp 1099-1134.

Comiso, J.C., 1994: Surface temperatures in the polar regions from Nimbus-7 temperature humidity infrared radiometer. Journal of Geophysical Research, 99(C3), pp 5181-5200.

Comiso, J.C., P. Wadhams, W.B. Drabill, R.N. Swift, J.P. Crawford and W.B. Tucker III, 1991: Top/bottom multisensor remote sensing of arctic sea ice. Journal of Geophysical Research, 96(C2), pp 2693-2709.

Alexandrov, V. Yu., S. Sandven, O. M. Johannessen, Ø. Dalen, and L. H. Pettersson. 2000. Winter navigation in the Northern Sea Route. Polar Record 36, no. 199:333–342.

Bertoia, C., M. Manore, H. Steen Andersen, C. O'Connors, K. Q. Hansen, and C. Evanego. 2004. Synthetic aperture radar for operational ice observation and analysis. Chapter 20, pages 417–442 in Synthetic Aperture Radar. Marine User's Manual, edited by C. R. Jackson and J. R. Apel. Washington, D.C.: National Oceanic and Atmospheric Administration.

Bjørgo, E., O. M. Johannessen, and M. W. Miles. 1997. Analysis of merged SMMR/SSMI time series of Arctic and Antarctic sea ice parameters. Geophys. Res. Lett. 24:413–416.

Breivik, L.-A., S. Eastwood, Ø. Godøy, H. Schyberg, S. Andersen, and R. Tonboe. 2001. Sea ice products for EUMETSAT satellite application facility. Can. J. Rem. Sens. 27, no. 5:403–410.

Campbell, W. J. 1973. NASA remote sensing of sea ice AIDJEX. Proceedings of the World Meteorological Organization Technical Conference, Tokyo, Japan, WMO No. 350, 55– 56.

Caneva, N. 1992. ERS-1 SAR Calibration. Frascati, Italy: European Space Agency.

Carsey, F. D., ed. 1992. Microwave Remote Sensing of Sea Ice. AGU Geophysical Monograph 68. Washington, D.C.: American Geophysical Union.

Cavalieri, D. J. and J. C. Comiso. 2000. Algorithm Theoretical Basis Document for the AMSR-E Sea Ice Algorithm, revised December 1. Landover, Md.: Goddard Space Flight Center.

Cavalieri, D. J., P. Gloersen, and W. J. Campbell. 1984. Determination of sea-ice parameters with the NIMBUS-7 SMMR. J. Geophys. Res. 89:5355–5369.

Cavalieri, D. J., J. P. Crawford, M. R. Drinkwater, D. T. Eppler, L. D. Farmer, R. R. Jentz, and C. C. Wackerman. 1991. Aircraft active and passive microwave validation of sea ice from the Defense Meteorological Satellite Program Special Sensor Microwave Imager. J. Geophys. Res. 96:21989–22008.

Cavalieri, D. J., P. Gloersen, C. E. Parkinson, H. J. Zwally, and J. C. Comiso. 1997. Observed hemispheric asymmetry in global sea ice changes. Science 278:1104–1106.

Comiso, J. C. 1983. Sea ice emissivities from satellite passive microwave and infrared observations. J. Geophys. Res. 88:7686–7704.

D. M. A. Aminou, B. Jacquet and F. Pasternak, ‘Characteristics of the Meteosat Second Generation Radiometer/Imager :SEVIRI’ Proceedings of SPIE, Europto series, Vol. 3221 pages 19-31.

D. M. A. Aminou, A. Ottenbacher, B. Jacquet and A. Kassighian, ‘Meteosat Second Generation : On-ground Calibration, Characterisation and Sensitivity Analysis of the SEVIRI Imaging Radiometer’. Proceedings of SPIE « Earth Observing Systems IV », Vol. 3750, page 419-430.

Earth Science Data and Information System (ESDIS). 1996. EOS Ground System (EGS) systems and operations concept. Greenbelt, MD: Goddard Space Flight Center.

Hall, D.K., J.L. Foster, D.L. Verbyla, A.G. Klein, and C.S. Benson. 1998. Assessment of snow cover mapping accuracy in a variety of vegetation cover densities in central Alaska. Remote Sensing of the Environment 66: 129-137.

Hall, D.K. and J. Martinec. 1985. Remote sensing of ice and snow. London: Chapman and Hall.

Hall, D.K., G.A. Riggs, and V.V. Salomonson. 1995. Development of methods for mapping global snow cover using Moderate Resolution Imaging Spectroradiometer (MODIS). Remote Sensing of the Environment 54(2): 127-140.

Hall, D.K., G.A. Riggs, and V.V. Salomonson. September 2001. Algorithm Theoretical Basis Document (ATBD) for the MODIS Snow-, Lake Ice- and Sea Ice-Mapping Algorithms. Greenbelt, MD: Goddard Space Flight Center.

Key, J.R., J.B. Collins, C. Fowler, and R.S. Stone. 1997.High latitude surface temperature estimates From thermal satellite data. Remote Sensing of the Environment 61:302-309.

Key, J.R., J.A. Maslanik, T. Papakyriakou, M.C. Serreze, and A.J. Schweiger. 1994. On the validation of satellite-derived sea ice surface temperature. Arctic 47: 280-287.

Van Leeuwen, B., 2002, “Engineering approach to water level uncertainties”, Chapter 9 in 'Proceedings of the 1st Workshop on the Caspian Sea', PHAJM van Gelder, ed., TU Delft, The Netherlands, pp. 139-152.

Farzaneh M., Savadjiev K., Druez J., 2001, “Icing event occurrence in Quebec: Statistical analysis of field data”, International journal of Offshore and Polar Engineering 11 (1) pp. 9-15.

Blanchet D., 1998, “Ice loads from first-year ice ridges and rubble fields”, Canadian Journal of Civil Engineering 25 (2), pp. 206-219.

Arrigo, K.R., Robinson,D.H., Worthen, D.L., Dunbar, R.B., DiTullio, G.R, VanWoert,

M. & Lizotte, M.P. (1999) Phytoplankton community structure and the drawdown of nutrients and CO2 in the Southern Ocean, Science, 283, pp. 365-367.

Blome, D. & Riemann, F. (1999) Antarctic sea ice nematodes, with description of Geomonhystera glaciei sp. nov. (Monhysteridae). Mitteilungen aus dem Hamburgischen Zoologischen Museum und Institut, 96, 15-20.

Bornemann, H., Kreyscher, M., Ramdohr, S., Martin, T., Carlini, A., Sellmann, L. & Plötz, J. (2000) Southern elephant seal movements and Antarctic sea ice. Antarctic Science, 12, 3-15.

Brieivk, L.-A., S. Eastwood, Ø. Godøy, H. Schyberg, S. Andersen, and R. Tonboe, Sea ice products for EUMETSAT Satellite Application Facility, Canadian Journal of Remote Sensing, Vol. 27, no. 5, 2001

Carrieres, T. and Others, An International Collaborative Effort Towards Automated Sea Ice Chart Production, Draft Working Document of the IICWG Science Subcommittee, 9 pp., April 2, 2003.

CEOS, The Use of Earth Observing Satellites for Hazard Support: Assessments and Scenarios, Final Report of the CEOS Disaster Management Support Group, NOAA, August 2002.

Barber, D., M. Manore, T. Agnew, H. Welch, E. Soulis, and E. LeDrew (1992), Science Issues Relating to Marine Aspects of the Cryosphere: Implications for Remote Sensing. Cdn J. of R.S., vol 18, No. 1. pp. 46-54.

Bertoia, C. and J. Falkingham, F.Fetterer (1998), Polar SAR Data for Operational Sea Ice Mapping, in R. Kwok and C. Tsatsoulis (Eds.). Recent Advances in the Analysis of SAR Data of the Polar Oceans, Springer Verlag: Berlin, pp. 201-234.

Bertoia,C., Michael Manore, Henrik Steen Andersen, Chris O'Connors, Keld Q. Hansen, Craig Evanego (To be published 2004), "Synthetic Aperture Radar Marine User's Manual", National Oceanic and Atmospheric Administration, Center for Satellite Application and Research, NOAA/NESDIS, C.R. Jackson and J.R. Apel, editors, Washington, D.C., USA.

CEOS (2001), "Ice Hazards", Final Report of the CEOS Disaster Management Support Group.

Duchoissois, G. and G. Sommeria (2003), WRCP Satellite Working Group Report on Update of Space Mission Requirements for WCRP.

European Environment Agency (2004), "Arctic Environment: European Perspectives".

Goss Gilroy Inc. (2001) "Economic Benefits from the Utilisation of RADARSAT-1 for Surveillance of Ice Conditions in Canada".

O. Cula and K. Dana, .Compact representation of bidirectional texture functions,. in Proc. IEEE Conf. on Comp. Vision and Patt. Recog., Kauai, Hawaii, December 2001.

K. Dana, B. Van Ginneken, S. Nayar, and J. Koenderink, .Re_ectance and texture of real-world surfaces,. ACM Transactions on Graphics, vol. 18, no. 1, pp. 1.34, January 1999.

A. Efros and W. Freeman, .Image quilting for texture synthesis and transfer,. in SIGGRAPH, Los Angeles, CA, August 2001, pp. 341.346.

A. Efros and T. Leung, .Texture synthesis by non-parametric sampling,. in ICCV, September 1999.

R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, .Appearance based object modeling using texture database: Acquisition, compression and rendering,. in Eighth Eurographics Workshop on Virtual Environments, 2002.

J. Koenderink and A. van Doorn, .Illuminance texture due to surface mesostructure,. J. Opt Soc. Am., vol. 13, no. 3, pp. 452.462, March 1996.

Darvishzadeh, R., et al., LAI and chlorophyll estimation for a heterogeneous grassland using hyperspectral measurements. ISPRS Journal of Photogrammetry & Remote Sensing (2008), doi:10.1016/j.isprsjprs.2008.01.001

Duckworth, J., 1998. Spectroscopic quantitative analysis. In: Workman, J., Springsteen, A. (Eds.), Applied Spectroscopy: A Compact Reference for Practitioners. Academic Press, San Diego, pp. 93–164.

Dunagan, S.C., Gilmore, M.S., Varekamp, J.C., 2007. Effects of mercury on visible/near-infrared reflectance spectra of mustard spinach plants (Brassica rapa P.). Environmental Pollution 148 (1), 301–311.

Haan C. T., 2002. Statistical methods in Hydrology, second edition. Iowa State University Press, Ames, Iowa.

Chokmani K., Ouarda T.B.M.J., Hamilton S., Ghedira M.H., Gingras H., 2008. Comparison of ice-affected streamflow estimates computed using artificial neural networks and multiple regression techniques. Journal of Hydrology 349, 383-396.

Sergent, M., Mathieu, D., Phan-Tan-Luu, R., Drava, G., 1995. Correct and incorrect use of multilinear regression. Chemometrics and Intelligent Laboratory Systems 27 (2), 153–162.

Snee R.D., 1977. Validation of regression models: Methods and examples, Technometrics 19, 415-428.

Vinod, H.D., 1976. Application of new ridge regression methods to a study of Bell system scale economies. Journal of the American Statistical Association 71, 835–841.

Benediktsson, J. A., P. H. Swain, and O. K. Ersoy, 1990: Neural network approaches versus statistical methods in classification of multi-source remote sensing data. *IEEE Trans. on Geoscience and Remote Sensing*, **vol. 28,** 540-552.

Ghedira, H., R. Khanbilvardi, and P. Romanov, 2007: Development of an advanced technique for mapping and monitoring sea and lake ice for the future GOES-R Advanced Baseline Imager (ABI). *IGARSS'2007, IEEE International Geosciences and Remote Sensing Symposium*, Barcelona, Spain.

Ghedira, H., J. C. AREVALO, T. Lakhankar, and R. Khanbilvardi, 2005: An Adaptive Neural Network System for Improving the Filtration of Non-Snow Pixels from SSM/I Images *IGARSS'2005, IEEE International Geosciences and Remote Sensing Symposium*, Seoul, Korea, pages 5459

Hillger, D. W., 2006: GOES-R ABI New Product Development. . *Proceedings of the 14th Conference on Satellite Meteorology and Oceanography.* , Atlanta GA, USA,.

Kouraev, A. V., Fabrice Papa, Nelly M. Mognard, Petr I. Buharizin,Anny Cazenave, Jean-Francois Cretaux, Julia Dozortseva, Frederique Remy., 2004. : Sea ice cover in the Caspian and Aral Seas from historical and satellite data. . *Journal of Marine systems.* , **Vol. 47.,** 89-100.

Kwok, R., D. A. Rothrock, H. L. Stern, and G. F. Cunningham, 1995: Determination of the age distribution of sea ice from Lagrangian observations of ice motion. *IEEE Transactions on Geoscience and Remote Sensing*, **vol.33,** pp.392–400.

Paola, J. D., and R. A. Schowengerdt, 1995: A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery. *International Journal of Remote Sensing*, **16,** 3033-3058.

Ramsay, B. H., 1998: The interactive multisensor snow and ice mapping system. *Hydrological Processes*, **vol. 12,** pp.1537-1546.