Towards Identification of Environmentally Vulnerable Areas to Coral Bleaching Using Remote Sensing and GIS in the Coral Triangle

Muhammad Barmawi

February, 2009

Towards Identification of Environmentally Vulnerable Areas to Coral Bleaching Using Remote Sensing and GIS in the Coral Triangle

by

Muhammad Barmawi

Thesis submitted to the International Institute for Geo-information Science and Earth Observation in partial fulfillment of the requirement for the degree of Master of Science in Geo-information Science and Earth Observation, specialization: Natural Resources Management

Thesis Assessment Board

Chairperson	: Dr. Ir. C.A.J.M. de Bie (Natural Resources Departement ITC)
External Examiner	: Dr. J. Janssen (Wageningen University)
First Supervisor	: Valentijn Venus, M.Sc. (Natural Resources Departement ITC)
Second Supervisor	: Drs. Joan Looijen (Natural Resources Departement ITC)

ITC

INTERNATIONAL INSTITUTE FOR GEO-INFORMATION SCIENCE AND EARTH OBSERVATION
ENSCHEDE, THE NETHERLANDS

Disclaimer

Abstract

The goal of this research is to test whether the bleaching vulnerability models (VM1 and VM2) that were developed in the Western Indian Ocean (WIO) gives satisfactory result when applied to the Coral Triangle compare against MaxEnt model (VM3). Coral bleaching observation data from Reefbase was used as the reference data. The environmental variable inputs to the models are sea surface temperature and its derived variables, chlorophyll-a concentration, photosynthetically active radiation (PAR), ultraviolet radiation, surface wind speeds, and ocean currents. Data collection and processing were done using distributed data access technologies.

Model parameters for the WIO were applied to the environmental variables in the Coral Triangle. The same variables were used to model bleaching vulnerability using Maximum Entropy modeling. Model comparison was done by utilizing the Reefbase bleaching observation data.

This research found the Reefbase bleaching observation data contains high variation because of the heterogeneous data sources. Chlorophyll-a and PAR showed negative relationship with the bleaching severity. The model outputs revealed that VM1 and VM2 models performed indifferently significance, but not to the VM3 model. The machine learning model provides better accuracy than model-based expert knowledge.

Keywords: coral bleaching, marine protected area, vulnerability, Maximum Entropy, Western Indian Ocean, Coral Triangle, IDV.

Acknowledgements

First of all, I thank to the Almighty Allah SWT for His unlimited blessing on me during all the time at ITC and also on my family.

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. I want to thank the Netherlands Education Support Office (NESO) Indonesia, formerly the Netherlands Education Centre (NEC), for giving me an opportunity to study in the Netherlands through StuNED scholarships scheme.

Thanks to my supervisors, Valentijn Venus, for his guidance and support during this research, and Joan Looijen for her comments and suggestions. I gratefully acknowledge the contribution of Willem Nieuwenhuis for his help on Jython scripting.

Thanks to Dr. Michael Weir, the NRM Course Director, to Bas Wesselman, the Student Advisor, and all Natural Resources Department ITC staff who be helpful and encouraging.

Thanks to all my colleagues in NRM course 2007-2009, especially cluster #4-010 for the discussion and sharing thoughts.

Thanks to Indonesia students in ITC and some Indonesia people living in the cities in the Netherlands. Our *ukhuwah* made my study period here full with experiences.

Thanks to my mother and father for their endless praying, to my lovely wife, dik Neni, for her *sabar, doa*, and taking care of my dear daughter, Rara. You were keeping me in spirit for this study.

Enschede, February 2009

Muhammad Barmawi

Table of contents

List of figures

List of tables

1. Introduction

1.1. Background

Coral reefs are the most important marine ecosystem which provides invaluable ecosystem services for human in direct or indirect way (Grimsditch, *et al.*, 2006). In coral reefs almost all of consumptive fish are found. Fish are integral part of coral reef ecosystems (Sale, 2002). In addition, coral reefs provide coastal protection, exotic environment for human live and also could provide value of US$30 billion to economies worldwide (Cesar, *et al.*, 2003). Coral reefs are like a tropical forest under the sea.

Coral reefs require a certain condition to survive. Beyond the suitable condition, it will have stress or event death. Threats to coral reefs are destructive fishing, pollution, waste disposal, coastal development, sedimentation, SCUBA diving, anchor damage, predator outbreak, invasive species, endemic diseases and bleaching (Grimsditch, *et al.*, 2006). The last threat has been destroyed 16% (Cesar, *et al.*, 2003) of coral reefs world wide in 1998 as the impact of El-Nino (Carriquiry, *et al.*, 2001). The same event has been recorded occur in the following years such as in the Great Barrier Reefs, Australia in 2002 (Berkelmans, *et al.*, 2004). It seems the bleaching event could potentially happen frequently in the future according to global warming.

Coral bleaching is a condition when the symbiosis between coral polyps and alga, zooxanthellae, is disturbed. The alga provide up to 95% of energy for coral reefs to grow, reproduce and feed (Cesar, *et al.*, 2003). Losing the alga will cause the coral reef bleach (whitening) and if the condition is prolong, they could die. If the coral reefs die, there will be subsequence serious consequence off losing coral reef ecosystem (Schuttenberg, 2001) which plays important role for marine environment as they become vulnerable to further structural degradation by algal overgrowth and bioerosion (Grimsditch, *et al.*, 2006). Coral bleaching has also socioeconomic impact. In the short term, tourism and fisheries will lose their main resource and also coastal protection by the coral reefs (Schuttenberg, 2001).

The elevated sea surface temperature has been suggested as the main cause of coral bleaching, beside other factors such as nutrient enrichment, disease, ultraviolet radiation. The last mass bleaching event in 1998 was coincided with El Nino event. This supports the statement that coral bleaching is caused by temperature increase (McClanahan, 2004). But scientists noted that there were also bleaching event happened in several regions (International Marine Sanctuaries; Berkelmans, *et al.*, 2004; McClanahan, *et al.*, 2005; Franklin, *et al.*, 2006; Ulstrup, *et al.*, 2006; Oxenford, *et al.*, 2008) which are not related to El Nino. This leads scientists to model coral bleaching to make better management response to possible future bleaching event due to global warming. The National Oceanic and Atmospheric Administration (NOAA) has recently launched the Coral Reef Watch that utilizes NOAA weather satellite to monitor physical environmental condition, predominantly sea surface temperature to help the management of coral reefs ecosystems. Their monitoring is intended for global scale as they produce sea surface temperature in 50 km resolution. Higher spatial resolution could be investigated to implementation in the regional scale.

Recently, in December 2007, the Governments of 6 countries in the Coral Triangle have agreed upon establishing international partnership to protect coral reefs in the region. Three largest conservation organizations, The Nature Conservancy (TNC), World Wide Fund for Nature (WWF) and Conservation International (CI) have put supports for initial start-up activities by pledged half a million U.S. dollars. This commitment and the facts have only increase the important to protect the marine ecosystems including coral reefs in the region (WWF, 2007).

Protection of the coral reef ecosystem could be conducted through marine protected area (MPA) tool. One of the important criteria of an area for MPA is low of natural catastrophes (Roberts*, et al.*, 2003), such as bleaching. Identification of areas which are vulnerable to bleaching is useful for success of MPA.

Ecological modeling to identify vulnerable areas to coral bleaching using remote sensing and GIS has been initiated by several researchers (Wooldridge*, et al.*, 2004; Baker*, et al.*, 2008; Buddemeier*, et al.*, 2008). In the Western Indian Ocean (WIO), a susceptibility model (SM) was developed to study the relative importance of environmental variables influencing coral bleaching (Maina*, et al.*, 2008). The researcher used susceptibility term instead of vulnerability. Both words area similar and seems to be used interchangeably. However, clear their definition is presented here. The Oxford Advance Learner's Dictionary defines susceptibility as the state of being very likely to be influenced, harmed or affected by something while vulnerability does not define clearly. Furthermore, Massachusetts Institute of Technology clearly differentiate susceptibility with vulnerability. Susceptibility is used where the increased risk is related to endogenous factors while vulnerability is related to external factor. SM seems to incorporate many environmental variables to predict coral bleaching. At this point, vulnerability term is preferred to be used in replacing susceptibility with the same meaning.

Furthermore, advance in computer processing allows the use of complex statistical analysis for predictive modelling. Maximum Entropy (MaxEnt) has been introduced as a machine learning-based model to measure species geographic distribution probability. The model uses species presence/ absence occurrence data to predict the species distribution based on environmental variable inputs (Phillips*, et al.*, 2006). This machine learning-based model is explored here to model coral bleaching vulnerability.

1.2. Research Problems

The need of establishing Marine Protected Areas (MPAs) in the Coral Triangle is high. The MPA sites should be located in such area where the degree of vulnerability to coral bleaching is least. Therefore the information about the vulnerability areas is becoming important. Applying an existing vulnerability model is the easiest way to provide the information but it should perform well (Figure 1).

In the Western Indian Ocean, vulnerability models were developed to identify specific areas with environmental variables that results in low bleaching and mortality (Maina*, et al.*, 2008). The model has not been tested for another area. The model was trained using in situ data and the environmental variables of the area. How the model performs in another area is an interesting research area explored in this study. Applying of the model to another area could improve the reliability of the model. MaxEnt modeling could potentially be used to identify vulnerable areas to coral bleaching in comparison to the existing models. It predicts species geographic distribution based on analysis on environmental variables.

The previous study used environmental data: sea surface temperature (SST), wind speeds, surface current, solar ultraviolet radiation (UV), photosynthetically active radiation (PAR) and chlorophyll-a concentration which are collected using conventional internet protocols such as FTP. The data should be stored in a local disk before processing. This is not efficient as it is slowing down processed data availability and also lot of works when there are new data available. All processing steps should be conducted again. Advances in the internet technology provide opportunity of data access sharing through internet protocol. Data are no longer necessary to be transferred to a local disk before data processing. A desktop computer installed with a data access client could send a request to a data server anywhere for a specific data (only area of interest) then retrieve the data or send instructions (data analysis) to the server for doing specific analysis then retrieve the output. IDV is one of distributed data access client freely available under General Public License (GPL). It has basic data analysis routines and provides high flexibility to advance them through Jython scripts. Once data analysis routines established, more data inputs including newly available can be processed.

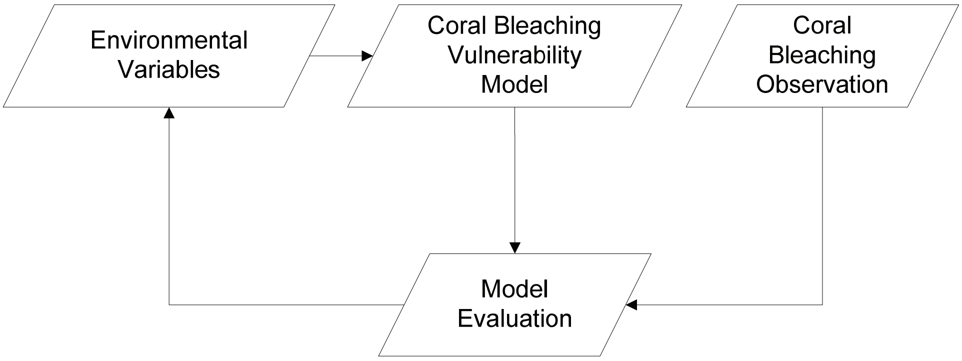


Figure 1. Conceptual framework

1.3. Research Approaches

Two models, vulnerability models (VM 1 and VM2) originally developed for the Western Indian Ocean (WIO) (Maina, 2007) were applied to the Coral Triangle. Input variables and model parameters were kept the same. The same environmental variables were used in the MaxEnt modelling to create the third a vulnerability model of coral bleaching (VM3). Comparison between these models and to the bleaching observation data extracted from Reefbase global bleaching database was also done. The best model could potentially be used to identify vulnerable areas to coral bleaching. Details research processes were given in the chapter 2.

This study used extensively Integrated Data Viewer (IDV) for data collection and processing. Data analysis advancement was explored using Jython scripting language, VisAD Java data model and Apache.org mathematical functions (see Appendix C). ERDAS Imagine Graphical Model (GMD) was used to further process MaxEnt model outputs. ESRI ArcGIS version 9.3 was utilized for maps productions.

1.4. Research Objectives

The overall objective of this study is identify areas vulnerable to coral bleaching using remote sensing and GIS in the Coral Triangle area.

The specific objectives are:

1. To evaluate the vulnerability models developed for the Western Indian Ocean applied in the Coral Triangle area.
2. To determine whether a model-based expert system provides better prediction than machine learning-based model
3. To provide a preliminary map showing vulnerable areas to coral bleaching in the Coral Triangle.

1.5. Research Questions

1. How do the vulnerability models developed for the Western Indian Ocean (WIO) perform in the Coral Triangle area without calibration?
2. Does the model-based expert system provide better prediction than machine learning-based model?

1.6. Research Hypothesis

<u>Hypothesis 1:</u>

Three different models were developed using different approaches on predicting environmental vulnerability area to coral bleaching. We test whether the models are significantly difference between each other or the other way around.

H_o = The accuracy difference between the models is significant. (VM1 = VM2 = 0, VM1 = VM3 = 0, VM2 = VM3 = 0)

H_a = The accuracy difference between the models is not significant. (VM1 <> VM2, VM1 <> VM3, VM2 <> VM3)

<u>Hypothesis 2:</u>

MaxEnt model used mathematical formulation (machine learning) to predict probability of species occurrence without so much interaction from the users. On the other hand, two models from the Western Indian Ocean (WIO) used human knowledge (expert system) to preprocess and interpret the model. We test whether human knowledge improves the model accuracy or the other way around.

H_o = Model attributable accuracy of the expert system-based model is equal to the machine learning–based model.

H_a = Model attributable accuracy of the expert system-based model is higher than the machine learning –based model.

2. Methods

2.1. Study Area

The study area of the research is in the Coral Triangle located in the Western Indo-Pacific area (Figure 2). Geographically it spans from 112° E - 165° and 20° N - 15° S covering eastern part of Indonesia, Malaysia, Philippines, Papua New Guinea (PNG), Solomon and Timor-Leste, Palau, and Macronesia. The area is characterized by its unique (thousands) islands structure along the Equator line. The temperature differences do not vary during a year ranging from 20° to 30°C. Precipitation is varied in difference places. Some places have very heavy rainfall throughout the year, some places are very dry. There are two seasonal variations which are caused by its location between Australia and Asia. West monsoon which happened during December, January and February brings heavy rain to western part of Indonesia. East monsoon happens during June, July and August which brings dry air from Australia. Thunderstorms frequently occur in the afternoon but strong cyclone and typhoons do not.

The Coral Triangle is the centre of the coral reef biodiversity in world. It was delineated in the expert workshop by The Nature Conservancy in 2003.. The workshop confirmed that the Coral Triangle area is the world's epicenter of marine life abundance and diversity (Green, *et al.*, 2004).with the largest coral reef surface area (Pet-Soede, *et al.*, 2007). The surface area is about 5.7 million km^2 (2% world ocean) but 75% of all coral species are found there. Moreover, more than 3,000 species of reef fish and commercially valuable pelagic species, six of the seven species of marine turtles, migrating whale sharks and manta rays, marine mammal including 22 species of dolphin, endangered dugong, Bryde's whale, short-finned pilot whale, three species of sperm whale, humpback whale, Cuvier's and Blainville's beaked whale and the world's least studied cetacean – Longman's beaked whale are found here (WWF, 2008).

The Coral Triangle marine ecosystem has supported about 150 million people in the surrounding area. They depend for their livelihood on marine resources. In 2002, it was recorded that the reef-based fisheries has value of $810 million. More than 75 percent of the world's aquaculture industry is centered in the Coral Triangle (WWF, 2007). The economic value generated from the area reaches estimation to US$2.3 billion annually.

Destruction of the marine resources has been reported in the Coral Triangle. Over-fishing and destructive fishing techniques have destroyed large part of the reefs. Coral bleaching also occurred in the region. During the last mass bleaching event in 1998, it was reported low and severe bleaching condition in several areas throughout the region (Goldberg, *et al.*, 2004; Pet-Soede, *et al.*, 2007). Some areas have been recovered from the mass bleaching, therefore it shows having good resilient to the bleaching threat.

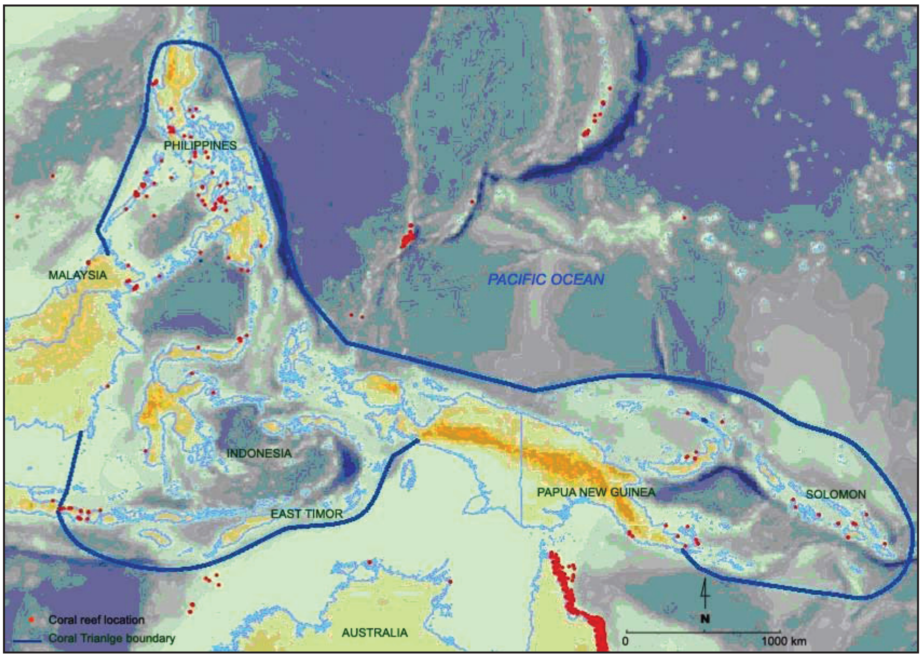


Figure 2. The study area: the Coral Triangle

2.2. Data

Data used in this research are collected according to previous study on vulnerability modelling in the Western Indian Ocean (Maina, 2007). Two data categories, coral bleaching observation and environmental data, were downloaded from the internet servers (Figure 3 and Appendix A).

2.2.1. Coral Bleaching Observations

The coral bleaching data were downloaded from the Reefbase. There are 504 point observations of coral bleaching from 1992 to 2006 in the study area. Data were collected from various reports from many research organizations and individual experts on coral reefs such as Global Coral Reef Monitoring Network (GCRMN), National Oceanic and Atmospheric Administration (NOAA), Reef Check, Great Barrier Reef Marine Protected Area (GBRMPA), and Diver/Tourism operators. The bleaching data was categorized into no-bleaching, low, medium and high severity. Not all of the data were used in this study. Selection was made to avoid over representation. 275 out of 504 were localized in the Great Barrier Reef (GBR) region which only covers less than 1% of the total study area. The data were removed except 10% points selected randomly to represent the GBR. Data bleaching records intersected with the islands were also removed because the island map from ESRI Digital Chart of the World 1993 is more accurate. The un-known bleaching category records were also removed. And the end, the non human induced bleaching records were also removed to avoid non environmental factors. Finally, there are 202 bleaching point observation used in this research.

2.2.2. Environmental Data

Six environmental data collected in this research area sea surface temperature, chlorophyll-a concentratraion, photosynthetically active radiation, ultraviolet radiation, sea surface currents, and sea surface winds.

2.2.2.1. Sea Surface Temperature

Sea Surface Temperature (SST) data were downloaded from NOAA CoastWatch OPeNDAP server (Appendix A). This Pathfinder version 5.0 sea surface temperature data is the highest quality data available (Kumar*, et al.*, 2000). It is science quality data which do not need for pre-processing (Kilpatrick*, et al.*, 2001). The pixel values already represent the SST values in the field. The SST data were recorded using the Advance Very High Resolution Radiometer (AVHRR) sensor flown in the NOAA-17 and NOAA-18. The spatial resolution is 4 km resolution and SST value accuracy at 0.3° Celsius. The data format is in the NetCDF and can be accessed directly to IDV. 23 years monthly SST data from 1985 to 2007 were subsetted and prepared for further analysis.

2.2.2.2. Chlorophyll-a

Chlorophyll-a monthly data, 1997 to 2007 were downloaded from NOAA CoastWatch OPeNDAP server (Appendix A). The data produced from the Sea-viewing Wide Field-of-View Sensor (SeaWiFS), Orbview-2 satellite. The Chlorophyll-a SeaWiFS data has 0.1 degree spatial resolution. It was processed using OC4v4 algorithm (O'Reilly*, et al.*, 1998). The accuracy of chlorophyll-a measurement of SeaWiFS data is 35% and the values are ranging from 0.001 to 64 mg/m^3. The pixel values of chlorophyll-a is already in a range that do not need pre-processing. The data format is in NetCDF and can be downloaded to IDV and stored in hard disk.

2.2.2.3. Photosynthetically Active Radiation (PAR)

PAR is defined as the quantum energy flux from the Sun in the visible spectral band (Frouin*, et al.*, 2000). The unit expressed the PAR is Einstein/m^2/day. The global coverage of PAR data is available from the SeaWiFS project. NOAA CoastWatch OPeNDAP server does not provide the PAR data. The PAR data were downloaded form Australia's Commonwealth Scientific and Industrial Research Organisation *(*CSIRO*)*. CSIRO provides OPeNDAP access to SeaWiFS PAR global coverage data for 8 day temporal resolution and 9 km spatial resolution. The time frame of the data is form 1997 to 207. The data are available in NetCDF format from the OPeNDAP server (Appendix A). There area thousand temporal layers of PAR data that easily were aggregated using IDV.

2.2.2.4. Ultraviolet Radiation

NASA has been producing global UV radiation maps which are derived from Total Ozone Mapping Spectrometer (TOMS). The maps are estimated using the ozone and aerosol amounts, cloud transmittance, surface reflectivity and the solar UV radiation backscattered. The data are available in the NetCDF format from NASA (Appendix A). Data download can not be done directly from IDV because the database was organised in a different way. Mirador data access user interface was used to locate the data and to make a catalogue file. The file was executed in Linux machine using *wget* command. There area 3,238 NetCDF files of UV variables covering the whole globe from 1997 to 2005. The spatial resolution of UV data is 1.25 degrees. Combining all the file in IDV was done using NetCDF Markup Language (NCML) script. NCML is a simple but useful script to manipulate NetCDF data (Unidata, NA).

2.2.2.5. Surface Currents

Sea surface current data were downloaded from the Pacific Marine Environmental Laboratory (PMEL) NOAA (Appendix A) using IDV. The data are organized in an OPeNDAP server and formatted in

NetCDF. The current are estimated from using satellite derive topography and wind stress using a physically base statistical model (Lagerloef, *et al.*, 1999). TOPEX/Poseidon sensor was the first generation of sea surface current satellite based measurement. Later Jason-1 and Jason-2 continue the TOPEX/Poseidon mission until now. There are two components inside the sea surface data, direction and magnitude of the current. Only the magnitude component was used in this study.

2.2.2.6. Surface Winds

The Special Sensor Microwave Imager (SSMI) provides global surface wind data. The SSMI is a seven channel, four frequency, passive microwave radiometric system which measures atmospheric, ocean and terrain microwave brightness at 19.35, 22.24, 37.00, and 85.80 µm (Austin, *et al.*, 2000). The data were collected in NetCDF format from the World Data Center for Climate (WDCC) (Appendix A).

2.3. Vulnerability Model

The data preparation and processing were done using IDV, a distributed data access client. All the environmental data were in the NetCDF format. Processing was done to locally stored data even though online processing could be done with the effect of reducing processing speed. Before saving the data to local disk, spatial and temporal data sub-setting were done. This function is available in almost all distributed data access clients.

Twelve 12 environmental variables were derived using IDV following the equations in the Table 1. There are some basic processing tools available in the standard package IDV, but in most cases advanced tools need to be developed by using Jython scripting language, Visualization for Algorithm Development (VisAD)-Java data model and Apache.org mathematical functions. Temporal aggregation using statistical function mean, max, standard deviation, and simple linear regression were done using Jython scripts Several mathematical calculations; addition, subtraction, multiplication, and division were also used. (see Appendix C). Some aggregated environmental variables still contain missing pixel values. A simple majority filtering was done to these data to fill the missing pixel values with the nearest neighbour pixel values.

Table 1. The equations used to derive variables

Derived variable	Data	Equation	Definition
Long term mean (*LtM*)	SST, PAR, Chlo, Currents, Wind, UV	$\frac{\sum_{i=1}^{n_m} Xm_i}{n_m}$ Eq. (1)	Xm is the monthly mean and n_m is the number of months
Average of maximum mean monthly ($\overline{Xm_{\max}}$)	SST	$\frac{\sum_{i=1}^{n_y} Xm_{\max}}{n_y}$ Eq. (2)	$Xm_{\max}$ is the maximum of the Xm's for year $i = 1 \ldots n_y$, n_y is the number of years
Coefficient of variation (*CV*)	SST	$\frac{\sigma \times 100}{LtM}$ Eq. (3)	σ is the standard deviation of Xm and LtM is the long term mean
Long term hotspot (*Hlt*)	SST	$\frac{\sum_{i=1}^{n_y} (Xm_{\max} - \overline{Xm_{\max}})}{n_y}$ Eq. (4)	n_y is the number of years where $Xm_{\max} > \overline{Xm_{\max}}$ (positive difference)

Slope of regression line, m	SST	$\frac{n_y \sum(xy) - \sum x \sum y}{n_y \sum(x^2) - (\sum x)^2}$ Eq. (5)	x is the time in years from $i = 0 \ldots n_y$ y is the average of the monthly mean for year $i = 1 \ldots n_y$

Source: Modified from Maina 2007.

2.3.1. Vulnerability Model of the WIO in the Coral Triangle

Two different models developed for the Western Indian Ocean were applied in the Coral Triangle. The first model (VM1) used Spatial Principal Component Analysis (SPCA) and the second model (VM2) used Analytical Hierarchy Process (AHP). Both models used expert knowledge. A detailed explanation can be found in the original paper (Maina, 2007).

The 12 variables derived from the environmental data were used as the input to the models. A fuzzy membership was applied to assign a degree of vulnerability of coral reefs changes as the variables changes. The fuzzy membership function takes values ranging from 0 to 1 to represent low to high vulnerability. There are three fuzzy functions used in this research; right trapezoidal, left trapezoidal and Gaussian. The right trapezoidal defines environmental data values lower than the low breaking point as 0, the values higher than the high breaking point as 1. The data values fall between the breaking points will have values ranging from 0 to 1. The left trapezoidal is the inversion of the right trapezoidal. The Gaussian membership used specifically for the mean max SST. Table 2 was used to assign a degree of vulnerability membership to the environmental variables. The results are 12 maps showing membership degree of environmental variables on coral bleaching stress (Appendix B).

Table 2. Fuzzy membership control points

Variable	Unit	Break point		
		a	b/c	d
Sea Surface Temperature (SST), mean	^oC	26	29	
Mean max SST	^oC	21	27	32
Ultraviolet Radiation (UV)	milli-watts/m^2	230	300	
Wind speed	m/s		5	8
Coefficient of Variation SST (CV)	^oC		4	10
Hotspot SST	^oC	0.4	0.9	
Chlorophyll-a	Mg/m^3		0.05	0.2
Photosythetically Active Radiation (PAR)	E/m^2/day	45	50	
Zonal currents	Cm/s	0.1	0.5	
Meridional currents	Cm/s	0.05	0.2	
Model bleaching		20	60	
Slope SST	^oC	0.01	0.03	

Source: (Maina, 2007)

The next step was calculating VM1 and VM2. VM1 was calculated using the inputs from the fuzzy membership function multiplied with the coefficients (Table 3) from Principal Component Analysis calculation. The first seven SPCA based on the eigen values were calculated for inputs to VM1 calculation. Here the formula of VM1 (Maina, 2007):

$$VM1 = 0.5649*PC1 + 0.1344*PC2 + 0.0837*PC3 + 0.0765*PC4 + 0.0416*PC5 + 0.0366*PC6 + 0.0272*PC7 \quad \text{Eq. (6)}$$

Table 3. PCA matrix coefficients used to calculate SCPA components

	CV	PAR	SST slope	SST mean	Meridional currents	Zonal currents	Wind speed	Max SST	UV	Hotspot	Model	Chlorophyll-a
PC1	-0.414	-0.146	-0.026	-0.407	-0.234	-0.007	-0.298	-0.432	-0.398	0.099	0.364	0.100
PC2	-0.229	0.228	-0.186	-0.317	0.671	0.101	-0.424	-0.024	0.124	-0.045	-0.308	-0.087
PC3	0.153	-0.554	0.155	0.154	0.585	-0.075	0.127	-0.091	-0.303	-0.051	0.082	0.385
PC4	-0.145	0.333	0.463	0.037	0.350	0.112	0.284	-0.038	0.017	0.144	0.535	-0.358
PC5	-0.088	0.140	-0.788	0.016	0.147	0.017	0.427	0.060	-0.064	0.226	0.269	0.123
PC6	-0.110	0.009	-0.121	-0.039	-0.010	0.185	0.247	-0.040	-0.267	-0.863	-0.013	-0.240
PC7	-0.180	0.396	0.142	0.007	-0.013	-0.239	0.027	-0.044	0.315	-0.329	0.171	0.703
PC8	0.310	-0.147	-0.249	0.172	0.067	-0.306	-0.460	-0.029	0.282	-0.238	0.534	-0.243
PC9	0.586	0.313	-0.046	0.004	0.014	0.280	0.014	-0.681	-0.046	0.023	-0.070	0.083
PC10	-0.074	0.107	0.000	0.016	0.061	-0.813	0.225	-0.337	-0.113	0.025	-0.276	-0.255
PC11	0.485	0.161	0.076	-0.653	0.006	-0.211	0.048	0.410	-0.273	-0.026	0.110	0.060
PC12	0.003	0.418	-0.021	0.501	0.006	-0.057	-0.353	0.222	-0.625	0.035	-0.007	0.071

Source: (Maina, 2007)

VM2 was calculated using the same input as VM1 inputs with the different coefficients (Table 4) developed using AHP (Maina, 2007). The formula to calculate VM2 is

$$VM2 = 0.098*P1 + 0.092*P2 + 0.091*P3 + 0.080*P4 + 0.097*P5 + 0.055*P6 + 0.091*P7 + 0.084*P8 + 0.074*P9 + 0.081*P10 + 0.073*P11 + 0.082*P12 \quad \text{Eq.(7)}$$

Table 4. Pair-wise parameter relation strentgth

	Max SST	SST mean	UV	Chlorophyll-a	CV	Bleaching Model	Wind speed	PAR	Zonal currents	Hotspot	Meridional currents	SST slope	Eigen value
P1	1												0.098
P2	0.92	1											0.092
P3	0.89	0.87	1										0.091
P4	0.51	0.38	0.43	1									0.08
P5	0.9	0.92	0.91	0.59	1								0.097
P6	0.21	0.01	0	0.61	0.11	1							0.055
P7	0.87	0.9	0.79	0.43	0.83	0.2	1						0.091
P8	0.79	0.7	0.9	0.37	0.78	0.02	0.66	1					0.084
P9	0.64	0.65	0.69	0.34	0.65	0.03	0.63	0.63	1				0.074
P10	0.48	0.66	0.42	0.66	0.48	0.59	0.39	0.41	0.28	1			0.081
P11	0.49	0.41	0.55	0.35	0.57	0.24	0.5	0.55	0.44	0.37	1		0.073
P12	0.65	0.59	0.54	0.44	0.62	0.44	0.57	0.44	0.38	0.47	0.42	1	0.082

Source: (Maina, 2007)

2.3.2. Maximum Entropy (MaxEnt) modelling

Maximum entropy is species distribution modelling that uses a simple and precise mathematical formulation to predict incomplete information based-on the presence/absence data (Phillips*, et al.*, 2006). A complete species distribution over an area can be generated together with the precision of the prediction. MaxEnt has been widely used in species suitability modelling as a complement of existing GARP model. Here MaxEnt was used to predict coral bleaching response to environmental changes. Observed bleaching data were considered as the species presence/absence data.

The MaxEnt software version 3.2.19 was used in the study. The inputs for MaxEnt are the original environmental variables without interaction to the expert knowledge (Figure 3). There are two categories of inputs required to run the MaxEnt model, a species present/absence data and environmental variables. The environmental variables can be either continuous or categorical spatial data. Full extent environmental variables are required to produce spatial distribution maps as a result of MaxEnt modelling. Point environmental variables can also be used if there are not available for the full extent or study area.

In this study, coral bleaching observations points are considered as species present/absence data input to MaxEnt. Each bleaching category is presented to MaxEnt as a species. All 4 bleaching categories were used. The 12 environmental variables used in the SM models were used as environmental layers in the MaxEnt. The bleaching categories were formatted as CSV format while the environmental variables were exported from IDV to ESRI© Grid using ArcGIS version 9.3 (ESRI, 2008) and later they were converted to ASCII grid. Resample operation was done to ensure all the environmental variables share same pixel size and spatial extent. The parameters were set as follows: random test percentage = 30, regularization multiplier = 1, maximum iteration = 1000, convergence threshold = 0.0001 and maximum number of background pixels = 10,000. The outputs of MaxEnt model are response curves, prediction maps and jackknife of variable importance.

These outputs were used to evaluate the environmental variables for best prediction of bleaching vulnerability. As there were four bleaching inputs categories (species), MaxEnt also produced 4 probability spatial distribution models. These model were combined using a rule proposed by (Phillips*, et al.*, 2006). A new pixel category of combined model was assigned according the input category which has the maximum probability. ERDAS imagine graphical model (GMD) was used to perform this combination.

2.3.3. Model Evaluation

Results of the models were compared each other and to observed bleaching data to evaluate the performance of models. The evaluation was done on both continuous and classified outputs. The continuous outputs were calculated by normalizing the raw VM1 and VM2 calculation results. Their values were stretched from zero (the minimum value) and to one (the maximum value). The classified model outputs, VM1 and VM2, were made by slicing the continuous output according to the natural's break classification (Table 5). VM3 model was classified using the rule explained in previous section (2.3.2). McNemar's statistical test was used to test the hypothesis 1 as it is recommended by (de Leeuw*, et al.*, 2006). McNemar's test is more sensitive to reject null hypothesis than Kappa-z test. The model attributable accuracies were calculated and compared among the models to see the model providing the highest accuracy. Descriptive statistics of the model results were produced to provide additional information. The best model output was used to show the preliminary map of environmentally vulnerable area to coral bleaching in the Coral Triangle.

Table 5. Classes in the VM1 and VM2

Class	VM1	VM2	Description
1	0 - 0.35	0 - 0.24	Not vulnerable
2	0.36 - 0.56	0.25 - 0.36	Low vulnerability
3	0.57 - 0.65	0.37 - 0.52	Medium vulnerability
4	0.66 - 1	0.53 - 1	High vulnerability

Source: (Maina, 2007)

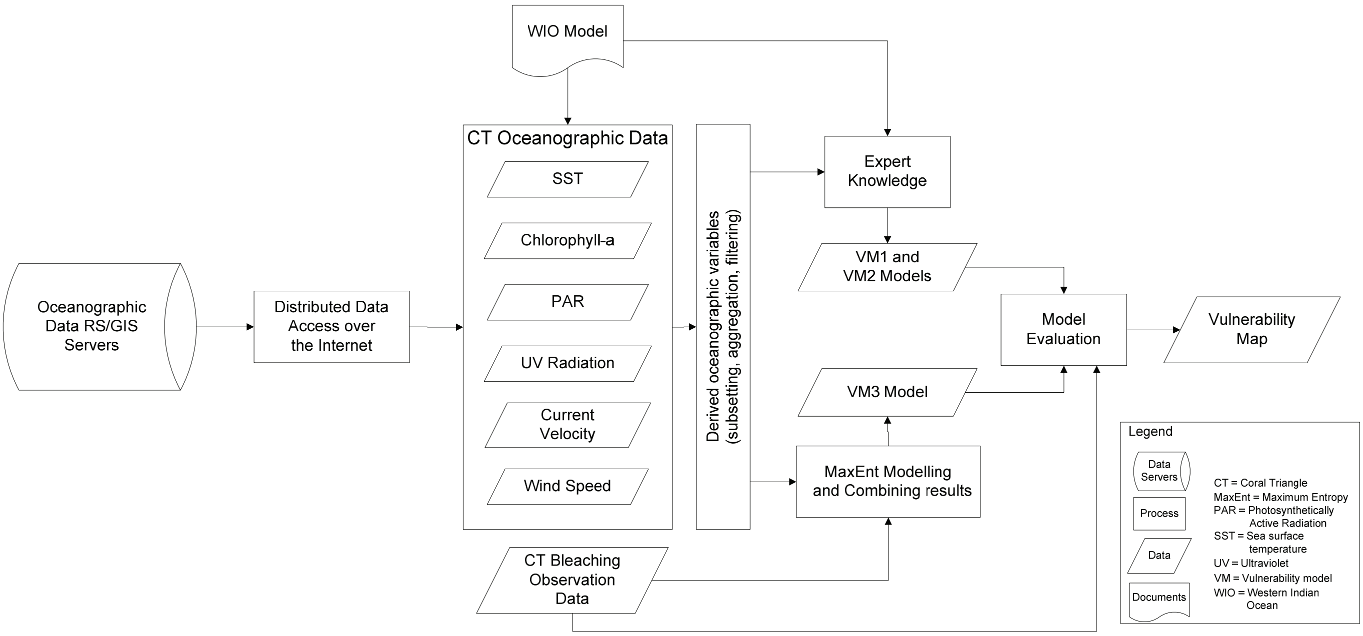


Figure 3. Flowchart of the research process

3. Results

3.1. Coral Bleaching Observation

Five hundred data points of observed bleaching were overlaid on top of the study area map (Figure 2). There were half of the data located in the Great Barrier Reef (GBR), Australia. From all the other areas, the GBR is the most intensive researched region covering only 1% of the total study area. 270 data points are located. In contrast to this, there are very limited data points in the eastern part of Indonesia. Including all of those data in the analysis does not improve the results because of misrepresentation. Representation of the data sample to the variation of the whole area is more important. Many data points found in the GBR were then dropped. Ten percent of the data were randomly selected and used for further data analysis. Two hundreds two data points were prepared for further analysis.

The 202 observed bleaching data points were plotted against the environmental variables in Figure 4. Their relationship can be seen by inspecting the trend of the data points. SST mean, Max SST, UV, and zonal currents are increasing as the bleaching severity increases (Figure 4: (a), (b), (c) and (g) while CV, PAR, wind, and chlorophyll-a are decreasing as the bleaching severity increases (Figure 4: (d), (e), (f) and (h). Among these relationships, PAR shows different pattern to its pattern in the WIO (Maina, 2007). In the WIO PAR is increasing as bleaching severity increases. In the study area PAR is decreasing as the bleaching severity increases. This is probably an artifact caused by the low quality of the Reefbase data.

Most the graphics show high standard deviations. These reflect to the fact that the observed bleaching data were collected by Reefbase from many difference organizations. They could use difference field survey methods or different quality of the surveyors. Assignment of the bleaching severity to a point observation is qualitative and subjective. 4 categories ob bleaching severity were used; high, medium, low and no bleaching. It is much accurate in differentiating high bleaching severity from no bleaching severity, but it will become more difficult and more subjective in differentiating high bleaching severity to medium or medium to low bleaching severity.

The 202 coral reefs observation point data were overlaid over the classified model output maps to extract corresponding values of the point locations. Their distribution frequencies were shown in Figure 5. The distribution frequencies between observed bleaching categories are comparable. These do not occur in the bleaching vulnerability categories of the models. The model VM1 is dominated by the high vulnerable class followed by the low vulnerable and not vulnerable classes respectively. The model VM2 is dominated by medium vulnerable class followed by high and low/not vulnerable classes. The model VM3 contains equally distributed classes. The Figure 5 also shows that VM3 is more similar to the observed bleaching categories distribution.

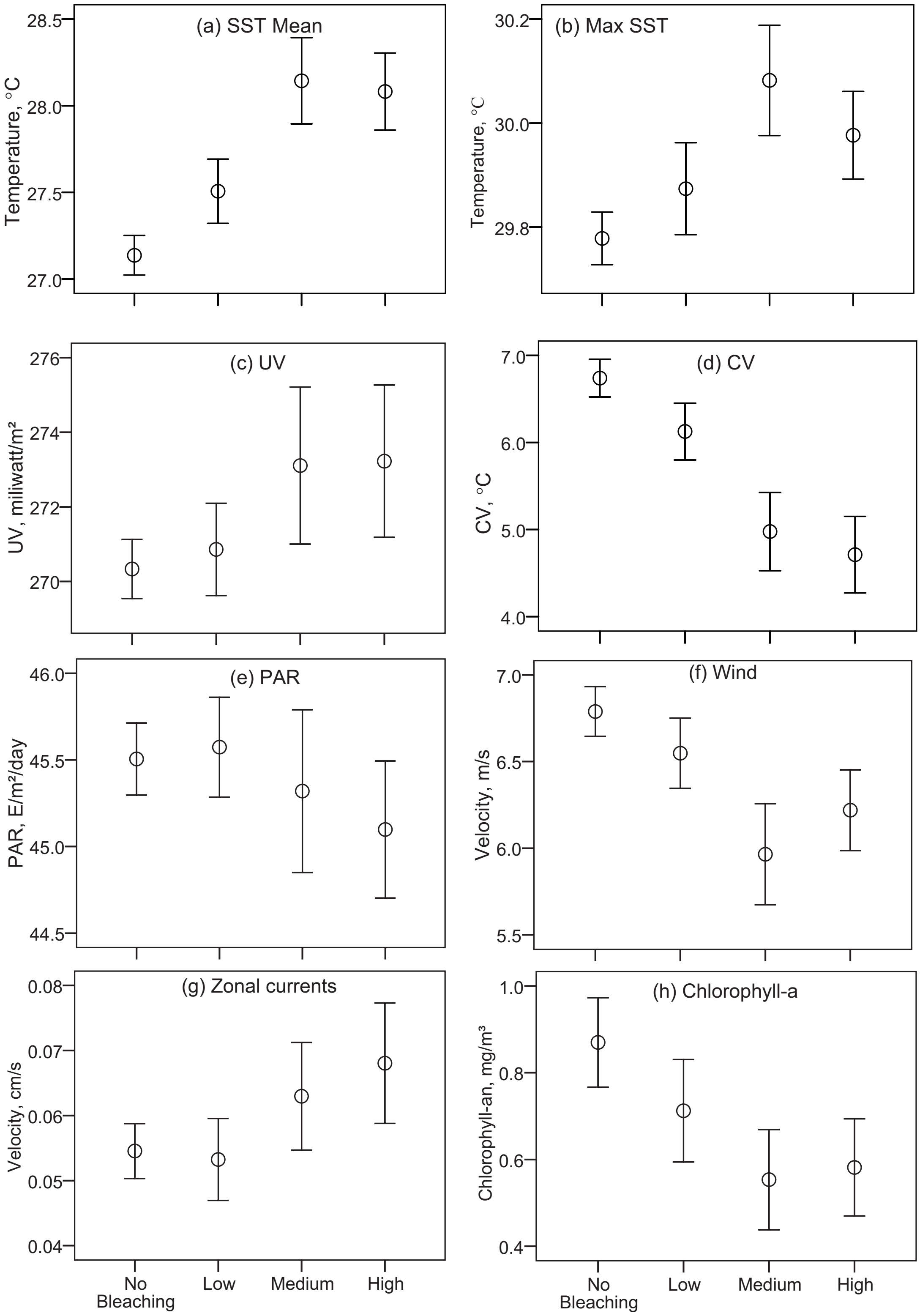


Figure 4. Bleaching intensity against the environmental variables (standard error of mean)

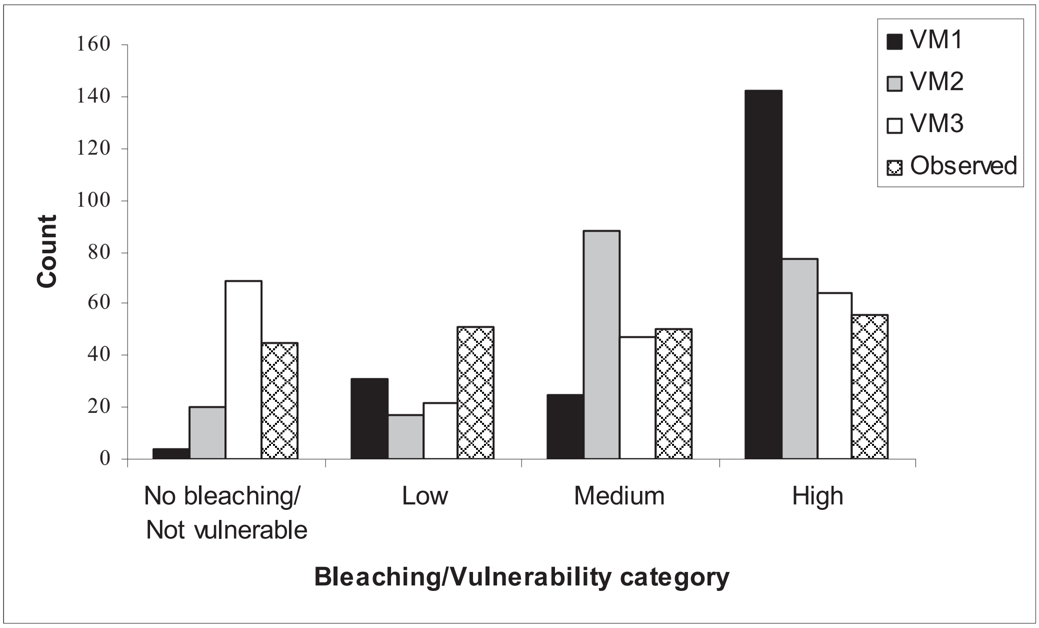


Figure 5. Comparison of observed bleaching to vulnerability categories of the models

3.2. Vulnerability Model in the Coral Triangle

Two models (VM1 and VM2) were produced based on the previous model developed for the WIO. Another model (VM3) was produced using Maximum Entropy modelling software.

3.2.1. Vulnerability models based on the WIO models

The first model output maps (VM1 and VM2) are displayed in Figure 6: (a) and (c). Both models have similar pattern. High vulnerability values are located in the horizontal centre along the equator line while low vulnerability values are located in the both side, top and down. Generally, the pattern varies according to the latitude line. Some differences can be detected in the upper left map next to the northern part of the Philippines. Different classes of vulnerable areas in the VM1 model are more fragmented than in the VM2 model. Both maps shared same locations for not vulnerable areas to coral bleaching located between Papua islands and Australia. In the maps, the areas were displayed as light brown. The area in the centre of the Coral Triangle shows the high vulnerability values.

Different vulnerability patterns are shown in the categorical vulnerability maps (Figure 6: (b), (d), and (e). The VM1 map shows larger, more compact vulnerability classes than the VM2 map. The high vulnerability class in the VM1 dominates the map coverage, more than 50 percent. The other vulnerability classes are equally distributed. The not vulnerable class is only found in the northern part of the Philippines. The map shows vulnerability categories where their boundaries reflect the latitude line. The VM2 map shows vulnerability classes more distributed than VM1 map. Boundary between the vulnerability classes is not sharp as the VM1 map. The not vulnerable class is found in the northern part of the Philippines and the area between Papua New Guinea and Australia.

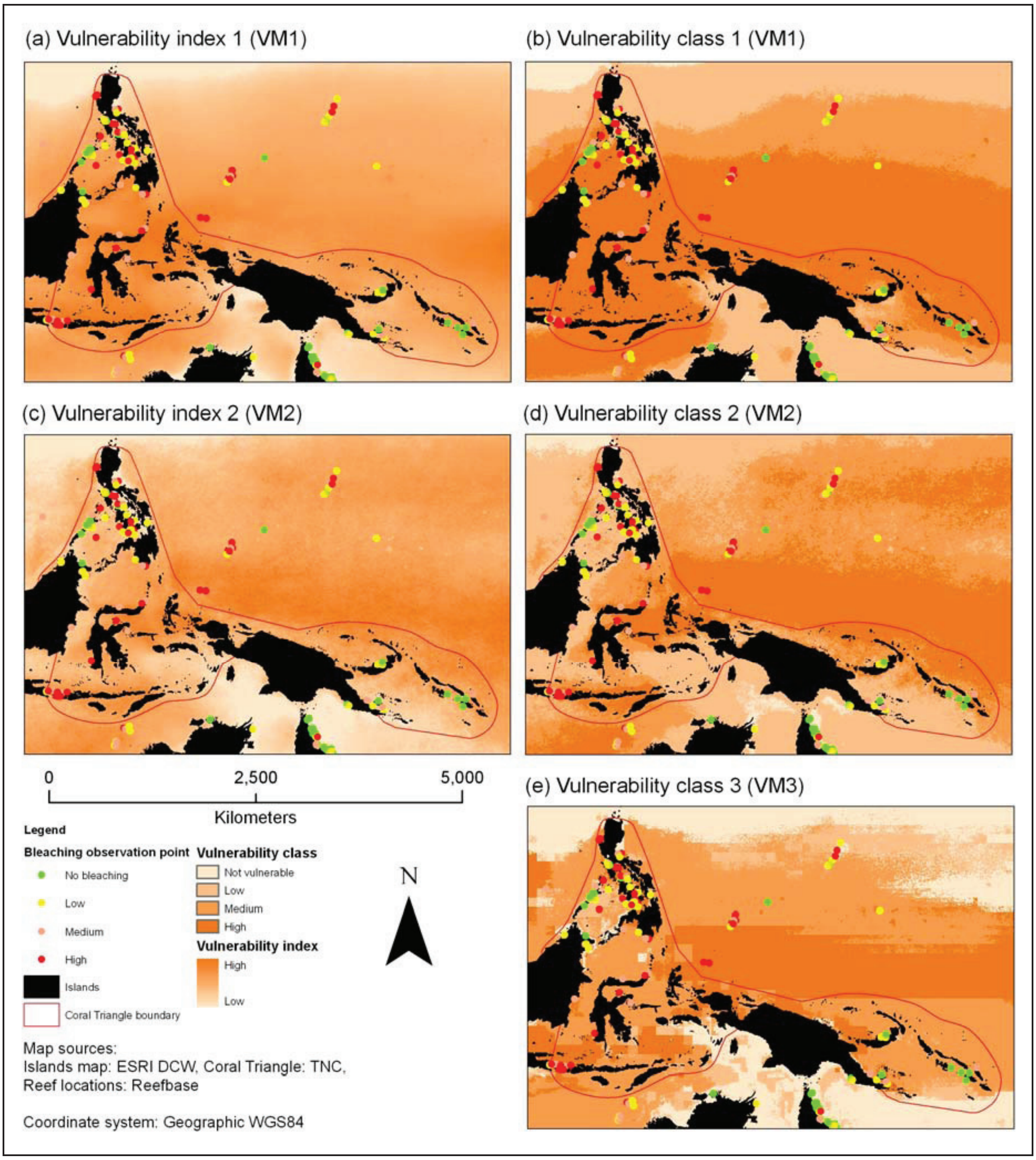


Figure 6. Vulnerability maps according to different models, continuous (a) VM1 (b) VM2 and categorical (c) VM1 (d) VM2 (e) VM3

3.2.2. Vulnerability Model based on MaxEnt

Contributions of each environmental variable to the MaxEnt model were shown in Figure 7. Chlorophyll-a is the highest contributor (>43%) to the MaxEnt model for all models. CV, PAR, currents, hotspot and wind have almost the same contribution around 10%. The other variables have very low contribution to the models. Figure 8 shows the gains of each environmental variable with only the variable or without the variable. The Jackknife graphs show that the chlorophyll-a is the most important variables to the model. The seven variables showing high contribution to the model were chosen to produce bleaching vulnerability model using MaxEnt.

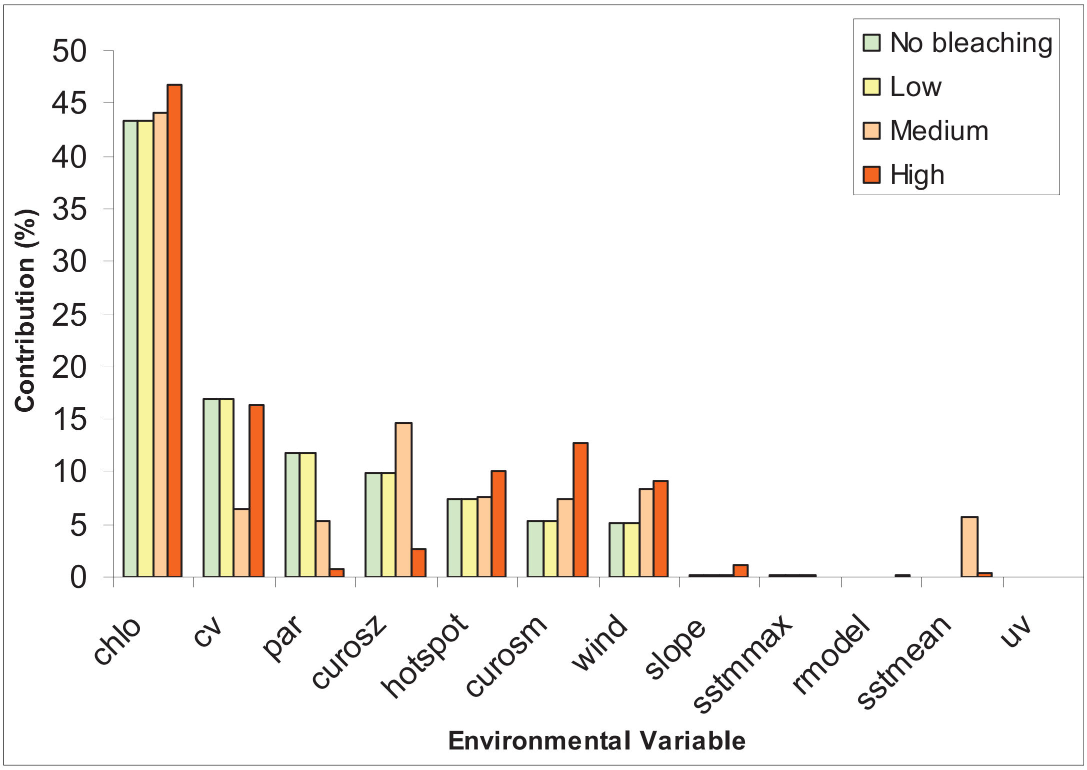


Figure 7. Percent contribution of the environmental variables to MaxEnt model

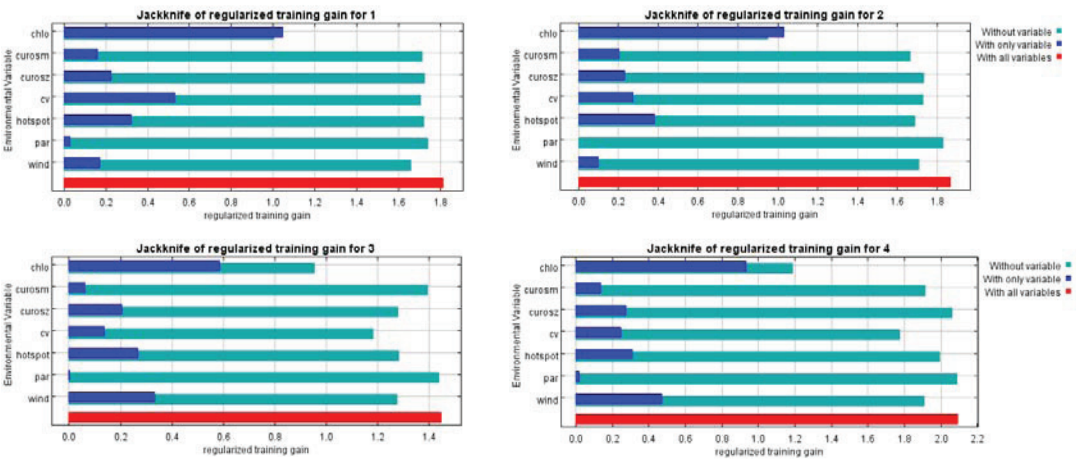


Figure 8. Jackknife graphs of the four models in MaexEnt

Four different maps resulted from the MaxEnt runs were combined using simple formulation base on Phillips, *et al.* (2006). The final model was calculated by assigning a new pixel according to the maximum probability value of the four model outputs. An illustration can be described here. A pixel location corresponding to superimposed 4 models contains the highest probability value belongs to a model using medium bleaching category presence/absence input. The pixel location will be assigned as medium vulnerable class in the final MaxEnt output. Figure 6 (e) shows the final output of MaxEnt model. Compared to the other model, the MaxEnt model shows relatively distributed vulnerable categories over the study area. Figure 9 shows enlarge model output with 2 zoom-in areas (A and B) for more detail comparison betwenn the model output and the observed bleaching data.

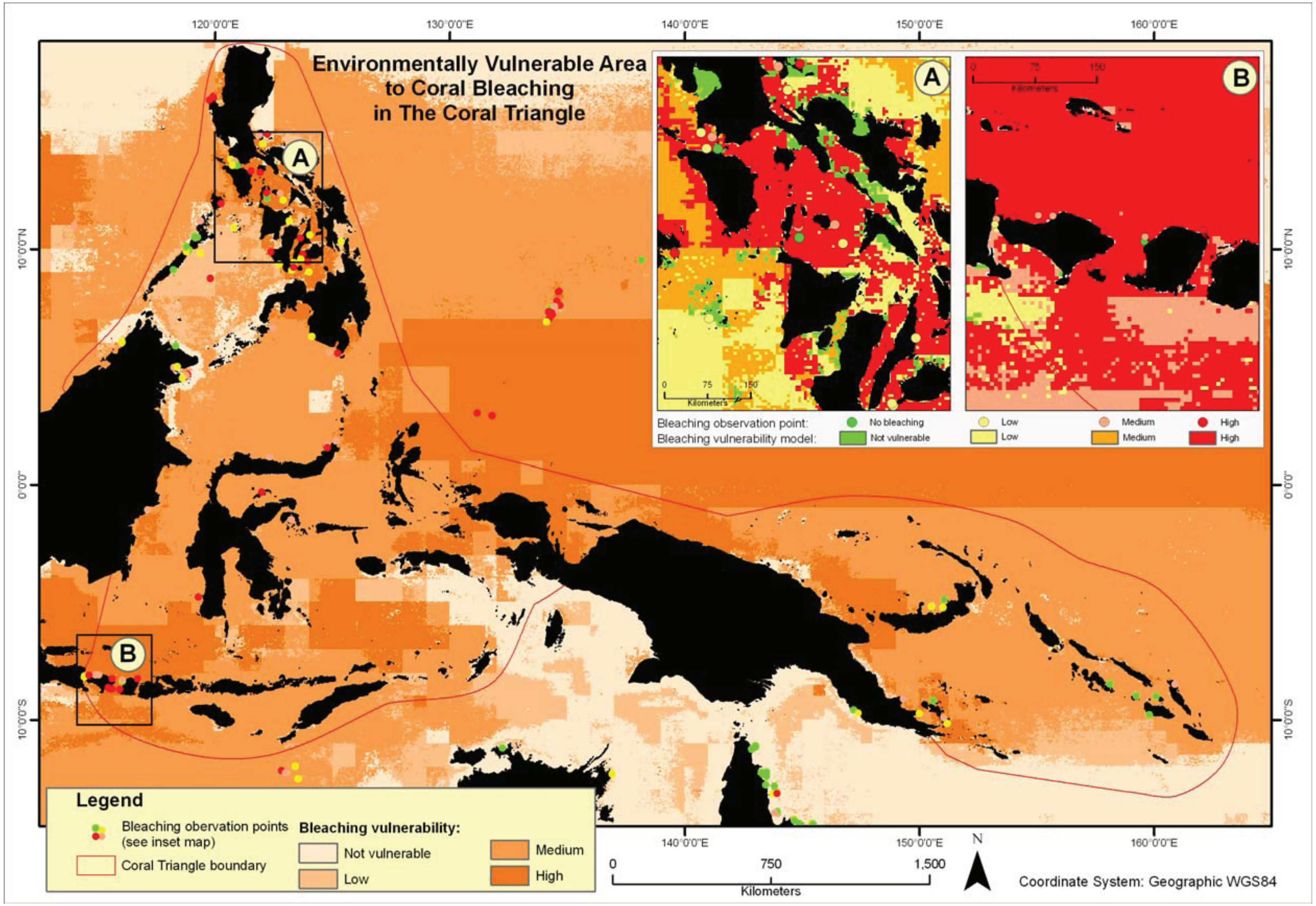


Figure 9. Environmentally Vulnerable Area in the Coral Triangle

3.3. Model Comparison

The number of pixels of each model for the whole study area is shown in Figure 10. The model VM1 has the biggest number of pixels belong to the high vulnerability class as well as the model VM2. The model VM3 shows dominant class for the medium vulnerability category. Model VM1 and VM2 show small number of pixels for not vulnerable category. Visually comparison of the model output maps, model VM2 appears more representative with field condition. Vulnerability classes in the model VM1 and VM3 are bulky. Each class is separated with very clear boundary. The model VM3 appear more realistic with boundaries between its vulnerability classes is rather gradual than sharp.

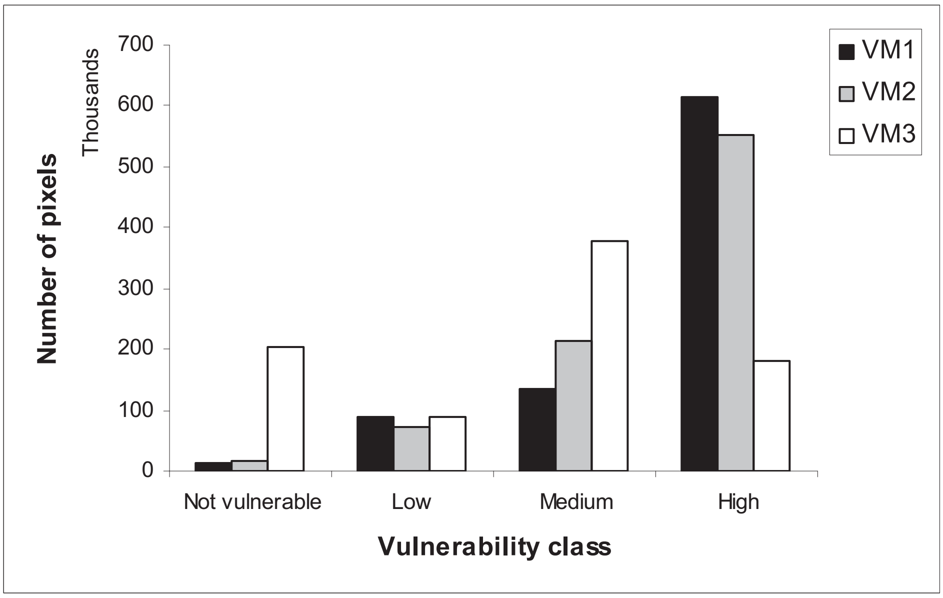


Figure 10. Distribution frequency of vulnerability values between the models

McNemar's test was conducted to the pairs of correct and wrong classified pixels of the models. McNemar's test for VM1 & VM2 shows the cqi-square value of 0.77 which is lower than 3.841 at the significant level of 0.05 (Table 6**Error! Reference source not found.**). Therefore we do not reject the null hypothesis 1 for VM1 and VM1. In another word, the two models VM1 and VM2 performed with no difference. Both McNemar's values for VM1/VM3 and VM2/VM3 are 7.44 and 12.10 respectively which are higher than 3.841. These mean we reject the null hypothesis 1 for VM1/VM3 and VM2/VM3. The model VM3 is significantly different with both VM1 and VM2. Table 7 shows the summary of correctly and wrongly classified pixels by the model VM1, VM2 and VM3 compared to the bleaching observation data.

Table 6. McNemar's test result

	VM1and VM2	VM1 and VM3	VM2 and VM3
N	202	202	202
Chi-Square	0.77	7.44	12.10
Asymp. Sig.	0.3816	0.0064	0.0005

Table 7. Summary of correctly and wrongly classified bleaching categories by VM1, VM2 and VM3

a

	VM2		
VM1	Wrong	Correct	Total
Wrong	111	28	139
Correct	36	27	63
Total	147	55	202

b

	VM3		
VM1	Wrong	Correct	Total
Wrong	84	55	139
Correct	29	34	63
Total	113	89	202

c

	VM3		
VM2	Wrong	Correct	Total
Wrong	85	62	147
Correct	28	27	55
Total	113	89	202

The outputs of the models (VM1, VM2 and VM3) were compared by utilizing the bleaching observation data (Figure 5). The description was given in the previous section. Table 8, Table 9, and Table 10 show the confusion matrices of the model outputs against bleaching observation data. Their overall accuracies are 0.31, 0.27, and 0.44 respectively whereas their Kappa values are VM1, VM2, and VM3 are 0.06, 0.02, and, 0.26. Comparison of "correct by methods", "correct by chance" and wrong classified between the models were shown in Figure 11. The "correct by chance" values of the models are same but differ for the "correct by method". The MaxEnt model provided the highest "correct by method" values followed by VM1 and VM2. From this statistic values, MaxEnt model is showing the best performance.

Table 8. Confusion matrix of VM1

VM1	Observed bleaching				Total
	No bleaching	Low	Medium	High	
Not vulnerable	3	1	0	0	4
Low	16	8	4	3	31
Medium	4	8	6	7	25
High	22	34	40	46	142
Total	45	51	50	56	202

Table 9. Confusion matrix of VM2

VM2	Observed bleaching				Total
	No bleaching	Low	Medium	High	
Not vulnerable	13	3	2	2	20
Low	5	5	4	3	17
Medium	15	23	18	32	88
High	12	20	26	19	77
Total	45	51	50	56	202

Table 10. Confusion matrix of VM3

VM3	Observed bleaching				Total
	No bleaching	Low	Medium	High	
Not vulnerable	27	17	13	12	47
Low	3	12	5	2	37
Medium	8	11	18	10	54
High	7	11	14	32	64
Total	45	51	50	56	202

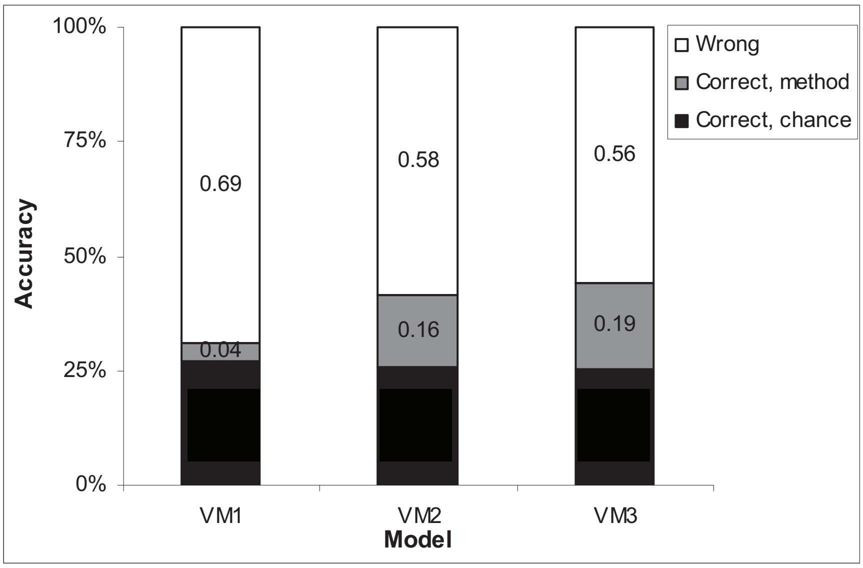


Figure 11. Stacked bar graph showing proportion of overall, method attributable and chance attributable accuracies of the models.

4. Discussion

4.1. Coral Bleaching Observation Data

The coral bleaching observation data used in this study were collected from the Reefbase database which is freely available and covers the whole globe. The result showed that the data has high variation. This can be understood by the fact that the data were collected from many difference organizations or individual scientist who might have difference survey methods. Some records contain no data and some areas do not have point observations. The other finding is that the point observation data are not equally distributed over the whole study area. We found more than 50% observation data were located in the Great Barrier Reef which cover less than 1% of total of study area. The error data will lead to bias conclusion. It is not necessary to use all data available. Good data quality is pre-requisite for good results. Data filtering to remove these “error” is important.

Data filtering was done in this research by removing the error data and selecting few points from the over sampled area in the GBR. The low data quality caused by unrepresented areas of bleaching observation point remains unsolved. No global bleaching observation data sources other than Reefbase found in this research. Collection from local research organizations requires more time to cover the whole study area. Better bleaching point observation data absolutely will improve the model outputs.

Another interesting finding in this research is the negative relationship of chlorophyll-a and PAR to the bleaching category. From biological perspective, coral bleaching experts generally explain that chlorophyll-a concentration and PAR have positive relationship with the coral bleaching severity. Higher chlorophyll-a concentration and PAR increase the bleaching severity. In this research, the bleaching severity was found increase as the chlorophyll-a and PAR increase. Why this happens to this study area?

The study area, the Coral Triangle is characterised by semi open sea area. Many islands from big to too small (sea mount) spread from the Philippines through Indonesia to Papua New Guinea and Solomon. The longest coastline with continental shelf is located in the area. This configuration influences its oceanographic conditions, such as up-welling, long shore currents, sedimentation from terrestrial discharges, and water chemical compounds. Furthermore, its geographical location along the Equator line with very high precipitation could also influence the coral resilience to bleaching. Moreover, the Coral Triangle is the area in the world having the highest marine biodiversity but least scientific researches done compare to other area such as Caribbean and GBR.

4.2. Vulnerability Models

Three bleaching vulnerability models were made in this study. The vulnerability model 3 (VM3) developed using the MaxEnt modelling software is the best model even though it is not the best representing the true field condition because of its very low accuracy, less than 50%. At least there are two reasons why the model VM3 provides the best performance. Firstly, the model VM3 was trained using completely the study area data while the other two models were trained using other than study area data. Secondly, the model VM3 does not use the expert system which in many cases very subjective to the researchers. MaxEnt is intensively used in biological modelling and it provides higher performance than the other modelling algorithm, GARP (Kayijamahe, 2008). A different result might be produced when a calibration to the model VM1 and VM2 with the study area data could be done.

Even though all models shared pixel same sizes and geographic extent, visual comparison between the model outputs reveal that the model VM1 and VM3 are bulky, shows sharp boundary between the vulnerability classes whereas the model VM2 shows its class boundary more natural. Furthermore, the models VM3 show some isolated blocks, a different vulnerability class in the middle of other classes. These blocky features also appeared on the boundary separating each vulnerability classes. This feature possibly caused by the difference software application. The first two models (VM1 and VM2) utilized IDV which provide better display capability than MaxEnt used for developing the model VM3.

Modelling bleaching vulnerability using MaxEnt provides a new interesting research area. The model showed that chlorophyll-a contributes the highest to the model output. This finding is similar to the finding when relating the bleaching observation data to the chlorophyll-a but not to the PAR. PAR is shown here contributing lower than chlorophyll-a. Chlorophyll-a acts an important rule in the Coral Triangle. The oceanographic factors described in section 4.2 influence this relationship.

4.3. Vulnerability Model and Marine Protected Area

The Coral Triangle has been attracted many organization during the last decade. It is recently found that the area has the highest coral biodiversity in the world, but an extensive research study has not done. In the same times, the marine resources in the area are threatened by human and environmental stressor such as climatic changes. A map showing the environmental vulnerable areas to coral bleaching is helpful for the MPA practitioners. In this research, the models could not provide a final map showing environmentally vulnerable area to coral bleaching with a sufficient accuracy. A preliminary map has been created to illustrate a desired model output.

The preliminary map shows the Coral Triangle area is dominated by medium and high vulnerability categories. Some small areas with not vulnerable category can be found between Papua and Australia, and in the coastal area along the north-eastern part of Kalimantan and the islands between Sulu and Mindanao seas. Low vulnerable category can also be found in the south-eastern part of Sulawesi islands, covering the Wakatobi National Park. These areas which are not vulnerable to bleaching stress caused by the environmental factors are the best candidate for Marine Protected Area (MPA) sites. New MPA could be established or existing MPA could be extended in the area characterized by not vulnerable or low vulnerability categories.

5. Conclusions and recommendations

5.1. Conclusions

1. Distributed data access technology is very useful technique to collect time series data that updated every time. IDV acts as one of the distributed data access clients with some basic processing is available. More sophisticated processing is possible by using Jython scripts.

2. Vulnerability models that originally developed in the Western Indian Ocean showed very low accuracy in the Coral Triangle areas compared to vulnerability model developed using MaxEnt. All the models, however, show very low accuracy that could be attributed to the low quality of the reference data which collected from many different organizations.

3. Both models, VM1 an dVM2 have similar performance in the Coral Triangle. Their differences were not significance. Both models VM1 and VM2 have different performance compared to the MaxEnt model.

4. Chlorophyll-a and ses surface temperature variables were found to behave differently against common coral bleaching research. This research showed the chlorophyll-a provid the highest contribution to the coral bleaching model far higher than the sea surface temperature which commonly considered as the most influence factor.

5. An uncalibrated model developed from a different area could not perform well in the study area.

5.2. Recommendations

1. The environmental variables, chlorophyll-a and PAR used to model bleaching vulnerability in the Coral Triangle area behaves differently compare to the Western Indian Ocean. Characterization of chlorophyll-a and PAR in the Coral Triangle in relation to coral bleaching is required for better understanding of their rule.

2. Some data quality issues were found in the Reefbase dataset. Many areas are still unrepresented by the database whereas some areas have much higher data sample density. Another data collection to fill the unrepresented area could improve the model quality.

3. Model calibration using area of interest data for a model developed from another area could improve the model performance.

References

Austin, J., D. Armstrong and M. Dahm. 2000. *Special sensor microwave/imager (ssm/i) user's interpretation guide*. Retrieved 5 December 2008, from http://www.ncdc.noaa.gov/oa/rsad/ssmi/fnoc-ssmi-manual.pdf.

Baker, A. C., P. W. Glynn and B. Riegl. 2008. Climate change and coral reef bleaching: An ecological assessment of long-term impacts, recovery trends and future outlook. *Estuarine Coastal and Shelf Science* 80(4): 435-471.

Berkelmans, R., G. De'ath, S. Kininmonth and W. J. Skirving. 2004. A comparison of the 1998 and 2002 coral bleaching events on the great barrier reef: Spatial correlation, patterns, and predictions. *Coral Reefs* 23(1): 74-83.

Buddemeier, R. W., P. L. Jokiel, K. M. Zimmerman, D. R. Lane, J. M. Carey, G. C. Bohling and J. A. Martinich. 2008. A modeling tool to evaluate regional coral reef responses to changes in climate and ocean chemistry. *Limnology and Oceanography-Methods* 6: 395-411.

Carriquiry, J. D., A. L. Cupul-Magana, F. Rodriguez-Zaragoza and P. Medina-Rosas. 2001. Coral bleaching and mortality in the mexican pacific during the 1997-98 el nino and prediction from a remote sensing approach. *Bulletin of Marine Science* 69(1): 237-249.

Cesar, H., L. Burke and L. Pet-Soede 2003. *The economics of worldwide coral reef degradation*. Arnhem (The Netherlands), Cesar Environmental Economics Counsulting**:** 23.

de Leeuw, J., H. Jia, X. Liu, K. Schmidt and A. K. Skidmore. 2006. Comparing accuracy assessments to infer superiority of image classification methods. *International Journal of Remote Sensing* 27(1): 223-232.

ESRI. 2008. *Arcgis desktop help 9.3*. Retrieved 14 January 2009, from http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=welcome.

Franklin, D. J., C. M. M. Cedres and O. Hoegh-Guldberg. 2006. Increased mortality and photoinhibition in the symbiotic dinoflagellates of the indo-pacific coral stylophora pistillata (esper) after summer bleaching. *Marine Biology* 149(3): 633-642.

Frouin, R., B. Franz and M. Wang 2000. *Algorithm to estimate par from seawifs data, version 1.2*. NASA.

Goldberg, J. and C. Wilkinson 2004. *Global threats to coral reefs: Coral bleaching, global climate change, disease, predator plagues, and invasive species*. Status of coral reefs of the world: 2004. C. WILKINSON. Townsville, Queensland, Australia, Australian Institute of Marine Science**:** 67-92.

Green, A. and P. Mous 2004. *Delineating the coral triangle, its ecoregions and functional seascapes*. The Nature Conservancy, Southeast Asia Center for Marine Protected Areas**:** 26.

Grimsditch, G. D. and R. V. Salm 2006. *Coral reef resilience and resistance to bleaching*. Gland Switzerland, IUCN**:** 52.

International Marine Sanctuaries. *International experts to meet in florida keys to discuss climate change effects on coral reefs*. Retrieved 20 August 2008, from http://sanctuaries.noaa.gov/news/press/2008/pr041408.html.

Kayijamahe, E. 2008. *Spatial modelling of mountain gorilla (gorilla beringei beringei) habitat suitability and human impact.* MSc, ITC. http://www.itc.nl/library/papers_2008/msc/nrm/kayijamahe.pdf

Kilpatrick, K. A., G. P. Podesta and R. Evans. 2001. Overview of the noaa/nasa advanced very high resolution radiometer pathfinder algorithm for sea surface temperature and associated matchup database. *Journal of Geophysical Research-Oceans* 106(C5): 9179-9197.

Kumar, A., P. Minnett, G. Podesta, R. Evans and K. Kilpatrick.2000. *Analysis of pathfinder sst algorithm for global and regional conditions*, Indian Academy Sciences.

Lagerloef, G. S. E., G. T. Mitchum, R. B. Lukas and P. P. Niiler. 1999. Tropical pacific near-surface currents estimated from altimeter, wind, and drifter data. *Journal of Geophysical Research-Oceans* 104(C10): 23313-23326.

Maina, J. 2007. *Modelling ecological susceptibility of coral reefs to environmental stress using remote sensing, gis and in situ observations: A case study in the western indian ocean.* MSc, ITC. http://www.itc.nl/library/papers_2007/msc/gem/mbui.pdf

Maina, J., V. Venus, T. R. McClanahan and M. Ateweberhan. 2008. Modelling susceptibility of coral reefs to environmental stress using remote sensing data and gis models. *Ecological Modelling* 212(3-4): 180-199.

McClanahan, T. R. 2004. The relationship between bleaching and mortality of common corals. *Marine Biology* 144(6): 1239-1245.

McClanahan, T. R., J. Maina, R. Moothien-Pillay and A. C. Baker. 2005. Effects of geography, taxa, water flow, and temperature variation on coral bleaching intensity in mauritius. *Marine Ecology-Progress Series* 298: 131-142.

O'Reilly, J. E., S. Maritorena, B. G. Mitchell, D. A. Siegel, K. L. Carder, S. A. Garver, M. Kahru and C. McClain. 1998. Ocean color chlorophyll algorithms for seawifs. *Journal of Geophysical Research-Oceans* 103(C11): 24937-24953.

Oxenford, H. A., R. Roach, A. Brathwaite, L. Nurse, R. Goodridge, F. Hinds, K. Baldwin and C. Finney. 2008. Quantitative observations of a major coral bleaching event in barbados, southeastern caribbean. *Climatic Change* 87(3-4): 435-449.

Pet-Soede, L. and S. Owen 2007. *The coral triangle at the center of marine biodiversity*. World Wildlife Fund.

Phillips, S. J., R. P. Anderson and R. E. Schapire. 2006. Maximum entropy modeling of species geographic distributions. *Ecological Modelling* 190(3-4): 231-259.

Roberts, C. M., S. Andelman, G. Branch, R. H. Bustamante, J. C. Castilla, J. Dugan, B. S. Halpern, K. D. Lafferty, H. Leslie, J. Lubchenco, D. McArdle, H. P. Possingham, M. Ruckelshaus and R. R. Warner. 2003. Ecological criteria for evaluating candidate sites for marine reserves. *Ecological Applications* 13(1): S199-S214.

Sale, P. F. 2002. *Coral reef fishes: Dynamics and diversity in a complex ecosystem*, Elsevier Escience.

Schuttenberg, H. Z. 2001. *Coral bleaching: Causes, consequences and response*. 9th International Coral Reef Symposium on Coral Bleaching: Assessing and Linking Ecological and Socioeconomic Impatcs, Future Trends and Mitigation Planning, Coastal Resources Center, University of Rhode Island**:** 108.

Ulstrup, K. E., R. Berkelmans, P. J. Ralph and M. J. H. van Oppen. 2006. Variation in bleaching sensitivity of two coral species across a latitudinal gradient on the great barrier reef: The role of zooxanthellae. *Marine Ecology-Progress Series* 314: 135-148.

Unidata. NA. *The netcdf markup language (ncml)*. Retrieved 5 December 2008, from http://www.unidata.ucar.edu/software/netcdf/ncml/.

Wooldridge, S. and T. Done. 2004. Learning to predict large-scale coral bleaching from past events: A bayesian approach using remotely sensed data, in-situ data, and environmental proxies. *Coral Reefs* 23(1): 96-108.

WWF. 2007. *Major steps taken to protect the coral triangle*. Retrieved 20 August 2008, from http://www.panda.org/news_facts/successes/index.cfm?uNewsID=121380.

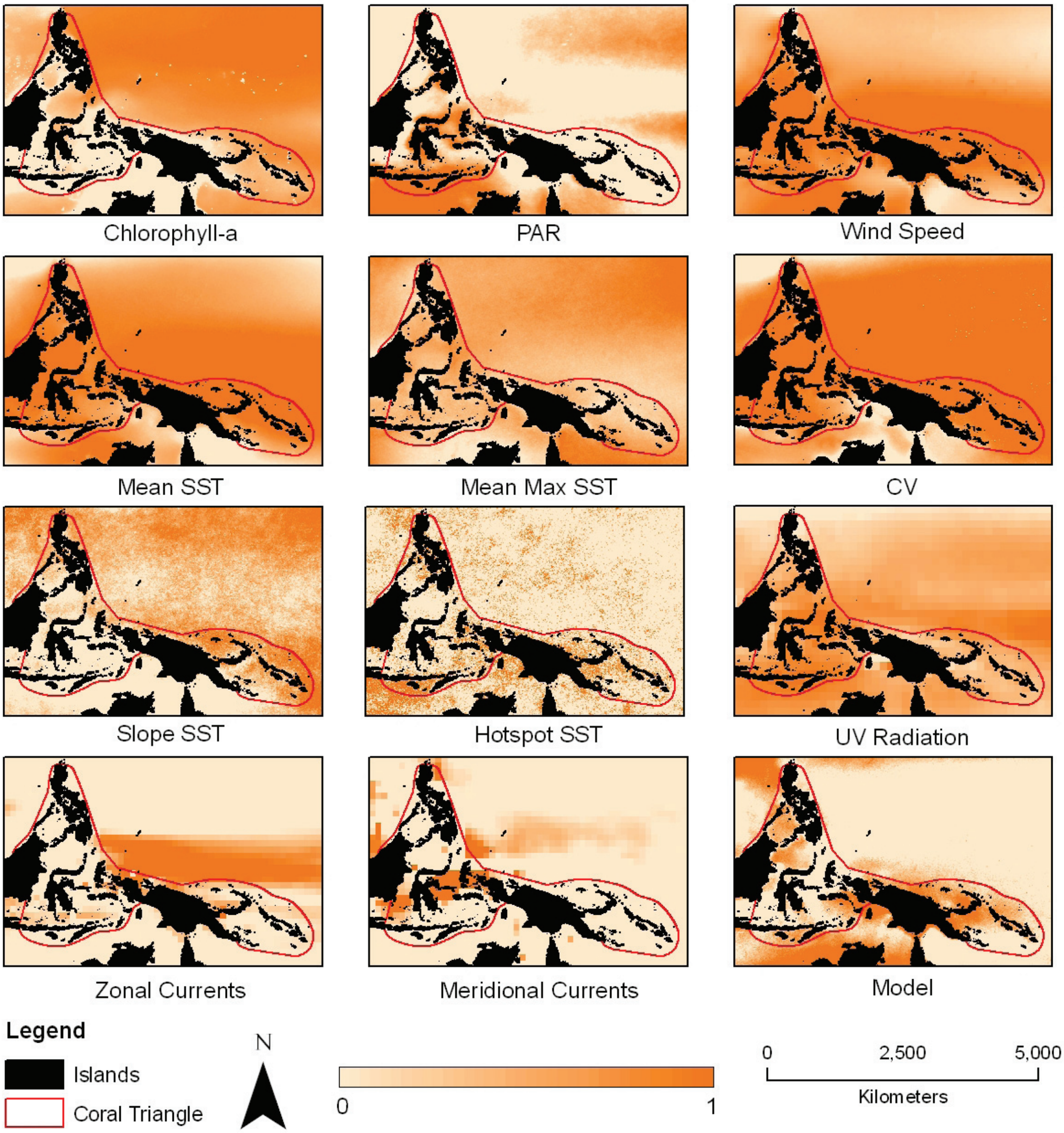
WWF. 2008. *Coral triangle: Home to the world's most abundant variety of corals and sea life* Retrieved Access date: 30 July 2008, from URL: http://www.worldwildlife.org/what/wherewework/coraltriangle/index.html.

Appendices

Appendix A – Data sources URL/OPeNDAP

Data	Satellite	URL/OpenDAP
SST	NOAA AVHRR	http://oceanwatch.pfeg.noaa.gov:8081/thredds/catalog.xml
		dods://thredds1.pfeg.noaa.gov:-1/thredds/dodsC/satellite/PH/ssta/mday
Chlorophyll-a	SeaWiFS	http://oceanwatch.pfeg.noaa.gov:8081/thredds/catalog.xml
		dods://thredds1.pfeg.noaa.gov:-1/thredds/dodsC/satellite/SW/chla/mday
Photosynthetically Active Radiation	SeaWiFS	dods://www.marine.csiro.au/dods/nph-dods/dods-data/ocean_colour/seawifs/global/8_day/netcdf/f_L3m_8D_PAR_9_yyyy.nc
Ultraviolet Radiation	TOMS	http://mirador.gsfc.nasa.gov/index.shtml
		wget http://acdisc.sci.gsfc.nasa.gov/daac-bin/OTF/HTTP_services.cgi?FILENAME=%2Fdata%2Fs4pa%2F%2FEarth_Probe_TOMS_Level3%2FTOMSEPL3%2F%2F2005%2FTOMS-EP_L3-TOMSEPL3_2005m1231_v8.HDF&LABEL=TOMS-EP_L3-TOMSEPL3_2005m1231_v8.nc&SHORTNAME=TOMSEPL3&SERVICE=NetCDF&VERSION=1.02" -O TOMS-EP_L3-TOMSEPL3_yyyymmmdd_v8.nc
Wind Speed	SSMI	http://cera-www.dkrz.de/WDCC/ui/Index.jsp
Surface Current	OSCAR	http://dapper.pmel.noaa.gov/dapper/oscar/world-filter-monthly.nc

Appendix B – Partial vulnerability maps



Appendix C – Jython scripts

```
#####################################################################################
# Purpose: To aggregate time series single grid/image
# Description: This script will aggregate time series grids using
# statistical function Mean, Max, Min, StandardDeviation.
# Operation:
# 1. Change the name after ‘def’ according to a function (in this case: Mean)
# 2. Uncomment the function to be run.
# 3. Change the ‘alg’ to the function.
#####################################################################################

def mbAggregateMeanSeq(timeSequence,singleBand):
  import sys;
```

```
  sys.add_package('visad.meteorology');
  sys.add_package('GridUtil');
  # here we may choose various aggregator offered by the apache.org "math" package
  sys.add_package('org.apache.commons.math.stat.descriptive.moment');
  from org.apache.commons.math.stat.descriptive.moment import Mean
  #sys.add_package('org.apache.commons.math.stat.descriptive.rank');
  #from org.apache.commons.math.stat.descriptive.rank import Max
  #sys.add_package('org.apache.commons.math.stat.descriptive.rank');
  #from org.apache.commons.math.stat.descriptive.rank import Min
  #sys.add_package('org.apache.commons.math.stat.descriptive.moment');
  #from org.apache.commons.math.stat.descriptive.moment import StandardDeviation
  from jarray import zeros
  from java.lang import Float
  from java.lang import Double
  #singleBand = timeSequence.getSample(0)
  domain = singleBand.getSample(0).getDomainSet()
  len =  domain.getLength()
  # Pass in 0 as false so we don't copy the values
  samples = domain.getSamples(0)
  # Clone the incoming objects
  news = singleBand.clone()
  lineValuesA = singleBand.getSample(0).getFloats()
  alg = Mean()
  #print "Starting for loop ... number of iterations = ", len
  # calculate aggregates for all coordinates (crd) for all times (time)
  for crd in xrange(len):
      #print "Pixel nr. ", crd + 1, " of ", len
      # get length of imageSequence
      seq = timeSequence.getDomainSet().getLength()
      # create array to store values
      arr = zeros(seq, "d")
      # get all values for one coordinate for all times time and put them into arr
      #print "Entering nested forloop 1"
      for time in xrange(seq):
          # get the values
          # Pass in 0 as false so we don't copy the values
          #print "    Retrieving pixel info for image nr. ", time + 1, " of ", seq
          values = timeSequence.getSample(time).getFloats(0)
          #print "    Value for current pixel (#", crd + 1, "): ",values[0][crd]
          arr[time] = values[0][crd]
      cnt = 0
      for i in xrange(seq):
        if (arr[i] > 0) and (Double(arr[i]).isNaN() == 0):
          cnt = cnt + 1
      validArray = zeros(cnt, "d")
      index = 0
      for i in xrange(seq):
        if (arr[i] > 0) and (Double(arr[i]).isNaN() == 0):
            validArray[index] = arr[i]
            index = index + 1
      aggregate = alg.evaluate(validArray)
      lineValuesA[0][crd] = aggregate
      #print "array from timeSequence: ",arr
      #print "array after filtering: ",validArray
      #print "aggegated array: ",aggregate
      alg.clear()
  news.getSample(0).setSamples(lineValuesA)
  return news


####################################################################################
# Purpose: To make a time series sequence grid
# Description: This script will combines many single time series grids into a new
# single time series grid
####################################################################################

def mbMakeTimeSequence(g):
  """ Merge a set of single time grids/images into a time sequence """
  from visad import FunctionType, FieldImpl, Gridded1DDoubleSet, QuickSort
  from jarray import array
  domain = getDomainSet(g[0])
  dt = getDomainType(g[0])
  v=[getDomain(g[i]).indexToDouble([0,])[0][0] for i in range(len(g))]
```

```
  va = array(v,'d')
  index = QuickSort.sort(va)
  ft=FunctionType(dt, getRangeType(g[0]))
  fld=FieldImpl(ft,Gridded1DDoubleSet.create(dt,va,None,domain.getSetUnits()[0],None))
  for i in range(len(g)):
     fld.setSample(i,g[index[i]].getSample(0),0)
  return fld

####################################################################################
# Purpose: To apply the right trapezoidal fuzzy membership function
# Description: This script will assign new values based on the control points
# entered. a value will be assigned to 0 while b value to 1
####################################################################################

from java.lang import Double
def mbNormalizedVariableR(image,user_a,user_b):
  singleBand = image.getSample(0)
  newd = singleBand.clone()
  lineValuesA = singleBand.getFloats()
  a = float(user_a)
  b = float(user_b)
  for i in xrange(singleBand.getDomainSet().getLength()):
      pixel = lineValuesA[0][i]
      if (pixel < a):
         fuzzy = 0
         #print fuzzy
      elif (pixel > b):
         fuzzy = 1
         #print fuzzy
      else:
         fuzzy = (pixel-a)/(b-a)
         #print fuzzy
      lineValuesA[0][i] = fuzzy
  newd.setSamples(lineValuesA)
  return newd

####################################################################################
# Purpose: To apply the left trapezoidal fuzzy membership function
# Description: This script will assign new values based on the control points
# entered. c value will be assigned to 1 while b value to 0
####################################################################################

from java.lang import Double
def mbNormalizedVariableL(image,user_c,user_d):
  singleBand = image.getSample(0)
  newd = singleBand.clone()
  lineValuesA = singleBand.getFloats()
  c = float(user_c)
  d = float(user_d)
  for i in xrange(singleBand.getDomainSet().getLength()):
      pixel = lineValuesA[0][i]
      if (pixel <= c):
         fuzzy = 1
         #print fuzzy
      elif (pixel >= d):
         fuzzy = 0
         #print fuzzy
      else:
         fuzzy = (d - pixel)/(d-c)
         #print fuzzy
      lineValuesA[0][i] = fuzzy
  newd.setSamples(lineValuesA)
  return newd

####################################################################################
# Purpose: To apply the Gaussian (S) trapezoidal fuzzy membership function
# Description: This script will assign new values based on the control points
# entered. a and d are values will be assigned as 0 while c value will be assigned
# to 1.
####################################################################################
```

```
import math
from java.lang import Double
def mbNormalizedVariableH(image,user_a,user_b,user_d):
  singleBand = image.getSample(0)
  newd = singleBand.clone()
  lineValuesA = singleBand.getFloats()
  a = float(user_a)
  b = float(user_b)
  d = float(user_d)
  for i in xrange(singleBand.getDomainSet().getLength()):
      pixel = lineValuesA[0][i]
      if (pixel <= a):
         fuzzy = 0
         #print "a:", fuzzy
      elif (pixel >= d):
         fuzzy = 0
         #print "d:", fuzzy
      elif (pixel <= b):
         value = 1-((pixel-a)/(b-a))*math.pi/2
         fuzzy = math.cos(value)*math.cos(value)
         #print "b:", fuzzy
      else:
        value = ((pixel-b)/(d-b))*math.pi/2
         fuzzy = math.cos(value)*math.cos(value)
         #print "b1:", fuzzy
      lineValuesA[0][i] = fuzzy
  newd.setSamples(lineValuesA)
  return newd


####################################################################################
# Purpose: To invert values in a grid
# Description: This script will assign the minimum value to 1, the maximum value
# to 0, and vary between them. a is the minimum value while b is the maximum value
# Operation:
####################################################################################

from java.lang import Double
def mbInvertingSM1(image,user_a,user_b):
  singleBand = image.getSample(0)
  newd = singleBand.clone()
  lineValuesA = singleBand.getFloats()
  a = float(user_a)
  b = float(user_b)
  for i in xrange(singleBand.getDomainSet().getLength()):
      pixel = lineValuesA[0][i]
      if (pixel < a):
         fuzzy = 1
         #print fuzzy
      elif (pixel > b):
         fuzzy = 0
         #print fuzzy
      else:
         fuzzy = (pixel-b)/(a-b)
         #print fuzzy
      lineValuesA[0][i] = fuzzy
  newd.setSamples(lineValuesA)
  return newd


####################################################################################
# Purpose: To classify a continuous grid
# Description: This script will classify a grid according to assign entered boundary
# classes. Here a grid will be classified into 5 classes.
####################################################################################

from java.lang import Double
def mbclusteringSM(image,user_a,user_b,user_c,user_d):
  singleBand = image.getSample(0)
  newd = singleBand.clone()
  lineValuesA = singleBand.getFloats()
  a = float(user_a)
```

```
  b = float(user_b)
  c = float(user_c)
  d = float(user_d)
  for i in xrange(singleBand.getDomainSet().getLength()):
      pixel = lineValuesA[0][i]
      if (pixel <= a):
         value = 1
      elif (pixel <= b):
         value = 2
      elif (pixel <= c):
         value = 3
      elif (pixel <= d):
         value = 4
      else:
         value = 5
      lineValuesA[0][i] = value
  newd.setSamples(lineValuesA)
  return newd


####################################################################################
# Purpose: To sample grid values at point locations
# Description: This script will extract grid values at random locations
####################################################################################

sys.add_package('org.python.core');
from org.python.core import PyReflectedFunction
import java.lang.StringBuffer as SB
# this is a workaround for the append() StringBuffer error
# which seems related to the problem that package private classes with public
# methods confuse Jython 2.1.
for n,f in java.lang.AbstractStringBuilder.__dict__.items():
    x = PyReflectedFunction(n)
    for a in f.argslist:
      if a is None: continue
      m = SB.getMethod(n,a.args)
      x.addMethod(m)
    SB.__dict__[n] = x
#This routine just calls the IDV function "exportCsvAllTimes" that performs the I/O
for file writing
def exportCsvAllTimes (buffer):
 exportCsvAllTimes
 sys.add_package('ucar.unidata.util')
 from ucar.unidata.util import FileManager
 from ucar.unidata.util import IOUtil
 try:
   filename = FileManager.getWriteFile(FileManager.FILTER_CSV,FileManager.SUFFIX_CSV);
   IOUtil.writeFile(filename, buffer.toString());
 except:
   return;

def mbGetValueAtLocation(g,a,s) :
  """Extract a value from a 2D grid at "Location."
     In Maxent this is a so-called 'samples with data',
     or in short, a SWD-file.

     Requires a text-file with data in these dimension:
(index -> ((Latitude, Longitude, Altitude), Time, range values)))
     Range values are rounded to two (2) decimals.
                                                 """
  from jarray import array
  from ucar.unidata.idv.control.DisplayControlBase import makeEarthLocation
  from visad.python.JPythonMethods import *

  domain = domainType(a)
  range = rangeType(a)
  dom = a.getDomainSet()
  pointData = a[0][2]
  types = pointData.getType()
  typesLength = len(types)
  domLength =  dom.getLength()
  buff = SB();
  sampleLength = len(g)
```

```
  buff.append("Species");
  buff.append(",");
  buff.append("Longitude");
  buff.append(",");
  buff.append("Latitude");
  for j in xrange(sampleLength):
    buff.append(",");
    buff.append(GridUtil.getParamType(g[j]))
  for q in xrange(typesLength):
    buff.append(",");
    buff.append(types[q])
  buff.append("\n");
  for i in xrange(domLength):
    pointData = a[i][2]
    latitude  = a[i][0].getComponent(0).getValue()
    longitude = a[i][0].getComponent(1).getValue()
    earthLocation = makeEarthLocation(latitude,longitude,1.0)
    location = earthLocation.getLatLonPoint()
    #print location
    buff.append(s);
    buff.append(",");
    buff.append(str(longitude));
    buff.append(",");
    buff.append(str(latitude));
    for j in xrange(sampleLength):
     samples = GridUtil.sample(g[j], location, 1)
     buff.append(",");
     try:
       buff.append(round(samples[0].getValue(),2));
     except:
       buff.append(round(samples[0][0].getValue(),2));  #   for   ADDE   images   use:
samples[0][0].getValue()
    for q in xrange(typesLength):
      buff.append(",");
      buff.append(pointData[q])
    buff.append("\n");
  exportCsvAllTimes (buff)


####################################################################################
# Purpose: To sample grid values at random point locations
# Description: This script will extract grid values at random locations
####################################################################################

def mbGetValueAtRandomLocation(g,ulLat,ulLon,lrLat,lrLon,n) :
  """Extract a value from a 2D grid at random "Locations"
     In Maxent this is a so-called 'samples with data',
     or short SWD-file

     Requires a text-file with data in these dimension:
(index -> ((Latitude, Longitude, Altitude), Time, range values)))
     Range values are rounded to two (2) decimals.
                                          """

  from jarray import array
  from ucar.unidata.idv.control.DisplayControlBase import makeEarthLocation
  from visad.python.JPythonMethods import *
  import random

  ulLat=float(ulLat)
  ulLon=float(ulLon)
  lrLat=float(lrLat)
  lrLon=float(lrLon)
  domLength = int(n)
  buff = SB();
  sampleLength = len(g)
  buff.append("background");
  buff.append(",");
  buff.append("Longitude");
  buff.append(",");
  buff.append("Latitude");
  for j in xrange(sampleLength):
    buff.append(",");
```

```
    buff.append(GridUtil.getParamType(g[j]))
  buff.append("\n");
  for i in xrange(domLength):
    latitude  = random.uniform(lrLat, ulLat)
    longitude = random.uniform(lrLon, ulLon)
    location = makeEarthLocation(latitude,longitude,1.0).getLatLonPoint()
    buff.append("background");
    buff.append(",");
    buff.append(str(longitude));
    buff.append(",");
    buff.append(str(latitude));
    for j in xrange(sampleLength):
     samples = GridUtil.sample(g[j], location, 1)
     buff.append(",");
     try:
       buff.append(round(samples[0].getValue(),2));
     except:
       buff.append(round(samples[0][0].getValue(),2));
    buff.append("\n");
  exportCsvAllTimes (buff)

####################################################################################
# Purpose: To export a grid to Microsoft Excel file format
# Description: This script is useful if you want to export the grid to ArcGIS
####################################################################################

def mbWriteGridToXls(grid,filename='grid.xls'):
    """Write out the grid data to an excel spreadsheet"""
    GridUtil.writeGridToXls(grid,filename)
    return grid


####################################################################################
# Purpose: To set a new unit for a grid
# Description: This script is useful when you need to make a index grid
####################################################################################

def mbNewUnit(field, varname, unitname):
  from ucar.visad import Util
  from ucar.unidata.data.grid import GridUtil
  newunit = Util.parseUnit(unitname)
  newType = Util.makeRealType(varname, newunit)
  return GridUtil.setParamType(field, newType)


####################################################################################
# Purpose: To calculate the Coeffcient of Variation (CV)
# Description: Coefficient of Variation is calculated by dividing Standard Deviation
# to mean.
# Operation: Call the first def
####################################################################################

#First
def mbCoefficientOfVariationSeq(a,b):
  import ucar.unidata.data.grid.GridUtil as gu
  if (gu.isTimeSequence(a)):
     newA = a.clone()
     for t in range(a.getDomainSet().getLength()):
         seqA = a.getSample(t)
         seqB = b.getSample(t)
         newA.setSample(t, mbCoefficientOfVariation(seqA,seqB))
     return newA
  else:
     return mbCoefficientOfVariation (a,b)

#Second
def mbCoefficientOfVariation(a,b):
  from jarray import zeros
  from java.lang import Float

  lineValuesA = a.getFloats()
  lineValuesB = b.getFloats()
  for i in xrange(a.getDomainSet().getLength()):
```

```
      pixelA = lineValuesA[0][i]
      pixelB = lineValuesB[0][i]
      value = (pixelA*100)/pixelB
      #print "A", pixelA
      #print "B", pixelB
      #print value
      lineValuesA[0][i] = value
  newd = a.clone()
  newd.setSamples(lineValuesA)
  return newd


#####################################################################################
# Purpose: To calculate the slope of regression
# Description: This script will calculate the coefficient of regression of time
# series grid by projecting grid value versus time
# Formula: mbSimpleRegressionSeq(timeSequence,singleBand)
#####################################################################################

def mbSimpleRegressionSeq(timeSequence,singleBand):
  import sys;
  sys.add_package('visad.meteorology');
  sys.add_package('GridUtil');
  import ucar.unidata.data.grid.GridUtil as gu
  # here we may choose various aggregator offered by the apache.org "math" package
  sys.add_package('org.apache.commons.math.stat.regression');
  from org.apache.commons.math.stat.regression import SimpleRegression
  #sys.add_package('org.apache.commons.math.stat.descriptive.rank');
  #from org.apache.commons.math.stat.descriptive.rank import Percentile

  from java.util import TimeZone
  from jarray import zeros
  from java.lang import Float, Double, Object

  #singleBand = timeSequence.getSample(0)
  domain = singleBand.getSample(0).getDomainSet()
  len =  domain.getLength()
  # Pass in 0 as false so we don't copy the values
  samples = domain.getSamples(0)
  # Clone the incoming objects
  news = singleBand.clone()
  lineValuesA = singleBand.getSample(0).getFloats()

  alg = SimpleRegression()

  # get length of imageSequence
  seq = timeSequence.getDomainSet().getLength()
  # create array to store values
  #data = zeros(2, "d")
  data = zeros(2, Object)
  # create array to store values
  for i in xrange(2):
      data[i] = zeros(seq, "d")
  #print data
  #y = zeros(seq, "d") #array of observation
  #x = zeros(seq, "d") #array of times
  tz = TimeZone.getTimeZone("GMT");
  for time in xrange(seq):
          dateTime = gu.getDateTimeList(timeSequence).get(time)
          year = Double(dateTime.formattedString("yyyy", tz))
          data[0][time] = year
  #print "Starting for loop ... number of iterations = ", len
  # calculate aggregates for all coordinates (crd) for all times (time)
  for crd in xrange(len):
      print "Pixel nr. ", crd + 1, " of ", len
      # get all values for one coordinate for all times time and put them into arr
      #print "Entering nested forloop 1"
      for time in xrange(seq):
          # get the values
          # Pass in 0 as false so we don't copy the values
          #print "    Retrieving pixel info for image nr. ", time + 1, " of ", seq
          values = timeSequence.getSample(time).getFloats()
          #print "    Value for current pixel (#", crd + 1, "): ",values[0][crd]
```

```
          data[1][time] = Double(values[0][crd])
          #print data[0][time],data[1][time]
          alg.addData(data[0][time],data[1][time])
      #cnt = 0
      #for i in xrange(seq):
      #  if (arr[i] > 0) and (Double(arr[i]).isNaN() == 0):
      #    cnt = cnt + 1
      #validArray = zeros(cnt, "d")
      #index = 0
      #for i in xrange(seq):
      #  if (arr[i] > 0) and (Double(arr[i]).isNaN() == 0):
      #      validArray[index] = arr[i]
      #      index = index + 1
      aggregate = alg.getSlope()
      lineValuesA[0][crd] = aggregate
      #print "array from timeSequence: ",arr
      #print "array after filtering: ",validArray
      print "aggegated array: ",aggregate
      alg.clear()
  news.getSample(0).setSamples(lineValuesA)
  return news


#####################################################################################
# Purpose: To append a grid dimension from another grid
# Description: Some grids have non compatible dimension that prevent it from further
# processing. This script will make the grid compatible.
# Operation: Run this script, save the result in cache, and further process the
# cached data. It is also possible to serialize the cache data for permanent
# storage.
# Formula: dummy(S1,S2)
#####################################################################################

#S1 = a grid to be processed further
#S2 = reference grid

def dummy(S1,S2):
  return quo(mul(S1,S2),S2)


#####################################################################################
# Purpose: To fill missing pixels by using 3 x 3 filter
# Description: Frequently, a grid has missing pixel values. This script will take
# the neighbour missing value and assign a new value according to selected function,
# Max, Mean, Median
# Formula: doMovingFilterImageSeqSampledN52(timeSequence)
# Author: Willem Neuwenhuis
# Note: ~ symbol to keep a next line with previous
#####################################################################################

from jarray import zeros
from java.lang import Double
sys.add_package('org.apache.commons.math.stat.descriptive.rank');
from org.apache.commons.math.stat.descriptive.rank import Max

# Get the indices of the pixels surrounding the current one at 1-D index: index
# parameters:
#   x_size: the number of pixels on a single line
#   y_size: the number of lines
#   index: the 1-D index of the pixel under investigation
# Return value
#   a 3x3 matrix with the 1-D indeces of the surrounding pixels
def getNeighborIndices2D(x_size, y_size, index):
  x = index % x_size
  y = index / x_size
  nb = [[0,0,0],[0,0,0],[0,0,0]]
  if x == 0:
      nb[0][0] = -1
  else:
    nb[0][0] = index - y_size - 1
  nb[0][1] = index - y_size
  if x == (x_size - 1):
    nb[0][2] = -1
```

```
  else:
    nb[0][2] = index - y_size + 1
  if x == 0:
      nb[1][0] = -1
  else:
    nb[1][0] = index - 1
  nb[1][1] = index
  if x == (x_size - 1):
    nb[1][2] = -1
  else:
    nb[1][2] = index + 1
  if x == 0:
      nb[2][0] = -1
  else:
    nb[2][0] = index + y_size - 1
  nb[2][1] = index + y_size
  if x == (x_size - 1):
    nb[2][2] = -1
  else:
    nb[2][2] = index + y_size + 1
  tot_size = x_size * y_size
  for x in xrange(3):
    for y in xrange(3):
      if nb[x][y] < 0: nb[x][y] = -1
      if nb[x][y] >= tot_size: nb[x][y] = -1
  return nb

def getNeighborSamples2D(source, x_size, y_size, index, edgeIsNan):
  nb = getNeighborIndices2D(x_size, y_size, index)
  vals = [[0.0,0.0,0.0],[0.0,0.0,0.0],[0.0,0.0,0.0]]
  for x in range(3):
    for y in range(3):
      if nb[x][y] >= 0:
        vals[x][y] = source[nb[x][y]]
      else:
        if edgeIsNan:
          vals[x][y] = Double.NaN
  return vals

def doMovingFilterImageSampledN52(singleBand):

  from jarray import zeros
  from java.lang import Double
  sys.add_package('org.apache.commons.math.stat.descriptive.rank');
  from org.apache.commons.math.stat.descriptive.rank import Median

  aggregator = Median()
  domain = singleBand.getDomainSet()
  dims = domain.getLengths()
  x_size = dims[1]
  y_size = dims[0]

  #Clone the incoming data object
  newd = singleBand.clone()

  # do the guts
  source = singleBand.getFloats()
  length =  domain.getLength()
  for i in xrange(length):
    if Double(source[0][i]).isNaN():
      # get all values in the 3x3 matrix around current pixel (including current)
      nbv = getNeighborSamples2D(source[0], x_size, y_size, i, 1)
      validArray = []
      for x in range(3):
        for y in range(3):
          if (Double(nbv[x][y]).isNaN() == 0):
            validArray.append(nbv[x][y])
      aggregate = aggregator.evaluate(validArray)
      # set the new value
      source[0][i] = aggregate
      #aggregator.clear()
  # end of for loop
  newd.setSamples(source)
```

```
  return newd

# Time series functions (call this jython function for image sequences)

def doMovingFilterImageSeqSampledN52(timeSequence):
  import ucar.unidata.data.grid.GridUtil as gu
  if (gu.isTimeSequence(timeSequence)):
     newA = timeSequence.clone()
     seq = timeSequence.getDomainSet().getLength()
     for t in range(seq):
         count = t+1
         print "    Computing image nr. ", count, " of ", seq, " (",~
float(count/seq)*100 ,"%)"
         singleB = timeSequence.getSample(t)
         newA.setSample(t,doMovingFilterImageSampledN52(singleB))
     return newA
  else:
     return doMovingFilterImageSampledN52(singleB)

def doMovingFilterImageN52(singleBand):

  domain = singleBand.getDomainSet()
  dims = domain.getLengths()
  x_size = dims[1]
  y_size = dims[0]

  #Clone the incoming data object
  newd = singleBand.clone()

  # do the guts
  source = singleBand.getFloats()

  length =  domain.getLength()
  for i in xrange(length):
    if Double(source[0][i]).isNaN():
      # get all indices of the 3x3 matrix around current pixel
      nbi = getNeighborIndices2D(x_size, y_size, i)

      # Check if we are at the edge of the grid
      v1 = nbi[1][0]   # left from current pixel
      v2 = nbi[1][2]   # right from current pixel
      if v1 < 0:
        if v2 < 0:
          avg = Double.NaN   # left and right are outside image (very unlikely!)
        else:
          avg = source[0][v2]   # left is outside image
      else:
        if v2 < 0:
          avg = source[0][v1]   # right is outside image
        else:
          avg = (source[0][v1] + source[0][v2]) / 2
# both are inside, calc the average
# check if the result is a NaN; this means that either one or both of the left/right
# pixels are NaN
# replace with the one that is not NaN
      if Double(avg).isNaN():
        if Double(source[0][v1]).isNaN():
          if Double(source[0][v2]).isNaN():
            avg = avg   # both are NaN, no change
          else:
            avg = Double(source[0][v1])   # right is NaN
        else:
          avg = Double(source[0][v2])   # left is NaN
    else:
      avg = source[0][i]

    # set the new value
    source[0][i] = avg
    # end of for loop

  newd.setSamples(source)
  return newd
```

```
# Time series functions (call this jython function for image sequences)

def doMovingFilterImageSeqN52(timeSequence):
  import ucar.unidata.data.grid.GridUtil as gu
  if (gu.isTimeSequence(timeSequence)):
     newA = timeSequence.clone()
     seq = timeSequence.getDomainSet().getLength()
     for t in range(seq):
         count = t+1
         print "    Computing image nr. ", count, " of ", seq, " (",~
float(count/seq)*100 ,"%)"
         singleB = timeSequence.getSample(t)
         newA.setSample(t,doMovingFilterImageN52(singleB))
     return newA
  else:
     return doMovingFilterImageN52(singleB)

##################################################################################
# Purpose: To calculate SPCA 1 to 7
# Description: This script will calculate SPCA 1 to 7 based on the constants
# developed by Maina (2007)
# Operation: Call this mbSPCA1(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12) one by one
# from the Define a grid diagnostic script
# Note: ~ symbol to keep a next line with previous
##################################################################################

#s = fuzzy layers
#s1 = slope of regression sst
#s2 = photosynthetically active radiation
#s3 = coefficient of variation sst
#s4 = mean sst
#s5 = zonal currents
#s6 = meridional currents
#s7 = wind speed
#s8 = mean max sst
#s9 = ultraviolet radiation
#s10 = hotspot sst
#s11 = chlorophyll-a
#s12 = regression model

#
def mbSPCA1(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12):
   return GridMath.add(add(add(add(add(add(add(add(add(add(add(s1*(-0.026),s2*(-~
0.146)),s3*(-0.414)),s4*-0.407),s5*(-0.234)),s6*(-0.007)),s7*(-0.298)),s8*(-~
0.432)),s9*(-0.398)),s10*(0.099)),s11*0.100),s12*0.364)
#
def mbSPCA2(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12):
   return GridMath.add(add(add(add(add(add(add(add(add(add(add(s1*-~
0.186,s2*0.228),s3*(-0.229)),s4*-0.317),s5*(0.671)),s6*(0.101)),s7*-0.424),s8*(-~
0.024)),s9*0.124),s10*(-0.045)),s11*-0.087),s12*(-0.308))
#
def mbSPCA3(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12):
   return GridMath.add(add(add(add(add(add(add(add(add(add(add(s1*0.155,s2*-~
0.554),s3*0.153),s4*0.154),s5*0.585),s6*(-0.075)),s7*0.127),s8*(-0.091)),s9*-~
0.303),s10*(-0.051)),s11*0.385),s12*0.082)
#
def mbSPCA4(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12):
   return
GridMath.add(add(add(add(add(add(add(add(add(add(add(s1*0.463,s2*0.333),s3*(-~
0.145)),s4*0.037),s5*(0.350)),s6*(0.112)),s7*0.284),s8*(-~
0.038)),s9*0.017),s10*(0.144)),s11*-0.358),s12*0.535)
#
def mbSPCA5(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12):
   return GridMath.add(add(add(add(add(add(add(add(add(add(add(s1*-~
0.788,s2*0.140),s3*(-~
0.088)),s4*0.016),s5*(0.147)),s6*(0.017)),s7*0.472),s8*(0.060)),s9*-~
0.064),s10*(0.226)),s11*0.123),s12*0.269)
#
def mbSPCA6(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12):
   return GridMath.add(add(add(add(add(add(add(add(add(add(add(s1*-~
0.121,s2*0.009),s3*(-0.110)),s4*-0.039),s5*(-0.010)),s6*(0.185)),s7*0.247),s8*(-~
0.040)),s9*-0.267),s10*(-0.863)),s11*-0.240),s12*(-0.013))
#
```

```
def mbSPCA7(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12):
   return
GridMath.add(add(add(add(add(add(add(add(add(add(add(s1*0.142,s2*0.396),s3*(-~
0.180)),s4*0.007),s5*(-0.013)),s6*(-0.239)),s7*0.027),s8*(-0.044)),s9*0.315),s10*(-
0.329)),s11*0.703),s12*0.171)


###################################################################################
# Purpose: To calculate The Vulnerability Model (SM)
# Description: This script will calculate SM1 and SM2 based on the coefficients
# developed by Maina (2007)
# Operation: Call this mbSM1(s1,s2,s3,s4,s5,s6,s7…) one by one
# from the Define a grid diagnostic script
# Note: ~ symbol to keep a next line with previous
###################################################################################

def mbSM1(s1,s2,s3,s4,s5,s6,s7):
#s1-7 = SPCA1 - 7
   return GridMath.add(add(add(add(add(add(s1*0.5649,s2*0.1344),s3*0.0837),~
s4*0.0765),s5*0.0416),s6*0.0366),s7*0.0272)

def mbSM2(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12):
#s = fuzzy layers
#s1 = mean sst
#s2 = coefficient of variation sst
#s3 = photosynthetically active radiation
#s4 = mean max sst
#s5 = hotspot sst
#s6 = ultraviolet radiation
#s7 = slope sst
#s8 = meridional currents
#s9 = zonal currents
#s10 = wind speed
#s11 = chlorophyll-a
#s12 = regression model
   return GridMath.add(add(add(add(add(add(add(add(add(add(add(s1*0.092,s2*0.097),~
s3*0.084),s4*0.098),s5*0.081),s6*0.091),s7*0.082),s8*0.073),s9*0.074),s10*0.091),~
s11*0.08),s12*0.055)
```