LAKE BERA AND LAKE CHINI WATER QUALITY MONITORING USING SUPPORT VECTOR MACHINE

SITI FATIHAH ASY SYURA BT MAT JUBIT

INSTITUTE OF GRADUATE STUDIES

UNIVERSITY OF MALAYA

KUALA LUMPUR

2012

LAKE BERA AND LAKE CHINI WATER QUALITY MONITORING USING SUPPORT VECTOR MACHINE

SITI FATIHAH ASY SYURA BT MAT JUBIT

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTSFOR THE DEGREE OF MASTER OF BIOINFORMATICS

INSTITUTE OF GRADUATE STUDIES

UNIVERSITY OF MALAYA

KUALA LUMPUR

2012

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Siti Fatihah Asy syura bt Mat Jubit

(I.C/Passport No:) 870915355416

Registration/Matric No: SGJ 100008

Name of Degree: MASTER OF BIOINFORMATICS

TITLE("this Work"): Lake Bera and Lake Chini Water Quality Monitoring Using Support Vector Machine (SVM)

Field of Study:

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any Copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:
Subscribed and solemnly declared before,

Witness's Signature Date:
Name:
Designation:

ABSTRACT

Water quality monitoring is very important to control the quality of water. Lake Bera and Lake Chini which are known as a very important wetland are used to apply SVM method to predict its water quality. The output used to predict the classification of high medium and low is the dissolved oxygen according to the standard provided by the Interim National Water Quality Standard of Malaysia and Department of Environment. The training and test data is divided to 80% for training data and 20% for testing data. The SVM is implemented using R software package kernlab which used ksvm as its implementation to do prediction. Kernel Anova was used to create the model. The result shows that the predicted accuracy is about 74%.

ABSTRAK

Kawalan kualiti air memainkan peranan yang sangat penting kerana ia membolehkan kualiti air dapat dikawal daripada dicemari. Tasik Bera dan Tasik Chini yang dikenali sebagai kawasan tanah lembap yang besar peranannya digunakan di dalam projek ini untuk dibuat ramalan kualiti air dengan mengaplikasikan kaedah SVM iaitu salah satu kaedah automatik. Kualiti air ditentukan dengan menggunakan oksigen terlarut sebagai kelas penentu untuk tasik ini berdasarkan sama ada oksigen terlarut berada di kelas yang tinggi, sederhana atau rendah berdasarkan Standard Kualiti Air Kebangsaan di Malaysia dan JabatanAlamSekitar. Data yang digunakan dibahagikan kepada data latihan dan data untuk percubaan di mana 80% digunakan sebagai data latihan dan 20% untuk percubaan. Kaedah SVM diaplikasikan dengan menggunakan perisian R di mana pakej kernlab digunakan dan ia mengaplikasikan ksvm yang dapat membuat ramalan untuk kaedah SVM. Berdasarkan kaedah kernel yang dapat digunakan kernel Anova digunakan dalam kaedah ini. Ketepatan ramalan yang di peroleh adalah sebanyak 74%.

ACKNOWLEDGEMENT

I would like to thanks my supervisor and my co supervisor Dr. Sorayya Bibi Malek binti Malek Abd Rashid and Dr. Sharifah Mumtazah binti Syed Ahmad Abdul Rahman whom have guided me in completing this project especially on using the R to do SVM. I also would like to thanks my friends who have helped me understanding the SVM. Lastly I would like to express my thanks to my family who has kept supporting me throughout this project.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

ANN	Artificial Neural Network
ASMA	AlamSekitar Malaysia Sdn. Bhd
BOD	Biochemical oxygen demand
C	Constant
COD	Chemical oxygen demand
COND	Electrical conductivity
CPU	Colony Forming Units
CSV	Comma delimited
DNA	Deoxyribonucleic acid
DO	Dissolved Oxygen
DOE	Department of Environment
DS	Dissolved solid
ERM	Empirical Risk Minimization
INWQS	Interim National Water Quality Standard, Malaysia
Kernlab	Kernel-based Machine Learning Lab
mg/l	Milligram per liter
Ml	Miligram
NH3N	Ammonia nitrogen
NO3	Nitrate nitrogen
NTU	Nephelometric turbidity units
PO4	Phosphate
Ppt	Parts per thousand
QP	Quadratic Problem
SLT	Statistical learning theory
SMO	Sequential Minimization Optimization
SS	Suspended solid
SVM	Support Vector Machine

1. INTRODUCTION

Lake Chini and Lake Bera are the largest natural lake situated in Pahang state of Malaysia. These lakes play role as wetland which help preventing flood and erosion also as habitat for various flora and fauna which some of them are endangered species. It is important to keep this source to stay in good condition to balance the ecosystem and indirectly the lakes also one of the daily water source for the local people. Water quality monitoring is important to help in managing the water quality so that it will be safe whether as drink water, daily chores uses or for the fish to live.

1.1. Lake Bera and Lake Chini

Listed as a RAMSAR in 1994 for its importance of nature conservation Lake Bera has been preserved from development because it can disturb its nature. It is an example of blackwater ecosystem which consists of swamp area and swamp forest with grassland on the periphery. It lies in the basin of Peninsula's Malaysia largest river of Pahang river. Lake Bera eventually discharging into the South China Sea from as it flow through Bera River and then Pahang River. Lake Bera is a known habitat for many types creature such as animals and plants which some are rare and endangered species .This situation give a great opportunity for researchers as it is an important area containing a great combination of research to conduct.

Lake Chini is ranked as after Lake Bera as the largest natural lake where it is in second place. Lake Chini also plays role which Lake Bera has because they both served as wetlands such as preventing the floods and erosion of its riverbank. Then it can help the surrounding area save from the damage made by the flood. Lake Chini also provided a very important source especially to the local people as they can get the fishes in its lake since the lake also create a very good nature for fishes to breed and a place for fishes that migrate. Lake Chini flows by Chini River and then flows into Pahang River the same as Lake Bera.

However somehow the lakes encounter threat from its basin or within the lake itself and this has made some changing with their ecosystem. With development happen around the lake it affected to the damages of the plant around the area, the aquatic habitat and their system and made the lake to be polluted. Their role as wetlands are lost due to these problems mentioned and thus decreases the water quality. Therefore a study is needed to assess the quality of water because it is important for managing its quality for a sustainable management.

1.2. Water Quality Monitoring

Water quality have different requirement based on what purpose it will be used for such as to supply clean water to household or to cool down generator. Water quality as it has different requirement can be defined as chemical, physical, and biological characteristics of the water.

Water-quality monitoring is important because we can monitor and use it to control pollution that happen with water so that it can stay clean and safe to be used. There are several factors that can influenced water quality as it comes from physical factor, human activities such as development project, meteorology, chemical effects and many other factor.

In Malaysia Water Quality Monitoring has been established through National Monitoring Network Established in which was started in 1978. Several aims been created where the water quality monitoring is created as a platform to monitor the water quality status of river water in Malaysia and also to check the development activities as if it is affecting the water quality. They check the water quality based on continuous basis. DOE and ASMA has been collaborating to do the water quality monitoring as a result from high demand of monitoring.

As establish by Interim Water Quality Standards for Malaysia which is Interim National River Water Quality Standards, there are five classes of water quality standards table as

below and several important parameter used in monitoring. As in this project we use DO as a parameter to be predict based on the class of high, medium or low because it is an important parameter which can easily show the water quality whether it is in good condition or it is not in good condition.

Table1. 1 Interim National River Water Quality Standards

Class	**Description**
Class I	Conservation of natural environment, Water Supply I – practically no treatment necessary, Fishery I – very sensitive aquatic species.
Class IIA	Water supply II – conventional treatment required, Fishery II – sensitive aquatic species.
Class IIB	Recreational use with body contact
Class III	Water supply III – extensive treatment required, Fishery III – common, of economic value, and tolerant species livestock drinking
Class IV	Irrigation
Class V	None of the above

Source: Department of Environment Malaysia

Table 1.2 Class and parameters standard

Class **Parameter**	**I**	**II**	**III**	**IV**	**V**
BOD	***<1***	***1-3***	***3-6***	***6-12***	***>12***
COD	***<10***	***10-25***	***25-50***	***50-100***	***>100***
NH3N	***<0.1***	***0.1-0.3***	***0.3-0.9***	***0.9-2.7***	***>2.7***
DO	***>7***	***5-7***	***3-5***	***1-3***	***<1***
pH	***>7***	***6-7***	***5-6***	***<5***	***>5***
SS	***<25***	***25-50***	***50-150***	***150-300***	***>300***
WQI	**>92.7**	**76.5-92.7**	**51.9-76.5**	**31.0-51.9**	**<31.0**

Source: Department of Environment Malaysia

1.3. Support Vector Machine (SVM)

SVM is a tools created to solve the classification problem. It works by separating data into training and testing data and are gaining popularity due to many attractive features. Furthermore SVM now can also solve the regression problems. SVM and neural network share a quite similar process and system however SVM is has more advantages in solving a complex and nonlinear data since it use kernel function that can provide more solving method for the problems.

SVM modeling works by finding the best line to separate training data according to the classification chosen and place the data in the plane made by the line that has been created. There are vector that will be place near the line and it is the support vector.

1.4. Scope

Water Quality Monitoring requires a lot of effort, time and also cost. Besides, calculating water quality parameter is hard to get the accurate results since it is a nonlinear system. Some model has been recognized to solve this problem such as the Artificial Neural Network and SVM. SVM has been applied in many research fields and successfully solve the nonlinear system with its regression algorithm.

This project will use data from 15 stations from Lake Chini and Lake Bera which were sampled every two month from February 2005 until October 2009. 4 parameters were analyzed such as pH, temperature, turbidity, dissolved oxygen (DO) as the output to be predicted. The SVM also will be implemented using R.

1.5. Objectives

This project serves two objectives as below:

1. To predict DO classification of low medium and high at Lake Chini and Lake Beraby using SVM model.
2. To develop the SVM model based on the DO classification.
3. To test the accuracy of SVM model for its accuracy in the predicted DO.

2. LITERATURE REVIEW

Water is very important in our daily life in fact all living creature needs water to survive however the importance of this source also means that it is very possible to be polluted which will reduce its quality hence it will become a bigger problems in the end. Xiang and Jiang (2009), Xul, Wang, Guan and Huang (2007) in their articles agreed that monitoring the water quality and the forecast for it is a very important task to do. They mentioned that development such as economical activities is contributing to the pollution of water and that make it is a crucial task to monitor the water quality as to quickly solve the water pollution problems. As for the arisen problem there have been many researches that conducted the water quality forecast model. They mentioned that when we can determine the water quality parameter it can show at which level the evaluation of the water is and thus we can prevent the water pollution. In addition to the monitoring the traditional method used before offer a lot range of the monitoring parameters used. However water quality unfortunately can be directly polluted by so many factors such as limited manpower, materials, climate, landform, hydrologic conditions therefore the traditional way to monitor water quality is not really efficient to solve these problems because to do water quality forecast it involves a complex and nonlinear data. Moreover there are many problems with traditional way that may be from the human source and limitation and some other technical problems arise during inspection and make the monitoring become not productive. Therefore a new method which can increase the forecast of water quality is necessary to maintain the water quality.

Regarding the methods used in water quality monitoring, currently there are two main methods for monitoring and evaluating water quality as first by physical and chemical analysis, and second, biological monitoring methods as mentioned by Liao, Xu and Wang (2012). Water quality is evaluated by determining the existence and content of hazardous substances within the water directly using a variety of instruments. These physical and chemical analysis methods are accurate and sensitive, but they are time-consuming and cannot be used continuously in situ. While biological monitoring is to detect if there is any changes whether it involves in water quality itself or if there is presence of pollution by identifying changes in the health status, physiological characteristics, and behavioral responses of individuals or populations of aquatic organisms, providing a basis for environmental quality monitoring and evaluation from a biological point of view. Biological methods are once a system is established it can provide automatic alarms and can be used for long-time online monitoring of water quality. Furthermore the response of aquatic organisms to water quality is more sensitive and reliable and biological methods are also useful for detecting mixed pollution. Lastly they have a low cost and can easily be incorporated into a digital system.

Liao, Xu and Wang (2012) also agreed that traditional method do not solve the complex nonlinear relationships between assessment factors and water quality, and the assessment result is greatly affected by subjective factors of the assessing person.

Bouamar and Ladjal (2007), Xul, Wang, Guan and Huang (2007), Liu, Chang and Ma (2009) in their articles said that the automation tool of artificial intelligence techniques can provide a better result from the data that will be used which get directly from the monitoring station or the raw data. These data is known to be complex and is a nonlinear data which is hard to deal so the tool mention can be used to do the decision making aid. One of the tools that gain attention is SVM which has been successfully applied in many areas to do forecasting such as in biological area. SVM is a statistical learning theory (SLT) where it uses structural risk minimization principle with good generalization ability. It can solve the problem that conventional methods face in assessing water quality and can overcome the defects of slow training speed, poor network generalization, and low learning accuracy in artificial neural networks (ANNs). It also can fully utilizes the distribution feature of training samples to construct discriminant function based on part of the training samples, describing such nonlinear relationship. They also said that calculation result of their data shows that SVM has favorable classification performance and can be applied in water quality assessment. Besides, Liu, Chang and Ma (2009) mentioned when using SVM there are three main issues need to be considered such as feature selection, kernel function selection, and the penalty and inner parameters of kernel function selection.

In Bouamar and Ladjal (2007) they use the SVMs technique to solve pattern recognition and clearly satisfied with the result produce from the SVM technique but there also error produce from it and they conclude that with the increasing of training data and new sensor the precision can be improved.

In Liu, Chang and Ma (2009) in their journal mentioned that as water quality assessment is a complex data (SVM) can transform the learning process into a convex quadratic planning problem to get a global optimization by using the rule of minimum structure risk, which is appropriate to solve small-sample, nonlinear classification and regression issues. They apply SVM in water quality assessment for karst groundwater sample at the Niangziguan fountain region of Haihe River basin to obtain the grade of water quality assessment. The result shows that such a method solves the complex nonlinear relationship between assessment factor and water quality grade. It offers high prediction accuracy and is a reasonable and feasible assessment method.

When we mentioned about SVM there is another tool we cannot forget which is really close to SVM. Researchers often do the evaluation between these two tools. Bouamar and Ladjal (2007), Xiang and Jiang (2009) evaluate both tools in their paper which other tool is the Artificial neural network (ANN). ANN can also process nonlinear data and it also can gives result with high accuracy however it is itself a complex structure and made it poor in its performance.

Bouamar and Ladjal (2007), in their finding found SVM can deal with complex and highly nonlinear data with good result even the training sample is only few. They did an evaluation of ANN and SVM techniques which both of the tools shows a highly good results about 86%. They found that the corresponding time of ANN to the training data is better but it is not very sensitive to the noise produce while SVM is good when dealing

with this noise and it shows that as for water quality monitoring the SVM is a better tool to do the forecasting of water quality.

SVM prediction method used to do DO prediction however has been done in many areas such as in Najah et al. (2011), they use different kind of machine learning method to do prediction of DO such as the ANN, ensemble and also SVM. The research was done for the river water of Johor state which is for Johor River. Based on the research SVM give the best performance among all the methods that have been used in doing the DO prediction. Other machine learning method used to do DO prediction such as ANN method which was establish earlier then SVM and also by using mathematical method. In Palani et al, (2008) they have agreed that DO prediction is successfully done using ANN method. Based ontheir finding the ANN method used gives a good result when they do the prediction for seawater for Singapore which they obtain acceptable accuracy. In Junsawang, Pomsathit and Areerachakul (2011), Naik and Manjapp (2010) they agreed that DO is the best parameter to indicate the water quality. The prediction done in their research for river in Thailand and India is by using regression method where they predicted the DO value and not the classification of DO. The results they get show a good production of predicted value of DO. It is shown that the prediction is almost as the actual value in their data. They are satisfied with the result obtain by using their methods.

3. METHODOLOGY

3.1. Support Vector Classification

SVM can solve classification or the regression problem and in this project SVM is used to solve classification problem. As the early development of SVM it was create to solve classification problems. The classification problem involves separating data into training and testing sets where it contains the class label and variables input of data. SVM then from the training data will create a model which going to be used to predict the class label on the testing data.

SVM works by predicting the labels of training data as $D = \{(\overrightarrow{x_i}, y_i), i = 1..N\}$ with $y_i \in \{-1, +1\}$ is separable by a hyperplane. Where if the data is positive it belongs to class 1 and if the data is negative it belongs to class -1. When data is linearly separable and support vector existed it can be derived as:

$$f(\vec{x}) = \vec{w}^T \vec{x} + b (= w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_n x_n + b)$$

W and b are determined during training process where it locates the support vector. W is defined as weight vector while b is term for bias and is a scalar. T stands for transpose operator.

Data in SVM can be a linear or nonlinear data. For a linear and separable dataw and b is minimize to $^1/_2 \|w\|^2$

Subject to constraints: $y_i(w^T x_i + b) \geq 1, \ \forall i$

To solve the optimization Lagrangian function defined as below is used:

$$L(w, b, a) = \frac{1}{2} w^T w - \sum_{i=1}^{N} a_i [y_i(w^T x_i + b) - 1, \qquad a_i \geq 0, \forall i]$$

α_i is required to express w. When $\alpha_i > 0$ it is called the support vectors and it resulted in

$$w = \sum_{i=1}^{N} a_i y_i x_i$$

$$\sum_{i=1}^{N} a_i y_i = 0$$

The Lagrangian then derived into the dual optimization problem as follow

maximizationα

$$\sum_i a_i - {}^1/_2 \sum_i \sum_j a_i a_j y_i \, y_j x_i^T x_j$$

then

$$\sum_{i=1}^{N} a_i y_i = 0 \, a_i \geq 0, \forall i$$

SVM for linear predictor can be expressed from as below

$$f(x) = w^T x_i + b = \sum_{i=1}^{N} a_i y_i x_i^T x + b$$

where

$$b = \frac{1}{\left|I_{support}\right|} \sum_{i \in I_{support}} \left(y_i - \sum_j a_j y_j x_j^T x_i \right)$$

andI_{support} is the set of support vectors.

If f(x) is positive the test data will belong to class of y_i =1 and for negative it will be classified into the other class.

The nonlinear data equation on the other hand can be derived as

$$^1/_2 \|w\|^2 + C \sum_i \varepsilon_i^2$$

$$y_i(w^T x_i + b) \geq 1 - \varepsilon_i, \qquad \varepsilon_i \geq 0, \forall i$$

Here $C > 0$ is a constant of positive number on a selected parameter. $\varepsilon_i$shows distances of data lying on false class side and its margin of predicted class.

As in the linear problem where it can be optimized into dual problem it also can be used in nonlinear case. Optimization problem can be converted into dual problem as:

$$\sum_i a_i - {}^1\!/_2 \sum_i \sum_j a_i a_j y_i \, y_j x_i^T x_j$$

subject to

$$\sum_{i=1}^{N} a_i y_i = 0,$$
$$0 \le a_i \le C, \ \ \forall i$$

By this it is dependable on the data to choose the appropriate C.

In SVM kernel function is introduced as it offer a better performance when dealing with a nonlinear data. Here data is mapped and derived as below where K is the kernel function

$$\Phi(x)^T \Phi(y) = K(x, y)$$

By introducing the kernel it can be derived as

$$\sum_i a_i - {}^1\!/_2 \sum_i \sum_j a_i a_j y_i \, y_j K(x_i x_j)$$

subject to

$$\sum_{i=1}^{N} a_i y_i = 0,$$
$$0 \le a_i \le C, \ \ \forall i$$

The equation produce is as below

$$f(x) = w^T \Phi(x_i) + b = \sum_{i=1}^{N} a_i y_i K(x_i, x) + b$$

The kernel selection depends on the data distribution but kernel selection also generally done through trial and error.

3.2. Kernel-based Machine Learning Lab (Kernlab)

The SVM was implemented using R software. Kernel-based Machine Learning Lab (Kernlab) package was used to do the SVM. Along kernlab there are a few more packages in R that can be used to do SVM. Such as R package e1071 which is very efficient SVM implementation. Another SVM related R package available is klaR.

In kernlab R user is provided with basic kernel functionality and other functions. Besides user can also create own function to the kernel based on the kernel is in the package. It is applied to class system of S4 where declaration is needed of every step taken by the user and it is stricter however it is still consistent.

There are prepared data included in this package such as spam data set which classifies spam or non-spam. Promotergene, ticdata data set, spirals data set and lastly the income data set are more of the data included in this package.

In kernlab it supports about seven kernels such as:

The linear kernel

$$k(x, \acute{x}) = \langle x, \acute{x} \rangle$$

Gaussian radial basis kernel

$$k(x, \acute{x}) = exp(-\sigma \|x - \acute{x}\|^2)$$

The polynomial kernel

$$k(x, \acute{x}) = (scale. \langle x, \acute{x} \rangle + offset)^{degree}$$

The Hyperbolic tangent kernel

$$k(x, \acute{x}) = tanh(scale. \langle x, \acute{x} \rangle + offset)$$

The Bessel kernel

$$k(x, \acute{x}) = \frac{Bessel^{n}_{(v+1)(\sigma \|x - \acute{x}\|)}}{(\|x - \acute{x}\|)^{-v(v+1)}}$$

The Laplace radial basis kernel

$$k(x, \acute{x}) = exp(-\sigma \|x - \acute{x}\|)$$

The ANOVA radial basis kernel

$$k(x, \acute{x}) = \left(\sum_{k=1}^{n} exp(-\sigma(x^k - \acute{x}^k)^2) \right)^d$$

The linear kernel is known as vanilladot is used when we predict the linear data. The Gaussian radial basis kernel is known as the rbfdot which is usually used for doing the classification prediction and for general purpose and in the rbfdot the parameter introduced is the sigma. This is the same as Laplacian radial basis kernel or the laplacedotwhich has the sigma parameter. Lapelacedot is usually used for general purpose also. The Polynomial kernel used three different parameter which known as scale, offset and degree in its function and usually the polynomial kernel that is known as polydot is used for the image classification. In Hyperbolic tangent kernel which is known as tanhdot is usually used in neural network purpose and the parameter we need to determine are scale and offset. The Bessel kernel is used for general purpose and it is known as besseldot. The parameters used in this kernel are sigma, order and degree. Lastly the Anova kernel is known as anovadot usually deal with the multidimensional regression problems and the parameters used are sigma and degree.

Ksvm as an implementation used in kernlab is a very efficient method because of it functions and also because it applied the C-SVM classification which can also predict for multiclass classification problems where some method that it uses are like one-against-one method and the other method is pairwise classification method. These methods used

voting method to do prediction and have been shown to produce good results when used with SVM.

Another method that can be used is by solving the problem by including the data from all classes such as derived:

$$t(w_{n,}\varepsilon) = \frac{1}{2}\sum_{n=1}^{k}\|w_n\|^2 + \frac{C}{m}\sum_{i=1}^{m}\varepsilon_i$$

subject to

$$\langle x_i, w_{yi}\rangle - \langle x_i, w_n\rangle \geq b_i^n - \varepsilon_i$$

where

$$b_i^n = 1 - \delta_{\mathrm{yi,n}}$$

where the decision function is

$$argmax_{m=1,\ldots\ldots,\mathrm{k}}\langle x_i, w_n\rangle$$

For R package kernlab it is adapted to R new and modern functions where it allows the used to explore using its package by bravely constructing new kernel function of the algorithm existing in it. As for ksvm it helps improving the prediction by allowing multiclass problem classification.

3.3. Data

Lake Bera consist of 26 000 hectares of its core zone and 27 500 hectares the buffer zone all has been preserved as RAMSAR sites and it is coordinate at 3°49′00″N102°25′00″E. Lake Chini consists of about 5026 hectares and is situated in coordinate 3°26′N102°55′E. Both Lakes have the climate of equator in Peninsular Malaysia which having the humidity, temperature and rain fall at an average characteristic.

Figure 3.1 Lake Bera Map, Source: go2travelmalaysia.com

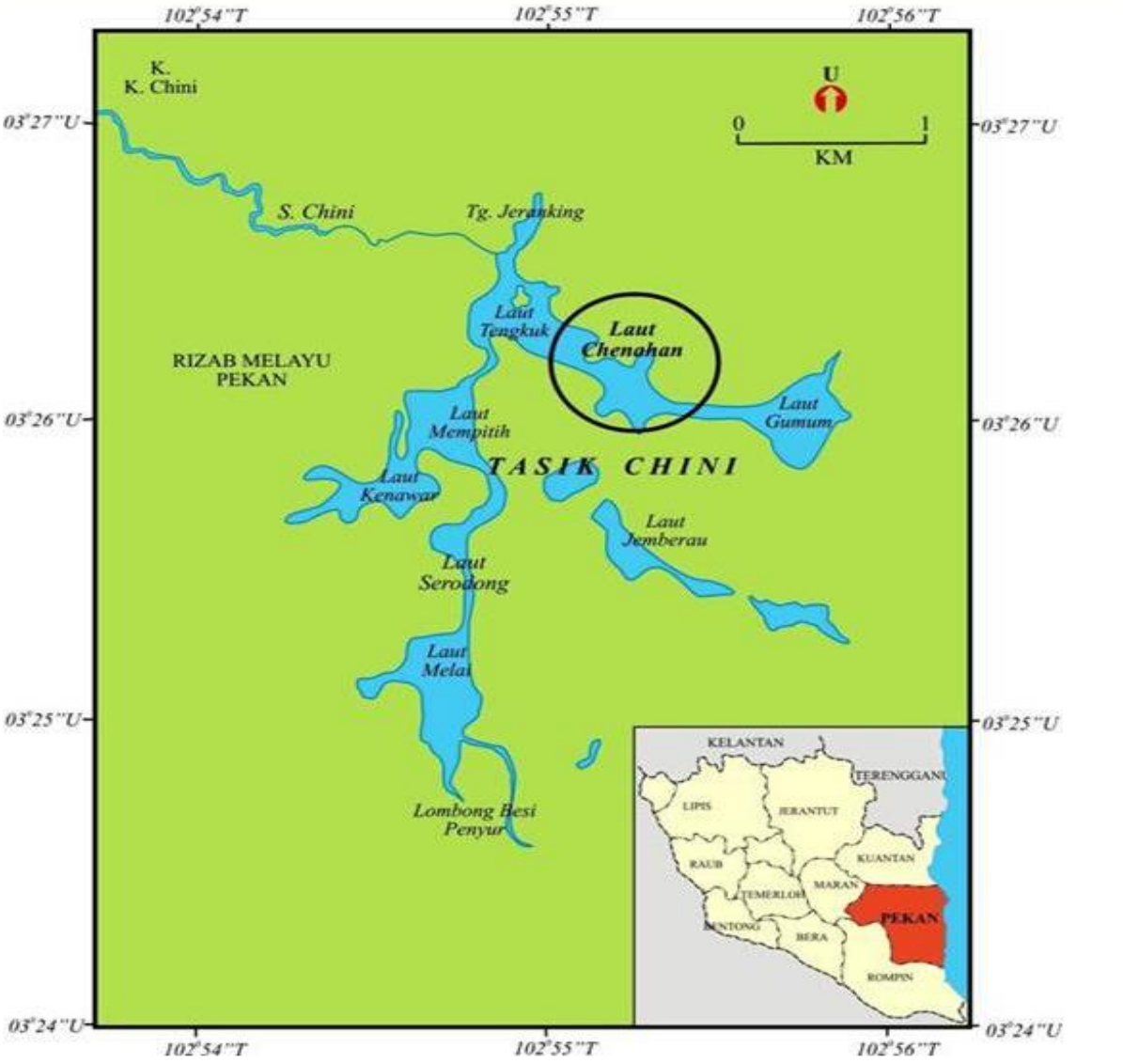


Figure 3.2 Lake Chini Map. Source: http://www.ukm.my/ahmad/tesispelajar/fitochenahan_files/image319.jpg

In Shuhaimi Othman, Lim and Mushrifah (2007) based on the research conduct on Lake Chini the water quality is decreasing due to the pollution that happen because of development.

Lake Bera and Lake Chini water quality data were collected starting from February 2005 until October 2009. The lakes were monitored regularly for every two month during mentioned years above which is from 2005 until 2009. It was monitored from six stations for Lake Bera and from Lake Chini it was monitored by nine stations. The stations monitored are as followed below:

Table3. 1 Monitored station of the lakes

Lake	Stations
Lake Bera	4PH03, 4PH07, 4PH66, 4PH67, 4PH71, 4PH72
Lake Chini	4PH75, 4PH76, 4PH77, 4PH78, 4PH79, 4PH80, 4PH81, 4PH82, 4PH83

In this project we used 4 parameters of the lake data. The parameters that were measured and put in the data such named as pH (pH), temperature, turbidity and dissolved oxygen (DO).

The pH is used as the indicator for the water to be determined whether it is acidic or alkaline. In the standard state by the Department of Environment the pH suitable and safe for Malaysian rivers range between 5.00 to 9.00 and the lakes results are within the range which is from the lowest are 4.97 until 7.94. The pH was slightly fall from the range when February 2005. Temperature varying from 26.51 until 33.63 degree Celsius and turbidity are from 1 and 282.2 *nephelometric turbidity units (*NTU). All of these variables are the factor that has the most affecting factor to the DO level whether it is good or not. As for the DO chosen to be the output because DO is a good indicator to know whether

the river is clean and save or not because it shows how many oxygen can be dissolved in water and if the creature in water can survive with the DO condition at certain level.

3.4. Data Prediction Using SVM

The water quality data has about 147 samples and 11 variables. 10 variables were the input used to do the SVM and DO is used as the output in this project. DO variables were labeled in classification as high, medium and low based on the classification standard by the Interim National Water Quality Standard, Malaysia (INWQS) and Department of Environment as in table 1.2. In this classification the high class of DO range for value 7 above while the medium class ranges between 5 until 7 and lastly for the DO that had value below 5 is in class low. Data selection then is done by eliminating the data that was empty or was not available data. Then data was divided into training and testing data set to do the SVM. It was divided by 80% for training data and 20% for testing data. 80% data for training data consist of 80% of class high, medium and low also for testing data is vice versa. Next data was converted into comma delimited (CSV) format before it can be used to run in the R software.

To run SVM in R we need to call the kernlab package. Then data that was in the CSV format was imported into R by using function read csv. Data was reviewed in R by using summary function where we can see the information of data such as the mean, median, the min and others. The parameter was tested one by one to see which data has the least error where here we can see which parameter is relevant to use to do the prediction using SVM. Parameter was tested by using linear kernel which known as vanilladot to

determine the cross validation error. Result from using the linear kernel function then was ranked from the least error result until the parameter that has the big error.

After determined which parameter was ranked in ascending ranking then we do the selection of the best parameter that can compute highest accuracy by doing forward selection. Parameter was added one by one and after that kernels function was tested on them to see which kernel suits best by seeing the error that produced when running the program. The kernel that has least error was then chosen to be used as kernel for the SVM model. In this step default value was used for each kernel.

Kernels have different parameter used in their function and depend on their parameter we need to choose the best parameter value which can generate least error and finally when we do prediction function it gives high accuracy. To determine the least error for parameter in kernel we did the loop function in the kernel chosen then from the resulted calculation we chose the value with least error. From the best value of each parameter we then inserted it to the model and run it. After that we did the prediction function to know the prediction that has been done by the SVM. To see the cross tabulation table of prediction and actual data table function is used. Lastly the calculation for accuracy of prediction is done automatically using R. The model evaluation is done by using the sampling of cross validation error and using the sensitivity, specificity and accuracy method.

4. RESULTS

The data of Lake Bera and Lake Chini shows that DO levels of these lakes mostly is at the level medium and from this level the lake is classified as the water need to be treated to be as a supply and it involves the sensitive aquatic species.

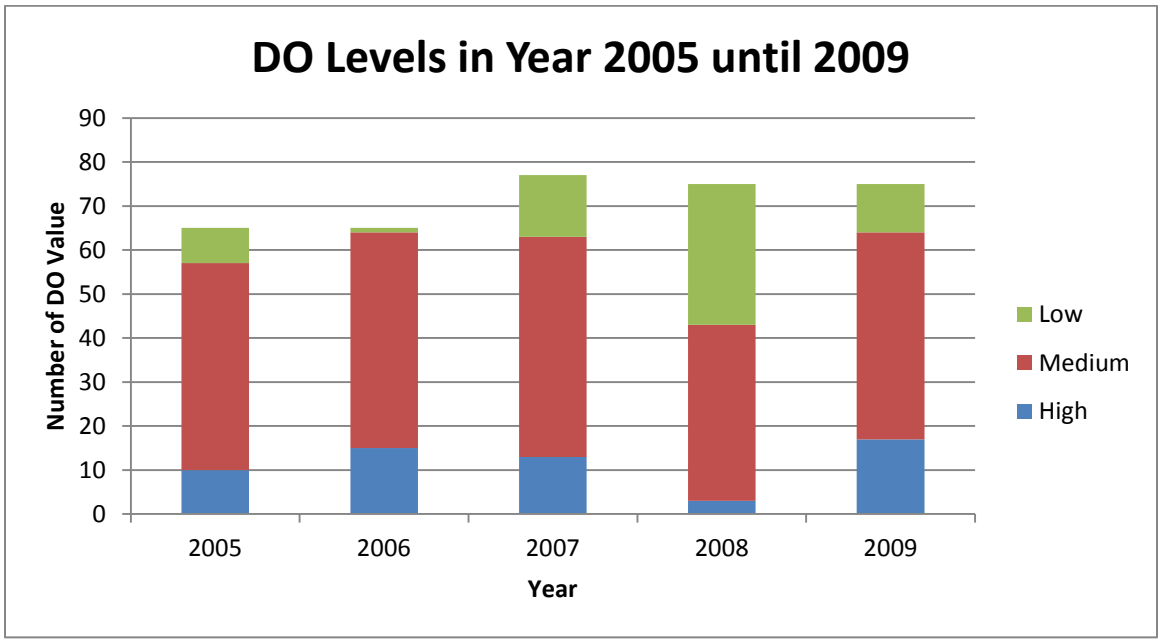


Figure 4. 1 DO Levels at Lake Bera and Lake Chini

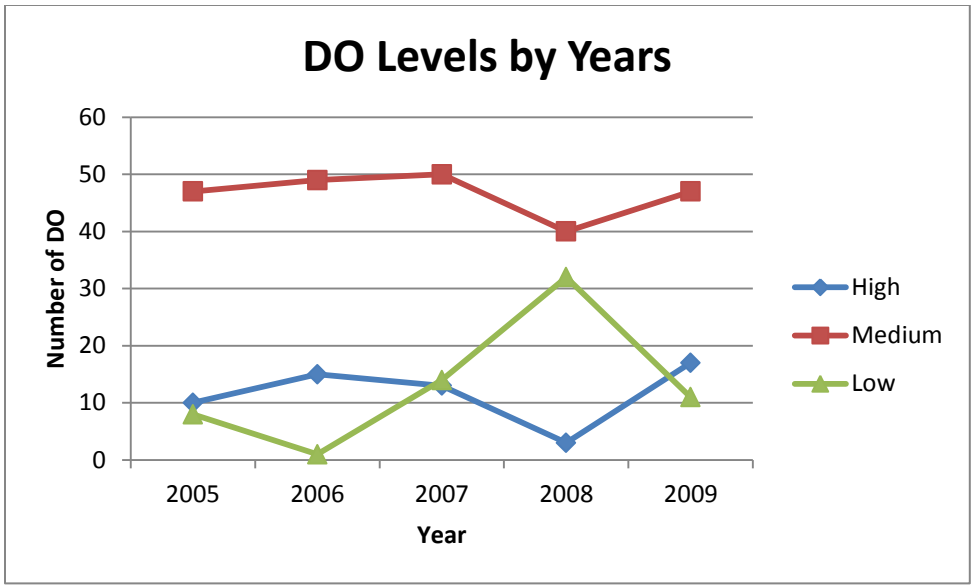


Figure 4.2 Levels of DO by year

As seen in the graph year 2008 shows that the lakes decreases in high level and low level of DO is increasing.

In the SVM method used to do prediction based on DO classification firstly each parameter is tested to see whether it is suitable to use as an input and from the result that generate by R shown below on its ranking with the least error until parameter that has big error:

Table 4.1 Parameter ranking

Ranking	Parameter	Error
1	Temperature	0.458333
2	pH	0.516667

Table 4.1 continue…

3	Condition	0.575
4	Coliform	0.608333
5	Turbidity	0.625
6	Natrium	0.675
7	Salinity	0.691667
8	Phosphate	0.7
9	*E.coli*	0.708333
10	Nitrate	0.8

When doing forward selection the data was added up one by one according to the ranking above. In each step of determining which parameter is good, the determination of which kernel was best also been done. When 2 and 10 inputs were used the best kernel was the radial basis and 3, 4, 5, 7, 8 and 9 inputs show that Anova is the best kernel. Lastly with 6 inputs of parameter it was suitable using Laplacian kernel. With the best kernel then SVM model are run and the accuracy produce from these kernels when run with certain inputs parameter is shown in table below.

Table 4.2 Accuracy for kernels

Inputs	Parameter	Kernel	Accuracy (%)
2	Temperature, pH	Radial Basis	44.44
3	Temperature, pH, Condition	Anova	74.07
4	Temperature, pH, Condition, Coliform	Anova	62.96
5	Temperature, pH, Condition, Coliform, Turbidity	Anova	70.37
6	Temperature, pH, Condition, Coliform, Turbidity, Natrium	Laplacian	66.67
7	Temperature, pH, Condition, Coliform, Turbidity, Natrium, Salinity	Anova	42
8	Temperature, pH, Condition, Coliform, Turbidity, Natrium, Salinity, Phosphate	Anova	48. 14
9	Temperature, pH, Condition, Coliform, Turbidity, Natrium, Salinity, Phosphate, *E.coli*	Anova	44.44
10	Temperature, pH, Condition, Coliform, Turbidity, Natrium, Salinity, Phosphate, *E.coli*, Nitrate	Radial Basis	51.85

From the table shown when using 3 inputs parameter using kernel Anova the highest accuracy that was obtain about 74%. This means that the best 3 parameter is relevant to use in doing the SVM prediction. Furthermore it is shown that when parameter is added

up the accuracy is decreasing. The best five parameters which shown good accuracy result is actually the factor that gives effect to the DO. With the result shown it is also prove that DO is affected by these five parameters. However in this project 3 inputs were used as it provides highest accuracy among all.

In kernel Anova the parameters need to be determined in this function are degree, sigma and cost C. By doing the loop function the degree obtain is 1.5, sigma is 1 and the cost C used for this model is 24. This model computes the cross validation error about 0.575. The accuracy as mentioned above is 74.07%. The prediction made by this model can be seen in the figure below.

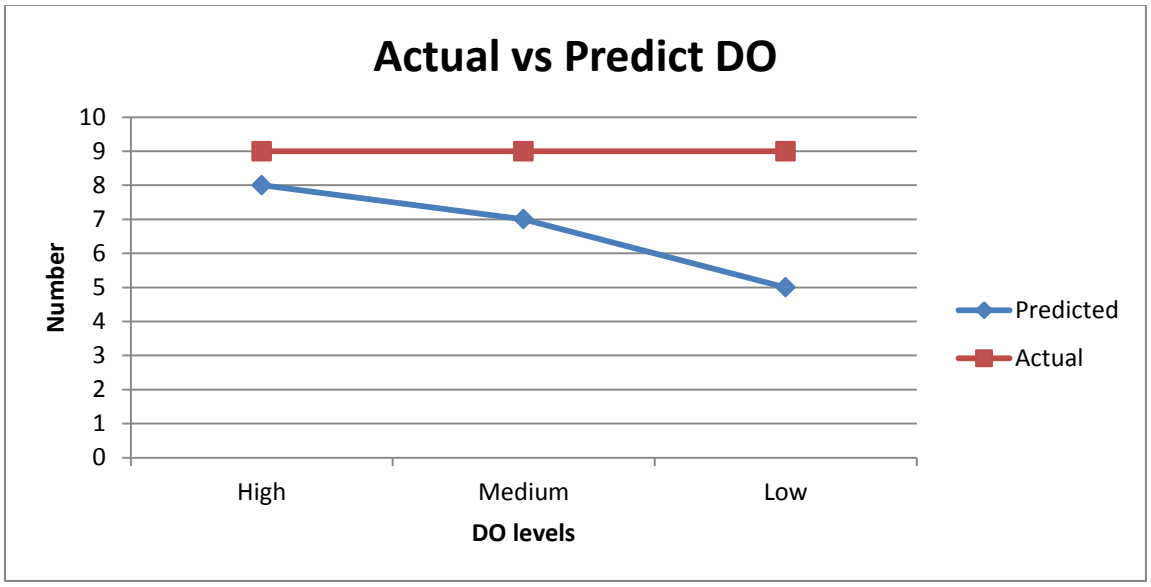


Figure 4.3 Predictions and actual value of testing data

Table 4.3 Prediction result on testing data

DO	pH	TEMP	COND	Prediction	Prediction result
Medium	6.18	30.13	28	Medium	TRUE
Medium	6.45	31.17	28	Medium	TRUE
Medium	6.34	30.97	29	Medium	TRUE
Medium	6.77	31.065	28	Medium	TRUE
Medium	6.24	30.67	32	Medium	TRUE
Medium	6.3	30.6	29	Medium	TRUE
Medium	6.07	29.93	28	Medium	TRUE
Medium	7.59	26.12	30	High	FALSE
Medium	7.77	27.7	31	Low	FALSE
Low	5.79	28.46	26	Low	TRUE
Low	5.81	28.12	22	Low	TRUE
Low	6.41	29.88	22	Low	TRUE
Low	7.03	29.67	78	Low	TRUE
Low	6.66	28.37	33	High	FALSE
Low	6.24	32.64	37	Medium	FALSE
Low	6.15	30.95	25	Medium	FALSE
Low	5.98	32.62	26	Medium	FALSE
Low	6.9	29.82	69	Low	TRUE
High	6.67	30.72	26	High	TRUE
High	6.68	30.47	25	High	TRUE
High	7.01	33.47	23	High	TRUE
High	7.26	30.71	125	Medium	FALSE
High	6.64	30.91	43	High	TRUE
High	6.71	30.28	26	High	TRUE
High	6.42	32.86	27	High	TRUE
High	7.89	31.54	21	High	TRUE
High	7.42	32.54	23	High	TRUE

Results shown that this model predicted the high classification as the most correctly predicted, and after that medium class and after that is the low class. From the prediction that has been made sensitivity, specificity and accuracy for each class can be calculate as this shown figure below.

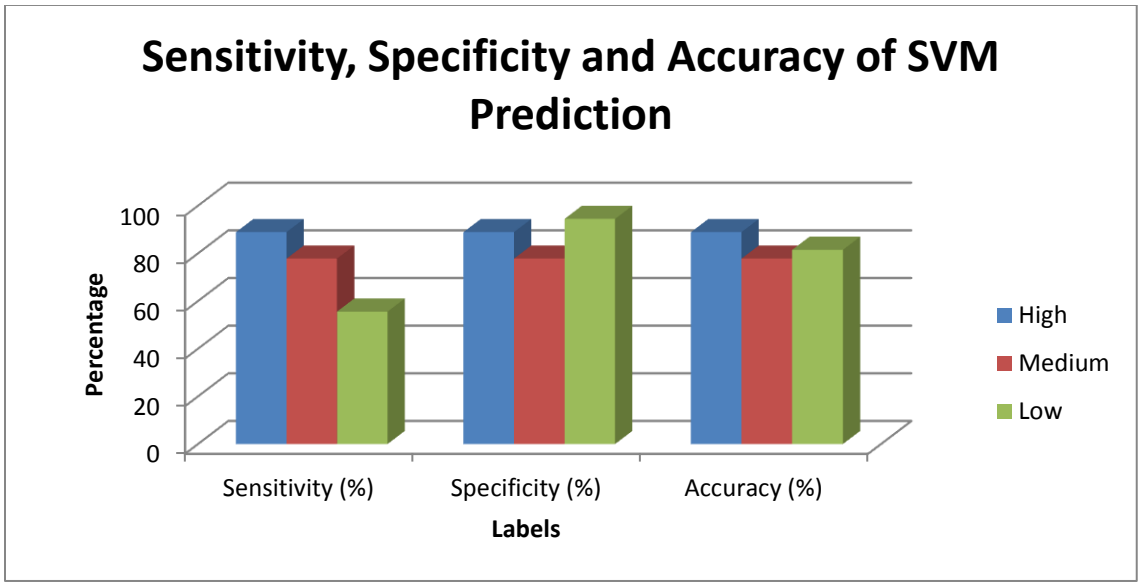


Figure 4.4 Sensitivity, Specificity and Accuracy Graph

Table 4.4 Calculated sensitivity, specificity and accuracy

	Sensitivity (%)	Specificity (%)	Accuracy (%)
High	88.89	88.89	88.89
Medium	77.78	77.78	77.78
Low	55.56	94.44	81.48

Overall accuracy is 74.07 % and as shown in the table the sensitivity for high class is 88.89% while specificity is 88.89% and for medium class the sensitivity is 77.78% while specificity is also 77.78%. For low class of DO sensitivity is 55.56% and specificity is 94.44%. The accuracy for high class calculated about 88.89% and medium class accuracy is 77.78%. Lastly class low accuracy is 81.48%.

5. DISCUSSIONS

Based on the result present in the previous chapter, the prediction made using SVM model produce error which is about 0.5 from the cross validation error method and the accuracy about 74% when done automatically using R. The accuracy is calculated for the prediction that matches the actual data over overall data. The research done by other researcher such as Najah et al. (2011), Bouamar and Ladjal (2007), Xul, Wang, Guan and Huang (2007), Liu, Chang and Ma (2009)when using SVM to predict data produce small value of error and the accuracy of their prediction is above 70%. The training was set to 80% containing each levels of DO also for 80% because we want the data to be sampled at fairly value. So SVM is predicted to give the best prediction when the entire sample is divided fairly.

This is proven in the prediction when data is divided properly and the sample use for training and testing data is chosen carefully the result can show a good result. Such as in the parameter selection when the parameter with high error was added it produced bad accuracy and it reduced the performances of SVM to do prediction correctly. Since in SVM the term bias is introduced it is affecting the sample selection where if the data has more high class then the prediction will be bias towards high class label and so on.

The introduction of empty sample also is problem if we do not remove it because the SVM will not include it in the calculation and we will have problems with the data in the future step when doing SVM. The forward selection is used because it is easy to

recognize which parameter actually works or not because we add the parameter one by one and if it gives bad result we can eliminated it and substitute with another parameter.

The sensitivity for each class is quite high except for the low class however it is acceptable because the high and medium class sensitivity is higher. This is important because the prediction can predict the DO high class and medium class very effectively as high class of DO is clean water and medium class require treatment and low class of DO need an extra treatment for the water. The sensitivity is high because the true negative result in this prediction is high because it is not wrongly predict the data that was not supposed to be predicted in wrong class but the accuracy shows that the prediction is accurate for all classes. However the error that obtains from this prediction is quite high and it is because the value used in the kernel parameter is affecting the error result. Such as the cost C is about 24 and it shows that this model tolerates more error.

Water quality since it is not easy to be predicted is same with these data because several factors which arise in these lakes such as climate and the development that happen around them. Such as in 2008 where the DO levels drop mainly from Lake Chini is because some factor of development such as the land activities and from the river flow that drained into Lake Chini especially during wet season. Prediction on water quality also expected to falls below predicted model because of random error cause by nature factor.

6. CONCLUSION

Water quality monitoring is hard to be forecast. However with the introduction of artificial intelligence techniques it helps in predicting the water quality successfully. SVM is nowadays recognized to do the forecast of this complex data. In this project prediction made by the SVM for the Lake Bera and Lake Chini produces accuracy about 74.07%.It accurately predicts the class of DO. This prediction can be used to predict new data for future data. Besides in ecology the precision is not as important as the range because we want to know whether the water quality is acceptable to be use or not. In the objective mentioned in chapter 1 show that these two lakes are in safe condition as based on the standard given for DO quality. All objective produces for this project is achieved.

INSTRUMENTATION

R coding use to do the SVM

```
>library(kernlab)

>### read data

>mydata<- read.csv('NApo.csv')

>summary(mydata)
```

In R we need to declare the package that we want to use by using the library fuction.

To import data into R we used function read.csv to read the data that we had saved in csv format. Summary function will gives information of the data such as mean, min, mod and other information.

```
### determine the pH selection

>pH<- mydata[1:120, 2,]

>ytrain<-mydata[1:120, 1:1]

>pH<-cbind(pH,pH)

>bestmodel<-ksvm (pH,ytrain,type="C-svc",  kernel='vanilladot', cross=10)

>bestmodel
```

This part is where the run the linear kernel to see the error produce by each parameter to determine the suitability of the parameter to be use as the input. In the first line we declare the parameter column and row of testing data and then in the second line declare the labels of classification for testing data. Then we bind the parameter with itself before the linear SVM is run. Other parameter is done the same way and we need to define the row and the column of each parameter.

```
>library(kernlab)

 ### read data

>mydata<- read.csv('NApo.csv')

>summary(mydata)
```

In the csv file we had determine which parameter need to be combined together by doing forward selection. Then data is imported in to R. Actually based on the parameter selection made before we can just use the cbind function to bind the best parameter together and run for the kernel test.

```
### set data to be training and test

>xtrain<- mydata[1:120, 2:4]

>ytrain<- mydata[1:120, 1:1]

>xtest<- mydata[121:147, 2:4]

>ytest<- mydata[121:147, 1:1]

### label for train and test data

>ytrain<- factor(ytrain, levels = c ("High","Medium","Low"),labels=c("H", "M", "L"))

>ytest<- factor(ytest, levels = c ("High","Medium","Low"),labels=c("H", "M", "L"))
```

In the first until fourth line train data for the input is set from data 1 until data 120 which start from column 2 while testing data is starting after training data. The label is same as training and testing data but it start from column 1. Next in fifth line we labeled the classification of DO as H for class high, M for class Medium and L for low class.

```
>table(ytrain)

>table(ytest)
```

Table function is used to see the value of each class in train and test data.

```
###Kernel selection (Each kernel is set with default value)

>model<-ksvm (ytrain~.,data=xtrain,type="C-svc",  kernel='vanilladot',cross=10)

>model2<-ksvm (ytrain~.,data=xtrain,type="C-svc",  kernel='rbf',cross=10)

>model3<-ksvm (ytrain~.,data=xtrain,type="C-svc",  kernel='polydot',cross=10)

>model4<-ksvm (ytrain~.,data=xtrain,type="C-svc",  kernel='laplacedot', cross=10)

>model5<-ksvm (ytrain~.,data=xtrain,type="C-svc",  kernel='tanhdot', cross=10)

>model6<-ksvm (ytrain~.,data=xtrain,type="C-svc",  kernel='besseldot', cross=10)

>model7<-ksvm (ytrain~.,data=xtrain,type="C-svc",  kernel='anovadot', cross=10)
```

Based on the kernel used we assign it with different name such as model, model2 and as follow. Each is done using the default value which is usually the value is 1.

```
>model

>model2

>model3

>model4

>model5

>model6

>model7
```

By calling the assign name we can see the error produce for each parameter and the best kernel is chosen when it resulted with least error.

```
###loop function to determined best kernel parameter value

>clist<- c(1:20)

>clist<- clist /10

>sigmalist<- clist



>cv_err<- matrix(0,nrow=length(clist),ncol=length (sigmalist))

+for (i in seq(length(clist))) {

+C <- clist[i]

+for (j in seq(length(sigmalist))) {

+model<-ksvm(ytrain~.,data=xtrain,type="C-svc",          kernel='anovadot',
+kpar=list(sigma=sigmalist[j],degree=1),  C=clist[i],cross=10)

+cv_err[i,j] <- cross(model)

+}

+}

>cv_err
```

After selecting the kernel we need to find the best parameter value as this can improve our result. By doing the SVM model using default value we still do not know whether the value is the best value and if it is the best combination that can gives us least error. For the best kernel use in this project is Anova kernel. In Anova the parameter used are sigma, C and degree. We use default value for degree value firstly. In the first until third line we assign the value to be used in finding the best sigma and C value. Next in fourth

line we assign which from sigma and C value to be in row and which to in the column line. We used for loop to find the best generated sigma and C value and input it in the model that has been chosen. In the tenth line from the row and column of sigma and C value we assign it to display the error produce from the combination of both parameter values.

```
>bestmodel<-ksvm (ytrain~.,data=xtrain,type="C-svc", kernel='anovadot', kpar=list(sigma=1,degree=1.5), C=24,cross=10)

>bestmodel
```

From the loop function we choose the best parameter value with least error to be input into our model.

```
>pred.ksvm<- predict(bestmodel, xtest)

>pred.ksvm

>ytest
```

The predict function is used to predict our test data with the model that has been created before. When calls the assigned name for predict function it will show the prediction made on the test data. ytest is called to see the comparison between the predicted and actual data class label.

```
>table(pred.ksvm, ytest)

>sum(pred.ksvm==ytest)/length(ytest)
```

The table function is used to generate the cross tabulation table of actual and predicted data. Then to calculate the accuracy we use the sum function.

APPENDIX

Appendix A

SVM on R Tutorial

Some facts about SVM on R:

- R is light – it doesn't come with all functions.
- Specialised functions come in packages – you only add the packages that you need.
- SVM functions are bundled in the "kernlab" package.
- Thus, prior to run SVM functions, we need to download the 'kernlab" package on our machine.

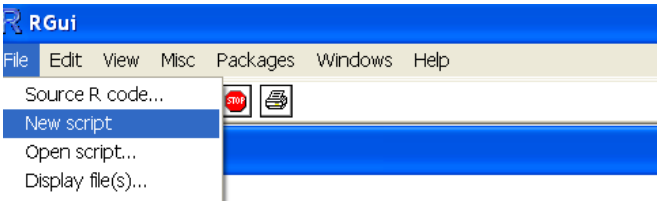
(1) In order to download "kernlab" package on our machine, type the following in the terminal:

install.packages("kernlab")

You would then need to choose the mirror. Once the installation is completed, you would be prompted accordingly.

(2) In this tutorial, we shall write all our R codes in scripts. Do the following:

File -> New Scripts



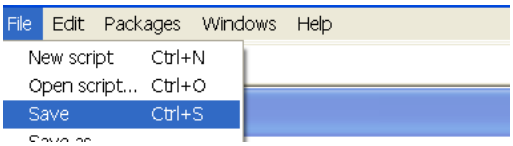
A new window of R Editor will be opened.

(3) Type the following codes in the R editor:

```
#This is my 1st attempt to run SVM on R

#attaching the kernlab package

library(kernlab)
```

(4) Save the file. Do the following:

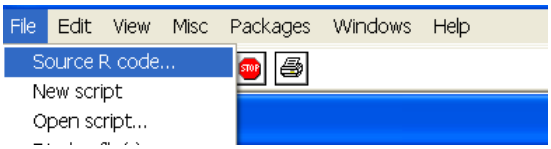
File -> Save



Save the file as ICIMU.R

(5) Invoke the file. Do the following:

File ->Source R code…



Chose the file ICIMU.R

Our tutorial:

- In this tutorial, we shall be doing things step by step.
- It is essential that every step is saved and compiled accordingly.
- The comments help you to understand each given step.

(6) Go back to your ICIMU.R file. Append the following codes in the R editor:

```
#These codes create two classes of 50 specimens each,
# namelyblueClass and redClass.
#Each class has two features, namely f1 and f2
#Both f1 and f2 are of normal distribution
#The blue class has a mean of 1 and sd of 0.2
#for its f1 and f2.
#The red class has a mean of 1.5 and sd of 0.2
#for its f1 and f2.


n <- 50


f1blueclass <- rnorm(n, mean = 1, sd = 0.2)
f2blueclass <- rnorm(n, mean = 1, sd = 0.2)
f1redclass <- rnorm(n, mean = 1.5, sd = 0.2)
```

```
f2redclass <- rnorm(n, mean = 1.5, sd = 0.2)


blueclass<- cbind(f1blueclass, f2blueclass)

redclass<- cbind(f1redclass, f2redclass)
```

Do not forget to save the file.

(7) Invoke the file ICIMU.R again. (Hint: Use the arrow key)

Type the following in the terminal:

```
>blueclass

>redclass
```

(8) Let us visualize the data. Type the following in the terminal:

```
>plot(blueclass, col = "red")

>plot(redclass, col = "blue")
```

Some facts on rnorm(n, mean, sd) function:

- This function creates a random n samples from a distribution with the given mean and standard deviation
- The values are different from one person to another
- The values are different from one run to another

(9) Go back to your ICIMU.R file. Append the following codes in the R editor:

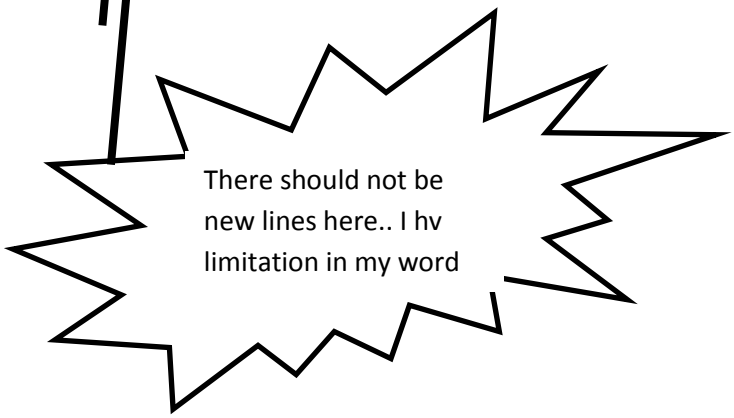
```
#Prepare data for SVM

#Data – regardless the number of features is often known as x

x <- rbind(blueclass, redclass)



#Generate the labels for the classes

#Labels are often known as y

#For blue class, we assign the value 1

#For red class, we assign the value -1

y<- matrix(c(rep(1,50),rep(-1,50)))
```

Do not forget to save the file.

(10) Invoke the file ICIMU.R again. (Hint: Use the arrow key)

Type the following in the terminal:

```
>x
>y
>plot(x,col=ifelse(y>0,"blue","red"), xlab = "feature2",
ylab="feature1")
>legend("topleft",c('Blue Class','Red
Class'),col=c("blue","red"),pch=1,text.col=c("blue","red"))
```



(11) Go back to your ICIMU.R file. Append the following codes in the R editor:

```
# Prepare a training and a test set randomly from the data set#


# ntrain is the total number of training samples
# Usually 80% of data is used for training
# whilst 20% of data is used for testing
ntrain<- round(n*0.8)


# tindex lists the indices of training samples (randomly chosen)
```

```
tindexblue<- sample(n,ntrain)
tindexred<- 50 + sample(n,ntrain)
tindex<- c(tindexblue, tindexred)
xtrain<- x[tindex,]
xtest<- x[-tindex,]
ytrain<- y[tindex]
ytest<- y[-tindex]
istrain=rep(0,2*n)
istrain[tindex]=1
```

Do not forget to save the file.

(12) Invoke the file ICIMU.R again. (Hint: Use the arrow key)

Type the following in the terminal:

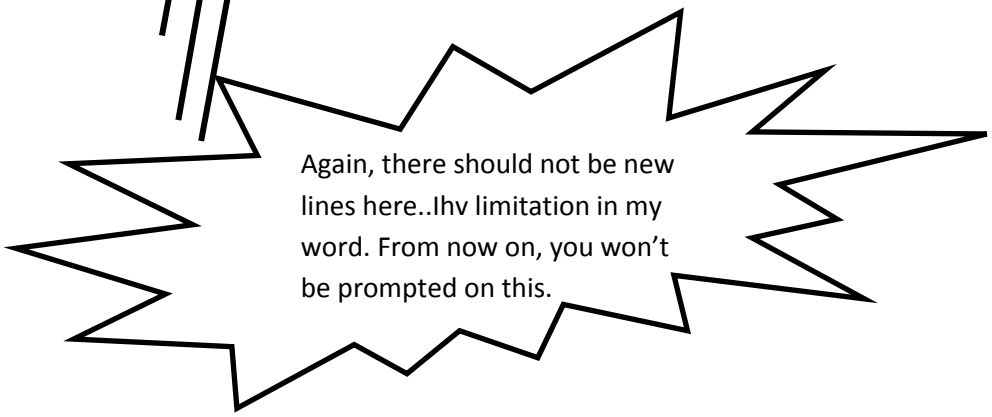
```
>tindexblue
>tindexred
>tindex
>xtrain
>xtest
>ytrain
>ytest
>istrain
```

Some facts on sample(n, x) function:

- This function selects random x numbers from a 0 to n integer list
- x < n
- There should not be duplicate values of x

(13) Let us visualize the data. Type the following in the terminal:

```
>plot(x,col=ifelse(y>0, "blue", "red"),pch=ifelse(istrain==1,1,2),
xlab = "feature2", ylab="feature1")
>legend("topleft",c('Blue Class Train','Blue Class Test','Red Class
Train','Red Class Test'),col=c("blue", "blue", "red",
"red"),pch=c(1,2,1,2),text.col=c("blue", "blue", "red", "red"))
```



(14) Now we are ready to run the SVM. Let us first build the SVM model using linear kernel on out training data. Go back to your ICIMU.R file. Append the following codes in the R editor:

```
# Train the SVM
# Assign our model to an object known as svm_model
svm_model<- ksvm(xtrain,ytrain,type="C-
svc",kernel='vanilladot',C=1,scaled=c())
```

Do not forget to save the file.

(15) Let us check our model. Invoke the file ICIMU.R again. (Hint: Use the arrow key)

Type the following in the terminal:

```
>svm_model
```

Some facts on ksvm(x, y ,...) function:

- This function builds the specified SVM model on our trained data x with label of classes y.
- In order to get more details on any function in R, just type help(function_name) on the terminal to go to the help page
- Kernel vanniladot is the linear kernel.
- By default, the C parameter is set to 1 (unless you set it differently).

- When you type the model name, you will be prompted on several information. The most important information is the training error rate.

(16) Let us now test the accuracy of our SVM model on the testing data. Go back to your ICIMU.R file. Append the following codes in the R editor:

```
# Predict labels on the testing data
ypred = predict(svm_model,xtest)


# Compute the model accuracy
# Our model is accurate if the predicted label = the actual label
# Otherwise, we have error
accuracy<- sum(ypred==ytest)/length(ytest)


# Compute at the prediction scores
# If the score is > 0, it belongs to class blue
# ifthescore is < 0, it belongs to class red
ypredscore = predict(svm_model,xtest,type="decision")
```

Do not forget to save the file.

(17) Invoke the file ICIMU.R again. (Hint: Use the arrow key)

Type the following in the terminal:

```
>ypredscore

>ypred

>ytest

>accuracy

>table(ypred, ytest)
```

(18) Let us repeat the above steps (training and testing) with different C parameters for our SVM models. Go back to your ICIMU.R file. Append the following codes in the R editor:

```
# Train the SVM with C = 20 & C = 30

svm_model2<- ksvm(xtrain,ytrain,type="C-

svc",kernel='vanilladot',C=20,scaled=c())

svm_model3<- ksvm(xtrain,ytrain,type="C-

svc",kernel='vanilladot',C=30,scaled=c())


# Predict labels (for model with C = 20 & C=30)

ypred2 = predict(svm_model2,xtest)

ypred3 = predict(svm_model3,xtest)


# Compute the model accuracy (for model with C = 20&C = 30)

accuracy2<- sum(ypred2==ytest)/length(ytest)

accuracy3<- sum(ypred3==ytest)/length(ytest)
```

Do not forget to save the file.

(19) Invoke the file ICIMU.R again. (Hint: Use the arrow key)

Type the following in the terminal:

```
>accuracy

> accuracy2

> accuracy3

>plot(svm_model,data=xtrain, xlab = "feature2", ylab="feature1")

>plot(svm_model2,data=xtrain, xlab = "feature2",

ylab="feature1")

>plot(svm_model3,data=xtrain, xlab = "feature2",

ylab="feature1")
```

You may re-run the experiment several times until you get different values of accuracy.

What we have done thus far:

- Basically we have divide our data (manually) into training and testing set.
- We built the model on the training data
- We investigate the accuracy by experimenting on the testing data.
- In research, data collection is expensive. Researches are often faced with the problems of data limitation.
- Cross validation is used when we have limited data.

- There is a build-in function for cross validation in R – simple, save us the trouble of dividing the data into separate training and testing set.

(20) Let us try the cross validation approach. Go back to your ICIMU.R file.

Append the following codes in the R editor:

```
# We perform cross validation on 5 folds data
svm_cv<- ksvm(x,y,type="C-
svc",kernel='vanilladot',C=1,scaled=c(),cross=5)
```

(21) Let us check the cross validation error rate.

Invoke the file ICIMU.R again. (Hint: Use the arrow key)

Type the following in the terminal:

```
>svm_cv
>cross(svm_cv)
```

Some facts aboutscross(svm_cv):

- It invokes the cross validation error rate from the object svm_cv.
- Training error rate is not as important as the cross validation error rate (i.e. there is no point getting a low training rate if it performs badly during cross validation testing)

(22) Tricks: We shall create a function that performs SVM cross validation in a loop for different parameter of C. Go back to your ICIMU.R file. Append the following codes in the R editor:

```
#Create a function that performs cross validation SVM repeatedly
#on a list of C parameters


clist<- c(1:20) #create an C list of integer from 1 to 20


#divide c list by 10
#thus our c parameter varry from 0.1 to 2.0
clist<- clist / 10


#create a list to hold the cross validation error
err<- numeric(length(clist))


#perform SVM in a loop
for (i in seq(length(clist))) {
        svm_cv_loop<- ksvm(x,y,type="C-
        svc",kernel='vanilladot',C=clist[i],scaled=c(),cross=5)
        err[i] <- cross(svm_cv_loop)
}
```

(23) Let us check the error rate performance.

Invoke the file ICIMU.R again. (Hint: Use the arrow key)

Type the following in the terminal:

```
>plot(clist,err,type='l',xlab="C",ylab="Error rate")
>cbind(clist, err)
```

What have given you thus far:

- Is the step by step exact codes (you just need to compile and run the codes)
- For the following session, you are required to add, change and modify the code yourselves.
- Good luck

(24) You have so have only tested the linear kernel. From now onwards you shall run your codes on non-linear kernel. Google package kernlab. What are other non-linear kernels available? Go back to your ICIMU.R file. Modify svm_model2 and svm_model3 using different kernels. Run. Compare the accuracy by using different kernels.

(Hint :- You may want to increase the overlapped of your data by increasing the standard deviation)

(25) You have so have only tested the linear kernel. From now onwards you shall run your codes on non-linear kernel. Google package kernlab. What are other non-linear kernels available? Go back to your ICIMU.R file. Modify svm_model2 and svm_model3 using different kernels. Run. Compare the accuracy by using different kernels.

(26) Try to make a nested loop for SVM RBF kernel. The outter loop should run with different value of C. Where as the inner loop should run with different value of sigma. You may want to view the help pages or the pdf documentation of kernlab for further elaboration on RBF kernel.

Appendix B

Data used for prediction

DO	pH	TEMP	COND
Medium	6.99	30.88	77
Medium	6.73	30.91	89
Medium	6.19	28.9	46
Medium	6.48	32.84	48
Medium	7.33	32.78	88
Medium	6.85	30.47	154
Medium	6.83	32.88	57
Medium	6.92	31.63	33
Medium	6.78	30.97	36
Medium	7.11	32.36	32
Medium	7.23	31.81	37
Medium	7.24	31.03	35
Medium	6.75	30.86	52
Medium	6.73	30.97	36
Medium	6.53	31.65	28
Medium	7.73	32.09	92
Medium	7.68	32.52	140
Medium	7.13	28.79	94
Medium	6.38	33.11	190
Medium	6.79	30.51	28
Medium	6.18	30.35	29
Medium	6.24	30.51	28
Medium	6.79	30.75	27
Medium	7.21	30.31	29
Medium	7.11	30.72	30
Medium	6.57	30.91	38
Medium	6.84	30.63	28
Medium	6.22	30.34	26
Medium	7.22	31.7	27
Medium	6.8	31.14	27
Medium	6.95	31.05	27
Medium	7.31	32.75	28
Medium	7.3	31.95	32
Medium	7.56	31.78	31
Medium	6.76	31.89	34
Medium	6.59	31.09	29
Medium	6.51	31.11	26
Medium	6.73	27.64	109
Medium	6.06	27.86	71
Medium	6.41	30.61	27
Low	7.64	32.96	139
Low	6.55	29.07	39
Low	6.86	30.48	93
Low	6.13	27.55	38
Low	6.51	31.17	28
Low	6.75	28.77	99
Low	6.25	27.67	72
Low	6.33	29.76	30
Low	6.41	27.59	32
Low	6.24	27.7	27
Low	5.27	27.67	24
Low	6.46	29.87	25
Low	7.33	28.19	90
Low	5.5	29.21	31
Low	5.61	30.2	31
Low	7.65	27.34	34
Low	7.6	29.99	40
Low	7.02	28.32	23
Low	6	30.11	33
Low	5.54	26.51	34
Low	5.59	30.04	35
Low	5.96	28.43	22
Low	5.84	28.62	31
Low	6.1	28.06	32
Low	6.01	27.39	49
Low	5.96	30.61	52
Low	6.09	28.02	39
Low	6.16	28.84	31
Low	6.71	29.32	36

Low	6.1	28.02	39
Low	6.67	28.4	44
Low	6.11	28.8	40
Low	5.5	27.09	36
Low	5.91	28.9	23
Low	5.84	27.87	29
Low	5.92	33.04	32
Low	6.32	29.32	33
Low	6.24	29.49	29
Low	6.51	29.34	33
Low	6.35	29.2	29
High	7.42	33.62	47
High	6.63	30.31	30
High	6.93	31.37	25
High	7.5	31.32	33
High	6.54	30.78	29
High	7.33	31.82	27
High	7.06	30.8	26
High	7.8	31.74	31
High	7.92	31.75	34
High	7.94	31.71	37
High	7.6	30.49	35
High	7.06	30.8	26
High	6.11	27.35	31
High	6.43	29	27
High	6.57	32.1	25
High	7.53	31.32	27
High	6.39	28.93	32
High	6.32	32.69	34
High	6.12	27.61	38
High	5.11	28.5	25
High	4.97	28.57	26
High	5.14	27.16	26
High	5.53	29.8	27
High	5.32	28.18	27
High	6.4	30.91	32
High	6.62	30.91	36
High	6.51	30.57	14
High	6.58	29.44	32

High	6.57	34.29	28
High	6.66	28.87	26
High	5.69	32.35	28
High	5.91	32.84	24
High	7.5	27.57	101
High	5.97	30.11	21
High	6.5	32.72	22
High	6.55	30.75	25
High	6.82	30.69	22
High	7.41	30.75	25
High	7.16	30.77	27
High	6.76	30.34	26
Medium	6.18	30.13	28
Medium	6.45	31.17	28
Medium	6.34	30.97	29
Medium	6.77	31.065	28
Medium	6.24	30.67	32
Medium	6.3	30.6	29
Medium	6.07	29.93	28
Medium	7.59	26.12	30
Medium	7.77	27.7	31
Low	5.79	28.46	26
Low	5.81	28.12	22
Low	6.41	29.88	22
Low	7.03	29.67	78
Low	6.66	28.37	33
Low	6.24	32.64	37
Low	6.15	30.95	25
Low	5.98	32.62	26
Low	6.9	29.82	69
High	6.67	30.72	26
High	6.68	30.47	25
High	7.01	33.47	23
High	7.26	30.71	125
High	6.64	30.91	43
High	6.71	30.28	26
High	6.42	32.86	27
High	7.89	31.54	21
High	7.42	32.54	23

Appendix C

Data eliminated for prediction

NH3-NL	TEMP	COND	SAL	TUR	NO3	PO4	E-coli	Coliform
0.03	30.88	77	0.03	23.7	0.04	0.01	100	20000
0.05	30.91	89	0.04	32	0.22	0.01	400	8700
0.09	28.9	46	0.02	29.2	0.05	0.01	2800	11000
0.03	32.84	48	0.02	15.1	0.01	0.01	140	7400
0.01	32.78	88	0.04	37.3	0.21	0.03	300	13000
0.01	30.47	154	0.07	56.2	1.06	0.13	100	3300
0.03	32.88	57	0.02	8.2	0.16	0.01	200	2800
0.01	31.63	33	0.01	4.7	0.17	0.04	80	1800
0.01	30.97	36	0.02	4.8	0.01	0.01	5400	11000
0.1	32.36	32	0.01	8.8	0.01	0.01	3000	10000
0.02	31.81	37	0.02	7.3	0.02	0.01	1800	7800
0.45	31.03	35	0.02	7.6	0.01	0.01	2100	6400
0.37	30.86	52	0.02	1.1	0.01	0.01	1800	8100
0.75	30.97	36	0.02	2.6	0.01	0.01	3000	7800
0.14	31.65	28	0.01	1.6	0.01	0.01	3000	6400
0.01	32.09	92	0.05	25.2	0.33	0.09	500	4500
0.05	32.52	140	0.06	27.2	0.47	0.07	300	4300
0.08	28.79	94	0.04	16.8	0.18	0.02	1600	5200
0.37	33.11	190	0.09	11.4	0.22	0.01	1300	1800
0.01	30.51	28	0.01	6.5	0.01	0.02	200	1300
0.01	30.35	29	0.01	6.5	0.05	0.23	500	1400
0.01	30.51	28	0.01	1.8	0.01	0.2	600	1900
0.01	30.75	27	0.01	11.3	0.01	0.3	100	1500
0.01	30.31	29	0.01	5.3	0.85	0.29	400	1700
0.01	30.72	30	0.01	5.9	0.04	0.23	300	2500
0.01	30.91	38	0.02	2.1	0.13	0.18	400	1000
0.01	30.63	28	0.01	7	0.01	0.19	300	1600
0.01	30.34	26	0.01	1	0.01	0.3	900	2400
0.01	31.7	27	0.01	6.4	0.01	0.07	70	170
0.2	31.14	27	0.01	9.1	0.46	0.01	20	110
0.26	31.05	27	0.01	3.8	0.75	0.01	60	310
0.5	32.75	28	0.01	8.3	0.53	0.01	60	230
0.62	31.95	32	0.01	8.7	0.78	0.12	60	180

0.33	31.78	31	0.01	11.4	0.47	0.03	0	20
0.04	31.89	34	0.01	1.5	0.29	0.01	120	340
0.08	31.09	29	0.01	2.2	0.15	0.01	30	130
0.08	31.11	26	0.01	3.1	0.11	0.01	40	410
0.02	27.64	109	0.05	129	0.8	0.08	2500	8400
0.01	27.86	71	0.03	13.2	0.03	0.07	3000	8500
0.5	30.61	27	0.01	3.5	0.01	0.04	400	2800
0.01	32.96	139	0.06	75.1	0.44	0.02	300	7800
0.01	29.07	39	0.02	39.8	0.07	0.01	400	14000
0.07	30.48	93	0.04	145	1.08	0.14	300	6000
0.01	27.55	38	0.02	28.9	0.25	0.01	200	2100
0.52	31.17	28	0.01	1.1	0.01	0.01	2100	7300
0.01	28.77	99	0.05	26.4	0.28	0.06	1300	8400
0.01	27.67	72	0.03	14.4	0.13	0.07	1600	5100
0.01	29.76	30	0.02	2	0.18	0.18	10	990
0.34	27.59	32	0.02	14.6	0.11	0.36	1100	3100
0.04	27.7	27	0.01	14.1	0.62	0.01	0	1000
0.03	27.67	24	0.01	0.8	0.71	0.01	100	2300
0.04	29.87	25	0.02	37	0.03	0.09	700	20800
0.01	28.19	90	0.04	125.5	0.76	0.05	100	5600
0.03	29.21	31	0.02	18.6	0.02	0.01	0	1800
0.01	30.2	31	0.03	2.2	0.31	0.01	0	4400
0.03	27.34	34	0.01	6.6	0.77	0.01	1300	10900
0.01	29.99	40	0.02	9.6	0.01	0.01	1000	14000
0.05	28.32	23	0.01	13	0.01	0.01	200	13000
0.81	30.11	33	0.01	8	0.31	0.01	100	4000
0.04	26.51	34	0.01	30	0.01	0.1	800	21000
0.01	30.04	35	0.01	30	0.01	0.1	2000	13000
0.01	28.43	22	0.01	7	0.01	0	100	7000
0.03	28.62	31	0.01	3	0.01	0	1000	13000
0.01	28.06	32	0.01	22.4	0.01	0.1	200	8100
0.01	27.39	49	0.02	33.7	0.01	0.05	2000	8000
0.01	30.61	52	0.02	36	0.01	0.07	4000	33000
0.01	28.02	39	0.02	29.1	0.01	0.05	600	2800
0.06	28.84	31	0.01	12.1	0.58	0.01	1100	7400
0.18	29.32	36	0.02	4.5	0.35	0.01	1200	7200
0.22	28.02	39	0.02	29.1	0.13	0.01	300	6100
0.5	28.4	44	0.02	27.8	1.06	0.06	200	800
0.23	28.8	40	0.02	2	0.75	0.01	1100	5500

0.23	27.09	36	0.02	5	0.93	0.01	900	10100
0.05	28.9	23	0.01	9.7	0.01	0.06	300	8000
0.01	27.87	29	0.01	17.4	0.01	0.04	400	8000
0.01	33.04	32	0.01	16.5	0.01	0.04	600	5000
0.01	29.32	33	0.01	8.9	0.02	0.05	200	3200
0.1	29.49	29	0.01	3.4	0.09	0.24	2000	4700
0.09	29.34	33	0.01	8.4	0.09	0.24	100	900
0.07	29.2	29	0.01	2.6	0.09	0.21	100	2100
0.01	33.62	47	0.02	10.9	0.01	0.01	100	2100
0.02	30.31	30	0.01	4	0.21	0.01	100	2000
0.01	31.37	25	0.02	1.7	0.01	0.01	300	1100
0.01	31.32	33	0.02	4.2	0.02	0.01	200	1200
0.01	30.78	29	0.02	0.8	0.02	0.01	200	1200
0.01	31.82	27	0.02	3.3	0.01	0.01	300	6600
0.01	30.8	26	0.02	0.4	0.01	0.01	500	1300
0.01	31.74	31	0.02	9.8	0.01	0.01	100	900
0.01	31.75	34	0.02	8.6	0.01	0.01	1100	2700
0.01	31.71	37	0.02	9.7	0.01	0.01	300	3400
0.01	30.49	35	0.02	1	0.01	0.01	100	31700
0.01	30.8	26	0.02	0.6	0.04	0.01	1200	2400
0.01	27.35	31	0.02	15.6	0.11	0.01	500	47100
0.01	29	27	0.02	1.1	0.01	0.07	2100	3300
0.01	32.1	25	0.02	4.2	0.02	0.08	100	2700
0.06	31.32	27	0.02	5.2	0.05	0.09	300	1200
0.01	28.93	32	0.02	1.8	0.01	0.08	100	1600
0.01	32.69	34	0.01	10.2	0.96	0.21	40	1560
0.03	27.61	38	0.02	16.2	0.11	0.01	1400	6400
0.01	28.5	25	0.01	1.5	0.09	0.01	1000	1900
0.01	28.57	26	0.01	3.1	0.04	0.01	1100	2200
0.01	27.16	26	0.01	0.7	0.31	0.01	50	1800
0.01	29.8	27	0.02	2.4	0.18	0.01	0	8100
0.01	28.18	27	0.02	2.7	0.53	0.01	200	5900
0.3	30.91	32	0.01	3.3	1.37	0.03	1000	13000
0.4	30.91	36	0.01	4	2.04	0.01	100	8000
0.13	30.57	14	0.02	5.2	0.66	0.01	100	9000
0.44	29.44	32	0.01	4.1	1.68	0.03	2000	18000
0.2	34.29	28	0.01	19	0.25	0.01	1000	31000
0.02	28.87	26	0.01	4	0.01	0.01	30	900
0.01	32.35	28	0.01	8.3	0.57	0.01	2000	6000

0.1	32.84	24	0.01	8.5	0.72	0.06	200	13800
0.21	27.57	101	0.05	36.5	0.24	0.05	200	31000
0.01	30.11	21	0.01	0.5	0.04	0.04	80	13300
0.01	32.72	22	0.01	8.7	0.24	0.01	2900	13900
0.01	30.75	25	0.01	5.7	0.03	0.09	164	12400
0.01	30.69	22	0.01	1.2	0.01	0.26	1328	24400
0.01	30.75	25	0.01	1.6	0.02	0.14	192	10900
0.01	30.77	27	0.01	2.5	0.01	0.22	384	15500
0.01	30.34	26	0.01	3	0.02	0.23	144	11300
0.01	30.13	28	0.02	1.3	0.01	0.12	500	1600
0.01	31.17	28	0.02	4	0.09	0.15	200	1500
0.01	30.97	29	0.02	6.4	0.71	0.03	20	1480
0.01	31.065	28	0.02	5.3	0.09	0.03	300	2500
0.01	30.67	32	0.02	1.1	0.18	0.12	300	3300
0.01	30.6	29	0.02	2.6	0.09	0.18	10	1180
0.01	29.93	28	0.02	1.5	0.75	0.21	500	3000
0.06	26.12	30	0.01	238	0.44	0.01	5200	16300
0.24	27.7	31	0.01	282.2	0.56	0.01	600	19100
0.06	28.46	26	0.01	1.1	0.09	0.31	1000	9000
0.01	28.12	22	0.01	74.2	0.37	0.02	1000	8600
0.02	29.88	22	0.01	16.6	0.02	0.01	2000	11000
0.01	29.67	78	0.03	30.8	0.01	0.02	100	7000
0.01	28.37	33	0.01	25.9	0.01	0.01	100	5000
0.01	32.64	37	0.02	10.4	0.01	0.01	16	700
0.01	30.95	25	0.01	9.2	0.01	0.07	100	900
0.01	32.62	26	0.01	1.8	0.01	0.04	100	3800
0.01	29.82	69	0.03	74.9	0.01	0.02	100	10000
0.01	30.72	26	0.01	1.1	0.01	0.2	160	3500
0.01	30.47	25	0.01	1.3	0.01	0.19	120	9400
0.01	33.47	23	0.01	4	0.04	0.01	1400	15100
0.05	30.71	125	0.06	15.7	0.01	0.05	100	284000
0.04	30.91	43	0.02	11	0.04	0.04	200	4600
0.04	30.28	26	0.01	6.9	0.01	0.03	400	17400
0.03	32.86	27	0.01	3.8	0.72	0.01	200	4700
0.21	31.54	21	0.01	12.2	0.01	0.01	1040	1300
0.08	32.54	23	0.01	11.7	0.01	0.01	800	1800

BIBLIOGRAPHY

1. Bouamar M., and Ladjal M., "Evaluation of the performances of ANN and SVM techniques used in water quality classification, "*Proceedings of ICECS'07*, 14.

2. Bouamar M., Ladjal M, Multisensor system using Support Vector Machines for water quality classification, *Proceedings of ISSPA'07, 9th IEEE International Symposium on Signal Processing and its Applications*, 12-15 Feb, Sharjah, UAE, 2007.

3. *DTREG*. (n.d.). Retrieved May 5, 2012 , from DTREG (Software For Predictive Modeling and Forecasting): http://www.dtreg.com.

4. He , T., & Chen, P. (2010). Prediction of water-quality based on wavelet transform using vector machine. *2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science.* 76 – 81.

5. Interim National Water Quality Standard in Malaysia.(n.d). Retrieved March 26, 2012, from WEPA (Water Environment Partnership in Asia): http://www.wepa-db.net.

6. Junsawang P, Pomsathit A, Areerachakul. Prediction of Dissolved Oxygen Using Artificial Neural Network. *2011 International Conference on Computer Communication and Management,Singapore,* 1-5.

7. Karatzoglou A.,Smola A., Hornik K. (2004). kernlab – An S4 Package for Kernel Methods in R. 11(9). 1-20.

8. Liao Y., Xu J., Wang Z. (2012). Application of biomonitoring and support vector machine in water quality assessment. 13(4): 327–334.

9. Liu JP, Chang MQ, Ma XY. Groundwater Quality Assessment Based on Support Vector Machine. HAIHE River Basin Research and Planning Approach-*Proceedings of 2009 International Symposium of HAIHE Basin Integrated Water and Environment Management*, Beijing, China. 2009, 173-178.

10. Medina, M. (2011). *Development and implementation of a water quality monitoring plan to support the creation of a geographic information system to assess water quality conditions of rivers in the state of veracruz in mexico.* (Master's thesis).

11. Naik, V. K. and Manjapp, S. (2010). Prediction of Dissolved Oxygen through Mathematical Modeling. Int. J. Environ. Res., 4(1), 153-160.

12. Najah, A., El-Shafie, A., Karim, O. A., and Jaafar, O. (2011)Integrated versus isolated scenario for prediction dissolved oxygen at progression of water quality monitoring stations, Hydrol. Earth Syst. Sci., 15, 2693–2708.

13. Najah, A., El-Shafie, A., Karim, O. A., Jaafar, O. El Shafie, H.O (2011).An application of different artificial intelligences techniques for water quality prediction.International Journal of the Physical Sciences, 6(22), 5298–5308.

14. Palani, S., Liong, S.Y., Tkalich, V., Palanichamy, J., (2008). Development of Neural Network Model for Dissolved Oxygen in Seawater. Indian Journal of Marine Sciences, 38(2), 151-159.

15. Prasad, M., W. Long, X. Zhang, R. Wood, and R. Murtugudde. 2011. Predicting dissolved oxygen in the Chesapeake Bay: Applications and implications. Aquatic Sciences—Research Across Boundaries: 1–15.