Ocean Engineering 36 (2009) 339–347

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Prediction of significant wave height using regressive support vector machines

J. Mahjoobi [a,*], Ehsan Adeli Mosabbeb [b]

[a] Ministry of Energy, Water Research Institute, Hakimieh, 4th Tehranpars Square, P.O. Box 16765-313, Tehran, Iran
[b] Computer Engineering Department, Iran University of Science and Technology, Narmak, Tehran, Iran

ARTICLE INFO

Article history:
Received 23 August 2008
Accepted 3 January 2009
Available online 20 January 2009

Keywords:
Regression
Support vector machines
Artificial neural networks
Wave prediction

ABSTRACT

Wave parameters prediction is an important issue in coastal and offshore engineering. In this literature, several models and methods are introduced. In the recent years, the well-known soft computing approaches, such as artificial neural networks, fuzzy and adaptive neuro-fuzzy inference systems and etc., have been known as novel methods to form intelligent systems, these approaches has also been used to predict wave parameters, as well. It is not a long time that support vector machine (SVM) is introduced as a strong machine learning and data mining tool. In this paper, it is used to predict significant wave height (H_s). The data set used in this study comprises wave wind data gathered from deep water locations in Lake Michigan. Current wind speed (u) and those belonging up to six previous hours are given as input variables, while the significant wave height is the output parameter. The SVM results are compared with those of artificial neural networks, multi-layer perceptron (MLP) and radial basis function (RBF) models. The results show that SVM can be successfully used for prediction of H_s. Furthermore, comparisons indicate that the error statistics of SVM model marginally outperforms ANN even with much less computational time required.

1. Introduction

The effects of waves in activities related to the ocean environment such as the building and maintenance of coastal and offshore structures, maritime transportation, environmental protection and etc. caused the research on waves from different perspectives to extract the wave characteristics. Different methods such as empirical, numerical and soft computing approaches have been proposed for significant wave height prediction. Numerical models are generally based on a form of the spectral energy or action balance equation. However, due to their complexity of implementation, high amount of processor time is required, and the need for accurate local bathymetric surveys, their implementation is not an easy task (Browne et al., 2007). When the huge amount of exogenous information is not available and the computational resources and expertise are limited, data mining and machine learning approaches would be very good choices.

Empirical data modeling could be taken into use in many engineering applications. Empirical data modeling is an induction process to construct a model. This model is meant to learn and infer the behavior of the problem. The performance of this empirical model depends on quantity and quality of the data. In traditional statistical methods for problem solving, one needs to make some prior model assumptions, such as normality, linearity and homoscedasticity. But in data mining and machine learning approaches these assumptions are not required. This is also expressed in the statistical learning theory (Vapnik, 1995) in other words, which tries to minimize the empirical risk of the model built over the data. Traditional neural networks face problems with generalization.

They produce models that can overfit the data. Furthermore, approaches like neural networks need to find network parameters, such as number of hidden layer and neuron, by trial and error, which is time-consuming. Recently, artificial neural networks (ANNs) have been widely used to predict wave parameters (e.g. Makarynskyy et al., 2005; Agrawal and Deo, 2002; Makarynskyy, 2004). A review of neural network applications in ocean engineering is given in Jain and Deo (2006). They indicated in their paper that neural networks can provide a good alternative to statistical regression, time series analysis, numerical methods and approaches of this kind. The advantages are due to the improved accuracy, less complexity, smaller computational efforts and in some cases reduced data requirements.

Recently, other soft computing techniques such as fuzzy inference system (FIS) and adaptive network-based fuzzy inference system (ANFIS) have been used to develop wave

* Corresponding author. Tel.: +98 9127119496; fax: +98 2177000502.
E-mail addresses: jmahjoobi@gmail.com (J. Mahjoobi), eadeli@iust.ac.ir (E. Adeli Mosabbeb).

0029-8018/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.oceaneng.2009.01.001

prediction models (e.g. Kazeminezhad et al., 2005; Ozger and Sen, 2007). These studies have shown that the wind speed is the most important parameter in wave parameters prediction. Results of Kazeminezhad et al. (2005) show that the ANFIS results are more accurate than the results of CEM (US Army, 2003) method. Ozger and Sen (2007) predict wave parameters by using fuzzy logic approach. The model results were compared with those of classical auto regressive moving average with exogenous input (ARMAX) models. Results indicate that fuzzy model outperformed ARMAX with high levels of difference.

Mahjoobi et al. (2008) compared different soft computing methods such as artificial neural networks, fuzzy inference system and adaptive network-based fuzzy inference system to hindcast wave parameters. Their results showed that the models skills are nearly the same. Furthermore, using sensitivity analysis, they showed that wind speed and direction are the most important parameters for wave hindcasting. Another example of using data mining approaches in this field is Mahjoobi and Etemad-Shahidi (2008). They have proposed an alternative approach based on classification and regression trees for prediction of significant wave height. Results of decision trees were compared with those of artificial neural networks. The error statistics of decision trees and ANNs were nearly similar. Results indicate that the decision tree, as an efficient novel approach with an acceptable range of error, can be used successfully for prediction of H_s. It is argued that the advantage of decision trees is that, in contrast to neural networks, they represent rules.

This paper illustrates a new approach to significant wave height prediction based on support vector machines. For this purpose, support vector regression (SVR) algorithm was employed for building and evaluating a model. The results are compared with the results acquired using artificial neural networks. The improvements and great independence of external parameter settings shows the efficiency of the utilization of the proposed approach. The foundations of support vector machines (SVMs) have been developed by Vapnik (1995) and are gaining popularity due to many attractive features, and promising empirical performance. The formulation embodies the structural risk minimization (SRM) principle, which has been shown to be superior, (Gunn, 1998), to traditional empirical risk minimization (ERM) principle, employed by conventional neural networks. SRM minimizes an upper bound on the expected risk, as opposed to ERM that minimizes the error on the training data. It is this difference which equips SVM with a greater ability to generalize, which is the goal in statistical learning. SVMs were developed to solve the classification problem, but recently they have been extended to the domain of regression problems (Vapnik, 1998). In the literature the terminology for SVMs can be slightly confusing. The term SVM is typically used to describe classification with support vector methods and support vector regression is used to describe regression with support vector methods.

Support vector machines have been applied in many applications in the field of water engineering.

Mohandes et al. (2004) used SVM for wind speed prediction and compared the results with multi-layer perceptron (MLP) neural network. They indicated that SVM outperforms MLP for their purpose. Another work on soil classification (Bhattacharya and Solomatine, 2006) used three different machine learning models namely ANN, SVM and decision trees. This work acquired nearly the same results for the tree models. Asefa et al. (2006) utilized SVM for multi-time scale stream flow predictions. They have achieved better results compared to those of physical models. Yu et al. (2006) also used SVM but for real-time flood stage forecasting and obtained satisfactory results. One important point that they have noted is that the SVM model is not easily understood and interpreted. This is one of the shortcomings of SVM in comparisons to traditional ANN. Also, SVMs are used for estimation of discharge and end depth in trapezoidal channel in Pal and Goel (2007). They noted that in comparisons to back propagation neural network, both radial basis function (RBF) and polynomial kernel-based approaches work better for different datasets. In addition, they have introduced a smaller computational time for SVM comparing with ANN. Singh et al. (2008) have employed SVM for estimation of removal efficiency to settle basins in canals. In their work, SVM is again compared with ANN. Both SVM and ANN methods worked equally well. They have also indicated that the computational cost involved with SVM is significantly smaller than ANN.

Different data mining approaches, as seen before, have been used for wave prediction. SVM was used for other applications as discussed above. However, this method (SVM) has not been applied in wave prediction. SVM showed satisfactory results in other fields. As far as SVM more or less outperformed other data mining tools, both in the accuracy criteria and the computational complexity measures, it would be a great effort to utilize this modeling approach to predict significant wave height. Furthermore, regarding the ability of regressive SVM in function approximation, by utilizing SVM we can obtain wave height from wind speed. This is the rationale for using regressive support vector machines in this work.

The paper is organized as the following: Section 2 gives an explanation of data mining approaches employed in this work. Section 3 describes the study area and the data set used in this study. The proceeding section explains the acquired results together with a discussion and comparison against artificial neural networks. Finally, Section 5 summarizes and concludes the work.

2. Data mining approaches

The patterns and relationships in data can be found using machine learning, statistical analysis, and data mining techniques. The activities to discover hidden knowledge contained in data sets have been attempted by researchers in different disciplines for a long time. Through a variety of techniques, data mining identifies nuggets of information in bodies of data. Data mining extracts information in such a way that it can be used in areas such as decision support, prediction, forecast, and estimation. Data mining takes advantage of advances in the fields of artificial intelligence (AI) and statistics. Both disciplines have been applied in pattern recognition and classification.

Support vector machines are methods of supervised learning, which are commonly used for classification and regression purposes. SVM is a tool for empirical risk minimization, a special property of SVMs is that they minimize the empirical classification or regression error and maximize the geometric margin, simultaneously; this is why they are also considered as maximum margin classifiers. A SVM constructs a separating hyperplane between the classes in the n-dimensional space of the inputs. This hyperplane maximizes the margin between the two data sets of the two input classes. This is one of the most advantageous features of SVMs comparing to ANNs. The margin is defined as the distance between the two parallel hyperplanes, on each side of the separating one, pushed against each of the two datasets. Simply, the larger the margin, the better the generalization error of the classifier would be. For the case of regression, the only difference is that SVM attempts to fit a curve, with respect to the kernel used in the SVM, on the data points such that the points lie between the two marginal hyperplanes as much as possible, the aim is to minimize the regression error. The formulations and the technical note are explained in more details in Appendix A.

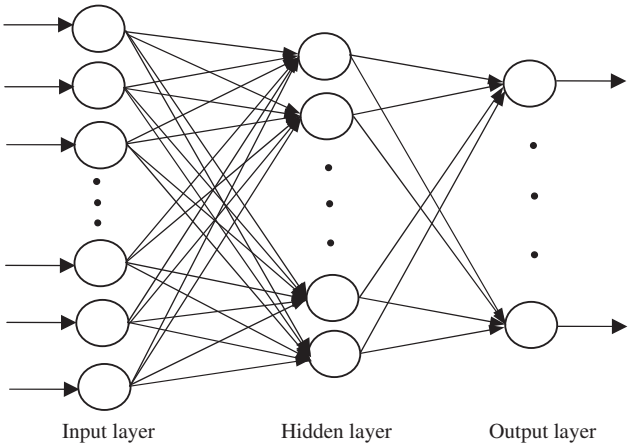

Fig. 1. The structure of a standard one-hidden-layer perceptron.

It is well-known that SVM generalization performance (estimation accuracy) depends on a good setting of meta-parameters, parameters C, ε and the kernel parameters. The choices of C and ε control the prediction (regression) model complexity. The problem of optimal parameter selection is further complicated by the fact that SVM model complexity (and hence its generalization performance) depends on all three parameters (Smola and Schölkopf, 1998). An algorithm for solving the problem of regression with support vector machines was proposed by Platt (1999) called sequential minimal optimization (SMO). It puts chunking to the extreme by iteratively selecting subsets only of size 2 and optimizing the target function with respect to them. This algorithm has much simpler background and is easier to implement. The optimization sub-problem could be analytically solved, without the need to use a quadratic optimizer. Shevade et al. (2000) had proposed an improvement, which enhanced the algorithm to perform significantly faster.

Another set of well-known data mining approaches are neural networks. There are many neural network techniques; multi-layer perceptron with back propagation (Haykin, 1999) is one of the choices for prediction problems. The multi-layered perceptron neural network (Fig. 1) consists of input, hidden, and output layers of interconnected neurons. Neurons in the one layer are combined according to a set of weights and fed to the next layer (feed-forward network). During the training phase, the data is fed to the neural network one by one, and the weights of the neurons are modified based on the error rates of the resulting outputs. Multiple passes are required over the data in order to train the network. As a result, the training times are quite large.

Another well-known neural network model is radial basis function. RBFs are typically used for function approximation, where the approximating function is represented as a sum of a number of radial basis functions, each associated with a different centers and appropriate coefficients.

3. Study area and data used

The data set used in this study comprises of wind and wave data gathered in Lake Michigan from 15 September to 10 December, 2002 and 14 September to 6 December, 2004. The data set was collected by National Data Buoy Center (NDBC) in station 45007 at 42°40′30″N and 87°01′30″W (Fig. 2), where water depth is 164.6 m. Wind and wave data were collected using 3 m discus buoy at 1 h intervals. The wind speed at buoy was measured at a height of 5 m above the mean sea level.

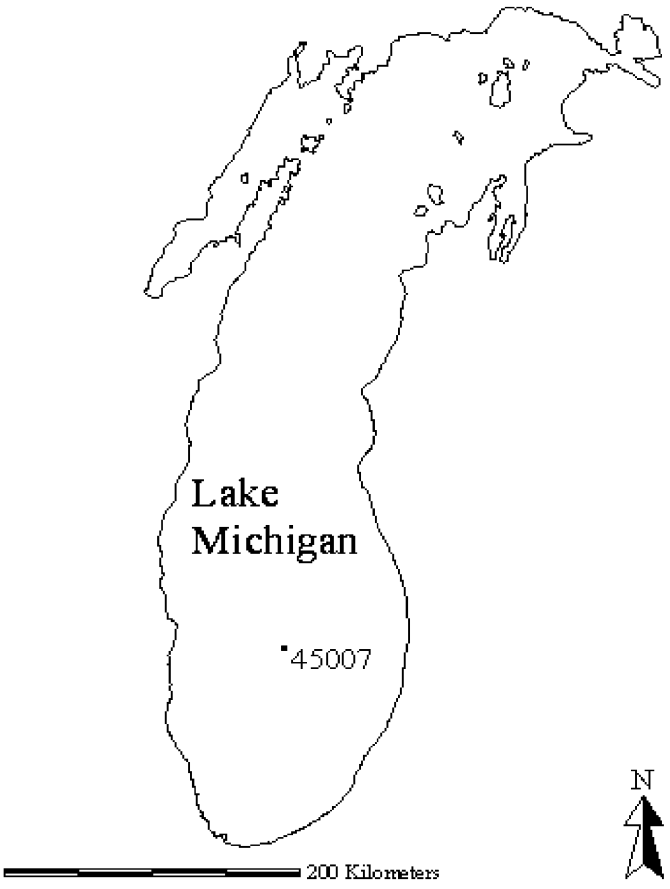

Fig. 2. Lake Michigan bathymetry and location of NDBC buoy 45007 located at 43°37′09″N and 77°24′18″W.

In this study, the data set was divided into two groups. The first one that comprises of 2086 wind and wave records (from 15 September to 10 December, 2002) was used as training data to develop the models. The second one that comprises 2007 wind and wave records (from 14 September to 6 December, 2004) was used as tasting data to verify the models.

4. Results and discussion

In this study, support vector machines are used for predicting significant wave heights. The SVM parameters used in this study are $C = 100$ and $\varepsilon = 0.001$. For the optimization process in the regression problem, improved SMO algorithm is used as discussed in Section 2. For the kernel function, two different kernel functions, radial basis function and polynomial are used. The parameters used for the kernels are $\gamma = 0.01$ for the RBF kernel and $p = 1.0$ for the polynomial kernel function. For statistical comparison of predicted and observed values; bias, coefficient of correlation (R), root mean square error ($RMSE$) and scatter index (SI) are used. These statistical measures are defined as

$$\text{bias} = \bar{y} - \bar{x} \tag{1}$$

$$R = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum(x_i - y_i)^2} \tag{3}$$

$$SI = \frac{RMSE}{\bar{x}} \quad (4)$$

In all formulas, the x_i's represent the observation, the y_i's represent the predicted value, n is the number of observations, $\bar{x}$ is the mean of x and $\bar{y}$ is the mean of y.

The inputs to be used in constructing the model are the current wind speed and those belonging to the previous hours. Evaluating the model with different number of previous wind speed values led to this conclusion that the best result could be achieved when using only six previous wind speed values together with the current wind speed. Adding more previous hours to the inputs did not change the results, so they were ignored. As a result, training and testing data has seven inputs; these inputs include the wind speed at the time of sampling and six others belonging up to six previous hours, one at every hour. Significant wave heights are considered as the output variable. Mahjoobi et al. (2008) using sensitivity analysis on wind speed, wind direction, duration and fetch length showed that wind speed is the most important parameter for wave hindcasting. They excluded fetch and duration in their work and developed an ANN model. Their results confirm that these two parameters do not play any important role in the model's accuracy. Similar results were obtained by Deo et al. (2001). They concluded that, unlike the deterministic models, fetch and duration do not seem to be important in ANN modeling. This fact can be generalized when we are using any kind of data mining approach (Mahjoobi et al., 2008). Therefore, in this work the two fetch and duration parameters are not taken into use. Table 1 shows minimum, maximum and average values of different parameters of training and testing data sets.

4.1. Wave–wind modeling

The model was developed using the training data. The training error statistics for the SVM with RBF kernel are $R = 0.97$ and $RMSE = 0.2$. These values for the polynomial kernel are $R = 0.94$ and $RMSE = 0.26$. The model is evaluated by the testing data set. The error statistics of the SVM for testing data are given in Table 2. As can be seen, the R of the model using the RBF kernel is 0.96, SI is 21.27% and, $RMSE$ and bias are 0.21 and −0.007 m, respectively. In addition, R, SI, $RMSE$ and bias for the polynomial kernel function are 0.92, 28.02%, 0.28 m and −0.05 m, respectively. As it is obvious, the RBF kernel outperforms the polynomial kernel. The model acquired form the RBF kernel is more reliable. The polynomial kernel results are given only to compare the two kernels' performance.

Table 1
Minimum, maximum and average values of different parameters in training and testing data.

Parameter	Minimum	Maximum	Average
Wind speed (m/s)-(training)	0.1	16.6	7.41
Wind speed (m/s)-(testing)	0	17.1	7.09
Significant wave height (m)-(training)	0.15	4.09	1.08
Significant wave height (m)-(testing)	0.04	3.93	1.01

A comparison between the SVM model and two different models of neural networks, multi-layer perceptron and radial basis function, is performed on the same data. Deo et al. (2001) implied that any nonlinear mathematical dependency structure can be approximated using a three layered feed-forward neural network. The network topology used here was achieve after examining different number of nodes, from 2 to 20, for the hidden layer and testing the performance of different transfer functions, e.g. sigmoid and hyperbolic tangent. Here, a three-layer feed-forward network with the sigmoid transfer functions is used. The best topology is found to be $7 \times 15 \times 1$ (neurons in the input × hidden × output layers) by trial and error. Results of these models indicate that error statistics of SVM are a bit better than both ANN models (Table 2). Comparison between observed and predicted H_s by support vector machine (RBF kernel), MLP and RBF neural networks are shown in Figs. 3–5, respectively.

Here only MLP and RBF models are tested. In spite of the fact that the best topology and parameters for these models are adopted, other ANN models may achieve better results. On the other hand, SVM parameters also have wide ranges. A good parameter selection may improve SVM performance. One can optimize SVM parameters using meta-heuristic search and optimization algorithms like genetic algorithms. Also, different choices for kernel functions in SVM exist. Choosing newly proposed kernel functions could still improve the results.

4.2. Data mining issues

The error rate estimate of the final model on validating the test data will be biased (smaller than the true error rate). In order to avoid this bias and have more trustworthy model and results, one can perform a cross-validation on the data set as a whole. K-fold cross-validation is the process of performing K different experiments. For each of K experiments, use $K-1$ folds for training and the remaining one for testing. The advantage of K-fold cross-validation is that all the examples in the data set are eventually used for both training and testing. The true error is estimated as the average error rate

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i \quad (5)$$

where E_i is the error for a single experiment. Fig. 6 illustrates the cross-validation process visually. In this work, 10-fold cross-validation is used. It is a common technique to evaluate learning algorithms on a dataset. In this method, a data set is randomly partitioned into 10 subsets (called "folds"). If there is m data points, then each fold should has $m/10$ data points. For each fold, the following processing is performed: the selected fold becomes the test set and the other 9 folds are used as the training set. The models are trained using the training set and tested on the test set.

The results comparing SVM and ANN using cross-validation are given in Table 3. The results in both the cross-validation and non-cross-validation case showed that SVMs' performance is slightly better than ANN. SVMs use the principle of structural risk

Table 2
Error statistics of predicted significant wave heights by SVM and ANN methods for testing data.

Model	Bias	$RMSE$ (m)	SI (%)	R
Support vector machine (RBF kernel)	−0.007	0.21	21.27	0.96
Support vector machine (polynomial kernel)	−0.05	0.28	28.03	0.92
Artificial neural network (MLP)	0.018	0.23	23.03	0.94
Artificial neural network (RBF)	0.024	0.25	25.07	0.93

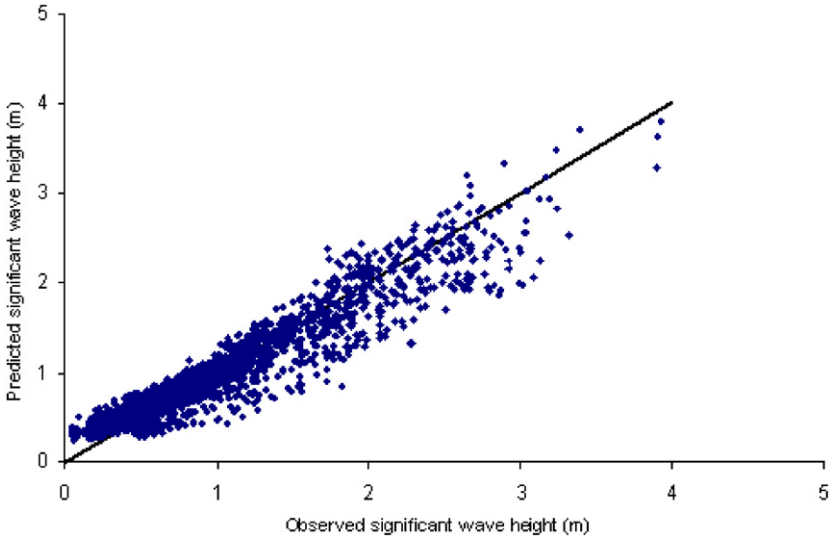

Fig. 3. Comparison between observed and predicted significant wave height by SVM (RBF kernel) method for testing data.

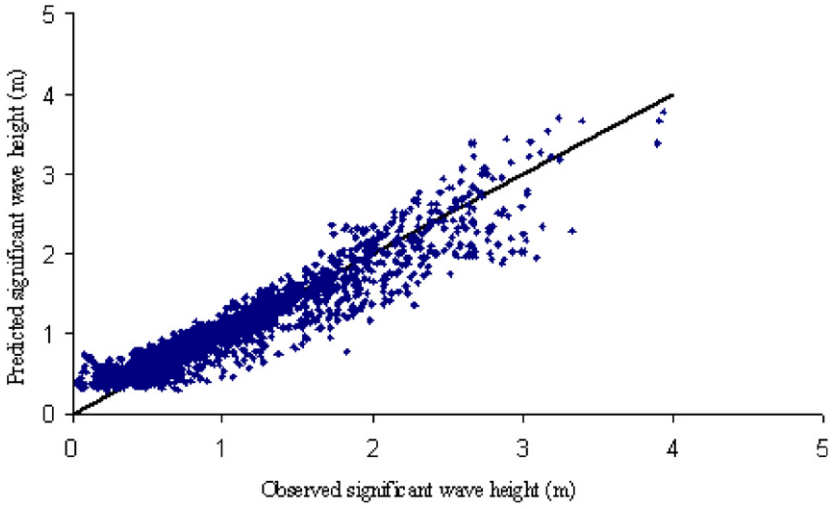

Fig. 4. Comparison between observed and predicted significant wave height by neural network method (MLP) for testing data.

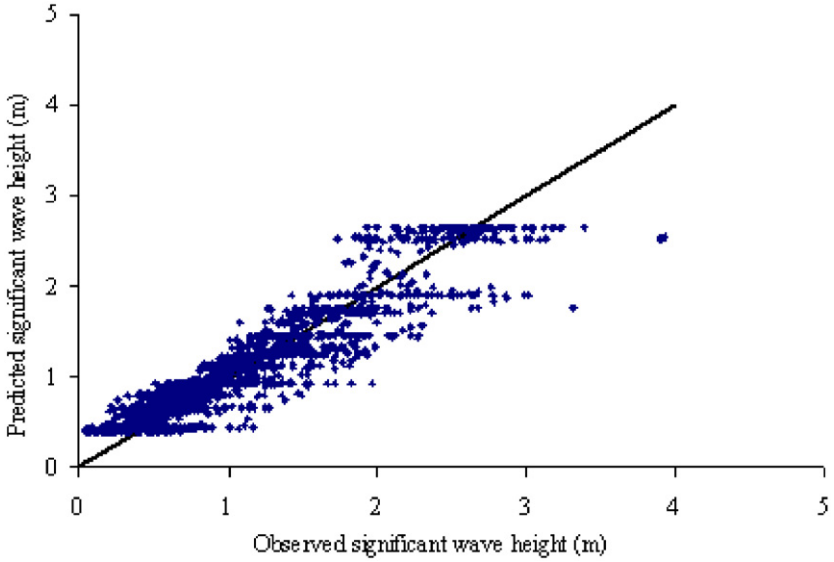

Fig. 5. Comparison between observed and predicted significant wave height by neural network method (RBF) for testing data.

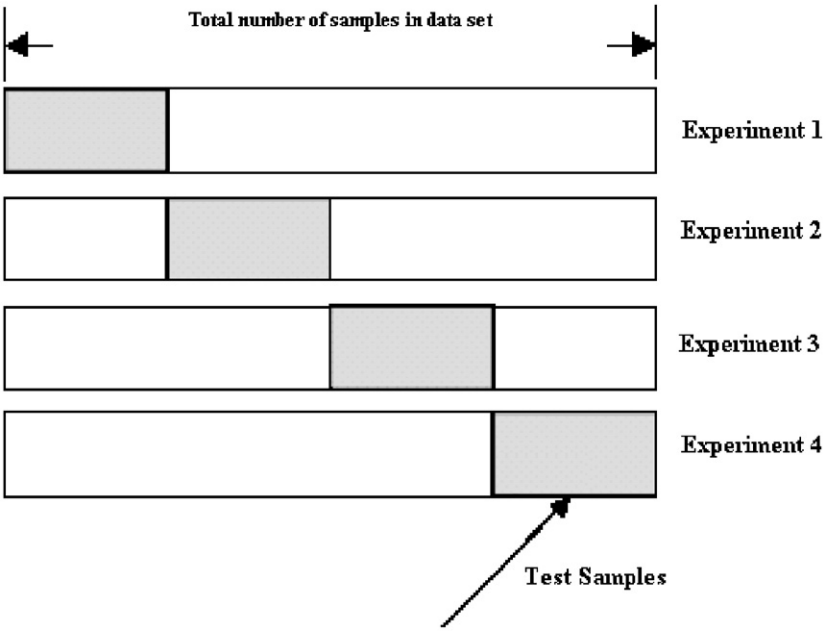

Fig. 6. *K*-fold cross-validation process.

Table 3
Error statistics of predicted significant wave heights by SVM and ANN methods (10-fold cross validation).

Model	Bias	*RMSE* (m)	*SI* (%)	*R*
Support vector machine (RBF kernel)	−0.03	0.22	20.62	0.95
Artificial neural network (MLP)	−0.014	0.24	22.74	0.93

minimization rather than empirical risk minimization in the traditional ANNs. The sequential minimal optimization also improved the performance of the SVMs both as the error statistics point of view and running time of the learner. An inherent advantage of support vector machines on using optimization algorithms, sequential minimal or quadratic programming optimization, is that they ensure a global optimum.

Furthermore, the use of support vector machines involves much lower computational cost than neural networks. This is because of using a straight forward algorithm in the learning phase. Also, the solution acquired form SVMs contributes a limited number of training points (support vectors).

Another advantage of the SVMs is that they prevent over fitting. The problem with over fitting is only disputable for ANNs. ANNs use an iterative algorithm for training the data points, if a particular section in the data is fed into the algorithm more than the other parts, the model over fits or in simpler words learns that section better. This may cause problems when testing the algorithm with a set of data that are not of the same nature of the training data. But the SVM solves a mathematical, numerical optimization problem which does not depend on the sequence or the number of repetitions of the data points (Thissen et al., 2003). Moreover, disordering the input data would lead to different models in ANNs, while the order of the input samples is not any important for SVMs.

On the other hand, SVM is less portable than ANN. Anybody can use a trained ANN, given only its architecture, weights and bias sets. But in the case of SVM, the model is much more complex and could not be used anywhere else in other implementations. Furthermore, ANNs are mostly implemented in different software packages, while this is not the case with the SVMs.

Generally, using inner kernel products leads to calculating nonlinear solutions, much simpler. Furthermore, one of the most important advantages of SVMs, in comparisons to the ANNs, is that they provide solutions with better generalizations. Due to the calculation of the support vectors and the marginal hyperplanes, the generalization error (test error in this case) would be much better. Running these algorithms on more realistic data, gathered on long time intervals would reveal this fact more obvious. There are few parameters for a SVM that needs to be determined prior the training phase. The behavior of changing these parameters is obvious. C and ε are the two SVM and parameters and in the case of RBF kernel a γ parameter is also needed for the kernel. But for ANNs, a larger number of parameters are required for the training phase, including the number of hidden layers, number of hidden nodes, momentum term, learning rate, number of training epochs, weight initialization methods, and transfer functions.

5. Summary and conclusion

Significant wave height prediction is an essential step for the study of many projects in coastal and offshore environments. In this paper, the performance of support vector machines for predicting significant wave height was investigated. For this purpose, wind wave data set gathered from deep water in Lake Michigan was invoked. It was found that the error statistics of the models for the prediction of the significant wave height decrease as the wind speed lag increases.

Application of the SVM with two different kernels, RBF and polynomial, was successfully demonstrated in this paper. The RBF showed a better performance comparing to the polynomial kernel. The results of these models were compared to those of MLP and RBF artificial neural networks. 10-fold cross-validation has also been used in the experiments, to avoid the bias and to achieve a reliable value for the model error rate.

The error statistics for the SVM with the RBF kernel were better than the results from ANN, for both using cross-validation and not using that.

The models were trained with the data gathered from 5 September to 10 December, 2002 and tested with the data from 14 September to 6 December, 2004. This time distance between the test and train data, with respect to the good results of the models, show that these machine learning models are well-constructed and tested.

The advantages of using SVM in comparisons to ANN were the fact that SVM creates a more reliable model with better generalization error, independent from the variations of the training data. SVMs do not over fit, while ANNs may face such a problem and need to deal with it. SVMs need way fewer parameters, comparing to ANNs. Also, SVMs required less computational time comparing to ANN. All these points motivate the use of SVMs in real-world engineering problems, rather than ANNs and similar machine learning tools.

Acknowledgements

The authors would like to thank Seyed Mohammad Mehdi Amiripour for his great helps preparing this document and Dr. Etemad-Shahidi and Mr. Reza Kamalian for their invaluable considerations. Also, the authors thank the three reviewers of the manuscript for their useful comments.

Appendix A

A.1. Support vector machines for pattern classification

Data classification has always been a need for data mining and machine learning applications. Assume that we have some data points, that each is a member of a class. The goal is to determine which class a new data point belongs to. As a result the objective is to maximize the separation margin. Fig. 7(a) shows different lines separating two data points, but the one illustrated in 7(b) gives the biggest margin. This separating line (say hyperplane) has the best generalization ability. SVM struggles extracting this hyperplane.

A.1.1. The optimal hyperplane (linear SVM)

Here we will briefly describe the basic SVM concepts for typical two-class classification problems. Given a training set of instance-label pairs (x_i, y_i), $i = 1, 2, \ldots, m$, where $x_i \in R$ and $y_i \in \{+1,-1\}$ for the linearly separable case, the data points will be correctly classified by

$$\begin{aligned} &\langle w.x_i\rangle + b \geqslant +1 \quad \text{for } y_i = +1 \\ &\langle w.x_i\rangle + b \leqslant -1 \quad \text{for } y_i = -1 \end{aligned} \tag{A1}$$

where w and b are the adjustable weight vector and the bias, respectively. Combining the two equations above into one set of inequalities, we will have

$$y_i(\langle w.x_i\rangle + b) - 1 \geqslant 0 \quad \forall\, i = 1, \ldots, m \tag{A2}$$

Fig. 8 illustrates an example for the linearly separable case. In this figure, you can see the separating hyperplane, the two marginal hyperplanes and the support vectors, the data points lying on the marginal hyperplanes.

The SVM finds an optimal separating hyperplane with the maximum margin by solving the following optimization problem

$$\underset{w,b}{Min}\ \frac{1}{2} w^T w \quad \text{subject to}: y_i(\langle w.x_i\rangle + b) - 1 \geqslant 0 \tag{A3}$$

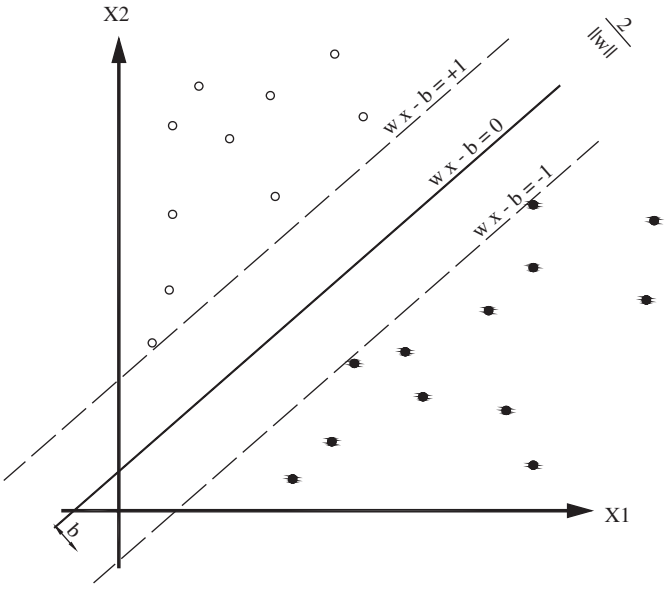

Fig. 8. Two different data points, the separating hyperplanes and the support vectors (points lying on the two marginal hyperplanes).

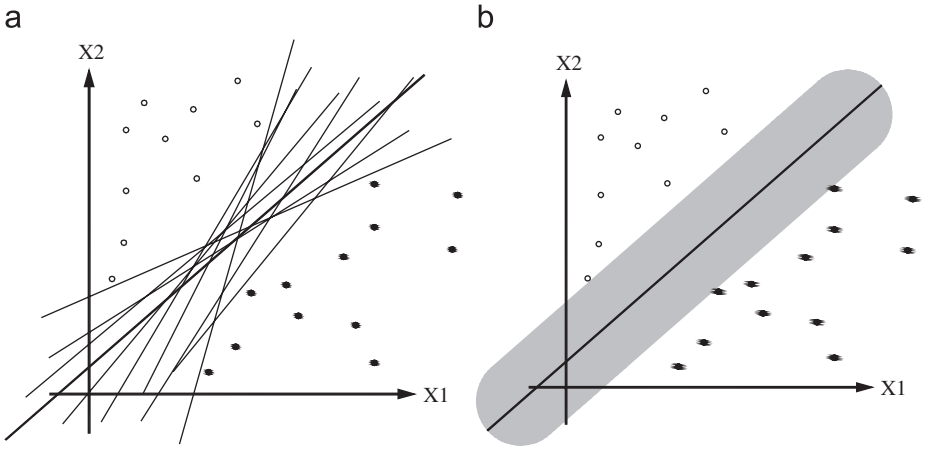

Fig. 7. (a) Different possible hyperplanes separating two classes of data points, (b) the one with the maximum marginal distance.

It is known that to solve this quadratic optimization problem, one must find the saddle point of the Lagrange function

$$L_p(w,b,\alpha)=\frac{1}{2}w^T.w-\sum_{i=1}^{m}(\alpha_i y_i(\langle w.x_i\rangle+b)-1) \tag{A4}$$

where α_i the denotes Lagrange multipliers, hence $\alpha_i \geqslant 0$. The search for an optimal saddle point is necessary because the L_p must be minimized with respect to the primal variables w and b and maximized with respect to the non-negative dual variable α_i. By differentiating with respect to w and b, the following equations are obtained:

$$\frac{\partial}{\partial w}L_p=0,\quad w=\sum_{i=1}^{m}\alpha_i y_i x_i$$
$$\frac{\partial}{\partial w}L_p=0,\quad \sum_{i=1}^{m}\alpha_i y_i=0 \tag{A5}$$

A.1.2. The optimal hyperplane for non-separable data (linear generalized SVM)

The above concepts can also be extended to the non-separable case, when there is no solution. The goal is to construct a hyperplane that makes the smallest number of errors. To get a formal setting of this problem, the non-negative slack variables $\xi_i \geqslant 0$, $i=1,\ldots,m$ are introduced (Haykin, 1999). Such that

$$\langle w.x_i\rangle+b\geqslant+1-\xi_i \quad \text{for } y_i=+1$$
$$\langle w.x_i\rangle+b\leqslant-1+\xi_i \quad \text{for } y_i=-1 \tag{A6}$$

In terms of these slack variables, the problem of finding the hyperplane that provides the minimum number of training errors, for example to keep the constraint violation as small as possible, has the formal expression

$$\underset{w,b}{Min}\frac{1}{2}w^T w+C\sum_{i=1}^{m}\xi_i$$
$$\text{subject to : } y_i(\langle w.x_i\rangle+b)+\xi_i-1\geqslant 0,\ \xi_i\geqslant 0 \tag{A7}$$

C is the parameter, concerning a tradeoff between the complexity of the SVM and the number of points misclassified. This optimization model can be solved using the Lagrangian method, which is almost equivalent to the method for solving the optimization problem in the separable case.

Accordingly, the coefficients α_i can be found by solving the following convex quadratic programming problem.

$$Q(\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\phi(x_i)^T\phi(x_j)$$
$$\text{Subject } \sum_{i=1}^{m}\alpha_i y_i=0,\quad 0\leqslant\alpha_i\leqslant C \tag{A8}$$

where $\varphi(.)$ denotes a set of nonlinear transformation between the input space and the feature apace. $K(x,x_i)=\varphi^T(x)\varphi(x_i)$ is called the inner-product kernel function, which is motivated by Mercer's theorem (Haykin, 1999). So the problem is changed to

$$Q(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \tag{A9}$$

Kernel functions are used to change the dimensionality of the input space, in order to perform the classification (or regression) task with more confidence. Two common kernel functions are radial basis function (RBF)

$$k(x,x')=\exp(-\gamma||x-x'||^2) \tag{A10}$$

and polynomial function

$$k(x,x')=(x.x'+1)^p \tag{A11}$$

The radial parameter $\gamma>0$ and p are the kernel specific parameters; they are set to values priory and used through out the training process. Other kernel functions are also introduced that may be used for specific purposes (Uestuen et al., 2006).

Support vector machines for regression

At the beginning the support vector machines where used for classification, another version SVMs was proposed by Drucker et al. (1997). Still it contains all the main features that characterize maximum margin algorithm: a nonlinear function is learned by linear learning machine mapping into high-dimensional kernel induced feature space. The capacity of the system is controlled by parameters that do not depend on the dimensionality of feature space. In the same way as with classification approach there is motivation to seek and optimize the generalization bounds given for regression. They relied on defining the loss function that ignores errors, which are situated within the certain distance of the true value. This type of function is often called epsilon intensive loss function. Fig. 9 shows an example of one-dimensional linear regression function with epsilon intensive band. ε is the degree of tolerating errors in constructing the

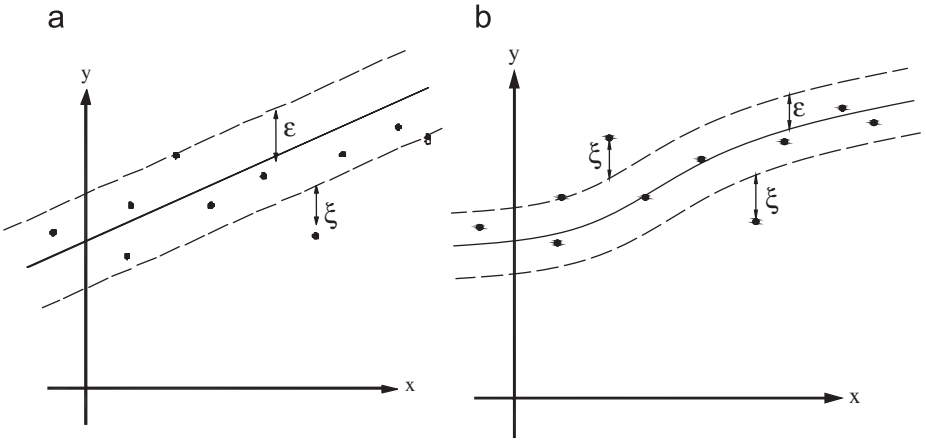

Fig. 9. (a) One-dimensional linear regression with epsilon intensive band and (b) non-linear regression.

predictor. Input symbols with errors more than ε will be penalized, the slack variables, ξ, determine this penalty. These variables measure the cost of the errors on the training points. These are zero for all points that are inside the band, Fig. 9 clearly illustrates this issue in one dimension. In SVM regression, the input x is first mapped onto an m-dimensional feature space using some fixed (nonlinear) mapping, and then a linear model is constructed in this feature space. Using mathematical notation, the linear model (in the feature space) $f(x,w)$ is given by

$$f(x,w)=\sum_{j=1}^{n} w_i g_i(x)+b \tag{A12}$$

where $g_j(x)$, $j=1,\ \ldots,\ n$ are a set of nonlinear transformations, and as before w and b are the weight vector and the bias terms. The quality of estimation is measured by the loss function $L(y,f(x,w))$. SVM regression uses a new type of loss function called ε insensitive loss function proposed by (Vapnik 1995, 1998)

$$L_\varepsilon(y,f(x,w))=\begin{cases}0 & \text{if } |y-f(x,w)|\leqslant\varepsilon\\ |y-f(x,w)|-\varepsilon & \text{otherwise}\end{cases} \tag{A13}$$

The empirical risk is

$$R_{emp}(w)=\frac{1}{m}\sum_{i=1}^{m} L_\varepsilon(y_i,f(x_i,w)) \tag{A14}$$

SVM regression performs linear regression in the high-dimension feature space using ε-insensitive loss and, at the same time, tries to reduce model complexity by minimizing $||w||^2$. This can be described by introducing (non-negative) slack variables ξ_i, ξ_i^*, $i=1,\ \ldots,\ m$ to measure the deviation of training samples outside ε-insensitive zone. Thus SVM regression is formulated as minimization of the following function:

$$\min\ \frac{1}{2}||w||^2+C\sum_{i=1}^{m}(\xi_i+\xi_i^*)$$
$$\text{such that}\begin{cases}y_i-f(x_i,w)\leqslant\varepsilon+\xi_i^*\\ f(x_i,w)-y_i\leqslant\varepsilon+\xi_i\\ \xi_i,\xi_i^*\geqslant 0,i=1,\ldots,m\end{cases} \tag{A15}$$

This optimization problem can be transformed into the dual problem and its solution is given by

$$f(x)=\sum_{i=1}^{n_{SV}}(\alpha_i-\alpha_i^*)\,k(x_i,x)$$
$$\text{Subject to}\quad 0\leqslant\alpha_i^*\leqslant C,0\leqslant\alpha_i\leqslant C \tag{A16}$$

where n_{SV} is the number of support vectors (SVs) and the kernel function is

$$k(x,x_i)=\sum_{j=1}^{n} g_j(x)g_j(x_i) \tag{A17}$$

References

Asefa, T., Kemblowski, M., McKee, M., Khalil, A., 2006. Multi-time scale stream flow predictions: the support vector machines approach. J. Hydrol. 318, 7–16.

Agrawal, J.D., Deo, M.C., 2002. On-line wave prediction. Mar. Struct. 15, 57–74.

Bhattacharya, B., Solomatine, D.P., 2006. Machine learning in soil classification. Neural Networks 19, 186–195.

Browne, M., Castelle, B., Strauss, D., Tomilnson, R., Blumenstein, M., Lane, C., 2007. Near-shore swell estimation from a global wind–wave model: spectral process, linear and artificial neural network models. Coastal Eng. 54, 445–460.

Deo, M.C., Jha, A., Chaphekar, A.S., Ravicant, K., 2001. Neural networks for wave forecasting. Ocean Eng. 28, 889–898.

Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V., 1997. Support vector regression machines. In: Mozer, M., Jordan, M., Petsche, T. (Eds.), Advances in Neural Information Processing Systems, Vol. 9. MIT Press, Cambridge, MA, pp. 155–161.

Gunn, S., 1998. Support Vector Machines for Classification and Regression. ISIS Tech Report. University of Southampton.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation, second ed. Prentice-Hall, NJ, 842pp.

Jain, P., Deo, M.C., 2006. Neural networks in ocean engineering. Int. J.Ships Offshore Struct. 1, 25–35.

Kazeminezhad, M.H., Etemad-Shahidi, A., Mousavi, S.J., 2005. Application of fuzzy inference system in the prediction wave parameters. Ocean Eng. 32, 1709–1725.

Mahjoobi, J., Etemad-Shahidi, A., 2008. An alternative approach for prediction of significant wave height based on classification and regression trees. Applied Ocean Research, Accepted.

Mahjoobi, J., Etemad-Shahidi, A., Kazeminezhad, M.H., 2008. Hindcasting of wave parameters using different soft computing methods. Appl. Ocean Res. 30, 28–36.

Makarynskyy, O., 2004. Improving wave predictions with artificial neural networks. Ocean Eng. 31 (5–6), 709–724.

Makarynskyy, O., Pires-Silva, A.A., Makarynska, D., Ventura-Soares, C., 2005. Artificial neural networks in wave predictions at the west coast of Portugal. Comput. Geosci. 31 (4), 415–424.

Mohandes, M.A., Halawani, T.O., Rehman, S., Hussain, Ahmed A., 2004. Support vector machines for wind speed prediction. Renewable Energy 29, 939–947.

Ozger, M., Sen, Z., 2007. Prediction of wave parameters by using fuzzy logic approach. Ocean Eng. 34, 460–469.

Pal, M., Goel, A., 2007. Estimation of discharge and end depth in trapezoidal channel by support vector machines. Water Resour. Manage. 21, 1763–1780.

Platt, J., 1999. Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (Eds.), Advances in Kernel Methods-Support Vector Learning. MIT Press, Cambridge, MA, pp. 185–208.

Singh, K.K., Pal, M., Ojha, C.S.P., Singh, V.P., 2008. Estimation of removal efficiency for settling basins using neural networks and support vector machines. J. Hydraulic Eng., ASCE 13 (3), 146–155.

Shevade, S.K., Keerthi, S.S., Bhattacharyya, C., Murthy, K.R.K., 2000. Improvements to the SMO algorithm for SVM regression. IEEE Trans. Neural Networks 11 (5), 1188–1193.

Smola, A.J., Schölkopf, B., 1998. A Tutorial on Support Vector Regression. Royal Holloway College, London, UK, NeuroCOLT Tech. Rep. TR 1998-030.

Thissen, U., van Brakel, R., de Weijer, A.P., Melssen, W.J., Buydens, L.M.C., 2003. Using support vector machines for time series prediction. Chemom. Intell. Lab. Syst. 69, 35–49.

Uestuen, B., Melssen, W.J., Buydens, L.M.C., 2006. Facilitating the application of support vector regression by using a universal Pearson VII function based kernel. Chemom. Intell. Lab. Syst. 81, 29–40.

US Army, 2003. Coastal Engineering Manual. Chapter II-2, Meteorology and Wave Climate. Engineer Manual 1110-2-1100. US Army Corps of Engineers, Washington, DC.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer, NY.

Vapnik, V., 1998. Statistical Learning Theory. Springer, NY.

Yu, P.S., Chen, S.T., Chang, I.F., 2006. Support vector regression for real-time flood stage forecasting. J. Hydrology 328, 704–716.