

## *Research Article*

# **Prediction of Hydrocarbon Reservoirs Permeability Using Support Vector Machine**

**R. Gholami,<sup>1</sup> A. R. Shahraki,<sup>2</sup> and M. Jamali Paghaleh<sup>3</sup>**

<sup>1</sup> Faculty of Mining, Petroleum and Geophysics, Shahrood University of Technology, P.O. Box 316, Shahrood, Iran

<sup>2</sup> Department of Industrial Engineering, University of Sistan and Baluchestan, Zahedan, Iran

<sup>3</sup> Young Researchers Club, Islamic Azad University, Zahedan Branch, Zahedan 98168, Iran

Correspondence should be addressed to M. Jamali Paghaleh, mj.paghaleh@gmail.com

Received 16 July 2011; Revised 8 October 2011; Accepted 1 November 2011

Academic Editor: P. Liatsis

Copyright © 2012 R. Gholami et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Permeability is a key parameter associated with the characterization of any hydrocarbon reservoir. In fact, it is not possible to have accurate solutions to many petroleum engineering problems without having accurate permeability value. The conventional methods for permeability determination are core analysis and well test techniques. These methods are very expensive and time consuming. Therefore, attempts have usually been carried out to use artificial neural network for identification of the relationship between the well log data and core permeability. In this way, recent works on artificial intelligence techniques have led to introduce a robust machine learning methodology called support vector machine. This paper aims to utilize the SVM for predicting the permeability of three gas wells in the Southern Pars field. Obtained results of SVM showed that the correlation coefficient between core and predicted permeability is 0.97 for testing dataset. Comparing the result of SVM with that of a general regression neural network (GRNN) revealed that the SVM approach is faster and more accurate than the GRNN in prediction of hydrocarbon reservoirs permeability.

## **1. Introduction**

In reservoir engineering, reservoir management, and enhanced recovery design point of view, permeability is the most important rock parameter affecting fluids flow in reservoir. Knowledge of rock permeability and its spatial distribution throughout the reservoir is of utmost importance. In fact, the key parameter for reservoir characterization is the permeability distribution. In addition, in most reservoirs, permeability measurements are rare and therefore permeability must be predicted from the available data. Thus, the accurate estimation of

permeability can be considered as a difficult task. Permeability is generally measured in the laboratory on the cored rocks taken from the reservoir or can be determined by well test techniques. The well testing and coring methods are, however, very expensive and time consuming compared to the wire-line logging techniques [1]. Moreover, in a typical oil or gas field, almost all wells are logged using various tools to measure geophysical parameters such as porosity and density, while both well test and core data are available only for a few wells [2, 3]. As the well log data are usually available for most of wells, many researchers attempt to predict permeability through establishing a series of statistical empirical correlation between permeability, porosity, water saturation, and other physical properties of rocks. This technique has been used with some success in sandstone and carbonate reservoirs, but it often shows short of the accuracy for permeability prediction from well log data in heterogeneous reservoirs. Proton magnetic resonance (PMR) is another modern approach for the prediction of the permeability as a continuous log, but it has significant technological constrains [4].

Alternatively, neural networks have been increasingly applied to predict reservoir properties using well log data [5–7]. In this way, previous investigations [8–15] have revealed that neural network is a proper tool for identifying the complex relationship among permeability, porosity, fluid saturations, depositional environments, lithology, and well log data. However, recent works on the artificial intelligence have resulted in finding a suitable machine learning theory called support vector machine (SVM). Support vector machines, based on the structural risk minimization (SRM) principle [16], seem to be a promising method for data mining and knowledge discovery. It was introduced in the early 90s as a nonlinear solution for classification and regression tasks [17, 18]. It stems from the frame work of statistical learning theory or Vapnik-Chervonenkis (VC) theory [19, 20] and was originally developed for pattern recognition problem [21, 22]. VC theory is the most successful tool by now for accurately describing the capacity of a learned model and can further tell us how to ensure the generalization performance for future samples [23] by controlling the capacity of a model. Its theory is mainly based on the consistency and rate of convergence of a learning process [20]. VC dimension and SRM principle are the two most important elements of VC theory. VC dimension is a measure of the capacity of a set of functions, and SRM principle can ensure that the model will be well generalized. SVM is firmly rooted in VC theory and has superiority compared to the traditional learning methods. As a matter of fact, there are at least three reasons for the success of SVM: its ability to learn well with only a very small number of parameters, its robustness against the error of data, and its computational efficiency compared with several other intelligent computational methods such as neural network, fuzzy network, and so forth [24, 25]. By minimizing the structural risk, SVM works well not only in classification [26–28] but also in regression [29, 30]. It was soon introduced into many other research fields, for example, image analysis [31, 32], signal processing [33], drug design [34], gene analysis [35], climate change [36], remote sensing [37], protein structure/function prediction [38], and time series analysis [39] and usually outperformed the traditional statistical learning methods [40]. Thus, SVM has been receiving increasing attention and quickly becomes quite an active research field. The objective of this study is to evaluate the ability of SVM for prediction of the reservoirs permeability. In this way, Kangan and Dallan gas reservoirs in the Southern Pars field, Iran are selected as case study in this research work. In addition, obtained results of SVM will be compared to those given by a general regression neural network (GRNN).



Figure 1: Geographical position of Southern Pars gas field.

## 2. Study Area

The Iranian South Pars field is the northern extension of Qatar's giant North Field. It covers an area of 500 square miles and is located 3,000 m below the seabed at a water depth of 65 m. The Iranian side accounts for 10% of the world's and 60% of Iran's total gas reserves. Iran's portion of the field contains an estimated 436 trillion cubic feet. The field consists of two independent gas-bearing formations, Kangan (Triassic) and Dalan (Permian). Each formation is divided into two different reservoir layers, separated by impermeable barriers. The field is a part of the N-trending Qatar Arch structural feature that is bounded by Zagros fold belt to the north and northeast. In the field, gas accumulation is mostly limited to the Permian-Triassic stratigraphic units. These units known as the "Kangan-Dalan Formations" constitute which are very extensive natural gas reservoirs in the Persian Gulf area and consist of carbonate-evaporate series also known as the Khuff Formation [41]. Figure 1 shows the geographical position of Southern Pars gas field.

## 3. Material and Method

### 3.1. Data Set

The main objective of this study is to predict permeability of the gas reservoirs by incorporating well logs and core data of three gas wells in the Southern Pars field. As a matter of fact, the well logs are considered as the inputs, whereas horizontal permeability ( $k_h$ ) is taken as the output of the networks. Available digitized well logs are consisted of sonic log (DT), gamma

ray log (GR), compensated neutron porosity log (NPHI), density log (ROHB), photoelectric factor log (PEF), microspherical focused resistivity log (MSFL), and shallow and deep latero-resistivity logs (LLS and LLD). For the present study, a total number of 175 well logs and core permeability datasets were obtained from 3 wells of Kangan and Dalan gas reservoirs. In view of the requirements of the networks computation algorithms, the data of the input and output variables were normalised to an interval by transformation process. In this study, normalization of data (inputs and outputs) was done for the range of  $(-1, 1)$  using (3.1), and the number of training (125) and testing data (50) was then selected randomly:

$$p_n = 2 \frac{p - p_{\min}}{p_{\max} - p_{\min}} - 1, \quad (3.1)$$

where  $p_n$  is the normalised parameter,  $p$  denotes the actual parameter,  $p_{\min}$  represents a minimum of the actual parameters, and  $p_{\max}$  stands for a maximum of the actual parameters. In addition, the leave-one-out (LOO) cross-validation of the whole training set was used for adjusting the associated parameters of the networks [42].

### 3.2. Independent Component Analysis

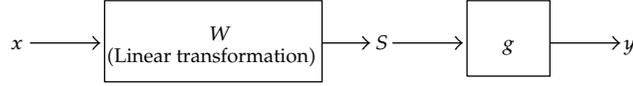
Independent component analysis (ICA) is a suitable method for feature extraction process. Unlike the PCA, this method both decorrelates the input signals and also reduces higher-order statistical dependencies [43, 44]. ICA is based on the non-Gaussian assumption of the independent sources and uses higher-order statistics to reveal interesting features. The applications of IC transformation include dimension reduction, extracting characteristics of the image, anomaly and target detection, feature separation, classification, noise reduction, and mapping. Compared with principal component (PC) analysis, IC analysis provides some unique advantages:

- (i) PC analysis is orthogonal decomposition. It is based on covariance matrix analysis and Gaussian assumption. IC analysis is based on non-Gaussian assumption of the independent sources;
- (ii) PC analysis uses only second-order statistics. IC analysis uses higher-order statistics. Higher-order statistics is a stronger statistical assumption, revealing interesting features in the usually non-Gaussian datasets [45, 46].

#### 3.2.1. The ICA Algorithm

Independent Component Analysis of the random vector  $x$  is defined as the process of finding a linear transform  $S = Wx$  such that  $W$  is a linear transformation and the components  $s_i$  are as independent as possible. They maximize a function  $F(s_1, \dots, s_m)$  that provides a measure of independence [46]. The individual components  $s_i$  are independent when their probability distribution factorizes (when there is no mutual information between them) and can be mentioned as (3.2):

$$f_s(s) = \prod_{s_i} f_{s_i}(s_i). \quad (3.2)$$



**Figure 2:** Mechanism of data transformation via ICA algorithm.

This approach used to minimize the mutual information involves maximizing the joint entropy  $H[g(s)]$ . This is accomplished using a stochastic gradient ascent method, termed Infomax. If the nonlinear function  $g$  is the cumulative density function of the independent components  $s_i$ , then this method also minimizes the mutual information. The procedure of transforming data in higher dimension is shown in Figure 2.

Notice that the nonlinear function  $g$  is chosen without knowing the cumulative density functions of the independent components. In case of a mismatch, it is possible that the algorithm does not converge to a solution [47]. A set of nonlinear functions has been tested, and it was found that super-Gaussian probability distribution functions converge to an ICA solution when the joint entropy is maximized. The optimization step in obtaining the independent components relies on changing the weights according to the entropy gradient that can be expressed as (3.3):

$$\Delta W \propto \frac{\partial H(y)}{\partial W} = E \left[ \frac{\partial \ln |J|}{\partial W} \right], \quad (3.3)$$

where  $E$  is the expected value,  $y = [g(s_1) \cdots g(s_n)]^T$ , and  $|J|$  is the absolute value of the determinant of the Jacobian matrix of the transformation from  $x$  to  $y$ . From this formula, it is shown that  $\Delta W$  can be calculated using

$$\Delta W = (W^T)^{-1} + \left( \frac{\partial g(s)/\partial s}{g(s)} \right) \cdot x^T. \quad (3.4)$$

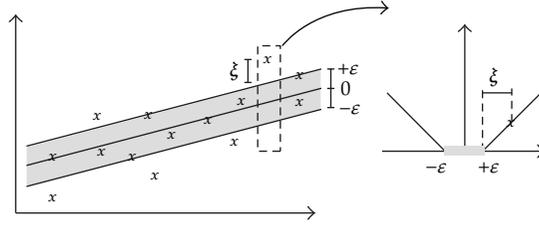
Equation (3.4) involves the matrix inversion, and thus an alternative formulation involving only simple matrix multiplication is preferred. This formula can be mentioned as [43, 47]:

$$\Delta W \propto \frac{\partial H(y)}{\partial W} W^T W = \left( (W^T)^{-1} + \left( \frac{\partial g(s)/\partial s}{g(s)} \right) \cdot x^T \right) W^T W = W + \left( \frac{\partial g(s)/\partial s}{g(s)} \right) s^T W. \quad (3.5)$$

### 3.3. Support Vector Machine

In pattern recognition, the SVM algorithm constructs nonlinear decision functions by training a classifier to perform a linear separation in high-dimensional space which is nonlinearly related to input space. To generalize the SVM algorithm for regression analysis, an analogue of the margin is constructed in the space of the target values ( $y$ ) using Vapnik's  $\varepsilon$ -insensitive loss function (Figure 3) [48–50]:

$$|y - f(x)|_\varepsilon := \max\{0, |y - f(x) - \varepsilon|\}. \quad (3.6)$$



**Figure 3:** Concept of  $\varepsilon$ -insensitivity. Only the samples out of the  $\pm\varepsilon$  margin will have a non-zero slack variable, so they will be the only ones that will be part of the solution [50].

To estimate a linear regression,

$$f(x) = (w \cdot x) + b, \quad (3.7)$$

where  $w$  is the weighting matrix,  $x$  is the input vector, and  $b$  is the bias term. With precision, one minimizes

$$\frac{1}{2}\|w\|^2 + C \sum_{i=1}^m |y - f(x)|_{\varepsilon}, \quad (3.8)$$

where  $C$  is a trade-off parameter to ensure that the margin  $\varepsilon$  is maximized and error of the classification  $\xi$  is minimized. Considering a set of constraints, one may write the following relations as a constrained optimization problem:

$$L(w, \xi, \xi') = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi'_i). \quad (3.9)$$

Subject to

$$y_i - w^T \cdot x - b \leq \xi_i + \varepsilon, \quad (3.10)$$

$$w^T \cdot x + b - y_i \leq \xi'_i + \varepsilon, \quad (3.11)$$

$$\xi_i, \xi'_i, x_i \geq 0. \quad (3.12)$$

According to relations (3.10) and (3.11), any error smaller than  $\varepsilon$  does not require a nonzero  $\xi_i$  or  $\xi'_i$  and does not enter the objective function (3.9) [51–53].

By introducing Lagrange multipliers ( $\alpha$  and  $\alpha'$ ) and allowing for  $C > 0$ ,  $\varepsilon > 0$  chosen a priori, the equation of an optimum hyper plane is achieved by maximizing the following relations:

$$L(\alpha, \alpha') = \frac{1}{2} \sum_{i=1}^N \sum_{i=1}^N (\alpha_i - \alpha'_i) x'_i x_i (\alpha_i - \alpha'_i) + \sum_{i=1}^N ((\alpha_i - \alpha'_i) y_i - (\alpha_i + \alpha'_i) \varepsilon) \quad (3.13)$$

$$\text{Subject to } 0 \leq (\alpha_i - \alpha'_i) \leq C,$$

**Table 1:** Polynomial, normalized polynomial, and radial basis function (Gaussian) kernels [24].

Kernel function	Type of classifier
$K(x_i, x_j) = (x_i^T x_j + 1)^\rho$	Complete polynomial of degree $\rho$
$K(x_i, x_j) = (x_i^T x_j + 1)^\rho / \sqrt{(x_i^T x_j) - (y_i^T y_j)}$	Normalized polynomial kernel of degree $\rho$
$K(x_i, x_j) = \exp[-\ x_i - x_j\ ^2 / 2\sigma^2]$	Gaussian (RBF) kernel with parameter $\sigma$ which controls the half-width of the curve fitting peak

where  $x_i$  only appears inside an inner product. To get a potentially better representation of the data in nonlinear case, the data points can be mapped into an alternative space, generally called feature space (a pre-Hilbert or inner product space) through a replacement:

$$x_i \cdot x_j \longrightarrow \varphi(x_i) \cdot \varphi(x_j). \quad (3.14)$$

The functional form of the mapping  $\varphi(x_i)$  does not need to be known since it is implicitly defined by the choice of kernel:  $k(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$  or inner product in Hilbert space. With a suitable choice of kernel, the data can become separable in feature space while the original input space is still nonlinear. Thus, whereas data for  $n$ -parity or the two-spiral problem is non-separable by a hyperplane in input space, it can be separated in the feature space by the proper kernels [54–59]. Table 1 gives some of the common kernels.

Then, the nonlinear regression estimate takes the following form:

$$y_i = \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha'_i) \varphi(x_i)^T \varphi(x_j) + b = \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha'_i) K(x_i, x_j) + b, \quad (3.15)$$

where  $b$  is computed using the fact that (3.5) becomes an equality with  $\xi_i = 0$  if  $0 < \alpha_i < C$  and relation (3.6) becomes an equality with  $\xi'_i = 0$  if  $0 < \alpha'_i < C$  [60, 61].

### 3.4. General Regression Neural Network

General regression neural network has been proposed by [62]. GRNN is a type of supervised network and also trains quickly on sparse data sets, rather than categorising it. GRNN applications are able to produce continuous valued outputs. GRNN is a three-layer network with one hidden neuron for each training pattern.

GRNN is a memory-based network that provides estimates of continuous variables and converges to the underlying regression surface. GRNN is based on the estimation of probability density functions, having a feature of fast training times, and can model nonlinear functions. GRNN is a one-pass learning algorithm with a highly parallel structure. GRNN algorithm provides smooth transitions from one observed value to another even with sparse data in a multidimensional measurement space. The algorithmic form can be used for any regression problem in which an assumption of linearity is not justified. GRNN can be thought as a normalised radial basis functions (RBF) network in which there is a hidden unit centred at every training case. These RBF units are usually probability density functions such as the Gaussian. The only weights that need to be learned are the widths of the RBF units. These widths are called “smoothing parameters.” The main drawback of GRNN is that it suffers badly from the curse of dimensionality. GRNN cannot ignore irrelevant inputs without major

modifications to the basic algorithm. So, GRNN is not likely to be the top choice if there are more than 5 or 6 nonredundant inputs. The regression of a dependent variable,  $Y$ , on an independent variable,  $X$ , is the computation of the most probable value of  $Y$  for each value of  $X$  based on a finite number of possibly noisy measurements of  $X$  and the associated values of  $Y$ . The variables  $X$  and  $Y$  are usually vectors. In order to implement system identification, it is usually necessary to assume some functional form. In the case of linear regression, for example, the output  $Y$  is assumed to be a linear function of the input, and the unknown parameters,  $a_i$ , are linear coefficients.

The method does not need to assume a specific functional form. A Euclidean distance ( $D_i^2$ ) is estimated between an input vector and the weights, which are then rescaled by the spreading factor. The radial basis output is then the exponential of the negatively weighted distance. The GRNN equation can be written as:

$$D_i^2 = (X - X^i)^T (X - X^i),$$

$$Y(X) = \frac{\sum_{i=1}^n Y_i \exp(-D_i^2/2\sigma^2)}{\sum_{i=1}^n \exp(-D_i^2/2\sigma^2)}, \quad (3.16)$$

where  $\sigma$  is the smoothing factor (SF).

The estimate  $Y(X)$  can be visualised as a weighted average of all of the observed values,  $Y_i$ , where each observed value is weighted exponentially according to its Euclidean distance from  $X$ .  $Y(X)$  is simply the sum of Gaussian distributions centred at each training sample. However, the sum is not limited to being Gaussian. In this theory, the optimum smoothing factor is determined after several runs according to the mean squared error of the estimate, which must be kept at minimum. This process is referred to as the training of the network. If a number of iterations pass with no improvement in the mean squared error, then smoothing factor is determined as the optimum one for that data set. While applying the network to a new set of data, increasing the smoothing factor would result in decreasing the range of output values [63]. In this network, there are no training parameters such as the learning rate, momentum, optimum number of neurons in hidden layer and learning algorithms as in BPNN, but there is a smoothing factor that its optimum is gained as try and error. The smoothing factor must be greater than 0 and can usually range from 0.1 to 1 with good results. The number of neurons in the input layer is the number of inputs in the proposed problem, and the number of neurons in the output layer corresponds to the number of outputs. Because GRNN networks evaluate each output independently of the other outputs, GRNN networks may be more accurate than BPNN when there are multiple outputs. GRNN works by measuring how far given samples pattern is from patterns in the training set. The output that is predicted by the network is a proportional amount of all the output in the training set. The proportion is based upon how far the new pattern is from the given patterns in the training set.

## 4. Implementation

### 4.1. ICA Implementation and Determination of the Most Relevant Well Logs

As it was mentioned, ICA is one of the suitable methods for extracting the most important and relevant features of any particular dataset. Hence, in this paper, we used this method for

**Table 2:** Correlation matrix of the well logs and permeability after applying ICA.

	X	Y	Z	GR	DT	RHOB	NPHI	PEF	MSFL	LLD	LLS	KH
X	1.000											
Y	-.999	1.000										
Z	.655	-.655	1.000									
GR	.419	-.216	.490	1.000								
DT	.525	.026	.447	.488	1.000							
RHOB	.611	.010	.597	.582	.915	1.000						
NPHI	.710	.107	.440	.567	.746	.615	1.000					
PEF	.125	.619	.161	.210	.150	.176	.009	1.000				
MSFL	.772	-.070	.564	.459	.627	.401	-.318	-.236	1.000			
LLD	.607	-.510	.506	.541	.553	.469	-.101	.278	.574	1.000		
LLS	.577	-.678	.523	.602	.539	.648	-.203	.215	.635	.767	1.000	
KH	.607	.052	.676	.689	.829	.534	.836	.045	.642	.720	.603	1.000

**Table 3:** Rotated component matrix of the parameters.

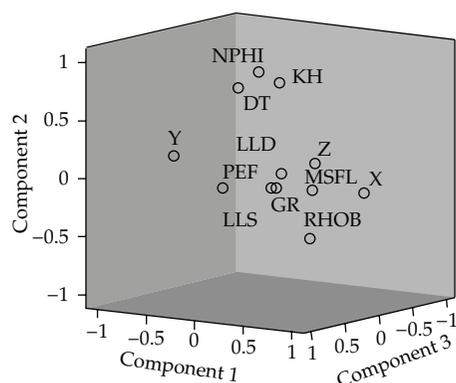
Parameters	Component 1	Component 2
X	.971	-.065
Y	-.065	.865
Z	.768	.129
GR	.653	-.018
DT	.808	.100
RHOB	.630	-.357
NPHI	.826	.092
PEF	.094	.802
MSFL	.677	-.218
LLD	.779	-.014
LLS	.608	-.132
KH	.816	.074

identification of those well logs that have a good relationship with permeability. Table 2 gives the correlation matrix of the well logs and permeability after applying ICA in the first step.

As it is seen in Table 2, most of the well logs have a good relationship with permeability except Y and PEF. Hence, it can be concluded that photoelectric factor log (PEF) and Y coordinate should be ignored as the inputs because of the weak relationship with permeability. For further analysis, rotated component matrix was determined showing the number of components of the dataset. This matrix is shown in Table 3.

Regarding Table 3, it can be observed that the entire well logs and their coordinate are divided into two different components. In the first component, eight parameters including X and Z coordinates, GR, DT, RHOB, NPHI, MSFL, LLD, and LLS well logs and permeability were categorized. The second component is consisted of Y coordinate and PEF well log. This table clearly shows that component no. 1 is the one which can be used for prediction of permeability. Figure 4 is another representation showing the obtained results of Table 3 in a graphical form.

As it is also shown in Figure 4, Y and PEF are not associated with the others in a same side of the cubic. Regarding the obtained results of ICA, five well logs including GR, DT,



**Figure 4:** A graphical form of representation for showing the relationship of well logs and permeability.

RHOB, NPHI, MSFL, LLD, and LLS and two coordinates, X and Z, are taken into account for prediction of permeability using the networks.

#### 4.2. Implementation of SVM

Similar to other multivariate statistical models, the performance of SVM for regression depends on the combination of several parameters. They are capacity parameter  $C$ ,  $\varepsilon$  (epsilon) of  $\varepsilon$ -insensitive loss function, and the kernel type  $K$  and its corresponding parameters.  $C$  is a regularization parameter that controls the tradeoff between maximizing the margin and minimizing the training error. If  $C$  is too small, then insufficient stress will be placed on fitting the training data. If  $C$  is too large, then the algorithm will overfit the training data. But, Wang et al. [64] indicated that prediction error was scarcely influenced by  $C$ . In order to make the learning process stable, a large value should be set up for  $C$  (e.g.,  $C = 100$ ).

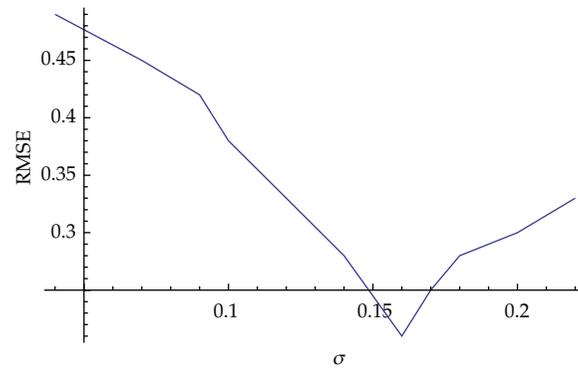
The optimal value for  $\varepsilon$  depends on the type of noise present in the data, which is usually unknown. Even if enough knowledge of the noise is available to select an optimal value for  $\varepsilon$ , there is the practical consideration of the number of resulting support vectors.  $\varepsilon$ -insensitivity prevents the entire training set meeting boundary conditions and so allows for the possibility of sparsity in the dual formulations solution. So, choosing the appropriate value of  $\varepsilon$  is critical from theory.

Since in this study the nonlinear SVM is applied, it would be necessary to select a suitable kernel function. The obtained results of previous published researches [64, 65] indicated that the Gaussian radial basis function has a superior efficiency than other kernel functions. As it is seen in Table 1, the form of the Gaussian kernel is as follow:

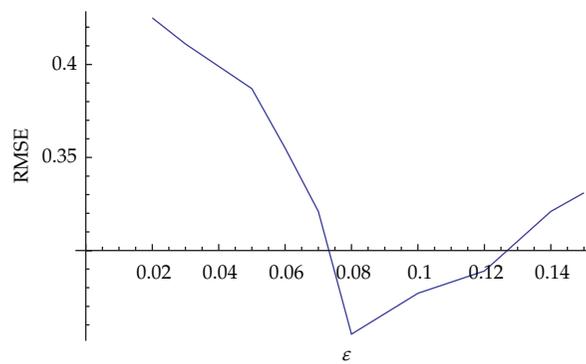
$$K(x_i, x_j) = e^{-|x_i - x_j|^2 / 2\sigma^2}, \quad (4.1)$$

where  $\sigma$  (sigma) is a constant parameter of the kernel. This parameter can control the amplitude of the Gaussian function and the generalization ability of SVM. We have to optimize  $\sigma$  and find the optimal one.

In order to find the optimum values of two parameters ( $\sigma$  and  $\varepsilon$ ) and prohibit the overfitting of the model, the data set was separated into a training set of 125 compounds and a test set of 50 compounds randomly and the leave-one-out cross-validation of the whole training



**Figure 5:** Sigma versus RMS error on LOO cross-validation.



**Figure 6:** Epsilon versus RMS error on LOO cross-validation.

set was performed. The leave-one-out (LOO) procedure consists of removing one example from the training set, constructing the decision function on the basis only of the remaining training data and then testing on the removed example [50]. In this fashion, one tests all examples of the training data and measures the fraction of errors over the total number of training examples. In this process, root mean square error (RMS) is commonly used as an error function to evaluate the quality of model.

Detailed process of selecting the parameters and the effects of every parameter on generalization performance of the corresponding model are shown in Figures 5 and 6. To obtain the optimal value of  $\sigma$ , the SVM with different  $\sigma$ s was trained, with the  $\sigma$  varying from 0.01 to 0.3, every 0.01. We calculated the RMS on different  $\sigma$ s, according to the generalization ability of the model based on the LOO cross-validation for the training set. The curve of RMS versus the sigma was shown in Figure 5. In this regard, the optimal  $\sigma$  was found as 0.16.

In order to find an optimal  $\epsilon$ , the RMS on different  $\epsilon$ s was calculated. The curve of the RMS versus the epsilon was shown in Figure 6. From Figure 6, the optimal  $\epsilon$  was found as 0.08.

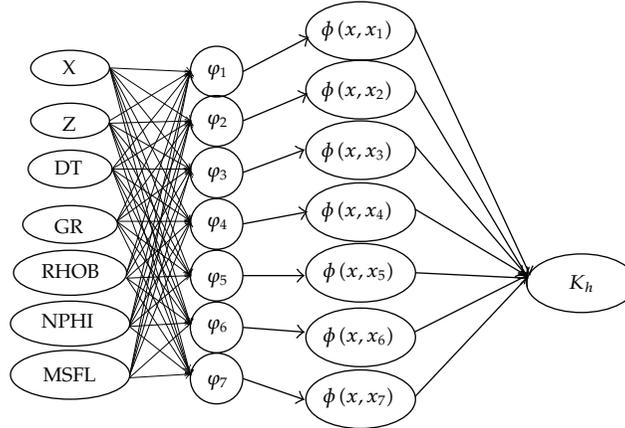


Figure 7: Schematic diagram of optimum SVM for prediction of permeability.

From the above discussion, the  $\sigma$ ,  $\varepsilon$ , and  $C$  were fixed to 0.16, 0.08, and 100, respectively, when the support vector number of the SVM model was 41. Figure 7 is a schematic diagram showing the construction of the SVM.

### 4.3. Implementation of GRNN

In order to check the accuracy of the SVM in prediction of permeability, obtained results of SVM are compared with those of the general regression neural network (GRNN). Constructed GRNN of this study was a multilayer neural network with one hidden layer of radial basis function consisting 49 neurons and an output layer containing only one neuron. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between the input and output vectors.

In addition, the performance of GRNN depends mostly on the choice of smooth factor (SF) which is in a sense equivalent to the choice of the ANN structure. For managing this issue, LOO cross-validation technique has been used and optimal SF was found as 0.23. Figure 8 shows the obtained results of this study.

## 5. Results

### 5.1. Prediction of Permeability Using SVM

After building an optimum SVM based on the training dataset, performance of constructed SVM was evaluated in testing process. Figure 9 is a demonstration showing the ability of SVM in prediction of permeability.

As it is illustrated in Figure 9, there is an acceptable agreement (correlation coefficient of 0.97) between the predicted and measured permeability. In fact, the SVM is an appropriate method for prediction of permeability. Nonetheless, the performance of this method should be compared with another suitable method for highlighting the strength of SVM.

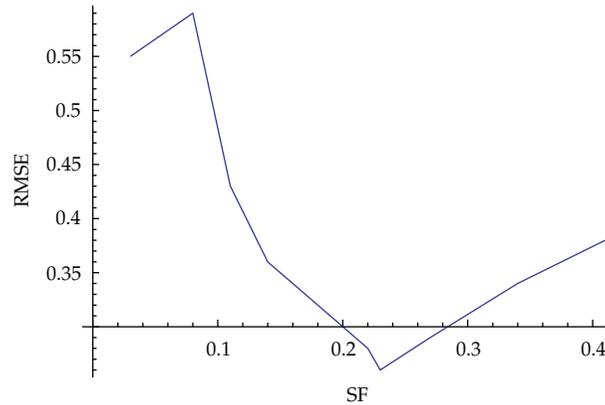


Figure 8: SF versus RMS error on LOO cross-validation.

Table 4: Comparing the performance of SVM and BPNN methods in the training and testing process.

Model	R (train)	R (test)	RMSE (train)	RMSE (test)
GRNN	0.998	0.94	0.16	0.35
SVM	0.998	0.96	0.16	0.28

## 5.2. Prediction of Permeability Using GRNN

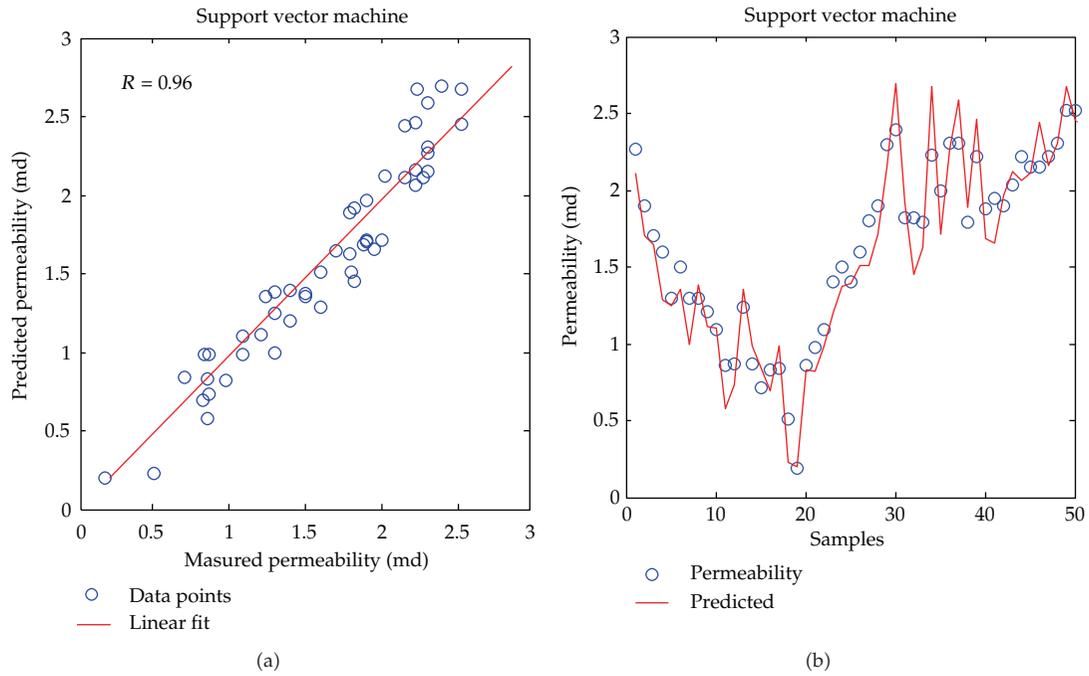
As it was already mentioned, to check the accuracy of the SVM in the prediction of permeability, obtained results of SVM are compared with those of the general regression neural network (GRNN). Figure 10 shows the obtained results of prediction via GRNN.

As it is shown in Figure 10, GRNN is a good network for prediction of permeability. However, it is not as strong as the SVM in prediction process. Hence, it can be considered as the second alternative after the SVM for prediction of permeability.

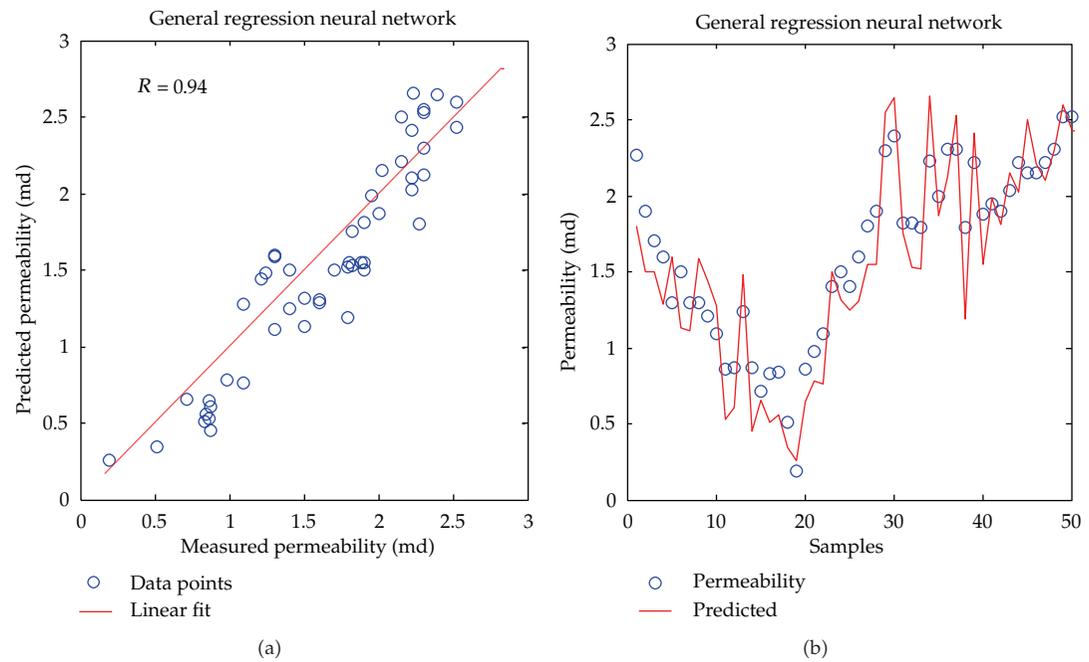
## 6. Discussion

In this research work, we have demonstrated one of the applications of artificial intelligence techniques in forecasting the hydrocarbon reservoirs permeability. Firstly, independent component analysis (ICA) has been used for determination of relationship between the well log data and permeability. In that section, we found that  $Y$  and PEF are not suitable inputs for the networks because of the weak relationship with the permeability (Figure 4). As a matter of fact, GR, DT, RHOB, NPHI, MSFL, LLD, and LLS well logs along with  $X$  and  $Z$  coordinates were used for the training of the networks. In this regard, we developed two Matlab software codes (i.e., M.file) and interrogated the performance of SVM with the best performed work of the GRNN method. When comparing SVM with this model (Table 4), it presented overall better efficiency over the GRNN in terms of RMS error in both training and testing process.

According to this table, the RMS error of the SVM is smaller than the GRNN. In terms of running time, the SVM consumes a considerably less time (3 second) for the prediction compared with that of the GRNN (6 second). All of these expressions can introduce the SVM as a robust algorithm for the prediction of permeability.



**Figure 9:** Relationship between the measured and predicted permeability obtained by SVM (a); estimation capability of SVM (b).



**Figure 10:** Relationship between the measured and predicted permeability obtained by GRNN (a); estimation capability of GRNN (b).

## 7. Conclusions

Support vector machine (SVM) is a novel machine learning methodology based on statistical learning theory (SLT), which has considerable features including the fact that requirement on kernel and nature of the optimization problem results in a uniquely global optimum, high generalization performance, and prevention from converging to a local optimal solution. In this research work, we have shown the application of SVM compared with GRNN model for prediction of permeability of three gas wells in the Kangan and Dalan reservoir of Southern Pars field, based on the digital well log data. Although both methods are data-driven models, it has been found that SVM makes the running time considerably faster with the higher accuracy. In terms of accuracy, the SVM technique resulted in an RMS error reduction relative to that of the GRNN model (Table 4). Regarding the running time, SVM requires a small fraction of the computational time used by GRNN, an important factor to choose an appropriate and high-performance data-driven model.

For the future works, we are going to test the trained SVM for predicting the permeability of the other reservoirs in the south of Iran. Undoubtedly, receiving meaningful results from other reservoirs using well log data can further prove the ability of SVM in prediction of petrophysical parameters including permeability.

## Acknowledgment

The authors appreciate anonymous reviewers for their constructive comments and contributions in improving the paper.

## References

- [1] A. Bhatt, *Reservoir properties from well logs using neural networks*, Ph.D. thesis, Department of Petroleum Engineering and Applied Geophysics, Norwegian University of Science and Technology, A dissertation for the partial fulfillment of requirements, 2002.
- [2] W. F. Brace, "Permeability from resistivity and pore shape," *Journal of Geophysical Research*, vol. 82, no. 23, pp. 3343–3349, 1977.
- [3] D. Tiab and E. C. Donaldsson, *Petrophysics, Theory and Practice of Measuring Reservoir Rock and Fluid Properties*, Gulf Publishing Co., 2004.
- [4] N. Kumar, N. Hughes, and M. Scott, *Using Well Logs to Infer Permeability*, Center for Applied Petrophysical Studies, Texas Tech University, 2000.
- [5] S. Mohaghegh, R. Arefi, S. Ameri, and D. Rose, "Design and development of an artificial neural network for estimation of formation permeability," in *Proceedings of the Petroleum Computer Conference*, SPE 28237, pp. 147–154, Dallas, Tex, USA, July–August 1994.
- [6] S. Mohaghegh, B. Balan, and S. Ameri, "State-of-the-art in permeability determination from well log data: part 2—verifiable, accurate permeability predictions, the touch-stone of all models," in *Proceedings of the Eastern Regional Conference*, SPE 30979, pp. 43–47, September 1995.
- [7] B. Balan, S. Mohaghegh, and S. Ameri, "State-of-the-art in permeability determination from well log data, part 1, a comparative study, model development," in *Proceedings of the SPE Eastern Regional Conference and Exhibition*, SPE 30978, Morgantown, WV, USA, 1995.
- [8] S. Mohaghegh, S. Ameri, and K. Aminian, "A methodological approach for reservoir heterogeneity characterization using artificial neural networks," *Journal of Petroleum Science and Engineering*, vol. 16, pp. 263–274, 1996.
- [9] Z. Huang, J. Shimeld, M. Williamson, and J. Katsube, "Permeability prediction with artificial neural network modeling in the Venture gas field, offshore eastern Canada," *Geophysics*, vol. 61, no. 2, pp. 422–436, 1996.
- [10] Y. Zhang, H. A. Salisch, and C. Arns, "Permeability evaluation in a glauconite-rich formation in the Carnarvon Basin, Western Australia," *Geophysics*, vol. 65, no. 1, pp. 46–53, 2000.

- [11] P. M. Wong, M. Jang, S. Cho, and T. D. Gedeon, "Multiple permeability predictions using an observational learning algorithm," *Computers and Geosciences*, vol. 26, no. 8, pp. 907–913, 2000.
- [12] K. Aminian, H. I. Bilgesu, S. Ameri, and E. Gil, "Improving the Simulation of Waterflood Performance With the Use of Neural Networks," in *Proceedings of the SPE Eastern Regional Meeting*, SPE 65630, pp. 105–110, October 2000.
- [13] K. Aminian, B. Thomas, H. I. Bilgesu, S. Ameri, and A. Oyerokun, "Permeability distribution prediction," in *Proceedings of the SPE Eastern Regional Conference*, October 2001.
- [14] L. Rolon, *Developing intelligent synthetic logs: application to upper devonian units in PA*, M.S. thesis, West Virginia University, Morgantown, WV, 2004.
- [15] E. Artun, S. Mohaghegh, and J. Toro, "Reservoir characterization using intelligent seismic inversion," in *Proceedings of the SPE Eastern Regional Meeting*, SPE 98012, West Virginia University, Morgantown, WV, September 2005.
- [16] M. Stitsion, A. Gammerman, V. Vapnik, V. Vovk, C. Watkins, and J. Weston, *Advances in Kernel Methods—Support Vector Learning*, MIT Press, Cambridge, Mass, USA, 1999.
- [17] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [18] M. Behzad, K. Asghari, M. Eazi, and M. Palhang, "Generalization performance of support vector machines and neural networks in runoff modeling," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7624–7629, 2009.
- [19] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 2nd edition, 2000.
- [20] V. N. Vapnik, *Statistical Learning Theory*, Wiley, New York, NY, USA, 1998.
- [21] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, UK, 2000.
- [22] C. M. Bishop, *Pattern recognition and machine learning*, Springer, New York, NY, USA, 2006.
- [23] C. Cortes, *Prediction of generalization ability in learning machines*, Ph.D. thesis, University of Rochester, Rochester, NY, USA, 1995.
- [24] L. Wang, *Support Vector Machines: Theory and Applications*, Springer, Berlin, Germany, 2005.
- [25] M. Martinez-Ramon and C. Cristodoulou, *Support Vector Machines for Antenna Array Processing and Electromagnetic*, Universidad Carlos III de Madrid, Morgan & Claypool, San Rafael, Calif, USA, 2006.
- [26] D. Zhou, B. Xiao, and H. Zhou, "Global geometric of SVM classifiers," Tech. Rep., Institute of automation, Chinese Academy of Sciences, AI Lab, 2002.
- [27] K. P. Bennett and E. J. Brendensteiner, *Geometry in Learning, Geometry at Work*, Mathematical Association of America, Washington, DC, USA, 1998.
- [28] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "A fast iterative nearest point algorithm for support vector machine classifier design," *IEEE Transactions on Neural Networks*, vol. 11, no. 1, pp. 124–136, 2000.
- [29] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear prediction of chaotic time series using support vector machines," in *Proceedings of the 7th IEEE Workshop on Neural Networks for Signal Processing (NNSP '97)*, pp. 511–520, Amelia Island, Fla, USA, September 1997.
- [30] J. T. Jeng, C. C. Chuang, and S. F. Su, "Support vector interval regression networks for interval regression analysis," *Fuzzy Sets and Systems*, vol. 138, no. 2, pp. 283–300, 2003.
- [31] K. K. Seo, "An application of one-class support vector machines in content-based image retrieval," *Expert Systems with Applications*, vol. 33, no. 2, pp. 491–498, 2007.
- [32] K. Trontl, T. Šmuc, and D. Pevec, "Support vector regression model for the estimation of  $\gamma$ -ray buildup factors for multi-layer shields," *Annals of Nuclear Energy*, vol. 34, no. 12, pp. 939–952, 2007.
- [33] A. Widodo and B. S. Yang, "Wavelet support vector machine for induction machine fault diagnosis based on transient current signal," *Expert Systems with Applications*, vol. 35, no. 1-2, pp. 307–316, 2008.
- [34] R. Burbidge, M. Trotter, B. Buxton, and S. Holden, "Drug design by machine learning: support vector machines for pharmaceutical data analysis," *Computers and Chemistry*, vol. 26, no. 1, pp. 5–14, 2001.
- [35] G. Valentini, "Gene expression data analysis of human lymphoma using support vector machines and output coding ensembles," *Artificial Intelligence in Medicine*, vol. 26, no. 3, pp. 281–304, 2002.
- [36] S. Tripathi, V. V. Srinivas, and R. S. Nanjundiah, "Downscaling of precipitation for climate change scenarios: a support vector machine approach," *Journal of Hydrology*, vol. 330, no. 3-4, pp. 621–640, 2006.
- [37] C. Sanchez-Hernandez, D. S. Boyd, and G. M. Foody, "Mapping specific habitats from remotely sensed imagery: support vector machine and support vector data description based classification of coastal saltmarsh habitats," *Ecological Informatics*, vol. 2, no. 2, pp. 83–88, 2007.

- [38] C. Z. Cai, W. L. Wang, L. Z. Sun, and Y. Z. Chen, "Protein function classification via support vector machine approach," *Mathematical Biosciences*, vol. 185, no. 2, pp. 111–122, 2003.
- [39] F. E. H. Tay and L. J. Cao, "Modified support vector machines in financial time series forecasting," *Neurocomputing*, vol. 48, pp. 847–861, 2002.
- [40] I. Steinwart, *Support Vector Machines*. Los Alamos National Laboratory, Information Sciences Group (CCS-3), Springer, New York, NY, USA, 2008.
- [41] M. Saemi, M. Ahmadi, and A. Y. Varjani, "Design of neural networks using genetic algorithm for the permeability estimation of the reservoir," *Journal of Petroleum Science and Engineering*, vol. 59, no. 1-2, pp. 97–105, 2007.
- [42] H. Liu, X. Yao, R. Zhang, M. Liu, Z. Hu, and B. Fan, "The accurate QSPR models to predict the bio-concentration factors of nonionic organic compounds based on the heuristic method and support vector machine," *Chemosphere*, vol. 63, no. 5, pp. 722–733, 2006.
- [43] T. W. Lee, *Independent Component Analysis*, Theory and Applications Kluwer Academic Publishers, 1998.
- [44] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.
- [45] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.
- [46] A. Hyvärinen, "Survey on independent component analysis," *Neural Computing Surveys*, vol. 2, pp. 94–128, 1999.
- [47] A. J. Bell and T. J. Sejnowski, "The "independent components" of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, 1997.
- [48] Q. A. Tran, X. Li, and H. Duan, "Efficient performance estimate for one-class support vector machine," *Pattern Recognition Letters*, vol. 26, no. 8, pp. 1174–1182, 2005.
- [49] S. Merler and G. Jurman, "Terminated Ramp-support vector machines: a nonparametric data dependent kernel," *Neural Networks*, vol. 19, no. 10, pp. 1597–1611, 2006.
- [50] H. Liu, S. Wen, W. Li, C. Xu, and C. Hu, "Study on identification of oil/gas and water zones in geological logging base on support-vector machine," *Fuzzy Information and Engineering*, vol. 2, pp. 849–857, 2009.
- [51] Q. Li, L. Jiao, and Y. Hao, "Adaptive simplification of solution for support vector machine," *Pattern Recognition*, vol. 40, no. 3, pp. 972–980, 2007.
- [52] H.-J. Lin and J. P. Yeh, "Optimal reduction of solutions for support vector machines," *Applied Mathematics and Computation*, vol. 214, no. 2, pp. 329–335, 2009.
- [53] E. Eryarsoy, G. J. Koehler, and H. Aytug, "Using domain-specific knowledge in generalization error bounds for support vector machine learning," *Decision Support Systems*, vol. 46, no. 2, pp. 481–491, 2009.
- [54] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [55] B. Walczak and D. L. Massart, "The radial basis functions—partial least squares approach as a flexible non-linear regression technique," *Analytica Chimica Acta*, vol. 331, no. 3, pp. 177–185, 1996.
- [56] R. Rosipal and L. J. Trejo, "Kernel partial least squares regression in reproducing kernel Hilbert space," *Journal of Machine Learning Research*, vol. 2, pp. 97–123, 2004.
- [57] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K. R. Müller, "Fisher discriminant analysis with kernels," in *Proceedings of the 9th IEEE Workshop on Neural Networks for Signal Processing (NNSP '99)*, pp. 41–48, August 1999.
- [58] B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, Mass, USA, 2002.
- [59] S. R. Gunn, "Support vector machines for classification and regression," Tech. Rep., Image Speech and Intelligent Systems Research Group, University of Southampton, Southampton, UK, 1997.
- [60] C. H. Wu, G. H. Tzeng, and R. H. Lin, "A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4725–4735, 2009.
- [61] V. D. A. Sánchez, "Advanced support vector machines and kernel methods," *Neurocomputing*, vol. 55, no. 1-2, pp. 5–20, 2003.
- [62] Donald F. Specht, "A general regression neural network," *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568–576, 1991.
- [63] Y. B. Dibike, S. Velickov, D. Solomatine, and M. B. Abbott, "Model induction with support vector machines: introduction and applications," *Journal of Computing in Civil Engineering*, vol. 15, no. 3, pp. 208–216, 2001.

- [64] W. Wang, Z. Xu, W. Lu, and X. Zhang, "Determination of the spread parameter in the Gaussian kernel for classification and regression," *Neurocomputing*, vol. 55, no. 3-4, pp. 643-663, 2003.
- [65] D. Han and I. Cluckie, "Support vector machines identification for runoff modeling," in *Proceedings of the 6th International Conference on Hydroinformatics*, S. Y. Liong, K. K. Phoon, and V. Babovic, Eds., pp. 21-24, Singapore, June 2004.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

