APPLICABILITY OF STATISTICAL LEARNING ALGORITHMS IN GROUNDWATER QUALITY MODELING

Abedalrazq Khalil\textsuperscript{1,‡}, Mohammad N. Almasri\textsuperscript{2}, Mac McKee\textsuperscript{1}, and Jagath J. Kaluarachchi\textsuperscript{1}

\textsuperscript{1}Department of Civil and Environmental Engineering
Utah Water Research Laboratory
Utah State University
Logan, Utah 84322-8200
USA

\textsuperscript{2}Water and Environmental Studies Institute
An-Najah National University
Nablus
Palestine

July 2004

\textsuperscript{‡} Corresponding author – Graduate Assistant [akhalil@cc.usu.edu, Tel: (435) 797-7176, Fax: (435) 797-3663]
Four algorithms are outlined, each of which has interesting features for predicting contaminant levels in groundwater. Artificial neural networks (ANN), support vector machines (SVM), locally weighted projection regression (LWPR), and relevance vector machines (RVM) are utilized as surrogates for a relatively complex and time-consuming mathematical model to simulate nitrate concentration in groundwater at specified receptors. Nitrates in the application reported in this paper are due to on-ground loadings from fertilizers and manures. The practicability of the four learning machines in this application is demonstrated for an agriculture-dominated watershed where nitrate contamination exceeds the maximum allowable contaminant level at many locations. Cross-validation and bootstrapping techniques are used for both training and performance evaluation. Prediction results of the four learning machines are rigorously assessed using different efficiency measures to ensure their generalization ability. Prediction results show the ability of learning machines to build accurate models with strong predictive capabilities and, hence, constitute a valuable means for saving effort in groundwater contaminant modeling and improving modeling performance.

Keywords: nitrate, contamination, groundwater, modeling, statistical learning theory, predictive learning.
1. INTRODUCTION

Groundwater provides one-third of the world’s drinking water. Since surface water is largely allocated, demand on the finite groundwater resources is increasing. However, groundwater is highly susceptible to contamination. This vulnerability poses serious threat to the environment and can limit the value of the resource to society as a whole. Groundwater can be contaminated by localized releases from waste disposal sites, landfills, and underground storage tanks. Pesticides, fertilizers, salt water intrusion, and contaminants from other nonpoint source pollutants are also major sources of groundwater pollution (CGER, 1993).

Recognition of groundwater contamination problems and the growing demand for quality water has generated a need for powerful quantitative predictive models that are reliable, accurate, and resilient against uncertainty. Such models must have high predictive capability to be utilized in mitigating groundwater contamination. Process-based contaminant transport simulations rely on solving the advection-dispersion-reaction governing equation (Atmadja and Bagtzoglou, 2001). This simulation entails a full understanding of the underlying physics controlling advection, dispersion, retardation, hydrodynamic, and chemical behavior. The utility of such models is constrained by their limited predictive power. Moreover, their reliability can be diminished by the paucity of data on aquifer structure, heuristic assumptions, and limited information for model validation. In addition, such models are generally computationally expensive (Hassan and Hamed, 2001; Wagner, 1992; Kunstmann et al., 2002).
To overcome these limitations, researchers have sometimes utilized approximation tools as surrogate for the mathematical models. These tools are characterized by their ability to quickly capture the underlying physics and provide predictions of system behavior. Many researchers have used learning machines, such as artificial neural networks (ANN), as surrogates for the mathematical model. The advantage of an ANN is that it does not require knowledge of the mathematical form of the relationship between the inputs and corresponding outputs. As a successful pattern recognition algorithm, ANNs have been utilized to “learn” to accurately mimic the behavior of a solute transport model so that it can be later employed in an optimization framework for remediation purposes (Rogers and Dowla, 1994; Rogers et al., 1995). Aziz and Wong (1992) further used ANNs to estimate aquifer parameters from pumping-test drawdown records. Morshed and Kaluarachchi (1998b) estimated saturated hydraulic conductivity and other parameters in the problem of free product migration and recovery using ANNs. Readers interested in ANN approximations are referred to ASCE Task Committee (2000a, b) and Maier and Dandy (2000).

ANNs have been combined with genetic search algorithms to dramatically accelerate the search process in groundwater optimization models. Primarily, ANNs are used to expedite the process of calculating the objective function in groundwater management and optimization problems (Rogers and Dowla, 1994; Rogers et al., 1995; Morshed and Kaluarachchi, 1998a, b; Aly and Peralta, 1999; Johnson and Rogers, 2000; Almasri, 2003). For instance, Rogers et al. (1995) demonstrated that an ANN was approximately $1.8 \times 10^7$ times faster than the groundwater flow and contaminant transport code used in their study. However, the ASCE Task Committee (2000b) concluded that
vigilance must be exercised when applying this combination. This caution stems in part from the potential for ANNs to fail to generalize well when trained with limited data.

In addition to the application of ANNs, the past decade has witnessed a growing advancement in data-driven modeling through the development of intelligent systems. Again, such systems “evolve” or “learn” reliable models using empirical records and qualitative physics that characterize the input-output behavior of physical phenomena. The intelligent systems approaches provide methods for flexible estimation (or “learning”) with limited data to achieve high levels of generalization and prediction accuracy. Among these approaches is a new learning methodology called support vector machines (SVMs), which were developed for such learning objectives (Vapnik, 1995). SVMs rely on the statistical learning theory (SLT) known as Vapnik-Chervonenkis theory (Vapnik, 1982, 1995, 1998). SVMs are now receiving enthusiastic attention similar to that of ANNs when they were first introduced, and are becoming an active field of machine learning research. Good prediction results have been reported in many SVM applications. For example, upon using SVMs for feature classification of digital remote sensing data and prediction of horizontal forces on a vertical breakwater, Dibike et al. (2001) concluded that SVMs produced results to comparable those of ANNs. However, the use of SVMs is expected to surpass ANN applications due to their superior performance in many problems that is due to its generalization capability ()

High dimensionality of the input space is often a serious problem associated with learning machines. A large training set that is able to provide a good distribution of high dimensional data is essential for successful learning. Locally weighted projection regression (LWPR) is an incremental nonparametric learning machine (not memory-
based) that uses special projection regression techniques to deal efficiently with high
dimensional spaces (Vijayakumar and Schaal, 2000a, b). LWPR is numerically robust
and of linear computational complexity in the number of input dimensions. The key
feature of the LWPR algorithm is the use of a spatially, locally nonlinear function
approximation for high dimensional input data that have redundant and irrelevant
components (Vijayakumar and Schaal, 2000a, b; Schaal et al., 2002). LWPR has shown
remarkable success in real-time robot learning and has outperformed models based on
simulation of the physical processes (Schaal et al., 2002). The robust incremental nature
of LWPR could be employed to handle the concerns of the ASCE Task Committee
(2000b) about the inability of ANNs to predict when the scope of the problem changes in
the context of a dynamic system. Thus, the motivation behind exploring LWPR models
originates from their suitability to operate in real time, and their resilience against
negative inference when new data are presented (Atkenson et al., 1997).

The absence of probabilistic outputs that provide estimates of the confidence and
reliability of the model predictions has led to the development of another learning
machine called the relevance vector machine (Tipping, 2001). Relevance vector machines
(RVM) address the uncertainty in both data and parameters that plague most of the
groundwater quality models (Kunstmann et al., 2002), for example, in an efficient and
effective manner. RVMs rely on the Bayesian concept and utilize an inductive modeling
procedure that allows incorporation of prior knowledge in the estimation process
(Tipping, 2000). The structure of the RVM model is identified parsimoniously and has
the potential for broad applications. The key features of RVMs are their good
generalization accuracy and sparse formulation. State-of-the-art prediction results have been reported in many applications where RVMs have been used (Li et al., 2002).

SVMs, LWPRs, and RVMs have not been previously utilized in groundwater related studies to mimic physically based relationships in the simulation of the fate and transport of contaminants in groundwater. The objective of this paper is to introduce several learning machines and examine their ability to produce models that can be effectively used to reduce the cost and complexity of transport simulation.

2. THEORETICAL BACKGROUND

The general pattern recognition problem can be described as follows. A learning machine is given a set, $D$, of $M$ training pairs of data, $[x_i, y_i], i = 1, \ldots, M$. The data training pairs are independent and identically distributed (i.i.d.) and consist of an $N$-dimensional vector, $x \in \mathbb{R}^N$, and the response or output, $y \in \mathbb{R}$. The goal of the learning machine, then, is to estimate an unknown continuous, real-valued function, $f(x)$ that makes accurate predictions of outputs, $y$, for previously unseen values of $x$.

2.1 Artificial Neural Networks

ANNs present an information-processing paradigm for pattern recognition (McCulloch and Pitts, 1943). ANNs use input-output response patterns to approximate the underlying governing rules of the output responses corresponding to specific inputs in a convoluted physical space (Morshed and Kaluarachchi, 1998b). The objective of the training process for ANNs is to calculate the optimal weights of the links in the neural net by minimizing the overall prediction error. This is known as empirical risk minimization.
In this work, ANNs are trained using the back-propagation algorithm (BPA) as developed by Rumelhart et al. (1986). For a detailed illustration of ANN functionality, the interested reader may refer to Maier and Dandy (2000), Kecman (2001), and Haykin (1999).

2.2 Support Vector Machines

SVMs represent a machine-learning model where prediction error and model complexity are simultaneously minimized. Unlike ANNs, the SVM structure is not fixed in advance with a specific number of adjustable parameters, but can adapt with data. Introduced by Vapnik (1995), the basic idea behind SVMs is mapping the input space into a high-dimensional feature space utilizing kernels (Vapnik, 1995). This so-called “kernel-trick” enables the SVM to work with feature spaces having very high dimensions. SVMs generally result in a function estimation equation analogous to the following form:

\[ f(x; w) = \sum_{i=1}^{m} w_i \times \phi_i(x) + w_0 \]  

(1)

where the functions \( \{\phi_i(x)\}_{i=1}^{m} \) are feature space representations of the input query \( x \), \( m \) is the number of patterns that contain all the information necessary to solve a given learning task, hereinafter referred to as support vectors, and \( w = \{w_0, w_1 \ldots w_m\} \) are the SVM parameters. The mapping of \( x \) by \( \phi(x) \) into a higher dimensional feature space is chosen in advance by selecting a suitable kernel function that satisfies Mercer’s conditions (Vapnik, 1995, 1998). By performing such a mapping, the learning algorithm seeks to obtain a hyperplane that is necessary for applying the linear regression in the SVM formulation (Kecman, 2001). Now the problem is to determine \( w \) and the corresponding \( m \) support vectors from the training data. To avoid the use of empirical
Vapnik (1995) proposed a structural risk minimization (SRM) in which one minimizes some empirical risk measure regularized by a capacity term. SRM is a novel inductive rule for learning from a finite data set and has shown good performance with small samples (Kecman, 2001). This is the most appealing advantage of SVMs, especially when data scarcity is a limitation on the use of process-based models or ANNs in groundwater quality modeling (ASCE Task Committee, 2000b; Kunstmann et al., 2002).

In line with SRM, therefore, the objective function of SVM is to minimize the following:

$$E(w) = \frac{1}{M} \sum_{i=1}^{M} |y_i - f(x_i, w)|_\varepsilon + \|w\|^2$$  \hspace{1cm} (2)

Vapnik (1995) employed the $\varepsilon$-insensitive loss function, $|y_i - f(x_i, w)|_\varepsilon$, where the difference between estimated output, $f(x_i, w)$, and the observed output, $y_i$, lies in the range of $\pm \varepsilon$ do not contribute to the output error. The $\varepsilon$-insensitive loss function is defined as:

$$|e|_\varepsilon = \begin{cases} 0 & \text{if } |e| < \varepsilon \\ |e| - \varepsilon & \text{if } |e| > \varepsilon \end{cases}$$  \hspace{1cm} (3)

Vapnik (1995) has shown that Equation (2) is equivalent to the following dual form:

$$\hat{y} = f(x, \alpha^*, \alpha) = \sum_{i=1}^{M} (\alpha_i^* - \alpha_i)K(x_i, x) + \lambda_o$$  \hspace{1cm} (4)

where the Lagrange multipliers $\alpha_i$ and $\alpha_i^*$ are required to be greater than zero for $i = 1, \ldots, M$, and $K(x_i, x)$ is a kernel function defined as an inner product in the feature space, $K(x_i, x) = \sum_{i=1}^{m} \phi(x_i) \cdot \phi(x)$. Typically, the optimal parameters of Equation (4) are found by solving its dual formulation:
\[
\begin{align*}
\min J_D(\alpha^*, \alpha) &= E \sum_{i=1}^{M} (\alpha_i^* + \alpha_i) - \sum_{i=1}^{M} y_i (\alpha_i^* + \alpha_i) + \\
&\frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} (\alpha_i^* + \alpha_i)(\alpha_j^* + \alpha_j)K(x_i, x_j) \\
such that \sum_{i=1}^{M} (\alpha_i^* + \alpha_i) &= 0 \\
\alpha_i, \alpha_i^* &\in [0, c] \forall i
\end{align*}
\] (5)

The parameter \( c \) is a user-defined constant that stands for the trade-off between model complexity and the approximation error. Equation (5) comprises a convex constrained quadratic programming problem (Vapnik, 1995, 1998). As a result, the input vectors that correspond to nonzero Lagrangian multipliers, \( \alpha_i \) and \( \alpha_i^* \), are considered as the support vectors. The SVM model thus formulated, then, is guaranteed to have a global, unique, and sparse solution. Despite the mathematical simplicity and elegance of SVM training, experiments prove they are able to deduce relationships of high complexity (Liong and Sivapragasam, 2002; Yu et al., 2004; Yu, 2004).

### 2.3 Relevance Vector Machines

RVMs adopt a Bayesian extension of learning. RVMs allow computation of the prediction intervals taking uncertainties of both the parameters and the data (Tipping, 2000). RVMs evade complexity by producing models that have structure and by a parameterization process that is appropriate to the information content of the data. RVMs have the identical functional form as SVMs, as in Equation (2), but using kernel terms, \( \{\phi(x)\}_{i=1}^{M} \equiv K(x, x_i) \), that correspond to nonlinear and fixed basis functions (Tipping, 2001). The RVM model seeks to forecast \( \hat{y} \) for any query \( x \) according to

\[
\hat{y} = f(x, w) + \epsilon_n, \text{ where } \epsilon_n \sim N(0, \sigma^2) \text{ and } w = (w_1, \ldots, w_M)^T \text{ is a vector of weights. The likelihood of the complete data set can be written as:}
\]
where $\Phi(x_i) = [1, K(x_i, x_1), K(x_i, x_2), \cdots, K(x_i, x_M)]^T$. Maximum likelihood estimation of $w$ and $\sigma^2$ in Equation (6) often results in severe overfitting. Therefore, Tipping (2001) recommended imposition of some prior constraints on the parameters, $w$, by adding a complexity penalty to the likelihood or the error function. This \textit{a priori} information controls the generalization ability of the learning system. Primarily, new higher-level hyperparameters are used to constrain \textit{an explicit} zero-mean Gaussian prior probability distribution over the weights, $w$ (Tipping, 2000):

$$p(w \mid \alpha) = \prod_{i=0}^{N} \mathcal{N}(w_i \mid 0, \alpha_i^{-1})$$  \hfill (7)

where $\alpha$ is a hyperparameter vector that controls how far from zero each weight is allowed to deviate (Schölkopf and Smola, 2002). For completion of hierarchical prior specifications, hyperpriors over $\alpha$ and the noise variance, $\sigma^2$, are defined. Consequently, using Bayes’ rule, the posterior over all unknowns could be computed given the defined noninformative prior distributions:

$$p(w, \sigma^2 \mid y) = \frac{p(y \mid w, \sigma^2).p(w, \sigma^2)}{\int p(y \mid w, \sigma^2).p(w, \sigma^2) \, dw \, d\sigma^2}$$  \hfill (8)

The analytical solution of the posterior in Equation (8) is intractable. Thus, decomposition of the posterior according to $p(w, \sigma^2 \mid y) = p(w \mid y, \sigma^2)p(\sigma^2 \mid y)$ is used to facilitate the solution (Tipping, 2001). The posterior distribution of the weights is:

$$p(w \mid y, \sigma^2) = \frac{p(y \mid w, \sigma^2).p(w \mid \alpha)}{p(y \mid \alpha, \sigma^2)}$$  \hfill (9)
This has an analytical solution where the posterior covariance and mean are, respectively,

\[ \Sigma = (\sigma \Phi^T \Phi + A)^{-1}, \]

with \( A = \text{diag}(\alpha_1, \alpha_2, \cdots, \alpha_{N_y}) \), and \( \mu = \Sigma \Phi^T \sigma^{-2} I_y t \) where \( I \) is the identity matrix. Therefore, learning becomes a search for the hyperparameter posterior most probable, i.e., the maximization of

\[ p(a, \sigma^2 \mid y) \propto p(y \mid a, \sigma^2) p(a) p(\sigma^2) \text{ with respect to } a \text{ and } \sigma^2. \]

For uniform hyperpriors over \( a \) and \( \sigma^2 \), one need only to maximize the term \( p(y \mid a, \sigma^2) \):

\[
p(y \mid a, \sigma^2) = \int p(y \mid w, \sigma^2) p(w \mid a) \, dw
\]

\[
= \left( \frac{1}{2\pi} \right)^{-N_y/2} \sqrt{\sigma^2 I + \Phi A^{-1} \Phi^T} \exp \left\{ -\frac{1}{2} y^T (\sigma I + \Phi A^{-1} \Phi^T)^{-1} y \right\}
\]

In related Bayesian models, Equation (10) is known as the marginal likelihood, and its maximization is known as the type II-maximum likelihood method (Berger, 1985; Wahba, 1985). MacKay (2003) refers to this term as the “evidence for hyperparameter” and its maximization as the “evidence procedure.” Hyperparameter estimation is carried out in iterative formulae, e.g., gradient descent on the objective function (Tipping, 2001; MacKay, 2003).

The evidence of the data allows the posterior probability distribution to concentrate at very large values of \( a \). Respectively, the posterior probability of the associated weight will be concentrated at zero. Therefore, one could consider the corresponding inputs irrelevant (Tipping, 2001). In other words, the outcome of this optimization is that many elements of \( a \) go to infinity such that \( w \) will have only a few nonzero weights that will be considered as relevant vectors. The relevant vectors (RV) can be viewed as counterparts to support vectors (SV) in SVMs; therefore, the resulting
model enjoys the properties of SVMs (i.e., sparsity and generalization) and, in addition, provides estimates of uncertainty bounds.

2.4 Locally Weighted Projection Regression

LWPR is a new algorithm that achieves a nonlinear function approximation in a high dimensional space that might have redundant input dimensions. LWPR is considered to be the first spatially localized incremental learning system that can efficiently work in high dimensional spaces (Vijayakumar and Schaal, 2000a). LWPR is embedded within a projection regression algorithm along with an incremental nonlinear function approximation. Projection regression (PR) was employed to cope with high dimensions through using single variate regressions along particular local projections in the input space to counter the curse of dimensionality. Local projection is used instead of global projection to accomplish local function approximation and to detect irrelevant input dimensions (Vijayakumar and Schaal, 2000b). Therefore, projection regression (PR) and function approximation are both utilized in LWPR. In PR algorithms, one seeks to spatially localize a linear function approximation along the desired projections. Partial least squares (PLS) is adopted here where one computes orthogonal projections of input data and consequently estimates a univariate regression along each component on the residuals of the previous step (Vijayakumar and Schaal, 2000a). Assume that the data are generated according to the standard linear regression model, \( y = \beta^T x + \epsilon \), where \( \epsilon \) represents white noise. In PLS projection regression, \( k \) orthogonal directions, \( u_1, \ldots, u_k \), are sought. Along each projection, finding the regression coefficient, \( \beta \), is found from linear regression. In the LWPR learning mechanism, weighing kernels, \( c \), that define the
locality are determined, each of which computes a weight \( w_{l,i} \), for each data point \((x_i, y_i)\). The estimated weight is a function of the distance of the query from the center of the weighing kernel \( \mathbf{c}_i \). For a Gaussian kernel, \( w_{l,i} \) is:

\[
w_{l,i} = \exp\left(\left(x_i - \mathbf{c}_i\right)^T \eta_k \left(x_i - \mathbf{c}_i\right)\right)
\]  

where \( \eta \) is the metric distance that determines the size and shape of the region of validity of the linear model, called the “receptive field”. For instance, in case of the \( L \) local linear models, to make a prediction for a given input vector \( \mathbf{x} \), each linear model must estimate a prediction \( y_{l,i} \), \( l = 1, \ldots, L \). Accordingly, the total output of the machine is a weighted mean of all linear models:

\[
\hat{y} = \frac{\sum_{l=1}^{L} w_l y_l}{\sum_{l=1}^{L} w_l}
\]  

Algorithmically, for a new training query \((\mathbf{x}, y)\), if no linear model is activated by more than a predefined threshold then a new receptive field is defined to be centered at that query. The metric distance \( \eta \) is of paramount importance to the concept of LWPR since it controls the validity of the local unit’s shape and size. Thus, optimizing such a parameter for each receptive field is necessary. Vijayakumar and Schaal (2000b) proposed to address this optimization problem through use of an incremental gradient descent algorithm based on a leave-one-out cross-validation criterion rather than the empirical error. Finally, the utility of LWPR in function approximation has been demonstrated in data sets of up to 50 dimensions and it has shown a very robust learning performance (Vijayakumar and Schaal, 2000a, b).
3. APPLICATIONS OF LEARNING MACHINES

The most pervasive groundwater contaminant is nitrate, which results from fertilizers and animal wastes (CGER, 1993). Agricultural practices, including fertilizer and manure applications, result in nonpoint source pollution of groundwater, and the effects of these practices accumulate over time (Schilling and Wolter, 2001). Hence, nitrate levels in groundwater have increased proportionally and concurrently with rises in fertilizer application (USDA, 1987; DeSimone and Howes, 1998). Identification of areas with heavy nitrogen loadings from nonpoint sources is important for land use planners and environmental regulators. Once such high-risk areas have been identified, preventative measures can be implemented to minimize the risk of nitrate leaching to groundwater (Lee, 1992; Tesoriero and Voss, 1997). The need to introduce alternatives to protect groundwater quality is of critical importance, especially in areas where groundwater is the sole source of drinking water and because of the high cost of mitigating contaminated groundwater (Tesoriero and Voss, 1997).

Aquifers can sustain a specific level of on-ground nitrogen applications without exceeding the maximum contaminant level (MCL). This sustainable loading, which might be considered the optimal loading, is a function of the on-ground nitrogen loadings from existing sources of nitrogen, nitrogen dynamics in the soil, the groundwater flow system, and the nitrate fate and transport processes in groundwater (see Figure 1). An optimization approach can be used to determine the sustainable loadings. In the optimization process, the objective function representing the sustainable loading is evaluated successively by executing the mathematical model depicted in Figure 1 to ultimately predict nitrate concentration in groundwater. The work reported in this paper is
motivated by the fact that the simulation of nitrate fate and transport in groundwater is a
time-consuming process when successive runs are needed in an optimization context or in
the assessment of management alternatives, especially when conducting a regional-scale
analysis for fine-resolution decision variables.

The following sections demonstrate the learning machines that have been
discussed. Pattern recognition is depicted through training, validation, and testing using
patterns generated from mathematical models of soil nitrogen dynamics and nitrate fate
and transport in groundwater. The resulting models are intended to capture the nitrogen
dynamics in the soil, the groundwater flow system, and the nitrate fate and transport
processes in groundwater (see Figure 1). Results are demonstrated and discussion is
provided to illustrate the predictive ability of the models. Comparison of prediction
efficiencies is made and conclusions are provided. Moreover, the practicability of these
learning machines is demonstrated through a case study of an actual regional aquifer in
an agriculture-dominated watershed.

3.1 Site Description

The Sumas-Blaine aquifer (see Figure 2) is located in the Nooksack watershed in
Whatcom County in the northwest corner of Washington State. The water table is mostly
shallow, typically less than 10 feet, but a few exceptions occur where the depth to the
water table ranges from 25 feet to 50 feet (Tooley and Erickson, 1996). Precipitation
ranges from over 60 inches per year in the northern uplands to about 40 inches per year in
the lowlands. Recharge to the aquifer is largely due to the infiltration of precipitation and
irrigation. The actual area considered in this work includes parts of Canada because there
is a substantial manure application on berry plantations located in the portions of the
watershed that lie in Canada. Since the groundwater flow is from north to south towards
the Nooksack River, the nitrogen-rich manure application in the Canadian side has a
major influence on groundwater quality in the south (Stasney, 2000; Mitchell et al.,
2003). The total area of the extended aquifer region is approximately 376 square miles
(Figure 2). There are 39 drainages representing the extended Sumas-Blaine aquifer
region. Due to the intensive agricultural activities in the study area (see Figure 2 for the
land cover distribution), groundwater quality in the aquifer has been continuously
degrad ing in recent decades and nitrate concentrations are increasing (Almasri and
Kaluarachchi, 2004b). Since the role of nitrate in eutrophication is well-recognized
(Wolfe and Patz, 2002), nitrate contamination of the surface water of the study area is a
concern as it greatly affects fish habitat. The transport of nitrate to surface water occurs
mainly via discharge of groundwater during baseflow conditions (Schilling and Wolter,
2001; Bachman et al., 2002). Therefore, the prevention of groundwater contamination
from nitrate also protects surface water quality.

3.2 Conceptualization of Nitrogen Transport

As depicted in Figure 1, the conceptual model of nitrate fate and transport in
groundwater includes (Almasri and Kaluarachchi, 2004a,c): (i) characterization of land
use cover to compute the spatial distribution of on-ground nitrogen loadings; (ii) detailed
assessment of all nitrogen sources in the study area and their allocation to the appropriate
land cover classes; (iii) simulation of the soil nitrogen dynamics; (iv) prediction of nitrate
leaching to groundwater; (v) modeling the groundwater flow system; and (vi) detailed
description of nitrate fate and transport processes in groundwater. In the next sections, a
general description of the integrated sub-systems is provided.

**On-Ground Nitrogen Loading** - A major step in calculating the amount of nitrate
leaching to groundwater is the estimation of the on-ground nitrogen loadings from
different nitrogen sources. There are many sources of nitrogen, natural and
anthropogenic, which can contribute to groundwater contamination (Hallberg and
Keeney, 1993). To differentiate between the different land application categories in order
to assign the appropriate nitrogen loadings, the national land cover data (NLCD) grid was
utilized in this study.

**Soil Nitrogen Dynamics** - The amount of nitrate found at any point in groundwater is the
product of various physical, chemical, and biological processes that are taking place in
the soil zone and groundwater (Johnsson et al., 2002). The major soil transformation
processes that greatly affect nitrate leaching are mineralization-immobilization,
nitrification, denitrification, and plant uptake (Addiscott et al., 1991). In addition, the soil
organic matter and crop residues influence the soil nitrogen content.

**Fate and Transport in Groundwater** - Many processes, including advection,
dispersion, and decay, can control the fate and transport of nitrate in groundwater.

Denitrification is the dominant chemical reaction that affects nitrate concentration in the
groundwater under anaerobic conditions (Frind et al., 1990; Postma et al., 1991; Korom,
1992; Tesoriero et al., 2000; Shamrukh et al., 2001). Denitrification can be expressed
using first-order kinetics with a first-order decay coefficient (Frind et al., 1990;
Shamrukh et al., 2001). Minerals rarely sorb nitrate because it is negatively charged. As a
result, it is highly mobile in mineral soils (Shamrukh et al., 2001).
Based on the above discussion, the long-term steady-state nitrate concentration distribution in groundwater can be expressed as a function of the soil and groundwater properties and other parameters that concurrently influence the nitrate concentration in groundwater, spatially and temporally. This illustrates the fundamental difficulty in the accurate modeling of fate and transport of nitrate in groundwater, especially at a regional scale.

### 3.3 Input and Predicted Output

The development of the learning machines requires the precise identification of the input and output vectors. Since the objective is to simulate the effect of on-ground nitrogen loadings from manure and fertilizers on nitrate concentrations in the groundwater at specified receptors, long-term nitrate concentrations, $C$, will be predicted according to the following formulation:

$$ C = f(\tau_F, \tau_M) $$  \hspace{1cm} (13)

where $\tau_F$ and $\tau_M$ are the on-ground nitrogen loadings from fertilizers and manure for each nitrate receptor. Although Equation (13) does not include all the applicable soil and groundwater properties and parameters, many studies have been successful in predicting the nitrate contamination of groundwater by considering only nitrogen loadings (Tesoriero and Voss, 1997; Nolan et al., 2002; Mitchell et al., 2003). Following this approach, machines in this work, the machines are used to predict the two-dimensional groundwater concentration distribution of nitrate only as a function of on-ground nitrogen loadings from manure and fertilizers.
3.4 Methodology

The conceptual model depicted in Figure 1 is applied to the study area to develop the input-output response patterns based on Equation (13). The models of on-ground nitrogen loadings and fate and transport of nitrate in the soil were developed by Almasri and Kaluarachchi (2004a, c), the groundwater flow model was developed by Kemblowski and Asefa (2003) using MODFLOW (Harbaugh and McDonald, 1996), and the model of nitrate fate and transport in groundwater was developed by Kaluarachchi and Almasri (2004) using MT3D.

Having estimated $\tau_F$ and $\tau_M$, the soil nitrogen model calculates the amount of nitrate leaching to groundwater and provides inputs to the nitrate fate and transport model, which in turn computes the corresponding $C$ vector at the specified receptors. Afterwards, the patterns of $C$ and $\tau_F$ and $\tau_M$ are allocated into training and testing sets and the learning machines are developed with the appropriate selection of machine parameters. A total of 56 nitrate receptors was selected, as depicted in Figure 3. The selected receptors have nitrate concentrations exceeding the MCL under current conditions. These receptors cover 14 selected drainages that contribute the majority of the on-ground nitrogen loadings in the study area. Such components of nitrogen loadings will comprise the inputs for the learning machines that is 28 inputs. Since the resulting models are to simulate the effect of managing fertilizer and manure applications on nitrate concentrations at the receptors depicted in Figure 3, two inputs are assigned for each drainage pertaining to fertilizer and manure loadings.
3.5 Learning Machines Construction

Obtaining an optimal level of performance for any learning machine entails a considerable number of design choices. The objectives of building optimal model architecture are to produce acceptable predictions and to assure generalization abilities. The approach of selecting an optimal architecture encompasses a rigorous statistical analysis and expert knowledge. Also, different models can be deduced given different data sets, which can further complicate the process of model selection. However, for successful model construction any training data set should carry enough idiosyncratic information about the processes involved. In this paper, 268 out of the available 440 patterns were randomly selected to develop the model specifications and structure. The justification for selecting 268 training patterns is that, as illustrated in Figure 4, no significant improvement in cross-validation error was achieved for greater numbers of patterns (see Results and Discussion section). The remaining 172 patterns were set aside for model validation. Intuitively, since training and testing sets were allocated randomly from the same domain (the pool of 440 patterns), they are likely to have similar information content and statistical significance. This should be expected to yield good performance of ANNs where overfitting is most likely to occur. For all the machines, input-output scaling is performed linearly using the minimum and maximum values of each input and output component.

The problem of choosing a suitable architecture for a multilayer perceptron (MLP) ANNs lies in specifying the type of activation function to be used and the number of neurons in the hidden layer. Four types of kernel functions —namely, polynomial kernel, radial basis function kernel, \( \text{sig}(\cdot) \), and \( \text{tanh}(\cdot) \) kernel—were used. For this case
study and data set trial-and-error analysis better performance was achieved with the sig(-) activation function. Upon producing the probability distribution function of the generalization error using cross-validation techniques, it was found that eight-hidden neurons produced an acceptable bias-variance trade-off. Different random initial weights may produce different training results, thus the training over the cross-validation sub-samples is performed at a fixed seed value.

Choosing a suitable kernel for both SVM and RVM models and receptive field shape for the LWPR is of paramount importance since these steps comprise the building blocks of the machines. While some authors recommend that the choice of kernel type and kernel parameters be done with knowledge of the underlying physical processes to be represented by the learning machine, in this study, a simple trial-and-error approach was used to select the type of kernel function for both the SVM and RVM models. For kernel parameter selection, cross-validation criteria were minimized over a specific range. The radial basis function, with a parameter value of 0.5, was used for the SVM model. The parameter \( \varepsilon \) and \( c \) had to be set to their optimal values during the model training. For a given data set proper \( \varepsilon \) and \( c \) selection ensure good generalization performance. The insensitive-error function parameter is largely selected to reflect the desired accuracy and could be optimally tuned to particular noise density and it was set at \( \varepsilon = 0.01 \) in this case study. Identification of the optimal value of the trade-off between model complexity and the approximation error was set at \( c = 1 \) (i.e., the tradeoff between an approximation error and model complexity) as a result of 10-fold cross-validation error. A Gaussian kernel function with width of 1.5 was used in the case of the RVM model, while in the LWPR analysis a Gaussian kernel was used, with the kernel metric distance optimized by
application of a gradient descent algorithm based on a leave-one-out cross-validation criterion. The RVM model was found to have the smallest number of parameters (e.g., only the kernel type and its width parameter). Netlab, a toolbox of Matlab® functions and scripts (Bishop, 1995; Nabney, 2001), was used for these analyses. For the SVM model, a Matlab interface to SVMlight, written by Schwaighofer (2004), was used. SVMlight is an implementation of Vapnik's support vector machine design (Vapnik, 1995). For development of the RVM and LWPR models, the Matlab implementation of Tipping (2001) and Vijayakumar and Schaal (2000a) was used.

To ensure good generalization of the inductive learning algorithm given scarce data, the machine performance was been tested on many bootstrap samples (i.e., 1000 bootstrap samples) from the original data set in order to explore the implications of the assumptions made about the nature of the data. This analysis provides a way to evaluate the significance of some indices and thus draw conclusions about model reliability. Using bootstrapping techniques, one can also deduce rough confidence bounds that are more revealing of model performance than single values (Willmott et al., 1985). Because of concerns about the underlying assumptions of each of the considered machines, rigorous model performance measures were performed to assess the capacity of each model (see Appendix I).

4. RESULTS AND DISCUSSION

While ANNs have been extensively employed in water resources (ASCE Task Committee, 2000a, b), the newer SVM, LWPR, and RVM approaches bring with them many potentially advantageous features, especially generalization performance and
sparse representation. It is with respect to these characteristics that the experimental
results on the performance of each machine are presented and discussed.

A widely advocated approach to the evaluation and comparison of inductive
learning machines involves training with known input-output data and then testing the
resulting machine against other data not used in training or validation.

There are 268 patterns used for model construction, specification, and training. To
support the selection of the number of patterns in the training set, Figure 4 was developed
and utilized. Specifically, the more examples that explain the underlying physics, the
better will be the predictability of the machine. Figure 4 provides information about the
number of data points required for the machine to have enough information about the
system (i.e., error becoming asymptotic as a function of the sample size). In the case of
utilizing more than 268 patterns, there is no significant contribution of additional data to
enhance the 5-fold cross-validation error as a measure of machine ability to generalize. In
other words, and according to Figure 4, about 39% of all samples in the data set can be
reserved for testing. It should be pointed out, however, that the recommended percentage
of samples for testing might be even higher for larger data sets. Good performance in the
testing phase is believed to be evidence for an algorithm’s practical plausibility and
provides an evaluation of the model’s predictive abilities. Achievement of this objective
is typically measured by the correlation coefficient, coefficient of efficiency, bias, root-
mean-square-error (RMSE), mean absolute error, and index of agreement. For more
details regarding these goodness-of-fit measures, the interested reader can refer to David
and Gregory (1999) and Willmott et al. (1985).
Table 1 presents the key statistics to evaluate the efficiency of the four learning machines in the training and testing phases. All the machines have higher performance in the training phase than in the testing phase. The loss of performance on the testing set addresses the machine susceptibility to the issue of overtraining. There is a noticeable reduction in performance on the testing data set (i.e., difference between machine performance on training and testing) for both the ANN and LWPR models. The small decline of performance on both RVM and SVM models indicates their ability to avoid overtraining and hence generalize well.

Figures 5 and 6 show scatter plots of predicted (from the learning machine) versus simulated (from the physical model) nitrate concentrations at two selected receptors. The results indicate that the four learning machines did provide good prediction performance. Figure 5 illustrates the prediction efficiency at the 19th receptor (see Figure 3). The SVM model shows the highest accuracy with a coefficient of efficiency of 0.866, followed by the RVM model at 0.864, the LWPR model at 0.837, and lastly the ANN model at 0.756. The SVM model shows an average underbias of 0.021, while the other machines show an overbias of 0.027, 0.031, and 0.037 for the RVM, LWPR, and ANN models, respectively. Figure 6 demonstrates the performance of the machines at the 34th receptor (see Figure 3). The RVM model has a coefficient of efficiency value of 0.993, followed by the SVM, ANN, and LWPR models with values of 0.988, 0.981, and 0.980, respectively. Again, the RVM model shows the lowest bias, followed by the ANN, SVM, and LWPR models. The ANN model experiences the highest variance as judged by a RMSE value of 0.113, while the lowest is for the RVM model with RMSE = 0.066.
Figure 7 shows the prediction performance of the four machines at each receptor in terms of RMSE. ANN performed the best for 25 receptors, while RVM performed the best for 19 followed by SVM for 12. As evaluated by the mean absolute bias, SVM performed the best for 21 receptors, ANN for 13 receptors, and RVM and LWPR for 11 receptors, each. From a bias-variance perspective, the ANN tends to produce a low variance but high bias. SVM produced the best unbiased machine, yet it showed high variance. A good tradeoff between bias and variance seems to be shown by the RVM for this application.

Figure 8 shows the coefficient of efficiency statistics for each receptor. The coefficient of efficiency represents an improvement over the coefficient of determination for model evaluation purposes in that it is sensitive to differences in the actual and model simulated means and variances (David and Gregory, 1999). For interpretation purposes for any machine, an efficiency coefficient of 0.9 indicates that the machine has a mean square error of 10 percent of the variance. The ANN model performed the best for 24 receptors, while RVM performed the best for 20, followed by SVM for 11 receptors and LWPR for only one receptor.

Table 2 provides empirical generalization estimates in terms of root-mean-square-error (RMSE) based on cross-validation and bootstrapping over scaled data. Linear scaling to [0, 1] is performed for mapping real world measurement to a range of values appropriate for model execution. Bootstrapping is useful in a situation where the underlying sampling distribution of the data and the parameters is unknown and difficult to estimate. Therefore, these statistics are mostly utilized for model selection purposes and model reliability evaluation (Willmott et al., 1985). The model selection procedure
focuses on selecting the optimal set of model hyper-parameters by minimizing bootstrapping or cross-validation estimates of the prediction error. For instance, the number of hidden nodes in the ANN model was obtained by minimizing the variance and the mean of the 10-fold cross-validation error. For development of the SVM model, the 10-fold cross-validation error was used to select the optimal trade-off, \( c \), between model complexity and the empirical risk. In their work with LWPR, Vijayakumar and Schaal (2000a) used the leave-one-out error estimates in the gradient descent algorithm in finding the metric parameters that specify the shape and region of validity of the receptive fields. One might notice that according to the hybrid bootstrap and 0.632+ estimator, the ANN model has significantly higher generalization capability than the other machines. However, the bootstrap estimates of the generalization error are optimistically biased which is evident in the case of the ANN model where overtraining results in a network that memorizes the individual examples rather than the trends in the data set. Besides using these statistics for model selection, one can also use them to provide confidence in the machine predictability, persistency, and robustness. As noticed in Table 2, the four machines produce almost similar generalization error.

The statistical results reported in Table 2 provide credible estimates of machine reliability and significance. The magnitude of the confidence interval for the accuracy measure of interest could be used as a measure of model reliability (Willmott et al., 1985). Principally, it is straightforward to estimate the confidence intervals of these statistics. The width of the bootstrapping confidence intervals indicates implicit uncertainty in the machine parameters. A wide confidence interval indicates that the available training data set is inadequate to find a robust parameter set (Kuan et al., 2003).
The RVM model shows the narrowest confidence bounds. For example in the case of hybrid bootstrap and 0.632+, the RVM model has $\text{RMSE} = 0.0232 \pm 0.000196$. The SVM model shows a 20 percent increase in the confidence interval width, and both the ANN and LWPR models show a 30 percent increase when compared to RVM. Owing to the nonincremental application of LWPR in the testing (validation) phase, it produces the lowest generalization performance. The use of LWPR is expected to be exceptional in problems that are highly dynamic and characterized by nonstationarity (i.e., streamflow predictions).

Degrees of freedom are often used as a model complexity measure in model selection criteria. An important aspect in machine learning and more specifically model selection is to avoid overparameterized models, or in other words, in accordance with Occam’s Razor, the most parsimonious model is the best (MacKay, 1992, 2003). While the ANN model requires a liberal number of parameters (i.e., linkage weights) to produce satisfactory results, the SVM and RVM models provide functional formulations that produce similar generalization abilities with many fewer degrees of freedom. According to Vapnik (1998), generalization from finite data is possible if and only if the estimator has limited capacity (i.e., enforced regularization).

The SVM model is characterized by a highly effective mechanism for avoiding overfitting that results in good generalization. The SVM formulation leads to a sparse model dependent only on a subset of training examples and their associated kernel functions (Vapnik, 1995). Tipping (2000) indicated that SVMs suffer from the absence of a probabilistic prediction capability that captures information about uncertainty and from the number of kernel functions that grows steeply with the size of the training data set.
from the necessity to manually tune some parameters, and from the selection of kernel function parameters (i.e., which also has to satisfy Mercer’s condition (Vapnik, 1995; Tipping, 2000)). Empirical results proved that RVMs are remarkable in producing an excellent generalization level while maintaining the sparsest structure. For example, the SVM utilized 120 patterns as support vectors out of the 268 patterns of the training set, while the RVM used only 26 patterns as relevance vectors, and LWPR used 40 receptive fields. However, the support vectors in the SVM model represent decision boundaries, while the RVM relevance vectors represent prototypical examples (Li et al., 2002). The prototypical examples exhibit the essential features of the information content and thus are able to transform the input data into the specified targets. This feature of both RVM and SVM could be further utilized to build up a sparse representation of the processes (e.g., monitoring network design).

5. SUMMARY AND CONCLUSIONS

The machine learning induction techniques examined here have shown the ability to build accurate models with strong predictive capabilities for groundwater quality and they offer a practical approach to some modeling difficulties encountered in water-related studies. Based on the evidence of the experiment, learning machines, other than ANNs, appear to be highly effective. The results of the analyses presented here show distinct performance preferences for each machine in a supervised-learning task. However, since the comparisons between the different learning machines were intended to be illustrative only, it should be strongly emphasized that no broader generalizations can be made about the superiority of any of the machines for all classes of problems. The complex nature of each of the learning algorithms that have been examined here makes it difficult to study
their statistical behavior in order to assess their performance objectively. Cross-validation techniques can be robust for tuning parameter selection because they make no assumptions about the data or noise distributions (Atkenson et al., 1997).

In the development of the models discussed here, significant effort is required to build the machine architecture. However, once developed and trained, the resulting models perform simulations in a small fraction of the time required by the process-based model. It can be concluded that learning machines could be confidently adopted as computationally efficient and sufficiently accurate substitutes for physical models in many applications. This feature is of great importance when conducting large numbers of consecutive model simulations, such as in an optimization context. Using traditional physically-based models, such simulations might be time-consuming to the extent that the entire process would be practically infeasible.

There are no criteria as when to use each of the presented machine other than to bear in mind that ANNs minimize only the empirical risk by finding an optimal set of weights for the chosen number of hidden nodes, while SVMs minimize the structural risk to achieve estimators that are less susceptible to overfitting, as evident by the results depicted in Table 1. Besides, owing to the quadratic optimization, SVMs are uniquely solvable and there is no need to train them in a repetitive manner. In contrast, ANNs require repeated training on the data set until a working model is attained. LWPR and RVM entail iterative solutions until some stopping criteria are achieved. In addition, SVMs achieve a global solution in the search for optimal parameter values and there is no need for trial-and-error procedures to determine the final machine architecture, which is directly obtained from the optimization solution. Also, ANNs rely heavily on the
structure of the networks, which is proven nontrivial and considered the most important drawback of ANNs (Liong and Sivapragasam, 2002). The choice of the number of hidden units in ANNs is problem-dependent and, therefore, it is difficult to determine a priori the optimal network configuration. However, the performance of SVMs and RVMs depend largely on the choice of kernel functions, which is in a sense equivalent to the choice of the ANN structure. One may resort to cascade correlation or pruning techniques to adjust the ANN structure to the complexity of the problem in an automatic way (Fahlman and Lebiere, 1990). Primarily, in this application, ANNs, SVMs, RVMs, and LWPRs all achieved their goal, namely pattern recognition in nitrate contamination occurrences in groundwater. The resulting models, once constructed, are many orders of magnitude faster than the process-based model. The comparison studies of learning machines mostly revolve around the fact that superiority in performance heavily depends on the problem in hand. In other words, there is a wide range of common applications that are of interest where one machine will be proffered choice over the others. Strictly speaking, an ANN prediction is more accurate in some problems, while SVM might be stronger in others. RVM is the strongest when uncertainty bounds are required, and LWPR is the most widely advocated in dynamic situations due to its incremental nature (e.g., when the input distribution of the training data changes over time).

One also has to keep in mind that ANNs and SVMs both suffer a decline in performance as the dimension of the data increase. Consequently, SVMs suffer from as many difficulties as ANNs and RVMs in finding the optimum solution when the size of the data set and/or the dimension of the input vector is large. When SVM is applied for solving large-size problems the computation time is prohibitively high. RVMs are
characterized by their ability to represent the information content of the data set without being degraded in terms of model complexity by an abundance of data yet it is also computationally exhaustive during the training. Both SVMs and RVMs exploit only the set of observations that contains all the information necessary for defining the final decision function.

ANNs, SVMs, and RVMs are global learning methods; however, many argue that they could be improved and applied in a much broader context if they could be localized by using locally weighted training criteria (Atkenson et al., 1997; Vapnik, 1992). The learning formalism in RVMs, SVMs, and LWPRs filters out noise. ANNs, if not well-trained, could learn the noise and hence result in overfitting.

In summary, this paper has surveyed four learning machines that could be viewed as powerful alternative approaches to process-based models in some applications. The advantages and disadvantages of learning machines have been presented in comparison to each other along with several statistical criteria for judging model performance. The authors agree with the popular No Free Lunch (NFL) theorem (Wolpert and Macready, 1995) and share the concern that “...for any algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class”. Similarly, quoting Magdon-Ismail (2000), “A learning algorithm that performs exceptionally well in certain situations will perform comparably poorly in other situations.” Essentially, the NFL theorem concludes that there is no learning algorithm that can be universally superior; therefore, one could fuse the advantageous features of the models in a “mixture experts system” (Jacobs et al., 1991; Jordan and Jacobs, 1994), which is a system that employs a set of experts trained independently on the same problem and thus benefits
from combining the recommendations of experts for making predictions. The outlook for
the use of learning machines in water resources research and applications is very
promising.
Model Performance

Various error estimation measures have been adopted to evaluate the accuracy of machine predictions, and this paper applies some of these error estimation methods, such as cross-validation and bootstrapping. These concepts of resampling are motivated by data scarcity. A validation test must be performed to evaluate the performance of an inductive learning algorithm to ensure good generalization capabilities. Since the true distribution of system inputs and outputs is unknown, it is necessary to estimate the generalization error. Using common notation (e.g., McLachlan, 1992; Shakhnarovich et al., 2001), an input data set, \( X = \{x_m\}_{m=1}^M = [x_1 \ldots x_M] \), will be referred to as \( X^{(m)} \) and its corresponding output set, or targets, is \( \{y_m\}_{m=1}^M \) where \( x \in \mathbb{R}^m \) and \( y \in \mathbb{R} \). The data set \( X^{(m)} \) is assumed to be i.i.d. and generated from a d-dimensional data space, \( D \), according to an unknown distribution, \( F \). The error function of any learning machine is denoted as:

\[
Q(x, X^{(m)}, A(X^{(m)})) = Q(x, X^{(m)})
\]  

(14)

where \( x \) is a random test point and \( A(X^{(m)}) \) is the set of hypotheses (a learning machine that assigns a prediction, \( \hat{y} \), to each \( x \) ) that have been produced by algorithm, \( A \), given a certain concept class over the training set \( X^{(m)} \) (Shakhnarovich et al., 2001). The conditional true error of a machine trained on \( X^{(m)} \) is:

\[
Err = Err(X^{(m)}, F) = E_{F(x)}[Q(x, X^{(n)})] = \int_D Q(x, X^{(n)}) dF(x)
\]  

(15)

The methods used for error estimation are as follows:
1. **Empirical error** $\overline{Err}$

A machine can be tested with the same data used for training. The empirical error (or redistribution error) results in an overoptimistic learning machine:

$$\overline{Err} = Q(X^{(m)}, X^{(m)})$$.

Again, this approach typically underestimates the true error and has a negative bias that is large for learning algorithms in which the susceptibility to overfitting is high (Shakhnarovich et al., 2001).

2. **Cross-validation and hold out**

Splitting the data into two sets, where the machine is trained on one set and tested on the other, to avoid underestimating the true error has a twofold disadvantage: (1) a problem of data reduction, and (2) statistical dependence between the two subsets (Blum et al. 1999; Shakhnarovich et al., 2001). The application of k-fold cross-validation is used to overcome these deficiencies. In using k-fold cross-validation, the data set is partitioned into k mutually disjointed folds (subsets) $S_j \forall j \in \{1,2,...,k\}$. For each $S_j$ the machine is trained on all folds except $S_j$. The final error is estimated as:

$$Err_{CV} = \frac{1}{k} \sum_{j=1}^{k} Q(S_j, X^{(m)}) \ , \ S_j \not\subset X^{(m)}$$

(16)

Leave-one-out-cross-validation error $Err_{CV,m}$ constitutes the extreme case where $k$ equals the number of training data sets $X^{(m)}$. Kohavi (1995) claimed that $Err_{CV,m}$ suffers from high variance estimates owing to the learning algorithm’s instability under small perturbations in data.

3. **Bootstrap error estimation**
Ordinary bootstrap estimator. This estimator is also called “naïve”. The algorithm is trained on $B$ set of bootstrap samples $X_b^{(m)}$, $b = 1, \cdots, B$, and tested on the original data set $X^{(m)}$ (Efron, 1992). The error, therefore, is calculated as:

$$Err_{BS} = \frac{1}{B} \sum_{b=1}^{B} Q(X^{(m)}, X_b^{(m)})$$ (17)

Intuitively, one should expect $Err_{BS}$ to be biased downward (Shakhnarovich et al., 2001).

Leave-one-out bootstrap. The learning machine quality can be evaluated using a number, $B$, of bootstrap samples $X_b^{(m-i)}$ that are drawn from the empirical distribution with the $i$-th sample, $x_i$, removed for testing (Efron and Tibshirani, 1997). The resulting error is:

$$Err_{BS}^{(i)} = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{B} \sum_{b=1}^{B} Q(X^{(m)}, X_b^{(m)})$$ (18)

Intuitively, as the number of samples increase, the error tends to decrease and thus upward bias is likely to occur.

Hybrid bootstrap and 0.632+. An estimator that minimizes the upward bias of $Err_{BS}^{(i)}$ is given by:

$$Err^2_h = \lambda Err_{BS}^{(i)} + (1 - \lambda)\overline{Err}$$ (19)

where $\lambda$ is a mixing parameter that is intended to minimize the bias. Davison and Hinkley (1998) reported that $\lambda = 0.632$ is the most favorable value and it is used to trade off between downward and upward bias. The probability that a test point $x_i$ will be included in the training bootstrap set $X_b^{(m)}$ is:

$$p(x_i \in X_b^{(m)}) = 1 - \left(1 - \frac{1}{m}\right)^m, \text{ and } p(x_i \in X_b^{(m)}) \approx 0.632 \quad \forall \ m \to \infty,$$ (20)
The 0.632+ estimator. This is a sophisticated estimator that accounts for the amount of overfitting and adjusts $\lambda$ accordingly. The relative overfitting rate, $\hat{R}$, is derived as

$$\hat{R} = \frac{Err_{BS}^{(1)} - \hat{\gamma} - Err}{Err_{BS} - \hat{\gamma} - Err},$$

where $\hat{\gamma}$ is the “no information error rate” which is the error rate of the learning machine when the data convey no information. It is given by:

$$\hat{\gamma} = m^{-2} \sum_{i=1}^{m} \sum_{j=1}^{m} Q(o_i, o_j, X^{(m)}).$$

For the no overfitting machine, $\hat{R} = 0$. The highest possible overfitting corresponds to $\hat{R} = 1$. The 0.632+ estimator is obtained as:

$$Err_{632+} = Err_{632} + \left(Err_{BS}^{(1)} - Err\right) \frac{.368 \times 632 \times \hat{R}}{1 - .368 \hat{R}}$$

where $Err_{632} = 0.632 Err_{BS} + (1 - 0.632) \overline{Err}$. For detail about these statistics, interested readers are referred to Shakhnarovich et al. (2001) and Efron and Tibshirani (1993).
6. REFERENCES


Blum A., A. Kalai, and J. Langford, 1999. Beating the holdout: Bounds for k-fold and
progressive cross-validation. Proceedings of the 12th Annual Conference on

vulnerability assessment: Predicting relative contamination potential under
conditions of uncertainty. National Academy Press, Washington, DC.

in hydrologic and hydroclimatic model validation. Water Resources Research,

Cambridge University Press.

receiving wastewater discharge: A mass balance approach. Water Resources

with support vector machines: introduction and applications. ASCE Journal of

York.


560.

Advances in Neural Information Processing Systems, 2, edited by D. S.
Touretzky, pp. 524-532, Morgan Kaufmann Publishers, Los Altos, CA.

Frind, E., W. Duynisveld, O. Strebel, and J. Boettcher, 1990. Modeling of
multicomponent transport with microbial transformation in groundwater: The

Regional ground-water quality. U.S. Geological Survey, Van Nostrand Reinhold,
New York.

Harbaugh, A.W., and M. G. McDonald, 1996. User's documentation for MODFLOW-96,
An update to the U.S. Geological Survey modular finite-difference ground-water


List of Tables

1. Key statistics for the prediction efficiency of the four learning machines in the training and testing phases (mean of the 56 receptors) .......................................................... 44

2. Different generalization performance measures for the four learning machines (scaled data). ......................................................................................................................... 45
Table 1. Key statistics for the prediction efficiency of the four learning machines in the training and testing phases (mean of the 56 receptors).

<table>
<thead>
<tr>
<th>Statistics</th>
<th>ANN Training</th>
<th>ANN Testing</th>
<th>SVM Training</th>
<th>SVM Testing</th>
<th>RVM Training</th>
<th>RVM Testing</th>
<th>LWPR Training</th>
<th>LWPR Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation coefficient</td>
<td>0.987</td>
<td>0.967</td>
<td>0.984</td>
<td>0.974</td>
<td>0.983</td>
<td>0.973</td>
<td>0.983</td>
<td>0.969</td>
</tr>
<tr>
<td>Coefficient of efficiency</td>
<td>0.974</td>
<td>0.933</td>
<td>0.966</td>
<td>0.948</td>
<td>0.966</td>
<td>0.946</td>
<td>0.966</td>
<td>0.911</td>
</tr>
<tr>
<td>Bias</td>
<td>0.000</td>
<td>0.021</td>
<td>-0.026</td>
<td>-0.004</td>
<td>0.000</td>
<td>0.015</td>
<td>0.000</td>
<td>-0.010</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.131</td>
<td>0.192</td>
<td>0.143</td>
<td>0.185</td>
<td>0.141</td>
<td>0.183</td>
<td>0.141</td>
<td>0.229</td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>0.085</td>
<td>0.131</td>
<td>0.074</td>
<td>0.115</td>
<td>0.095</td>
<td>0.128</td>
<td>0.095</td>
<td>0.172</td>
</tr>
<tr>
<td>Index of agreement</td>
<td>0.993</td>
<td>0.982</td>
<td>0.992</td>
<td>0.986</td>
<td>0.991</td>
<td>0.985</td>
<td>0.991</td>
<td>0.975</td>
</tr>
</tbody>
</table>
Table 2. Different generalization performance measures for the four learning machines (data scaled linearly to $[0, 1]$).

<table>
<thead>
<tr>
<th>Generalization Error (RMSE)</th>
<th>ANN</th>
<th>SVM</th>
<th>RVM</th>
<th>LWPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empirical error</td>
<td>0.0214</td>
<td>0.0210</td>
<td>0.0206</td>
<td>0.0216</td>
</tr>
<tr>
<td>5-fold cross-validation</td>
<td>0.0237</td>
<td>0.0267</td>
<td>0.0248</td>
<td>0.0244</td>
</tr>
<tr>
<td>10-fold cross-validation</td>
<td>0.0234</td>
<td>0.0262</td>
<td>0.0261</td>
<td>0.0250</td>
</tr>
<tr>
<td>Leave-one-out error</td>
<td>0.0231</td>
<td>0.0245</td>
<td>0.0269</td>
<td>0.0252</td>
</tr>
<tr>
<td>Ordinary bootstrap estimator</td>
<td>0.0222</td>
<td>0.0258</td>
<td>0.0242</td>
<td>0.0261</td>
</tr>
<tr>
<td>Leave-one-out bootstrap</td>
<td>0.0221</td>
<td>0.0256</td>
<td>0.0247</td>
<td>0.0259</td>
</tr>
<tr>
<td>Hybrid bootstrap and 0.632+</td>
<td>0.0218</td>
<td>0.0239</td>
<td>0.0232</td>
<td>0.0243</td>
</tr>
<tr>
<td>0.632 bootstrap</td>
<td>0.0218</td>
<td>0.0239</td>
<td>0.0232</td>
<td>0.0244</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1. Schematic representing the integrated modeling framework for simulating nitrate concentration in groundwater. ................................................................. 47

Figure 2. Layout of the model domain consisting of the extended Sumas-Blaine aquifer and land use classes. ................................................................. 48

Figure 3. The spatial distribution of the nitrate receptors in the study area. .......... 49

Figure 4. Variability of the 5-fold cross-validation RMSE with the number of data points for the four learning machines (scaled data). ........................................ 50

Figure 5. Scatterplot of the observed versus predicted nitrate concentrations at the 19th receptor for (a) ANN, (b) SVM, (c) RVM, and (d) LWPR. ............................ 51

Figure 6. Scatterplot of the observed versus predicted nitrate concentrations at the 34th receptor for (a) ANN, (b) SVM, (c) RVM, and (d) LWPR. ............................ 52

Figure 7. RMSE for the testing efficiency of the four learning machines for the 56 receptors. ........................................................................................................... 53

Figure 8. Coefficients of efficiency for the testing efficiency of the four learning machines for the 56 receptors. .......................................................... 54
Figure 1. Schematic of the integrated modeling framework for simulating nitrate concentration in groundwater.
Figure 2. Physical model domain, consisting of the extended Sumas-Blaine aquifer and land use classes.
Figure 3. The spatial distribution of the nitrate receptors in the study area.
Figure 4. Variability of the 5-fold cross-validation RMSE with the number of data points for the four learning machines (data scaled linearly to [0, 1]).
Figure 5. Scatterplot of the observed versus predicted nitrate concentrations at the 19th receptor for (a) ANN, (b) SVM, (c) RVM, and (d) LWPR.
Figure 6. Scatterplot of the observed versus predicted nitrate concentrations at the 34th receptor for (a) ANN, (b) SVM, (c) RVM, and (d) LWPR.
Figure 7. RMSE for the testing efficiency of the four learning machines for the 56 receptors.
Figure 8. Coefficients of efficiency for the testing efficiency of the four learning machines for the 56 receptors.