APPLICABILITY OF STATISTICAL LEARNING ALGORITHMS IN GROUNDWATER QUALITY MODELING

Abedalrazq Khalil[1,‡], Mohammad N. Almasri[2], Mac McKee[1], and Jagath J. Kaluarachchi[1]

[1]Department of Civil and Environmental Engineering
Utah Water Research Laboratory
Utah State University
Logan, Utah 84322-8200
USA

[2]Water and Environmental Studies Institute
An-Najah National University
Nablus
Palestine

July 2004

‡ Corresponding author – Graduate Assistant [akhalil@cc.usu.edu, Tel: (435) 797-7176, Fax: (435) 797-3663]

ABSTRACT

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 1.8×10^7 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-

122 based) that uses special projection regression techniques to deal efficiently with high
123 dimensional spaces (Vijayakumar and Schaal, 2000a, b). LWPR is numerically robust
124 and of linear computational complexity in the number of input dimensions. The key
125 feature of the LWPR algorithm is the use of a spatially, locally nonlinear function
126 approximation for high dimensional input data that have redundant and irrelevant
127 components (Vijayakumar and Schaal, 2000a, b; Schaal et al., 2002). LWPR has shown
128 remarkable success in real-time robot learning and has outperformed models based on
129 simulation of the physical processes (Schaal et al., 2002). The robust incremental nature
130 of LWPR could be employed to handle the concerns of the ASCE Task Committee
131 (2000b) about the inability of ANNs to predict when the scope of the problem changes in
132 the context of a dynamic system. Thus, the motivation behind exploring LWPR models
133 originates from their suitability to operate in real time, and their resilience against
134 negative inference when new data are presented (Atkenson et al., 1997).

135 The absence of probabilistic outputs that provide estimates of the confidence and
136 reliability of the model predictions has led to the development of another learning
137 machine called the relevance vector machine (Tipping, 2001). Relevance vector machines
138 (RVM) address the uncertainty in both data and parameters that plague most of the
139 groundwater quality models (Kunstmann et al., 2002), for example, in an efficient and
140 effective manner. RVMs rely on the Bayesian concept and utilize an inductive modeling
141 procedure that allows incorporation of prior knowledge in the estimation process
142 (Tipping, 2000). The structure of the RVM model is identified parsimoniously and has
143 the potential for broad applications. The key features of RVMs are their good

144 generalization accuracy and sparse formulation. State-of-the-art prediction results have
145 been reported in many applications where RVMs have been used (Li et al., 2002).

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

151 ## 2. THEORETICAL BACKGROUND

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

158 ### 2.1 Artificial Neural Networks

159 ANNs present an information-processing paradigm for pattern recognition
160 (McCulloch and Pitts, 1943). ANNs use input-output response patterns to approximate
161 the underlying governing rules of the output responses corresponding to specific inputs in
162 a convoluted physical space (Morshed and Kaluarachchi, 1998b). The objective of the
163 training process for ANNs is to calculate the optimal weights of the links in the neural net
164 by minimizing the overall prediction error. This is known as empirical risk minimization.

165 In this work, ANNs are trained using the back-propagation algorithm (BPA) as developed
166 by Rumelhart et al. (1986). For a detailed illustration of ANN functionality, the interested
167 reader may refer to Maier and Dandy (2000), Kecman (2001), and Haykin (1999).

168 **2.2 Support Vector Machines**

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

177 $$f(\mathbf{x};\mathbf{w}) = \sum_{i=1}^{m} w_i \times \phi_i(\mathbf{x}) + w_o \qquad (1)$$

178 where the functions $\{\phi_i(\mathbf{x})\}_{i=1}^{m}$ are feature space representations of the input query $\mathbf{x}$, m
179 is the number of patterns that contain all the information necessary to solve a given
180 learning task, hereinafter referred to as support vectors, and $\boldsymbol{w} = \{w_o\ w_1 \ldots w_m\}$ are the
181 SVM parameters. The mapping of $\mathbf{x}$ by $\phi(\mathbf{x})$ into a higher dimensional feature space is
182 chosen in advance by selecting a suitable kernel function that satisfies Mercer's
183 conditions (Vapnik, 1995, 1998). By performing such a mapping, the learning algorithm
184 seeks to obtain a hyperplane that is necessary for applying the linear regression in the
185 SVM formulation (Kecman, 2001). Now the problem is to determine $\boldsymbol{w}$ and the
186 corresponding m support vectors from the training data. To avoid the use of empirical

187 risk minimization (e.g., quadratic residual function), which may result in overfitting,
188 Vapnik (1995) proposed a structural risk minimization (SRM) in which one minimizes
189 some empirical risk measure regularized by a capacity term. SRM is a novel inductive
190 rule for learning from a finite data set and has shown good performance with small
191 samples (Kecman, 2001). This is the most appealing advantage of SVMs, especially
192 when data scarcity is a limitation on the use of process-based models or ANNs in
193 groundwater quality modeling (ASCE Task Committee, 2000b; Kunstmann et al., 2002).
194 In line with SRM, therefore, the objective function of SVM is to minimize the following:

195 $$E(\mathbf{w}) = \frac{1}{M}\sum_{i=1}^{M} | y_i - f(\mathbf{x}_i, \mathbf{w}) |_{\varepsilon} + \| \mathbf{w} \|^2 \quad (2)$$

196 Vapnik (1995) employed the ε-insensitive loss function, $| y_i - f(\mathbf{x}_i, \mathbf{w}) |_{\varepsilon}$, where
197 the difference between estimated output, $f(\mathbf{x}_i, \mathbf{w})$, and the observed output, y_i, lies in
198 the range of $\pm\varepsilon$ do not contribute to the output error. The ε-insensitive loss function is
199 defined as:

200 $$| e |_{\varepsilon} = \begin{cases} 0 & \text{if } |e| < \varepsilon \\ | e | - \varepsilon & \text{if } |e| > \varepsilon \end{cases} \quad (3)$$

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

202 $$\hat{y} = f(\mathbf{x}, \boldsymbol{\alpha}^*, \boldsymbol{\alpha}) = \sum_{i=1}^{M} (\alpha_i^* - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + \lambda_o \quad (4)$$

203 where the Lagrange multipliers α_i and α_i^* are required to be greater than zero for i = 1,
204 …, M, and $K(\mathbf{x}_i, \mathbf{x})$ is a kernel function defined as an inner product in the feature space,
205 $K(\mathbf{x}_i, \mathbf{x}) = \sum_{i=1}^{m} \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x})$. Typically, the optimal parameters of Equation (4) are found
206 by solving its dual formulation:

207
$$\left[\begin{array}{ll} \min\limits_{\alpha^*,\alpha} J_D(\alpha^*,\alpha) = \varepsilon\sum_{i=1}^{M}(\alpha_i^*+\alpha_i) - \sum_{i=1}^{M} y_i(\alpha_i^*+\alpha_i) + \\ \qquad\qquad \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M}(\alpha_i^*+\alpha_i)(\alpha_i^*+\alpha_i)K(\mathbf{x}_i,\mathbf{x}_j) \\ \text{such that} \quad \sum_{i=1}^{M}(\alpha_i^*+\alpha_i) = 0 \\ \qquad\qquad \alpha_i,\alpha_i^* \in [0,c], \forall_i \end{array}\right] \qquad (5)$$

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

216 **2.3 Relevance Vector Machines**

217 RVMs adopt a Bayesian extension of learning. RVMs allow computation of the
218 prediction intervals taking uncertainties of both the parameters and the data (Tipping,
219 2000). RVMs evade complexity by producing models that have structure and by a
220 parameterization process that is appropriate to the information content of the data. RVMs
221 have the identical functional form as SVMs, as in Equation (2), but using kernel terms,
222 $\{\phi_i(\mathbf{x})\}_{i=1}^{m} \equiv K(\mathbf{x},\mathbf{x}_i)$, that correspond to nonlinear and fixed basis functions (Tipping,
223 2001). The RVM model seeks to forecast $\hat{y}$ for any query $\mathbf{x}$ according to
224 $\hat{y} = f(\mathbf{x},\mathbf{w}) + \varepsilon_n$, where $\varepsilon_n \sim N(0,\sigma^2)$ and $\mathbf{w} = (w_o \ldots w_M)^T$ is a vector of weights. The
225 likelihood of the complete data set can be written as:

$$p(\mathbf{y} \mid \mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2} \| \mathbf{y} - \mathbf{\Phi w} \|^2\right\} \quad (6)$$

where $\mathbf{\Phi}(\mathbf{x}_i) = \left[1, K(\mathbf{x}_i, \mathbf{x}_1), K(\mathbf{x}_i, \mathbf{x}_2), \cdots, K(\mathbf{x}_i, \mathbf{x}_M)\right]^T$. Maximum likelihood estimation of $\mathbf{w}$ and σ^2 in Equation (6) often results in severe overfitting. Therefore, Tipping (2001) recommended imposition of some prior constraints on the parameters, $\mathbf{w}$, by adding a complexity penalty to the likelihood or the error function. This *a priori* information controls the generalization ability of the learning system. Primarily, new higher-level hyperparameters are used to constrain *an explicit* zero-mean Gaussian prior probability distribution over the weights, $\mathbf{w}$ (Tipping, 2000):

$$p(\mathbf{w} \mid \boldsymbol{\alpha}) = \prod_{i=0}^{N} \mathsf{N}(w_i \mid 0, \alpha_i^{-1}) \quad (7)$$

where $\boldsymbol{\alpha}$ 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 $\boldsymbol{\alpha}$ and the noise variance, σ^2, are defined. Consequently, using Bayes' rule, the posterior over all unknowns could be computed given the defined noninformative prior distributions:

$$p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2 \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{w}, \boldsymbol{\alpha}, \sigma^2).p(\mathbf{w}, \boldsymbol{\alpha}, \sigma)}{\int p(\mathbf{y} \mid \mathbf{w}, \boldsymbol{\alpha}, \sigma^2) p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2)\, d\mathbf{w}\, d\boldsymbol{\alpha}\, d\sigma^2} \quad (8)$$

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

$$p(\mathbf{w} \mid \mathbf{y}, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\mathbf{y} \mid \mathbf{w}, \sigma^2).p(\mathbf{w} \mid \boldsymbol{\alpha})}{p(\mathbf{y} \mid \boldsymbol{\alpha}, \sigma^2)} \quad (9)$$

This has an analytical solution where the posterior covariance and mean are, respectively, $\boldsymbol{\Sigma} = (\sigma \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \mathbf{A})^{-1}$, with $\mathbf{A} = diag(\alpha_1, \alpha_2, \cdots, \alpha_{N+1})$, and $\boldsymbol{\mu} = \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \sigma^{-2} \mathbf{I}_N \mathbf{t}$ where $\mathbf{I}$ is the identity matrix. Therefore, learning becomes a search for the hyperparameter posterior most probable, i.e., the maximization of $p(\boldsymbol{\alpha}, \sigma^2 \mid \mathbf{y}) \propto p(\mathbf{y} \mid \boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{\alpha}) p(\sigma^2)$ with respect to $\boldsymbol{\alpha}$ and σ^2 . For uniform hyperpriors over $\boldsymbol{\alpha}$ and σ^2 , one need only to maximize the term $p(\mathbf{y} \mid \boldsymbol{\alpha}, \sigma^2)$:

$$
\begin{aligned}
p(\mathbf{y} \mid \boldsymbol{\alpha}, \sigma^2) &= \int p(\mathbf{y} \mid \mathbf{w}, \sigma^2) p(\mathbf{w} \mid \boldsymbol{\alpha}) \, d\mathbf{w} \\
&= \left((2\pi)^{-N/2} \Big/ \sqrt{\left| \sigma^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^T \right|} \right) \exp\left\{ -\frac{1}{2} \mathbf{y}^T (\sigma \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^T)^{-1} \mathbf{y} \right\}
\end{aligned} \tag{10}
$$

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 $\boldsymbol{\alpha}$. 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 $\boldsymbol{\alpha}$ go to infinity such that $\mathbf{w}$ 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

266 model enjoys the properties of SVMs (i.e., sparsity and generalization) and, in addition,
267 provides estimates of uncertainty bounds.

268 **2.4 Locally Weighted Projection Regression**

269 LWPR is a new algorithm that achieves a nonlinear function approximation in a
270 high dimensional space that might have redundant input dimensions. LWPR is considered
271 to be the first spatially localized incremental learning system that can efficiently work in
272 high dimensional spaces (Vijayakumar and Schaal, 2000a). LWPR is embedded within a
273 projection regression algorithm along with an incremental nonlinear function
274 approximation. Projection regression (PR) was employed to cope with high dimensions
275 through using single variate regressions along particular local projections in the input
276 space to counter the curse of dimensionality. Local projection is used instead of global
277 projection to accomplish local function approximation and to detect irrelevant input
278 dimensions (Vijayakumar and Schaal, 2000b). Therefore, projection regression (PR) and
279 function approximation are both utilized in LWPR. In PR algorithms, one seeks to
280 spatially localize a linear function approximation along the desired projections. Partial
281 least squares (PLS) is adopted here where one computes orthogonal projections of input
282 data and consequently estimates a univariate regression along each component on the
283 residuals of the previous step (Vijayakumar and Schaal, 2000a). Assume that the data are
284 generated according to the standard linear regression model, $\mathbf{y} = \beta^T \times \mathbf{x} + \varepsilon$, where ε
285 represents white noise. In PLS projection regression, k orthogonal directions, $\mathbf{u}_1,....,\mathbf{u}_k$,
286 are sought. Along each projection, finding the regression coefficient, β, is found from
287 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 $(\mathbf{x}_i, y_i)$. The estimated weight is a function of the distance of the query from the center of the weighing kernel $\mathbf{c}_l$. For a Gaussian kernel, $w_{l,i}$ is:

$$w_{l,i} = \exp\left((\mathbf{x}_i - \mathbf{c}_l)^T \boldsymbol{\eta}_k (\mathbf{x}_i - \mathbf{c}_l)\right) \tag{11}$$

where $\boldsymbol{\eta}$ 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, $l = 1, \cdots, L$. Accordingly, the total output of the machine is a weighted mean of all linear models:

$$\hat{y} = \frac{\sum_{l=1}^{L} w_l y_l}{\sum_{l=1}^{L} w_l} \tag{12}$$

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 $\boldsymbol{\eta}$ 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

353 is a substantial manure application on berry plantations located in the portions of the
354 watershed that lie in Canada. Since the groundwater flow is from north to south towards
355 the Nooksack River, the nitrogen-rich manure application in the Canadian side has a
356 major influence on groundwater quality in the south (Stasney, 2000; Mitchell et al.,
357 2003). The total area of the extended aquifer region is approximately 376 square miles
358 (Figure 2). There are 39 drainages representing the extended Sumas-Blaine aquifer
359 region. Due to the intensive agricultural activities in the study area (see Figure 2 for the
360 land cover distribution), groundwater quality in the aquifer has been continuously
361 degrading in recent decades and nitrate concentrations are increasing (Almasri and
362 Kaluarachchi, 2004b). Since the role of nitrate in eutrophication is well-recognized
363 (Wolfe and Patz, 2002), nitrate contamination of the surface water of the study area is a
364 concern as it greatly affects fish habitat. The transport of nitrate to surface water occurs
365 mainly via discharge of groundwater during baseflow conditions (Schilling and Wolter,
366 2001; Bachman et al., 2002). Therefore, the prevention of groundwater contamination
367 from nitrate also protects surface water quality.

368 **3.2 Conceptualization of Nitrogen Transport**

369 As depicted in Figure 1, the conceptual model of nitrate fate and transport in
370 groundwater includes (Almasri and Kaluarachchi, 2004a,c): (i) characterization of land
371 use cover to compute the spatial distribution of on-ground nitrogen loadings; (ii) detailed
372 assessment of all nitrogen sources in the study area and their allocation to the appropriate
373 land cover classes; (iii) simulation of the soil nitrogen dynamics; (iv) prediction of nitrate
374 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:

$$C = f\left(\tau_F, \tau_M\right) \tag{13}$$

where $\boldsymbol{\tau}_{\mathrm{F}}$ and $\boldsymbol{\tau}_{\mathrm{M}}$ 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 τ_F and τ_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 τ_F and τ_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, $sig(\cdot)$, and $\tanh(\cdot)$ kernel—were used. For this case

study and data set trial-and-error analysis better performance was achieved with the $\mathrm{sig}(\cdot)$ 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 ε and c had to be set to their optimal values during the model training. For a given data set proper ε 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).

530 Table 1 presents the key statistics to evaluate the efficiency of the four learning
531 machines in the training and testing phases. All the machines have higher performance in
532 the training phase than in the testing phase. The loss of performance on the testing set
533 addresses the machine susceptibility to the issue of overtraining. There is a noticeable
534 reduction in performance on the testing data set (i.e., difference between machine
535 performance on training and testing) for both the ANN and LWPR models. The small
536 decline of performance on both RVM and SVM models indicates their ability to avoid
537 overtraining and hence generalize well.

538 Figures 5 and 6 show scatter plots of predicted (from the learning machine) versus
539 simulated (from the physical model) nitrate concentrations at two selected receptors. The
540 results indicate that the four learning machines did provide good prediction performance.
541 Figure 5 illustrates the prediction efficiency at the 19th receptor (see Figure 3). The SVM
542 model shows the highest accuracy with a coefficient of efficiency of 0.866, followed by
543 the RVM model at 0.864, the LWPR model at 0.837, and lastly the ANN model at 0.756.
544 The SVM model shows an average underbias of 0.021, while the other machines show an
545 overbias of 0.027, 0.031, and 0.037 for the RVM, LWPR, and ANN models, respectively.
546 Figure 6 demonstrates the performance of the machines at the 34th receptor (see Figure
547 3). The RVM model has a coefficient of efficiency value of 0.993, followed by the SVM,
548 ANN, and LWPR models with values of 0.988, 0.981, and 0.980, respectively. Again, the
549 RVM model shows the lowest bias, followed by the ANN, SVM, and LWPR models. The
550 ANN model experiences the highest variance as judged by a RMSE value of 0.113, while
551 the lowest is for the RVM model with RMSE = 0.066.

552 Figure 7 shows the prediction performance of the four machines at each receptor
553 in terms of RMSE. ANN performed the best for 25 receptors, while RVM performed the
554 best for 19 followed by SVM for 12. As evaluated by the mean absolute bias, SVM
555 performed the best for 21 receptors, ANN for 13 receptors, and RVM and LWPR for 11
556 receptors, each. From a bias-variance perspective, the ANN tends to produce a low
557 variance but high bias. SVM produced the best unbiased machine, yet it showed high
558 variance. A good tradeoff between bias and variance seems to be shown by the RVM for
559 this application.

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

568 Table 2 provides empirical generalization estimates in terms of root-mean-square-
569 error (RMSE) based on cross-validation and bootstrapping over scaled data. Linear
570 scaling to [0, 1] is performed for mapping real world measurement to a range of values
571 appropriate for model execution. Bootstrapping is useful in a situation where the
572 underlying sampling distribution of the data and the parameters is unknown and difficult
573 to estimate. Therefore, these statistics are mostly utilized for model selection purposes
574 and model reliability evaluation (Willmott et al., 1985). The model selection procedure

575 focuses on selecting the optimal set of model hyper-parameters by minimizing
576 bootstrapping or cross-validation estimates of the prediction error. For instance, the
577 number of hidden nodes in the ANN model was obtained by minimizing the variance and
578 the mean of the 10-fold cross-validation error. For development of the SVM model, the
579 10-fold cross-validation error was used to select the optimal trade-off, c, between model
580 complexity and the empirical risk. In their work with LWPR, Vijayakumar and Schaal
581 (2000a) used the leave-one-out error estimates in the gradient descent algorithm in
582 finding the metric parameters that specify the shape and region of validity of the
583 receptive fields. One might notice that according to the hybrid bootstrap and 0.632+
584 estimator, the ANN model has significantly higher generalization capability than the
585 other machines. However, the bootstrap estimates of the generalization error are
586 optimistically biased which is evident in the case of the ANN model where overtraining
587 results in a network that memorizes the individual examples rather than the trends in the
588 data set. Besides using these statistics for model selection, one can also use them to
589 provide confidence in the machine predictability, persistency, and robustness. As noticed
590 in Table 2, the four machines produce almost similar generalization error.

591 The statistical results reported in Table 2 provide credible estimates of machine
592 reliability and significance. The magnitude of the confidence interval for the accuracy
593 measure of interest could be used as a measure of model reliability (Willmott et al.,
594 1985). Principally, it is straightforward to estimate the confidence intervals of these
595 statistics. The width of the bootstrapping confidence intervals indicates implicit
596 uncertainty in the machine parameters. A wide confidence interval indicates that the
597 available training data set is inadequate to find a robust parameter set (Kuan et al., 2003).

598 The RVM model shows the narrowest confidence bounds. For example in the case of
599 hybrid bootstrap and 0.632+, the RVM model has $RMSE = 0.0232 \pm 0.000196$. The
600 SVM model shows a 20 percent increase in the confidence interval width, and both the
601 ANN and LWPR models show a 30 percent increase when compared to RVM. Owing to
602 the nonincremental application of LWPR in the testing (validation) phase, it produces the
603 lowest generalization performance. The use of LWPR is expected to be exceptional in
604 problems that are highly dynamic and characterized by nonstationarity (i.e., streamflow
605 predictions).

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

615 The SVM model is characterized by a highly effective mechanism for avoiding
616 overfitting that results in good generalization. The SVM formulation leads to a sparse
617 model dependent only on a subset of training examples and their associated kernel
618 functions (Vapnik, 1995). Tipping (2000) indicated that SVMs suffer from the absence of
619 a probabilistic prediction capability that captures information about uncertainty and from
620 the number of kernel functions that grows steeply with the size of the training data set,

621 from the necessity to manually tune some parameters, and from the selection of kernel
622 function parameters (i.e., which also has to satisfy Mercer's condition (Vapnik, 1995;
623 Tipping, 2000)). Empirical results proved that RVMs are remarkable in producing an
624 excellent generalization level while maintaining the sparsest structure. For example, the
625 SVM utilized 120 patterns as support vectors out of the 268 patterns of the training set,
626 while the RVM used only 26 patterns as relevance vectors, and LWPR used 40 receptive
627 fields. However, the support vectors in the SVM model represent decision boundaries,
628 while the RVM relevance vectors represent prototypical examples (Li et al., 2002). The
629 prototypical examples exhibit the essential features of the information content and thus
630 are able to transform the input data into the specified targets. This feature of both RVM
631 and SVM could be further utilized to build up a sparse representation of the processes
632 (e.g., monitoring network design).

633 ## 5. SUMMARY AND CONCLUSIONS

634 The machine learning induction techniques examined here have shown the ability
635 to build accurate models with strong predictive capabilities for groundwater quality and
636 they offer a practical approach to some modeling difficulties encountered in water-related
637 studies. Based on the evidence of the experiment, learning machines, other than ANNs,
638 appear to be highly effective. The results of the analyses presented here show distinct
639 performance preferences for each machine in a supervised-learning task. However, since
640 the comparisons between the different learning machines were intended to be illustrative
641 only, it should be strongly emphasized that no broader generalizations can be made about
642 the superiority of any of the machines for all classes of problems. The complex nature of
643 each of the learning algorithms that have been examined here makes it difficult to study

644 their statistical behavior in order to assess their performance objectively. Cross-validation
645 techniques can be robust for tuning parameter selection because they make no
646 assumptions about the data or noise distributions (Atkenson et al., 1997).

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

656 There are no criteria as when to use each of the presented machine other than to
657 bear in mind that ANNs minimize only the empirical risk by finding an optimal set of
658 weights for the chosen number of hidden nodes, while SVMs minimize the structural risk
659 to achieve estimators that are less susceptible to overfitting, as evident by the results
660 depicted in Table 1. Besides, owing to the quadratic optimization, SVMs are uniquely
661 solvable and there is no need to train them in a repetitive manner. In contrast, ANNs
662 require repeated training on the data set until a working model is attained. LWPR and
663 RVM entail iterative solutions until some stopping criteria are achieved. In addition,
664 SVMs achieve a global solution in the search for optimal parameter values and there is no
665 need for trial-and-error procedures to determine the final machine architecture, which is
666 directly obtained from the optimization solution. Also, ANNs rely heavily on the

667 structure of the networks, which is proven nontrivial and considered the most important
668 drawback of ANNs (Liong and Sivapragasam, 2002). The choice of the number of hidden
669 units in ANNs is problem-dependent and, therefore, it is difficult to determine a priori the
670 optimal network configuration. However, the performance of SVMs and RVMs depend
671 largely on the choice of kernel functions, which is in a sense equivalent to the choice of
672 the ANN structure. One may resort to cascade correlation or pruning techniques to adjust
673 the ANN structure to the complexity of the problem in an automatic way (Fahlman and
674 Lebiere, 1990). Primarily, in this application, ANNs, SVMs, RVMs, and LWPRs all
675 achieved their goal, namely pattern recognition in nitrate contamination occurrences in
676 groundwater. The resulting models, once constructed, are many orders of magnitude
677 faster than the process-based model. The comparison studies of learning machines mostly
678 revolve around the fact that superiority in performance heavily depends on the problem in
679 hand. In other words, there is a wide range of common applications that are of interest
680 where one machine will be proffered choice over the others. Strictly speaking, an ANN
681 prediction is more accurate in some problems, while SVM might be stronger in others.
682 RVM is the strongest when uncertainty bounds are required, and LWPR is the most
683 widely advocated in dynamic situations due to its incremental nature (e.g., when the input
684 distribution of the training data changes over time).

685 One also has to keep in mind that ANNs and SVMs both suffer a decline in
686 performance as the dimension of the data increase. Consequently, SVMs suffer from as
687 many difficulties as ANNs and RVMs in finding the optimum solution when the size of
688 the data set and/or the dimension of the input vector is large. When SVM is applied for
689 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

713 from combining the recommendations of experts for making predictions. The outlook for
714 the use of learning machines in water resources research and applications is very
715 promising.

APPENDIX I

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, $\mathbf{X} = \{\mathbf{x}_m\}_{m=1}^{M} = [\mathbf{x}_1\ \mathbf{x}_2 ... \mathbf{x}_M]$, will be referred to as $X^{(m)}$ and its corresponding output set, or targets, is $\{y_m\}_{m=1}^{M}$ where $\mathbf{x} \in \mathrm{R}^m$ and $\mathrm{y} \in \mathsf{R}$. 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:

$$Q\left(\mathbf{x}, X^{(m)}, A(X^{(m)})\right) = Q(\mathbf{x}, X^{(m)}) \tag{14}$$

where $\mathbf{x}$ is a random test point and $A(X^{(m)})$ is the set of hypotheses (a learning machine that assigns a prediction, $\hat{\mathrm{y}}$, to each $\mathbf{x}$) 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:

$$Err = Err\left(X^{(n)}, F\right) = E_{F(\mathbf{x})}\left[Q(\mathbf{x}, X^{(n)})\right] = \int_D Q(\mathbf{x}, X^{(n)})\, dF(\mathbf{x}) \tag{15}$$

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

$$Err_{CV \times k} = \frac{1}{k} \sum_{j=1}^{k} Q(S_j, X^{(m)}) \;,\; S_j \not\subset X^{(m)} \tag{16}$$

Leave-one-out-cross-validation error $Err_{CV \times m}$ constitutes the extreme case where k equals the number of training data sets $X^{(m)}$. Kohavi (1995) claimed that $Err_{CV \times m}$ 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:

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

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, $\mathbf{x}_i$, removed for testing (Efron and Tibshirani, 1997). The resulting error is:

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

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 $Err_{BS}^{(1)}$ is given by:

$$Err_h^{\lambda} = \lambda Err_{BS}^{(1)} + (1-\lambda)\overline{Err} \tag{19}$$

where λ 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 $\mathbf{x}_i$ will be included in the training bootstrap set $X_b^{(m)}$ is:

$$p\left(x_i \in X_b^{(m)}\right) = 1 - \left(1 - \frac{1}{m}\right)^m, \; and \quad p\left(x_i \in X_b^{(m)}\right) \approx 0.632 \quad \forall \; m \to \infty, \tag{20}$$

The 0.632+ estimator. This is a sophisticated estimator that accounts for the amount of overfitting and adjusts λ accordingly. The relative overfitting rate, $\hat{R}$, is derived as $\hat{R} = Err_{BS}^{(1)} - \overline{Err} \big/ \hat{\gamma} - \overline{Err}$, 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: $\hat{\gamma} = m^{-2} \sum_{i=1}^{m} \sum_{j=1}^{m} Q\left(\left\langle \mathbf{x}_i, y_j \right\rangle, X^{(m)}\right)$. For the no overfitting machine, $\hat{R} = 0$. The highest possible overfitting corresponds to $\hat{R} = 1$. The 0.632+ estimator is obtained as:

$$Err_{.632+} = Err_{.632} + \left(Err_{BS}^{(1)} - \overline{Err}\right) \frac{.368 \times .632 \times \hat{R}}{1 - .368\hat{R}} \qquad (21)$$

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

6. REFERENCES

Addiscott, T. M., A. P. Whitmore, and D. S. Powlson, 1991. Farming, fertilizers and the nitrate problem. CAB International, Wallingford, United Kingdom. 170 p.

Almasri, M. N. and J. J. Kaluarachchi, 2004b. Assessment and management of long-term nitrate pollution of groundwater in agriculture-dominated watersheds. Journal of Hydrology, 295(1-4): 225-245.

Almasri, M. N. and J. J. Kaluarachchi, 2004c. Modular neural networks to predict the nitrate distribution in groundwater using the on-ground nitrogen loading and recharge data. Environmental Modelling and Software. In press.

Almasri, M. N., 2003. Optimal management of nitrate contamination in groundwater. Unpublished PhD dissertation. Utah State University, Logan, Ut.

Almasri, M. N., and J. J. Kaluarachchi, 2004a. Implications of on-ground nitrogen loading and soil transformations on groundwater quality management. Journal of the American Water Resources Association (JAWRA), 40(1): 165-186.

Aly, A. H., and R. C. Peralta, 1999. Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm. Water Resources Research 35(8): 2523-2532.

ASCE Task Committee on Application of the Artificial Neural Networks in Hydrology, 2000a. Artificial neural networks in hydrology, I: Preliminary concepts. Journal of Hydrologic Engineering, ASCE, 5(2): 115-123.

ASCE Task Committee on Application of the Artificial Neural Networks in Hydrology, 2000b. Artificial neural networks in hydrology II: Hydrologic applications. Journal of Hydrologic Engineering, ASCE, 5(2): 124-137.

Atkenson, C. G., A. W. Moore, and S. Schaal, 1997. Locally weighted learning. Artificial Intelligence Review, 11: 11-73.

Atmadja, J., and A. C. Bagtzoglou, 2001. Pollution source identification in heterogeneous porous media. Water Resources Research, 37(8), pp.2113-2125.

Aziz A. R. A., and K. F. V. Wong, 1992. Neural network approach to the determination of aquifer parameters. Groundwater, 30(2): 164-166.

Bachman, L. J., D. E. Krantz, and J. Böhlke, 2002. Hydrogeologic framework, ground-water, geochemistry, and assessment of N yield from base flow in two agricultural watersheds, Kent County, Maryland. US Environmental Protection Agency, EPA/600/R-02/008, p. 46.

Berger, J. O., 1985. Statistical Decision Theory and Bayesian Analysis 2 Ed., Springer, New York.

Bishop, C. M., 1995. Neural Networks for Pattern Recognition. Oxford University Press.

823 Blum A., A. Kalai, and J. Langford, 1999. Beating the holdout: Bounds for k-fold and
824 progressive cross-validation. Proceedings of the 12th Annual Conference on
825 Computational Learning Theory, pp. 203–208.

826 CGER - Commission on Geosciences, Environment and Resources, 1993. Groundwater
827 vulnerability assessment: Predicting relative contamination potential under
828 conditions of uncertainty. National Academy Press, Washington, DC.

829 David, R. L., and M. J. Gregory, 1999. Evaluating the use of "goodness-of-fit" measures
830 in hydrologic and hydroclimatic model validation. Water Resources Research,
831 35(1) : 233–241.

832 Davison, A. C., and D. V. Hinkley, 1998. Bootstrap Methods and Their Application.
833 Cambridge University Press.

834 DeSimone, L., and B. Howes, 1998. N transport and transformations in a shallow aquifer
835 receiving wastewater discharge: A mass balance approach. Water Resources
836 Research, 34(2): 271-285.

837 Dibike, Y. B., S. Velickov, D. P. Solomatine, and M. B. Abott, 2001. Model induction
838 with support vector machines: introduction and applications. ASCE Journal of
839 Computing in Civil Engineering, 15(3): 208-216.

840 Efron B., R. J. Tibshirani, 1993. An Introduction to the Bootstrap. Chapman-Hall, New
841 York.

842 Efron, B., 1992. Jackknife-after-bootstrap standard errors and influence functions.
843 Journal of Royal Statistical Society, 54(1): 83-127.

844 Efron, B., and R. J. Tibshirani, 1997. Improvements on cross-validation: The .632+
845 bootstrap method. Journal of the American Statistical Association, 92(438): 548–
846 560.

847 Fahlman, S. E. and C. Lebiere, 1990. The cascade-correlation learning architecture. In
848 Advances in Neural Information Processing Systems, 2, edited by D. S.
849 Touretzky, pp. 524-532, Morgan Kaufmann Publishers, Los Altos, CA.

850 Frind, E., W. Duynisveld, O. Strebel, and J. Boettcher, 1990. Modeling of
851 multicomponent transport with microbial transformation in groundwater: The
852 Fuhrberg case. Water Resources Research 26(8): 1707-1719.

853 Hallberg, G. R., and D. R. Keeney, 1993. Nitrate, p. 297-321. In William M. Alley (Ed.).
854 Regional ground-water quality. U.S. Geological Survey, Van Nostrand Reinhold,
855 New York.

856 Harbaugh, A.W., and M. G. McDonald, 1996. User's documentation for MODFLOW-96,
857 An update to the U.S. Geological Survey modular finite-difference ground-water
858 flow model. U.S. Geological Survey Open-File Report 96-485, 56 p.

859 Hassan, A., and K. H. Hamed, 2001. Prediction of plume migration in heterogeneous
860 media using artificial neural networks. Water Resources Research, 37(3): 605-
861 623.

862 Haykin S., 1999. Neural networks a Comprehensive Foundation. 2 Ed., Macmillan
863 College Publishing Company, Englewood Cliffs, NJ.

864 Jacobs, R.A., M.I. Jordan, S.J. Nowlan, and G.E. Hinton, 1991. Adaptive mixtures of
865 local experts. Neural Computation, 3: 79-87.

866 Johnson, V. M., and L. L. Rogers, 2000. Accuracy of neural network approximator in
867 simulation-optimization. Journal of Water Resources Planning and Management,
868 126(2): 48-56.

869 Johnsson, H., M. Larsson, K. Mårtensson, and M. Hoffmann, 2002. SOILNDB: A
870 decision support tool for assessing nitrogen leaching losses from arable land.
871 Environmental Modelling and Software, 17(6): 505-517.

872 Jordan, M. I., and R. A. Jacobs, 1994. Hierarchical mixtures of experts and the EM
873 algorithm. Neural Computation, 6: 181-214.

874 Kaluarachchi, J. J., and M. N. Almasri, 2004. A mathematical model of fate and transport
875 of nitrate for the extended Sumas-Blaine Aquifer, Whatcom County, Washington.
876 Phase III Report. Utah State University, Logan, Ut. 146 p.

877 Kecman, V., 2001. Learning and Soft Computing: Support Vector Machines, Neural
878 Networks, and Fuzzy Logic Models. MIT Press, Cambridge, MA.

879 Kemblowski, M., and T. Asefa, 2003. Groundwater modeling of the lowlands of WRIA 1
880 watersheds. Draft Report, Utah State University, Logan, Ut.

881 Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and
882 model selection. Proceedings of the 14th International Joint Conference on
883 Artificial Intelligence (2): 1137-1145.

884 Korom, S. 1992. Natural denitrification in the saturated zone: A review. Water Resources
885 Research, 28(6): 1657-1668.

886 Kuan, M. M., C. P. Lim, and R. F. Harrison, 2003. On operating strategies of the fuzzy
887 ARTMAP neural network: A comparative study. International Journal of
888 Computational Intelligence and Applications, 3: 23-43.

889 Kunstmann, H., W. Kinzelbach, and T. Siegfried, 2002. Conditional first-order second
890 moment method and its application to the quantification of uncertainty in
891 groundwater modeling. Water Resources Research, 38 (4): 1035.

892 Lee, Y. W., 1992. Risk assessment and risk management for nitrate-contaminated
893 groundwater supplies. Unpublished PhD dissertation. University of Nebraska,
894 Lincoln, NE. 136 p.

895 Li, Y., C. Campbell, and M. Tipping, 2002. Bayesian automatic relevance determination
896 algorithms for classifying gene expression data. Bioinformatics, 18(10): 1332-
897 1339.

898 Liong, S., and C. Sivapragasam, 2002. Flood stage forecasting with support vector
899 machines. Journal of the American Water Resources Association, 38 (1): 173-
900 186.

MacKay, D. J., 1992. Bayesian methods for adaptive models. Ph.D. thesis, Dept. of Computation and Neural Systems, California Institute of Technology, Pasadena, CA.

MacKay, D., 2003. Information Theory, Inference, and Learning Algorithms. Cambridge University Press.

Magdon-Ismail, M., 2000. No free lunch for noise prediction. Neural Computation, 12(3): 547-564.

Maier, H. R., and G. C. Dandy, 2000. Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications. Environmental Modeling and Software, 15: 101-124.

McCulloch, W. S., and W. Pitts, 1943. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5: 115-133.

McLachlan, G. J., 1992. Discriminant Analysis and Statistical Pattern Recognition. Chapter 10, pp. 337-377. Wiley, New York.

Mitchell, R. J., R. S. Babcock, S. Gelinas, L. Nanus, and D. E. Stasney, 2003. Nitrate distributions and source identification in the Abbotsford-Sumas aquifer, Northwestern Washington State. Journal of Environmental Quality, 32: 789-800.

Morshed, J., and J. J. Kaluarachchi, 1998a. Application of artificial neural network and genetic algorithm in flow and transport simulations. Advances in Water Resources, 22 (2), pp. 145-158.

Morshed, J., and J. J. Kaluarachchi, 1998b. Parameter estimation using artificial neural network and genetic algorithm for free product and recovery. Water Resources Research, 34(5): 1101-1113.

Nabney, I., 2001. Netlab: Algorithms for Pattern Recognition. Springer, New York.

Nolan, B. T., K. Hitt, and B. Ruddy, 2002. Probability of nitrate contamination of recently recharged groundwaters in the conterminous United States. Environmental Science and Technology, 36(10): 2138-2145.

Postma, D., C. Boesen, H. Kristiansen, and F. Larsen, 1991. Nitrate reduction in an unconfined sandy aquifer: Water chemistry, reduction processes, and geochemical modeling. Water Resources Research, 27(8): 2027-2045.

Rogers L. L., F. U. Dowla, and V. M. Johnson, 1995. Optimal field scale groundwater remediation using neural networks and genetic algorithm. Environmental Science and technology, 29(5): 1145-1155.

Rogers L.L., and F. U. Dowla, 1994. Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. Water Resources Research, 30(2): 457-481.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1986. Learning internal representations by error propagation. In Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1, edited by D. E. Rumelhart and J. L. McClelland, Chapter 8, pp. 318-362, MIT Press, Cambridge, MA.

941 Schaal, S., C. Atkeson, and S. Vijayakumar, 2002. Scalable locally weighted statistical
942 techniques for real time robot learning. Applied Intelligence - Special issue on
943 Scalable Robotic Applications of Neural Networks, 17(1): 49-60.

944 Schilling, K. E., and C. F. Wolter, 2001. Contribution of base flow to nonpoint source
945 pollution loads in an agricultural watershed. Groundwater, 39(1): 49-58.

946 Schölkopf, B. and A. J. Smola, 2002. Learning with Kernels: Support Vector Machines,
947 Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA.

948 Schwaighofer, A., 2004. http://www.cis.tugraz.at/igi/aschwaig/software.html. Access
949 date: June 2004.

950 Shakhnarovich, G., R. El-Yaniv, and Y. Baram, 2001. Smoothed bootstrap and statistical
951 data cloning for classifier evaluation. Proceedings of International Conference on
952 Machine Learning: 521-528.

953 Shamrukh, M., M. Corapcioglu, and F. Hassona, 2001. Modeling the effect of chemical
954 fertilizers on groundwater quality in the Nile Valley Aquifer, Egypt.
955 Groundwater, 39(1): 59-67.

956 Stasney, D., 2000. Hydrostratigraphy, groundwater flow and nitrate transport within the
957 Abbotsford-Sumas Aquifer, Whatcom County, Washington. M.S. thesis. Western
958 Washington University, Bellingham.

959 Tesoriero, A. J., and F. D. Voss, 1997. Predicting the probability of elevated nitrate
960 concentrations in the Puget Sound Basin: Implications for aquifer susceptibility
961 and vulnerability. Groundwater, 35(6): 1029-1039.

962 Tesoriero, A., H. Liecscher, and S. Cox, 2000. Mechanism and rate of denitrification in
963 an agricultural watershed: Electron and mass balance along groundwater flow
964 paths. Water Resources Research, 36(6)" 1545-1559.

965 Tipping, M., 2000. The relevance vector machine. In Advances in Neural Information
966 Processing Systems, 12, edited by S. Solla, T. Leen, and K.-R. Muller, pp. 652–
967 658, MIT Press, Cambridge, MA.

968 Tipping, M.E., 2001. Sparse Bayesian learning and the relevance vector machine. Journal
969 of Machine Learning, 1: 211-244.

970 Tooley, J., and D. Erickson, 1996. Nooksack watershed surficial aquifer characterization.
971 Ecology Report #96-311. Washington State Department of Ecology, Olympia,
972 WA, p.12.

973 U.S. Department of Agriculture (USDA), 1987. The magnitude and cost of groundwater
974 contamination from agricultural chemicals, a national perspective. Staff Report
975 AGES870318. U.S. Department of Agriculture, Environmental Research Service,
976 Washington, D.C. p. 54.

977 Vapnik, V., 1982. Estimation of Dependencies Based on Empirical Data. Springer, New
978 York.

979 Vapnik, V., 1992. Principles of risk minimization for learning theory. In J. E. Moodey,
980 S.J. Hanson, and R. P. Lippmann (Eds.), Advances in Neural Information
981 Processing Systems, 4: 831-838.

982 Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer, New York.

983 Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York.

984 Vijayakumar, S., and S. Schaal, 2000b. Real time learning in humanoids: A challenge for
985 scalability of online algorithms. Humanoids 2000, First IEEE-RAS Intl. Conf. on
986 Humanoid Robots, MIT, Cambridge, MA.

987 Vijayakumar, S., and S. Schaal, 2000a. LWPR: An O(n) algorithm for incremental real
988 time learning in high dimensional space. Proc. of 17th International Conference
989 on Machine Learning (ICML 2000), Stanford, CA, pp.1079-1086.

990 Wagner, B. J., 1992. Simultaneous parameter estimation and contaminant source
991 characterization for couples groundwater flow and contaminant transport
992 modeling. Journal of Hydrology, 135: 275-303.

993 Wahba, G., 1985. A Comparison of GCV and GML for choosing the smoothing
994 parameter in the generalized spline-smoothing problem. The Annals of Statistics,
995 4:1378-1402.

996 Willmott, C. J., S. G. Ackleson, R. E. Davis, J. J. Feddema, K. M. Klink, D. R. J.
997 Legates, O. Donnell, and C. M. Rowe, 1985. Statistics for the evaluation and
998 comparison of models. Journal of Geophysical Research, 90 (C5): 8995-9005.

999 Wolfe, A. H., and J. A. Patz, 2002. Reactive nitrogen and human health: Acute and long-
1000 term implications. Ambio, 31(2): 120-125.

1001 Wolpert, D.H., and W. G. Macready, 1997. No free lunch theorems for optimization.
1002 IEEE Transactions on Evolutionary Computation, 1(1): 67-82.

1003 Wolpert, D.H., and, W.G. Macready, 1995. No Free Lunch Theorems for search. Santa
1004 Fe Institute Technical Report SFI-TR-05-010, Santa Fe, NM.

1005 Yu, X.Y., 2004. Support vector machine in chaotic hydrological time series forecasting.
1006 Ph.D. dissertation, National University of Singapore, Singapore.

1007 Yu, X.Y., S.Y. Liong, and V. Babovic, 2004. EC-SVM approach for real time hydrologic
1008 forecasting. Journal of Hydroinformatics 6: 209-223.

List of Tables

1015 Table 1. Key statistics for the prediction efficiency of the four learning machines in the
1016 training and testing phases (mean of the 56 receptors).

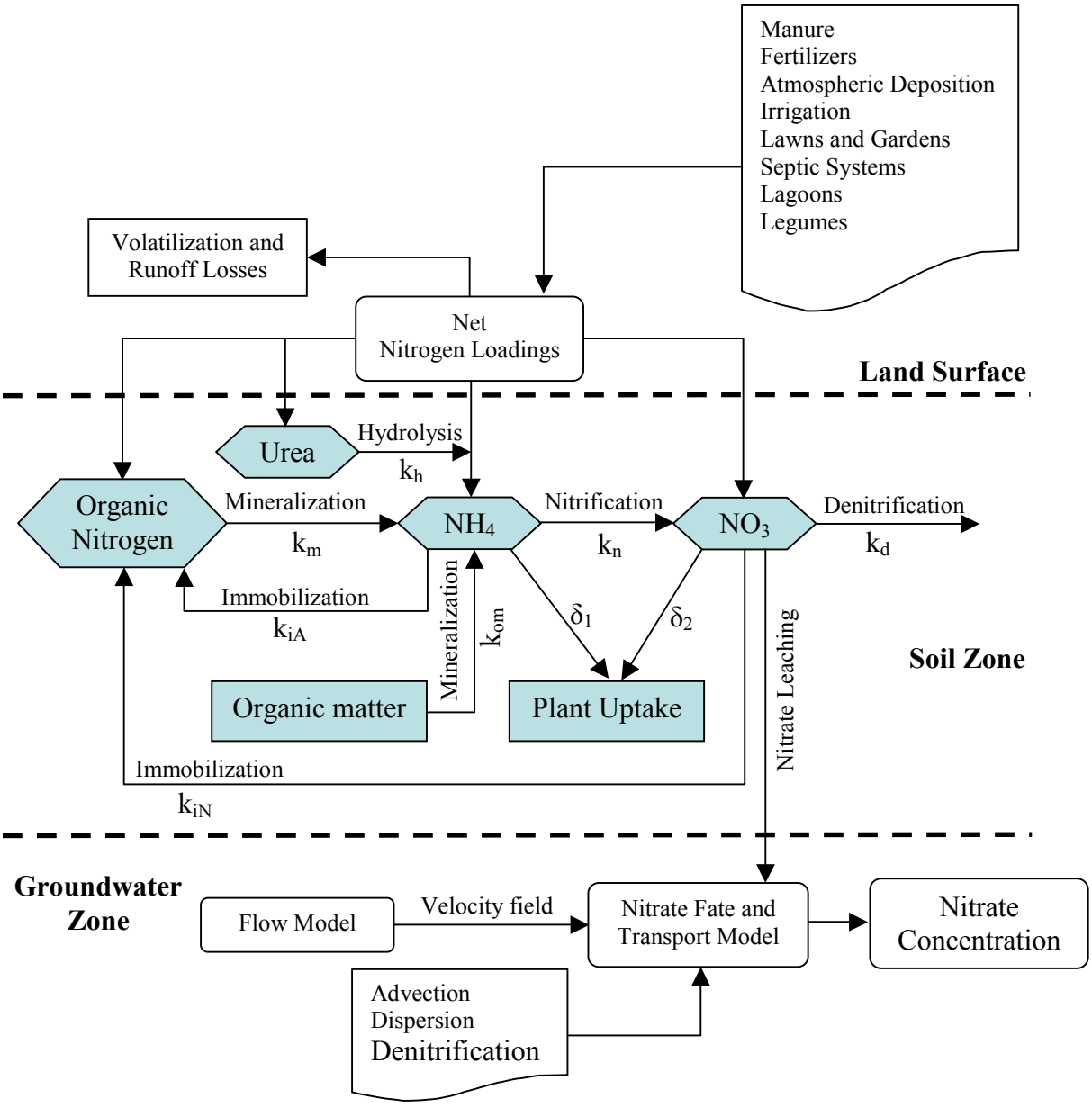
Statistics	ANN		SVM		RVM		LWPR	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Correlation coefficient	0.987	0.967	0.984	0.974	0.983	0.973	0.983	0.969
Coefficient of efficiency	0.974	0.933	0.966	0.948	0.966	0.946	0.966	0.911
Bias	0.000	0.021	-0.026	-0.004	0.000	0.015	0.000	-0.010
RMSE	0.131	0.192	0.143	0.185	0.141	0.183	0.141	0.229
Mean absolute error	0.085	0.131	0.074	0.115	0.095	0.128	0.095	0.172
Index of agreement	0.993	0.982	0.992	0.986	0.991	0.985	0.991	0.975

1017

1018 Table 2. Different generalization performance measures for the four learning machines
1019 (data scaled linearly to [0, 1]).

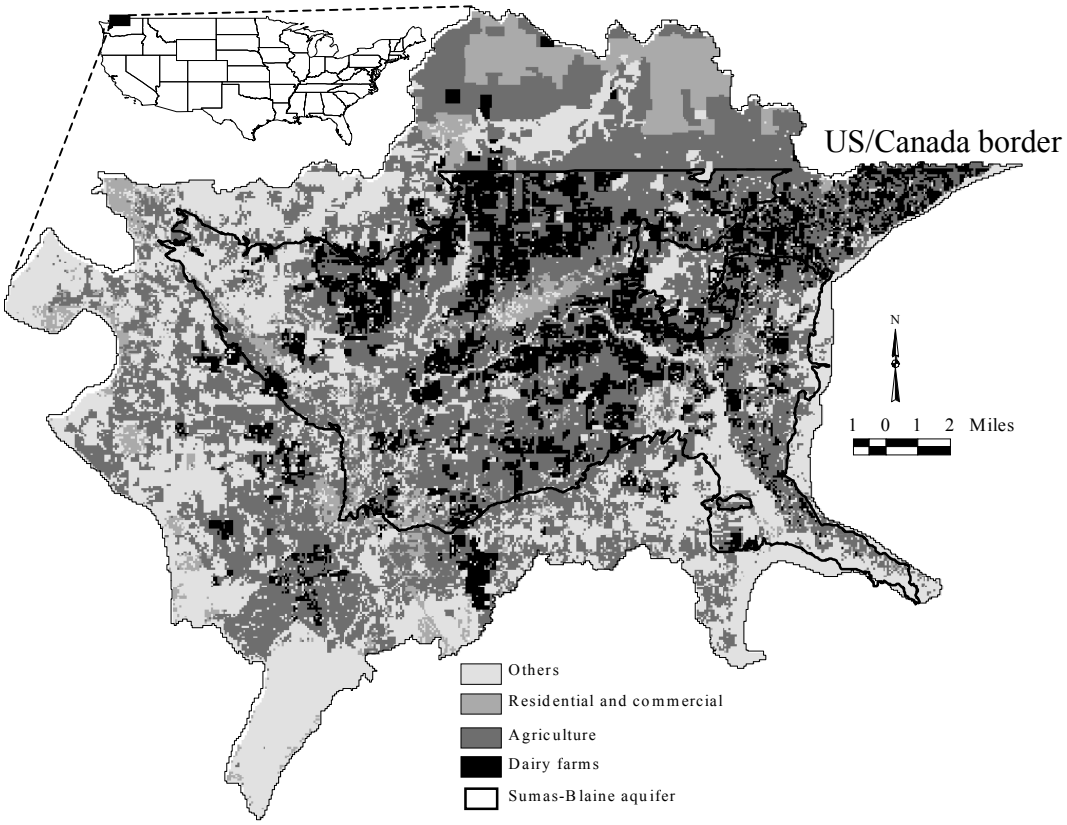
Generalization Error (RMSE)	ANN	SVM	RVM	LWPR
Empirical error	0.0214	0.0210	0.0206	0.0216
5-fold cross-validation	0.0237	0.0267	0.0248	0.0244
10-fold cross-validation	0.0234	0.0262	0.0261	0.0250
Leave-one-out error	0.0231	0.0245	0.0269	0.0252
Ordinary bootstrap estimator	0.0222	0.0258	0.0242	0.0261
Leave-one-out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

List of Figures



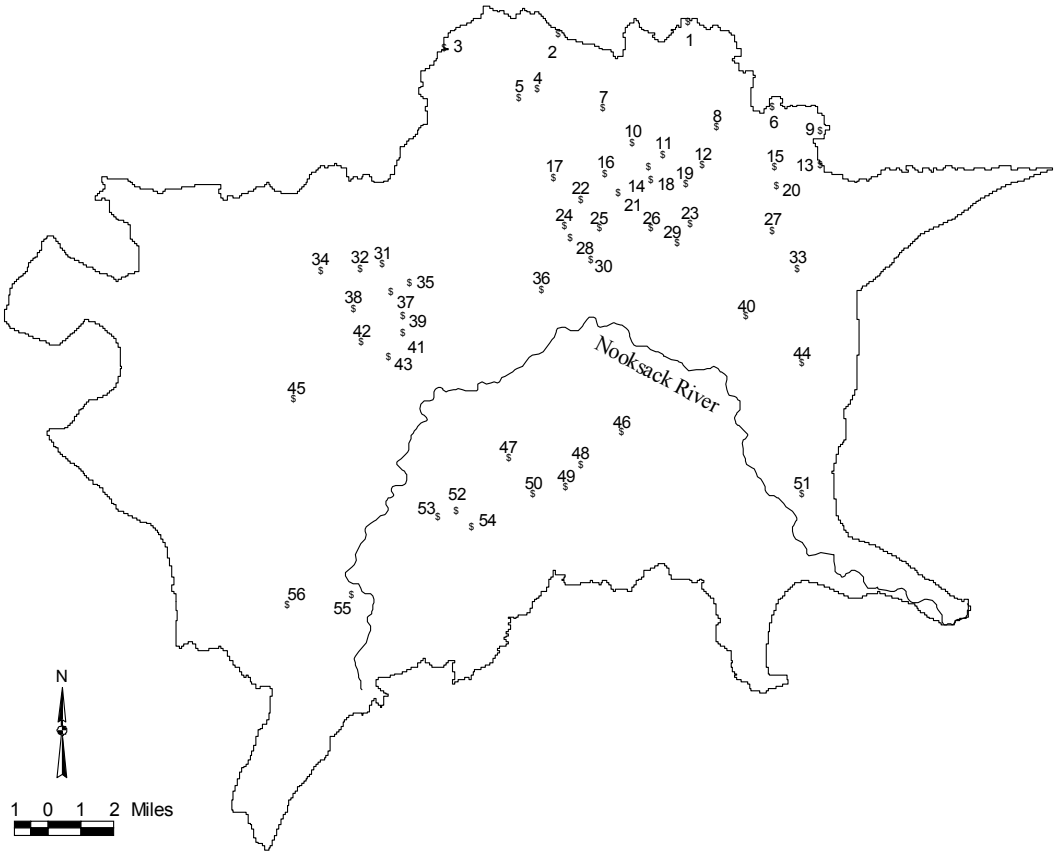
1038

1039 Figure 1. Schematic of the integrated modeling framework for simulating nitrate
1040 concentration in groundwater.



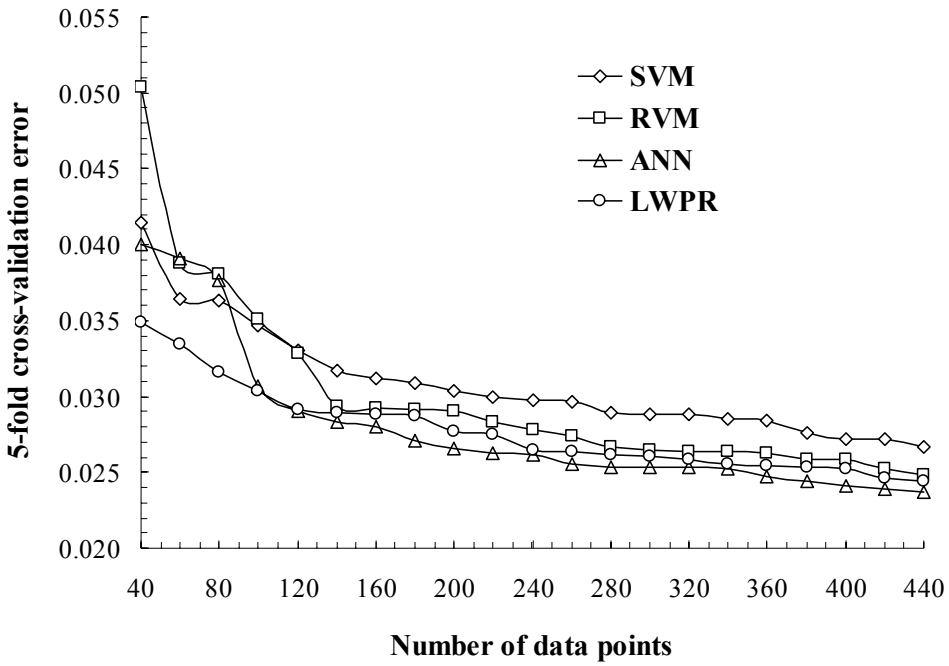
1041

1042 Figure 2. Physical model domain, consisting of the extended Sumas-Blaine aquifer and
1043 land use classes.



1044

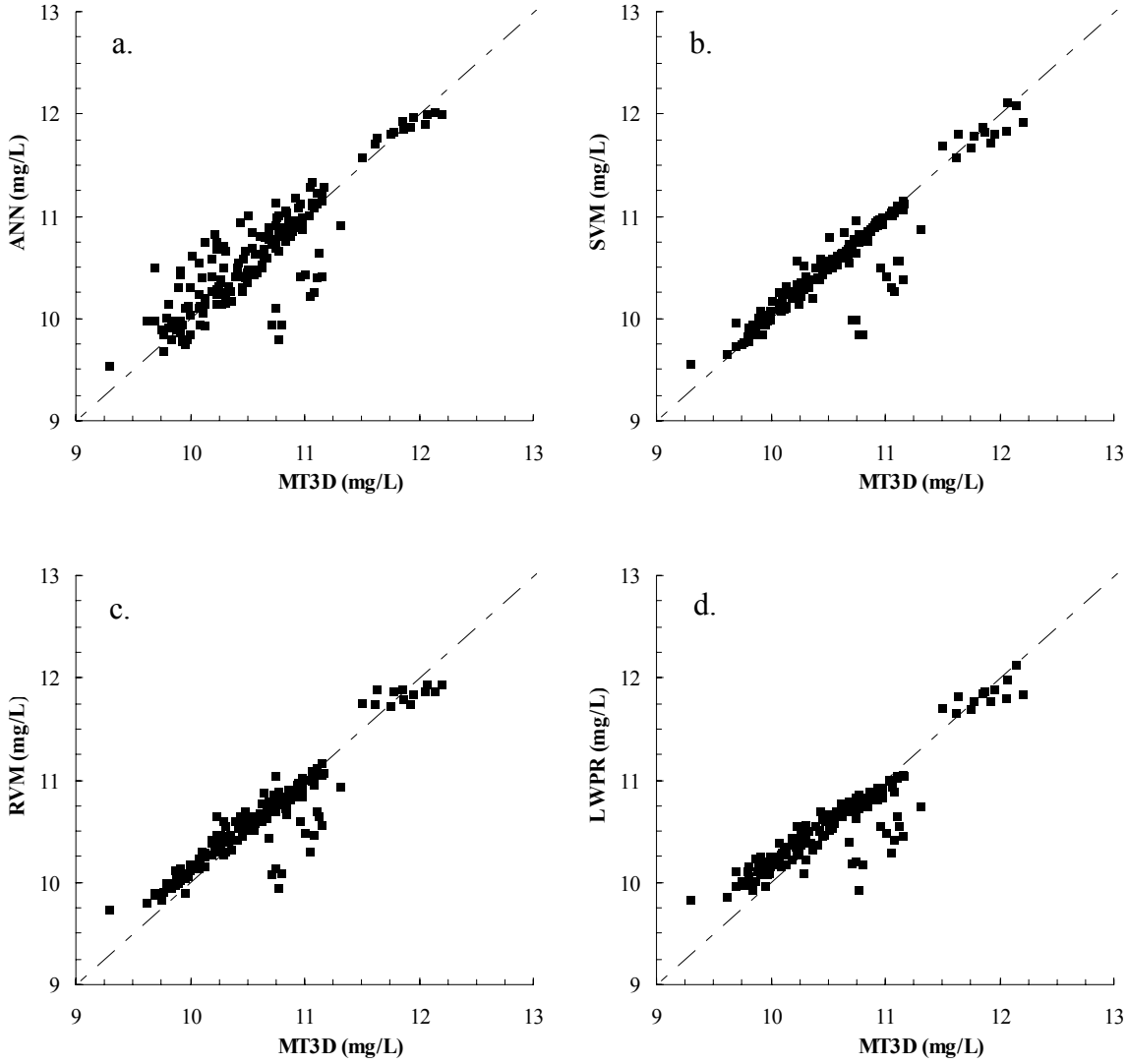
1045 Figure 3. The spatial distribution of the nitrate receptors in the study area.



1046

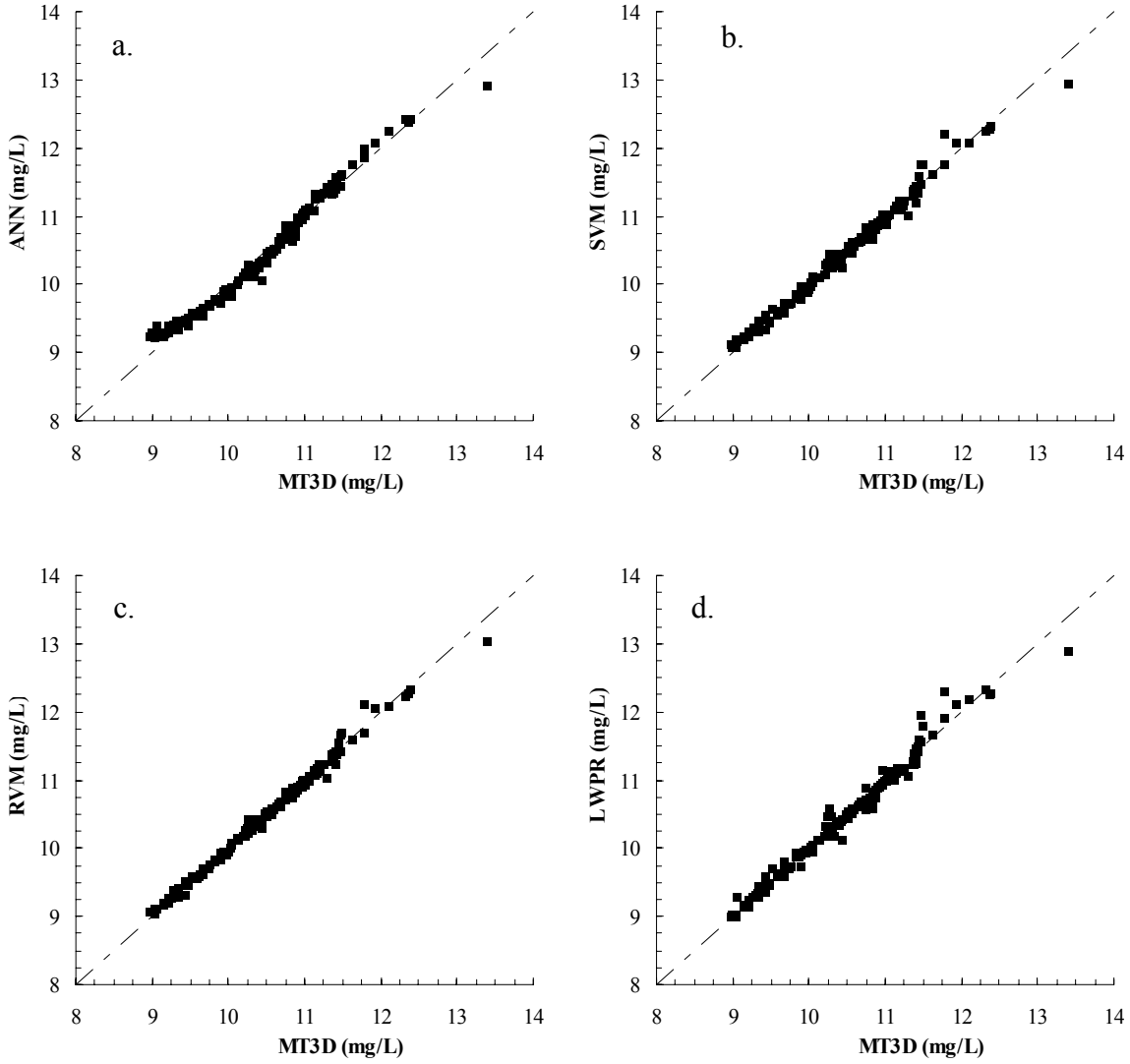
1047 Figure 4. Variability of the 5-fold cross-validation RMSE with the number of data points
1048 for the four learning machines (data scaled linearly to [0, 1]).

1049



1050 Figure 5. Scatterplot of the observed versus predicted nitrate concentrations at the 19th
1051 receptor for (a) ANN, (b) SVM, (c) RVM, and (d) LWPR.

1052



1053 Figure 6. Scatterplot of the observed versus predicted nitrate concentrations at the 34th
1054 receptor for (a) ANN, (b) SVM, (c) RVM, and (d) LWPR.

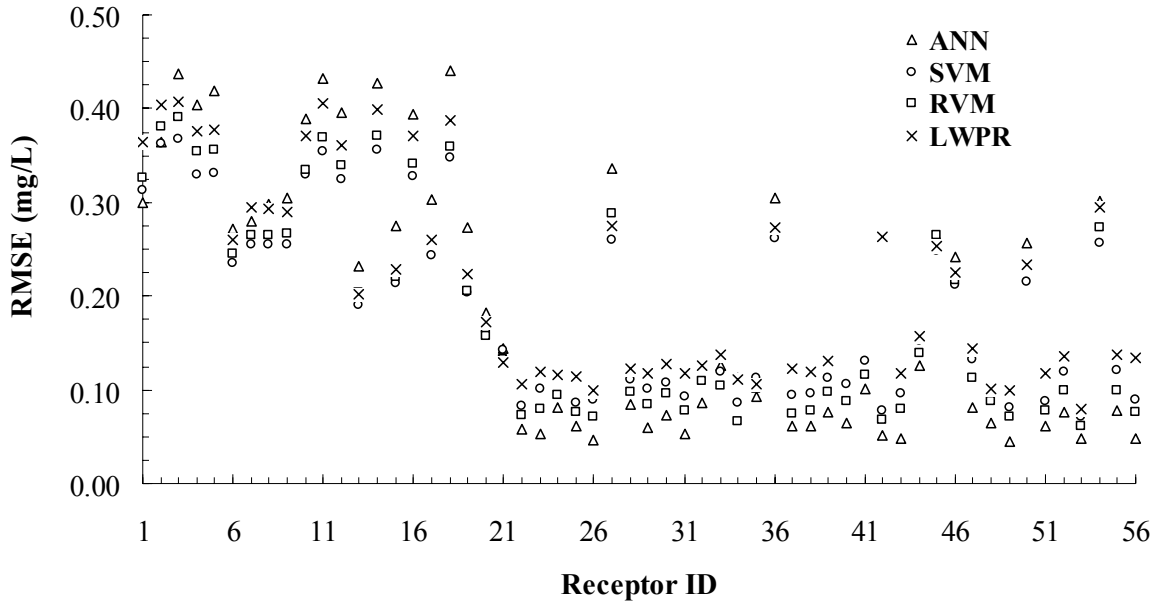
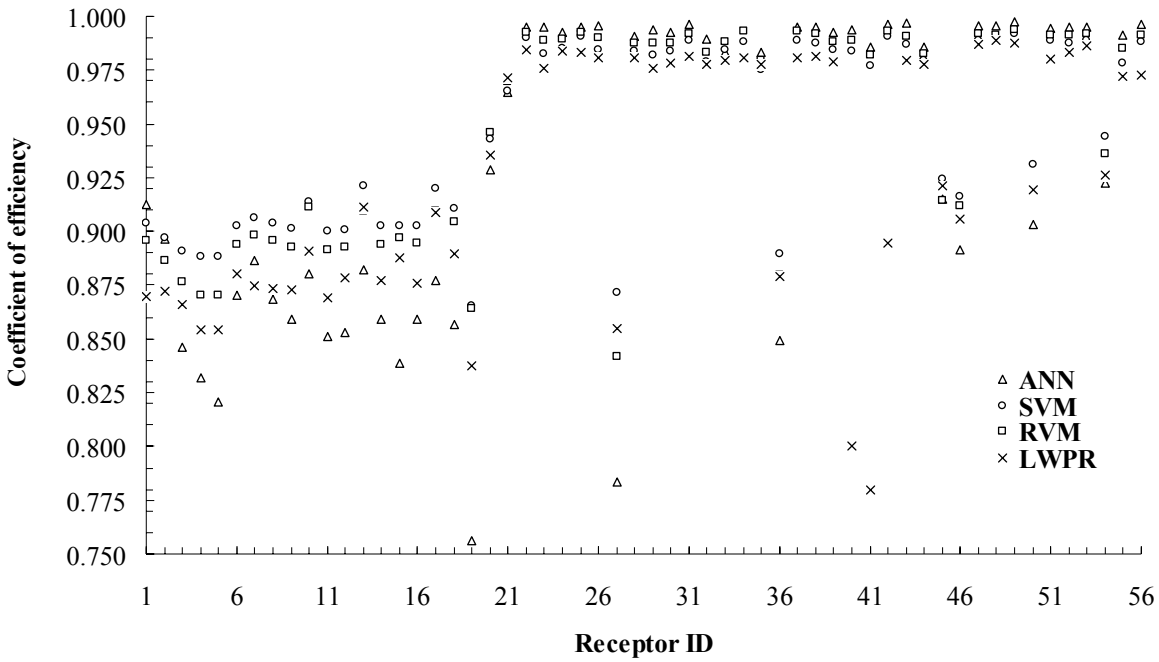


Figure 7. RMSE for the testing efficiency of the four learning machines for the 56 receptors.



1063

1064 Figure 8. Coefficients of efficiency for the testing efficiency of the four learning
1065 machines for the 56 receptors.
1066