QSAR/QSPR Studies Using Probabilistic Neural Networks and Generalized Regression Neural Networks

Philip D. Mosier and Peter C. Jurs*

Department of Chemistry, The Pennsylvania State University, 152 Davey Laboratory, University Park, Pennsylvania 16802

Received June 9, 2002

The Probabilistic Neural Network (PNN) and its close relative, the Generalized Regression Neural Network (GRNN), are presented as simple yet powerful neural network techniques for use in Quantitative Structure−Activity Relationship (QSAR) and Quantitative Structure−Property Relationship (QSPR) studies. The PNN methodology is applicable to classification problems, and the GRNN is applicable to continuous function mapping problems. The basic underlying theory behind these probability-based methods is presented along with two applications of the PNN/GRNN methodology. The PNN model presented identifies molecules as potential soluble epoxide hydrolase inhibitors using a binary classification scheme. The GRNN model presented predicts the aqueous solubility of nitrogen- and oxygen-containing small organic molecules. For each application, the network inputs consist of a small set of descriptors that encode structural features at the molecular level. Each of these studies has also been previously addressed in this research group using more traditional techniques such as *k*-nearest neighbor classification, multiple linear regression, and multilayer feed-forward neural networks. In each case, the predictive power of the PNN and GRNN models was found to be comparable to that of the more traditional techniques but requiring significantly fewer input descriptors.

INTRODUCTION

Although they are based on the classical and well-understood mathematics of probability and statistics, the probabilistic neural network[1] and generalized regression neural network[2,3] are still largely underutilized in the field of chemistry. Recently, however, some chemistry-related applications using PNNs have appeared and include the modeling of acute toxicity,[4−6] chromatographic pattern recognition,[7] characterizing aluminum hydroxide particles,[8] proton NMR classification,[9,10] and pattern recognition using chemical sensor arrays.[11] Recent chemistry-related applications of GRNNs include prediction of cetane number and density of diesel fuel,[12] leaf wetness,[13] proton NMR shifts,[14] and the performance of plasma-polymerized thin films.[15] The ability of the probabilistic neural network to classify and the generalized regression neural network to map any continuous function makes them ideally suited for Quantitative Structure−Activity Relationship (QSAR) and Quantitative Structure−Property Relationship (QSPR) studies.

To test the efficacy of the PNN/GRNN methodology using real-world data, two data sets were selected for study that have previously been addressed in this research group using more conventional modeling methods. The PNN study represents a biologically oriented QSAR and involves the classification of inhibitors of soluble epoxide hydrolase in humans.[16] Human soluble epoxide hydrolase (HSEH), E.C.3.3.2.3, is an important enzyme that catalyzes the conversion of epoxides to the corresponding diols. The ability to predict which molecules will efficiently bind to HSEH can identify possible therapeutic agents. The GRNN study represents a physical property-based QSPR and quantitatively predicts the aqueous solubility of small heteroatom-containing organic molecules.[17] The ability to accurately predict water solubility is important to many areas of chemistry. Two examples of this importance are environmental chemistry, where the concentrations of hazardous chemicals in the groundwater are sought, and medicinal chemistry, where the ADME (Absorption/Distribution/Metabolism/Excretion) properties of drug molecules must be optimized in order for the drug to be effective.

THEORY

Bayesian classification is a classical statistical technique in which predictions are made based on probabilities. The method is based upon Bayes' optimal decision rule, eq 1:

$$h_k \cdot c_k \cdot f_k(\mathrm{x}) > h_m \cdot c_m \cdot f_m(\mathrm{x}) \qquad (1)$$

In eq 1, h_k is the prior probability for class k, c_k is the cost of misclassification for class k, $\mathbf{x}$ is a vector of input variables, and $f_k(\mathbf{x})$ is a probability density function (pdf) related to the number and proximity of known (training set) cases of class k about the observed vector $\mathbf{x}$. Class k is assigned to vector $\mathbf{x}$ if the product of the three terms is greater for class k than for any other class m not equal to k.

The *prior probabilities* represent what are believed to be the true distribution of observations in each class. For example, the training set may be made up of 80% class 1 and 20% class 2. If this represents the actual distribution of class 1 and class 2 for the entire population of observations, then the priors may be assigned so that class 1 will be more heavily favored (by a factor of 4 in this case) when making the predictions for new observations. This will lower the misclassification rate because there are statistically four times as many instances of class 1 than of class 2, and the algorithm

10.1021/ci020039i CCC: $22.00
Published on Web 09/18/2002

will be correspondingly four times as likely to choose class 1 as class 2. In QSAR/QSPR applications, however, the ultimate class distribution is usually not known. In this case, the priors for each class are initially assigned such that no preference is given to any class, even if the class memberships differ.

The *cost of misclassification* is important if there are greater consequences for misclassifying one class over another. For example, a pharmaceutical company may wish to predict whether a potential therapeutic agent will be toxic or not. The two possibilities for misclassification are (1) classifying a nontoxic compound as toxic ("false positive") and (2) classifying a toxic compound as nontoxic ("false negative"). It is generally understood that false negatives are more costly than false positives because the toxicity may not be discovered until late in the drug development process when a great deal of time and money has been spent developing the drug. In this case, the cost of misclassifying a toxic compound is higher than the cost of misclassifying an inactive compound. The corresponding c's may be adjusted in eq 1 so that a more conservative approach is taken in that new compounds will be more likely to be classified as being toxic. This will lower the number of false negatives, usually at the expense of increasing the number of false positives.

The function $f_k(\mathbf{x})$ is a *probability density function* and embodies the inherent relationship between the inputs and output k. More specifically, the pdf function $f_k(\mathbf{x})$ is the probability or likelihood that that the observation $\mathbf{x}$ belongs to class k. Under normal circumstances, the true form of this function is not known. However, $f_k(\mathbf{x})$ can be approximated by a function $g_k(\mathbf{x})$ by using the information about the training set members that belong to class k. One method of achieving this approximation is attributed to Parzen,[18] later expanded to the multivariate case by Cacoullos,[19] and is given in eqs 2 and 3.

$$D(\mathrm{x},\mathrm{x}_i) = \sum_{j=1}^{p}\left(\frac{x_j - x_{ij}}{\sigma_j}\right)^2 \tag{2}$$

$$g_k(\mathrm{x}) = \frac{1}{n}\sum_{i=1}^{n} \exp(-D(\mathrm{x},\mathrm{x}_i)) \tag{3}$$

Using Parzen and Cacoullos' methods, a squared weighted Euclidean distance $D(\mathbf{x},\mathbf{x}_i)$ between an observation $\mathbf{x}$ and each of the other observations $\mathbf{x}_i$ in the training set belonging to class k may be calculated. In eq 2, p is the number of elements of the input vector $\mathbf{x}$. The σ variable is an adjustable parameter that is optimized as part of the learning process. There may be a separate σ weight for each of the p inputs as shown in eq 2, or there may be a single σ that is common to all of the input variables. The multiple-sigma models are more general and usually perform significantly better than the corresponding single-sigma models. In eq 3, the distance function $D(\mathbf{x},\mathbf{x}_i)$ is then transformed using a *kernel function*. Most commonly the kernel function is the Gaussian kernel, $\exp(-x^2)$, but other kernel functions may be used. Masters[20] discusses the requirements for a function to be used as a kernel. Note that the sigma parameter corresponds to the width of the Gaussian curve in a manner that is analogous to the classical standard deviation statistic. By summing the contributions from each of the training set members' kernels from class k, a continuous approximation to the true pdf is constructed. This is illustrated in Figure 1 for a single input variable x and a single class k.

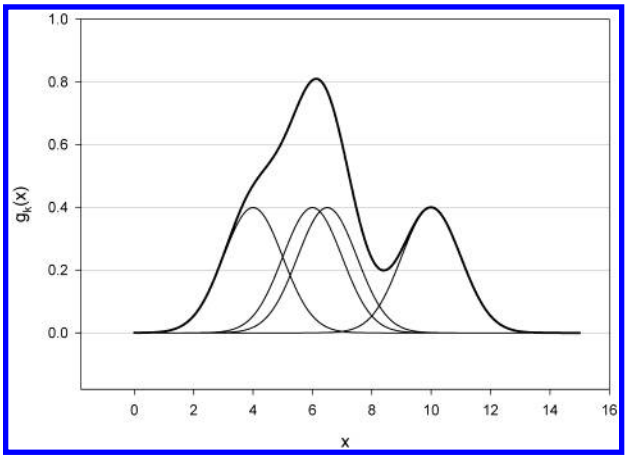

Figure 1. Parzen pdf approximation using normalized Gaussian kernels. An approximate pdf (indicated by the heavy line) is constructed for a single class in which there are four training set members. The x values of the TSET members are 4.0, 6.0, 6.5, and 10.0. The lighter curves represent the kernel functions for each of the TSET members. The σ value is 1.0.

The theory described thus far pertains to Bayesian classification. However, the pdf that is used in the Bayes optimal decision rule for classification may be generalized such that a continuous function mapping is achieved, allowing prediction of one or more continuous dependent variables. Predictions made in this way are based on the classical statistical conditional expected value of $\mathbf{y}$ given $\mathbf{x}$, eq 4:

$$E_{\mathrm{Y|X}}(\mathrm{x}) = \frac{\int_{-\infty}^{\infty} \mathrm{y}\cdot f_{\mathrm{XY}}(\mathrm{x},\mathrm{y})\cdot d\mathrm{y}}{\int_{-\infty}^{\infty} f_{\mathrm{XY}}(\mathrm{x},\mathrm{y})\cdot d\mathrm{y}} \tag{4}$$

In eq 4, $\mathbf{x}$ is the input vector and $\mathbf{y}$ is the output vector. The function $f_{XY}(\mathbf{x},\mathbf{y})$ is known as the *joint probability density function* and embodies the interrelationship between the input and output variables. As before, the true form of this function is not known. However, the joint pdf may be approximated by concatenating the $\mathbf{y}$ vector to the $\mathbf{x}$ vector and using Parzen and Cacoullos' method with the compounds of the training set.

$$\hat{\mathrm{y}}(\mathrm{x}) = \frac{\int_{-\infty}^{\infty} \mathrm{y}\cdot g_{\mathrm{XY}}(\mathrm{x},\mathrm{y})\cdot d\mathrm{y}}{\int_{-\infty}^{\infty} g_{\mathrm{XY}}(\mathrm{x},\mathrm{y})\cdot d\mathrm{y}} \tag{5}$$

Schiøler and Hartmann[2] have shown that when the Gaussian function is used as the kernel function, eq 5 can be simplified further:

$$\hat{\mathrm{y}}(\mathrm{x}) = \frac{\sum_{i=1}^{n} \mathrm{y}_i\cdot\exp(-D(\mathrm{x},\mathrm{x}_i))}{\sum_{i=1}^{n} \exp(-D(\mathrm{x},\mathrm{x}_i))} \tag{6}$$

Equation 6 is the fundamental equation of the PNN (classification) and the GRNN (function mapping). For the PNN

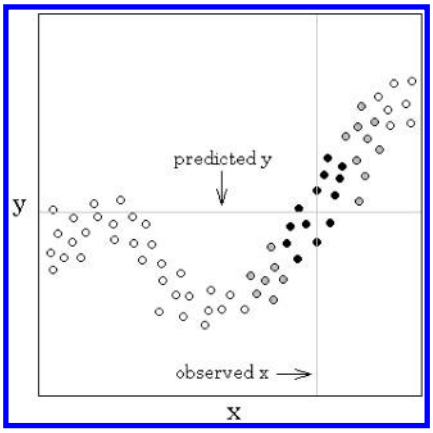

Figure 2. Illustration of the operation of a Generalized Regression Neural Network (GRNN), adapted from an illustration by Masters.[20] The predicted *y* value (indicated by a horizontal line) is a weighted average of the *y* values of the training set members, where the weights are the kernel-modified distances from an *x* value (indicated by a vertical line) whose *y* value is not known. Training set members that are closer to the *x* value are more influential in determining the predicted *y* value and are indicated by filled circles. Training set members that lie further away have less influence, and these are represented by gray and open circles.

classifier, there is an element in the class vector $\mathbf{y}_i$ corresponding to each class. Each element is zero except for the element corresponding to the class of observation *i* which is set to 1. This "delta vector" allows each training set member to contribute its kernel function only to the class that it belongs to in the numerator. The denominator of eq 6 sums the kernel contributions from all of the training set members. The elements of the prediction vector $\hat{\mathbf{y}}(\mathbf{x})$ thus range from 0.0 to 1.0 and represent Bayesian confidences. These confidence measures are known as *output activations* and sum to 1.0 across all classes. The PNN output activations represent the probability that the observation belongs to each class. For GRNN function mapping, each element of $\mathbf{y}$ corresponds to an individual continuous output (dependent variable) and the elements of $\hat{\mathbf{y}}(\mathbf{x})$ are the expected (predicted) values for the dependent variable(s). A practical explanation of the operation of a GRNN is illustrated in Figure 2 for a single input and dependent variable. A series of training set observations representing an arbitrary nonlinear function is shown as a set of circles. The vertical line represents an *x* value for which the predicted value of *y* is sought. The predicted value of *y*, represented by the horizontal line, may be found by using eq 6, and is a weighted average of the *i* training set members' *y* values where the weights are the kernel-transformed distances $\exp(-D(\mathbf{x},\mathbf{x}_i))$. The training set members closest to the vertical line will have the most influence in the prediction of the expected *y* value and these are indicated by solid circles. Training set members further away will have a diminished influence and are indicated by gray filled circles, and the training set members furthest away will have practically no influence and are indicated by open circles. Adjusting the sigma parameter changes the width of the kernel function, and this will determine how much influence nearby neighbors have in determining the predicted *y* value relative to the more distant members. Smaller values of sigma will produce a predicted *y* value that is a weighted average of only the closest neighbors; larger values of sigma will result in more distant observations having a larger influence in the predicted *y* value.

Training the PNN and GRNN involves finding the optimal values for the σ parameters in eq 2. For single-sigma models, there is one parameter to optimize, and an optimal value for sigma can be found using a good line minimization routine such as that of Brent.[21] For multiple-sigma models, a multivariate optimization routine such as conjugate gradients[22] is required. To use these optimization routines, the error or cost function to minimize must be defined. The cost function used by both PNN and GRNN is a mean squared error (MSE) and is given in eq 7.

$$COST = \frac{\sum_{I=1}^{N} e(\mathrm{x}_I)}{N} \tag{7}$$

Here, *N* is the number of training set members and $e(\mathbf{x}_I)$ is the squared error of prediction (error function) for training set member *I*. Internal validation in both the PNN and GRNN is achieved using the leave-one-out (LOO) method, in which the predicted value for the *I*'th training set member is derived from each of the other *N*-1 training set members. The error functions for the PNN and GRNN are somewhat different. For the PNN classifier, a continuous and differentiable error function is given in eqs 8 and 9.

$$e(\mathrm{x}_I) = [1 - b_k(\mathrm{x}_I)]^2 + \sum_{m \neq k} [b_m(\mathrm{x}_I)]^2 \tag{8}$$

$$b_k(\mathrm{x}_I) = \frac{\sum_{i=1}^{n} \delta_k(i) \cdot \exp(-D(\mathrm{x}_I,\mathrm{x}_i))}{\sum_{i=1}^{n} \exp(-D(\mathrm{x}_I,\mathrm{x}_i))} \tag{9}$$

In eq 8, training set member $\mathbf{x}_I$ is known to belong to class *k*. The first term in eq 8 is the probability (as determined by the other *N*-1 training set members) that $\mathbf{x}_I$ does not belong to the correct class *k*. The remaining terms represent the probability that $\mathbf{x}_I$ belongs to an incorrect class *m*. Equation 9 specifies that the output activations themselves are used in the evaluation of the error function and is equivalent to eq 6 when the $\mathbf{y}_i$ values of eq 6 take on only values of 0 and 1. For the GRNN, the usual mean squared error of prediction is used as the cost function. The pertinent equations are given in eqs 10 and 11 for the training set and a single dependent variable *y*.

$$e(\mathrm{x}_I;y_I) = [b(\mathrm{x}_I) - y_I]^2 \tag{10}$$

$$b(\mathrm{x}_I) = \frac{\sum_{i=1}^{n} y_i \cdot \exp(-D(\mathrm{x}_I,\mathrm{x}_i))}{\sum_{i=1}^{n} \exp(-D(\mathrm{x}_I,\mathrm{x}_i))} \tag{11}$$

Again, the equation for $b(\mathbf{x})$, the expected value of *y* in eq 11, is a special case of the more general eq 6 in which there may be more than one dependent variable. In eqs 9 and 11,

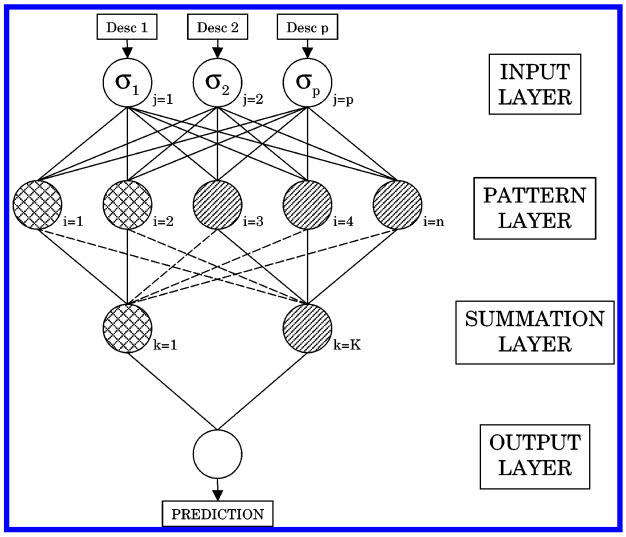

Figure 3. The network architecture of the PNN and GRNN. Nodes with checkered crosshatching in the pattern and summation layers represent class 1; nodes with diagonal line crosshatching represent class 2. Dashed lines connecting nodes of the pattern and summation layers indicate connections found only in the GRNN.

n represents the number of training set members. When LOO validation is used, only the summation terms where *I* is not equal to *i* are included in the calculation of b_k and *b*.

The techniques of Bayesian classification and Parzen pdf approximation long predated the modern computational neural network, but in 1990 Donald Specht[1] showed that these methods could be implemented as the probabilistic neural network. Shortly after this discovery, both Specht[3] and Schiøler and Hartmann[2] showed that the PNN could be generalized to predict continuous functions, resulting in the GRNN. The PNN and GRNN are very similar mathematically, and they have very similar network architectures. The combined architectures of the PNN and GRNN are shown in Figure 3. There are four layers of nodes: the input layer, the pattern layer, the summation layer, and the output layer. Unlike the more traditional multilayer feedforward neural network (MLFN), there are no weights associated with the connections in the PNN or GRNN. Instead, the connections depicted in Figure 3 indicate only that a given node supplies a piece of information to a node in the subsequent layer. All operations in the PNN and GRNN are performed in the nodes themselves. The *p* nodes of the input layer represent the input descriptors, and no special operation is performed in these nodes. There are as many nodes in the input layer as there are inputs to the network, and there is a σ weight associated with each input layer node. The nodes of the input layer are fully connected to the nodes of the pattern layer. There is a node in the pattern layer for each member of the training set. For the PNN, the class of each training set member is indicated by a different crosshatched pattern. Each of the *n* nodes in this layer calculates the weighted Euclidean distance function $D(\mathbf{x},\mathbf{x}_i)$ given in eq 2. In the PNN, there is a node in the summation layer for each of the *K* classes. Nodes in the pattern layer of like class are connected to a single node in the summation layer representing that particular class. These connections are represented by solid lines in Figure 3. Each node in the summation layer of a PNN calculates the numerator portion of eq 6. Finally, the single node of the output layer of the PNN calculates the denominator portion of eq 6 and the output activations and assigns a class to the input vector based on the optimal decision rule in eq 1. In a GRNN, there are no classes, and each node of the pattern layer is fully connected to exactly two nodes in the summation layer. The additional connections required for a GRNN are indicated in Figure 3 by dashed lines. In the case of the GRNN, the two nodes of the summation layer calculate the numerator and denominator of eq 6. The output layer, consisting of one node, performs the division and generates the vector of predicted *y* values.

EXPERIMENTAL METHOD

The process used to develop the models described here consisted of four stages: (1) structure entry and optimization, (2) descriptor generation, (3) objective feature selection, and (4) subjective feature selection. A general description of each of these steps is given here.

Structure Entry and Optimization. In the structure entry and optimization stage, the molecular structures of the compounds used in each study were entered into the computer using the molecular modeling program HyperChem (HyperCube, Inc., Waterloo, ON) on a desktop PC. Atom types and coordinates and bond connectivities were stored in individual HyperChem input (.hin) files. These were then transferred to a UNIX workstation where each molecule was assigned a low-energy conformation using the semiempirical molecular orbital methods found in the MOPAC 6.0 software package.[23] The PM3 Hamiltonian was used to obtain the low-energy conformations, and the AM1 Hamiltonian was used to assign charges to the atoms in the structures. The molecules were then ready to be used with the Automated Data Analysis and Pattern Recognition Toolkit (ADAPT).[24,25] The entire set of compounds in each study was divided into two subsets: a training set (TSET) whose information was used to build the actual models and a prediction set (PSET), consisting of molecules not found in the TSET, which was used to validate the models once they were built. An additional cross-validation set (CVSET) of compounds was selected and removed from the TSET and used to internally validate models whose fitness evaluators required it. Members of each set were assigned randomly, but with the condition that the values of the dependent variable for none of the PSET or CVSET members exceed the dependent variable range defined by the TSET.

Descriptor Generation. In the descriptor generation stage, various properties of the molecules were calculated and stored using routines found in the ADAPT package. Four classes of descriptors[26] were generated in each of the studies presented here: topological, geometric, electronic, and hybrid. Topological descriptors are based on graph theory and encode information about the types of atoms and bonds in a molecule and the nature of their connections. Examples of topological descriptors include counts of atom and bond types and indexes that encode the size, shape, and types of branching in a molecule. Geometric descriptors encode information about the three-dimensional nature of the molecule. Examples of geometric descriptors are solvent-accessible surface areas, moments of inertia, and shadow areas. Electronic descriptors encode the electronic character of the molecule. Examples include HOMO and LUMO energies, dipole moments, and atomic charges. Finally, hybrid descriptors incorporate information from two or more

of the above categories. Notable examples include the charged partial surface area descriptors,[27] which combine information about both the surface area (a geometric property) and the atomic charges (an electronic property) of individual atoms. The CPSA descriptors correlate well with the molecular polar surface area (PSA) descriptors of Stenberg et al.[28] Additional whole-molecule descriptors were generated using the DRAGON[29] computer program of Todeschini et al.

Objective Feature Selection. Objective feature selection was used to reduce the pool of descriptors by eliminating those descriptors (features) that contained little or redundant information. Any descriptor whose values were identical for at least 90% of the compounds in the training set was removed due to insufficient information content. In addition, one of two descriptors whose pairwise correlation coefficient exceeded 0.90 for the training set members was also removed to eliminate redundant information. Objective feature selection is carried out using only the independent variables (descriptors); the dependent variable is not used. The resulting reduced pool of descriptors was submitted to the subjective feature selection routines for model development.

Subjective Feature Selection. Subjective feature selection was used to select subsets of descriptors from the reduced pool that optimally modeled the physical property or biological activity of interest. This is achieved through the use of a *subjective feature selection* algorithm which selects subsets of descriptors coupled with a *fitness evaluator* that determines the value of the cost function associated with each of the descriptor subsets. For the models presented here, the subjective feature selection routines used were Generalized Simulated Annealing (GSA) and Genetic Algorithm (GA). The fitness evaluators used were the *k*-Nearest Neighbor (*k*-NN) algorithm, the Multi-Layer Feed-forward Neural Network (MLFN), the Probabilistic Neural Network (PNN), and the Generalized Regression Neural Network (GRNN). A GA was used to select descriptor subsets for the *k*-NN and MLFN fitness evaluators; a GSA feature selection algorithm was used to select descriptor subsets for the PNN and GRNN. For the MLFN, PNN, and GRNN, optimization involved modifying the adjustable parameters so that the predefined cost function was minimized. Internal validation was used in these fitness evaluators to prevent overtraining, a phenomenon in which the model learns the idiosyncrasies of the TSET members and loses the ability to generalize. Depending on the particular model building routine, this was achieved either by the leave-one-out (LOO) method (for PNN and GRNN) or by the use of a separate CVSET (for MLFN). In the LOO method, one TSET member was predicted at a time, while using the remaining *n*-1 TSET members as part of the network. The cost function used for the PNN and GRNN was eq 7. A CVSET was used for building MLFN models. The members of the CVSET were selected and removed from the training set. A separate root-mean-square error (RMSE) was then calculated for the members of the TSET and the external CVSET, and eq 12 was used as the cost function.

$$COST = RMSE_{TSET} + 0.4 \cdot |RMSE_{TSET} - RMSE_{CVSET}| \quad (12)$$

The subjective feature selection routines in each case provide a list of the top-performing models, and the predictive power of these was assessed using the compounds of the PSET. Models whose PSET errors were comparable to or lower than their TSET error were considered to have good predictive ability.

PNN and GRNN Feature Selection. A generalized simulated annealing-driven, PNN/GRNN based feature selection routine was used to select an optimal set of descriptors from the reduced descriptor pool as inputs to the PNN or GRNN. Each descriptor in the reduced pool was transformed linearly on the range 0 to 1 prior to its use as a network input. The networks were trained using a quasi-Newton Polak-Ribière conjugate gradient optimization algorithm.[22] To maintain consistency, the reduced descriptor pool and the TSET and PSET members used in each study were retained from their original sources. The PNN/GRNN algorithm uses the LOO cross-validation method, in which each member of the TSET is assigned a predicted value based on a model built using each of the other members of the TSET. The originally defined CVSET in each study was thus not needed and was merged with the TSET. Each model was built using a separate sigma weight for each input descriptor and using the Gaussian kernel function to construct Parzen pdfs. The reduced pools of descriptors were also retained from the original sources. PNN and GRNN models were constructed using varying numbers of inputs in determining the optimal descriptor subset. The models were then validated using the compounds of the external PSET.

All PNN and GRNN computations were performed on a 1.0-GHz Athlon Thunderbird class workstation running the Microsoft Windows 2000 operating system. The PNN/GRNN software used in this work was written in-house in FORTRAN 90 and compiled on a PC workstation using the Compaq Visual FORTRAN 6.1 compiler and is based upon the C++ code of Masters.[20,30] Calculation of the DRAGON descriptors was performed on an Intel Pentium 4 desktop PC. All other computations were performed on 500-MHz DEC Alpha workstations running the OSF/1 operating system. Additional software that was used in the development of the QSAR models described here, including ADAPT and code implementing computational neural networks, simulated annealing, and genetic algorithms, was also written in-house.

RESULTS AND DISCUSSION

Soluble Epoxide Hydrolase Inhibitor Data Set. 339 compounds were tested for their ability to inhibit the soluble epoxide hydrolase (SEH) enzyme in humans and classified as either inactive or active. Models describing the compounds' activity were developed by McElroy et al.[16] The compounds were subdivided into a 304-member training set and a 35-member external prediction set. The training set contained 106 inactive and 198 active inhibitors of human soluble epoxide hydrolase. The prediction set consisted of 11 inactive and 24 active inhibitors. The cost of misclassification was set equal to 1.0 for all classes, and the prior probabilities for each class were assigned so as to negate the effect of unequal class populations. That is, no class was favored over another. Using the original 119-descriptor reduced pool, GRNN models containing from 2 to 6 descriptors were examined for overall quality. The four-descriptor subset that was selected as the best performer is

Table 1. Descriptors Selected for the HSEH GSA-PNN Model

descriptor	type[a]	sigma	range	explanation[b]
MOLC-9	T	0.0283	1.10 to 4.56	topological J index
MDE−33	T	0.0262	0.0 to 179.0	distance-edge between 3° carbon atoms
PND-5	T	0.0047	0.0 to 312.0	oxygen-only superpendentic index
SCAA-2	G/E	0.1168	−15.1 to 0.0	SA-charge hydrogen bonding

[a] T = topological; G/E = geometric/electronic hybrid. [b] MOLC-9, Balaban topological J index; MDE-33, distance-edge index between tertiary carbon atoms; PND-5, oxygen-only superpendentic index; SCAA-2, sum of (surface area × charges) of acceptor atoms divided by the number of acceptor atoms.

Table 2. Confusion Matrices for the HSEH GSA-PNN Model Using Costs of Misclassification that (a) Favor Neither the Active nor Inactive Class and (b) Favor the Active Class

(a) Favor Neither the Active nor Inactive Clas

	predicted class		
actual class	inactive	active	% correct
	training set		**86.2**
inactive	88	18	83.0
active	24	174	87.9
	prediction set		**91.4**
inactive	11	0	100.0
active	3	21	87.5

(b) Favor the Active Class

	predicted class		
actual class	inactive	active	% correct
	training set		**86.5**
inactive	80	26	75.5
active	15	183	92.4
	prediction set		**91.4**
inactive	9	2	81.8
active	1	23	95.8

presented in Table 1. The sigma values for the input descriptors ranged from 0.0047 to 0.1168. This model is presented in Table 2a. This LOO-validated model was able to predict 83.0% of the inactives correctly and 87.9% of the actives correctly for the training set for an overall prediction rate of 86.2%. For the prediction set compounds, 100.0% of the inactives and 87.5% of the actives were classified correctly for an overall prediction rate of 91.4%. Pairwise squared correlation coefficients (r^2) among the four descriptors were very low and ranged from 0.0029 to 0.098 with a mean of 0.044. No trends were evident between any of the four descriptors and the activity of the training set members.

Three of the descriptors in the model are topological in nature and one is a hybrid descriptor with both a geometric and an electronic component. The first topological descriptor, MOLC-9, is the topological J index as defined by Balaban.[31] The J index is an *averaged distance sum connectivity* index and can be thought of as an extension of the simple connectivity index,[32] where the adjacency matrix has been replaced with the topological distance matrix. The J index is primarily a branching index; however, there is a size dependence that will result in larger J indices for larger molecules. For molecules of the same size, larger J indices arise from structures that are more highly branched because the shortest path between atoms is significantly smaller in molecules with more branches or rings. This descriptor may thus be indicating that there is a dependence on molecular size and shape. This seems reasonable since the HSEH enzyme likely has a ligand binding pocket that will accept only molecules of certain size and shape. The second topological index in the model is MDE-33, a molecular distance-edge vector.[33] MDE vectors encode the interconnectedness of the underlying carbon skeleton of a molecule. MDE-33 encodes the interaction between tertiary carbon atoms. This descriptor is encoding the shape requirements for the carbon skeleton and in particular that tertiary carbon atoms are important. The final topological descriptor is PND-5, the superpendentic index[34] calculated for pendant oxygen atoms. The superpendentic index employs a topological distance matrix to calculate the shortest path between pairs of (in this case) oxygen atoms. Larger PND-5 values occur when there are more oxygen atoms in a molecule and when their through-bond distances are greater. PND-5 is contributing information that suggests that the presence of oxygen atoms is important as well as the oxygen atoms' relationship to one another. The hybrid descriptor included in this model is SCAA-2. This is a hydrogen bonding charged partial surface area (CPSA) descriptor[27] that is sum of (surface area × charges) of acceptor atoms divided by the number of acceptor atoms. This descriptor is conveying information about the hydrogen bonding properties of the molecule, which is an important aspect of many enzyme-ligand interactions.

The three compounds that were misclassified in the prediction set are all false negatives; that is, they were truly active but were classified as being inactive. According to the output activations from the model, the probability that each of these compounds is active was 0%, 45%, and exactly 50%. The compound with zero percent probability of being active was most likely classified as such because its only neighbors in the sigma-weighted Euclidean input descriptor space were all inactive. For the compound whose output activation for the active class was 45%, both classes had nonzero contributions, but the inactive compounds had slightly more influence. An output activation of exactly 50% can mean that there is a nonzero but exactly equal chance that the compound belongs to each of the two classes. That is, the compound has Euclidean neighbors of both classes and they produce exactly equal output activations. However, in most all cases this means that the model simply does not have enough information from the available training set to make a confident prediction of the activity of a new molecule. When this happens, there are no training set members in the vicinity of the new molecule in the weighted Euclidian space to compare activities to.

To reduce the number of false negatives in the model, the cost of misclassification in eq 1 was adjusted, and the subjective feature selection routine repeated for models with four input descriptors. Since there were more active compounds in the training set than inactives (198 active; 106 inactive), the cost of misclassification for the active class was set to 198 and the cost for the inactive class was set to 106. Thus, the active class was nearly twice as likely to be chosen as the inactive class when the pdf functions from each class are equal. The best model had the same four descriptors as the original model but with slightly different sigma values for each descriptor. The sigma values are 0.0282 for MOLC-9, 0.0250 for MDE-33, 0.0027 for PND-

Table 3. Descriptors Selected for the HSEH GA-*k*-NN Model

descriptor	type[a]	range	explanation[b]
MOLC-9	T	1.10 to 4.56	topological J index
1SP3-1	T	0 to 10	count of sp^3 carbons attached to one carbon
2SP3-1	T	0 to 17	count of sp^3 carbons attached to two carbons
PND-5	T	0 to 312	oxygen-only superpendentic index
SAAA-3	G/E	0.0 to 0.745	SA of acceptor atoms/total SA
CHAA-1	G/E	−2.19 to −0.286	sum of charges on acceptor atoms
WHIM-42	G/E	0.373 to 0.697	weighted holistic invariant metric

[a] T = topological; G/E = geometric/electronic hybrid. [b] MOLC-9, Balaban topological J index; 1SP3-1, count of sp^3-hybridized carbon atoms bonded to one other carbon atom; 2SP3-1, count of sp^3-hybridized carbon atoms bonded to two other carbon atoms; PND-5, superpendentic index between oxygen atoms; SAAA-3, surface area of hydrogen bond acceptor atoms divided by the total molecular surface area; CHAA-1, sum of the atomic charges on hydrogen bond acceptor atoms; WHIM-42, first component accessibility directional WHIM index weighted by atomic Sanderson electronegativities.

Table 4. Confusion Matrices for the HSEH GA-*k*-NN Model

	predicted class		
actual class	inactive	active	% correct
	training set		**89.1**
inactive	88	18	83.0
active	15	183	92.4
	prediction set		**91.4**
inactive	9	2	81.8
active	1	23	95.8

5, and 0.0848 for SCAA-2. The order of the descriptors with respect to the sigma values remains the same as in the original PNN model. As expected, the number of compounds classified as active increased in both the training and prediction sets. This effectively reduced the number of false negatives, while increasing the number of false positives. As shown in Table 2, the number of false negatives in the training set was lowered from 24 to 15 and the number of false positives was increased from 18 to 26. For the compounds of the prediction set, the number of false negatives was reduced from 3 to 1 and the number of false positives was increased from 0 to 2. The one compound left as a false negative had an output activation for the active class of zero and was the same compound in the original PNN model with zero output activation for the active class.

The predictive power of the improved PNN model is identical to that of the best model obtained using a genetic algorithm subjective feature selection routine and *k*-nearest neighbor classification fitness evaluator, reported by McElroy et al.[16] The descriptors included in the *k*-NN model are given in Table 3, and the associated confusion matrix is presented in Table 4. The training set performance of the *k*-NN model matched the original PNN model with respect to the inactives (88 of 106 predicted correctly) and the improved PNN model with respect to the actives (183 of 198 correct). In both the *k*-NN model and the improved PNN model, however, the prediction set consisted of one false negative and two false positives. Only one of the false positives is common to both the *k*-NN model and the PNN model.

The seven descriptors included in the *k*-NN model include four topological descriptors and three hybrid geometric/electronic descriptors. Two descriptors in the *k*-NN model are found in the PNN models: MOLC-9 and PND-5. The two additional topological descriptors in the *k*-NN model are 1SP3-1 and 2SP3-1. These are counts of sp^3-hybridized carbon atoms bonded to one and two other carbon atoms, respectively. These encode information about the carbon backbone in the molecule as does MDE-33 in the PNN models. Two of the hybrid descriptors in the k-NN model, SAAA-3 and CHAA-1, are CPSA descriptors and convey information about the hydrogen bonding capability of the molecules. SAAA-3 is the surface area of the hydrogen bond acceptor atoms divided by the total molecular surface area and CHAA-1 is the sum of the atomic charges of the hydrogen bond acceptor atoms. The counterpart to these in the PNN models is SCAA-2. The third hybrid descriptor chosen for this model is WHIM-42, a weighted holistic invariant molecular descriptor.[26] WHIM descriptors are designed to capture 3D information regarding molecular size, shape, and symmetry and atom distribution with respect to the molecule as a whole. WHIM-42 is the first component accessibility directional WHIM index weighted by atomic Sanderson electronegativities and conveys information about the distribution of electronic charge within the molecule. This is not unreasonable, as electrostatic interactions are important in the process of ligand binding in an active site.

To show that the results obtained for the PNN models were not due to chance correlations, a randomizing experiment was performed. The first part of the experiment involved randomly scrambling the dependent variable, in this case the HSEH inhibition. The second part of the experiment was an attempt to construct a PNN model using the same methodology as was used to build the actual PNN model but using the scrambled dependent variable data. Uniform priors and equal costs of misclassification were used for the randomizing experiment. The best four-descriptor model found was only able to predict 72% of the inactives correctly and 63% of the actives correctly (66% overall) for the training set. For the prediction set, only 40% of the inactives and 64% of the actives (57% overall) were classified correctly. Based on the populations of actives and inactives in the training set, random classification should result in 55% of the molecules being correctly classified. This is consistent with the overall classification rate for the prediction set. These results show that the best PNN models were unlikely to have been found due to chance correlation effects.

Aqueous Solubility Data Set. 399 nitrogen- and oxygen-containing small organic compounds and their associated aqueous solubility values, reported as log S, were studied by McElroy and Jurs.[17] The compounds' log *S* values ranged from −8.77 to 1.57, with a mean of −2.11 log units. The molecular weights ranged from 53.1 to 959.2 with a mean of 190.5 amu. This set of 399 compounds was subdivided into a 348-member TSET (from the original 298-member TSET and 50-member CVSET) and a 51-member PSET.

Table 5. Descriptors Selected for the Aqueous Solubility GSA-GRNN Model

descriptor	type[a]	sigma	range	explanation[b]
WTPT-2	T	0.1067	1.67 to 2.15	molecular ID/number of heavy atoms
EAVE−2	T	0.0568	2.24 to 12.2	mean heteroatom e-state
MOMI-2	G	0.0334	38.5 to 14026	second principal moment of inertia
RPCS-1	G/E	0.0836	0.0 to 11.5	relative positively charged surface area
CTDH-0	G/E	0.0523	0 to 11	number of donatable hydrogen atoms

[a] T = topological; G = geometric; G/E = geometric/electronic hybrid. [b] WTPT-2, molecular ID divided by the number of heavy atoms; EAVE-2, mean heteroatom electrotopological state index; MOMI-2, second principal moment of inertia; RPCS-1, surface area of the most positively charged atom × (charge of the most positively charged atom/total positive charge); CTDH-0, number of donatable hydrogen atoms.

Table 6. Descriptors Selected for the Aqueous Solubility GA-MLFN Model

descriptor	type[a]	range	explanation[b]
KAPA-6	T	0.0 to 18.1	atom-corrected third-order κ-index
NO-3	T	0 to 16	number of oxygen atoms
NN-4	T	0 to 6	number of nitrogen atoms
NDB-13	T	0 to 6	number of double bonds
WTPT-3	T	2.35 to 43.2	sum of path weights from heteroatoms
MDE−44	T	0.0 to 83.3	distance-edge between 4° carbon atoms
SYMM-25	T/G	0.05 to 1.0	geometric symmetry index
GEOM-1	G	0.66 to 48.4	first geometric moment
DPSA-2	G/E	64.3 to 4334	difference in partial surface areas
CHDH-1	G/E	0.0 to 2.32	charge on donatable hydrogens
SAAA-3	G/E	0.10 to 62.1	surface area of acceptor atoms

[a] T = topological; G = geometric; T/G = topological/geometric hybrid; G/E = geometric/electronic hybrid. [b] KAPA-6, third-order kappa index using atom and bond type corrections; NO-3, number of oxygen atoms in the molecule; NN-4, number of nitrogen atoms in the molecule; NDB-13, number of double bonds in the molecule; WTPT-3, sum of path weights from heteroatoms; MDE-44, molecular distance-edge index between quaternary carbons; SYMM-25, geometric symmetry through five bonds; GEOM-1, first geometric moment; DPSA-2, difference in charge-weighted partial surface areas; CHDH-1, sum of charges on donatable hydrogens; SAAA-3, sum of surface areas of hydrogen bond acceptor atoms.

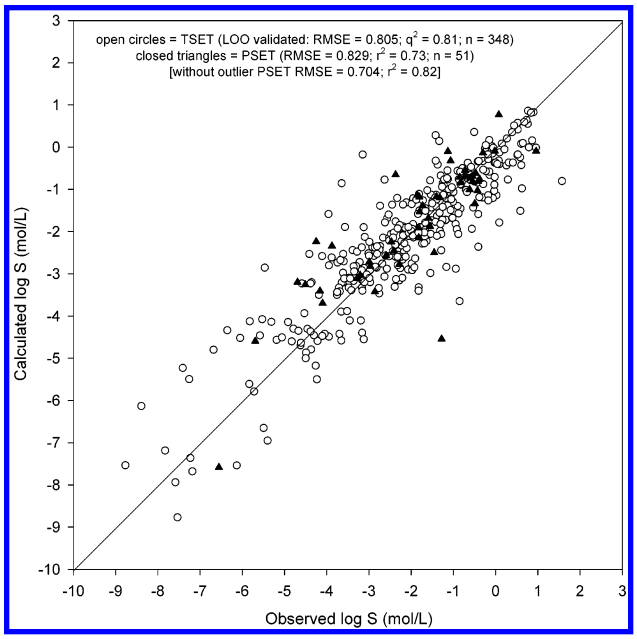

Figure 4. Plot of calculated log S (mol/L) versus the experimentally determined log S (mol/L) for the 5-descriptor GSA-GRNN model. The descriptors included in this model are listed in Table 5. Open circles represent TSET members and closed triangles represent PSET members.

Using the original 89-descriptor reduced pool, GRNN models containing from 2 to 7 descriptors were examined for overall quality. The five-descriptor subset that was selected as the best performer is presented in Table 6. The sigma values ranged from 0.0334 to 0.1067. This model is presented in Figure 5 and has a training set LOO root-mean-square error (RMSE) of 0.805 log units ($q^2 = 0.81$) and a prediction set RMSE of 0.829 log units ($r^2 = 0.73$). The prediction that showed the worst agreement with experiment was obtained for octachlorodibenzo-*p*-dioxin which was underpredicted by 3.27 log units. Although the descriptor values for this compound were within the ranges of those defined by the TSET for the descriptors in the model, it represented the most highly chlorinated compound in the TSET or PSET. Removal of this outlier reduced the prediction set error to 0.704 log units ($r^2 = 0.81$). Pairwise squared correlation coefficients (r^2) among the five descriptors ranged from 6.18×10^{-5} to 0.163 with a mean of 0.059. Pairwise squared correlation coefficients (r^2) between the five descriptors and the dependent variable were more pronounced and ranged from 0.0315 to 0.458 with a mean of 0.190.

Two of the descriptors in the model are topological, one is geometric, and two are geometric/electronic hybrids. The first topological descriptor, WTPT-2, is the molecular ID, as defined by Randić,[35] divided by the number of atoms in the molecule. The molecular ID is a *weighted path* descriptor, where the weights for each bond in a given path are determined by the connectivities of the two atoms defining the bond in the hydrogen-suppressed graph representing the molecule. The equation used to determine the molecular ID is given in eq 13.

$$ID = NA + \sum_{i=1}^{m}\prod_{j=1}^{n}(\delta_{ij,1}\cdot\delta_{ij,2})^{-1/2} \qquad (13)$$

Here, m is the number of paths in the molecule, n is the number of bonds in path i, and $d_{ij,1}$ and $d_{ij,2}$ are the connectivities (number of adjacent atoms) of atom 1 and 2 in bond ij. NA is the number of atoms in the molecule and is included because each atom by definition contributes 1 path of length zero (with weight = 1.0) to the total path count. Division of the molecular ID by the number of atoms reduces the contribution from paths of length zero to a constant (1.0) and normalizes the contributions from paths of longer length. Examination of eq 13 and of the molecular IDs of molecules having the same number of atoms reveals that the ID is larger when there are more short paths in a molecule and when the connectivities of the atoms remain small. For chemically relevant graphs, this means that structures that have more atoms in rings and structures with

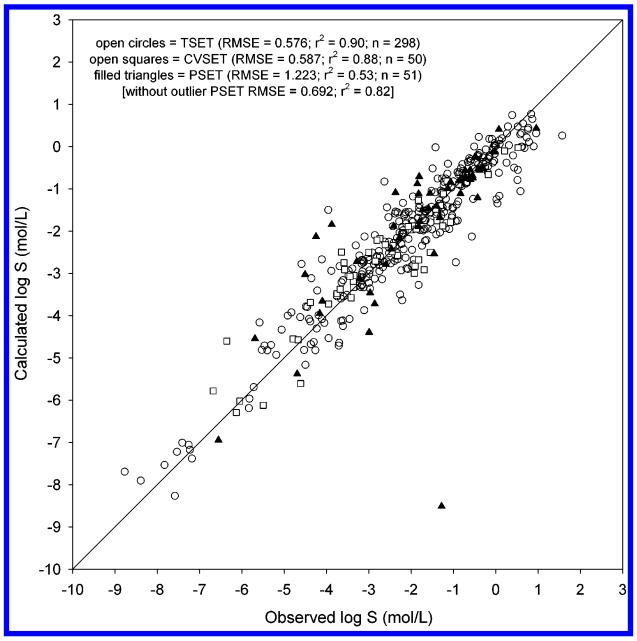

Figure 5. Plot of calculated log S (mol/L) versus the experimentally determined log S (mol/L) for the 11-descriptor GA-MLFN model. The descriptors included in this model are listed in Table 6. Open circles represent TSET members, open squares represent CVSET members, and closed triangles represent PSET members.

less branching will have higher molecular ID numbers. There is a modest inverse relationship ($r^2 = 0.458$) between WTPT-2 and the aqueous solubility indicating that molecules with higher WTPT-2 values are less soluble. This makes sense for two reasons. First, a hydrophilic functional group is better able to solubilize a hydrocarbon skeleton that is more highly branched and compact than one with the same number of atoms but less branched and more extended. Second, the presence of more rings in the structure lowers the aqueous solubility because many of these rings are aromatic and hydrophobic. The second topological descriptor in the model, EAVE-2, is the mean electrotopological state (e-state) index[36] for all heteroatoms. The e-state is calculated for each heavy atom in the molecule, and this descriptor combines information about the number of valence electrons and the number of neighboring atoms. Atoms with more neighbors are considered to be less accessible from the periphery of the molecule and thus less likely to participate in interactions at the molecule's surface. Atoms with a larger number of valence electrons will have a higher e-state index, as will those atoms with fewer neighboring atoms. It makes intuitive chemical sense that both of these factors are important for good aqueous solubility: more valence electrons translate to more and stronger polar interactions with the surrounding water molecules; fewer neighboring atoms mean more accessibility by the water molecules. The geometric descriptor included in the model is MOMI-2, the second principal moment of inertia. This is a measure of overall size and reflects the fact that in general larger molecules are less soluble than smaller molecules. Indeed, a significant decrease in log S was observed as MOMI-2 increased. The remaining two descriptors in the model represent geometric/electronic hybrid descriptors. The first of these is RPCS-1 or relative positive charged surface area, one of the CPSA descriptors. RPCS-1 is defined as the *surface area* of the most positively charged atom multiplied by the *charge* of the most positively charged atom divided by the total positive charge. Like EAVE-2, this descriptor contributes information about the molecule's ability to interact with the surrounding water molecules. The second hybrid descriptor included in the model is CTDH-0, the number of donatable hydrogen atoms in the molecule. This descriptor obviously contributes information about the ability of the molecule to interact with nearby water molecules via hydrogen bonding, a critical component of aqueous solubility.

The GRNN model presented here compares quite favorably to the MLFN model presented by McElroy and Jurs. The descriptors used in that 11−5−1 model (11 input nodes, 5 hidden layer nodes, and 1 output layer node) are presented in Table 6, and the predicted versus observed plot is presented in Figure 5. This model has a training set RMSE of 0.576 log units ($r^2 = 0.90$), a cross-validation set RMSE of 0.587 log units ($r^2 = 0.88$), and a prediction set RMSE of 1.223 log units ($r^2 = 0.53$). The prediction that showed the worst agreement with experiment, as in the GRNN model, was obtained for octachlorodibenzo-*p*-dioxin which was severely underpredicted by 7.23 log units. Removal of this outlier decreased the prediction set RMSE to 0.692 ($r^2 = 0.81$). The predictive power of the MLFN model (as measured by the PSET RMSE) is virtually identical to that of the GRNN model. A discussion of the descriptors included in this model will not be presented here but can be found in the original work. It will be noted, however, that many descriptors found in the MLFN model are closely related to those of the GRNN model. In particular, the descriptors CHDH-1 and SAAA-3 in the MLFN model and CTDH-0 in the GRNN both encode aspects of hydrogen bonding, DPSA-2 and RPCS-1 encode charged partial surface areas, GEOM-1 and MOMI-2 encode moments, and WTPT-3 and WTPT-2 both represent weighted paths. The TSET and CVSET RMS errors for the MLFN model (0.576 and 0.587 log units, respectively) are both significantly lower than the LOO-validated TSET RMS error for the GRNN model (0.805 log units). Aside from the differences in network architectures between the MLFN and GRNN, this difference in TSET RMSE is believed to be a result of two other potential factors. First, the way in which the cost function is calculated in each fitness evaluator is different. In the GRNN model, a LOO method is used to achieve internal validation. The cost function associated with a LOO method reflects the model's ability to predict each compound in the TSET without prior knowledge of that compound. This will generally result in higher TSET RMS errors. In addition, it is well-known that a model having a low squared cross-validated correlation coefficient (q^2) can still be quite predictive. However, Golbraikh and Tropsha[37] have shown that although a model might have a high q^2, an external prediction set is still necessary for proper model validation. In contrast, in a system where there is a fixed TSET and CVSET, the prediction for a given TSET member is obtained from a model that incorporates information about that training set member. This will tend to decrease the TSET RMS error, and this is important because eq 12 shows that the TSET RMS error is a major part of the overall cost function. In the TSET/CVSET system, only the CVSET members are predicted without the model's prior knowledge of them. In a MLFN, however, it is difficult to ascertain what a given training set member's contribution to the overall model is.

Compare this to the PNN or GRNN, where the contribution of a TSET member to the model is obvious (it is represented by an individual node in the pattern layer, which is clipped out during the training process). A second cause for the difference in TSET errors may be that the number of descriptors selected for the best MLFN model contains 11 descriptors, more than double the number selected for the best GRNN model. Due to computational time constraints, only MLFN models with 11 descriptors were considered. The MLFN model may thus be overtrained by including extra descriptors that code for features found primarily in the TSET and CVSET but that are not prominent in the PSET. The somewhat smaller RMS errors in the TSET and CVSET compared to that of the PSET for the MLFN model indicate that some overtraining may have taken place.

An additional randomizing experiment was performed to show that the results obtained for the GRNN model were not due to chance correlations. The aqueous solubility (log S) values were scrambled and models were generated in that same way as the original GRNN models were generated. The best five-descriptor model found had a TSET RMSE of 1.71 log units and a PSET RMSE of 1.74 log units. These results show that the best GRNN model was unlikely to have been found due to chance correlation effects.

CONCLUSIONS

This work has shown that the Probabilistic Neural Network (PNN) and its close relative, the Generalized Regression Neural Network (GRNN), are simple and powerful neural network techniques for use in Quantitative Structure−Activity Relationship (QSAR) and Quantitative Structure−Property Relationship (QSPR) studies. The PNN methodology was applied to classification problems, whereas the GRNN was applied to continuous function mapping problems. Effective PNN models were presented that identified molecules as potential human soluble epoxide hydrolase inhibitors using a binary classification scheme. A GRNN model was presented that predicted the aqueous solubility of nitrogen- and oxygen-containing small organic molecules. For each application, the network inputs consisted of a small set of descriptors that encode structural features at the molecular level. For the applications presented, the predictive power of the PNN and GRNN models was found to be equivalent to previously examined methodologies such as *k*-NN classification and MLFN function approximation but requiring significantly fewer input descriptors.

REFERENCES AND NOTES

(1) Specht, D. Probabilistic Neural Networks. *Neural Networks* **1990**, *3*, 109−118.

(2) Schiøler, H.; Hartmann, U. Mapping Neural Network Derived from the Parzen Window Estimator. *Neural Networks* **1992**, *5*, 903−909.

(3) Specht, D. A General Regression Neural Network. *IEEE T. Neural Networks* **1991**, *2*, 568−576.

(4) Kaiser, K. L. E.; Niculescu, S. P. Modeling Acute Toxicity of Chemicals to *Daphnia Magna*: a Probablistic Neural Network Approach. *Environ. Toxicol. Chem.* **2001**, *20*, 420−431.

(5) Kaiser, K. L. E.; Niculescu, S. P. On the PNN Modeling of Estrogen Receptor Binding Data for Carboxylic Acid Esters and Organochlorine Compounds. *Water Qual. Res. J. Can.* **2001**, *36*, 619−630.

(6) Niculescu, S. P.; Kaiser, K. L. E.; Schultz, T. W. Modeling the Toxicity of Chemicals to *Tetrahymena Pyriformis* Using Molecular Fragment Descriptors and Probabilistic Neural Networks. *Arch. Environ. Contam. Toxicol.* **2000**, *39*, 289−298.

(7) Magelssen, G. R.; Elling, J. W. Chromatography Pattern Recognition of Aroclors using Iterative Probabilistic Neural Networks. *J. Chromatogr. A* **1997**, *775*, 231−242.

(8) Zaknich, A. Characterization of Aluminum Hydroxide Particles from the Bayer Process using Neural Network and Bayesian Classifiers. *IEEE Trans. Neural Networks* **1997**, *8*, 919−931.

(9) Holmes, E.; Nicholson, J. K.; Tranter, G. Metabonomic Characterization of Genetic Variations in Toxicological and Metabolic Responses using Probabilistic Neural Networks. *Chem. Res. Toxicol.* **2001**, *14*, 182−191.

(10) Bathen, T. F.; Engan, T.; Krane, J.; Axelson, D. Analysis and Classification of Proton NMR Spectra of Lipoprotein Fractions from Healthy Volunteers and Patients with Cancer or CHD. *Anticancer Res.* **2000**, *20*, 2393−2408.

(11) Shaffer, R. E.; Rose-Pehrsson, S. L. Improved Probabilistic Neural Network Algorithm for Chemical Sensor Array Pattern Recognition. *Anal. Chem.* **1999**, *71*, 4263−4271.

(12) Yang, H.; Ring, Z.; Briker, Y.; McLean, N.; Friesen, W.; Fairbridge, C. Neural Network Prediction of Cetane Number and Density of Diesel Fuel from its Chemical Composition Determined by LC and GC-MS. *Fuel* **2002**, *81*, 65−74.

(13) Chtioui, Y.; Panigrahi, S.; Francl, L. A Generealized Regression Neural Network and its Application for Leaf Wetness Prediction to Forecast Plant Disease. *Chemometr. Intell. Lab. Syst.* **1999**, *48*, 47−58.

(14) Bathen, T. F.; Krane, J.; Engan, T.; Bjerve, K. S.; Axelson, D. Quantification of Plasma Lipids and Apolipoproteins by use of Proton NMR Spectroscopy, Multivariate and Neural Network Analysis. *NMR Biomed.* **2000**, *13*, 271−288.

(15) Moser, E. M.; Faller, C.; Pietrzko, S.; Eggimann, F. Modeling the Functional Performance of Plasma Polymerized Thin Films. *Thin Solid Films* **1999**, *356*, 49−54.

(16) McElroy, N. R.; Jurs, P. C.; Morisseau, C.; Hammock, B. D. QSAR and Classification of Murine and Human Soluble Epoxide Hydrolase Inhibition by Urea-like Compounds. *J. Med. Chem.*, submitted for publication.

(17) McElroy, N. R.; Jurs, P. C. Prediction of Aqueous Solubility of Heteroatom-Containing Organic Compounds from Molecular Structure. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1237−1247.

(18) Parzen, E. On Estimation of a Probability Density Function and Mode. *Ann. Math. Stat.* **1962**, *33*, 1065−1076.

(19) Cacoullos, T. Estimation of a Multivariate Density. *Ann. I. Stat. Math.* **1966**, *18*, 179−189.

(20) Masters, T. *Advanced Algorithms for Neural Networks: A C++ Sourcebook*; John Wiley and Sons: New York, 1995.

(21) Brent, R. *Algorithms for Minimization Without Derivatives*; Prentice Hall: Englewood Cliffs, NJ, 1973.

(22) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed; Cambridge University Press: 1992.

(23) Stewart, J. P. MOPAC: A Semiempirical Molecular Orbital Program. *J. Comput. Aid. Mol. Des.* **1990**, *4*, 1−105.

(24) Jurs, P. C.; Chow, J. T.; Yuan, M. Studies of Chemical Structure-Biological Activity Relations Using Pattern Recognition. In *Computer-Assisted Drug Design*; Olson, E. C., Christoffersen, R. E., Eds.; The American Chemical Society: Washington, DC, 1979.

(25) Stuper, A. J.; Brügger, W. E.; Jurs, P. C. *Computer-Assisted Studies of Chemical Structure and Biological Function*; John Wiley & Sons: New York, 1979.

(26) Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*; Mannhold, R., Kubinyi, H., Timmerman, H., Eds.; Methods and Principles in Medicinal Chemistry; Wiley-VCH: Weinheim, 2000.

(27) Stanton, D. T.; Jurs, P. C. Development and Use of Charged Partial Surface Area Structural Descriptors in Computer-Assisted Quantitative Structure−Property Relationship Studies. *Anal. Chem.* **1990**, *62*, 2323−2329.

(28) Stenberg, P.; Luthman, K.; Ellens, H.; Lee, C. P.; Smith, P. L.; Lago, A.; Elliot, J. D.; Arturrson, P. *Pharm. Res.* **1999**, *16*, 1520.

(29) Todeschini, R.; Consonni, V.; Pavan, M. *DRAGON*, v. 2.1; Talete SRL: Milan.

(30) Masters, T. *Practical Neural Network recipes in C++*; Academic Press: Boston, 1993.

(31) Balaban, A. T. Highly Discriminating Distance-Based Topological Index. *Chem. Phys. Lett.* **1982**, *89*, 399−404.

(32) Randic, M. On Characterization of Molecular Branching. *J. Am. Chem. Soc.* **1975**, *97*, 6609−6615.

(33) Liu, S.; Cao, C.; Li, Z. Approach to Estimation and Prediction for Normal Boiling Point (NBP) of Alkanes Based on a Novel Molecular Distance-Edge (MDE) Vector, *λ*. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 387−394.

(34) Gupta, S.; Singh, M.; Madan, A. K. Superpendentic Index: A Novel Topological Descriptor for Predicting Biological Activity. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 272−277.

(35) Randic, M. On Molecular Identification Numbers. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 164−175.
(36) Kier, L. B.; Hall, L. H. An Electrotopological State Index for Atoms in Molecules. *Pharmaceut. Res.* **1990**, *7*, 801−807.
(37) Golbraikh, A.; Tropsha, A. Beware of q^2! *J. Mol. Graphics Mod.* **2002**, *20*, 269−276.

CI020039I