Aquat Ecol (2007) 41:491–508
DOI 10.1007/s10452-007-9093-3

Applications of artificial neural networks predicting macroinvertebrates in freshwaters

Peter L. M. Goethals · Andy P. Dedecker · Wim Gabriels · Sovan Lek · Niels De Pauw

Received: 28 March 2007 / Accepted: 2 April 2007 / Published online: 9 May 2007

Abstract To facilitate decision support in freshwater ecosystem protection and restoration management, habitat suitability models can be very valuable. Data driven methods such as artificial neural networks (ANNs) are particularly useful in this context, seen their time-efficient development and relatively high reliability. However, specialized and technical literature on neural network modelling offers a variety of model development criteria to select model architecture, training procedure, etc. This may lead to confusion among ecosystem modellers and managers regarding the optimal training and validation methodology. This paper focuses on the analysis of ANN development and application for predicting macroinvertebrate communities, a species group commonly used in freshwater assessment worldwide. This review reflects on the different aspects regarding model development and application based on a selection of 26 papers reporting the use of ANN models for the prediction of macroinvertebrates. This analysis revealed that the applied model training and validation methodologies can often be improved and moreover crucial steps in the modelling process are often poorly documented. Therefore, suggestions to improve model development, assessment and application in ecological river management are presented. In particular, data pre-processing determines to a high extent the reliability of the induced models and their predictive relevance. This also counts for the validation criteria, that need to be better tuned to the practical simulation requirements. Moreover, the use of sensitivity methods can help to extract knowledge on the habitat preference of species and allow peer-review by ecological experts. The selection of relevant input variables remains a critical challenge as well. Model coupling is a missing crucial step to link human activities, hydrology, physical habitat conditions, water quality and ecosystem status. This last aspect is probably the most valuable aspect to enable decision support in water management based on ANN models.

P. L. M. Goethals (✉) · A. P. Dedecker · W. Gabriels · N. De Pauw
Department of Applied Ecology and Environmental Biology, Laboratory of Environmental Toxicology and Aquatic Ecology, Ghent University, J. Plateaustraat 22, Ghent 9000, Belgium
e-mail: peter.goethals@UGent.be

S. Lek
CESAC UMR 5576, CNRS-University Paul Sabatier, 118, route de Narbonne, Toulouse cedex 31062, France

Keywords Data driven models · Decision support systems · Ecological modelling · Habitat suitability models · Knowledge extraction · Water management

Introduction

Artificial neural networks (ANNs) are non-linear mapping structures that can be applied for predictive

modelling and classification. Various types of neural networks exist, suitable to solve different kinds of problems. The choice of the type of network depends on the nature of the problem to be solved. The most popular ANNs are multi-layer feed-forward neural networks with the backpropagation algorithm, i.e. backpropagation networks (Rumelhart et al. 1986; Hagan et al. 1996) and Self-organizing Maps, i.e. Kohonen networks (SOMs) (Kohonen 1982). However, the latter are mainly interesting for clustering data and will not be further discussed in this review. A backpropagation network is based on the 'supervised' procedure and can be used for the development of predictive models. The network constructs a model based on examples of data with known outputs. It has to build up the model solely from the examples presented, which are together assumed to contain the information necessary to establish the relation. An example can be the relation between the presence/absence or abundance of macroinvertebrate taxa (such as Gammaridae (Crustacea, Amphipoda), Baetidae (Insecta, Ephemeroptera), Chironomidae (Insecta, Diptera)) and river characteristics such as dissolved oxygen, pH, flow velocity, river depth, ...

A backpropagation network typically comprises three types of neuron layers: an input layer, one or more hidden layers and an output layer, each including one or more neurons (Fig. 1). In a backpropagation network neurons from one layer are connected to all neurons in the subsequent layer, but no lateral connections within a layer, nor feedback connections are possible. With the exception of the input neurons, which only connect one input value with its associated weight values, the net input for each neuron is the sum of all input values x_n, each multiplied by its weight w_{jn}, and a bias term z_j which may be considered as the weight from a supplementary input equalling one (Fig. 2):

$$a_j = \sum w_{ji} x_i + z_j \quad (1)$$

The output value, y_j, can be calculated by feeding the net input into the transfer function of the neuron:

$$y_j = f(a_j) \quad (2)$$

Before training, the values of the weights and biases are initially set to small random numbers. Subsequently, a set of input/output vector pairs is presented to the network. For each input vector, the output vector is calculated by the neural network model, and an error term is calculated for the outputs of all hidden and output neurons, by comparing the calculated output vector and the actual output vector. Using this error term, the weights and biases are updated in order to decrease the error, so future outputs are more likely to be correct. This procedure is repeated until the errors become small enough or a predefined maximum number of iterations is reached. This iterative process is termed 'training'. After the training, the ANN can be validated using independent data. A more detailed description can be found in Lek and Guégan (1999).

This paper analyses ANN development procedures from both technical and ecological perspectives and studies ANN applications to predict macroinvertebrate communities in aquatic ecosystems, as these communities have been proven to be good indicators

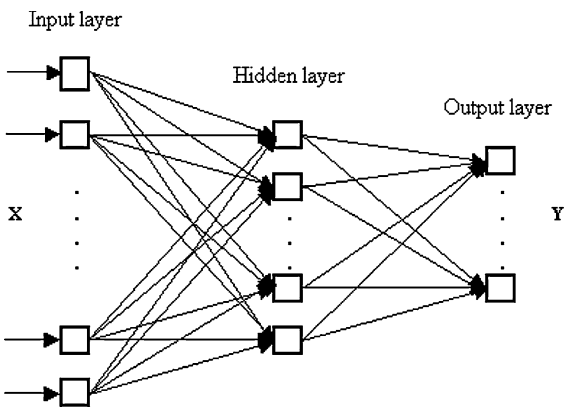

Fig. 1 Schematic illustration of a three-layered feed-forward neural network consisting of one input layer, one hidden layer and one output layer

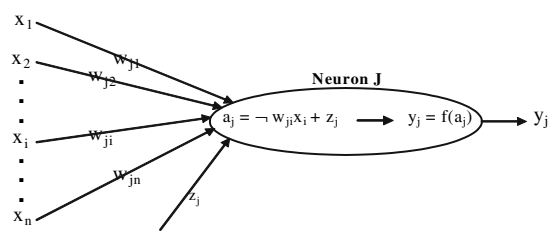

Fig. 2 Scheme of a neuron in a backpropagation network receiving input values from n neurons, each associated with a weight, as well as a bias z_j. The resulting output value y_j is computed according to the presented equations

for the assessment of surface waters. Based on this overview, suggestions to improve model development, assessment and application in ecological river management are presented.

Predictive ANN development

Data analysis and processing

Data processing

Generally, most variables span different numerical ranges. In order to ensure that all variables can receive equal attention during the training process, standardization is recommended. In addition, the variables have to be scaled in such a way as to be commensurate with the limits of the activation functions used in the output layer (Maier and Dandy 2000). Several authors (Chon et al. 2001, 2002; Gabriels et al. 2007; Obach et al. 2001; Park et al. 2003a, b; Schleiter et al. 1999; Schleiter et al. 2001; Wagner et al. 2000) proportionally normalized the data between zero and one [0 1] in the range of the maximum and minimum values, while Dedecker et al. (2004, 2005a) and Gabriels et al. (2002) used the interval [−1 1]. Moreover the division of the dataset in folds for cross-validation is crucial for a good model training and evaluation. However this fold optimization process is not described in the analysed publications. This aspect will be further discussed in the paragraphs on validation.

Band width

Lek and Guégan (1999) stated that ANN models are built solely from the examples presented during the training phase, which are together assumed to implicitly contain the information necessary to establish the relation between input and output. As a result, ANNs are unable to extrapolate beyond the range of the data used for training. Consequently, poor and unreliable predictions can be expected when simulations have to be made based on values outside of the range of those used for training (Maier and Dandy 2000). Dedecker et al. (2005a) tested the sensitivity and robustness of the ANN models when data, containing variables with values beyond the range of the data for initial training, was added. Therefore, they created a virtual dataset based on ecological expert knowledge to introduce a small set of instances with 'extreme' values to the model. Their work demonstrated that coupling of data driven modelling techniques with expert knowledge can be very valuable.

Input variable selection

Data driven approaches, such as ANN models, have the ability to determine which model inputs are critical to obtain the best possible predictions within the presented dataset. However, presenting a large number of inputs to ANN models, and relying on the network to determine the critical model inputs, usually increases network size. This has a number of disadvantages, for example decreasing processing speed and increasing the amount of data required to estimate the network parameters efficiently (Maier and Dandy 2000). In this way, selection of input variables can considerably reduce the model calculation time, but also the related field data collection efforts.

According to Walczak and Cerpa (1999), three steps can be followed to determine the optimal set of input variables. The first one is to perform standard knowledge acquisition. Typically, this involves consultation with multiple domain experts. Walczak (1995) has indicated the requirement for extensive knowledge acquisition utilizing domain experts to specify ANN input variables. The primary purpose of the knowledge acquisition phase is to guarantee that the input variable set is not under-specified, providing all relevant domain criteria to the ANN. Once a base set of input variables is defined through knowledge acquisition, the set can be pruned to eliminate variables that contribute noise to the ANN and consequently reduce ANN generalization performance. ANN input variables should not be highly correlated. Correlated variables degrade ANN performance by interacting with each other as well as other elements to produce a biased effect. From an ecological point of view, relationships between environmental variables and taxonomic richness should be considered with caution, as these analyses, based on correlation, do not necessarily involve relevant ecological processes. However, the only way to establish reliable causal relationships between input and output, is to use experimental designs

(Beauchard et al. 2003). For macroinvertebrates, this can be done on the basis of spiking tests in situ or with artificial river systems for instance. These are however both very time consuming, expensive and are moreover limited regarding their practical set-up (collection of individuals, controlling physical–chemical conditions,...). A first filter to help identify 'noise' variables is to calculate the correlation of pairs of variables. If two variables are strongly correlated, then one of these two variables may be removed without adversely affecting the ANN performance. The cut-off value for variable elimination is a heuristic value and must be determined separately for every ANN application, but any correlation absolute value of 0.20 or higher indicates a probable noise source to the ANN (Walczak and Cerpa 1999). When a significant correlation ($P < 0.01$) was found between two variables, Brosse et al. (2003) removed the one accounting for less variation in the single-scale models.

In addition, there are distinct advantages in using analytical techniques to help determine the inputs for ANN models (Maier and Dandy 2000). However, these methods can merely be applied when large datasets are available. Beauchard et al. (2003), Obach et al. (2001), Schleiter et al. (1999, 2001) used a stepwise procedure to identify the most influential variables. In this approach, separate networks are trained for each input variable. The network performing best is retained and the effect of adding each of the remaining inputs in turn is assessed. This process is repeated for three, four, five, etc. input variables, until the addition of extra variables does not result in a significant improvement in model performance. On the other hand, one can start with all the available variables and remove one by one the least important ones (e.g. Beauchard et al. 2003; Gabriels et al. 2007). Disadvantages of these approaches are that they are computationally intensive and that they are unable to capture the importance of certain combinations of variables that might be insignificant on their own. Obach et al. (2001), Schleiter et al. (1999, 2001) and Wagner et al. (2000) applied a special variant of the backpropagation network type, the so-called senso-net, to determine the most important input variables (sensitivity analysis). Senso-nets include an additional weight for each input neuron representing the relevance (sensitivity) of the corresponding input parameter for the neural model. The sensitivities are adapted during the training process of the network. Appropriate subsets of potential input variables can be selected according to these sensitivities. A third frequently used technique is genetic algorithms (e.g. D'heygere et al. 2006; Obach et al. 2001; Schleiter et al. 2001). This technique automatically selects the relevant input variables (Goldberg 1989). However, the dataset needs to contain a sufficient number of instances to enable the application of these methods.

Model architecture

According to Haykin (1999), generalization capability of a neural network is influenced by three factors: the size of the training set and how representative it is of the environment of interest, the architecture of the neural network, and the complexity of the problem studied. The architecture is the only of these three factors that can be influenced in the modelling process, making it a crucial step, which should be considered carefully.

Walczak and Cerpa (1999) distinguish four design criteria for ANNs which should be decided upon in subsequent steps: knowledge-based selection of input values, selection of a learning method, design of the number of hidden layers and selection of the number of hidden neurons for each layer. Input variable selection was already discussed in the previous section.

Learning method

The suitability of a particular method is often a trade-off between performance and calculation time. The majority of the ANNs used for prediction are trained with the backpropagation method (e.g. Cherkassky and Lari-Najafi 1992; Maier and Dandy 2000). Due to its generality (robustness) and ease of implementation, backpropagation is the best choice for the majority of ANN-based predictions. Backpropagation is the superior learning method when a sufficient number of relatively noise-free training examples are available, regardless of the complexity of the specific domain problem (Walczak and Cerpa 1999). Although backpropagation networks can handle noise in the training data (and may actually generalize better if some noise is present in the training data), too many erroneous training values may prevent the

ANN from learning the desired model. When only a few training examples or very noisy training data are available, other learning methods should be selected instead of backpropagation (Walczak and Cerpa 1999). Radial basis function networks perform well in domains with limited training sets (Barnard and Wessels 1992 in Walczak and Cerpa 1999) and counterpropagation networks perform well when a sufficient number of training examples is available, but may contain very noisy data (Fausett and Elwasif 1994 in Walczak and Cerpa 1999).

In order to optimize the performance of backpropagation networks, it is essential to note that the performance is a function of several internal parameters including the transfer function, error function, learning rate and momentum term. The most frequently used transfer functions are sigmoid ones such as the logistic and hyperbolic tangent functions (Maier and Dandy 2000). However, other transfer functions may be used, such as hard limit or linear functions (Hagan et al. 1996). The error function is the function that is minimized during training. The most commonly used error function is the mean squared error (MSE) function. However, in order to obtain optimal results, the errors should be independently and normally distributed, which is not the case when the training data contain outliers (Maier and Dandy 2000). To overcome this problem, Liano (1996) proposed the least mean log squares (LMLS) error function. The learning rate is directly proportional to the size of the steps taken in weight space. Traditionally, learning rates remain fixed during training (Maier and Dandy 2000) and optimal learning rates are determined by trial and error. However, heuristics have been proposed which adapt the learning rate as training progresses to keep the learning step size as large as possible while keeping learning stable (Hagan et al. 1996). A momentum term is usually included in the training algorithm in order to improve learning speed (Qian 1999) and convergence (Hagan et al. 1996). The momentum term should be less than 1.0, otherwise the training procedure does not converge (Dai and Macbeth 1997). Dai and Macbeth (1997) suggest a learning rate of 0.7 with a momentum term of at least 0.8 and smaller than 0.9 or a learning rate of 0.6 with a momentum term of 0.9. Qian (1999) derived the bounds for convergence on learning rate and momentum parameters, and demonstrated that the momentum term can increase the range of learning rates over which the system converges.

Number of hidden layers

A greater number of hidden layers enables an ANN to improve its closeness-of-fit, while a smaller quantity improves the smoothness or extrapolation capabilities of the ANN (Walczak and Cerpa 1999). Theoretically, an ANN with one hidden layer can approximate any function as long as sufficient neurons are used in the hidden layer (Hornik et al. 1989). Flood and Kartam (1994) suggest using two hidden layers as a starting point. However, it must be stressed that optimal network geometry is highly problem dependent and therefore trial and error is in most cases the only option to determine the optimal number of hidden layers based on 'experience' with the dataset.

Number of hidden neurons

The number of neurons in the input layer is fixed by the number of model inputs, whereas the number of neurons in the output layer equals the number of model outputs. The critical aspect however is the choice of the number of neurons in the hidden layer(s). More hidden neurons result in a longer training period, while fewer hidden neurons provide faster training at the cost of having fewer feature detectors (Bebis and Georgiopoulos 1994). For two networks with similar errors on training sets, the simpler one (the one with fewer hidden units) is likely to produce more reliable predictions on new cases, while the more complex model implies an increased chance of overfitting on the training data and reducing the model's ability to generalize on new data (Hung et al. 1996; Özesmi and Özesmi 1999). Hecht-Nielsen (1987) showed that any continuous function with N_i inputs in the range [0 1] and N_o outputs can be represented exactly by a feedforward network with $2N_i + 1$ hidden neurons.

Various authors propose rules of thumb for determining the number of hidden neurons. Some of these rules are based on the number of input and/or output neurons, whereas others are based on the number of training samples available. Walczak and Cerpa (1999) warn that these heuristics do not use domain knowledge for estimating the quantity of

hidden nodes and may be counterproductive. Table 1 shows the rules that suggest the number of hidden neurons based on the number of input (N_i) and/or output (N_o) nodes.

Some authors suggest rules to determine the necessary number of training samples (S) based on the number of connection weights. Given the number of training samples is fixed, inverting these rules gives an indication of the maximum number of connection weights to avoid overfitting (Table 2).

The number of hidden neurons necessary can be calculated given the number of connection weights and the number of input and output neurons.

Rules of thumb are clearly divergent and when selecting the number of hidden neurons, one should take both S and N_i into account. Assuming only one hidden layer is used, the number of connection weights should not exceed S/10 and the number of hidden neurons should be at least, roughly, $(N_i + N_o)$/2. Evidently, in order to be able to meet both constraints, the number of training samples has to be sufficiently large.

According to Walczak and Cerpa (1999), the number of hidden neurons in the last layer should be set equal to the number of decision factors used by domain experts to solve the problem. Decision factors are the distinguishable elements that serve to form the unique categories of the input vector space. The number of decision factors is equivalent to the number of heuristic rules or clusters used in an expert system (Walczak and Cerpa 1999).

Alternatively, techniques for automatically selecting ANN architecture with the required number of hidden units may be used. Such techniques were proposed by e.g. Bartlett (1994), Nabhan and Zomaya (1994) and Anders and Korn (1999).

Table 1 Rules suggesting the number of hidden neurons based on the number of input (N_i) and/or output (N_o) nodes

Rule	Reference
(2/3) * N_i	Wang (1994)
0.75 * N_i	Lenard et al. (1995)
0.5 * (N_i + N_o)	Piramuthu et al. (1994)
2 * N_i + 1	Fletcher and Goss (1993), Patuwo et al. (1993)
2 * N_i or 3 * N_i	Kanellopoulos and Wilkinson (1997)

Table 2 Indication of the maximum number of connection weights to avoid overfitting based on the number of training samples (S)

Maximum number of connection weights	Reference
S	After Rogers and Dowla (1994)
S/2	After Masters (1993)
S/4	After Walczak and Cerpa (1999)
S/10	After Weigend et al. (1990)
S/30	After Amari et al. (1997)

Model validation and interpretation

Model validation

To validate the model performance, a set with data independent from the training set is required (Lek and Guégan 1999; Maier and Dandy 2000). In the validating phase, the input patterns are fed into the network and the desired output patterns are compared with those given by the ANN model. The agreement or disagreement of these two sets gives an indication of the performance. As mentioned before, the data used for validation must be within the range of the data used for training. Therefore, this is a key element to take care of during data preparation (data stratification). It is also imperative that the training and validation sets are representative of the same population. The optimal solution is to have two independent databases (Lek and Guégan 1999). In this way, the first can be used for training and the second for validation of the model (e.g. Mastrorillo et al. 1998; Obach et al. 2001). However, when limited data are available, it might be necessary to split the available data into a training and a validation set. A frequently used procedure, is the k-fold cross-validation method (e.g. Dedecker et al. 2002, 2004, 2005a, b, 2007; D'heygere et al. 2006). In this case, the data set is equally divided into k parts. The ANN model is trained with $(k-1)$ parts, and validated with the remaining part. This is repeated k times. The variance of the performance results gives an indication of the robustness of the induced model(s). To determine the optimal k-value, it is best to try out a set of combinations of k between 3 and 10, and find a balance between the robustness and reliability of the developed models. In many software packages, 10-fold cross validation is used as default setting, however, when the dataset is relatively small, lower

k-values can result in more robust ANN-models, but with a relatively low performance. Therefore, a high *k*-value is recommended for small datasets. Beauchard et al. (2003), Brosse et al. (2001, 2003) and Guégan et al. (1998) for example used the 'leave-one-out' cross-validation method (Efron 1983). This procedure is a special case of *k*-fold cross-validation, where *k* equals the sample size (number of instances in the dataset).

Performance measures

Based on the output, different performance measures can be distinguished. When presence/absence of the macroinvertebrates is predicted, the percentage of correctly classified instances (CCI) is frequently used to assess model performance. There is however clear evidence that this CCI is affected by the frequency of occurrence of the test organism(s) being modelled (Fielding and Bell 1997; Manel et al. 1999). Among the different measures, which are based on a confusion matrix (Table 3), proposed to assess the performance of presence/absence models (Table 4), Fielding and Bell (1997) and Manel et al. (1999) recommended Cohen's kappa as a reliable performance measure, since the effect of prevalence on Cohen's kappa appeared to be negligible (e.g. Dedecker et al. 2004, 2005a; D'heygere et al. 2006). Therefore, kappa provides a more reliable representation of model performance (Cohen 1960). However, these kappa values also represent the information content of the dataset, and each dataset has a limit regarding extractable information. Consequently, differences in kappa threshold values (and evaluation classes) can be expected between disciplines in general and datasets in particular (Gabriels et al. 2007). In an ecological context, Randin et al. (2006) assess kappa values as follows: 0.00–0.40: poor; 0.40–0.75: good; 0.75–1.00: excellent. These classes can consequently be used to assess model reliability, but can not be used to evaluate the modelling method in an absolute manner, only on a relative basis (e.g. comparison of models based on the same dataset). In this context, the reliability of the monitoring procedure of the input and output variables has to be taken into account. Comparison of monitoring reliability with model performance can give valuable insight in how reliable models can potentially be and how much they can be improved.

When the output of the ANN model consists of the species abundance, richness, diversity, density or a derived index, commonly used performance measures are the correlation (r) or determination (R^2) coefficient and the (root) mean squared error ((R)MSE) or a derivative between observed (O) and predicted (P) values (Table 5).

Model interpretation

Although in many studies ANNs have been shown to exhibit superior predictive power compared to traditional approaches, they have also been labelled as a

Table 3 The confusion matrix as a basis for the performance measures with true positive values (TP), false positives (FP), false negatives (FN) and true negative values (TN)

		Observed	
		+	−
Predicted	+	a	b
	−	c	d

Table 4 Measures based on the confusion matrix to assess the performance of presence/absence models (after Fielding and Bell 1997)

Performance measure	Calculation
CCI	$\frac{(a+d)}{N}$
Misclassification rate	$\frac{(b+c)}{N}$
Sensitivity	$\frac{a}{(a+c)}$
Specificity	$\frac{d}{(b+d)}$
Positive predictive power	$\frac{a}{(a+b)}$
Negative predictive power	$\frac{d}{(c+d)}$
Odds-ratio	$\frac{(ab)}{(cd)}$
Cohen's kappa	$\frac{[(a+d)-(((a+c)(a+b)+(b+d)(c+d))/N)]}{[N-(((a+c)(a+b)+(b+d)(c+d))/N)]}$
NMI	$\frac{[-a.\ln(a)-b.\ln(b)-c.\ln(c)-d.\ln(d)+(a+b).\ln(a+b)+(c+d).\ln(c+d)]}{[N.\ln(N)-((a+c).\ln(a+c)+(b+d).\ln(b+d))]}$

CCI is the percentage correctly classified instances, NMI is the normalised mutual information statistic and N is the total number of instances

Table 5 Measures based on observed (O) and predicted (P) values to assess the performance of ANN models using abundance, richness, diversity, density or a derived index as model output

N is the total number of instances

Performance measure	Calculation
Correlation coefficient (r)	$\frac{\sum(P\times O)-((\sum P\times\sum O)/N)}{\sqrt{(\sum P^2-((\sum P)^2/N))\times(\sum O^2-((\sum O)^2/N))}}$
Determination coefficient (R^2)	$\left(\frac{\sum(P\times O)-((\sum P\times\sum O)/N)}{\sqrt{(\sum P^2-((\sum P)^2/N))\times(\sum O^2-((\sum O)^2/N))}}\right)^2$
Root mean squared error (RMSE)	$\sqrt{\frac{1}{N}\sum(P-O)^2}$
Mean squared error (MSE)	$\frac{1}{N}\sum(P-O)^2$

''black box'' because they provide little explanatory insight into the relative influence of the independent variables in the prediction process (Olden and Jackson 2002). This lack of explanatory power is a major concern to ecologists since the interpretation of statistical models is desirable for gaining knowledge of the causal relationships driving ecological phenomena. As a consequence, various authors have explored this problem and proposed several algorithms to illustrate the role of variables in ANN models. Sensitivity analysis is frequently used (Brosse et al. 2003; Chon et al. 2001; Dedecker et al. 2002, 2005b; Guégan et al. 1998; Hoang et al. 2001, 2002; Laë et al. 1999; Marshall et al. 2002; Mastrorillo et al. 1997a; Olden and Jackson 2002) and is based on a successive variation of one input variable while the others are kept constant at a fixed value (Lek et al. 1995, 1996a, b). The 'Perturbation' method (Yao et al. 1998; Scardi and Harding 1999) assesses the effect of small changes to each input on the neural network output (e.g. Park et al. 2003a; Gevrey et al. 2003; Dedecker et al. 2005b, 2007). This method can thus be seen as a 'local' sensitivity analysis. Gevrey et al. (2003), Dedecker et al. (2005b, 2007) and Beauchard et al. (2003) used the 'PaD' method (Dimopoulos et al. 1995; Dimopoulos et al. 1999) which consists in a calculation of the partial derivatives of the output according to the input variables. Several authors (Brosse et al. 1999, 2001, 2003; Dedecker et al. 2005b, 2007; Gevrey et al. 2003; Mastrorillo et al. 1997b; Olden and Jackson 2002) applied Garson's algorithm (Garson 1991; Goh 1995). This algorithm is based on a computation using the connection weights. Gevrey et al. (2003) and Dedecker et al. (2005b, 2007) applied the 'Stepwise' procedure, as discussed earlier, to identify the most influential variables. Özesmi and Özesmi (1999) described the Neural Interpretation Diagram (NID) to provide a visual interpretation of the connection weights among neurons. The relative magnitude of each connection weight is represented by line thickness and line shading represents the direction of the weight. Olden and Jackson (2002) proposed a randomization test for input–hidden–output connection weight selection in ANN models. By eliminating connection weights that do not significantly differ from random, they simplified the interpretation of neural networks by reducing the number of axon pathways that have to be examined for direct and indirect (i.e. interaction) effects on the response variable, for instance when using NIDs. Olden et al. (2004) compared these methodologies using a Monte Carlo simulation experiment with data exhibiting defined numeric relationships between a response variable and a set of independent predictor variables. Using simulated data with known properties, they could accurately investigate and compare the different approaches under deterministic conditions and provide a robust comparison of their performance.

Model optimization

Traditionally, optimal network geometries were searched for by trial and error (Maier and Dandy 2000). However, a number of systematic approaches for determining optimal network geometry have been proposed, including pruning and constructive algorithms. The basic idea of pruning algorithms is to start with a network that is large enough to capture the desired input–output relationship and to subsequently remove or disable unnecessary weights and/or neurons. A review of pruning algorithms is given by Reed (1993). Constructive algorithms approach the problem of optimizing the number of hidden layer neurons from the opposite direction to

pruning algorithms. The smallest possible network is used at the start. Hidden layer neurons and connections are then added one at a time in an attempt to improve model performance. A review of constructive algorithms is given by Kwok and Yeung (1997a). Several disadvantages of these approaches are mentioned in literature (Maier and Dandy 2000). For example, the networks generally have to be trained several times, i.e. each time a hidden neuron is added or deleted (Kwok and Yeung 1997b). It has also been suggested that the pruning and constructive algorithms are susceptible to becoming trapped in structural local optima (Angeline et al. 1994). In addition, they 'only investigate restricted topological subsets rather than the complete class of network architectures' (Angeline et al. 1994). Algorithms based on evolutionary programming and genetic algorithms have been proposed to overcome these problems and have successfully been used to determine optimal network architecture (e.g. Fang and Xi 1997; Kim and Han 2000; Zhao et al. 2000; Wicker et al. 2002). Evolutionary approaches are significantly different from the previous techniques described. They produce more robust solutions because they use a population of networks in the search process. A complete review of the use of evolutionary algorithms in neural networks is given by Yao (1993). Beside the optimization of the network geometry, input variable selection can also be seen as model optimization. However, this has already been discussed in Section ''Input variable selection''.

Applications of macroinvertebrate predictions using ANNs in water management

This paper focussed on macroinvertebrates. However, the general aspects are very similar for other freshwater communities regarding model development approaches. However, differences can be expected regarding reliabilities as a result of natural dynamics (algae blooms), behavioural complexity (e.g. fish migration), monitoring methods,... Also, the relevant input variables will differ among communities. For algae nutrients and climate will play a crucial role, whereas for fish the habitat related variables are essential. For fish, moreover age dependent models might be necessary, since depending on the age, different habitat conditions are preferred. For algae often time series are used for predictions (e.g. Recknagel et al. 2006).

Table 6 gives an overview of articles discussing case studies on the prediction of macroinvertebrates by means of ANNs. A total of 26 cases were found in literature. These papers were however produced by a far smaller number of research groups, since most of the research groups published more than one paper on the subject. Among them, there is a French, Belgian, German, British, South-Korean and Australian research group, counting up to six groups although this number is debatable because the groups do not work completely independently, as some cooperative papers clearly testify. All papers are very recent, the earliest dating from 1998.

About one out of two papers mention the software package used for modelling. Three different packages were cited: MATLAB, WEKA and NNEE. Several of the modellers not mentioning the software package use their own code, implemented in an existing modelling environment such as MATLAB. Evidently, the software package used should not influence the modelling results although neither the use of own programming nor existing software is an absolute guarantee that no errors will occur, which means that any system should be used with care.

The number of input variables ranged from 3 to 39, usually between 5 and 15. These variables included geographical, seasonal and habitat quality parameters (sinuosity, vegetation,...) as well as physical–chemical properties (dissolved oxygen, water temperature, pH, nutrient concentrations, COD, ...) and characteristics of toxicity. The performance of neural networks with more input variables was not necessarily higher, as shown in some studies (e.g. Walley and Fontama 1998). The target variables were usually presence/absence (nine cases) or abundance (six cases) of macroinvertebrate taxa or derived properties such as taxa richness, ASPT score or exergy.

The neural networks were in almost all cases of the feedforward connection type, in some cases combined with a Self-organizing Map. Exceptions included real-time recurrent neural networks, an Elman recurrent neural network and a forward only neural network. Most Self-organizing Maps were trained with the Kohonen learning rule, one was trained with a radial basis function. Most feedforward

Table 6 Overview of publications discussing case studies on the prediction of macroinvertebrates by means of artificial neural networks

Reference	Software package	Input variables	Output	Location(s)	Connection type	Training algorithm	Network architecture	No. train. samples – No. test samples	Determination of network architecture	Transfer functions	Scaling of variables	Perf. measure
Beauchard et al. (2003)	N/S	A, P, Lo, R, DISTs, SDA	richn	Morocco, Algeria, Tunisia	FF	BP	7-4-1	210-1 (leave-one-out)	Empirically	STF	N/S	*r*
Brosse et al. (2001)	MATLAB	A, SDA, SO, CA, VEG, AE, D, W, S	div	Taieri river (New Zealand)	FF	BP	10-4-1	96-1 (leave-one-out)	N/S	STF	N/S	*r*, PI
Brosse et al. (2003)	MATLAB	LU, SDA, A, CA, PR, SO, W, D, S	div	Taieri river (New Zealand)	FF	BP	(10, 8)-4-1	96-1 (leave-one-out)	N/S	STF	N/S	*r*, MSE
Céréghino et al. (2003)	N/S	A, SO, DISTs, T	richn	Adour-Garonne river basin (France)	FF	BP	4-5-1	130-25	Trial and error	N/S	N/S	*r*
Chon et al. (2001)	N/S	MI, FV, D, OM, S	comm	Yangjae stream (Korea)	RTRC	RL	(7+4)-13-7	N/S	Trial and error	N/S	[0 1]	*r*
Chon et al. (2002)	N/S	MI	dens	Yangjae stream (Korea)	FF	BP	(5-25)-(8-30)-5	N/S	Empirically	STF	[0 1]	*r*
					ERC	BP	5-30-5					
		MI, FV, D, OM, S	comm		RTRC	RL	(7+4)-13-7					
Dedecker et al. (2002)	MATLAB	T, pH, DO, Cond, SS, D, W, S, Sh, VEG, FV, Me, HRB, PR, AE	pr/ab	Zwalm river basin (Belgium)	FF	BP	15-10-1	40-20 (3-fold) 45-15 (4-fold)	Trial and error	STF	N/S	CCI
Dedecker et al. (2004)	MATLAB	T, pH, DO, Cond, SS, D, W, S, Sh, VEG, FV, Me, HRB, PR, AE	pr/ab	Zwalm river basin (Belgium)	FF	BP, LM	15-(2, 5, 10, 20, 25)-(5, 10)-1	108-12 (10-fold)	Trial and error	STF	[−1 1]	CCI, CK
Dedecker et al. (2007)	MATLAB	T, pH, DO, Cond, SS, D, W, S, FV, Me, HRB, PR, AE, NO_3^-, PO_4^{3-}, NH_4^+, COD, Ph, Ni, SO, DISTm	abu	Zwalm river basin (Belgium)	FF	BP	24-10-1	119-60 (3-fold)	N/S	STF	IN:N/S OUT[log	
(abu + 1)]	*r*											
Dedecker et al. (2005a)	MATLAB	T, pH, DO, Cond, SS, D, W, S, Sh, VEG, FV, Me, HRB, PR, AE	pr/ab	Zwalm river basin (Belgium)	FF	BP, LM	15-N/S-1	108-12 (10-fold)	Trial and error	STF	[−1 1]	CCI, CK
Dedecker et al. (2005b)	MATLAB	T, pH, DO, Cond, SS, D, W, S, FV, Me, HRB, PR, AE, NO_3^-, PO_4^{3-}, NH_4^+, COD, Ph, Ni, SO, DISTm	abu	Zwalm river basin (Belgium)	FF	BP	24-10-1	119-60 (3-fold)	N/S	STF	IN:N/S OUT[log (abu+1)]	*r*

Table 6 continued

Reference	Software package	Input variables	Output	Location(s)	Connection type	Training algorithm	Network architecture	No. train. samples – No. test samples	Determination of network architecture	Transfer functions	Scaling of variables	Perf. measure
D'heygere et al. (2006)	WEKA	Day, W, D, FV, S, T, pH, DO, Cond, TOX, TOC, OM, Ni, Ph	pr/ab	Flemish river sediment (Belgium)	FF	BP	(6-17)-10-2	324-36 (10-fold)	N/S	N/S	N/S	CCI, CK, RMSE
Gabriels et al. (2002)	MATLAB	S, DM, T, pH, DO, Cond, TOC, OM, Ni, Ph	abu	Flemish river sediment (Belgium)	FF	BP	20-10-92	250-95	Arbitrarily chosen	N/S	IN[−1 1], OUT[0, 1]	*r*, CCI
Gabriels et al. (2007)	N/S	Day, W, D, FV, S, pH, DO, Cond, Ni, Ph	pr/ab	Flemish river sediment (Belgium)	FF	BP	12-N/S-(1, 92)	294-49 (7-fold)	Trial and error	N/S	[−1 1]	CCI, CK
Goethals et al. (2002)	MATLAB	T, pH, DO, Cond, SS, D, W, S, Sh, VEG, FV, Me, HRB, PR, AE	pr/ab	Zwalm river basin (Belgium)	FF	BP	15-10-52	40-20	Trial and error	STF	N/S	CCI
Hoang et al. (2001)	N/S	A, SO, R, SoilT, VEG, S, T, ...	pr/ab	Queensland streams (Australia)	FF	BP	39-15-37	650-167	N/S	STF	N/S	CCI
Hoang et al. (2002)	N/S	SO, Lo, Ni, ...	pr/ab	Queensland streams (Australia)	FF	BP	N/S	N/S	N/S	STF	N/S	CCI
Marshall et al. (2002)	N/S	A, SO, R, SoilT, VEG, S, T, ...	pr/ab	Queensland streams (Australia)	FF	BP	39-15-37	650-167	N/S	STF	N/S	CCI
Obach et al. (2001)	N/S	T, DI, P	abu	Hesse (Germany)	FF	Mod BP	N/S	N/S	N/S	N/S	[0 1]	R^2, RMSE
					FF	GRNN						
					FF	LNN						
					SOM	RBF	N/S-120			N/A		
Park et al. (2001)	N/S	Ex		Suyong river (Korea)	SOM	KLR	N/S	N/S	N/S	N/A	[0.01 0.99]	*r*
		Comm	Ex		FF	BP	N/S-5-N/S			STF		
Park et al. (2003a)	N/S	EPTC	EPTC	Adour-Garonne river basin (France)	SOM	KLR	N/S-140	130-25	N/S	N/A	[0 1]	*r*
		A, SO, DISTs, T			FF	BP	4-N/S-1			N/S		
Park et al. (2003b)	N/S	S, VEG, DO, W, Cond, T, FV, D, NO_3^-, PO_4^{3-}, NH_4^+, ...	richn, SH	The Netherlands	Forward only	CPN	N/S	500-164	N/S	N/S	[0 1]	*r*

Table 6 continued

Reference	Software package	Input variables	Output	Location(s)	Connection type	Training algorithm	Network architecture	No. train. samples – No. test samples	Determination of network architecture	Transfer functions	Scaling of variables	Perf. measure
Schleiter et al. (1999)	N/S	T, P, pH, DO, Cond, D, W, S, DI, NO_3^-, NO_2^-, NH_4^+, COD, BOD, Ph, ...	abu	Kuhbach, Lahn and Breitenbach (Germany)	FF	BP	N/S	150-150 200-100 225-75	N/S	N/S	[0 1]	MSE, R^2
Schleiter et al. (2001)	NNEE	pr/ab, abu	BOD, Cond,NH_3, NO_3^-, NO_2^-, NH_4^+, Ni, pH, Ph, T, DO, SI	Hesse (Germany)	FF	Mod BP	N/S	45-6	N/S	N/S	[0 1]	R^2, RMSE, CVE
					FF	GRNN						
					FF	LNN						
Wagner et al. (2000)	N/S	T, P, DI	abu	Breitenbach (Germany)	FF	BP	N/S	216-54	N/S	N/S	[0 1]	R^2
Walley and Fontama (1998)	N/S	Coord, DISTs, SL, Alk, DI, A, S, W, D	ASPT, NFAM	UK	FF	BP	13-6-6-1	307-307 (2-fold)	N/S	N/S	No, log (DISTs), log(SL)	r

N/S = not specified; N/A = not applicable; NNEE = neural network experimental environment; *Input and output variables*: A = altitude; abu = abundance; Alk = alkalinity; AE = artificial embankment structures; ASPT = average score per taxon; BOD = biological oxygen demand; CA = catchment area; COD = chemical oxygen demand; Cond = conductivity; Comm = community data; Coord = *X* and *Y* coordinates; D = depth; Day = day; dens = density; DI = discharge; DISTs = distance from river source; DISTm = distance to mouth; div = diversity; DM = dry matter; DO = dissolved oxygen; EPTC = richness of Ephemeroptera, Plecoptera, Trichoptera and Coleoptera; Ex = exergy from the MI communities; FV = flow velocity; HRB = hollow river banks; Lo = longitude; LU = land use; Me = meandering; MI = macroinvertebrates; NFAM = number of families; NH_3 = ammonia; NH_4^+ = ammonium; Ni = nitrogen; NO_2^- = nitrite; NO_3^- = nitrate; OM = organic matter; P = precipitation; Ph = phosphorus; PO_4^{3-} = phosphate; PR = pool/riffle; pr/ab = presence/absence; richn = species richness; S = substrate; SDA = surface of the drainage area; SH = Shannon diversity index; Sh = shadow; SI = saprobic index; SL = slope; SO = stream order; SoilT = soil type; SS = suspended solids; T = water temperature; TOC = total organic carbon; TOX = toxicity; VEG = vegetation; W = width; *Connection type*: ERC = Elman recurrent neural network; FF = feedforward; RTRC = real-time recurrent neural network; SOM = Kohonen self-organizing mapping; *Training algorithm*: BP = backpropagation; CPN = counterpropagation network; GRNN = general regression neural network; KLR = Kohonen learning rule; LM = Levenberg-Marquardt; LNN = linear neural network; Mod BP = modified backpropagation; RBF = radial basis function; RL = recurrent learning; *Transfer functions*: STF = sigmoid transfer function; *Scaling of variables* IN = input; OUT = output; *Performance measure*: CCI = percentage of correctly classified instances; CK = Cohen's kappa; CVE = cross-validation error; MSE = mean squared error between observed and estimated values; PI = performance index (proportion of predictions within 10% of the observed value); r = correlation coefficient between observed and predicted values; R^2 = determination coefficient between observed and predicted values; RMSE = root mean squared error between observed and estimated values

neural networks were trained with backpropagation or a modification of it. In some cases the Levenberg–Marquardt algorithm, general regression, a linear neural network and/or counterpropagation were tested. The real-time recurrent neural networks were trained with recurrent learning and the Elman recurrent neural network was trained with backpropagation.

The network architecture was reported in most cases. The number of hidden layers was usually one and in none of the reported cases higher than two. The number of hidden neurons was usually of the same order of magnitude as the number of input nodes. Network architecture was determined, if stated, by 'trial and error' (seven cases), 'empirically' (two cases) or 'arbitrarily chosen' (one case). In the majority of the cases, the choice of network architecture was not discussed at all. Clearly, this crucial step in the modelling process is poorly documented for this type of applications. In general, rules of thumb were not (explicitly) used while trial and error was applied without a clear strategy. However, it is recommended to use rules of thumb as a starting point for a systematic trial and error process in order to refine and validate the choice of neural network architecture. In addition, techniques for model optimization were hardly used to optimize network geometry.

The transfer functions, where specified, were of the sigmoid transfer function type.

The data were generally rescaled to the interval [−1 1] or [0 1]. Maier and Dandy (2000) recommend avoiding the extreme limits of the transfer function when rescaling the outputs. However, in only one study (Park et al. 2001) an interval smaller than the transfer function allows was used.

A variety of performance measures was used, strongly related to the type of output parameter. For predictions of presence/absence, the percentage of CCI was the most frequently used performance measure. In some cases Cohen's kappa was calculated and in one case also the RMSE. When predicting continuous variables such as abundance or taxa richness, a variety of criteria were calculated in the cited case-studies: r, R^2, MSE, RMSE. Also the cross-validation error (CVE) and /or the proportion of predictions within a specified distance of the observed value (PI) were applied as alternatives to these more common performance measures. Two other measures were used after transforming the abundance outputs into abundance classes: CCI and Cohen's kappa.

Among the articles that specify the number of samples used for training, the number ranges from 40 to 650. The ratio of the number of training samples versus the number of hidden neurons ranges from 4.5 to 52.5 with an average of 16.8, when all specified combinations are taken into account.

Discussion and needs for further research

Predictive ANN model development and application

So far, several rules of thumb for determining model geometry have been proposed. Alternatively, techniques for automatically selecting model architecture are suggested. However, in most of the studies discussing the prediction of macroinvertebrates in aquatic systems, model geometry was decided with trial and error. But details on this process are in most cases missing. Consequently, there is a need to develop guidelines to clearly identify the circumstances under which particular approaches should be used and how to optimize the parameters that control neural network architecture (Özesmi et al. 2006). A major aspect in this context is the data splitting for training and validation. It is important to determine the optimal number of folds for cross-validation. A balance needs to be determined between reliability and robustness. For this, it is recommended to train and validate models based on at least five different fold options (e.g. 3, 5, 10, total number of instances divided by two as well as total number of instances minus one) and select the best fold number.

The use of sensitivity methods can allow peer-review by ecological experts, and offer an additional method to guarantee the quality of ANN models. Testing the model in a wider range of situations (in space and time) will permit to define the range of applications for which the model predictions are suitable. It is moreover crucial to provide information about the range of the training for all input variables. Based on this, a user has information about the reliability of the simulations and in what range the predictions are relevant. In this manner, the quality of the models is assessed from several perspectives and reduces the chance to develop theoretically very good models, that are from a practical or ecological point of view misleading (Tan et al. 2006).

The contribution and related selection of input variables is another very important aspect that calls for more research (Jeong et al. 2006). Many variables are missing, while others have a high variability, caused by measurement difficulties or by the natural dynamics in the river systems (e.g. flow velocities, water temperatures). Therefore, also the effect of monitoring methods needs more research, in particular the incorporation of 'new' variables which are less straightforward to be used in a model. This is in particular the case for structural and morphological variables that often need to be visually monitored, but also for heavy metals and other potential toxicants, since their effects are often related to the environment in which they are released (bio-availability, accumulation, ...). These toxicants may be a new challenge in the field of soft computing models to predict river communities, in particular macroinvertebrates.

ANNs versus other habitat suitability modelling techniques

Recently, numerous computational and statistical approaches have been developed (Chon and Park 2006). The combination of methods (datadriven and expert knowledge based) is moreover an important new trend, e.g. Salski and Holsten (2006). Nevertheless, at present, there is a lack of comparative papers (e.g. Skidmore et al. 1996; Manel et al. 1999) in which more than two statistical methods have been applied to the same data set. Most published ecological modelling studies use only one of the many techniques that may be properly used, and little information is available on the respective predictive capacity of each approach. The debate is usually restricted to the intrinsic suitability of a particular method for a given data set. When starting a static modelling study the choice of an appropriate method would be much facilitated by having access to publications that show the advantages and disadvantages of different methods in a particular context (Guisan and Zimmermann 2000).

When looking at the different soft computing techniques, they all seem to have particular strengths and weaknesses. ANNs for instance can provide well performing models, but the integration of expert knowledge is difficult. Fuzzy logic can be used to develop models merely on expert knowledge, but the number of input variables is very limited, because the rule sets become very complex when more than five input variables are used (e.g. Adriaenssens et al. 2006). Bayesian Belief Networks have the interesting characteristic to be able to reveal how different variables interact, on the other hand, the information necessary to build these networks and to set-up the variable distributions is also huge (Adriaenssens et al. 2005). Goethals (2005) compared two different data-driven methods (ANN and classification trees) for the prediction of macroinvertebrates. Crucial findings of this study were that different methods seem to use different input variables and extract other relations, but the reliability seems in particular to be limited by the information in the dataset. Moreover, the outcome of the two methods is quite different, in the case of the classification trees threshold values about river characteristics can be obtained, whereas in case of ANN (in combination with sensitivity analyses), habitat preference curves for the river characteristics could be generated.

However, based on the rather limited set of case studies in which several methods were compared, it is up to now nearly impossible to have clear insight in when to use what kind of method. For this, meta-models (technique selection models) based on a large set of datasets should be developed. However, several methods such as Bayesian Belief Networks and fuzzy logic have to date rarely been applied in ecology, so the methods themselves need further exploration as well, since the quality of the application of a technique is to a high extent related on a well understanding of the modelling method. Also the use of different evaluation measures and methods (validation) is crucial, and should be related to the type of predictions. Most likely, the type of predictions needed, the timeframe and available data will mainly determine what technique is most relevant.

Integrating and combining models

Recently, several practical concepts and software systems were developed related to environmental decision support (e.g. Argent 2004; Lam et al. 2004; Voinov et al. 2004; Poch et al. 2005; Pereira et al. 2006). From a technical point of view, one can opt to build a new model for each application (integrative approach) or to utilize existing models where possible (combinatory approach). The first approach has the benefit of control in the models' design and linkage,

but requires longer development time. The second approach saves on the development time, but requires additional work to link up existing models (Lam et al. 2004). However, when a lot of models are already available, it is probably the best option. However, up to now, this type of coupling is missing in scientific literature and remains a crucial aspect to support decision making in integrated ecological water management.

Conclusion

Although there is quite some experience gained with data driven models to predict macroinvertebrates, several aspects need more attention in future ANN development studies. Regarding model training and validation, many rules of thumb and performance indicators are provided in technical literature, and this review suggested a subset that is relevant to be used in ecological modelling. Moreover, data preparation for training and validation was usually decided by an undefined trial and error process in most of the studies available in the literature. Suggestions for optimization of the number of folds (data preprocessing) and considering more input variables were raised in this paper. Many essential variables, such as heavy metals and other potential toxicants, are often not taken into account so far, whereas other variables are superfluous and might be removed. The use of sensitivity analyses is probably a major need to increase the credibility of these often called 'black box methods' to ecologists and valid practical simulations are necessary to gain trust among river managers. Furthermore, there is also a need for more comparative research that shows the strengths and weaknesses of different types of habitat suitability models in a particular context. In this way, it would be possible to have insight in when to use what kind of method, and which data need to be collected to be able to answer the relevant questions of water managers.

Acknowledgments Andy Dedecker is a grant holder of the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT). The authors also like to thank the Flemish Government (Administration of Higher Education and Scientific Research) for the mobility grants received within the context of the TOURNESOL 2003 programme (Project T2003.01) allowing to exchange modelling experience in Ghent University with the European PAEQANN project (EVK1-CT1999–00026) in the Université Paul-Sabatier in Toulouse.

References

Adriaenssens V, Goethals PLM, De Pauw N (2006) Fuzzy knowledge-based models for prediction of Asellus and Gammarus in watercourses in Flanders (Belgium). Ecol Model 195(1–2):3–10

Adriaenssens V, Goethals PLM, Trigg D, De Pauw N (2005) Application of Bayesian belief networks for the prediction of macroinvertebrate taxa in rivers. Ann Limnol – Int J Lim 40(3):181–191

Amari S, Murata N, Müller KR, Finke M, Yang HH (1997) Asymptotic statistical theory of overtraining and cross-validation. IEEE T Neural Networks 8(5):985–996

Anders U, Korn O (1999) Model selection in neural networks. Neural Networks 12(2):309–323

Angeline PJ, Sauders GM, Pollack JB (1994) An evolutionary algorithm that constructs recurrent neural networks. IEEE T Neural Networks 5:54–65

Argent RM (2004) An overview of model integration for environmental applications – components, frameworks and semantics. Environ Model Softw 19:219–234

Barnard E, Wessels L (1992) Extrapolation and interpolation in neural network classifiers. IEEE Control Syst 12(5):50–53

Bartlett EB (1994) Dynamic node architecture learning: an information theoretic approach. Neural Networks 7(1):129–140

Beauchard O, Gagneur J, Brosse S (2003) Macroinvertebrate richness patterns in North African streams. J Biogeogr 30:1821–1833

Bebis G, Georgiopoulos M (1994) Feed-forward neural networks. IEEE Potentials 13(4):27–31

Brosse S, Guégan JF, Tourenq JN, Lek S (1999) The use of neural networks to assess fish abundance and spatial occupancy in the littoral zone of a mesotrophic lake. Ecol Model 120:299–311

Brosse S, Lek S, Townsend CR (2001) Abundance, diversity, and structure of freshwater invertebrates and fish communities: an artificial neural network approach. New Zeal J Mar Fresh 35:135–145

Brosse S, Arbuckle CJ, Townsend CR (2003) Habitat scale and biodiversity: influence of catchment, stream reach and bedform scales on local invertebrate diversity. Biodiv Conserv 12:2057–2075

Céréghino R, Park YS, Compin A, Lek S (2003) Predicting the species richness of aquatic insects in streams using a limited number of environmental variables. J N Am Benthol Soc 22(3):442–456

Cherkassky V, Lari-Najafi H (1992) Data representation for diagnostic neural networks. IEEE Intell Syst 7(5):43–53

Chon TS, Park YS (2006) Ecological informatics as an advanced interdisciplinary interpretation of ecosystems. Ecol Inf 1(3):213–217

Chon TS, Kwak IS, Park YS, Kim TH, Kim Y (2001) Patterning and short-term predictions of benthic macroinvertebrate community dynamics by using a recurrent artificial neural network. Ecol Model 146:181–193

Chon TS, Park YS, Kwak IS, Cha EY (2002) Non-linear approach to grouping, dynamics and organizational informatics of benthic macroinvertebrate communities in streams by Artificial Neural Networks. In: Recknagel F (ed) Ecological Informatics. Understanding ecology by biologically-inspired computation. Springer, Berlin, pp 127–178

Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20:37–46

Dai HC, Macbeth C (1997) Effects of learning parameters on learning procedure and performance of a BPNN. Neural Networks 10(8):1505–1521

Dedecker AP, Goethals PLM, De Pauw N (2002) Comparison of artificial neural network (ANN) model development methods for prediction of macroinvertebrate communities in the Zwalm river basin in Flanders, Belgium. The Sci World J 2:96–104

Dedecker A, Goethals PLM, Gabriels W, De Pauw N (2004) Optimisation of Artificial Neural Network (ANN) model design for prediction of macroinvertebrate communities in the Zwalm river basin (Flanders, Belgium). Ecol Model 174(1–2):161–173

Dedecker AP, Goethals PLM, De Pauw N (2005a) Sensitivity and robustness of stream model based on artificial neural networks for the simulation of different management scenarios. In: Lek S, Scardi M, Verdonschot PFM, Descy JP, Park YS (eds) Modelling community structure in freshwater ecosystems. Springer-Verlag, pp 133–146

Dedecker AP, Goethals PLM, D'heygere T, Gevrey M, Lek S, de Pauw N (2005b) Application of artificial neural network models to analyse the relationships between *Gammarus pulex* L. (Crustacea, Amphipoda) and river characteristics. Environ Monit Assess 111(1–3):223–241

Dedecker AP, Goethals PLM, D'heygere T, Gevrey M, Lek S, De Pauw N (2007) Habitat preference study of *Asellus* (Crustacea, Isopoda) by applying input variable contribution methods to artificial neural network models. Environ Model Assess (in press)

D'heygere T, Goethals PLM, De Pauw N (2006) Genetic algorithms for optimisation of predictive ecosystems models based on decision trees and neural networks. Ecol Model 195(1–2):20–29

Dimopoulos Y, Bourret P, Lek S (1995) Use of some sensitivity criteria for choosing networks with good generalization ability. Neural Process Lett 2:1–4

Dimopoulos I, Chronopoulos J, Chronopoulou Sereli A, Lek S (1999) Neural network models to study relationships between lead concentration in grasses and permanent urban descriptors in Athens city (Greece). Ecol Model 120:157–165

Efron B (1983) Estimating the error rate of a prediction rule: some improvements on cross-validation. J Am Stat Assoc 78:316–331

Fang J, Xi Y (1997) Neural network design based on evolutionary programming. Artif Intell Eng 11:155–161

Fausett L, Elwasif W (1994) Predicting performance from test scores using backpropagation and counterpropagation. Proceedings of IEEE international conference on neural networks, pp 3398–3402

Fielding AH, Bell JF (1997) A review of methods for the assessment of prediction errors in conservation presence/absence models. Environ Conserv 24(1):38–49

Fletcher D, Goss E (1993) Forecasting with neural networks: an application using bankruptcy data. Inform Manage 24(3):159–167

Flood I, Kartam N (1994) Neural networks in civil engineering. I: Principles and understanding. J Comput Civil Eng 8(2):131–148

Gabriels W, Goethals PLM, De Pauw N (2002) Prediction of macroinvertebrate communities in sediments of Flemish watercourses based on artificial neural networks. Verh Internat Verein Limnol 28(2):777–780

Gabriels W, Goethals PLM, Dedecker AP, Lek S, De Pauw N (2007) Analysis of macrobenthic communities in Flanders, Belgium, using a stepwise input variable selection procedure with artificial neural networks. Aquat Ecol

Garson GD (1991) Interpreting neural-network connection weights. Artif Intell Expert 6:47–51

Gevrey M, Dimopoulos I, Lek S (2003) Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecol Model 160:249–264

Goethals, PLM (2005) Data driven development of predictive ecological models for benthic macroinvertebrates in rivers. PhD thesis, Ghent University, Ghent, Belgium, 400 pp

Goethals P, Dedecker A, Gabriels W, De Pauw N (2002) Development and application of predictive river ecosystem models based on classification trees and artificial neural networks. In: Recknagel F (ed) Ecological Informatics. Understanding ecology by biologically-inspired computation. Springer, Berlin, pp 91–108

Goh ATC (1995) Back-propagation neural networks for modelling complex systems. Artif Intell Eng 9:143–151

Goldberg DE (1989) Genetic Algorithms in search, optimization and machine learning. Addison-Wesley Publishing Company, Reading, MA, 412 pp

Guégan JF, Lek S, Oberdorff T (1998) Energy availability and habitat heterogeneity predict global riverine fish diversity. Nature 391:382–384

Guisan A, Zimmermann NE (2000) Predictive habitat distribution models in ecology. Ecol Model 135:147–168

Hagan MT, Demuth HB, Beale M (1996) Neural network design. PWS Publishing Company, Boston

Haykin S (1999) Neural networks: a comprehensive foundation, 2nd edn. Prentice Hall, New Jersey

Hecht-Nielsen R (1987) Kolmogorov's mapping neural network existence theorem. In: First IEEE international conference on neural networks. San Diego, USA, pp 11–14

Hoang H, Recknagel F, Marshall J, Choy S (2001) Predictive modelling of macroinvertebrate assemblages for stream habitat assessments in Queensland (Australia). Ecol Model 146:195–206

Hoang H, Recknagel F, Marshall J, Choy S (2002) Elucidation of hypothetical relationships between habitat conditions and macroinvertebrate assemblages in freshwater streams by artificial neural networks. In: Recknagel F (ed) Ecological informatics. Understanding ecology by biologically-inspired computation. Springer, Berlin, pp 179–192

Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Networks 2(5):359–366

Hung MS, Hu MY, Shanker MS, Patuwo BE (1996) Estimating posterior probabilities in classification problems with neural networks. Int J Comput Intell Org 1(1):49–60

Jeong KS, Kim DK, Joo GJ (2006) River phytoplankton prediction model by artificial neural network: model performance and selection of input variables to predict time-series phytoplankton proliferations in a regulated river system. Ecol Inf 1(3):235–245

Kanellopoulos I, Wilkinson GG (1997) Strategies and best practice for neural network image classification. Int J Remote Sens 18(4):711–725

Kim K, Han I (2000) Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert Syst Appl 19:125–132

Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biol Cybern 43:59–69

Kwok TY, Yeung DY (1997a) Constructive algorithms for structure learning in feedforward neural networks for regression problems. IEEE T Neural Network 8(3):630–645

Kwok TY, Yeung DY (1997b) Objective functions for training new hidden units in constructive neural networks. IEEE T Neural Network 8(5):1131–1148

Laë R, Lek S, Moreau J (1999) Predicting fish yield of African lakes using neural networks. Ecol Model 120:325–335

Lam D, Leon L, Hamilton S, Crookshank N, Bonin D, Swayne D (2004) Multi-model integration in a decision support system: a technical user interface approach for watershed and lake management scenarios. Environ Model Softw 19:317–324

Lek S, Beland A, Dimopoulos I, Lauga J, Moreau J (1995) Improved estimation, using neural networks, of the food consumption of fish populations. Mar Freshwater Res 46:1229–1236

Lek S, Belaud A, Baran P, Dimopoulos I, Delacoste M (1996a) Role of some environmental variables in trout abundance models using neural networks. Aquat Living Resour 9:23–29

Lek S, Delacoste M, Baran P, Dimopoulos I, Lauga J, Aulagnier S (1996b) Application of neural networks to modeling nonlinear relationships in ecology. Ecol Model 90:39–52

Lek S, Guégan JF (1999) Artificial neural networks as a tool in ecological modelling, an introduction. Ecol Model 120:65–73

Lenard MJ, Alam P, Madey GR (1995) The application of neural networks and a qualitative response model to the auditor's going concern uncertainty decision. Decision Sci 26(2):209–227

Liano K (1996) Robust error measure for supervised neural network learning with outliers. IEEE T Neural Networks 7(1):246–250

Maier HR, Dandy GC (2000) Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environ Model Softw 15:101–124

Manel S, Dias JM, Buckton ST, Ormerod SJ (1999) Alternative methods for predicting species distribution: an illustration with Himalayan river birds. J Appl Ecol 36:734–747

Marshall J, Hoang H, Choy S, Recknagel F (2002) Relationships between habitat properties and the occurrence of macroinvertebrates in Queensland streams (Australia) discovered by a sensitivity analysis with artificial neural networks. Verh Internat Verein Limnol 28:1415–1419

Masters T (1993) Practical neural network recipes in C++. Academic Press, San Diego

Mastrorillo S, Dauba F, Oberdorff T, Guégan JF, Lek S (1998) Predicting local fish species richness in the Garonne river basin. Ecology 321:423–428

Mastrorillo S, Lek S, Dauba F (1997a) Predicting the abundance of minnow *Phoxinus phoxinus* (Cyprinidae) in the River Ariege (France) using artificial neural networks. Aquat Living Resour 10:169–176

Mastrorillo S, Lek S, Dauba F, Beland A (1997b) The use of artificial neural networks to predict the presence of small-bodied fish in a river. Freshwater Biol 38:237–246

Nabhan TM, Zomaya AY (1994) Toward generating neural network structures for function approximation. Neural Networks 7(1):89–99

Obach M, Wagner R, Werner H, Schmidt HH (2001) Modelling population dynamics of aquatic insects with artificial neural networks. Ecol Model 146:207–217

Olden JD, Jackson DA (2002) Illuminating the ''black box'': a randomization approach for understanding variable contributions in artificial neural networks. Ecol Model 154:135–150

Olden JD, Joy MK, Death RG (2004) An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. Ecol Model 178:389–397

Özesmi SL, Özesmi U (1999) An artificial neural network approach to spatial habitat modelling with interspecific interaction. Ecol Model 116(1):15–31

Özesmi SL, Tan CO, Özesmi U (2006) Methodological issues in building, training, testing artificial neural networks in ecological applications. Ecol Model 195(1–2):83–93

Park YS, Kwak IS, Chon TS, Kim JK, Jorgensen SE (2001) Implementation of artificial neural networks in patterning and prediction of exergy in response to temporal dynamics of benthic macroinvertebrate communities in streams. Ecol Model 146:143–157

Park YS, Céréghino R, Comin A, Lek S (2003a) Applications of artificial neural networks for patterning and predicting aquatic insect species richness in running waters. Ecol Model 160:265–280

Park YS, Verdonschot PFM, Chon TS, Lek S (2003b) Patterning and predicting aquatic macroinvertebrate diversities using artificial neural networks. Water Res 37:1749–1758

Patuwo E, Hu MY, Hung MS (1993) Two-group classification using neural networks. Decision Sci 24(4):825–845

Pereira A, Duarte P, Norro A (2006) Different modelling tools of aquatic ecosystems: a proposal for a unified approach. Ecol Inf 1(4):407–421

Piramuthu S, Shaw M, Gentry J (1994) A classification approach using multi-layered neural networks. Decis Support Syst 11(5):509–525

Poch M, Comas J, Rodríguez-Roda I, Sànchez-Marrè M, Cortés U (2004) Designing and building real environmental decision support systems. Environ Model Softw 19(9):857–873

Qian N (1999) On the momentum term in gradient descent learning algorithms. Neural Networks 12(1):145–151

Randin CF, Dirnböck T, Dullinger S, Zimmermann NE, Zappa M, Guisan A (2006) Are niche-based species distribution models transferable in space? J Biogeogr 33:1689–1703

Recknagel F, Cao H, Kim B, Takamura N, Welk A (2006) Unravelling and forecasting algal population dynamics in two lakes different in morphometry and eutrophication by neural and evolutionary computation. Ecol Inf 1(2):133–151

Reed R (1993) Pruning algorithms – a review. IEEE T Neural Networks 4(5):740–747

Rogers LL, Dowla FU (1994) Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. Water Resour Res 30(2):457–481

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagation errors. Nature 323:533–536

Salski A, Holsten B (2006) A fuzzy and neuro-fuzzy approach to modelling cattle grazing on pastures with low stocking rates in Central Europe. Ecol Inf 1(3):269–276

Scardi M, Harding LW (1999) Developing an empirical model of phytoplankton primary production: a neural network case study. Ecol Model 120(2–3):213–223

Schleiter IM, Borchardt D, Wagner R, Dapper T, Schmidt KD, Schmidt HH, Werner H (1999) Modelling water quality, bioindication and population dynamics in lotic ecosystems using neural networks. Ecol Model 120(2–3):271–286

Schleiter IM, Obach M, Borchardt D, Werner H (2001) Bio-indication of chemical and hydromorphological habitat characteristics with benthic macro-invertebrates based on artificial neural networks. Aquat Ecol 35:147–158

Skidmore AK, Gauld A, Walker P (1996) Classification of Kangaroo habitat distribution using three GIS models. Int J Geogr Inf Syst 10:441–454

Tan CO, Özesmi U, Beklioglu M, Per E, Kurt B (2006) Predictive models in ecology: comparison of performances and assessment of applicability. Ecol Inf 1(2):195–211

Voinov A, Fitz C, Boumans R, Costanza R (2004) Modular ecosystem modeling. Environ Model Softw 19:285–304

Wagner R, Dapper T, Schmidt HH (2000) The influence of environmental variables on the abundance of aquatic insects: comparison of ordination and artificial neural networks. Hydrobiologia 422–423:143–152

Walczak S (1995) Developing neural nets currency trading. Artif Intell Financ 2(1):27–34

Walczak S, Cerpa N (1999) Heuristic principles for the design of artificial neural networks. Inform Software Tech 41(2):107–117

Walley WJ, Fontama VN (1998) Neural network predictors of average score per taxon and number of families at unpolluted river sites in Great Britain. Water Res 32(3): 613–622

Wang F (1994) The use of artificial neural networks in a geographical information system for agricultural land-suitability assessment. Environ Plann A 26(2):265–284

Weigend AS, Huberman BA, Rumelhart DE (1990) Predicting the future: a connectionist approach. Int J Neural Syst 1(3):193–209

Wicker D, Rizki MM, Tamburino LA (2002) E-Net: evolutionary neural network synthesis. Neurocomputing 42:171–196

Yao X (1993) A review of evolutionary artificial neural networks. Int J Intell Syst 8(4):539–567

Yao J, Teng N, Poh HL, Tan CL (1998) Forecasting and analysis of marketing data using neural networks. J Inf Sci Eng 14:843–862

Zhao W, Chen D, Hu S (2000) Optimizing operating conditions based on ANN and modified Gas. Comput Chem Eng 24:61–65