Journal of Industrial and Systems Engineering
Vol. 3, No. 1, pp 38-55
Spring 2009

Knowledge Extraction from the Neural 'Black Box' in Ecological Monitoring

Weckman G.R.[1*], Millie D.F.[2], Ganduri C.[1], Rangwala M.[1], Young W.[1], Rinder M.[1], Fahnenstiel G.L.[3]

[1] Industrial Systems Engineering, Ohio University, Athens, OH 45701-2979, USA
weckman@bobcat.ent.ohiou.edu

[2] Florida Institute of Oceanography, University of South Florida, Saint Petersburg, Florida 33701, USA

[3] Lake Michigan Field Station, Great Lakes Environmental Research Laboratory, National Oceanic & Atmospheric Administration, Muskegon, Michigan 49441,USA

ABSTRACT

Phytoplankton biomass within the Saginaw Bay ecosystem (Lake Huron, Michigan, USA) was characterized as a function of select physical/chemical indicators. The complexity and variability of ecological systems typically make it difficult to model the influences of anthropogenic stressors and/or natural disturbances. Here, Artificial Neural Networks (ANNs) were developed to model chlorophyll *a* concentrations, a measure for water-column phytoplankton biomass and a proxy for system-level health. ANNs act like "black boxes" in the sense that relationships are encoded as weight vectors within the trained network and as such, cannot easily support the generation of scientific hypotheses unless these relationships can be explained in a comprehensible form. Accordingly, the 'knowledge' and/or rule-based information embedded within ANNs needs to be extracted and expressed as a set of comprehensible 'rules'. Such extracted information would enhance the delineation and understanding of ecological complexity and aid in developing usable prediction tools. Comparisons of various computational approaches (including TREPAN, an algorithm for constructing decision trees from neural networks) used in extracting rule-based information from trained Saginaw Bay ANNs are discussed.

Keywords: Ecological monitoring, artificial neural networks, chlorophyll prediction, knowledge extraction.

1. INTRODUCTION

A crucial shortcoming in modeling ecological systems via machine learning is their lack of insight into biological processes and/or relationships and autocorrelations between/among environmental (input) variables. An exploration of machine learning techniques would capture predictive knowledge regarding such processes and relationships with the desired modeled (output) variable. This computational approach for ecosystem modeling is called Ecological Informatics, with artificial neural networks (ANNs) being one of the core computational approaches (Recknagel and Friedrich, 2003). ANNs are recognized as a powerful and general technique for machine learning

[*] Corresponding Author

because of their non-linear modeling abilities and capability to learn, adapt and exhibit some very basic human-like intelligence. Successful 'learning' of variable associations and trends is the key attribute in the successful development of ANNs and their practical applications. Many important knowledge-based network systems have been developed and successfully applied to diverse databases such as speech recognition, game playing, medical diagnosis, financial forecasting and industrial control (Mitchell, 1997).

A further benefit of such computational models is the transformation of knowledge into the form of human-comprehensible rules that could aid in the simulation of the system. The extracted knowledge could then allow for the development of more sophisticated models of the system. Here, we discuss the use of ANNs as the machine learning tool of choice to study ecological systems and various techniques for extracting knowledge from trained networks.

2. BACKGROUND

In recent years, a number of studies in ecological modeling have emerged. Groups of studies use ANN to model time series such as: water quality and population dynamics in ecosystems (Schleiter et al., 1999), function approximation based on satellite imaging of near-surface oceans (Gross et al., 1999) and classification of species of dinoflagellates in coastal waterways (Culverhouse et al., 1996). In these instances, the ANN is used as a prediction tool based on the 'black box' concept, whereas no knowledge is extracted from the network. Articles that investigate knowledge extraction techniques typically include three basic approaches and/or conceptualizations: 1) Neural Interpretation Diagrams (NIDs), 2) Garson's Algorithm, and 3) Sensitivity Analysis. For example, the NID and Garson's Algorithm were used in gaining predictive and explanatory insight into fish-habitat relationships (Olden and Jackson, 2001), and for studying phytoplankton succession by interpreting the interacting contributions based on a number of biotic and abiotic factors to account for seasonal variations (Olden, 2000). Another approach typically used in extracting knowledge from ANN's is Sensitivity Analysis in conjunction with Garson's Algorithm (Recknagel et al., 1997). An example of this approach is Lee et al.(2003), when they were used to model the occurrence of algal blooms in aquatic systems.

The first attempt of extraction rules from a neural network was research conducted by Gallant (1988) on connectionist expert systems. Classification rules describing the network's behavior were obtained by analyzing the role of attribute ordering in correctly classifying a problem. A variety of rule extraction methods have been developed since then for addressing the problem of comprehensibility in neural networks. Andrews et al. (1995) classifies rule extraction approaches into the following three categories, based on the view taken by the algorithms of the underlying network topology: decompositional (Fu, 1991; Towell and Shavlik, 1993), pedagogical (Craven and Shavlik, 1996a) and eclectic (Sestito and Dillon, 1992).

This paper only explores the pedagogical technique. The pedagogical techniques extract rules that map network inputs to outputs directly, effectively treating the neural network as a black box. A common strategy employed by these approaches for rule generation is that a candidate rule antecedent is generated based on domain knowledge or analysis of ranges of input attributes and the network's response is treated as the rule consequent. In Saito and Nakano(1990), useful rules are selected from a candidate rule set that is generated by examining input activation values of the network which then activates a given output unit. Several pedagogical approaches have also been developed for extracting decision tree representations of the neural network. Craven and Shavlik (1996) extract decision trees from trained neural networks using a novel algorithm named

TREPAN. This algorithm employs a greedy gain ratio criterion for evaluating attribute splits. Binary and M-of-N decision trees can be derived by this method.

One of the more pronounced drawbacks of neural networks is their lack of explanation capability. They act like 'black boxes' and do not provide any reasoning behind the conclusion of a learning system. However it is essential to understand the basis of decisions as this type of computer support systems are often used for decision critical applications. These decisions are extremely useful when they are in human comprehensible form. Representing the extracted knowledge in the form of 'if-then' rules is most likely the best method of explaining the output of a neural network model. Decision trees can be easily represented in the form of 'if-then' rules and hence extracting decision trees are probably one of the best methods of interpreting a neural network. This idea is described in Figure 1.

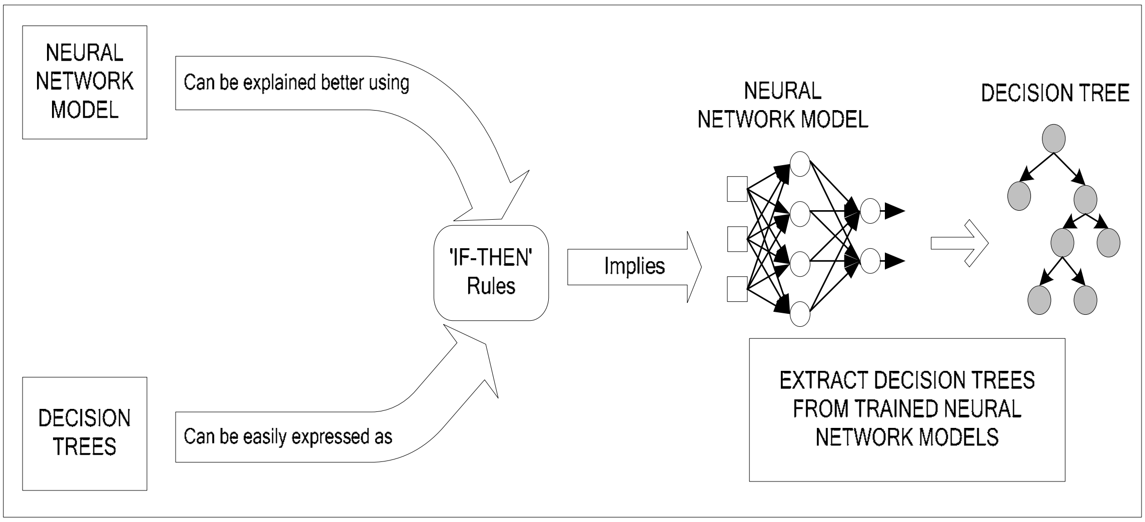
NEURAL NETWORK MODEL
Can be explained better using
'IF-THEN' Rules
Implies
NEURAL NETWORK MODEL
DECISION TREE
DECISION TREES
Can be easily expressed as
EXTRACT DECISION TREES FROM TRAINED NEURAL NETWORK MODELS

Figure 1. Extracting decision trees from neural networks

Decision trees are fast, simple to implement and can compile the model's learned hypothesis to a set of easily interpreted rules. More importantly, decision makers can utilize the derived trees because complex interacting factors are explained more effectively.

3. DEVELOPMENT OF A NEURAL NETWORK MODEL

There are three aspects related to development of a neural network model:

Choice of the training, cross-validation (CV) and testing data sets and their sizes;

Selection of suitable architecture, training algorithm and learning constants;

Determination of the termination criteria.

Considerable experimentation is necessary to achieve a good network model of the data. The software NeuroSolutions developed by NeuroDimensions Incorporated is used for development and testing of the neural network model (NeuroSolutions, 1995).

Saginaw Bay

This dataset comprises of real world data collected during 1991-1996 from the Saginaw Bay ecosystem in Michigan. Various indicator variables include water temperature, Secchi depth – a measure of water clarity (Secchi), total suspended solids (TSS), total phosphorous (TP), soluble reactive phosphorus (SRP), nitrate (NO3), ammonia (NH4), silica (SiO2), particulate silica (PSiO2), chloride (CL), particulate organic carbon (POC), and dissolved organic carbon (DOC). These indicators are used to predict water-column chlorophyll *a* concentration (CHL), a measure for total phytoplankton biomass. The Saginaw Bay data set is extremely complex with a large percentage of the output data concentrated in a relatively small region of space but, in a few instances, dispersed over a large range. These latter instances are comparable to rare event situations where very little data is available and modeling would be typically construed to be difficult. A sample of the Saginaw Bay dataset is shown in Table 1.

Table 1. Saginaw Bay-sample dataset

Sr. No.	**Temp**	**Secchi**	**TSS**	**TP**	**SRP**	**NO3**	**NH4**	**SiO2**	**PSiO2**	**CL**	**POC**	**DOC**	**CHL**
1	22.11	0.9	16	32.019	1.055	0.009	17.213	0.245	1.51	23.966	3.79	4.08	20.32
2	7.8	6.5	0.76	16.215	0.588	0.329	14.543	1.713	0.241	5.9	0.37	1.719	1.58
3	22.25	0.8	11.3	37.406	1.071	0.025	7.643	2.344	1.013	23.619	1.97	3.86	9.10769
4	15.3	0.4	23.91	32.602	1.46	0.647	18.143	0.085	3.69	20.087	1.97	3.769	22.628
5	8.77	2	4.32	13.648	0.92	1.17	13.6	1	1.36	26.3	0.95	4.12	7.45

A Multilayer Perceptron (MLP) neural network that included 12 inputs with 8 processing elements in the first hidden layer, 4 in the second hidden layer and one continuous output was trained. The best model resulted with a training mean squared error (MSE) of 0.00270 and a testing result as shown in Table 2. Both hidden layers contain neurons that employ hyperbolic tangent functions to map activations from input to output layers. The neural network architecture for the Saginaw Bay model is shown in Figure 2.

Table 2. Test results for the 12-8-4-1 MLP

Performance	**CHL**
MSE	6.268538215
MAE	1.495080971
Min Abs Error	0.027110829
Max Abs Error	13.64970128
r	0.897142826

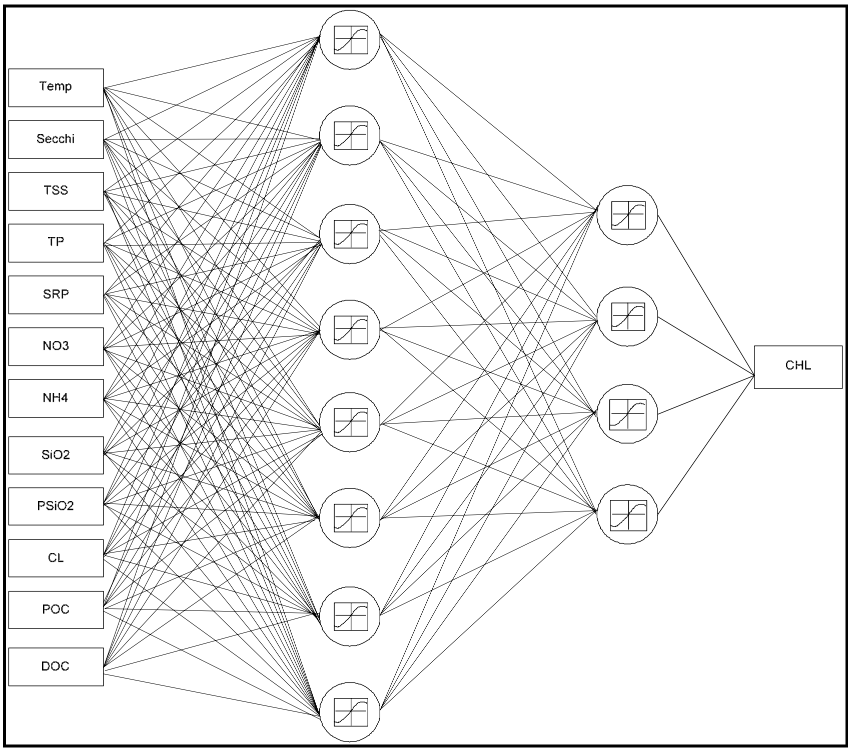

Figure 2. ANN Architecture of 12-8-4-1 MLP

4. NETWORK INTERPRETATION DIAGRAM (NID)

Approaches using node connection weights to interpret predictor variable contributions in neural networks have been used (Aoki and Komatsu, 1999; Chen and Ware, 1999) . A novel approach called the Neural Interpretation Diagram (NID) was proposed by Özesmi & Özesmi(1999). NID provides a visual tool to identify significant neurons or nodes by means of the connection weights. To construct a NID diagram, the nodes in various layers are arranged in a cascade as in Figure 2. The connections between various nodes are represented by lines, where the line thickness represents the relative magnitude of the connection weight. Thicker lines indicate greater connection strength andand thinner lines indicate a weaker connection. A distinction is also made regarding the sign of the connection weight in drawing the lines. A solid line indicates a positive or excitatory signal, while a negative or inhibitory connection is indicated by a dashed line. Thus, this diagram allows us to interpret visually both the strength and the sign of the connections. This can aid in identification of individual contributions on input nodes. Figure 3 illustrates the NID for the MLP 12-8-4-1 model and the relative influence of each input variable in predicting the output response.

Interpretation of the NID diagram for networks with more than one hidden layer is complex, particularly for networks of large size because of interaction. Interaction between nodes can be identified in a NID diagram when there is more than one significant connection coming into a node. An examination of the neurons in hidden layer 1 provides some possible clues regarding the significance of predictor variables and their interactions. A visual inspection of the connections between the input and the first hidden layer reveals that there are significant connections to nodes HN1, HN2 and HN8. The predictor variables corresponding to these connections are Temp, TP, NO3, SiO2, PSiO2, CL and POC. Thus seven of the thirteen input variables can be interpreted as significant based on the NID diagram. For possible interactions between these variables, the focus shifts to each significant node in the hidden layer. As an example, consider node HN1 which has significant connections from inputs TP and PSiO2. These two variables present a case of possible interaction. Following a similar approach, the following groups of variables have been identified as cases for potential interaction: {TP, PSiO2} based on HN1, {Temp, CL, and POC} based on HN2, {TP, SiO2, and CL} based on node HN8.

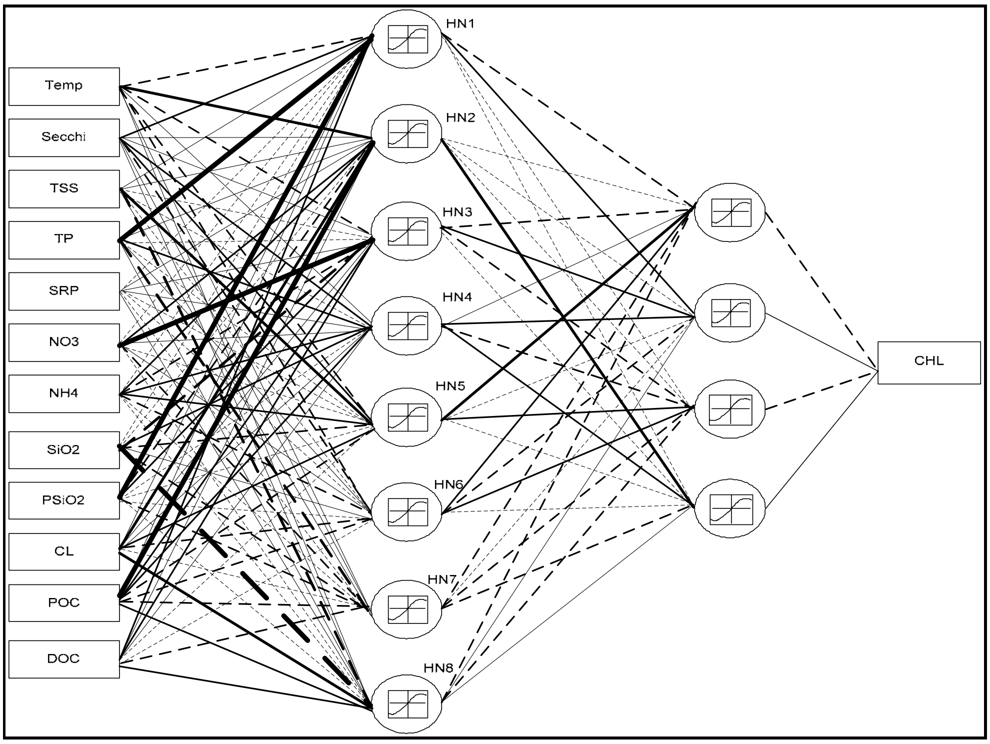

Figure 3. Network Interpretation Diagram (NID) of the MLP

The process can be repeated for the second hidden layer and the output layers. However, the interpretation becomes difficult as the outputs of hidden layers now represent a linear combination of inputs which is further modified by the use of transfer functions. It is preferable to use NID for simpler networks than for the one currently under consideration.

5. GARSON'S ALGORITHM

Garson (1991) proposed a method for partitioning the neural network connection weights in order to determine the relative importance of each input variable in the network. An example showing the application of Garson's algorithm in a single hidden layer Feed Forward MLP with two Processing Elements (PEs) is shown in Figure 4.

The basic steps in determining the relative weights of the inputs are as follows:

Step 1: Construct matrix containing input to hidden and hidden to output neuron connection weights.

Step 2: Calculate the contribution of each input neuron to the output (C_{A1}) via each hidden neuron as the product of the input-hidden connection (W_{A1}) and the hidden-output connection (W_{OA}).

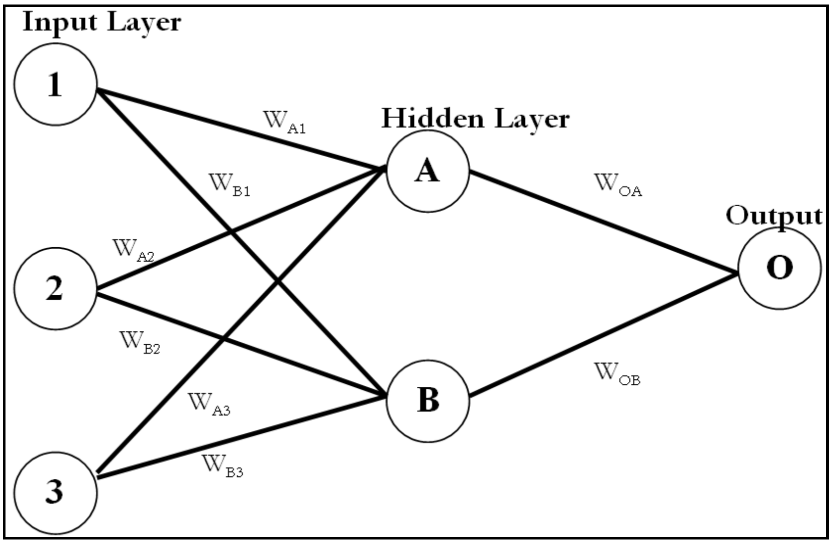

Figure 4. Network Diagram

$$C_{A1} = W_{A1} \times W_{OA} \quad (1)$$

Step 3: Calculate the relative contribution of each input neuron to the outgoing signal for each hidden neuron (R_{A1}) and the sum of all input neuron contributions (S_1)

$$R_{A1} = |C_{A1}| / (|C_{A1}| + |C_{A2}| + |C_{A3}|) \quad (2)$$

$$S_1 = R_{A1} + R_{B1} \quad (3)$$

Step 4: Calculate the relative importance (RI_1) of each input variable.

$$RI_1 = S_1 / ((S_1 + S_2 + S_3) \times 100) \quad (4)$$

For a more detailed step by step example of the calculations refer to Garson(1991). In this example, a smaller ANN was created for demonstration purposes, to identify the relative weights of six key factors taken from the Saginaw Bay database as identified by the researchers. The relative importance of the 6 major inputs are shown in Figure 5, where the most important attributes are DOC and TP.

Garson’s method is a good method to determine the overall influence of each predictor variable but may not provide accurate information regarding the interactions of predictor variables among themselves. For example, a large negative connection weight between the input-hidden layers coupled with a large positive connection weight between the hidden-output layers would result in a relatively important variable significance. However, in some cases, a negative connection weight between the input-hidden and a positive connection weight between the hidden-output layers could

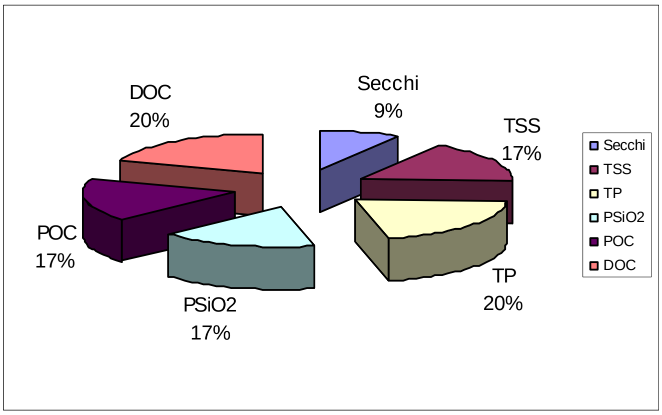

Figure 5. Pie Chart illustrating the relative importance of inputs

counteract each other and result in a final output which is not significantly important. Furthermore, Garson's algorithm uses the absolute values of the connection weights when calculating the variable contribution and would not always identify the countering effect of negative weights.

6. SENSITIVITY ANALYSIS

Sensitivity analysis is a method for extracting the cause and effect relationship between the inputs and outputs of the network. Traditional sensitivity analysis involves varying each input variable across its entire range while holding all other input variables constant, so that the individual contributions of each variable can be assessed. The inputs to the network are shifted slightly (defined number of standard deviations, both +/-) and the corresponding change in the output is observed. Figure 6 illustrates how CHL (output) varies over a range of values for PSiO2 (input).

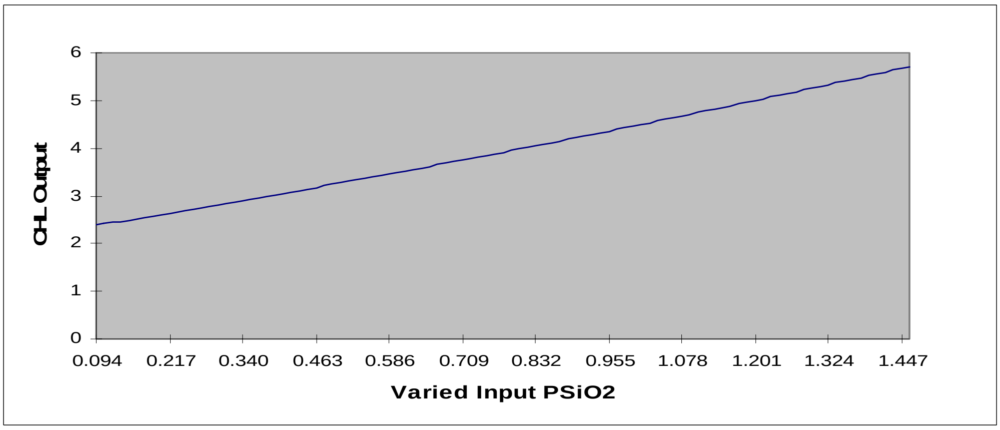

Figure 6. Sensitivity of input indicator PSiO2 versus output Chlorophyll

Sensitivity analysis provides feedback as to which input variables are the most significant relative to other input variables. An example of the significance of the input variables is shown in Figure 7. Based on feedback, the decision becomes which inputs could be pruned by removing the insignificant variables. This approach would reduce the size of the network, which in turn would help reduce the complexity of the network and the training times, but would also remove the impact and relationships that the input variable has to the output and other input variables.

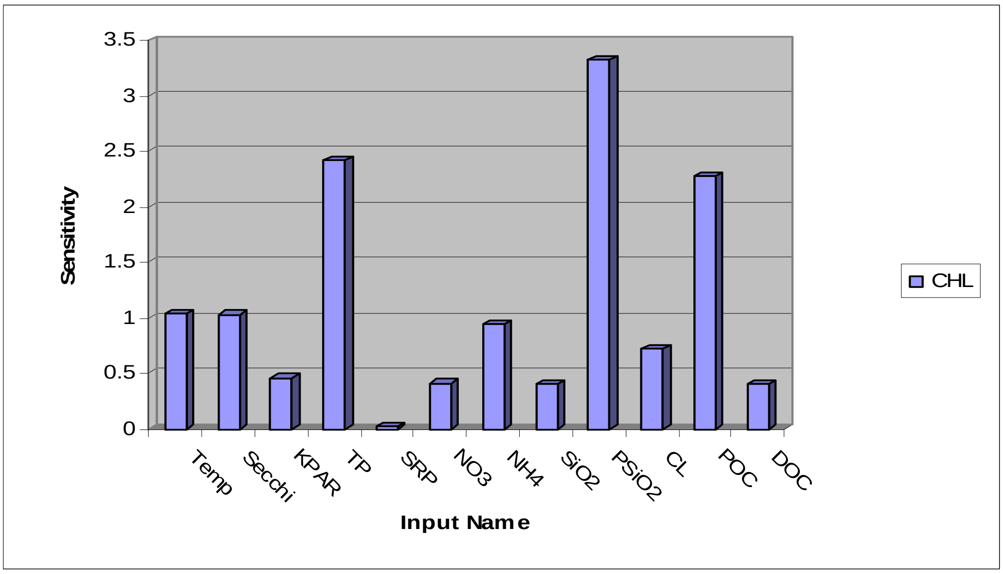

Figure 7. Sensitivity for CHL based on a MLP model with 12 inputs

In addition to the traditional sensitivity analysis, relationships between input variables (such as TP and Temp) can be explored by holding the output variable (CHL) constant, as shown in Figure 8. The relationships between the two input variables that correspond to the constant output can then be observed.

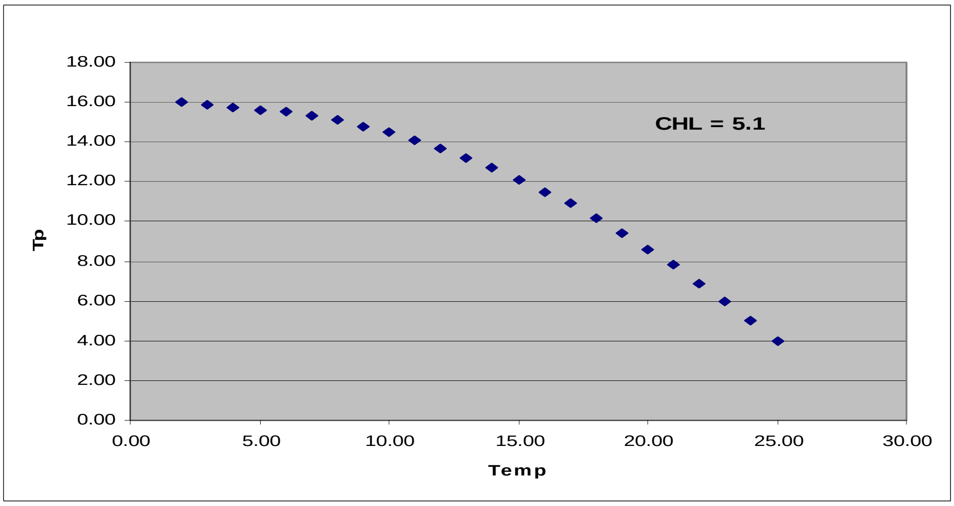

Figure 8. Relationship between TP and Temp at a constant Chlorophyll output

These relationships can be determined for multiple levels of the output variable (CHL) and a family of curves can be created as shown in Figure 9. This type of analyses shows that similar physical relationships exist between TP and Temp at CHL values of 1.52 to 10.08. However, it can also be observed that the physical relationships at a CHL level of 17.07 are not consistent with the lower values of CHC. These types of relationships can give scientists a better understanding of their ecological system.

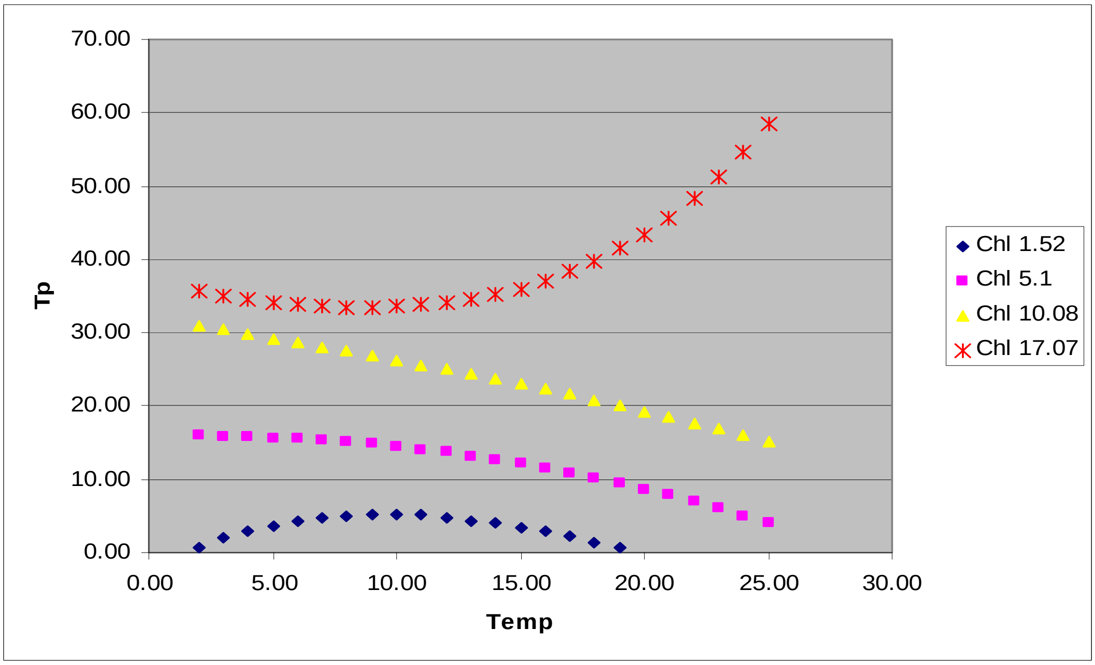

Figure 9. Family of Relationship curves between TP and Temp at a variety of constant CHL

A similar approach can be extended to investigate a family of surfaces based on knowledge obtained from training a neural network. An ANN model can easily be implemented into computer software by knowing the model's weights, biases, and structure. The inputs to the model can be assigned values based on statistical quantities, user specific, or values from a sample. A combination of input variables can be varied over a range, and the response of the output, CHL, can be observed. Figure 10 shows the response surface of CHL when values are varied for inputs Temp and POC.

It should be mentioned that this type of analysis has a distinct advantage of traditional sensitivity analysis, where values are locked to their mean values. In this type of knowledge extraction, the user assigns inputs to the desired value, which is important especially in a physical system such as this ecological system. It may not be physically possible to lock all of the input values to their mean.

Though this technique is very useful to understanding the neural network models, it is not limited to only investigating the system's response when input values are varied. Output values can also be varied over a specified range. Statistical optimization techniques can be utilized to generate additional surfaces by solving for input values that correspond to the desired output values. The goal of this optimizing strategy is to solve for an input value that minimizes the error of the desired value and the value obtained from the network.

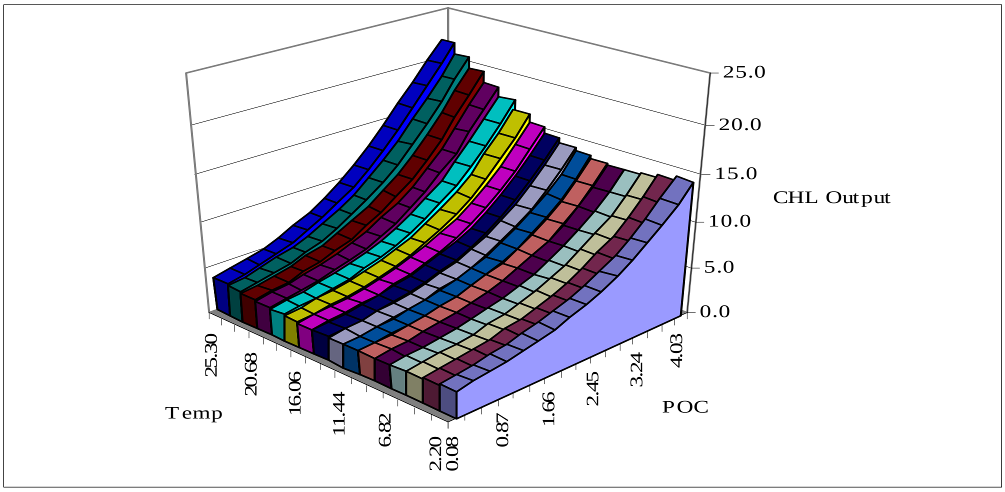

Figure 10. Family of Relationship Surfaces between Temp, POC, and Chlorophyll

It is also important that the solved value does not exceed the normalization limits of the neural network. If a solved value exceeds the normalization limit, it may not be a realistic value for the input. In this study, input parameters are normalized from negative to positive one. Therefore, the optimization condition is to minimize the error while not allowing input parameters to exceed the normalization limits. Figure 11 shows the effects of varying sensitive values (TP) and the output (CHL) and solving for another sensitive value (PSiO2).

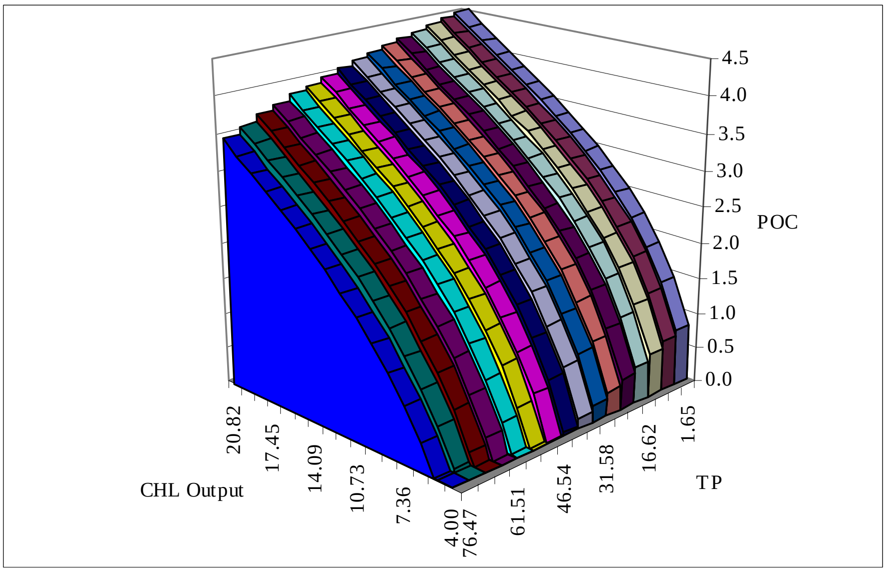

Figure 11. Family of Relationship Surfaces between CHL, TP, and POC

The surfaces generated can be used to better understand the physical phenomena being modeled. Maybe more importantly, the surfaces can also be used to eliminate the mentality of neural networks being a black box. If some of the relations being modeled by a neural network are known, this type of analysis can be used to provide evidence that the network is modeling the system correctly.

This type of modeling can also lead to developments in optimization. If, for example, the health of the aquatic life in the bay is in danger due to a particular attribute being high or low, a scientist may be able to observe the relationships of a controllable attribute and the one influencing aquatic life and control the effect to promote better condition for the aquatic life.

7. KNOWLEDGE EXTRACTION UTILIZING TREPAN

Although neural networks are known to be robust classifiers, they have found limited use in decision-critical applications such as in medical systems. Trained neural networks act like black boxes and are often difficult to interpret (Towell and Shavlik, 1993). The availability of a system that would provide an explanation of the input/output mappings of a neural network in the form of rules would thus be very useful. Rule extraction is one such system that tries to elucidate to the user, how the neural network arrived at its decision in the form of if-then rules.

Two explicit approaches have been defined to date for transforming the knowledge and weights contained in a neural network into a set of symbolic rules-decompositional and pedagogical (Craven and Shavlik, 1995). In the decompositional approach, the focus is on extracting rules at an individual hidden and/or output level into a binary outcome. It involves the analysis of the weight vectors and biases associated with the processing elements in general. The Subset algorithm is an example of this category (Fu, 1991). The pedagogical approach treats neural networks like black boxes and aims to extract rules that map inputs directly to its outputs. The Validity Interval Analysis (VIA) proposed by Thrun (1995) and TREPAN (Craven, 1996) is an example of one such technique. Andrews et al (1995) propose a third category called eclectic which combines the elements of the two basic categories. The DEDEC algorithm is representative of this category (Tickle et al., 1996). In this paper, we will concentrate on the pedagogical approach using TREPAN.

Decision Trees

Decision trees classify data through recursive partitioning of the data set into mutually exclusive subsets which best explain the variation in the dependent variable under observation(Biggs et al., 1991; Liepins et al., 1990). Decision trees classify instances (data points) by sorting them down the tree from the root node to some leaf node. This leaf node gives the classification of the instance. Each branch of the decision tree represents a possible scenario of decision and its outcome. Decision tree algorithms depict concept descriptions in the form of a tree structure. Each node of a decision tree specifies a test of some attribute and each branch that descends from the node corresponds to a possible value for this attribute.

Decision trees have been proved useful in their applications to various real world problems. Leech (1986) applied a decision tree induction approach to a chemical nuclear power plant process. Michie (1989) used an induction algorithm to produce a decision tree for making decisions whether to grant credit to a loan applicant or not. Evans and Fisher (1994) applied decision tree induction to the problem of banding in Rotogravure printing. In the manufacturing and production industry, decision trees have been used in non-destructive testing of spot weld quality(Ercil, 1993), productivity enhancements(Kennedy, 1993), semiconductor manufacturing(Irani et al., 1993), material procurement method selection(Das and Bhambri, 1994), process optimization(Famili, 1994), assembly line scheduling of printed circuit boards(Piramathu et al., 1994), to uncover flaws in a Boeing manufacturing process(Riddle et al., 1994), separation of gas from oil(Guilfoyle, 1986), quality control (Guo and Dooley, 1994) and a hybrid system to extract rules for job shop scheduling (Ganduri, 2004).

Other applications include the areas of Biomedical Engineering, Image Processing, Language Processing, Law, Medicine, Molecular Biology, Pharmacology, Physics, and Plant diseases (Murthy, 1998). Decision trees continue to be an active research area, the current focus being on improving methods for building, controlling and executing the decision tree algorithms to achieve maximum efficiency.

TREPAN Algorithm

The TREPAN algorithm developed by Craven (1996a, b) is a novel rule-extraction algorithm that mimics the behavior of a neural network. Given a trained Neural Network, TREPAN extracts decision trees that provide a close approximation to the function represented by the network. This work is concerned with its application to trained neural network models. It can be applied not only to neural networks but to a wide variety of learned models as well. TREPAN uses a concept of recursive partitioning similar to other decision tree induction algorithms. In contrast to the depth-first growth used by other decision tree algorithms, TREPAN expands using the best first principle. That node which increases the fidelity of the tree when expanded is deemed the best. Fidelity measures the extent to which the extracted decision tree faithfully replicates the behavior of the neural network.

In conventional decision tree construction algorithms, the amount of training data decreases as one traverses down the tree by selecting splitting tests. Thus there is not enough data at the bottom of the tree for deciding class memberships of the training data present at the leaf node. To increase the amount of data points at these nodes, TREPAN uses the trained neural network as an ‘Oracle’ to answer queries to hypothetical data points created from known distributions of input attributes, in addition to the training samples during the inductive learning process. This learning from larger samples can avoid the lack of examples for the splitting tests at lower levels of the tree, which is usually a problem with conventional decision tree learning algorithms.

In the case of TREPAN analysis for regression problems, it is required that the range of the response variable is segmented into classes. This is done by dividing the output into discrete values based on population density. The class sizes for the level of chlorophyll for the Saginaw Bay dataset are shown in Table 3.

Table 1. Chlorophyll level class labels

CHL	Class
0-8.99	Cl1
9-17.99	Cl2
18-26.99	Cl3
27-35.99	Cl4
36 and above	Cl5

The confusion matrix for the TREPAN model (decision tree) having a classification accuracy of 88.11% is shown in Table 4. A decision tree of size 15 (number of leaf nodes) was obtained and is depicted in Figure 12.

Table 4. Saginaw Bay Confusion Matrix (TREPAN)

Saginaw Bay: Confusion Matrix (TREPAN)					
Actual/Desired	Cl1	Cl2	Cl3	Cl4	Cl5
Cl1	183	13	0	0	0
Cl2	8	29	3	1	0
Cl3	0	1	3	1	0
Cl4	0	1	1	0	0
Cl5	0	0	0	0	0
Classification Accuracy (%)	95.81%	65.91%	42.86%	0.00%	0.00%
Total Accuracy (%)	88.11%				

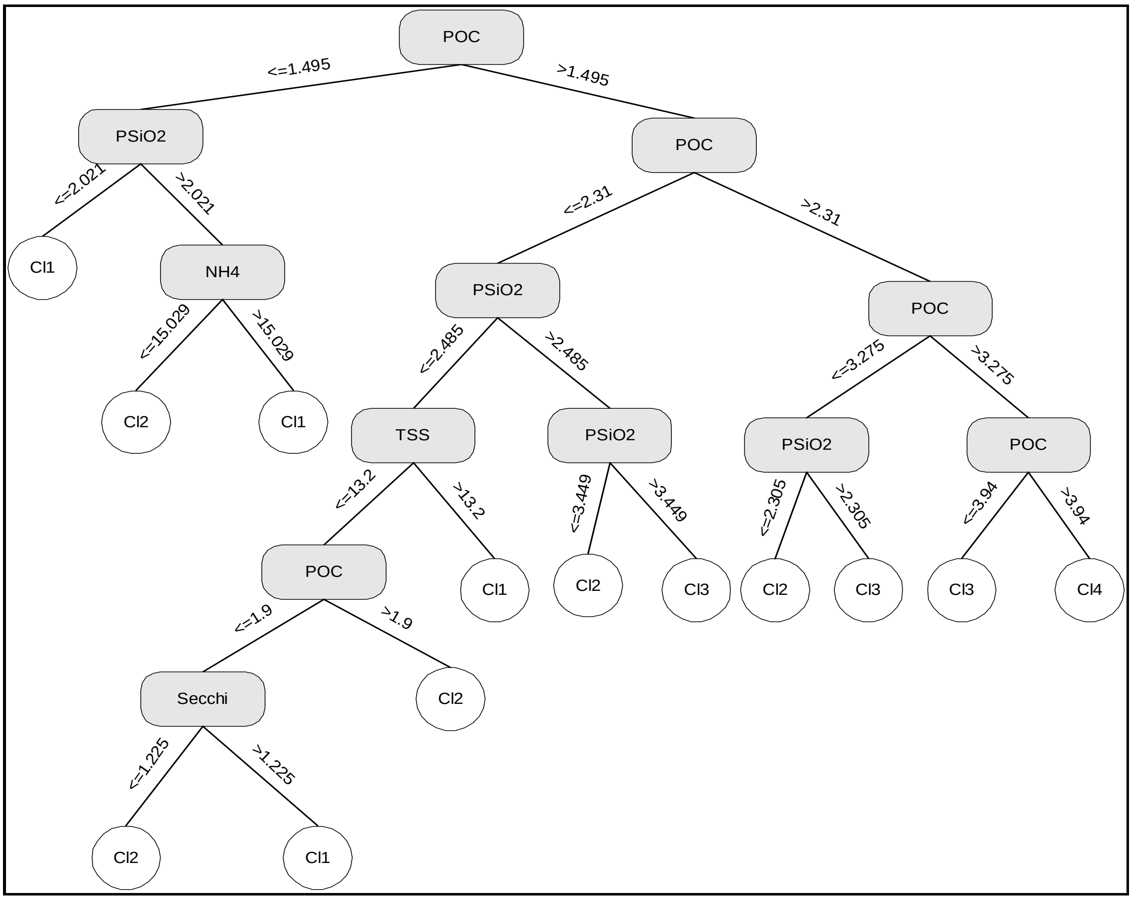

Figure 12. Saginaw Bay: TREPAN Decision Tree

8. CONCLUSIONS AND FUTURE RESEARCH

This paper explored a number of knowledge extraction techniques for ANNs, from basic to complex algorithms and the quality of information gained from these techniques was discussed. It has demonstrated how these techniques can be used to extract various levels of knowledge from various ANNs that were used to model a complex ecological system such as Saginaw Bay.

Chlorophyll *a* concentrations were accurately predicted by the ANN models, and as such, proved to be an acceptable methodology for assessing the generalized health of the system. Techniques were explored that would allow the investigator to go beyond the limitations typically referred to as a "black box".

The current research explores different tools for extracting knowledge from a trained neural network model for ecological modeling. The Neural Interpretation Diagram (NID) provides a visual tool for identifying the relative significance and interactions between different inputs. While this can be a useful technique for smaller networks, interpretation is difficult for larger, multi-layered neural network models as those utilized in this research. Garson's method and sensitivity analysis are two different techniques used to assess the relative contribution of each predictor variable to the output variable. While both techniques suffer from some drawbacks particularly when a high degree of interaction between predictor variables can be expected, there are two key benefits. They could be used in the model-building stage to reduce the complexity of the model by elimating variables which either does not significantly impact the output variable or whose impact could be assessed by an input variable already in the model. A second benefit is in the model-validation stage when existing domain knowledge could be used to evaluate the variables deemed significant by these methods. It is important to consider domain knowledge as a good predictive model implies association between the input variables and the output variable, but not necessarily causation. If the purpose of the research is to reveal the underlying mechanistic model, it is important to include all inputs which are considered relevant. Of the two approaches, sensitivity analysis provides more useful information particularly when used to generate response surfaces of the output variables with several inputs.

A more comprehensible approach to knowledge extraction was also illustrated by the use of a TREPAN algorithm for extracting decision trees from neural networks. An equivalent decision tree representation of the neural network was constructed using this method. The induced decision tree had good classification accuracy on the testing data set. This approach provides the best means of interpreting the neural network because it combines the power of the highly incomprehensible neural network black box and simple to understand decision trees. This considerably increases the confidence in using the neural network as a predictive tool. By extracting knowledge into a comprehensible form, further relationships between input and output variables can be explored and can be used to support the generation rule based systems. These extracted comprehensible rules aid in developing a more usable prediction tool and enhance the understanding of the bay's actual ecological system.

Future research in this area will take the next step in the development of a knowledge based model. In addition, knowledge will be derived from the first principles on the "true" process behavior of an ecological system and combine them to form the initial mechanistic model as shown in Figure 13. The mechanistic model will represent the complex ecological process through a set of equations that express both existing knowledge and what can be derived from the extracted ANN knowledge. It is anticipated that the initial development of the mechanistic model may not forecast ecological behavior with sufficient accuracy. The complexity and variability of ecological systems make it difficult to model its effects. This is typically due to the fact that the validity of the results is closely linked to the amount of data available and the experience and knowledge that accompany the analysis. In addition, the complexity of the physical/chemical indicators makes it difficult to generalize their behavior or to develop a mechanistic model.

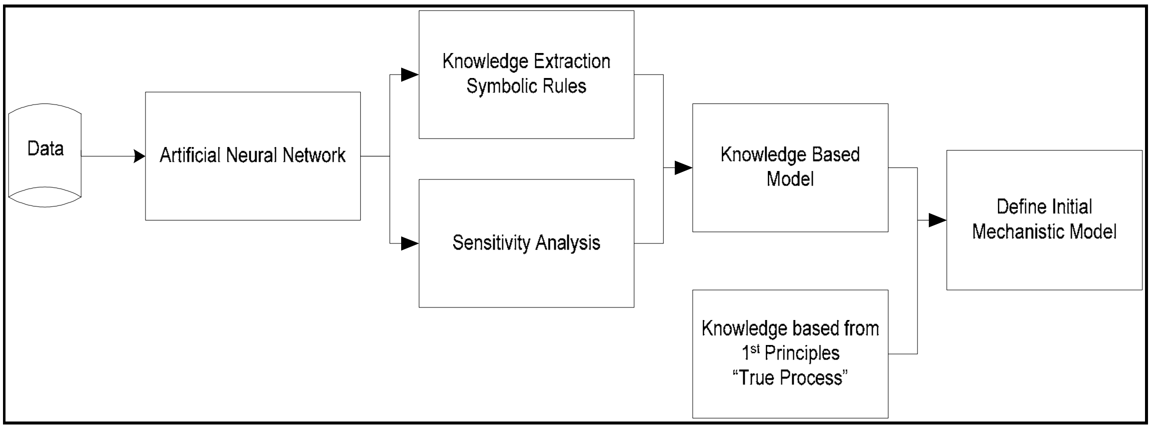

Figure 13. Overview of Initial Development of Mechanistic Model

REFERENCES

[1] Andrews R., Diederich, Tickle A. B. (1995), Survey and critique of techniques for extracting rules from trained artificial neural networks; *Knowledge Based Systems* 8; 373-389.

[2] Aoki I., Komatsu T. (1999). Analysis and prediction of the fluctuation of sardine abundance using a neural network; *Oceanol. Acta* 20; 81–88.

[3] Biggs D., de Ville B., Suen E. (1991), A method of choosing multiway partitions for classification and decision tree; *Journal of Applied Statistics* 18(1); 49-62.

[4] Chen D.G., Ware D.M. (1999), A neural network model for forecasting fish stock recruitment, *Can. J. Fish. Aquat. Sci.* 56; 2385–2396

[5] Craven M.W, Shavlik J.W. (1995), Using sampling and queries to extract rules from trained neural networks; Machine Learning. Proceedings of the Eleventh Inter-national Conference, Cohen W.W & Hirsh H. (Eds.), San Francisco, CA: Morgan Kaufmann.

[6] Craven M.W., Shavlik J.W. (1996a), Extracting tree-structured representations of trained networks; *Advances in Neural Information Processing* 8; 24-30.

[7] Craven M.W. (1996b), Extracting Comprehensible models from trained Neural Networks; *PhD Thesis*; Computer Science Department, University of Wisconsin, Madison, WI.

[8] Culverhouse PF., Simpson RG., Ellis R., Lindley JA., Williams R., Parisini T., Reguera B., Bravo I., Zoppoli R., Earnshaw G., McCall H., Smith G. (1996), Automatic Classification of Field Collected Dinoflagellates by Artificial Neural Network; *Mar. Ecol. Prog. Ser.* 139(1-3); 281-287.

[9] Das S.K., Bhambri S. (1994), A decision tree approach for selecting between demand based, reorder and JIT/Kanban methods for material procurement; *Production Planning and Control* 5(4); 342.

[10] Ercil A. (1993), Classification trees prove useful in non destructive testing of spot weld quality, *Welding Journal, Sept., Issue title: Special emphasis: Rebuilding America's roads, railways and bridges* 72(9); 59.

[11] Evans B., Fisher D. (1994), Overcoming process delays with decision tree induction; *IEEE Expert* 9; 60-66.

[12] Famili A. (1994), Use of Decision Tree Induction for Process Optimization and Knowledge Refinement of an Industrial Process, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM), Winter* 8(1); 63-75.

[13] Fu L. (1991), Rule learning by searching on adapted nets; *In Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA*; 590-595.

[14] Ganduri C.G. (2004), Rule driven job shop scheduling derived from neural networks through extraction; *M. S. Thesis*, Department of Industrial Engineering, Ohio University, Athens, Ohio.

[15] Gallant S.I. (1988), Connectionist expert systems, *Communications of the ACM* 31; 152-169.

[16] Garson G.D. (1991), Interpreting neural network connection weights; *Artificial Intelligence Expert* 6; 47-51.

[17] Gross L., Thiria S., Frouin R. (1999), Applying artificial neural network methodology to ocean color remote sensing; *Ecological Modeling* 120; 237-246.

[18] Guilfoyle C. (1986), Ten minutes to lay the foundations; *Expert Systems User* (Aug.), 16-19.

[19] Guo Y., Dooley K.J. (1994), Distinguishing between mean, variance and autocorrelation changes in statistical quality control; *International Journal of Production Research* 33(2); 497-510.

[20] Irani K., Jie C., Fayyad U. M., Zhaogang Q. (1993). Applying machine learning to semiconductor manufacturing; *IEEE Expert, Feb.*, 8(1); 41-47.

[21] Kennedy D.M. (1993), Decision tree bears fruit; *Products Finishing* 57(10); 66.

[22] Lee J., Huang Y., Dickman M., Jayawardena A.W. (2003), Neural network modelling of coastal algal blooms; *Ecological modeling* 159(2-3); 179-201.

[23] Leech W.J. (1986), A rule based process control method with feedback; *Advances in Instrumentation* 41; 169-175.

[24] Liepins G., Goeltz R., Rush R. (1990), Machine learning techniques for natural resource data analysis; *AI Applications* 4(3); 9-18.

[25] Michie D. (1989), Problems of computer-aided concept formation, In Quinlan, J.R., (Ed). Applications of Expert Systems Volume 2. Wokingham, UK: Addison-Wesley, 310-333.

[26] Mitchell T. (1997), Machine learning; 1st edition, Computer Science Series, Boston, MA; WCB McGraw-Hill.

[27] Murthy S.K. (1998), Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey; *Data Mining and Knowledge Discovery* 2(4); 345-389.

[28] NeuroSolutions (1995), Software developed and distributed by Neurodimension Incorporated; [http://www.neurosolutions.com/products/ns/].

[29] Olden J.D., Jackson D.A. (2001), Fish-Habitat Relationships in Lakes: Gaining Predictive and Explanatory Insight by Artificial Neural Networks; *Transactions of the American Fisheries Society* 130; 878-897.

[30] Olden J.D. (2000), An artificial neural network approach for studying phytoplankton succession; *Hydrobiologia* 436; 131-143.

[31] Özesmi S.L., Özesmi U. (1999), An artificial neural network approach to spatial habitat modelling with interspecific interaction; *Ecol. Model*. 116; 15–31.

[32] Piramuthu S., Raman N., Shaw M.J. (1994), Learning-based scheduling in a flexible manufacturing flow line, *IEEE Trans. on Engineering Management* 41(2); 172-182.

[33] Recknagel Friedrich (2003), Ecological Informatics: Understanding Ecology by Biologically-Inspired Computation; New York, NY, Springer.

[34] Recknagel F., French M., Harkonen P., Yabunake K. (1997), *Ecological Modelling* 96; 11-28.

[35] Riddle P., Segal R., Etzioni O. (1994), Representation, Design and Brute-force Induction in a Boeing manufacturing domain; *Applied Artificial Intelligence* 8(1); 125-147.

[36] Saito K., Nakano R. (1990), Rule extraction from facts and neural networks; *Proceedings of the International Neural Network Conference*; San Diego, CA, 379-382.

[37] Schleiter I.M., Borrchardt D., Wagner R., Dapper T., Schmidt K., Schmidt H., Werner H. (1999), Modeling water quality, bioindication and population dynamics in lotic ecosystems using neural networks; *Ecological Modelling* 120; 271-286.

[38] Sestito S., Dillon T. (1992), Automated knowledge acquisition of rules with continuously valued attributes; Proceedings of the Twelfth International Conference on Expert Systems and their Application, Avignon, France, 645-656.

[39] Thrun S. (1995), Extracting rules from artificial neural networks with distributed representations, In Tesauro G.,Touretzky D., and Leen T. eds. Advances in Neural Information Processing Systems 7; Cambridge, MA: MIT Press; 505-512.

[40] Tickle A.B., Orlowski M., Diederich J. (1996), DEDEC: a methodology for extracting rule from trained artificial neural networks; In: Proceedings of the AISB'96 Workshop on Rule Extraction from Trained Neural Networks, Brighton, UK; 90-102.

[41] Towell G.G., Shavlik J.W. (1993), Extracting refined rules from knowledge-based neural networks; *Machine Learning* 13; 71-101.