1995

Nuclear plant diagnostics using neural networks with dynamic input selection

Anujit Basu
Iowa State University

Nuclear plant diagnostics using neural networks with dynamic input selection.

by

Anujit Basu

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department: Mechanical Engineering
Major: Nuclear Engineering

Approved:

Signature was redacted for privacy

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Members of the Committee:

Signature was redacted for privacy.

Signature was redacted for privacy.

Signature was redacted for privacy

Signature was redacted for privacy.

Iowa State University
Ames, Iowa
1995

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I wish to thank IES Utilities and the United States Department of Energy for partial funding of this research. Thanks are also due to the Duane Arnold Energy Center for their cooperation in providing information and use of their control room simulator. Without their help, this research would not have been possible. I would also like to thank the faculty, staff, and students of the Nuclear Engineering program who have made my stay at Iowa State University a very enjoyable learning experience. I would especially like to thank the members of the Adaptive Computing Laboratory for their support and input to this research. Special thanks to Dr. Eric Bartlett who has been both friend and guide over the past four years. I would also like to thank Dr. Bullen, Dr. Leucke, Dr. Sheble and Dr. Udpa for being on my program committee. I would like to thank my parents who have always emphasized the importance of higher education. Last, but not least, I would like to thank my wife, Maria, who has been a source of strength through very trying situations.

GENERAL INTRODUCTION

Concern for nuclear power plant safety has been steadily increasing since the beginning of the nuclear power era. This increase in public concern has been accompanied by advances made by the nuclear power industry in the design and operation of nuclear power plants. Presently, plant safety systems monitor various instrumentation readings to check if the plant conforms to preset safety limits. If any of the instrumentation readings exceeds the preset limits, the information is relayed to the plant operators through alarms and warnings. The response of the operators to the abnormal situation is procedural in nature. These procedures are based on the symptoms of the plant, and are developed through prior extensive analysis of the plant. The operator response is based on the symptoms of the plant behavior, and not on the cause of the symptoms.

An automated real-time fault diagnostic system would provide information to the operators about the cause of the abnormality. The operators can use this additional information to validate the appropriateness of their procedural actions. An early diagnosis of an impending safety trip can provide the operators adequate lead time to bring the plant to a safe shutdown, which has a lesser impact on the plant components than a safety trip. Such a fault diagnostic system can also be used at the technical support center to monitor the health of the plant. The information from the system

can be used during a post-event analysis to prevent a future recurrence.

The work presented in this dissertation explores the design and development of a nuclear power plant fault diagnostic system based on artificial neural networks (ANNs), a form of artificial intelligence (AI). ANNs have been chosen over other forms of AI due to the advantages they provide. The ANNs learn from example data and thus do not need to be provided with explicit rules regarding the system that is being modeled. Moreover, the fault and noise tolerant capabilities make the ANNs well suited to the task of fault diagnosis in a system as complex as a nuclear power plant.

One of the problems commonly encountered by ANN users is the selection of an ANN architecture. Currently, the user guesses at the input variables required to model a system. This guess is based on the user's expertise of the system being modeled. Also, the size of the hidden layer is often guessed at using rules of thumb. The common practice is to train different ANNs with different inputs and hidden nodes, and use the model that displays the best post - training characteristics. This approach is computation intensive, and is not guaranteed to provide the best possible model.

One of the objective of this research is to develop a scheme that allows an ANN to automatically select the input variables that it deems necessary during the training process. This scheme should also allow the ANN to vary the number of hidden nodes during training to arrive at a viable model. Such a scheme would significantly improve the usability of ANNs. This scheme will be used to develop a fault diagnostic adviser for a nuclear power plant. Using these techniques, the adviser will be assembled out o individual ANNs. each of which will act as a module. This modular design

will allow for future expansion of the existing adviser. The ability to develop ANN models without guesswork will lead to significant reduction in developmental time for the adviser. The multiple ANN based design of the adviser would require such a capability for practical development of the adviser.

Dissertation Organization

The main body of this dissertation comprises of three self-contained papers that represent three distinct phases of this research. They have been presented in chronological order. Anujit Basu is the principal investigator and the first author of the papers presented here; Dr. E.B. Bartlett appears as the second author.

The first paper has been published in Nuclear Science and Engineering. This paper describes a dynamic node architecture (DNA) scheme applied to developing an ANN-based fault diagnostic adviser for a nuclear power plant. The DNA scheme allows the dynamic growth and contraction of the hidden layer to arrive at the optimum hidden layer size without recourse to guesswork. The adviser described in the first paper is capable to detecting and classifying 27 distinct transients.

The second paper has been submitted to IEEE Transactions on Neural Networks. This paper describes a dynamic input selection (DIS) scheme that allows an ANN to dynamically select inputs and hidden layers during training to arrive at a viable model. DIS is an extension and modification of the DNA scheme. Examples are provided to explain the workings of DIS, and demonstrate its advantages. The figures and tables for this paper have been collected at the end of the paper.

The third paper has been submitted to Neural Computing and Applications. This paper presents the development of a nuclear power plant fault diagnostic adviser

based on a modular design. This design allows the adviser to be built up of several different ANNs, each performing a part of the diagnostic task. This modular design also allows the expansion of the adviser to include new transients. The ANNs in this adviser are trained using DIS. The figures and tables of this paper are collected at the end of the paper.

The three papers are followed by a general summary that summarizes the entire work. It also addresses some of the issues arising from this work, and future avenues for investigation. A list of additional references cited in the general summary are also included. The transient scenarios used to develop the advisers in this work are described in the appendix.

CHAPTER 1. DETECTING FAULTS IN A NUCLEAR POWER PLANT USING DYNAMIC NODE ARCHITECTURE ARTIFICIAL NEURAL NETWORKS

A paper published in Nuclear Science and Engineering

Anujit Basu and Eric B. Bartlett

ABSTRACT

This paper describes an artificial neural network- (ANN-) based diagnostic adviser capable of identifying the operating status of a nuclear power plant. A dynamic node architecture (DNA) scheme is used to optimize the architectures of the two backpropagation ANNs which embody the adviser. The first or root network is used to determine whether the plant is in a normal operating condition or not. If the plant is not in a normal condition, the second, or classifier network is used to recognize the particular off-normal condition or transient taking place. These networks are developed using simulated plant behavior during both normal and abnormal conditions. Data from the nuclear generating station training simulator at Iowa Electric Light and Power Company's Duane Arnold Energy Center (DAEC) is used in this work. The adviser is effective at diagnosing 27 distinct transients based on 43 scenarios simulated at various severities which contain up to 3% noise.

1. INTRODUCTION

The safe operation of nuclear power reactors is very important to the nuclear engineering community and vital to creating a more positive attitude towards nuclear energy. This paper explains how artificial neural networks (ANNs) can be used to increase the operational safety of nuclear reactors by being the basis of a fault diagnostic system for a nuclear power plant. Faults in the various systems and components of nuclear power plants cause distinct behavioral transients. These transients can be identified and the particular faults that cause them detected. It is hoped that neurocomputing, as the science of neural networks is sometimes called, will provide an important contribution towards the recognition and classification of operational transients at nuclear power plants and thereby improve their safety.

The feasibility of using ANNs for a fault diagnostic adviser has been demonstrated in earlier work [3]. The present work describes the evolution of this earlier research and develops and enhances the previously used approaches for boiling water reactors (BWRs) as opposed to pressurized water reactors (PWRs). The present work also demonstrates the usefulness of ANN techniques for diagnosing many more transients over a wider range of severities. The diagnostic task investigated here requires the detection of 27 transients at a BWR using 97 plant variables as inputs as opposed to 7 transients and 27 variables in the earlier investigation. Not only does the increase in the number of variables and transients make the adviser more applicable but it also significantly increases the difficulty of obtaining a solution. This is due to the non-polynomial-time completeness of neural network learning [16]. Therefore, as the number of variables n in the problem increases, the complexity increases faster than a polynomial of order n [17]. This makes larger problems very difficult to solve

using ANNs. In this work the training complexity is reduced by using a modular hierarchy of two ANNs which distributes the burden of the diagnosis. This hierarchical approach has many advantages as the number of transients and variables and therefore ANNs is increased. For example, the individual component networks in this hierarchy are trained independently and simultaneously. Moreover, all the ANNs in the hierarchy do not need to use the same input variables. The system developer is then free to use only those variables needed for each ANN in the hierarchy to complete its particular task. This work demonstrates the feasibility of the hierarchical approach.

The ANNs used in this work utilize the backpropagation learning algorithm in contrast to the stochastic learning algorithm used in the previous work [3]. Backpropagation is a much more widely understood and accepted paradigm. A dynamic node architecture (DNA) scheme was developed for backpropagation networks that eliminates the need for preset architectures, thus eliminating another difficulty associated with ANN applications [14]. The successful implementation of the DNA scheme in a backpropagation network enables the development of the relatively large networks without recourse to guesswork inherent with the usual architecture preselection.

Power plants currently employ automatic safety systems that allow the plant to operate within a predefined operating range. These systems verify that plant variables conform to preassigned safety limits. If a plant variable exits from its safety range, safety systems either trigger an automatic reactor shutdown (safety trip) or notify the operators of the violation through alarms or indicators. Corrective actions performed by the operators following such events are based on procedures, which address symptoms. The operators are trained to react to each symptom by follow-

ing specific procedures. Developed by extensive offsite analysis, these procedures are designed to put the plant in a safe condition with minimal risk or damage to the environment. The root cause of a transient however does not effect the procedural response. After the plant is in a safe condition, the sequence of events leading to a plant shutdown or alarm are analyzed by technical review teams comprising of operators and shift and plant engineers [5]. The purpose of such reviews is to determine root cause and methods to avoid similar transients.

The operators in a nuclear power plant are expected to monitor a large number and variety of plant parameters. These parameters, or variables are relayed to the operator in the control room by instrumentation located in the various systems that comprise the plant. It is conceivable that the operators can misread or misinterpret instrumentation readings during the stressful situation following an abnormal event. The responsibility on the operator can be overwhelming. In such situations, a diagnosis by the ANN-based plant diagnostic adviser could reassure the operators about the adequacy of their procedural actions. The adviser's diagnosis could also provide clues to the operator about certain symptoms that might have been overlooked or are likely to develop in the near future. The adviser would be able to diagnose many transients in enough time to warn the operators of an impending automatic safety trip. The operators can then, in accordance with procedures, carry out a manual reactor trip that would be safer and easier on the plant components as compared to a fast automatic shutdown. Later during a technical review, a time record of the instantaneous response of the ANN adviser could be a valuable tool in understanding the evolution of the abnormal event. The quick diagnosis of a transient by the ANN adviser could help the reviewers evaluate the appropriateness of the emergency

operating procedures. The adequacy of operator actions can also be evaluated in this light.

Nuclear power plant diagnostics can also be performed by expert systems. Such diagnostic systems usually rely on elaborate decision trees to evaluate the current plant status [6, 7, 8]. The most notable effort in this regard is the Reactor Safety Assessment System (RSAS) developed jointly by the Idaho National Engineering Laboratory, the University of Maryland, and the U. S. Nuclear Regulatory Commission [6, 9]. Expert systems such as RSAS require knowledge to be specifically enumerated and incorporated into their knowledge base about every aspect of the system that is analyzed. This knowledge can be stored as numerous if-then logic statements that assist the expert system in performing a fault-tree type analysis [10, 11]. For nuclear power plants, this enumeration process for an exhaustive root cause analysis can require large amounts of personnel time. Every conceivable scenario should be analyzed to assure that as many transients as possible can be differentiated. The temporal behavior of all important variables is also required. All of this information must be manually incorporated into the expert system in a precise and methodical way. On the other hand, ANNs do not need an explicit knowledge base of this kind. The designer need only have a working model of the plant and a knowledge that certain sensors are important indicators of the health of the plant. Most nuclear power plants have such a working model in the form of a training simulator. ANNs learn their responses from sets of training data and then generalize this information to the actual operating plant. A training data set is a collection of input-output patterns that are used by the network to infer the functional relationship between their inputs and outputs. Generalization in this context is the ability to quantitatively estimate

certain characteristics or features of a phenomenon never before encountered on the basis of similarities with things previously known [14].

Most of the possible transients at nuclear power plants have never actually occurred and data for such scenarios are obtained through computer simulation. The noise and fault tolerant capabilities of neural networks are advantageous in developing an adviser based on such simulated data. These data are generated on a computer and do not contain any noise. But the adviser is expected to perform under real plant conditions where the input data can be expected to be corrupted by noise. The generalization capability of ANNs enables them to correctly classify patterns containing such noise. Fault tolerance of the neural networks is advantageous when there is failure of plant instrumentation. In such situations, an expert system might fail catastrophically if certain decisions in its analysis are based on such failed inputs. On the other hand, the ANN adviser could perform relatively accurately, disregarding the particular sensor fault. This fault tolerance capability of ANNs is due to their distributed memory. As the number of failed sensors increase, the performance of the ANN adviser will gradually deteriorate. Moreover, a neural network-based adviser system can provide better clues during event evaluations because it can produce its output in real time.

The following section gives a brief introduction to ANNs. This section also introduces the difficulty of determining network architectures and the solution using our DNA scheme. Section 3 gives a description of the design and development of the nuclear power plant fault diagnostic adviser. Section 4 presents the performance results of the adviser with both pure simulator data and data corrupted by 3% Gaussian noise. Section 5 contains concluding remarks.

2. NEURAL NETWORKS

Neural networks are a fast-emerging branch of the science of artificial intelligence. Hecht-Nielsen [35] defines a neural network as a "parallel, distributed information processing structure consisting of processing elements interconnected together with unidirectional signal channels called connections." The dense interconnection of these simple processing elements or nodes provides the ANN models with their characteristic properties. Lippmann [13] describes the ANN process as a simultaneous exploration of competing hypotheses utilizing massively parallel networks composed of individual processing elements connected by variable weights. This is opposed to a von Neumann computer that performs a program of sequential instructions. The layered feed-forward ANN, as shown in Figure 1.1, was used in this work. This model consists of nodes arranged in layers, with the nodes of any layer being connected to the nodes in an adjacent layer through variable weights. A node takes its input from every node of the layer below it, and feeds its output to every node of the layer above it. There are no connections between nodes of the same layer. In such networks, the first layer is the input layer where the nodes are inactive, their outputs being equal to their inputs. The last layer is the output layer. The layers in between consist of "hidden" nodes, so called because they are isolated from the outside environment. The design of a network architecture is rather arbitrary; only the number of nodes in the input and output layers are fixed by the problem at hand. The simplest node, shown in Figure 1.2, sums weighted inputs and passes the result through a squashing function, sometimes also called the transfer function. The output signal of a node can be of any mathematical type desired [14] but is usually nonlinier and analog and is governed by the transfer function. In this work, the transfer function was a sigmoid

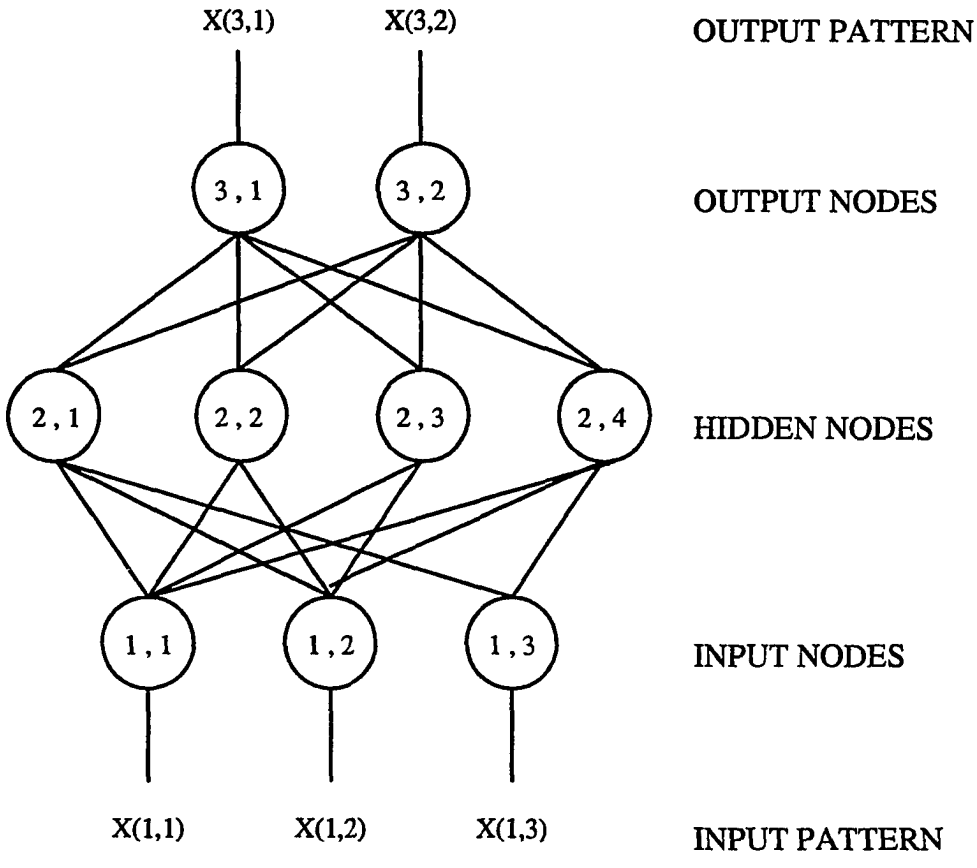

Figure 1.1: A three-layered feed-forward backpropagation neural network.

of the form

$$f(x) = \frac{1}{1 + e^{x}} \tag{1}$$

where x is the weighted sum of the outputs of the nodes in the previous layer, and $f(x)$ is the output. The behavior of this function can be seen in Figure 1.3.

The publication of the backpropagation technique by Rumelhart *et al.* [36] has been the most influential development in the field of neural networks in the past decade. This learning scheme involved the presentation of a set of input-output

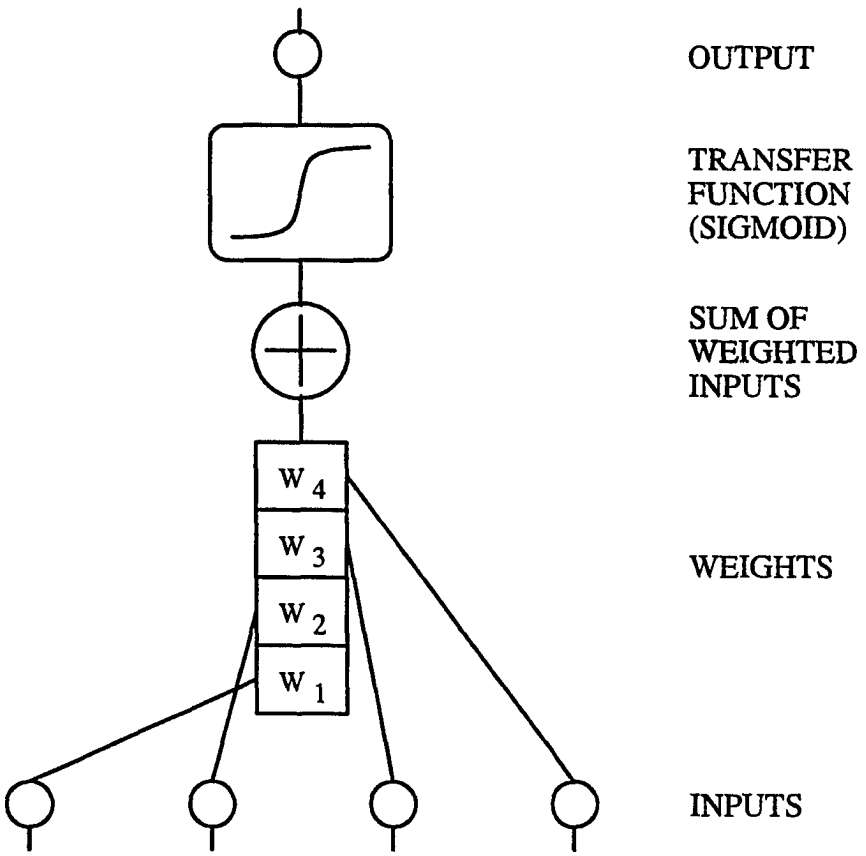

Figure 1.2: A simple node detailed.

patterns. The ANN uses the input vector to produce an output vector which is then compared with the expected or desired output vector to calculate the error for all the active nodes. The weights are changed using the delta and the generalized delta rules [35, 36, 20] such that the errors decrease during the next iteration. The error in the output nodes is easily calculated as the difference between the actual output and the desired output. The delta rule calculates the change in weights for the output layer. It is not possible to calculate the error in the hidden nodes in this way as their

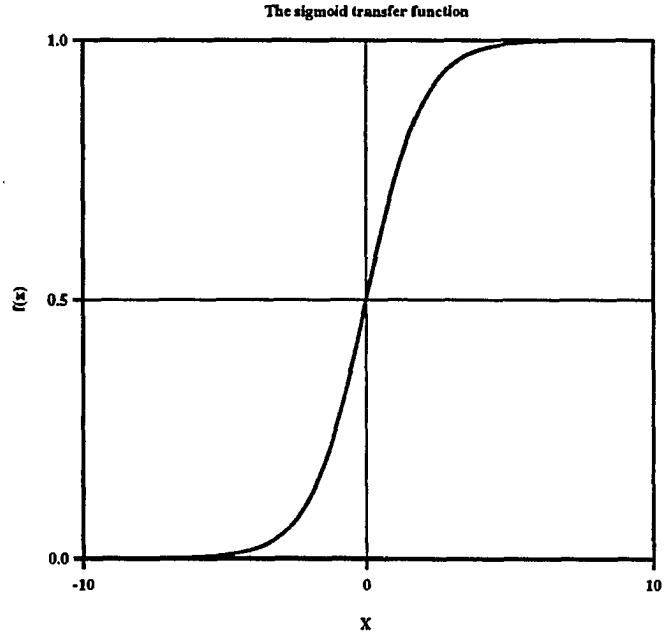

Figure 1.3: The sigmoid transfer function.

correct or expected outputs are not known. An error is assigned to each hidden node by backpropagating the error of the output nodes to the hidden nodes using the very same weights that were used to propagate the error to the output nodes in the first place [20]. The generalized delta rule, which is the delta rule modified for the hidden nodes, is used to change the weights in the hidden layers.

Most backpropagation networks, including those used in this work, are layered and feed-forward; they always consist of at least three layers of nodes. The number of inputs and outputs are fixed by the problem at hand. The only choice of network architecture is the number of hidden nodes. This choice needs to be exercised carefully. If the hidden layer is too large, it will encourage the network to memorize the input

patterns rather than generalize the input into features [35, 20]. The large number of nodes and weights give the network more ways to distinguish features allowing the specifics to be learned rather than the generalities. This reduces the network's ability to correctly classify novel patterns after training is complete. On the other hand, a hidden layer that is too small will drastically increase the number of iterations and thus the computer time required to train the network. In the extreme case, the network might be untrainable [20]. There are no hard and fast rules for determining the optimum architecture; most rules in use at this time are empirical in nature and are derived by heuristic methods [17, 37, 11].

The present work develops an ANN based diagnostic adviser for a nuclear power plant. This adviser uses 97 plant variables as inputs. The ANN has five binary outputs allowing for 2^5 possible output classes. As the design of the adviser progresses, the network needs to be trained a number of times with different numbers of patterns in the training set. If a search for the appropriate architecture is attempted before each training phase, the problem would assume mammoth proportions involving significant amounts of guesswork. It is therefore imperative to develop a systematic method that will derive optimum, or near optimum, architectures for a given problem. The DNA scheme described below is one such method.

2.1 Dynamic Node Architectures

Other researchers have attempted to develop various dynamic architecture schemes but have left some facets of the problem unanswered. For example, one approach prunes unnecessary weights from a fully connected and trained network [3, 21, 7]. This results in a sparsely connected network devoid of redundant weights. This

approach wastes computer time by training unnecessary nodes and weights in large networks. Moreover, some amount of guesswork is still involved in deciding the initial network size which needs to be more than the optimum architecture. Another approach [8, 24, 25] starts with small networks and builds them up until the network can successfully learn the training set. Most of these algorithms stop when the training set is first learned. But it is known that learning a functional relationship requires more nodes than recalling the same function [36, 27]. Work by Hirose *et al.* [24] addresses this issue by eliminating the last added node after the network can successfully learn the training set. The process of deleting nodes continues until the smallest network is found that can learn the problem. This process assumes that the last added node should be eliminated first. But Hirose *et al.* have no definite way to verify this assertion.

The derivative DNA scheme developed by Bartlett and Basu [14] progresses in a systematic method to arrive at the appropriate architecture for any particular problem by adding and deleting nodes as required and using a quantitative measure of the importance of each hidden node. Training is initiated with one hidden node. This architecture is obviously unable to arrive at a good solution unless the training task is exceedingly simple. As training progresses, the network soon reaches a learning plateau where it cannot reduce it's error below a certain value during a number of successive iterations. A node is then added to the hidden layer. The weights connecting this new node are assigned very small random values so that the addition of this node does not disrupt the network performance much. Training is resumed and the network soon reaches another plateau when another node is added. This process is continued until the network reaches a satisfactory level of performance. At this stage,

not all of the hidden nodes may be necessary to perform the mapping. Therefore, the hidden node with the least importance is removed from the network. The resultant network might require further training, depending on the deleted node's importance to the ANN function. Upon continued training, the smaller network might be able to learn the mapping. Then the next node with the least importance is deleted. This process is continued until the network is too small to learn the problem. Now nodes are added until the problem is relearned. The process of deleting and adding nodes is continued until the algorithm oscillates about some relative minimum architecture. Note that the final minimum arcitecture arrived at by the DNA scheme is influenced by the selection of various training parameters such as the learning rate, the initial random starting weights used in the initial architecture and the number of iterations spent on a learning plateau before a node is added.

2.2 Importance of a Node

The importance of a node to the ANN function can be defined in many ways. The skeletonization method [4, 5] measures the change in the error of the network output with the deletion of each node and assigns a greater importance to the node whose deletion causes a greater change. But if a hidden node is biasing the output of the network, i.e., it has a near constant output over the training set, this method will assign a very high importance to this node even though it is unimportant to the dynamics of the network. Another importance measure [30] is based on the values of the weights leading to and from the node and is binary in nature. Our importance measure, on the other hand, assumes that if changes in the output of a hidden node are more influential in deciding the output of the network than a similar change in

the output of another hidden node, then the former node is more important to the dynamic functioning of the network than the latter node. In this case, the importance of the jth hidden node with respect to the kth output node is defined as [14]

$$I_{(x_j|x_k)} = E[|\frac{\delta x_{k,n}}{\delta x_{j,n}}|] * dx_j^{max} \tag{2}$$

where $E[...]$ is the expectation over the entire training set and dx_j^{max} is the maximum change in the output of the jth hidden layer node over the entire training set. The above equation gives the partial importance of the jth hidden node with respect to the kth output node. If the network consists of more than one hidden layer, the partial importance of any of these hidden nodes with respect to any output node can be found by using the chain rule. The total importance of the jth hidden node is the sum of the partial importances of that node with respect to all the output nodes. Mathematically,

$$I_{(x_j)} = \sum_{k=1}^{kmax} I_{(x_j|x_k)} \tag{3}$$

where $kmax$ in the number of output layer nodes. The importance of a layer can be similarly defined as the sum of the importances of the nodes in that layer.

2.3 DNA Training: An Example

A very simple network learning task is the exclusive-nor problem. The training data for this example is shown in Table 1.1. The ANN training is initiated with a starting architecture of 2 x 1 x 1, i.e., two input, one hidden, and one output nodes. The training target was an RMS error of 0.01. Table 1.2 shows the training history for this problem. It is known from previous results [14] that while one hidden node is too small to solve this problem, two are sufficient. The DNA scheme adds a second

Table 1.1: Exclusive-nor training data

Pattern	Input1	Input2	Output
1	0.0	0.0	1.0
2	0.0	1.0	0.0
3	1.0	0.0	0.0
4	1.0	1.0	1.0

node when the network is unable to reduce its error during many iterations. This architecture reaches a plateau above the target RMS error, and a third node is added. This 2 x 3 x 1 architecture also is unable to learn the problem before a plateau is reached, and a fourth node is added. The 2 x 4 x 1 architecture is successful in learning the problem. Now, the node with the least importance is deleted. This leaves the network with three hidden nodes and is now able to reach the target RMS error. Another node is deleted and upon further training, the 2 x 2 x 1 network is also able to learn the problem to the desired level of accuracy. The further elimination of a node renders the network with only one hidden node. This is not sufficient to learn the problem and a node is added. This process is continued and the algorithm oscillates around the optimum architecture of 2 x 2 x 1 as can be seen in Table 1.2.

Computer simulations were carried out to compare the performance of DNA-generated networks and networks trained by conventional fixed node architecture (FNA) schemes. For the exclusive-nor problem, the DNA algorithm gave a minimum network of size 2 x 2 x 1. Two more networks with 5 and 10 hidden nodes are trained to the same level of accuracy on the exclusive-nor problem by using an FNA scheme. Then all three networks were used to recall on data corrupted by 5% uniform noise.

Table 1.2: DNA training history for the exclusive-nor problem

Architecture	RMS Error	Architecture	RMS Error	Architecture	RMS Error
2 x 1 x 1	0.50122	2 x 3 x 1	0.01793	2 x 4 x 1	0.02828
2 x 1 x 1	0.38725	2 x 4 x 1	0.01929	2 x 4 x 1	0.00997
2 x 2 x 1	0.39071	2 x 4 x 1	0.00993	2 x 3 x 1	0.02877
2 x 2 x 1	0.11059	2 x 3 x 1	0.02243	2 x 3 x 1	0.00982
2 x 3 x 1	0.11902	2 x 3 x 1	0.00972	2 x 2 x 1	0.06837
2 x 3 x 1	0.03376	2 x 2 x 1	0.04227	2 x 2 x 1	0.00999
2 x 4 x 1	0.04133	2 x 2 x 1	0.00999	2 x 1 x 1	0.49962
2 x 4 x 1	0.00997	2 x 1 x 1	0.49979	2 x 1 x 1	0.49224
2 x 3 x 1	0.02521	2 x 1 x 1	0.49286	2 x 2 x 1	0.42182
2 x 3 x 1	0.00981	2 x 2 x 1	0.32188	2 x 2 x 1	0.14522
2 x 2 x 1	0.06242	2 x 2 x 1	0.06229	2 x 3 x 1	0.18256
2 x 2 x 1	0.02109	2 x 3 x 1	0.06681	2 x 3 x 1	0.01644
2 x 3 x 1	0.03027	2 x 3 x 1	0.01483	2 x 4 x 1	0.02388

Table 1.3: Recall performance of ANNs derived by DNA and FNA schemes

Architecture	Scheme	Training RMS error	Recall RMS error
2x 2x1	DNA	9.9956E-03	0.00999
2x 5x1	FNA	9.9278E-03	0.01831
2x10x1	FNA	9.7582E-03	0.01983

The results in Table 1.3 show that the DNA architecture network outperformed the larger networks in the noisy situation. This indicates better generalization due to lesser number of nodes and weights.

3. DESIGNING THE NUCLEAR POWER PLANT ADVISER

This section describes the development of the ANN-based nuclear power plant fault diagnostic adviser. This adviser is capable of detecting and classifying a large

variety of transients and therefore needs to draw information from many plant variables. The choice of transients and variables is therefore very important. Two documents were consulted for the purpose: the *Updated Final Safety Analysis Report* [37] and the *Malfunction Cause and Effects Report* [38], both pertaining to the Duane Arnold Energy Center (DAEC). These documents describe many power plant transients. Discussions between personnel at DAEC and fellow researchers at Iowa State University (ISU) [39] resulted in a preliminary list of transients to be simulated on the operator training simulator and plant variables to be monitored [26]. Twenty seven of these transients, some of which can occur at varying severities, were selected for this work. The simulation of some transients at different severities enabled the adviser to detect a transient irrespective of its intensity. The training data therefore consisted of 43 scenarios representing the 27 distinct transients.

3.1 Data Collection and Processing

The raw data were obtained from the DAEC operators' training simulator. This data consisted of the numerical values of 97 plant variables at intervals of one second. The data also included a Yes/No binary switch that indicated the onset of the transient. The variables were selected from the complete list of 2,369 variables available on the simulator. These 97 variables were judged to be sufficient for an operator to diagnose the transients being investigated [39] and were therefore selected as the inputs for the proposed adviser. These variables covered a wide variety of plant instrumentation such as pressures and temperatures in the various systems of the plant, radiation monitors, and flow meters. Table 1.4 contains a complete list of variable used in this work. Raw data obtained from DAEC were reformatted and normalized

in the range zero to one. Normalization was based on the maximum and minimum possible values of the variables.

3.2 The Structure of the Adviser

In earlier approaches [3, 4, 36] one network was trained to output a particular binary code for the normal operating conditions and each transient investigated. The adviser presented here is a collection of two networks. The root network determines if the plant is in a normal condition or not, and the classifier network identifies the particular transient in progress. The values of the 97 chosen variables at any single instant of time are assumed to contain enough information to diagnose the plant status. This information proved to be adequate to solve the problem. Therefore, both networks have 97 input nodes. The root network has only one output node which gives an output of 0 if the plant is in a normal condition and 1 otherwise. The classifier network needs to be able to distinguish between 27 different transients. The ANN therefore has five output nodes. A listing of the transients investigated is shown in Table 1.5.

3.3 Training the Adviser

The training data for both the networks were chosen in an iterative manner. For the first trial of the root network, one pattern at the beginning and end of each of the 43 simulations were taken to form the training set. The initial training set therefore contained 86 patterns. Training was initiated with one hidden node. The target RMS error was 0.10. As the training progressed, the DNA scheme added more nodes and gave a final architecture with five hidde nodes. The network was used

Table 1.4: Plant variables used to train the ANN adviser.

Var. Num.	Variable Desig.	Description	Min. Value	Max. Value	Var. Unit
1	A041	Local power range monitor 16-25 flux level B	0.0	125.0	% power
2	A091	Source range monitor channel B	0.0	100.0	%
3	B000	Average power range monitor A Flux level	0.0	125.0	% power
4	B012	Reactor total core flow	0.0	60.0	Mlb/hr
5	B013	Reactor core pressure-differential	0.0	30.0	psid
6	B014	Control rod drive system flow	0.0	0.025	Mlb/hr
7	B015	Reactor feedwater loop A flow	0.0	4.0	Mlb/hr
8	B016	Reactor feedwater loop B flow	0.0	4.0	Mlb/hr
9	B017	Cleanup system flow	0.0	0.07691	Mlb/hr
10	B022	Total steam flow	0.0	8.0	Mlb/hr
11	B023	Cleanup system inlet temperature	0.0	755.0	$^{\circ}$F
12	B024	Cleanup system outlet temperature	0.0	600.0	$^{\circ}$F
13	B026	Recirculation loop A1 drive flow	0.0	15.1	Mlb/hr
14	B028	Recirculation loop B1 drive flow	0.0	15.1	Mlb/hr
15	B030	Reactor feedwater channel A1 temperature	280.0	430.0	$^{\circ}$F
16	B032	Reactor feedwater channel B1 temperature	280.0	430.0	$^{\circ}$F
17	B034	Recirculation loop A1 inlet temperature	260.0	580.0	$^{\circ}$F
18	B036	Recirculation loop B1 inlet temperature	260.0	580.0	$^{\circ}$F
19	B038	Recirculation A wide range temperature	50.4	789.6	$^{\circ}$F
20	B039	Recirculation B wide range temperature	50.4	789.6	$^{\circ}$F
21	B061	Reactor coolant total jet pumps 1-8 flow B	0.0	36.7	Mlb/hr
22	B062	Reactor coolant total jet pumps 9-16 flow A	0.0	36.7	Mlb/hr
23	B063	Reactor coolant total outlet steam flow A	0.0	2.0	Mlb/hr

Table 1.4 (Continued)

Var. Num.	Variable Desig.	Description	Min. Value	Max. Value	Var. Unit
24	B064	Reactor coolant total outlet steam flow B	0.0	2.0	Mlb/hr
25	B065	Reactor coolant total outlet steam flow C	0.0	2.0	Mlb/hr
26	B066	Reactor coolant total outlet steam flow D	0.0	2.0	Mlb/hr
27	B079	Reactor recirculation pump A motor vibration	0.0	10.0	MILS
28	B080	Reactor recirculation pump B motor vibration	0.0	10.0	MILS
29	B083	Control rod drive cooling-water differential pressure	0.0	500.0	dpsi
30	B084	Control rod drive cooling-water differential pressure	0.0	60.0	dpsi
31	B085	Torus air temperature #1	0.0	500.0	°F
32	B086	Torus air temperature #2	0.0	500.0	°F
33	B087	Torus air temperature #3	0.0	500.0	°F
34	B088	Torus air temperature #4	0.0	500.0	°F
35	B089	Drywell temperature azmuth 0 elevation 750	0.0	500.0	°F
36	B090	Drywell temperature azmuth 245 elevation 750	0.0	500.0	°F
37	B091	Drywell temperature azmuth 90 elevation 765	0.0	500.0	°F
38	B092	Drywell temperature azmuth 270 elevation 765	0.0	500.0	°F
39	B093	Drywell temperature azmuth 270 elevation 765	0.0	500.0	°F
40	B094	Drywell temperature azmuth 180 elevation 780	0.0	500.0	°F
41	B095	Drywell temperature azmuth 270 elevation 830	0.0	500.0	°F
42	B096	Drywell temperature center elevation 750	0.0	500.0	°F
43	B098	Torus water temperature	0.0	752.0	°F

Table 1.4 (Continued)

Var. Num.	Variable Desig.	Description	Min. Value	Max. Value	Var. Unit
44	B099	Torus water temperature	0.0	752.0	°F
45	B103	Drywell pressure	0.0	100.0	psia
46	B104	Torus pressure	0.0	100.0	psia
47	B105	Torus water level	-10.0	10.0	in.
48	B120	Torus radiation monitor A	-1.0	100.0	%
49	B121	Torus radiation monitor B	-1.0	100.0	%
50	B122	Reactor water level	158.0	218.0	in.
51	B124	Reactor water level	158.0	218.0	in.
52	B1257	Fuel zone level indication	-153.0	218.0	in.
53	B126	Reactor water level	158.0	458.0	in.
54	B127	Reactor vessel pressure	0.0	1200.0	psig
55	B128	Reactor vessel pressure	0.0	1200.0	psig
56	B129	Reactor vessel pressure	0.0	1500.0	psig
57	B130	Reactor vessel pressure	0.0	1500.0	psig
58	B137	Torus water level	1.5	16.0	ft
59	B138	Torus water level	1.5	16.0	ft
60	B150	Core spray A flow	-1767.8	5000.0	gpm
61	B151	Core spray B flow	-1767.8	5000.0	gpm
62	B160	Reactor core isolation cooling flow	-62.5	500.0	gpm
63	B161	High-pressure core injection flow	-437.5	3500.0	gpm
64	B162	Residual heat removal A flow	-75.0	15000.0	gpm
65	B163	Residual heat removal B flow	-75.0	150.0	gpm
66	B164	Drywell radiation monitor A	-1.0	100.0	%
67	B165	Drywell radiation monitor B	0.0	100.0	%
68	B166	Post-treat activity	0.0	100.0	%
69	B168	Pretreat activity	0.0	100.0	%
70	B171	Analyzer A — O_2 concentration	-1.25	10.0	%
71	B172	Analyzer A — H_2 concentration	-1.25	10.0	%
72	B173	Analyzer B — O_2 concentration	-1.25	10.0	%
73	B174	Analyzer B — H_2 concentration	-1.25	10.0	%
74	B180	Clean-up system flow	0.0	200.0	gpm
75	B196	Reactor water level-fuel zone A	-153.0	218.0	in.
76	B197	Reactor water level-fuel zone B	-153.0	218.0	in.

Table 1.4 (Continued)

Var. Num.	Variable Desig.	Description	Min. Value	Max. Value	Var. Unit
77	B247	Turbine steam bypass	0.0	500.0	°F
78	B248	Turbine steam bypass	0.0	500.0	°F
79	E000	4160 V switch gear bus 1A1 A-B	0.0	5.25	KV
80	F004	Condensate pump A&B discharge pressure	0.0	600.0	psig
81	F005	Low-pressure condenser circulating water inlet temperature A	0.0	200.0	°F
82	F010	High-pressure condenser circulating water outlet temperature A	0.0	200.0	°F
83	F011	Low-pressure condenser circulating water pressure differential A	0.0	10.0	dpsi
84	F015	Circulating water pump A&B discharge pressure	0.0	100.0	psig
85	F018	Cooling tower A discharge water temperature	0.0	752.0	°F
86	F019	Cooling tower B discharge water temperature	0.0	752.0	°F
87	F040	1P-1A reactor feed pump suction pressure	0.0	600.0	psig
88	F041	1P-1B reactor feed pump suction pressure	0.0	600.0	psig
89	F042	1P-1A reactor feed pump discharge pressure	0.0	2000.0	psig
90	F043	1P-1B reactor feed pump discharge	0.0	2000.0	psig
91	F044	Condensate total flow	0.0	8.0	Mlb/hr
92	F045	Condensate makeup flow	-10.0	100.0	Klb/H
93	F046	Condensate rejection flow	0.0	50.0	Klb/H
94	F094	Feedwater final pressure	0.0	2000.0	psig
95	G001	Generator gross watts	0.0	720.0	MWE
96	T039	Low-pressure condenser pressure	0.0	30.0	in.-Hg
97	T040	High-pressure condenser pressure	0.0	30.0	in.-Hg

Table 1.5: The twenty-seven transients, and the fourty-three scenarios used to design the adviser.

No.	Scenario	Description
1	cu10	Reactor water clean-up coolant leakage
2	cu10gp5	Reactor water clean-up coolant leakage with failure of Group 5 isolation valves.
3	fw04a	Condensate filter demineralizer resin injection
4	fw09a	Reactor feedwater pump trip
5	fw12c0	Feedwater regulator valve controller stuck closed
6	fw12c1	Feedwater regulator valve controller stuck open
7	fw17a	Main feedwater line break inside primary containment
8		Main feedwater line break outside primary containment
	fw18a	- 100% severity
	fw18a_2	- 60% severity
	fw18a_3	- 30% severity
9		High-pressure core injection (HPCI) steam supply line break in HPCI room
	hp05	- 100% severity
	hp05_2	- 60% severity
	hp05_3	- 30% severity
10		High-pressure core injection steam supply line break in torus room
	hp08	- 100% severity
	hp08_2	- 60% severity
	hp08_3	- 30% severity
11	ic20scr2	Spurious scram with effective operator action to avoid feedwater pump trip. Initial condition IC20 : 100% power, End of Cycle
12	ic20scrm	Spurious scram with no operator action. Initial condition IC20 : 100% power, End of Cycle.
13	ic22scra	Spurious scram. Initial condition IC22: 25% power, Beginning of Cycle
14	ic23scrm	Spurious scram. Initial condition IC23 : 75% power, Beginning of Cycle
15	ic24scrm	Spurious scram. Initial condition IC24 : 100% power, Middle of Cycle

Table 1.5 (Continued)

No.	Scenario	Description
16		Steam leak inside primary containment
	ms02	- 100% severity
	ms02_2	- 60% severity
	ms02_3	- 30% severity
17		Main steam line rupture inside primary containment
	ms03a	- 100% severity
	ms03a_2	- 60% severity
	ms03a_3	- 30% severity
18		Main steam line rupture outside primary containment
	ms04a	- 100% severity
	ms04a_2	- 60% severity
	ms04a_3	- 30% severity
19	ms19ab	Spurious group 1 isolation
20	ms32	Spurious group 7 isolation
21	rp05tc01	Reactor protection system SCRAM circuit failure (ATWS) with alternate rod injection
22	rp5actc1	Reactor protection system SCRAM circuit failure (ATWS) with failure of alternate rod injection
23	rr10	Recirculation pump speed feedback signal failure
24		Recirculation loop rupture (design basis Loss of Coolant Accident)
	rr15a	- 100% severity
	rr15a_2	- 60% severity
	rr15a_3	- 30% severity
25		Coolant leakage inside primary containment
	rr30	- 100% severity
	rr30_2	- 60% severity
	rr30_3	- 30% severity
26	rx01	Fuel cladding (5%) failure
27	tc02	Electrical hydraulic control (EHC) system hydraulic pump failure

to recall all the patterns over the entire length of the 43 simulations. The network, as expected, did not do a very good job of classifying all the patterns. The patterns with the worst recall errors were added to the training set and the network from the previous trial was trained further. This process of adding patterns to the training set was continued until the network could detect the onset of all the transients within a reasonable amount of time. A similar approach was used to train the classifier network. In this case, the normal operating conditions were not used because this network would be used to classify only those patterns that the root network had detected as abnormal.

This network training problem is slightly different from the conventional problems solved using neural networks. In most cases, a training set is given and is used to train a network. The recall set is not known beforehand, and the trained network is used to recall on unseen patterns. But in our case, the recall set is known from the simulations. We try to define the training set through the various trials so that it is a fairly accurate representation of the recall set. Consequently, good generalization is extremely important to keep the number of patterns in the training set to a minimum as well as to make the adviser fault and noise tolerant.

The final architecture of the root network was 97 x 9 x 1 and the classifier network was 97 x 26 x 5. The advantages of the DNA algorithm can be appreciated here. Suppose a conventional fixed architecture scheme was used to solve this problem. In that case we need to guess a few architectures for the first trial, train all of them, and choose the architecture with the best performance for later trials. But the training set changes, in fact increases, with each trial. Consequently, there is no guarantee that the best architecture for a given trial will also be good enough for the following trials.

Thus a fixed node architecture scheme would have been extremely inconvenient for developing the nuclear power plant status diagnostic adviser.

4. RESULTS

A total of 43 different scenarios representing 27 distinct transients were used for this study. The fault diagnostic adviser was trained on pure non-noisy simulator data, and its performance was tested on both pure data and data corrupted by 3% Gaussian noise. The computer generated noisy data contained Gaussian noise at a 3% standard deviation of the actual data. Table 1.6 gives a summary of the diagnostic performance of the adviser. The root network was able to detect all the transients irrespective of noise within 40 seconds. Table 1.6 shows that a vast majority of these were detected within 20 seconds. The root network needed only 192 patterns to be trained to this level of performance, which is less than 2.5% of the total of 7,854 patterns. Similarly, the classifier network was able to identify all the transients within 2 minutes of the initiating events. This network required 464 patterns, which is less than 6% of the total patterns.

The results can be best understood by looking at certain examples from Table 1.6. Transient scenario "fw12c1" (transient number 6), assigned the output binary code 10110, is the failure of the feedwater regulator valve controller which leaves the valve stuck open. This caused the plant to undergo a safety trip 90 seconds after the initiating event. The root network was able to detect an abnormality in the plant status immediately after the initiating event. This ability was not hampered even when the input data was corrupted by noise. The classifier network diagnosed the transient after 104 seconds after its advent. Under noisy condition, successful

Table 1.6: The output binaries, time to safety trip, detection time (by the root network) and the classification time (by the classifier network) for each scenario. The diagnosis and classification times are since the initiation of the transient. Fourty-three scenarios representing twenty-seven transients were used to design the adviser. The performance results are for both pure and noisy data.

Trans. Num.	Scenario Code	Output Binaries	Safety Trip Time (sec)	Root Diagnosis Time (sec)		Classification Time (sec)	
				Pure	Noise	Pure	Noise
1	cu10	1 1 0 0 0	n/s	5	5	62	62
2	cu10gp5	1 1 0 0 1	44	0	0	104	112
3	fw04a	1 1 1 0 1	n/s	25	27	40	52
4	fw09a	1 1 1 0 0	n/s	17	16	53	55
5	fw12c0	0 1 1 0 0	10	7	7	38	41
6	fw12c1	1 0 1 0 0	90	0	0	104	109
7	fw17a	0 1 0 0 1	1	0	1	74	86
8	fw18a	1 0 0 0 1	8	5	5	57	55
	fw18a_2		24	8	12	58	62
	fw18a_3		n/s	26	33	93	101
9	hp05	0 1 1 0 1	n/s	14	14	67	63
	hp05_2		n/s	22	18	2	18
	hp05_3		n/s	30	32	2	18
10	hp08	1 0 1 0 1	n/s	14	15	71	77
	hp08_2		n/s	34	32	96	108
	hp08_3		n/s	38	34	115	112
11	ic20scr2	0 0 1 0 0	1	2	5	93	93
12	ic20scrm	0 1 0 0 0	1	2	2	116	118
13	ic22scra	1 0 0 0 0	1	9	12	49	51
14	ic23scrm	0 0 0 0 0	1	2	2	81	75
15	ic24scrm	1 0 0 1 0	1	10	8	64	62
16	ms02	0 1 1 1 1	3	4	4	46	46
	ms02_2		3	4	4	56	59
	ms02_3		4	5	5	61	60
17	ms03a	1 0 1 1 1	2	0	0	85	101
	ms03a_2		1	0	0	38	38
	ms03a_3		2	2	2	71	73

Table 1.6 (Continued)

Trans. Num.	Scenario Code	Output Binaries	Safety Trip Time (sec)	Root Diagnosis Time (sec)		Classification Time (sec)	
				Pure	Noise	Pure	Noise
18	ms04a	1 0 0 1 1	2	0	1	69	68
	ms04a_2		2	0	1	68	68
	ms04a_3		2	0	3	69	68
19	ms19ab	1 0 1 1 0	17	20	26	76	91
20	ms32	0 0 0 1 1	98	9	12	105	112
21	rp05tc01	0 0 1 0 1	20	5	5	63	62
22	rp5actc1	0 0 1 1 0	n/s	5	5	54	66
23	rr10	0 1 0 1 0	n/s	17	26	66	71
24	rr15a	1 1 0 1 0	1	0	2	23	26
	rr15a_2		1	0	2	81	78
	rr15a_3		1	2	2	21	31
25	rr30	1 1 0 1 1	18	7	9	101	96
	rr30_2		30	21	24	83	88
	rr30_3		59	29	27	112	109
26	rx01	1 1 1 1 0	n/s	11	18	18	23
27	tc02	0 0 1 1 1	5	6	5	70	83

n/s = no safety trip

diagnosis was achieved at 109 seconds.

There are many transients that occur over a prolonged period of time and do not cause a safety trip. An example is transient "rx01" (transient number 26). This transient is caused by a 5% failure of the fuel cladding, resulting in radioactive contamination of the coolant system. The root network detects an abnormality 11 seconds after the initiating event, and the classifier network diagnoses the transient at 18 seconds. During this time the operators will be undertaking tasks mandated by the plant operating procedures. A definite diagnosis by the adviser in such a situation would help the shift technical adviser assess the situation and judge the validity and applicability of the procedures being performed. Altough the procedures are quite extensive, it is conceivable that the plant might enter a situation not foreseen. If the shift technical adviser feels that information provided by the diagnostic adviser conflicts with the procedures being carried out, he may take appropriate action.

Some of the transients analyzed here cause immediate safety trips of the reactor. For example transient "ms04a" (transient number 18) is caused by the rupture of the main steam line outside the primary containment. We analyzed three scenarios for this transient corresponding to 100% (ms04a), 60% (ms04a_2), and 30% (mso4a_3) severity of the rupture. All these transients were assigned the same binary output code 10011 since the adviser should classify the transient irrespective of severity. The plant underwent a safety trip within two seconds of the initiating event for each of the three severities. The root network was able to detect an anomaly in the plant status almost immediately. But the short time before the automatic safety trip will not give the operators a chance to take any action before the trip. In this case the diagnosis can later provide valuable clues during technical reviews of the transient.

5. CONCLUSIONS

The first conclusion from this work is the ability of neural networks to detect a very wide variety of operational transients at a boiling water reactor (BWR) nuclear power plant. An ANN-based adviser was successfully trained to detect and classify 27 distinct transients and the four normal conditions. The adviser performed well even when the data was corrupted by noise. Two neural networks functioning in a hierarchical manner were used in this adviser. Further evolution of the hierarchical approach could produce advisers capable of drawing information from more plant instrumentation and diagnosing a wider range of transients.

The second conclusion is the use of DNA allowed the speedy development of the adviser by eliminating a large amount of guesswork that would otherwise have been required to determine appropriate architectures. Use of a fixed architecture scheme would have been a major hurdle because the training set increased as the development of the adviser progressed. The DNA scheme is based on a derivative importance function and can arrive at a relative optimum network architecture for any problem.

The third significant conclusion from this work is the feasibility of a modular approach to solving a problem using a collection of individual networks. This was demonstrated by the successful use of two networks to solve two parts of the diagnostic problem. The practical acceptance of this technology would depend largely on the ability to develop a modular design, making it possible to increase the diagnostic capability of the adviser using the existing system. It is conceivable that not all of the 97 variables are needed to identify each transient. A simplification of the system design incorporating a number of small ANNs will make it possible to identify the

important variables for the task of transient identification.

ACKNOWLEDGMENTS

This work was made possible by the gracious support of the United States Department of Energy under Special Research Grant No. DEFG02-92ER75700, entitled "Neural Network Recognition of Nuclear Power Plant Transients," and the Iowa Electric Light and Power Company who provided the use of the simulator at the Duane Arnold Energy Center. Their support does not however constitute an endorsement of the views expressed in this paper.

BIBLIOGRAPHY

[1] E. B. BARTLETT and R. E. UHRIG, "Nuclear Power Plant Status Diagnostics Using Artificial Neural Networks," *Nucl. Technol.*, **97**, 272 (March 1992).

[2] J. S. JUDD, *Neural Network Design and the Complexity of Learning*, The MIT Press, Cambridge, Mass. (1990).

[3] H. S. WILF, *Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, New Jersey. (1986).

[4] E. B. BARTLETT and A. BASU, "A Dynamic Node Architecture Scheme for Backpropagation Neural Networks," *Intelligent Engineering Systems Through Artificial Neural Networks*, p. 101, C. H. DAGLI, S. R. T. KUMARA, and Y. C. SHIN, Eds., ASME Press, New York (1991).

[5] F. J. L. BINDON, "The Role of the Power Station Operator," *Proc. International Conference on Training for Nuclear Power Plant Operation*, Bristol, England, May 20-21, 1982, p.4:05.

[6] D. E. SEBO, M. A. BRAY, and M. A. KING, "Reactor Safety Assessment System," *Artificial Intelligence and Other Innovative Computer Applications in the Nuclear Industry*, p. 221, M. C. MAJUMDAR, D. MAJUMDAR, and J. I. SACKETT, Eds., Plenum Press, New York (1988).

[7] B. NASSERSHARIF, J. PATHAK, and A. KINI, "A Rule Based Approach to Model-Based Fault Diagnostic Expert Systems," *Proc. American Nuclear Society Meeting on Frontiers in Innovative Computing for the Nuclear Industry*, Jackson Lake, Wyoming, September 15-18, 1991, p. 35.

[8] C. R. HARDY, J. HA, B. K. HAJEK, and D. W. MILLER, "A Real-Time Expert System for the Detection and Diagnosis of Abnormal Conditions in Nuclear Power Plants," *Proc. American Nuclear Society Meeting on Frontiers in Innovative Computing for the Nuclear Industry*, Jackson Lake, Wyoming, September 15-18, 1991, p. 319.

[9] T. A. BALLARD and G. A. CORDES, "Adapting a Reactor Safety Assessment System for Specific Plants," *Proc. American Nuclear Society Meeting on Frontiers in Innovative Computing for the Nuclear Industry*, Jackson Lake, Wyoming, September 15-18, 1991, p. 526.

[10] A. BARR and E. A. FEIGENBAUM, Eds., *The Handbook of Artificial Intelligence*,vol. 1, Pitman, London, (1981).

[11] J. LIEBOWITZ and D. A. DE SALVO, Eds., *Structuring Expert Systems - Domain, Design, and Development*, Yourdon Press, Englewood Cliffs, NJ. (1989).

[12] R. HECHT-NIELSEN, "Theory of the Backpropagation Neural Network," *Proc. International Joint Conference on Neural Networks*, Vol. 1, Washington, D.C., June 18-22, 1989, p. 593.

[13] R. P. LIPPMANN, "An Introduction to Computing with Neural Nets," *IEEE Acoustics Speech and Signal Processing Magazine*, 4, 4 (April 1987).

[14] M. CAUDILL, "Neural Networks Primer, Part 1," *AI Expert*, 46 (December 1987).

[15] E. D. RUMELHART, G. E. HINTON, and R. J. WILLIAMS, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, p. 318, The MIT Press, Cambridge, Mass. (1986).

[16] M. CAUDILL, "Neural Networks Primer, Part 3," *AI Expert*, 53 (June 1988).

[17] N. E. COTTER, "The Stone-Weierstrass Theorem and its Application to Neural Networks," *IEEE Transaction on Neural Networks* **1**, 290 (1990).

[18] R. HECHT-NIELSEN, "Kolmogorov's Mapping Neural Network Existence Theorem," *Proc. IEEE International Conference on Neural Networks*, Vol. 2, San Diego, California, June 21-24, 1987, p. 11.

[19] B. R. UPADHYAYA and E. ERYUREK, "Application of Neural Networks for Sensor Validation and Plant Monitoring," *Nucl. Technol.*, **97**, 170 (February 1992).

[20] E. D. KARNIN, "A Simple Procedure for Pruning Back-Propagation Trained Neural Network," *IEEE Transactions on Neural Networks*, **1.2**, 239 (June 1990).

[21] A. S. WEIGEND, D. E. RUMELHART, and B. A. HUBERMAN, "Generalization by Weight Elimination Applied to Currency Exchange Rate Prediction," *Proc. International Joint Conference on Neural Networks*, Vol. 1, Seattle, Washington, July 8-12, 1991, p. 837.

[22] Y. WON and R. PIMMEL, "A Comparison of Connection Pruning Algorithms with Back-Propagation." *Intelligent Engineering Systems Through Artificial Neural Networks*, p. 113, C. H. DAGLI, S. R. T. KUMARA, and Y. C. SHIN, Eds., ASME Press, New York (1991).

[23] T. ASH, "Dynamic Node Creation in Backpropagation Networks," *Proc. International Joint Conference on Neural Networks*, Vol. 2, Washington, D.C., June 18-22, 1989, p. 623.

[24] Y. HIROSE, K. YAMASHITA, and S. HIJIYA, "Back Propagation Algorithm That Varies the Number of Hidden Nodes," *Neural Networks*, **4**, 61 (1991).

[25] J. VAARIO and S. OHSUGA, "Adaptive Neural Architectures Through Growth Control," *Intelligent Engineering Systems Through Artificial Neural Networks*, p. 11, C. H. DAGLI, S. R. T. KUMARA, and Y. C. SHIN, Eds., ASME Press, New York (1991).

[26] E. B. BARTLETT, "Dynamic Node Architecture Learning: An Information Theoretic Approach," *Neural Networks*, In press.

[27] E. B. BARTLETT, "A Dynamic Node Architecture Scheme for Layered Neural Networks," *Journal of Artificial Neural networks*, In press.

[28] M. C. MOZER and P. SMOLENSKY, "Skeletonization: A Technique for Trimming the Fat from a Neural Network via Relevance Assessment," *Advances in Neural Information Processing Systems*, Vol. 1, D. TOURETZKY, Ed., (1989).

[29] V. EIGEL-DANIELSON and M. F. AUGUSTEIJN, "Neural Network Pruning and its Effect on Generalization, Some Experimental Results," *Neural Parallel & Scientific Computations*, **1**, 59 (1993).

[30] S. V. ODRI, D. P. PETROVACKI, and G. A. KRSTONOSIC, "Evolutional Development of a Multilevel Neural Network," *Neural Networks*, **6**, 583 (1993).

[31] "Chapter 15: Accident Analysis" in *Updated Final Safety Analysis Report*, Vol. XI, Duane Arnold Energy Center, Iowa Electric Light and Power Company, Palo, Iowa (1984).

[32] J. GOULD, *Malfunction Cause and Effects Report*, Task no. 06000004, Duane Arnold Energy Center, Iowa Electric Light and Power Company, Palo, Iowa (1991).

[33] D. VEST, C. HUNT, and D. BERCHENBRITER, Personal discussions and correspondence with Duane Arnold Energy Center simulator complex employees, Iowa Electric Light and Power Company, Palo, Iowa (1991-1992).

[34] T. L. LANC, "The Importance of Input Variables to a Neural Network Fault-Diagnostic System for Nuclear Power Plants," MS Thesis, Iowa State University (1991).

[35] E. B. BARTLETT and R. E. UHRIG, "Nuclear Power Plant Status Diagnostics Using Artificial Neural Networks," *Proc. American Nuclear Society Meeting on Frontiers in Innovative Computing for the Nuclear Industry*, Jackson Lake, Wyoming, September 15-18, 1991, p. 644.

[36] A. BASU, "Nuclear power plant status diagnostics using a neural network with dynamic node architecture," MS Thesis, Iowa State University (1992).

CHAPTER 2. A DYNAMIC INPUTS SELECTION SCHEME FOR ARTIFICIAL NEURAL NETWORKS

A paper submitted to IEEE Transactions on Neural Networks

Anujit Basu and Eric B. Bartlett

ABSTRACT

This paper presents a dynamic input selection (DIS) scheme that enables an artificial neural network (ANN) to select appropriate input variables during the training process. The scheme also enables the ANN to add and subtract hidden nodes as required. DIS completely eliminates the need to pre-select network architectures including the number of inputs. A statistical analysis of the training data patterns involving principal component analysis (PCA) and information theory (IT) is used to rank the input variables in order of their relative importance to the desired mapping. The network begins training with the higherst ranked input and one hidden node, and adds inputs and hidden nodes until the mapping is learned. Any hidden or input nodes that are unnecessary once the network performs satisfactorily are subsequently removed from the ANN. In this work, DIS is demonstrated on two simple benchmark problems and a more complex temperature prediction problem. Results show DIS to be capable of developing viable ANN models without needless trial and error with

respect to input selection.

1. INTRODUCTION

Artificial neural networks (ANNs) learn to map functions from example data, and hopefully generalize this information to novel data [19]. An ANN's ability to generalize is highly dependent on its architecture and input vector [20]. Although no one else, to our knowledge, has investigated input vector determination, many researchers have investigated methods to arrive at the optimum number of nodes in a hidden layer. Most of these approaches involve node or weight elimination [3, 4, 5, 6, 7]. These methods train a network that is larger than optimal, thus wasting computing time training unnecessary nodes and weights. They then attempt to eliminate these same unnecessary elements from the trained ANN. Other researchers have approached the problem of hidden node architecture optimization by adding nodes during the training process [8, 35, 10]. These methods usually terminate when the problem is first learned, possibly leaving the network with extra hidden nodes. In one such case, where node elimination is attempted after learning, the nodes are eliminated in the reverse order in which they are added [35]. Thought it makes intuitive sense, no quantitative evidence is presented to prove that the last added node has the least contribution to the ANN's performance. Still others use heuristic rules to preselect the optimal hidden architecture and do not vary the number of hidden nodes during training. However, mathematical justification for such rules are not completely rigorous [11]. Previous work by one of the authors [21] is the only instance to our knowledge where an attempt was made to have the network determine the size of the input layer during training. That work however had considerable difficulty

taking into account the information presented jointly by two or more input variables. The authors of the work described here have previously demonstrated a systematic method called dynamic node architecture (DNA) that arrives at the appropriate number of nodes in the hidden layer of a backpropagation network [1, 14]. The work presented here improves on the authors' previous DNA work and extends the DNA approach to the input layer. The method presented in this paper successfully accounts for joint information among input variables.

Selecting the inputs for a well known problem is a simple exercise. However, for most real world applications, input selection can be extremely complicated. Most ANN users currently rely on their expertise for selecting the required input variables for specific problems. However, it is desirable to develop a general method that does not rely on user expertise. Such a general method would make ANNs more useful by reducing the complexity of their application. This is because neural network training is a non-polynomial time complete problem [16, 17] and smaller networks are therefore much less difficult to train. Our dynamic input selection (DIS) scheme was developed to eliminate the guesswork associated with selecting appropriate ANN inputs. DIS proceeds systematically by building up the input and hidden layers during the training process, thus arriving at a viable model without any guesswork.

There are many methods used by statisticians to estimate the number and appropriateness of the variables that could be used to develop a model [23, 24, 25]. Testing the linear correlation of each input variable with respect to the output(s) over the available data would be an easy method for selecting input variables [26]. Unfortunately, such techniques account for only the linear relationship between the input and output variables [25, 27]. Measures of nonlinear correlation such as infor-

mation theory might also be used [28, 29, 30]. But, as we will show in Section 3.1, an information theoretic interdependency analysis (ITIA) of individual inputs x and y with respect to the output z might not provide an appropriate measure of the importance of the inputs. For example, if $z = g(x, y)$ but $z \neq h(x)$ and $z \neq f(y)$, then ITIA on the individual variables x and y with respect to the output z would indicate that the two variables do not have much importance to determining the output. But we know that the two inputs together define the output, and this is not reflected in the interdependency analysis of individual inputs. The exclusive-or problem is a simple example of such a dilema. It cannot be guaranteed that such situations will not occur in real world problems. This is where our previous work falls short [21]. Accounting for all such joint information using ITIA would lead to a combinatorial explosion of calculations as ITIA has to be performed for all possible combinations of input variables. Such an exhaustive analysis will be intractable for large models. Therefore, before an information theoretic analysis can be performed on the training data, it is important to assure that the variables being subjected to ITIA contain as little joint information as possible.

To solve this problem, we first perform a principal component analysis (PCA) on the input data to transform the n-dimensioned input data vector into an n-dimensioned vector of uncorrelated and mutually orthogonal principal components (PC's) [31, 32, 33]. We shall show in Section 3.1 that ITIA can be performed on the principal components without loss of accountability, and a measure of nonlinear correlation between the PC's and the outputs can be determined. Neural network researchers have been attracted to PCA for its ability to extract linear features from the data set [28]. Many researchers have implemented algorithms that perform PCA

like operations [29, 30, 31, 32]. However, for our analysis, the PC's are numerically computed by determining the eigenvalues and the corresponding eigenvectors of the data covariance matrix as described in Section 2.1. We have also developed and will describe a method of backtracking from the principal component space to the original input space to arrive at a relative measure of the importance of each input with respect to the output vector.

Once the input variables are ranked, the training process can begin with just the most important input and one hidden node. The DIS scheme will then add inputs and hidden nodes as needed until the ANN has learned the training data. Our experience has shown that ANNs typically require more hidden nodes to learn a problem than to recall it [1, 14]. This same behavior can be exhibited in the input layer. Therefore, the DIS scheme attempts to prune out any input or hidden nodes from the trained network that are not required. It is possible that the trained ANN disagrees with our original input variable ranking. This is because our original analysis is based on data and not on the characteristics of the ANN. The input variables are therefore pruned not on the basis of the original ranking of the variables, but on a new ranking involving the inputs and the outputs of the ANN.

In the next section we present the required concepts of principal component analysis and information theory applied to the DIS scheme. Section 3 presents the DIS scheme and its three constituent phases. Section 4 presents the results from various computer simulations which benchmark the scheme. Conclusion are presented in the final section.

2. STATISTICAL CONCEPTS

2.1 Principal Component Analysis

Principal component analysis has traditionally been used as a method of reducing the dimensionality of a data set that contains a large number of interrelated variables [31, 32, 33]. In the process, PCA preserves as much of the variation in the data set as possible. PCA has become an essential technique in data compression and linear feature extraction [30]. Our interest in PCA, however, is limited to the transformation of input variables into principal components that are mutually orthogonal.

Let $\mathbf{x}$ be an n-dimensioned input vector that has been transformed such that each variable has zero mean. The first PC is the linear combination $\alpha_1^T \mathbf{x}$ which has the maximum possible variance. Here, α_1 is a vector of n constants, $\{\alpha_{1,1}, \alpha_{1,2}, \ldots, \alpha_{1,n}\}$, so that

$$\alpha_1^T \mathbf{x} = \alpha_{1,1} x_1 + \alpha_{1,2} x_2 + \ldots + \alpha_{1,n} x_n = \sum_{j=1}^{n} \alpha_{1,j} x_j \quad . \tag{1}$$

The second PC is the linear combination $\alpha_2^T \mathbf{x}$ such that this second PC is uncorrelated to the first PC and has the maximum possible variance. Similarly we determine the remaining PCs such that they are all uncorrelated to the principal components before them and have the maximum possible variance.

Let the data covariance matrix $\mathbf{C}$ be given by

$$\mathbf{C} = E[\mathbf{x}\mathbf{x}^T] \tag{2}$$

where $E[\bullet]$ is the expectation value. Note that the covariance matrix is given by $E[\mathbf{x}\mathbf{x}^\mathbf{T}]$ and not by $E[\mathbf{x}\mathbf{x}^\mathbf{T}] - E^2[\mathbf{x}]$ because the data vector $\mathbf{x}$ had been modified to have a zero mean and therefore $E[\mathbf{x}] = 0$.

The vectors $\alpha_1, \alpha_2, \ldots, \alpha_n$ that define the PC's are the n eigenvectors of the data covariance matrix $\mathbf{C}$. These eigenvectors are the orthogonal unit vectors given by

$$\mathbf{C}\alpha_i = \lambda_i \alpha_i , \qquad i = 1, ..., n \tag{3}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the n eigenvalues of matrix $\mathbf{C}$ in descending order of magnitude. Note that while $\alpha_1^T \mathbf{x}$ is called the first principal component, $\alpha_n^T \mathbf{x}$ is called the first minor component.

Statistically, PCA concentrates on variances, though it does not ignore covariances and correlations [31, 32]. The variance in the input variables is redistributed among the principal components such that the sum of the variances of the n input variables is equal to the sum of the variances of the n principal components. The traditional method of reducing dimensionality using PCA is to use only those PCs that account for a certain amount of the total variance in the original input data. Typically, the selected principal components account for 70% to 80% of the total variance [31].

2.2 Information Theory

Let us assume for the moment that we are dealing with discrete data. Then, a variable x would take on any one of N discrete values $\{x_1, x_2, ..., x_N\}$. Let p_i be defined as the probability of occurrence of any particular value x_i.

$$p_i = p(x = x_i) \quad . \tag{4}$$

This probability can be estimated for any x_i as the number of occurrences of the value x_i in a data set divided by the total number of patterns in the data set. The

entropy, or information, presented by the variable x can be defined as [29, 30, 34]

$$H(x) = -\sum_{i=1}^{N} p_i \ln(p_i) \quad . \tag{5}$$

Entropy is always a positive number since the probabilities are less than or equal to one, and therefore the logarithmic terms are negative or zero. The entropy of the variable x is zero when x takes only one value over the entire data set, in which case $N = 1$ and $p_1 = 1$. However, if x takes more than one value over a data set, then the entropy is maximized when the probability of occurrence of each of the N values is the same. In this case, $p_i = (1/N)$ for $i = 1, ..., N$, and $H(x) = -\ln(1/N)$.

We can extend Equation 5 to define the information presented by two variables x and y as

$$H(x,y) = -\sum_{i,j} p_{ij} \ln(p_{ij}) \tag{6}$$

where the probability p_{ij} is given by

$$p_{ij} = p(x{=}x_i \text{ and } y{=}y_j) \quad . \tag{7}$$

The entropy of y given x is a measure the information in the variable y when the value of x is known. Mathematically,

$$H(y|x) = -\sum_{i,j} p_{ij} \ln \frac{p_{ij}}{p_i} \tag{8}$$

where the subscripts i and j refer to the variables x and y respectively. It can also be shown that [34]

$$H(y|x) = H(x,y) - H(x) \quad . \tag{9}$$

Based on the above, we can now define an asymmetric measure of association between two variables x and y, denoted by $U(y|x)$ [29]:

$$U(y|x) = \frac{H(y) - H(y|x)}{H(y)} \quad . \tag{10}$$

This measure explains the extent to which y's entropy $H(y)$ is lost if x is already known. $U(y|x)$ will always lie between zero and one. A value of 0 indicates $H(y) = H(y|x)$, which means that knowledge of x does not have any effect on the entropy of y. In that case, x and y have no functional relationship. On the other hand, a value of 1 indicates that $H(y|x) = 0$, which means that knowing x eliminates the entropy of y. Thus y is completely predicted by x [28, 29, 34].

The above discussion assumed that the variables took discrete values, and thus it was possible to calculate the probabilities of each variable taking any particular value. But variables in real world are continuous rather than discrete. In such cases, the range of each variable can be broken into bins, and each data pattern can be converted such that the value of each variable is replaced by the bin number in which it falls [36]. Information theoretic interdependency analysis can then be accomplished, where p_i refers to the probability of the value of x falling in the ith bin.

3. DYNAMIC INPUT SELECTION

The ANNs used in this work are three layered feed-forward neural networks trained using the backpropagation learning algorithm [20, 35, 36]. (See Figure 2.1.) The input layer nodes are inactive, their output being equal to their input. The hidden and output layers, however, contain active nodes. The activation function used in this work is the familiar sigmoid function [20] given by

$$f(x) = \frac{1}{1 + e^{-x}} \quad . \tag{11}$$

The size of the output layer is fixed by the problem at hand since the output vector is usually well defined. The objective of the DIS scheme is to select the appropriate

input variables from a given set during training, thus determining the size of the input layer. Our training method also includes DNA from previous work which varies the number of hidden nodes to arrive at the optimum size for the hidden layer. This DNA is only coincidental to our discussion of DIS and the interested reader is refered to our earlier work [1, 14, 36]. Only one hidden layer is used in this work as it is adequate to solve virtually any mapping problem [37].

The DIS scheme can be divided into three phases. Phase I is the statistical ranking of the input variables. Phase II is the training of the network as it builds up from one input and one hidden node. In Phase III, DIS optimizes the input vector by pruning unnecessary input variables if they exist.

3.1 Phase I: Input Variable Ranking

Let us now focus the discussion on the input and output vectors. Let $\mathbf{x}$ be the n-dimensioned input vector $\{x_1, x_2, ..., x_n\}$, and $\mathbf{y}$ be the m-dimensioned output vector $\{y_1, y_2, ..., y_m\}$. Assume that N data patterns $\{\mathbf{x}, \mathbf{y}\}$ have been collected. Also assume that the input provides enough information to describe the output. All the variables have been normalized to lie between zero and one. Let the N patterns of the input vector $\mathbf{x}$ form the matrix $\mathbf{X}$, and the N patterns of the output vector $\mathbf{y}$ form the matrix $\mathbf{Y}$.

Our interest is to arrive at an appropriate measure of the importance of each of the input variables x_i with respect to the output $\mathbf{y}$. One might think that $U(\mathbf{y}, x_i)$ would be an easy and useful measure. However, this measure will not work well in all cases. For example, let us take the data contingency table given in Table 2.1. This is the traditional exclusive-or problem with two additional spurious input variables.

One of these is a constant and the other is a random variable. It is well known that the variables x_1 and x_2 are the most important with respect to the output $\mathbf{y}$. However, ITIA on the four individual input variables gives us $U(\mathbf{y}|x_1) = 0$, $U(\mathbf{y}|x_2) = 0$, $U(\mathbf{y}|x_3) = 0$, and $U(\mathbf{y}|x_4) = 0.1556$. If these values were used as a measure of the importance of the input variables, x_4 would have a higher importance than either x_1 or x_2. In order to properly evaluate the importance of the input variables, we would have to look at the asymmetric association between the output vector $\mathbf{y}$ and all possible combinations of the input variables. In the present example, this would require us to look at all possible combinations $U(\mathbf{y}|x_i x_j)$, $U(\mathbf{y}|x_i x_j x_k)$, and $U(\mathbf{y}|x_i x_j x_k x_l)$, where $i \neq j \neq k \neq l$. Such an analysis would reveal that the assymetric association between $\mathbf{y}$ and any combination of the inputs that involves both x_1 and x_2 will be 1.0, leading us to deduce that x_1 and x_2 are the most important variables. This deduction would be reinforced by the fact that the assymetric association between $\mathbf{y}$ and any combination of inputs that did not involve both x_1 and x_2 is less than 1.0. Such an approach may be plausible for this example, but we will soon encounter a combinatorial explosion of computations as the number of input variables begins to increase. This approach will not be practical for large data sets or models with potentially large numbers of inputs.

Principal component analysis allows us to transform the input data into a form where the variables are uncorrelated to each other. In order to perform PCA, let us convert the input data matrix $\mathbf{X}$ to the matrix $\tilde{\mathbf{X}}$ whose columns, corresponding to the input variables, have been centered to zero mean. The elements of $\tilde{\mathbf{X}}$ are given by

$$\tilde{x}_i = x_i - \bar{x}_i \; ; \quad i = 1, \ldots, n \tag{12}$$

where $\bar{x_i}$ is the mean of the ith variable in the original data matrix $\mathbf{X}$.

Let PCA convert the zero-centered input vector $\tilde{\mathbf{x}}$ into the principal component vector $\boldsymbol{\omega}$, which is also an n-dimensioned vector $\{\omega_1, \omega_2, \ldots, \omega_n\}$. The N patterns of the principal component vector $\boldsymbol{\omega}$ form the matrix $\boldsymbol{\Omega}$. Let $\boldsymbol{\alpha}$ be the matrix of weights α_{ij} that converts $\tilde{\mathbf{X}}$ to $\boldsymbol{\Omega}$.

$$\boldsymbol{\Omega} = \boldsymbol{\alpha}\tilde{\mathbf{X}} \quad . \tag{13}$$

Thus, the ith principal component is given by

$$\omega_i = \sum_{j=1}^{n} \alpha_{ij}\tilde{x_j} = \alpha_{i1}\tilde{x_1} + \alpha_{i2}\tilde{x_2} + \ldots + \alpha_{in}\tilde{x_n} \quad . \tag{14}$$

The values of the conversion matrix $\boldsymbol{\alpha}$ are so chosen that Equation 13 is an orthonormal linear transformation [31]. Each of the PCs are uncorrelated to each other, therefore, if ω_i and ω_j are two principal components, then

$$r = \frac{\sigma_{\omega_i\omega_j}}{\sigma_{\omega_i}\sigma_{\omega_j}} = \frac{E[\omega_i\omega_j] - E[\omega_i]E[\omega_j]}{\sqrt{E[\omega_i^2] - E^2[\omega_i]}\sqrt{E[\omega_j^2] - E^2[\omega_j]}} = 0 \tag{15}$$

where r is the correlation coefficient. The above equation implies that

$$E[\omega_i\omega_j] = E[\omega_i]E[\omega_j] \tag{16}$$

since the denominator in Equation 15 is bounded. Further, since the variables $\tilde{\mathbf{x}}_i$ have been centered to zero mean and the principal components are linear combinations of the $\tilde{\mathbf{x}}_i$'s, each principal component also has a mean, or expectation value, of zero. Equation 16 can therefore be rewritten as

$$E[\omega_i\omega_j] = E[\omega_i]E[\omega_j] = 0 \quad . \tag{17}$$

The above discussion tells us that the principal components are mutually uncorrelated. They are also orthogonal since $E[\omega_i \omega_j] = 0$. A principal component derived in this way can be interpreted as a projection of a data point on an axis in an n-dimensioned hyperspace. The orientation of the axis is given by the coefficients α_{ij} that define the principal component. The n axes corresponding to the n principal components are mutually orthogonal.

The original inputs have now been projected into the space spanned by the PC's. Each PC contains the information in the original data set along one direction. Since the direction represented by this PC is orthogonal to all the other directions represented by the other PC's, we are assured that the information in any particular PC is exclusive of the information in any other PC. Moreover, since each PC is a linear combination of all the input variables, each PC contains some aspects of information from each variable individually as well as some aspects of the information in all combinations of the variables. Together, all the PC's contain all the information present in the input vector. Virtually all of the joint information in the original inputs can be accounted for during ITIA of the individual PC's with respect to the outputs.

The information theoretic measure of association $U(\mathbf{y}|\omega_i)$ can now be considered an appropriate measure of the importance of the ith PC with respect to $\mathbf{y}$. If we are interested in the importance of the principal components with respect to the output vector, then a comparison of $U(\mathbf{y}|\omega_i)$ for all PC's, ω_i, would be in order. However, since the PC's are abstractions and are not given to easy interpretation, we would like to determine the importance of the input variables since they represent the physical system in terms that are easy to understand. Our interest therefore lies not in the

PCs, but in the original input variables. Thus, we need to find some method to determine the inputs' contribution to the PCs.

Statistically, the weight α_{ij} assigns a part of the variance of variable x_j to the principal component ω_i. Therefore, α_{ij} can be viewed as the contribution of x_j to ω_i. Further, $U(\mathbf{y}|\omega_i)$ is the association of ω_i with the output vector $\mathbf{y}$. We define the partial importance of the variable x_j by way of the PC ω_i as

$$I_{x_j|\omega_i} = \alpha_{ij} U(\mathbf{y}|\omega_i) \quad . \tag{18}$$

The variable x_j contributes to all the principal components, and each principal component has a certain measure of association with the output vector. We define the total importance of the variable x_j as

$$I_{x_j} = \alpha_{1j} U(\mathbf{y}|\omega_1) + \alpha_{2j} U(\mathbf{y}|\omega_2) + \ \ldots \ + \alpha_{nj} U(\mathbf{y}|\omega_n) = \sum_{i=1}^{n} \alpha_{ij} U(\mathbf{y}|\omega_i) \quad . \tag{19}$$

The term I_{x_j} is a measure of relative importance as the absolute value of I_{x_j} has little physical meaning. A higher value for I_{x_j} than for I_{x_i} indicates that the jth input variable x_j is more important to the input-output relationship than the ith input variable x_i.

3.2 Phase II: Network growth and training

This phase of the DIS scheme starts by arranging the inputs in the order of their relative importance as defined in the previous section. Training is then initiated with one input node and one hidden node. This first input is the variable with the highest relative importance. As training progresses, the network soon reaches a learning plateau where the network error cannot be reduced below a certain value regardless of the number of successive iterations. At this stage, a node needs to be added to

the network. This node can be added to either the input or the hidden layer. The layer with the lower information theoretic association with the output of the network is chosen as the recipient of the additional node. If $\hat{\mathbf{x}}$ is the input vector, $\hat{\mathbf{h}}$ is the hidden layer vector, and $\hat{\mathbf{y}}$ is the ANN output vector (Figure 2.1), then the condition

$$U(\hat{\mathbf{y}}|\hat{\mathbf{x}}) > U(\hat{\mathbf{y}}|\hat{\mathbf{h}}) \quad . \tag{20}$$

implies that the hidden layer is information deficient compared to the input layer. If the above condition is true, then the new node is added to the hidden layer, or else the new node is added to the input layer. Note that in the above condition, the dimension of the input vector $\hat{\mathbf{x}}$ is determined by the number of inputs being used by the ANN at this stage of training. The dimension of the hidden layer vector $\hat{\mathbf{h}}$ is also determined by the number of hidden nodes used by the ANN. The output vector $\hat{\mathbf{y}}$ is the actual ANN output at the present stage and has a fixed dimension.

The weights connecting the new node are assigned very small random values so that the addition of this node does not disrupt the network performance much. Training is resumed and the network soon reaches another plateau when another node is added to the layer with the lower information theoretic association with the ANN output. This process is continued until the network reaches a satisfactory level of performance as predetermined by the user.

Experience has shown that ANNs typically require more hidden nodes to learn a mapping than to recall the same mapping [14]. It is reasonable to assume that the ANN may also require more inputs to learn than to recall. Thus, it is quite possible that at this stage not all the hidden or input nodes may be necessary. To deal with this situation, we first look at the hidden nodes. The hidden node with the least importance is removed from the network. The importance of each of the

hidden nodes can be determined by a method very similar to the one used to rank the available inputs in Phase I of DIS. But such a calculation is fairly computation intensive since the importance of all the hidden nodes needs to be determined every time a hidden node is to be eliminated. This can significantly slow down the training process. Instead, the importance function used here for the hidden nodes is the derivative importance function used in the DNA scheme from the authors' previous work [1, 14]. The derivative importance measure works well and is computationally inexpensive. It assumes that if changes in the output of a hidden node are more influential in deciding the output of the network than a similar change in the output of another hidden node, then the former node is more important to the dynamic functioning of the network than the latter node. The importance of the ith hidden node with respect to the jth output node is given by

$$I_{(\hat{h}_i|\hat{y}_j)} = E[|\frac{\delta \hat{y}_j}{\delta \hat{h}_i}|]d\hat{h}_i^{max} \quad . \tag{21}$$

where $E[\bullet]$ is the expectation over the entire training set and $d\hat{h}_i^{max}$ is the maximum change in the output of the ith hidden layer node over the entire training set. The above equation gives the partial importance of the ith hidden node with respect to the jth output node. The total importance of the ith hidden node is the sum of the partial importances of that node with respect to all the output nodes. Mathematically,

$$I_{(\hat{h}_i)} = \sum_{j=1}^{m} I_{(\hat{h}_i|\hat{y}_j)} \quad . \tag{22}$$

where m in the number of output layer nodes. The derivatives in Equation 21 are readily calculable given the sigmoidal relationship between $\hat{h}_i$ and $\hat{y}_j$, and this results in very little computational overhead for this importance measure.

Following the elimination of a node, the ANN might require further training, depending on the deleted node's importance to the ANN function. Upon continued training, the smaller network might be able to learn the mapping. If it does, then, again the hidden node with the least importance is deleted. This process is continued until the network is too small to learn the problem. The smallest architecture capable of learning the problem is taken as the model from this phase of DIS.

3.3 Phase III: Removal of unnecessary inputs

Phase III deals with the possibility that the ANN has used some input variables during the training phase that are not important. It is desirable to eliminate such inputs as it could provide a more compact model. For this purpose, we perform a Phase I style ranking of the variables actually being used by the network with respect to the actual output vector $\hat{y}$. The ranking obtained might place the variables used by the network in a different order than the initial ranking in Phase I. This is a reflection of the mapping learned by the network. The input variable with the lowest relative importance is eliminated, and the resultant network is trained further if necessary. It is possible that this network is unable to reduce its error below the preselected target, in which case the network from the end of Phase II is retained as the final model. If however the ANN is able to learn the mapping with one input less than in Phase II, we proceed to use the derivative importance function on the hidden layer to determine if any of the hidden nodes can be eliminated.

The input variables now remaining in the ANN are re-ranked using PCA and ITIA with respect to the actual ANN output, and the least important variable is again eliminated. The whole process described in the previous paragraph is repeated, and

either a new model is achieved or the previous model is retained. The algorithm is stopped when a successful model is not found following the deletion of an input node. In the process of reducing the number of inputs, it is a good practice to test a model on a validation data set, and accept it only if its generalization performance exceeds that of the previous model. This is important as the elimination of an input variable is not guaranteed to provide better generalization. On the other hand we do not validate a model while removing only the hidden nodes as previous work has shown the capability of ANNs to generalize more effectively with lesser hidden nodes [35, 14, 36]. Note that the final architecture arrived at by the DIS scheme is influenced by the selection of various training parameters such as the learning rate, the random weights used in the starting architecture, the pre-selected learning target and the number of iterations spent on a learning plateau before a node is added.

4. COMPUTER SIMULATION RESULTS

4.1 Exclusive-or problem

For our first demonstration we return to the exclusive-or problem. The data given in Table 2.1 is used for training. The problem is defined by input 1, input 2 and the output as shown in the table. Input 3 and input 4 are spuriously generated data unimportant to the output value. Input 3 is a constant and input 4 is a random variable.

Phase I of the DIS scheme is the statistical ranking of the variables, and the results of this analysis can be seen in Table 2.2. The input variable ranking analysis shows that input 2 has the highest relative importance followed closely by input 1. Ideally, input 1 and input 2 should have equal importance, but the slight difference

seen in Table 2.2 is due to machine round off error during the numerical computations. Input 4 (random values) ranks third with a lower relative importance. Also, the table shows that input 3 (constant value) has no importance to determining the output.

In Phase II, an ANN is trained on the data. The preset training target for the ANN was an RMS error of 0.01 over the training data. The training history of this ANN can be seen in Table 2.3. The network starts with one input (input 2) and one hidden node. This is not sufficient to provide the desired mapping, so the ANN reaches a learning plateau and another node needs to be added. An information theoretic interdependency analysis shows that the input layer is better associated than the hidden layer with the output layer. Thus, a second node is added to the information deficient hidden layer. The addition of this node disrupts the network and increases the RMS error slightly. Continued training of this one input, two hidden, one output (1 x 2 x 1) network does not decrease the RMS error significantly, and a node is added to the input layer on the basis of ITIA of the hidden and input layers. The new input is input 1 since it is the second most important variable (see Table 2.2). The ANN is now able to significantly reduce its RMS error before reaching another plateau. A hidden node is added, and the 2 x 3 x 1 network reaches a plateau short of the target RMS error. Analysis of this network suggests the addition of another hidden node. Upon continued training, the 2 x 4 x 1 network is able to reach an RMS error below the preset target of 0.01. The derivative importance of each of the four hidden nodes is now calculated as given by Equations 21 and 22. The node with the lowest importance is deleted, and the resultant 2 x 3 x 1 network is trained until it can learn the function. Again, the hidden node with the least derivative importance is deleted from the network, and the 2 x 2 x 1 network is further trained. This

network is also able to learn the functional mapping and another node is removed. The 2 x 1 x 1 network is incapable of learning this function and a second node is added. The process of adding and deleting nodes is continued until it is apparent that 2 x 2 x 1 is the optimum architecture at the end of this phase of DIS. Phase III, wherein any unnecessary inputs are removed, is not attempted with this model as we know that the two inputs used by the present model are the appropriate ones and eliminating any of them is an exercise in futility.

4.2 Continuous function mapping

The function selected for this demonstration is the two dimensioned (2-D) sine function given by the equation

$$z = sin(\pi x)sin(\pi y) \tag{23}$$

where x and y lie in the range [0,1]. There were 600 patterns in the training data set, each pattern corresponding to a randomly selected point on the x-y square. Table 2.4 shows ten sample patterns. The first two input variables are x and y. The next two variables are derived from x and y as shown in the table. The fifth variable is randomly generated. The output is given by equation 23.

The Phase I ranking of the input variables (see Table 2.5) shows that input 1 and input 2 are the most important in determining the functional relationship between the inputs and the output. Input 5, input 4 and input 3 follow in that order. The Phase II training history of an ANN on this data can be seen in Table 2.6. The target RMS error for this example is 0.05. The training followed a pattern very similar to the exclusive-or problem. The network with the architecture 2 x 8 x 1 and RMS error 0.0487 is selected as the model from this phase. The two inputs used by this model

are known to be the appropriate variables for this problem. Therefore, Phase III is not implemented in this example.

It is possible that the choice of random numbers for input 1, input 2, and input 5 are such that a ranking of the variables gives us a different order of relative importance than in the previous case. Such a ranking analysis with a new data set for the 2-D sine function can be seen in Table 2.7. The random variable input 5 is ranked higher than the y variable input 2. The phase II training history for this data is shown in Table 2.8. At the end of this phase, a model is obtained with the architecture 3 x 13 x 1 and training RMS error of 0.0489. The three input variables used by this model are statistically re-ranked in Phase III. The output vector used for this ranking is the actual output of the ANN model. The results of this new ranking can be seen in Table 2.9. On the basis of this ranking, input 5 is deleted from the model and the resultant network is trained further. The training progress in Phase III DIS can be seen in Table 2.10. The final model has an architecture of 2 x 8 x 1, the same as in the previous case, and an RMS error of 0.0479. This example shows that an inappropriate ranking in Phase I can be corrected in Phase III based on the ANN learning in Phase II.

4.3 Temperature prediction problem

We have seen the DIS scheme applied to two known benchmark problems. In this section we will apply the scheme to a real world problem, the solution to which is not known. This problem involves the prediction of noon-time temperatures at the Chicago lake-front area based on historical data [38]. The historical variables that we expect to use for the prediction of noon temperatures are shown in Table 2.11. Data

for the months of June, July and August of 1991 and 1992 were used for training, and data for those same months in 1993 were used to test the model. Table 2.11 also shows the Phase I relative importance and ranking of each of the variables.

In Phase II, an ANN was trained using the DIS scheme. The target RMS error of 0.01 was achieved with an architecture of 23 x 32 x 1. The 23 variables used by this model are indicated with an asterisk in Table 2.11. This ANN model was tested to predict the noon temperatures in June, July and August of 1993. The model had an average error of 3.8°F for 1993. In Phase III, the 23 variables were re-ranked. This new ranking can be seen in Table 2.12. The variable "wind chill 4 days ago" was ranked 22nd in Phase I but is now ranked 23rd. This variable is eliminated and the resultant network trained further. The number of hidden nodes is varied and the network learns the mapping with an architecture of 22 x 34 x 1. But this network had slightly worse performance on the recall of the 1993 test data than the network with the 23 inputs. Thus, the earlier network obtained at the end of Phase II is taken as the model. The performance of this model on the 1993 data can be seen in Figure 2.2. The average error of 3.8°F is comparable to the 3°F average error obtained by meteorologists in predicting temperatures with a 24 hour lead time [39]. The square of the linear correlation between the actual temperature and the ANN prediction is 0.80. Taking into account that the summer of 1993 was an abnormally rainy summer, causing extensive floods in the American Midwest, this degree of accuracy in prediction is remarkable. A careful examination of Table 2.11 reveals that most of the variables used by the ANN were historical data of precipitation, wind speed, relative humidity and sky cover. In fact, historical data regarding temperature, heat index and wind chill were almost totally neglected. This input variable selection

was based solely on the data corresponding to the years 1991 and 1992. The model built on this analysis looked at precisely those variables that would indicate the rainy conditions prevailing in 1993. This possibly explains the resilience of the ANN model.

5. CONCLUSIONS

The DIS scheme presented in this paper is a very powerful technique that completely eliminates the need to preselect network architectures. The scheme automatically selects the input variables needed for the model by analyzing the given data. It also selects the number of hidden nodes needed for the training process, and subsequently eliminates the hidden nodes and input variables not necessary for recalling the learned functional mapping. The computer simulations show that DIS works effectively in learning both discrete and continuous mapping functions. Furthermore, the results in the temperature prediction problem demonstrate the usefulness of DIS with respect to real world applications. Using the DIS scheme, the ANN completely ignored the intuitive variables such as temperature one day ago, and built an extremely resilient model based on wind speeds, sky cover and humidity. The DIS scheme can be used to model almost any real world system provided there is adequate representative data. The user needs to only make a broad decision as to which variables might be needed to model the system, and the DIS scheme will determine the appropriate variables from this set. Such a degree of automation will allow ANN models to be developed without guesswork and will lead to substantial reductions in developmental time. This will significantly increase ANN applicability and use.

BIBLIOGRAPHY

[1] R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE Acoustics Speech and Signal Processing Magazine*, **4**, 4 (April 1987).

[2] M. Caudill, "Neural Networks Primer, Part 3," *AI Expert*, 53 (June 1988).

[3] E. D. Karnin, "A Simple Procedure for Pruning Back-Propagation Trained Neural Network," *IEEE Transactions on Neural Networks*, **1.2**, 239 (June 1990).

[4] M. C. Mozer and P. Smolensky, "Skeletonization: A Technique for Trimming the Fat from a Neural Network via Relevance Assessment," *Advances in Neural Information Processing Systems*, Vol. 1, D. Touretzky, Ed., (1989).

[5] V. Eigel-Danielson and M. F. Augusteijn, "Neural Network Pruning and its Effect on Generalization, Some Experimental Results," *Neural Parallel & Scientific Computations*, **1**, 59 (1993).

[6] A. S. Weigend, D. E. Rumelhart, and B. A. Huberman, "Generalization by Weight Elimination Applied to Currency Exchange Rate Prediction," *Proc. International Joint Conference on Neural Networks*, Vol. 1, Seattle, Washington, July 8-12, 1991, p. 837.

[7] Y. Won and R. Pimmel, "A Comparison of Connection Pruning Algorithms with Back-Propagation." *Intelligent Engineering Systems Through Artificial Neural Networks*, p. 113, C. H. Dagli, S. R. T. Kumara, and Y. C. Shin, Eds., ASME Press, New York (1991).

[8] T. Ash, "Dynamic Node Creation in Backpropagation Networks," *Proc. International Joint Conference on Neural Networks*, Vol. 2, Washington, D.C., June 18-22, 1989, p. 623.

[9] Y. Hirose, K. Yamashita, and S. Hijiya, "Back Propagation Algorithm That Varies the Number of Hidden Nodes," *Neural Networks*, **4**, 61 (1991).

[10] J. Vaario and S. Ohsuga, "Adaptive Neural Architectures Through Growth Control," *Intelligent Engineering Systems Through Artificial Neural Networks*, p. 11, C. H. Dagli, S. R. T. Kumara, and Y. C. Shin, Eds., ASME Press, New York (1991).

[11] B. R. Upadhyaya and E. Eryurek, "Application of Neural Networks for Sensor Validation and Plant Monitoring," *Nucl. Technol.*, **97**, 170 (February 1992).

[12] E. B. Bartlett, "Self Determination of Input Variable Importance Using Neural Networks," *Neural, Parallel & Scientific Computations*, **2**, 103 (1994).

[13] A. Basu and E. B. Bartlett, "Detecting Faults in a Nuclear Power Plant by Using Dynamic Node Architecture Artificial Neural Networks," *Nuclear Science and Engineering*, **116**, 313 (1994).

[14] E. B. Bartlett and A. Basu, "A Dynamic Node Architecture Scheme for Backpropagation Neural Networks," *Intelligent Engineering Systems Through Artificial Neural Networks*, p. 101, C. H. Dagli, S. R. T. Kumara, and Y. C. Shin, Eds., ASME Press, New York (1991).

[15] J. S. Judd, *Neural Network Design and the Complexity of Learning*, The MIT Press, Cambridge, Mass. (1990).

[16] H. S. Wilf, *Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, New Jersey. (1986).

[17] D. F. Morrison, *Applied Linear Statistical Methods*, Prentice-Hall, Englewood Cliffs, N.J. (1983).

[18] S. J. Press, *Applied Multivariate Analysis*. Holt, Rinehart and Winston. New York, (1972).

[19] Y. Isogawa, *Comparing Multivariate Linear Functional Relationships*, Kobe University of Commerce, (1991).

[20] T. L. Lane, "The Importance of Input Variables to a Neural Network Fault-Diagnostic System for Nuclear Power Plants," MS Thesis, Iowa State University (1991).

[21] A. Papoulis, *Probability, random variables, and Stochastic Processes*, McGraw-Hill Book Company, New York (1965).

[22] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. University of Illinois Press. Urbana. Ill. (1971).

[23] S. Watanabe, *Knowing and Guessing: A Quantitative Study of Inference and Information*, John Wiley and Sons, New York (1969).

[24] R. B. Ash, *Information Theory*, Dover Publications, New York (1990).

[25] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York (1986).

[26] G. H. Dunteman, *Principal Components Analysis*, Sage, Newbury Park, Calif., (1989).

[27] A. N. Kshirsagar, *Multivariate Analysis*, Marcel Dekker, New York (1972).

[28] P. Földiák, "Adaptive Network for Optimal Linear Feature Extraction," it Proc. International Joint Conference on Neural Networks, Vol. 1, Washington, D.C., June 18-22, 1989, p. 401.

[29] P. Baldi and K. Hornik, "Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima," *Neural Networks*, **2**, 53 (1989).

[30] E. Oja, "Principal Components, Minor Components, and Linear Neural Networks," *Neural Networks*, **5**, 927 (1992).

[31] Y. Chauvin, "Principal Component Analysis by Gradient Descent on a Constrained Linear Hebbian Cell," *Proc. International Joint Conference on Neural Networks*, Vol. 1, Washington, D.C., June 18-22, 1989, p. 373.

[32] K. Hornik and C. M. Kuan, "Convergence Analysis of Local Feature Extraction Algorithms," *Neural networks*, **5**, 229 (1992).

[33] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipies*, Cambridge University Press, Cambridge (1986).

[34] E. B. Bartlett, "Dynamic Node Architecture Learning: An Information Theoretic Approach," *Neural Networks*, **7**, 129 (1994).

[35] R. Hecht-Nielsen, "Theory of the Backpropagation Neural Network," *Proc. International Joint Conference on Neural Networks*, Vol. 1, Washington, D.C., June 18-22, 1989, p. 593.

[36] E. D. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, p. 318, The MIT Press, Cambridge, Mass. (1986).

[37] R. Hecht-Nielsen, "Kolmogorov's Mapping Neural Network Existence Theorem," *Proc. IEEE International Conference on Neural Networks*, Vol. 2, San Diego, California, June 21-24, 1987, p. 11.

[38] D. Hansen, Commonwealth Edison Electric Company, Hourly data for weather in the Chicago area obtained through personal communication.

[39] "A 20-year summary of national weather service verification results for temperature and precipitation," *NOOA Technical Memorandum NWS FCST 31*, National Weather Service, Silver Spring, Maryland (1986).

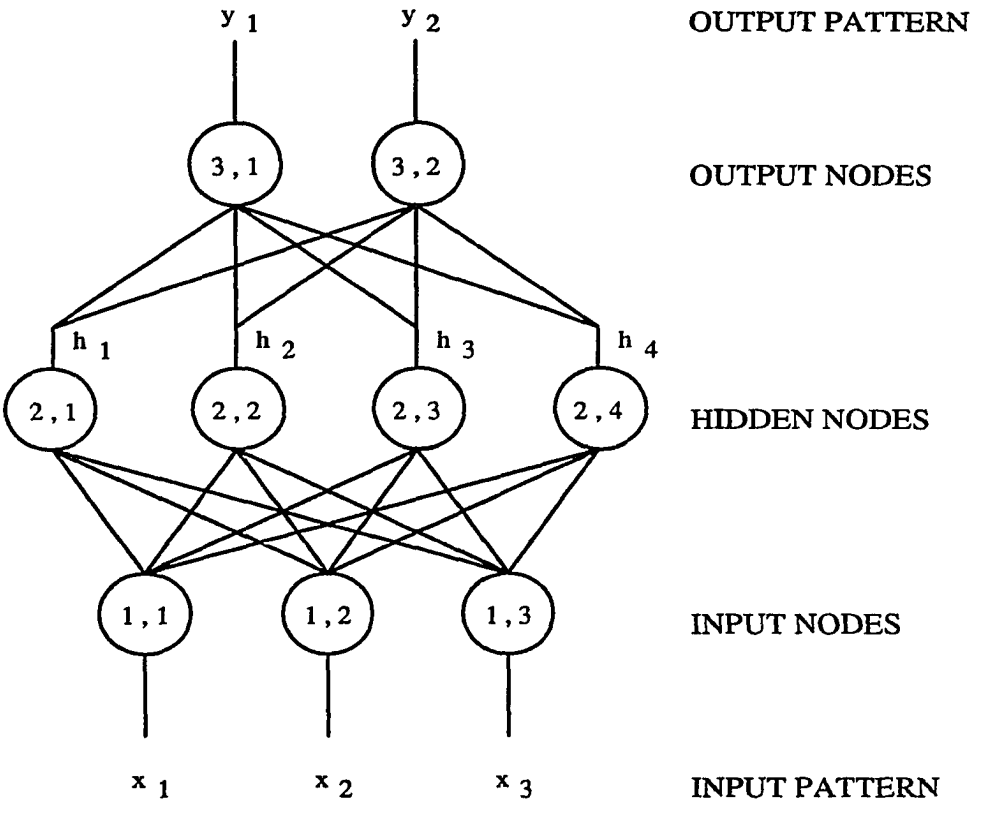

Figure 2.1: A three-layered feed-forward backpropagation neural network.

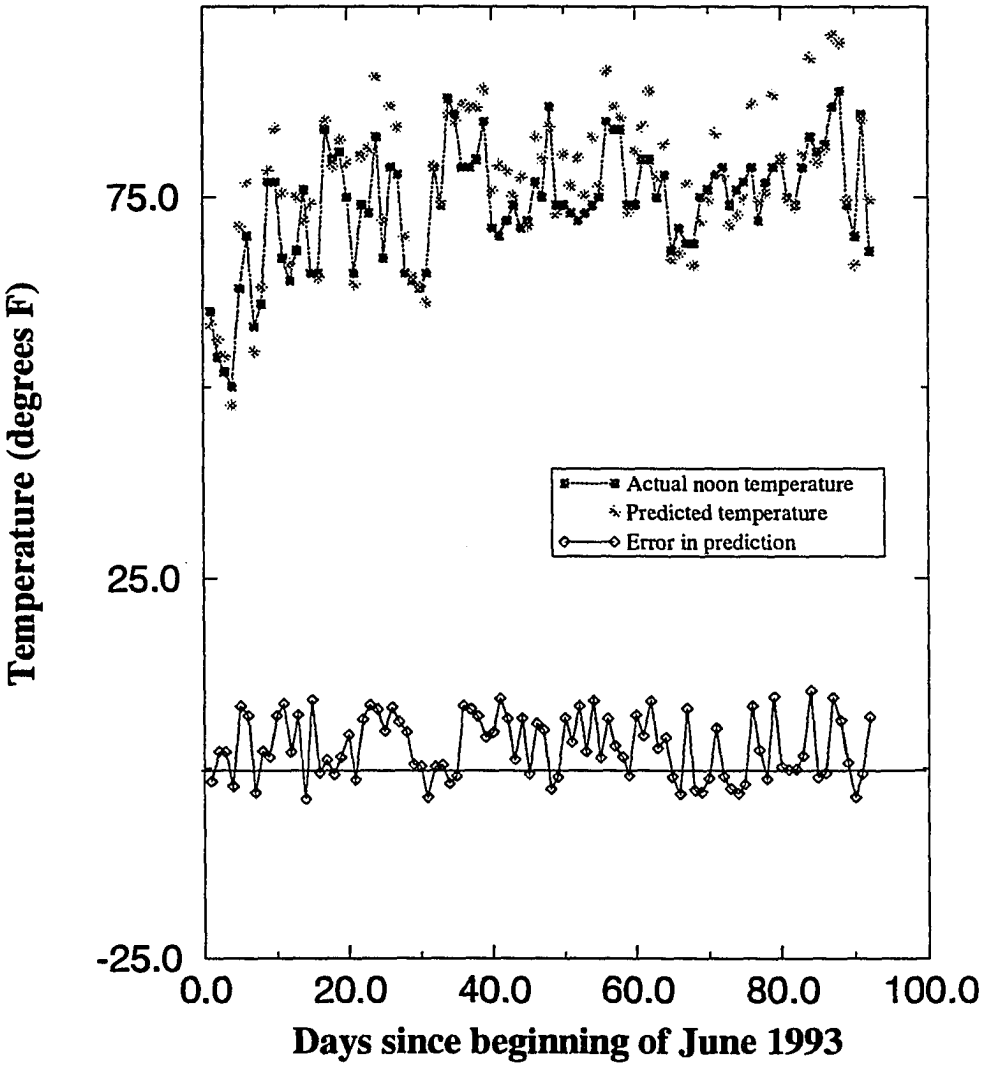

Figure 2.2: Noon-time temperature prediction for months of June, July, and August, 1993.

Table 2.1: Exclusive-or training data

Pattern	Input1 (switch)	Input2 (switch)	Input3 (constant)	Input4 (random)	Output
1	0.0	0.0	0.5	0.338	0.0
2	0.0	1.0	0.5	0.417	1.0
3	1.0	0.0	0.5	0.793	1.0
4	1.0	1.0	0.5	0.981	0.0
5	0.0	0.0	0.5	0.144	0.0
6	0.0	1.0	0.5	0.583	1.0
7	1.0	0.0	0.5	0.891	1.0
8	1.0	1.0	0.5	0.510	0.0

Table 2.2: Phase I ranking analysis of exclusive-or training data

Rank	Relative Importance	Variable Number
1	0.319	2
2	0.318	1
3	0.117	4
4	0.000	3

Table 2.3: Phase II training history for exclusive-or problem

Architecture			RMS	Architecture			RMS
Input	Hidden	Output	Error	Input	Hidden	Output	Error
Start: Input2 & 1 hidden node				Delete hidden node			
1	1	1	0.5842	2	2	1	0.0417
1	1	1	0.4919	2	2	1	0.0092
Add hidden node				Delete hidden node			
1	2	1	0.5528	2	1	1	0.3718
1	2	1	0.4692	2	1	1	0.3373
Add Input1				Add hidden node			
2	2	1	0.4921	2	2	1	0.3136
2	2	1	0.1822	2	2	1	0.0181
Add hidden node				Add hidden node			
2	3	1	0.2011	2	3	1	0.0201
2	3	1	0.0813	2	3	1	0.0098
Add hidden node				Delete hidden node			
2	4	1	0.0886	2	2	1	0.0318
2	4	1	0.0097	2	2	1	0.0096
Delete hidden node				Delete hidden node			
2	3	1	0.0366	2	1	1	0.4181
2	3	1	0.0098	2	1	1	0.3117

Table 2.4: 2-D sine continuous function sample training data

Input1 x	Input2 y	Input3 (x+y)/2	Input4 xy	Input5 random	Output $\sin(\pi x)\sin(\pi y)$
0.3490	0.6115	0.4802	0.2134	0.6883	0.8357
0.7335	0.8809	0.8072	0.6461	0.3155	0.2716
0.4137	0.2395	0.3266	0.0991	0.4524	0.6585
0.8166	0.8650	0.8408	0.7063	0.6887	0.2243
0.8356	0.8996	0.8676	0.7517	0.1402	0.1532
0.5348	0.3394	0.4371	0.1815	0.0091	0.8702
0.3167	0.7625	0.5396	0.2415	0.8803	0.5693
0.4034	0.8048	0.6041	0.3247	0.0911	0.5492
0.0888	0.7212	0.4050	0.0641	0.4517	0.2116
0.8448	0.8104	0.8276	0.6847	0.6096	0.2628

Table 2.5: Phase I ranking analysis of 2-D sine continuous function mapping problem (case 1).

Rank	Relative Importance	Variable Number
1	0.086	2
2	0.080	1
3	0.071	5
4	0.047	4
5	0.042	3

Table 2.6: Phase II training history for 2-D sine continuous function mapping (case 1).

Architecture			RMS	Architecture			RMS
Input	Hidden	Output	Error	Input	Hidden	Output	Error
Start: Input2 & 1 hidden node				Delete hidden node			
1	1	1	0.4421	2	9	1	0.0966
1	1	1	0.3615	2	9	1	0.0614
Add hidden node				Delete hidden node			
1	2	1	0.4049	2	10	1	0.0819
1	2	1	0.3384	2	10	1	0.0492
Add hidden node				Add hidden node			
1	3	1	0.3968	2	9	1	0.1184
1	3	1	0.3137	2	9	1	0.0498
Add Input1				Add hidden node			
2	3	1	0.4161	2	8	1	0.0922
2	3	1	0.2718	2	8	1	0.0487
Add hidden node				Delete hidden node			
2	4	1	0.3193	2	7	1	0.1637
2	4	1	0.2066	2	7	1	0.0833
Delete hidden node				Delete hidden node			
2	5	1	0.2828	2	8	1	0.0982
2	5	1	0.1196	2	8	1	0.0702
Delete hidden node				Delete hidden node			
2	6	1	0.1311	2	9	1	0.1281
2	6	1	0.0992	2	9	1	0.0499
Delete hidden node				Delete hidden node			
2	7	1	0.1362	2	8	1	0.0896
2	7	1	0.0986	2	8	1	0.0496
Delete hidden node				Delete hidden node			
2	8	1	0.1465	2	7	1	0.1662
2	8	1	0.0752	2	7	1	0.0822

Table 2.7: Phase I ranking analysis of 2-D sine continuous function mapping problem (case 2).

Rank	Relative Importance	Variable Number
1	0.066	2
2	0.062	5
3	0.061	1
4	0.039	3
5	0.036	4

Table 2.8: Phase II training history for 2-D sine continuous function mapping (case 2).

Architecture			RMS	Architecture			RMS
Input	Hidden	Output	Error	Input	Hidden	Output	Error
Start: Input2 & 1 hidden node				Delete hidden node			
1	1	1	0.3988	3	9	1	0.0966
1	1	1	0.2432	3	9	1	0.0614
Add hidden node				Add hidden node			
1	2	1	0.2817	3	10	1	0.1202
1	2	1	0.2300	3	10	1	0.0731
Add Input5				Add hidden node			
2	2	1	0.3121	3	11	1	0.1182
2	2	1	0.1422	3	11	1	0.0642
Add hidden node				Add hidden node			
2	3	1	0.1757	3	12	1	0.0961
2	3	1	0.1392	3	12	1	0.0641
Add hidden node				Add hidden node			
2	4	1	0.1801	3	13	1	0.0993
2	4	1	0.1358	3	13	1	0.0518
Add hidden node				Add hidden node			
2	5	1	0.1842	3	14	1	0.0839
2	5	1	0.1355	3	14	1	0.0492
Add hidden node				Delete hidden node			
2	6	1	0.1907	3	13	1	0.1166
2	6	1	0.1270	3	13	1	0.0497
Add Input1				Delete hidden node			
3	6	1	0.1362	3	12	1	0.1386
3	6	1	0.0986	3	12	1	0.0772
Add hidden node				Add hidden node			
3	7	1	0.1287	3	13	1	0.0983
3	7	1	0.0752	3	13	1	0.0489
Add hidden node				Delete hidden node			
3	8	1	0.1118	3	12	1	0.1226
3	8	1	0.0711	3	12	1	0.0819

Table 2.9: Phase III ranking analysis of the three variables used in the Phase II model for the 2-D sine continuous function mapping problem (case 2).

Rank	Relative Importance	Variable Number
1	0.064	2
2	0.061	1
3	0.058	5

Table 2.10: Phase III training history for 2-D sine continuous function mapping (case 2) after deletion of an input.

Architecture			RMS	Architecture			RMS
Input	Hidden	Output	Error	Input	Hidden	Output	Error
Delete Input5				Delete hidden node			
2	13	1	0.0912	2	7	1	0.2210
2	13	1	0.0490	2	7	1	0.0767
Delete hidden node				Add hidden node			
2	12	1	0.1803	2	8	1	0.1478
2	12	1	0.0496	2	8	1	0.0591
Delete hidden node				Add hidden node			
2	11	1	0.1383	2	9	1	0.1272
2	11	1	0.0492	2	9	1	0.0475
Delete hidden node				Delete hidden node			
2	10	1	0.1774	2	8	1	0.0919
2	10	1	0.0559	2	8	1	0.0496
Add hidden node				Delete hidden node			
2	11	1	0.1903	2	7	1	0.1103
2	11	1	0.0488	2	7	1	0.0816
Delete hidden node				Add hidden node			
2	10	1	0.1019	2	8	1	0.1229
2	10	1	0.0499	2	8	1	0.0479
Delete hidden node				Delete hidden node			
2	9	1	0.2016	2	7	1	0.1857
2	9	1	0.0497	2	7	1	0.0721
Delete hidden node				Add hidden node			
3	8	1	0.1296	2	8	1	0.1692
3	8	1	0.0492	2	8	1	0.0494

Table 2.11: The variables available for the temperature prediction problem, the Phase I relative importance of the variables, and their ranking. All values are at the noon hour.

Var. #	Variable Description	Rel. Imp.	Var. Rank	Var. #	Variable Description	Rel. Imp.	Var. Rank
	Temperature				Wind speed		
1	- 1 year ago	0.026	48	25	- 1 year ago*	0.196	11
2	- 1 day ago	0.033	39	26	- 1 day ago*	0.200	9
3	- 2 days ago	0.032	44	27	- 2 days ago*	0.206	7
4	- 3 days ago	0.032	45	28	- 3 days ago*	0.196	12
5	- 4 days ago	0.030	47	29	- 4 days ago*	0.203	8
6	- 5 days ago	0.033	41	30	- 5 days ago*	0.198	10
	Heat index				Sky cover		
7	- 1 year ago	0.031	46	31	- 1 year ago*	0.062	19
8	- 1 day ago	0.032	43	32	- 1 day ago*	0.057	20
9	- 2 days ago	0.034	37	33	- 2 days ago*	0.055	23
10	- 3 days ago	0.035	36	34	- 3 days ago*	0.056	21
11	- 4 days ago	0.035	35	35	- 4 days ago	0.048	29
12	- 5 days ago	0.039	31	36	- 5 days ago	0.050	28
	Wind chill				Precipitation		
13	- 1 year ago	0.048	30	37	- 1 year ago*	0.252	6
14	- 1 day ago	0.051	27	38	- 1 day ago*	0.267	2
15	- 2 days ago	0.054	24	39	- 2 days ago*	0.253	5
16	- 3 days ago	0.052	26	40	- 3 days ago*	0.260	3
17	- 4 days ago*	0.056	22	41	- 4 days ago*	0.270	1
18	- 5 days ago	0.053	25	42	- 5 days ago*	0.257	4
	Wind direction				Relative humidity		
19	- 1 year ago	0.033	42	43	- 1 year ago*	0.106	14
20	- 1 day ago	0.033	40	44	- 1 day ago*	0.095	18
21	- 2 days ago	0.034	38	45	- 2 days ago*	0.105	15
22	- 3 days ago	0.039	32	46	- 3 days ago*	0.103	17
23	- 4 days ago	0.037	33	47	- 4 days ago*	0.104	16
24	- 5 days ago	0.036	34	48	- 5 days ago*	0.110	13

* Variables used in the Phase II ANN model.

Table 2.12: The Phase III re-ranking of the variables based on the output of the ANN from Phase II.

ANN Var.	Variable Description	New Importance	New Rank
1	Precipitation 4 days ago	0.189	2
2	Precipitation 1 day ago	0.198	1
3	Precipitation 3 days ago	0.160	5
4	Precipitation 5 days ago	0.157	6
5	Precipitation 2 days ago	0.176	4
6	Precipitation 1 year ago	0.185	3
7	Wind speed 2 days ago	0.135	11
8	Wind speed 4 days ago	0.154	7
9	Wind speed 1 day ago	0.136	9
10	Wind speed 5 days ago	0.136	10
11	Wind speed 1 year ago	0.140	8
12	Wind speed 3 days ago	0.125	12
13	Relative humidity 5 days ago	0.070	17
14	Relative humidity 1 year ago	0.065	18
15	Relative humidity 2 days ago	0.073	15
16	Relative humidity 4 days ago	0.077	13
17	Relative humidity 3 days ago	0.074	14
18	Relative humidity 1 day ago	0.071	16
19	Sky cover 1 year ago	0.042	19
20	Sky cover 1 day ago	0.040	20
21	Sky cover 3 days ago	0.037	22
22	Wind chill 4 days ago	0.036	23
23	Sky cover 2 days ago	0.038	21

CHAPTER 3. A MULTIPLE NEURAL NETWORK SYSTEM FOR DETECTING FAULTS IN A NUCLEAR POWER PLANT

A paper submitted to IEEE Transactions on Nuclear Science

Anujit Basu and Eric B. Bartlett

ABSTRACT

This paper describes a modular design approach for an artificial neural network-(ANN-) based diagnostic adviser capable of identifying the operating status of a nuclear power plant. The diagnostic capabilities of an adviser based on this modular design can be expanded without having to start afresh. The adviser consists of two components, the root network and the classifier group of networks. The root network monitors the operating condition of a nuclear power plant on a continuous basis, and triggers the classifier group of networks upon detecting an abnormality. The networks in the classifier group subsequently try to classify the abnormal plant status as one of the transients they are trained to recognize. Each network in the classifier group has the capability to differentiate one particular transient from all the other transients used in the development of the adviser. A dynamic input selection (DIS) scheme is used to select the appropriate input variables and optimize the hidden layer for each network. The networks are developed using simulated plant behavior during both

normal and abnormal conditions. Data from the nuclear generating station training simulator at IES Utility's Duane Arnold Energy Center (DAEC) is used in this work. An adviser is initially developed to detect and classify 30 distinct transients based on 47 simulations at various severities. This adviser is later expanded to detect and classify 36 distinct transients using data from 58 transient simulations. In both cases, the noise tolerance capabilities of the adviser is demonstrated.

1. INTRODUCTION

The past decade has witnessed a steady increase in the emphasis on safety and reliability of nuclear power plant operations. Artificial intelligence techniques are being investigated to further sustain such efforts. Artificial neural network- (ANN-) based models are presently being developed for a wide variety of applications in the nuclear power plant environment. Such applications include fault diagnostics [1, 2, 3, 4, 5, 6, 7], component monitoring [8, 9, 10], sensor validation [11], and parameter estimation [12, 13, 14]. Previous work by the authors has effectively demonstrated the feasibility of using ANNs for a broad based fault diagnostic adviser (refered to hereafter as Adviser 1) for a nuclear power plant [1]. Faults in the various systems and components of nuclear power plants cause distinct behavioral transients. These transients are identified by the adviser and the particular faults that cause them are thus detected. This paper explains the development of an ANN based nuclear power plant adviser with an extent of design flexibility and performance unmatched by any previous attempts to develop such diagnostic systems.

One of the shortcomings with the previously developed advisers [1, 2, 3, 15] was their inability to adapt to changing requirements such as detecting an increasing

number of transients. Most of these advisers depended on one ANN to detect and classify a transient. Such networks formed extremely complex decision boundaries to accomplish their respective diagnostic tasks. This made it difficult to change the existing model in the event of plant modifications. Moreover, once a network was trained, the chances of expanding the number of transients were limited by the number of output nodes used in the network. One of the advisers (Adviser 1) [1] attempted to separate the problem by using a hierarchy of two networks. The "root" network detected the onset of a transient by classifying the plant status as "abnormal." The classifier network subsequently attempted to classify the abnormal condition as one of 27 distinct transient conditions. In spite of separating the problem into the two tasks of detecting and classifying the transients, the classifier network had to perform a fairly complex classification task. This network needed five output nodes to differentiate between 27 classes. The maximum number of output classes that can be formed by this network is 2^5, assuming binary outputs. Thus, any subsequent attempt to further train this network to classify additional transients would be limited to a total of 32 transients.

The fault diagnostic adviser design described in this paper takes the hierarchical approach further by utilizing numerous ANNs with a corresponding decrease in the complexity of the task performed by each network. The new design utilizes a root network that continuously monitors the status of the power plant. Its classification task is very simple, to decide if the plant is in a normal or abnormal condition. In this respect, the root network is similar to the root network in Adviser 1 [1]. Once the root network decides that the plant is in an abnormal condition, numerous networks in the classifier group try to classify the transient in progress. Each network in the classifier

group is trained to recognize one transient to the exclusion of all the other transients used to develop the adviser. Thus, the classifier group consists of the same number of ANNs as the number of transients that the adviser is required to classify. The task of transient classification has now been distributed between numerous ANNs. Since each ANN is required to differentiate its assigned transient from all other transients, its task is much simpler than a single classifier network that tries to classify all the different transients.

The classifier network in Adviser 1 used all the 97 available instrumentation data variables as inputs to perform its task. Given the complexity of its task, most of these variables would likely have been necessary. In fact, these 97 variables were selected with this precise scenario in mind, wherein one network would try to classify all the transients of interest. On the other hand, ANNs in the classifier group of the present design solve a much simpler problem, and it is entirely possible that many of them can perform their task by using only a small subset of the 97 variables. Reducing the number of input variables by selecting an appropriate subset of the available variables and reducing the number of output variables by simplifying the classification task would lead to a significant decrease in the training effort. The non-polynomial-time completeness of neural network learning [16] implies that as the number of variables n increases, the complexity of obtaining a solution increases faster than a polynomial of order n [17]. The Dynamic Input Selection (DIS) [18] scheme developed by the authors makes it possible to realize this advantage of reduced learning complexity. DIS is an advancement on the Dynamic Node Architecture (DNA) scheme used to develop Adviser 1. While DNA attempted to optimize the hidden layer of a network during training, DIS attempts to optimize both the input and hidden layers.

The next section outlines the working of DIS, and illustrates the algorithm using a benchmark problem. Section 3 gives a description of the design and development of the nuclear power plant fault diagnostic adviser. This section also demonstrates the ease with which the diagnostic capability of this adviser can be increased to classify a larger number of transients. Section 4 details the performance of the adviser as initially developed, and after the expansion of its capabilities. Both pure simulator data and data corrupted by 3% uniform noise are used to test the adviser. Conclusions are presented in Section 5.

2. ARTIFICIAL NEURAL NETWORK ARCHITECTURES

The artificial neural networks used in this work are two-layered feed-forward networks trained with the backpropagation training algorithm. Such ANNs learn to map functions from example data, and hopefully generalize this information to novel data [19]. An ANN's ability to generalize is highly dependent on its architecture and input vector [20]. If the hidden layer is too large, the ANN will memorize the example data and will have poor generalization. On the other hand, too small a hidden layer will increase the training time, and might even be unable to learn the example data. Similarly, too big an input vector will increase the training complexity, and too small an input vector will lack adequate information to solve the problem. Although no one else [18, 21], to our knowledge, has investigated input vector determination, many researchers have investigated methods to arrive at the optimum number of nodes in a hidden layer. Our DNA scheme [1, 22] is a systematic method that adds and deletes hidden nodes during training to arrive at the optimum number of nodes for a given training problem. DNA was improved and extended to the input layer to arrive at

the dynamic input selection scheme.

2.1 Dynamic input selection

Selecting the inputs for a well known problem is a simple exercise. However, for most real world applications such as the development of a power plant status diagnostic adviser, input selection can be extremely complicated. Most ANN users currently rely on their expertise for selecting the required input variables for specific problems. DIS is a general method that does not rely on user expertise, proceeding systematically by building up the input and hidden layers during the training process. The scheme thus arrives at a viable model without any guesswork.

Dynamic input selection consists of three distinct phases. In Phase I, the available input variables ranked according to their importance to the problem. Phase II is the training phase. The network starts training with the most important input and one hidden node, and adds input variables and hidden nodes as required until the problem is learned. Any input variables that are added during Phase II training but are subsequently deemed unnecessary for recall are eliminated from the network in Phase III. Although a complete description of DIS can be found in reference [18], a brief review is presented below.

Phase I: Input Variable Ranking The object of this phase of DIS is to arrive at an appropriate measure of the importance of each input variable x_i with respect to the output vector $\mathbf{y}$. There are many methods used by statisticians to estimate the number and appropriateness of the variables that could be used to develop a model [23, 24, 25]. Testing the linear correlation of each input variable with respect

to the output(s) over the available data would be an easy method for selecting input variables [26]. Unfortunately, such techniques account only for the linear relationship between the input and output variables [25, 27] and will be incorrect in the context of ANNs. Measures of nonlinear correlation such as information theory might also be used [28, 29, 30]. But an information theoretic interdependency analysis (ITIA) of individual inputs x and y with respect to the output z would not provide an appropriate measure of the importance of the inputs if $z = g(x, y)$ but $z \neq h(x)$ and $z \neq f(y)$. The exclusive-or problem is a simple example of this. It cannot be guaranteed that such a situation will not occur in real world problems. Accounting for all such joint information using ITIA would lead to a combinatorial explosion of calculations and will be intractable for large models. Therefore, before an information theoretic analysis can be performed on the training data, it is important to assure that the variables being subjected to ITIA contain as little joint information as possible.

Principal component analysis (PCA) [31, 32, 33] is a statistical method that transforms an input vector into a set of mutually uncorrelated and orthogonal principal components (PC's). Each PC is a linear combination of the original input variables. In order to perform PCA, we first convert the input data matrix $\mathbf{X}$ to the matrix $\tilde{\mathbf{X}}$ whose columns, corresponding to the input variables, have been centered to zero mean. The elements of $\tilde{\mathbf{X}}$ are given by

$$\tilde{x}_i = x_i - \bar{x}_i : \quad i = 1, \ldots, n \tag{1}$$

where $\bar{x}_i$ is the mean of the ith variable in the original data matrix $\mathbf{X}$. In the above equation, n is the number of input variables. PCA converts the zero-centered input vector $\tilde{\mathbf{x}}$ into the principal component vector $\boldsymbol{\omega}$, which is also an n-dimensioned vector. If the total number of observations, or patterns, is N, the collection of the

N patterns of the principal component vector $\boldsymbol{\omega}$ form the matrix $\boldsymbol{\Omega}$. Let $\boldsymbol{\alpha}$ be the matrix of weights α_{ij} that converts $\tilde{\mathbf{X}}$ to $\boldsymbol{\Omega}$.

$$\boldsymbol{\Omega} = \boldsymbol{\alpha}\tilde{\mathbf{X}} \quad . \tag{2}$$

Thus, the ith principal component is given by

$$\omega_i = \sum_{j=1}^{n} \alpha_{ij}\tilde{x_j} = \alpha_{i1}\tilde{x_1} + \alpha_{i2}\tilde{x_2} + ... + \alpha_{in}\tilde{x_n} \quad . \tag{3}$$

The values of the conversion matrix $\boldsymbol{\alpha}$ are so chosen that Equation 2 is an orthonormal linear transformation [31]. Each of the PC's ω_i are uncorrelated to each other. They are also mutually orthogonal.

An n-dimensioned data point converted into an n-dimensioned PC vector can be interpreted as a projection of the data point on n mutually perpendicular axes in an n-dimensioned hyperspace. This set of axes is unique in the sense that it preserves all the information present in the original data point. But that information has been spread among the PC's such that there is very little joint information between the PC's. All the PC's are as distinct from each other as possible. Any attempt to move two particular PC's further apart will result in them getting closer to other PC's.

With each PC now containing the information in the original data set along one direction, we proceed to perform an ITIA on the individual PC's with respect to the output vector. The measure $U(\mathbf{y}|\omega_i)$ defined as

$$U(\mathbf{y}|\omega_i) = \frac{H(\mathbf{y}) - H(\mathbf{y}|\omega_i)}{H(\mathbf{y})} \tag{4}$$

can be considered a fair measure of the importance of the ith PC with respect to the output vector. In Equation 4, the term $H(\mathbf{y})$ is known as the entropy of $\mathbf{y}$ and

$H(\mathbf{y}|\omega_i)$ is the entropy of $\mathbf{y}$ given ω_i. Entropy of a variable x is defined as

$$H(x) = -\sum_{i=1}^{N} p_i \ln(p_i) \tag{5}$$

where the variable x takes N distinct values, and p_i is the probability of occurrence of any of these values. Conditional entropy $H(y|x)$ is defined as

$$H(y|x) = -\sum_{i,j} p_{ij} \ln \frac{p_{ij}}{p_i} \tag{6}$$

where the subscripts i and j refer to the variables x and y respectively. It can also be shown that [34]

$$H(y|x) = H(x,y) - H(x) \quad . \tag{7}$$

We have now ranked the PC's in the order of their importance with respect to the output vector. But PC's are abstractions not given to easy interpretation and we would like to determine the importance of the input variables since they represent the physical system in terms that are easy to understand. Our interest therefore lies not in the PC's, but in the original input variables. Thus, we need to find some method to determine the inputs' contribution to the PC's. Statistically, the weight α_{ij} (in Equation 3) assigns a part of the variance of variable x_j to the principal component ω_i. Therefore, α_{ij} can be viewed as the contribution of x_j to ω_i. Further, $U(\mathbf{y}|\omega_i)$ is the association of ω_i with the output vector $\mathbf{y}$. We define the partial importance of the variable x_j by way of the PC ω_i as

$$I_{x_j|\omega_i} = \alpha_{ij} U(\mathbf{y}|\omega_i) \quad . \tag{8}$$

The variable x_j contributes to all the principal components, and each principal component has a certain measure of association with the output vector. We define the

total importance of the variable x_j as

$$I_{x_j} = \alpha_{1j} U(\mathbf{y}|\omega_1) + \alpha_{2j} U(\mathbf{y}|\omega_2) + \ \ldots \ + \alpha_{nj} U(\mathbf{y}|\omega_n) = \sum_{i=1}^{n} \alpha_{ij} U(\mathbf{y}|\omega_i) \quad . \quad (9)$$

The term I_{x_j} is a measure of relative importance; the absolute value of I_{x_j} has little physical meaning. A higher value for I_{x_j} than for I_{x_i} indicates that the jth input variable x_j is more important to the input-output relationship than the ith input variable x_i.

Phase II: Network Growth and Training This phase of the DIS scheme starts by arranging the inputs in the order of their relative importance as arrived at in Phase I. Training is then initiated with one input node and one hidden node. This first input is the variable with the highest relative importance. As training progresses, the network soon reaches a learning plateau where the network error cannot be reduced below a certain value regardless of the number of successive iterations. At this stage, a node needs to be added to the network. This node can be added to either the input or the hidden layer. The layer with the lower information theoretic association with the output of the network is chosen as the recipient of the additional node. If $\hat{\mathbf{x}}$ is the input vector, $\hat{\mathbf{h}}$ is the hidden layer vector, and $\hat{\mathbf{y}}$ is the ANN output vector, then the condition

$$U(\hat{\mathbf{y}}|\hat{\mathbf{x}}) > U(\hat{\mathbf{y}}|\hat{\mathbf{h}}) \quad . \quad (10)$$

implies that the hidden layer is information deficient compared to the input layer. If the above condition is true, then the new node is added to the hidden layer, or else the new node is added to the input layer. Note that in the above condition, the dimension of the input vector $\hat{\mathbf{x}}$ is determined by the number of inputs being used by

the ANN at this stage of training. The dimension of the hidden layer vector $\hat{\mathbf{h}}$ is also determined by the number of hidden nodes used by the ANN. The output vector $\hat{\mathbf{y}}$ is the actual ANN output at the present stage and has a fixed dimension.

The weights connecting the new node are assigned very small random values so that the addition of this node does not disrupt the network performance much. Training is resumed and the network soon reaches another plateau when another node is added to the layer with the lower information theoretic association with the ANN output. This process is continued until the network reaches a satisfactory level of performance as predetermined by the user.

Experience has shown that ANNs typically require more hidden nodes to learn a mapping than to recall the same mapping [22]. It is reasonable to assume that the ANN may also require more inputs to learn than to recall. Thus, it is quite possible that at this stage not all the hidden or input nodes may be necessary. To deal with this situation, we first look at the hidden nodes. The hidden node with the least importance is removed from the network. The importance of each of the hidden nodes can be determined by a method very similar to the one used to rank the available inputs in Phase I of DIS. But such a calculation is fairly computation intensive since the importance of all the hidden nodes needs to be determined every time a hidden node is to be eliminated. This can significantly slow down the training process. Instead, the importance function used here for the hidden nodes is the derivative importance function used in the DNA scheme from the authors' previous work [1, 22]. The derivative importance measure works well and is computationally inexpensive. It assumes that if changes in the output of a hidden node are more influential in deciding the output of the network than a similar change in the output

of another hidden node, then the former node is more important to the dynamic functioning of the network than the latter node. The hidden node with the least importance, as defined by the derivative importance function, is eliminated from the network.

Following the elimination of a node, the ANN might require further training, depending on the deleted node's importance to the ANN function. Upon continued training, the smaller network might be able to learn the mapping. If it does, then again the hidden node with the least importance is deleted. This process is continued until the network is too small to learn the problem. The smallest architecture capable of learning the problem is taken as the model from this phase of DIS.

Phase III : Removal of Unnecessary Inputs Phase III deals with the possibility that the ANN has used some input variables during the training phase that are not important. It is desirable to eliminate such inputs as it could provide a more compact model. For this purpose, we perform a Phase I style ranking of the variables actually being used by the network with respect to the actual ANN output vector $\dot{\mathbf{y}}$. The ranking obtained might place the variables used by the network in a different order than the initial ranking in Phase I. This is a reflection of the mapping learned by the network. The input variable with the lowest relative importance is eliminated, and the resultant network is trained further if necessary. It is possible that this network is unable to reduce its error below the preselected target, in which case the network from the end of Phase II is retained as the final model. If however the ANN is able to learn the mapping with one input less than in Phase II, we proceed to use the derivative importance function on the hidden layer to determine if any of

the hidden nodes can be eliminated.

The input variables now remaining in the ANN are re-ranked using PCA and ITIA with respect to the actual ANN output, and the least important variable is again eliminated. The whole process described in the previous paragraph is repeated, and either a new model is achieved or the previous model is retained. The algorithm is stopped when a successful model is not found following the deletion of an input node. In the process of reducing the number of inputs, it is a good practice to test a model on a validation data set, and accept it only if its generalization performance exceeds that of the previous model. This is important as the elimination of an input variable is not guaranteed to provide better generalization. On the other hand we do not validate a model while removing only the hidden nodes as previous work has shown the capability of ANNs to generalize more effectively with lesser hidden nodes [22, 35, 36]. Note that the final architecture arrived at by the DIS scheme is influenced by the selection of various training parameters such as the learning rate, the random weights used in the starting architecture, the pre-selected learning target and the number of iterations spent on a learning plateau before a node is added.

2.2 DIS Training: An Example

For a demonstration of the DIS scheme, we use the exclusive-or problem. The data given in Table 3.1 is used for training. The problem is defined by input 1, input 2 and the output as shown in the table. Input 3 and input 4 are spuriously generated data unimportant to the output value. Input 3 is a constant and input 4 is a random variable. This example is significant as a direct information theoretic interdependency analysis on the individual inputs would assign zero importance to

input 1 and input 2 while assigning a greater importance to the random variable, input 4. Clearly, such a ranking is inappropriate.

Phase I of the DIS scheme is the statistical ranking of the variables using PCA and ITIA. The results of this analysis can be seen in Table 3.2. The input variable ranking analysis shows that input 2 has the highest relative importance followed closely by input 1. Ideally, input 1 and input 2 should have equal importance, but the slight difference seen in Table 3.2 is due to machine round off error during the numerical computations. Input 4 ,which is a random variable, ranks third with a lower relative importance. Also, the table shows that input 3, a constant, has no importance to determining the output. Based on our knowledge of the problem, we see that the results of the ranking analysis are appropriate.

In Phase II, an ANN is trained on the data. The preset training target for the ANN was an RMS error of 0.01 over the training data. The training history of this ANN can be seen in Table 3.3. The network starts with one input (input 2) and one hidden node. This is not sufficient to provide the desired mapping, so the ANN reaches a learning plateau and another node needs to be added. An information theoretic interdependency analysis shows that the input layer is better associated than the hidden layer with the output layer. Thus, a second node is added to the information deficient hidden layer. The addition of this node disrupts the network and increases the RMS error slightly. Continued training of this one input, two hidden, one output (1 x 2 x 1) network does not decrease the RMS error significantly, and a node is added to the input layer on the basis of ITIA of the hidden and input layers. The new input is input 1 since it is the second most important variable (see Table 3.2). The ANN is now able to significantly reduce its RMS error before reaching another

plateau. A hidden node is added, and the 2 x 3 x 1 network reaches a plateau short of the target RMS error. Analysis of this network suggests the addition of another hidden node. Upon continued training, the 2 x 4 x 1 network is able to reach an RMS error below the preset target of 0.01. The derivative importance of each of the four hidden nodes is now calculated. The node with the lowest importance is deleted, and the resultant 2 x 3 x 1 network is trained until it can learn the function. Again, the hidden node with the least derivative importance is deleted from the network, and the 2 x 2 x 1 network is further trained. This network is also able to learn the functional mapping and another node is removed. The 2 x 1 x 1 network is incapable of learning the function and a second node is added. The process of adding and deleting nodes is continued until it is apparent that 2 x 2 x 1 is the optimum architecture at the end of this phase of DIS. Phase III, wherein any unnecessary inputs are removed, is not attempted with this model as we know that the two inputs used by the present model are the appropriate ones and eliminating any of them will render the network incapable of solving the problem.

3. DEVELOPMENT OF THE POWER PLANT ADVISER

This section describes the development of the ANN-based nuclear power plant fault diagnostic adviser. In order to be useful, any adviser should be capable of detecting and classifying a large variety of transients and therefore needs to draw information from many plant variables. The choice of transients and variables is therefore very important. Two documents were consulted for the purpose: the *Updated Final Safety Analysis Report* [37] and the *Malfunction Cause and Effects Report* [38], both pertaining to the Duane Arnold Energy Center (DAEC). These documents

describe many power plant transients. Discussions between personnel at DAEC and fellow researchers at Iowa State University (ISU) [39] resulted in a preliminary list of transients to be simulated on the operator training simulator and plant variables to be monitored [26]. Thirty of these transients, some of which can occur at varying severities, were selected for the development of the initial adviser. This adviser shall be referred to as Adviser 2a. The simulation of some transients at different severities enabled the adviser to detect a transient irrespective of its intensity. The training data for Adviser 2a therefore consisted of 47 scenarios representing the 30 distinct transients. This adviser was later expanded to detect six more transients and we shall refer to this expanded adviser as Adviser 2b. The training data for Adviser 2b consisted of 58 scenarios representing 36 distinct transients.

3.1 Data Collection and Processing

The raw data were obtained from the DAEC operators' training simulator. This data consisted of the numerical values of 97 plant variables at intervals of one second. The data also included a Yes/No binary switch that indicated the onset of the transient. The variables were selected from the complete list of 2,369 variables available on the simulator. These 97 variables were judged to be sufficient for an operator to diagnose the transients being investigated [39] and were therefore selected as the inputs for the proposed adviser. These variables covered a wide variety of plant instrumentation such as pressures and temperatures in the various systems of the plant, radiation monitors, and flow meters. Table 3.4 contains a complete list of the variable used in this work. Raw data obtained from DAEC were reformatted and normalized in the range zero to one. Normalization was based on the maximum and

minimum possible values of the variables.

3.2 Structure of the Adviser

The largest adviser developed in our earlier work (Adviser 1) [1] utilized a hierarchy of two networks. The structure of that adviser can be seen in Figure 3.1. The root network determined if the plant was in a normal condition or not, and the classifier network identified the particular transient in progress. The values of the 97 chosen variables at any single instant of time was assumed to contain enough information to diagnose the plant status. Both the networks used all the 97 variables as inputs. This information proved to be adequate to solve the problem.

The adviser(s) developed in this work are a step further in the evolution of the hierarchical concept initially demonstrated during the development of Adviser 1. The structure common to both Adviser 2a and Adviser 2b can be seen in Figure 3.2. The root network in this design has the same function as in Adviser 1, namely to detect an abnormal condition in the plant. The classification task is distributed among different ANNs that form the classifier group. Each ANN in the classifier group recognizes one transient. Therefore, Adviser 2a has 30 networks in the classifier group to identify the 30 transients it is expected to identify. Similarly, Adviser 2b has 36 ANNs in the classifier group. Since the DIS scheme is used to train the networks in these advisers, each network uses a subset of the 97 available variables. Thus the number of input variables vary among the networks, but each network has only one output node. A listing of the transients and scenarios used in this work can be found in Table 3.5. Transients 1 through 30 were used to develop Adviser 2a and transients 31 through 36 were added to develop Adviser 2b.

3.3 Development of Adviser 2a

The first phase in DIS training requires us to rank all the available inputs using a combination of PCA and ITIA. In this particular case, we need to perform this analysis for each of the individual networks. The simulation of the 47 scenarios used for developing Adviser 2a had a total of 11,792 data patterns, each containing the values of 97 instrumentation readings. We assigned an output to each of the networks according to the task of the network. For the root network, all abnormal patterns were assigned an output of 1 and all normal patterns were assigned an output of 0. A phase I analysis using all the 11,792 data patterns gave us a ranking for the 97 variables. Of the 11,792 data patterns, 376 corresponded to normal operating conditions, and were not relevant in the context of the ANNs in the classifier group. These ANNs would be used to recall only those patterns that have been identified as abnormal by the root network. The data set for the ANNs in the classifier group thus consisted of 11,416 patterns. For each of the classifier ANNs, these patterns were assigned an output of 0 or 1 depending on the particular transient that the given ANN had to recognize. Patterns corresponding to those scenarios were assigned an output of 1, and the patterns corresponding to all the other simulations were assigned an output of 0. A Phase I ranking analysis was performed for each of the network using the 11,416 patterns and a ranking obtained in each case.

The training data for all the networks were chosen in an iterative manner. For the first trial of the root network, one pattern at the beginning and one at the end of each of the 47 simulations were taken to form the training set. The initial training set therefore contained 94 patterns. Training was initiated with one hidden node and the highest ranked input. The target RMS error was 0.10. As the training progressed.

the DIS scheme added more inputs and hidden nodes and gave a final architecture of 8 x 16 x 1. This network was used to recall all the patterns over the entire length of the 47 simulations. The network, as expected, did not do a very good job of classifying all the patterns. The patterns with the worst recall errors were added to the training set and the network from the previous trial was trained further. This process of adding patterns to the training set was continued until the network could detect the onset of all the transients within a reasonable amount of time. The final architecture of the root network was 16 x 21 x 1.

A similar approach was taken to train all the networks in the classifier group. In this case, the networks were tested on only the patterns corresponding to abnormal plant conditions. Thus, each network had only one pattern from the end of each simulation during its first training attempt. Each network grew in size and improved its performance over successive training cycles. The networks ended up with different subsets of the available 97 variables as inputs. The input layer size ranged from 3 to 40. All the networks were trained to perform their individual classification task within 100 seconds of initiation of the transients.

All the networks were put together to form the diagnostic system. When the root network diagnosed a data pattern to be normal, nothing further happened. But as soon as the root network detected an abnormal pattern, it triggered on the networks in the classifier group. All of these networks attempted to classify the abnormal condition. The classification ability of the adviser is dependent on the outputs of all the 30 networks in the classifier layer. A definite classification is made when one of the networks gives an output of 1 and all the others give an output of 0. Thus, the speed of classification is controlled by the worst performing network in the classifier group.

But all the ANNs had been trained to perform their tasks within 100 seconds since the initiation of the transient. The adviser as a whole can thus arrive at the correct diagnosis within 100 seconds. Instead of having the adviser react to the output of the 30 networks at every one second interval, the adviser was designed to report only those trends that lasted for at least five seconds. Thus, the adviser would change an existing diagnosis only if it observes the new diagnosis consistently over five seconds, thus preventing it from making rapid changes in its diagnosis. This measure also adds robustness to the design of the adviser.

3.4 Expansion of Adviser 2a to Adviser 2b

We now decided to increase the capabilities to the adviser by modifying it to detect and classify six additional transients, using data from 11 simulations. The first step in this expansion is to account for the additional data represented by the 11 new simulations. The cumulative data now stood at 14,176 patterns, of which 460 patterns corresponded to normal operating conditions. The input variable ranking performed earlier did not take into account this new data, and is thus not valid any more. However, the existing networks are already trained using the previous ranking, and it is necessary to preserve these networks. So, we perform input variable ranking for each existing network using the complete set of data, and preserve the ranking of all the variables not already used by the network. We will also need to add six new networks to the classifier group, and we perform the input variable ranking for these networks.

The new ANNs in the classifier group are trained in a process similar to the training of the classifier group networks during the development of Adviser 2a. But

instead of 47 scenarios, these networks are tested on the new set of 58 scenarios. These networks are repeatedly trained until they can perform their task within the target of 100 seconds since the initiation of the transients.

At this stage, we shift our attention to the ANNs which constituted the original Adviser 2a. We test these networks on all the 58 simulations. Obviously, these networks will perform satisfactorily on the 47 scenarios that were originally used to develop Adviser 2a. Their performance on the 11 new scenarios might not be adequate, and they need to be further trained. Once these networks are trained on additional data, they need to be tested on all the 58 scenarios to make sure that their performance on the previous 47 scenarios did not deteriorate. This process is continued until the networks from Adviser 2a have learn their additional task. In the present work, none of the pre-existing networks needed any additional inputs to adapt to their expanded task. In fact, the root network was the only network that needed any significant retraining. The modified root network, the 30 networks retained and modified from the previous adviser, and the six new classifier networks are put together to form Adviser 2b.

4. RESULTS

The performance of Adviser 2a and Adviser 2b are tabulated in Table 3.6. The pure simulator data was corrupted by 3% uniform noise, and the advisers were tested on this noisy data. Results from these can also be seen in Table 3.6. It is evident from the table that the diagnostic capability of the advisers did not undergo any significant degradation under noisy conditions. This can be directly attributed to the "five-second observation" feature incorporated in the adviser.

The performance of Adviser 2b is comparable to that of Adviser 2a. The advantage of the modular design can be best appreciated by reflecting on the expansion of the adviser if it were based on the design that produced Adviser 1 [1]. The 30 transients would have been trained using a network with five output nodes. But this network would be unable to accommodate the six new networks. So, the existing classifier network would have to be discarded and a whole new classifier network trained. However, the root network could have been modified exactly in the same way as with the modular design.

Figures 3.3 and 3.4 graphically represent the performance of Adviser 2a on a selected transient. The transient in this case is Spurious Group 7 Isolation (ms32). The top half of Figure 3.3 displays the performance of the root network on pure simulator data, while the top half of Figure 3.4 shows the performance of this same network when it is tested on the pure data corrupted by 3% uniform noise. The bottom half of the two figures show the classification performed by the ANNs in the classifier group. "Normal" means that all the networks are showing an output of 0, which "Uncertain" means that ether some of the networks have an output in the intermediate range of 0.3 to 0.7 or more than one network has an output of 1. A "Wrong Diagnosis" means that only one network has an output of 1 and this network is not the ANN corresponding to the transient in progress. This kind of incorrect diagnosis would require the network corresponding to the transient in progress to have an output of 0, and one of the other networks to have an output of 1, while the remaining networks output 0. And this situation should be maintained for a minimum of five seconds. In this adviser structure, when the ANN corresponding to the transient in progress is giving an output of 0 instead of 1, the remaining

networks are usually all stable with outputs of 0 ("Normal") or in a very unstable state, their outputs changing every second. Thus the adviser rarely settles into a "Wrong Diagnosis." Such incorrect classification were detected in four instances over the duration of the fifty-eight transient scenarios. Finally, a "Correct Diagnosis" is obviously when the only network showing an output of 1 is the ANN corresponding to the transient in progress.

The utility of the advisers developed here can be best understood by looking at certain examples from Table 3.6. Transient scenario "ms32" (transient number 24) is the spurious group 7 isolation in the plant. This transient caused the plant to undergo a safety trip 98 seconds after the initiating event. The root network (in Adviser 2a) was able to detect an abnormality in the plant status 11 seconds after the initiating event. This ability was not hampered even when the input data was corrupted by noise. The classifier networks diagnosed the transient 74 seconds after its advent. Under noisy condition, successful diagnosis was achieved at 80 seconds. Table 3.6 also shows that Adviser 2b had a similar performance for this transient. The diagnosis might be able to indicate to the operators the cause of the transient, and that a safety trip was imminent. The operators could then take measures to shut down the plant, thus sparing the plant from an abrupt and stressful safety trip. A controlled shutdown has a much lower impact on the plant components.

There are many transients that occur over a prolonged period of time and do not cause a safety trip. An example is transient "rx01" (transient number 30). This transient is caused by a 5% failure of the fuel cladding, resulting in radioactive contamination of the coolant system. The root network detects an abnormality within 10 seconds after the initiating event, and the classifier network diagnoses the transient

around the 32 seconds mark. During this time the operators will be undertaking tasks mandated by the plant operating procedures. A definite diagnosis by the adviser in such a situation would help the shift technical adviser assess the situation and judge the validity and applicability of the procedures being performed. Although the procedures are quite extensive, it is conceivable that the plant might enter a situation not foreseen. If the shift technical adviser feels that information provided by the diagnostic adviser conflicts with the procedures being carried out, he may take appropriate action.

The use of DIS significantly decreased training time, and allowed the networks to use only those variables that they deemed necessary to perform their task. Table 3.7 shows the architecture of the 37 networks that form Adviser 2b, and the input variables used by the networks. The separation of the classification problem into the modular components allowed the various networks to be trained on different computers, thus quickening the process of developing the adviser. However, one possible drawback of this approach is the need to test the performance of all the networks on all the simulations during each training cycle, and adding new data patterns to the training data set. This can be overwhelming while developing many networks simultaneously, but this inconvenience is more than compensated by the fact that these networks can be retained in future upgradations of the adviser. Thus, the process of adding a few transients to the diagnostic capability of the existing adviser is a much more manageable task than it would have been without the modular design.

5. CONCLUSIONS

The first conclusion from this work is the ability to detect a very wide variety of operational transients at a boiling water reactor (BWR) nuclear power plant using multiple neural networks. An ANN-based adviser was successfully designed and trained to detect and classify 30 distinct transients and the five normal conditions. This adviser consisted of a total of 31 networks working in a hierarchical manner. The adviser performed well even when the data was corrupted by noise. The diagnosis of the adviser was a result of the outputs of all the individual ANNs. Even though the ANNs based their outputs on the data at one second intervals, the adviser made its decisions based on trends in the output of the ANNs. The adviser changed its diagnosis only when a new diagnosis was detected consistently for five seconds. This provided a measure of stability to the adviser, and was responsible for its admirable performance under noisy conditions.

The second conclusion from this work is the ability of the modular design to accommodate an increase in the diagnostics capabilities of an existing adviser. The adviser capable of detecting and classifying 30 transients was expanded to diagnose six more transients. The expansion process involved the training of six new networks, and minor retraining of the 31 existing networks. This expansion was performed without sacrificing the performance of the existing networks.

The third conclusion is the use of DIS allowed the speedy development of the adviser by automating the architecture selection process for all the networks. The dynamic growth of the input and hidden layers allowed the networks to be developed without any guesswork. The dimensionality reduction derived by the process helped in decreasing the training time required for all the networks. Successful application

of DIS in this problem demonstrated the advantages of the scheme in developing ANN-based solutions for large real world problems.

BIBLIOGRAPHY

[1] A. Basu and E. B. Bartlett, "Detecting Faults in a Nuclear Power Plant by Using Dynamic Node Architecture Artificial Neural Networks," *Nuclear Science and Engineering*, vol. 116, pp. 313-325, 1994.

[2] Keehoon Kim and Eric B. Bartlett, "Error prediction for a nuclear power plant fault-diagnostic advisor using neural networks," *Nuclear Technology*, vol. 108, November 1994, pp. 283-297.

[3] E. B. Bartlett and R. E. Uhrig, "Nuclear Power Plant Status Diagnostics Using Artificial Neural Networks," *Nucl. Technol.*, vol. 97, pp. 272-281, March, 1992.

[4] E. B. Bartlett and R. E. Uhrig, "Nuclear Power Plant Status Diagnostics Using Artificial Neural Networks," *Proc. American Nuclear Society Meeting on Frontiers in Innovative Computing for the Nuclear Industry*, Jackson Lake, Wyoming, September 15-18, 1991, p. 644.

[5] Y. Ohga, and H. Seki, "Abnormal event identification in nuclear power plants using a neural network and knowledge processing," *Nuclear Technology*, vol. 101, pp. 159-167, 1993.

[6] S. W. Cheon, and S. H. Chang, "Application of neural networks to a connectionist expert system for transient identification in nuclear power plants," *Nuclear Technology*, vol. 102, pp. 177-191, 1993.

[7] A. G. Parlos, J. Muthusami, and A. F. Atiya, "Incipient fault detection and identification in process systems using accelerated neural network learning," *Nuclear Technology*, vol. 105, pp. 145-161, 1994.

[8] A. Ikonomopoulos, L. H. Tsoukalas and R. E. Uhrig, "Monitoring nuclear reactor systems using neural networks and fuzzy logic," *Proc. Top. Mtg. Advances in Reactor Physics*, Charlston, South Carolina, March 1992, pp.2-14.

[9] Z. Guo and R. E. Uhrig, "Use of artificial neural networks to analyze nuclear power plant performance," *Nuclear Technology*, vol. 99, pp. 36-42, 1992.

[10] I. E. Alguindigue, R. E. Uhrig, M. Cai, and A. Trendy, "Discrimination of ex-core neutron noise signature using artificial neural networks," *Proc. of the American Power Conference*, Chicago, IL, April 12-14, 1993.

[11] B. R. Upadhyaya and E. Eryurek, "Application of Neural Networks for Sensor Validation and Plant Monitoring," *Nuclear Technology*, vol. 97, pp. 170-176, February, 1992.

[12] M. Roh, S. Cheon, and S. Chang, "Thermal power prediction of nuclear power plant using neural network and parity space model," *IEEE Transactions on Nuclear Science*, vol. 38, no. 2, pp. 866-872, April, 1991.

[13] H. Kim and S. Lee, "Neural network model for estimating departure from nuclear boiling performance of a pressurized water reactor core," *Nuclear Technology*, vol. 101, pp. 111-122, 1993.

[14] H. G. Kim, S. H. Chang, and B. H. Lee, "Pressurized water reactor core parameter prediction using an artificial neural network," *Nuclear Science and Engineering*, vol. 113, pp. 70-76, 1993.

[15] A. Basu, "Nuclear power plant status diagnostics using a neural network with dynamic node architecture," MS Thesis, Iowa State University (1992).

[16] J. S. Judd, *Neural Network Design and the Complexity of Learning*, The MIT Press, Cambridge, Mass., (1990).

[17] H. S. Wilf, *Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, New Jersey, (1986).

[18] A. Basu and E. B. Bartlett, "A Dynamic Inputs Selection Scheme for Artificial Neural Networks," Under review at *IEEE Transactions on Neural Networks.*

[19] R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE Acoustics Speech and Signal Processing Magazine*, 4, 4 (April 1987).

[20] M. Caudill, "Neural Networks Primer, Part 3," *AI Expert*, 53 (June 1988).

[21] E. B. Bartlett, "Self Determination of Input Variable Importance Using Neural Networks," *Neural, Parallel & Scientific Computations*, **2**, 103 (1994).

[22] E. B. Bartlett and A. Basu, "A Dynamic Node Architecture Scheme for Backpropagation Neural Networks," *Intelligent Engineering Systems Through Artificial Neural Networks*, p. 101, C. H. Dagli, S. R. T. Kumara, and Y. C. Shin, Eds., ASME Press, New York (1991).

[23] D. F. Morrison, *Applied Linear Statistical Methods*, Prentice-Hall, Englewood Cliffs, N.J. (1983).

[24] S. J. Press, *Applied Multivariate Analysis*, Holt, Rinehart and Winston, New York, (1972).

[25] Y. Isogawa, *Comparing Multivariate Linear Functional Relationships*, Kobe University of Commerce, (1991).

[26] T. L. LANC, "The Importance of Input Variables to a Neural Network Fault-Diagnostic System for Nuclear Power Plants," MS Thesis, Iowa State University (1991).

[27] A. Papoulis, *Probability, random variables, and Stochastic Processes*, McGraw-Hill Book Company, New York (1965).

[28] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, Ill. (1971).

[29] S. Watanabe, *Knowing and Guessing: A Quantitative Study of Inference and Information*, John Wiley and Sons, New York (1969).

[30] R. B. Ash, *Information Theory*, Dover Publications, New York (1990).

[31] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York (1986).

[32] G. H. Dunteman, *Principal Components Analysis*, Sage, Newbury Park, Calif., (1989).

[33] A. N. Kshirsagar, *Multivariate Analysis*, Marcel Dekker, New York (1972).

[34] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipies*, Cambridge University Press, Cambridge (1986).

[35] Y. Hirose, K. Yamashita, and S. Hijiya, "Back Propagation Algorithm That Varies the Number of Hidden Nodes," *Neural Networks*. 4, 61 (1991).

[36] E. B. Bartlett, "Dynamic Node Architecture Learning: An Information Theoretic Approach," *Neural Networks*, 7, 129 (1994).

[37] "Chapter 15: Accident Analysis" in *Updated Final Safety Analysis Report*, Vol. XI, Duane Arnold Energy Center, Iowa Electric Light and Power Company, Palo, Iowa (1984).

[38] J. Gould, *Malfunction Cause and Effects Report*, Task no. 06000004, Duane Arnold Energy Center, Iowa Electric Light and Power Company, Palo, Iowa (1991).

[39] D. Vest, C. Hunt, and D. Berchenbriter, Personal discussions and correspondence with Duane Arnold Energy Center simulator complex employees. Iowa Electric Light and Power Company, Palo, Iowa (1991-1992).

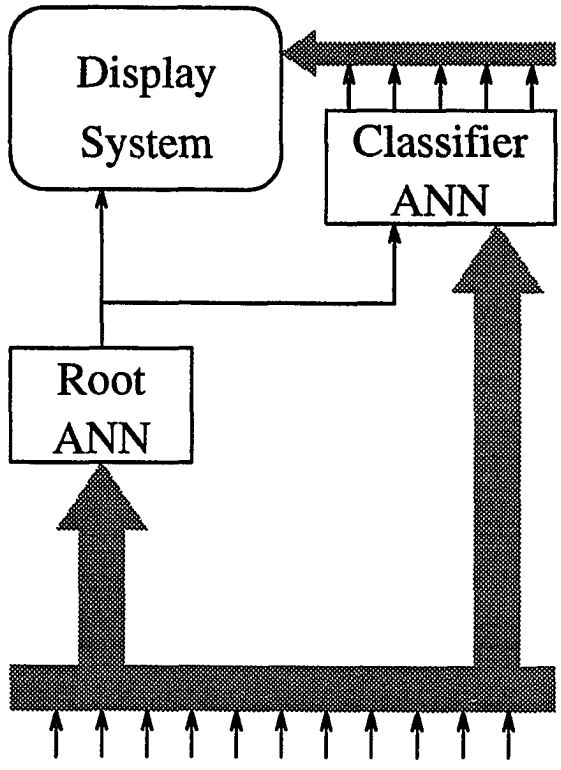

Figure 3.1: Structure of Adviser 1.

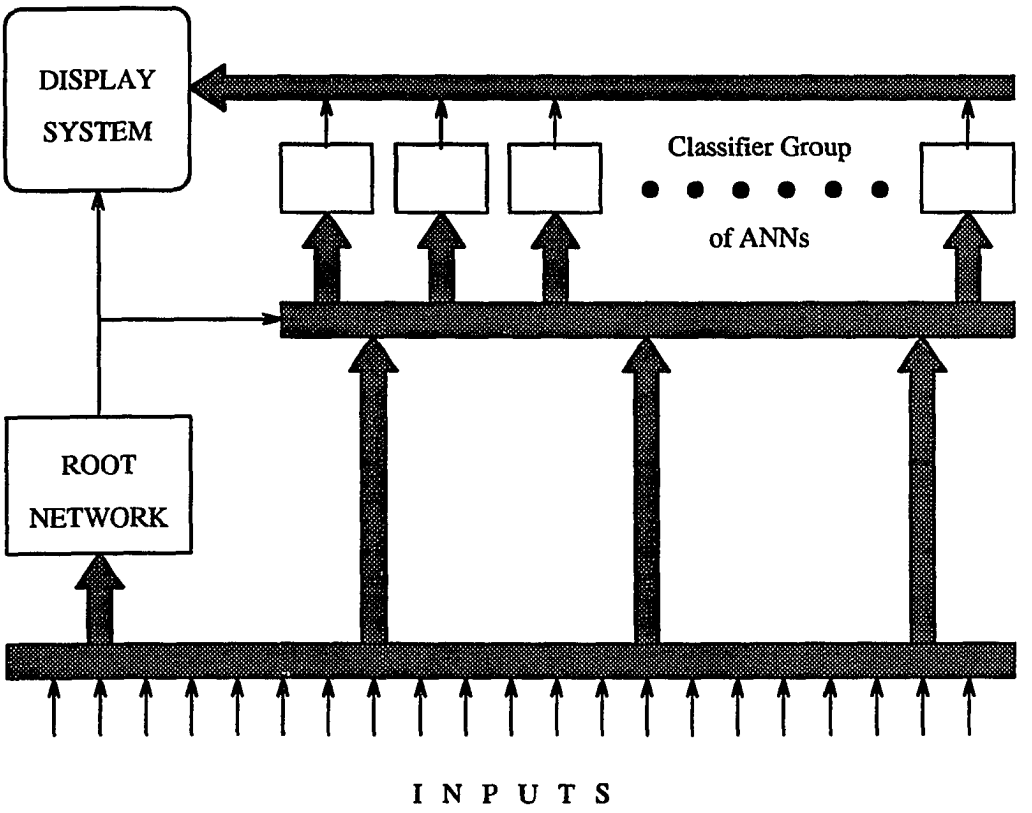

Figure 3.2: Structure of Adviser 2.

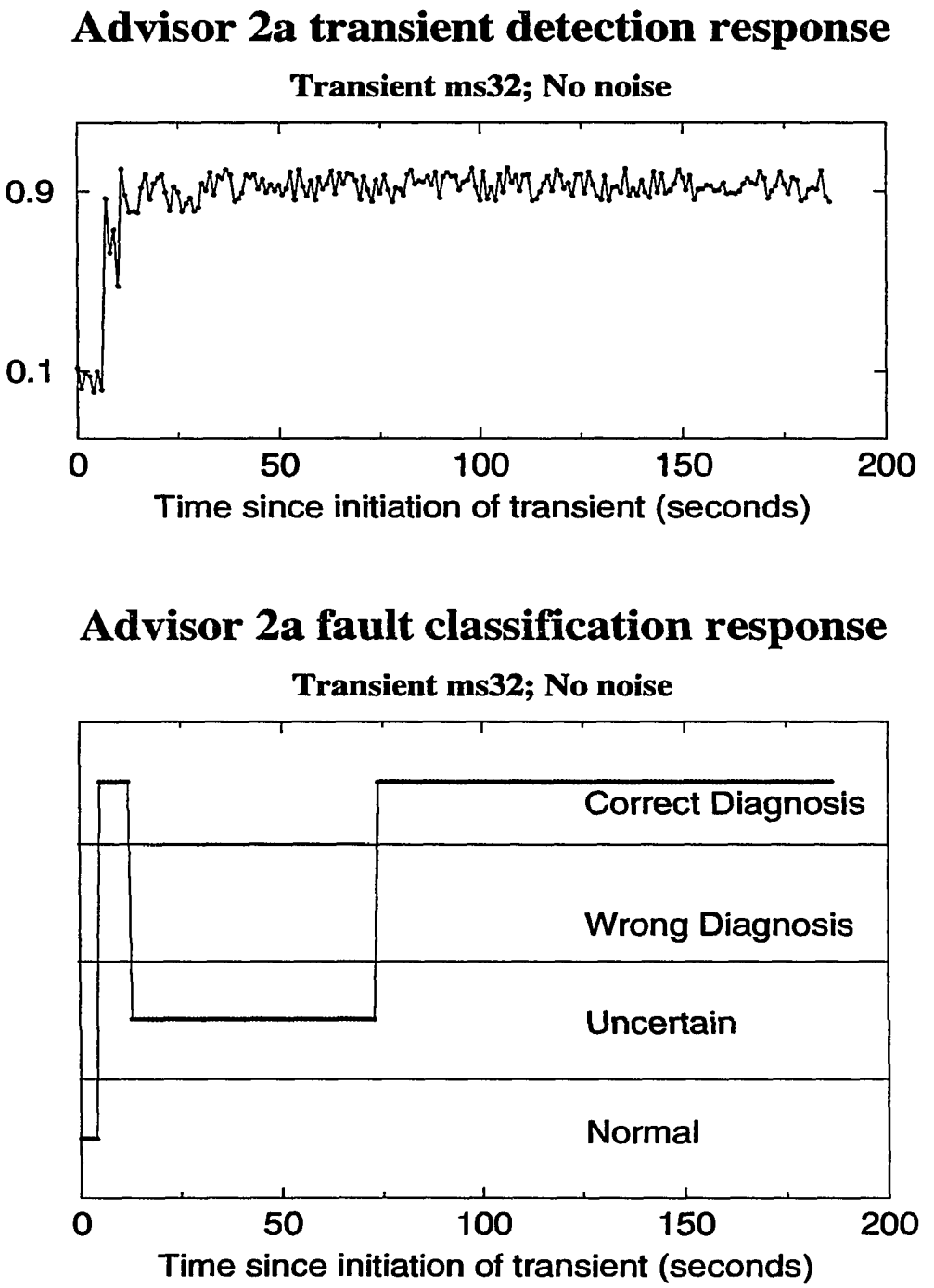

Figure 3.3: Performance of Adviser 2a during spurious group 7 isolation without any noise.

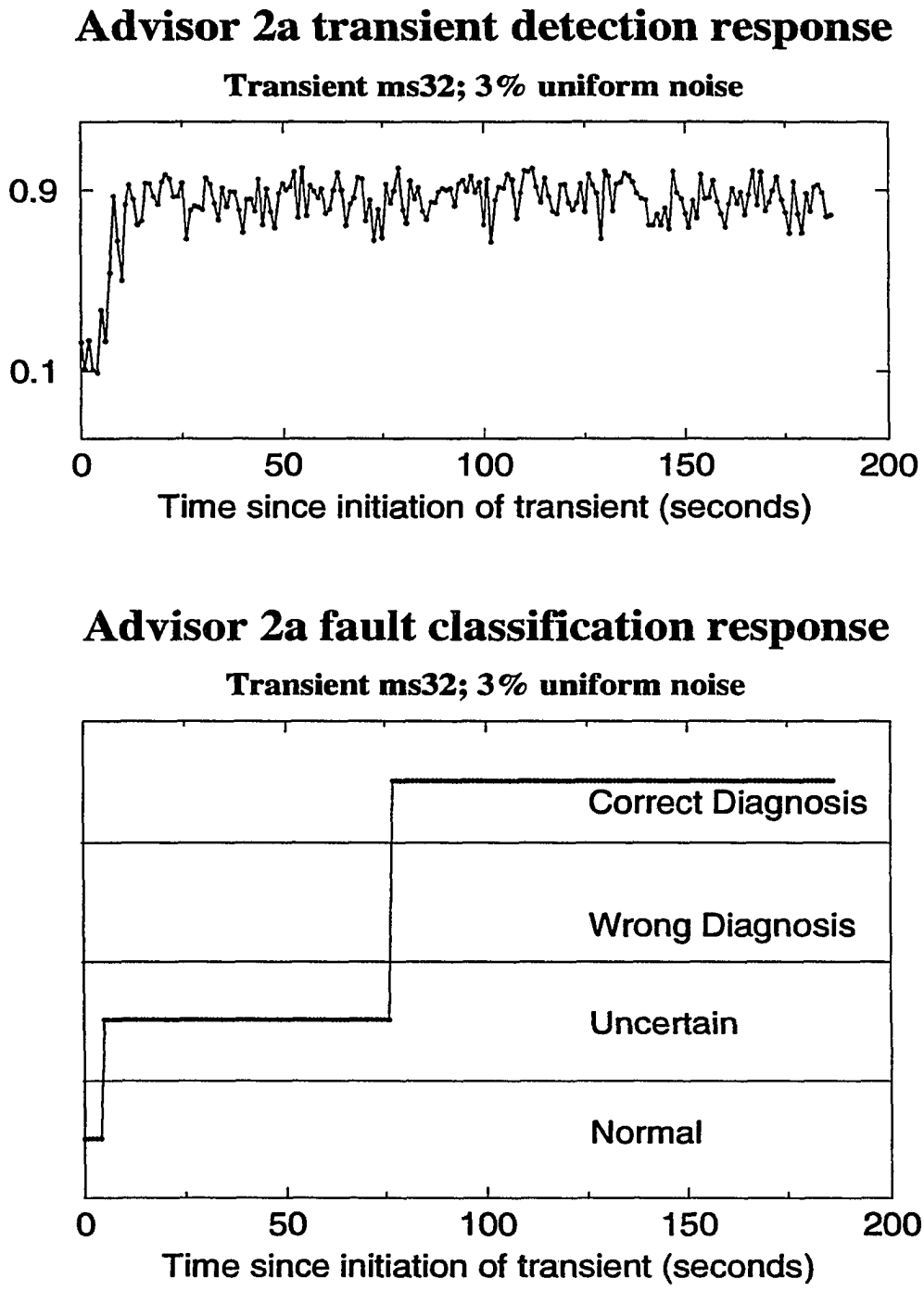

Figure 3.4: Performance of Adviser 2a during spurious group 7 isolation with 3% uniform noise.

Table 3.1: Exclusive-or training data

Pattern	Input1 (switch)	Input2 (switch)	Input3 (constant)	Input4 (random)	Output
1	0.0	0.0	0.5	0.338	0.0
2	0.0	1.0	0.5	0.417	1.0
3	1.0	0.0	0.5	0.793	1.0
4	1.0	1.0	0.5	0.981	0.0
5	0.0	0.0	0.5	0.144	0.0
6	0.0	1.0	0.5	0.583	1.0
7	1.0	0.0	0.5	0.891	1.0
8	1.0	1.0	0.5	0.510	0.0

Table 3.2: Phase I ranking analysis of exclusive-or training data

Rank	Relative Importance	Variable Number
1	0.319	2
2	0.318	1
3	0.117	4
4	0.000	3

Table 3.3: Phase II training history for exclusive-or problem

Architecture			RMS	Architecture			RMS
Input	Hidden	Output	Error	Input	Hidden	Output	Error
Start: Input2 & 1 hidden node				Delete hidden node			
1	1	1	0.5842	2	2	1	0.0417
1	1	1	0.4919	2	2	1	0.0092
Add hidden node				Delete hidden node			
1	2	1	0.5528	2	1	1	0.3718
1	2	1	0.4692	2	1	1	0.3373
Add Input1				Add hidden node			
2	2	1	0.4921	2	2	1	0.3136
2	2	1	0.1822	2	2	1	0.0181
Add hidden node				Add hidden node			
2	3	1	0.2011	2	3	1	0.0201
2	3	1	0.0813	2	3	1	0.0098
Add hidden node				Delete hidden node			
2	4	1	0.0886	2	2	1	0.0318
2	4	1	0.0097	2	2	1	0.0096
Delete hidden node				Delete hidden node			
2	3	1	0.0366	2	1	1	0.4181
2	3	1	0.0098	2	1	1	0.3117

Table 3.4: Plant variables available for training the ANN adviser.

Var. Num.	Variable Desig.	Description	Min. Value	Max. Value	Var. Unit
1	A041	Local power range monitor 16-25 flux level B	0.0	125.0	% power
2	A091	Source range monitor channel B	0.0	100.0	%
3	B000	Average power range monitor A Flux level	0.0	125.0	% power
4	B012	Reactor total core flow	0.0	60.0	Mlb/hr
5	B013	Reactor core pressure-differential	0.0	30.0	psid
6	B014	Control rod drive system flow	0.0	0.025	Mlb/hr
7	B015	Reactor feedwater loop A flow	0.0	4.0	Mlb/hr
8	B016	Reactor feedwater loop B flow	0.0	4.0	Mlb/hr
9	B017	Cleanup system flow	0.0	0.07691	Mlb/hr
10	B022	Total steam flow	0.0	8.0	Mlb/hr
11	B023	Cleanup system inlet temperature	0.0	755.0	$^{\circ}$F
12	B024	Cleanup system outlet temperature	0.0	600.0	$^{\circ}$F
13	B026	Recirculation loop A1 drive flow	0.0	15.1	Mlb/hr
14	B028	Recirculation loop B1 drive flow	0.0	15.1	Mlb/hr
15	B030	Reactor feedwater channel A1 temperature	280.0	430.0	$^{\circ}$F
16	B032	Reactor feedwater channel B1 temperature	280.0	430.0	$^{\circ}$F
17	B034	Recirculation loop A1 inlet temperature	260.0	580.0	$^{\circ}$F
18	B036	Recirculation loop B1 inlet temperature	260.0	580.0	$^{\circ}$F
19	B038	Recirculation A wide range temperature	50.4	789.6	$^{\circ}$F
20	B039	Recirculation B wide range temperature	50.4	789.6	$^{\circ}$F
21	B061	Reactor coolant total jet pumps 1-8 flow B	0.0	36.7	Mlb/hr
22	B062	Reactor coolant total jet pumps 9-16 flow A	0.0	36.7	Mlb/hr
23	B063	Reactor coolant total outlet steam flow A	0.0	2.0	Mlb/hr

Table 3.4 (Continued)

Var. Num.	Variable Desig.	Description	Min. Value	Max. Value	Var. Unit
24	B064	Reactor coolant total outlet steam flow B	0.0	2.0	Mlb/hr
25	B065	Reactor coolant total outlet steam flow C	0.0	2.0	Mlb/hr
26	B066	Reactor coolant total outlet steam flow D	0.0	2.0	Mlb/hr
27	B079	Reactor recirculation pump A motor vibration	0.0	10.0	MILS
28	B080	Reactor recirculation pump B motor vibration	0.0	10.0	MILS
29	B083	Control rod drive cooling-water differential pressure	0.0	500.0	dpsi
30	B084	Control rod drive cooling-water differential pressure	0.0	60.0	dpsi
31	B085	Torus air temperature #1	0.0	500.0	°F
32	B086	Torus air temperature #2	0.0	500.0	°F
33	B087	Torus air temperature #3	0.0	500.0	°F
34	B088	Torus air temperature #4	0.0	500.0	°F
35	B089	Drywell temperature azmuth 0 elevation 750	0.0	500.0	°F
36	B090	Drywell temperature azmuth 245 elevation 750	0.0	500.0	°F
37	B091	Drywell temperature azmuth 90 elevation 765	0.0	500.0	°F
38	B092	Drywell temperature azmuth 270 elevation 765	0.0	500.0	°F
39	B093	Drywell temperature azmuth 270 elevation 765	0.0	500.0	°F
40	B094	Drywell temperature azmuth 180 elevation 780	0.0	500.0	°F
41	B095	Drywell temperature azmuth 270 elevation 830	0.0	500.0	°F
42	B096	Drywell temperature center elevation 750	0.0	500.0	°F
43	B098	Torus water temperature	0.0	752.0	°F

Table 3.4 (Continued)

Var. Num.	Variable Desig.	Description	Min. Value	Max. Value	Var. Unit
44	B099	Torus water temperature	0.0	752.0	°F
45	B103	Drywell pressure	0.0	100.0	psia
46	B104	Torus pressure	0.0	100.0	psia
47	B105	Torus water level	-10.0	10.0	in.
48	B120	Torus radiation monitor A	-1.0	100.0	%
49	B121	Torus radiation monitor B	-1.0	100.0	%
50	B122	Reactor water level	158.0	218.0	in.
51	B124	Reactor water level	158.0	218.0	in.
52	B1257	Fuel zone level indication	-153.0	218.0	in.
53	B126	Reactor water level	158.0	458.0	in.
54	B127	Reactor vessel pressure	0.0	1200.0	psig
55	B128	Reactor vessel pressure	0.0	1200.0	psig
56	B129	Reactor vessel pressure	0.0	1500.0	psig
57	B130	Reactor vessel pressure	0.0	1500.0	psig
58	B137	Torus water level	1.5	16.0	ft
59	B138	Torus water level	1.5	16.0	ft
60	B150	Core spray A flow	-1767.8	5000.0	gpm
61	B151	Core spray B flow	-1767.8	5000.0	gpm
62	B160	Reactor core isolation cooling flow	-62.5	500.0	gpm
63	B161	High-pressure core injection flow	-437.5	3500.0	gpm
64	B162	Residual heat removal A flow	-75.0	15000.0	gpm
65	B163	Residual heat removal B flow	-75.0	150.0	gpm
66	B164	Drywell radiation monitor A	-1.0	100.0	%
67	B165	Drywell radiation monitor B	0.0	100.0	%
68	B166	Post-treat activity	0.0	100.0	%
69	B168	Pretreat activity	0.0	100.0	%
70	B171	Analyzer A — O_2 concentration	-1.25	10.0	%
71	B172	Analyzer A — H_2 concentration	-1.25	10.0	%
72	B173	Analyzer B — O_2 concentration	-1.25	10.0	%
73	B174	Analyzer B — H_2 concentration	-1.25	10.0	%
74	B180	Clean-up system flow	0.0	200.0	gpm
75	B196	Reactor water level-fuel zone A	-153.0	218.0	in.
76	B197	Reactor water level-fuel zone B	-153.0	218.0	in.
77	B247	Turbine steam bypass	0.0	500.0	°F

Table 3.4 (Continued)

Var. Num.	Variable Desig.	Description	Min. Value	Max. Value	Var. Unit
78	B248	Turbine steam bypass	0.0	500.0	°F
79	E000	4160 V switch gear bus 1A1 A-B	0.0	5.25	KV
80	F004	Condensate pump A&B discharge pressure	0.0	600.0	psig
81	F005	Low-pressure condenser circulating water inlet temperature A	0.0	200.0	°F
82	F010	High-pressure condenser circulating water outlet temperature A	0.0	200.0	°F
83	F011	Low-pressure condenser circulating water pressure differential A	0.0	10.0	dpsi
84	F015	Circulating water pump A&B discharge pressure	0.0	100.0	psig
85	F018	Cooling tower A discharge water temperature	0.0	752.0	°F
86	F019	Cooling tower B discharge water temperature	0.0	752.0	°F
87	F040	1P-1A reactor feed pump suction pressure	0.0	600.0	psig
88	F041	1P-1B reactor feed pump suction pressure	0.0	600.0	psig
89	F042	1P-1A reactor feed pump discharge pressure	0.0	2000.0	psig
90	F043	1P-1B reactor feed pump discharge	0.0	2000.0	psig
91	F044	Condensate total flow	0.0	8.0	Mlb/hr
92	F045	Condensate makeup flow	-10.0	100.0	Klb/H
93	F046	Condensate rejection flow	0.0	50.0	Klb/H
94	F094	Feedwater final pressure	0.0	2000.0	psig
95	G001	Generator gross watts	0.0	720.0	MWE
96	T039	Low-pressure condenser pressure	0.0	30.0	in.-Hg
97	T040	High-pressure condenser pressure	0.0	30.0	in.-Hg

Table 3.5: The thirty-six transients, and the fifty-eight scenarios used to design Adviser 2a and Adviser 2b.

No.	Scenario	Description
1	cu10	Reactor water clean-up coolant leakage
2	cu10gp5	Reactor water clean-up coolant leakage with failure of Group 5 isolation valves.
3	fw02a	Condensate pump trip
4	fw04a	Condensate filter demineralizer resin injection
5		Feedwater tube leak inside heaters 6A & 6B
	fw08-6	- 100% severity
	fw08-6_2	- 60% severity
	fw08-6_3	- 30% severity
6	fw09a	Reactor feedwater pump trip
7	fw12c0	Feedwater regulator valve controller stuck closed
8	fw12c1	Feedwater regulator valve controller stuck open
9		Main feedwater line break inside primary containment
	fw17a	- 100% severity
	fw17a_2	- 60% severity
	fw17a_3	- 30% severity
10		High-pressure core injection (HPCI) steam supply line break in HPCI room
	hp05_2	- 60% severity
	hp05_3	- 30% severity
11		High-pressure core injection steam supply line break in torus room
	hp08_2	- 60% severity
	hp08_3	- 30% severity
12		Complete loss of instrumentation air
	ia01	- 100% severity
	ia01_2	- 60% severity
13	ic14scra	Spurious scram with no operator action. Initial condition IC14 : 100% power, Beginning of Cycle
14	ic20scr2	Spurious scram with effective operator action to avoid feedwater pump trip. Initial condition IC20 : 100% power. End of Cycle
15	ic20scrm	Spurious scram with no operator action. Initial condition IC20 : 100% power, End of Cycle.

Table 3.5 (Continued)

No.	Scenario	Description
16	ic22scra	Spurious scram. Initial condition IC22: 25% power, Beginning of Cycle
17	ic23scrm	Spurious scram. Initial condition IC23 : 75% power, Beginning of Cycle
18	ic24scrm	Spurious scram. Initial condition IC24 : 100% power, Middle of Cycle
19		Main condenser air inleakage
	mc04	- 100% severity
	mc04_2	- 60% severity
	mc04_3	- 30% severity
20		Steam leak inside primary containment
	ms02	- 100% severity
	ms02_2	- 60% severity
	ms02_3	- 30% severity
21		Main steam line rupture inside primary containment
	ms03a	- 100% severity
	ms03a_2	- 60% severity
	ms03a_3	- 30% severity
22	ms14-6	Loss of feedwater heating to feedwater heaters 6A & 6B
23	ms19ab	Spurious group 1 isolation
24	ms32	Spurious group 7 isolation
25	rp05tc01	Reactor protection system SCRAM circuit failure (ATWS) with alternate rod injection
26	rp5actc1	Reactor protection system SCRAM circuit failure (ATWS) with failure of alternate rod injection
27	rr10	Recirculation pump speed feedback signal failure
28		Recirculation loop rupture (design basis Loss of Coolant Accident)
	rr15a	- 100% severity
	rr15a_2	- 60% severity
	rr15a_3	- 30% severity
29		Coolant leakage inside primary containment
	rr30	- 100% severity
	rr30_2	- 60% severity
	rr30_3	- 30% severity

Table 3.5 (Continued)

No.	Scenario	Description
30	rx01	Fuel cladding (5%) failure
31	ad05	Inadvertent initiation of Automatic Depressurization System (ADS)
32		Main feedwater line break outside primary containment
	fw18a	- 100% severity
	fw18a_2	- 60% severity
	fw18a_3	- 30% severity
33	mc01a	Main circulation water pump "A" trip
34		Main steam line rupture outside primary containment
	ms04a	- 100% severity
	ms04a_2	- 60% severity
	ms04a_3	- 30% severity
35		Loss of air pressure to control rod drive (CRD) hydraulic control units (HCUs)
	rd13	- 100% severity
	rd13_2	- 60% severity
36	tc02	Electrical hydraulic control (EHC) system hydraulic pump failure

Table 3.6: Performance of Adviser 2a and Adviser 2b on pure and noisy data. All times are in seconds since the initiation of the transients.

Scenario	Scram Time	Adviser 2a				Adviser 2b			
		Pure data		Noisy data		Pure data		Noisy data	
		Root	Class.	Root	Class.	Root	Class.	Root	Class.
cu10	n/s	0	87	22	87	0	88	10	88
cu10gp5	44	1	95	19	96	1	95	6	94
fw02a	498	0	76	8	78	1	76	9	88
fw04a	n/s	1	30	13	31	1	32	12	32
fw08-6	22	0	60	5	72	2	60	8	62
fw08-6_2	n/s	0	86	1	88	0	82	6	91
fw08-6_3	n/s	1	94	9	92	0	89	9	94
fw09a	n/s	0	96	6	102	0	96	2	112
fw12c0	10	1	63	10	66	0	64	18	64
fw12c1	90	1	99	12	96	0	96	14	99
fw17a	1	0	81	2	80	2	81	5	107
fw17a_2	4	1	83	9	86	2	83	9	84
fw17a_3	8	1	65	12	72	1	65	6	72
hp05_2	n/s	1	85	6	80	1	80	3	87
hp05_3	n/s	1	84	4	90	1	86	4	85
hp08_2	n/s	1	99	1	99	1	102	3	118
hp08_3	n/s	1	34	2	72	1	42	3	66
ia01	n/s	0	23	1	24	0	23	12	23
ia01_2	n/s	0	58	3	58	0	52	7	52
ic14scra	1	2	93	17	92	1	92	9	101
ic20scr2	1	1	85	12	99	1	85	22	112
ic20scrm	1	1	79	4	82	1	79	4	80
ic22scra	1	18	85	16	102	12	85	13	89
ic23scrm	1	1	85	10	91	0	86	6	99
ic24scrm	1	2	81	12	80	2	80	7	82
mc04a	35	0	72	26	77	1	72	22	72
mc04a_2	47	0	82	24	95	4	80	18	79
mc04a_3	58	26	92	39	107	22	87	26	91
ms02	3	0	77	0	68	0	81	4	102
ms02_2	3	0	85	9	88	0	93	3	98
ms02_3	4	0	95	12	103	0	92	6	106

Table 3.6 (Continued)

Scenario	Scram Time	Adviser 2a				Adviser 2b			
		Pure data		Noisy data		Pure data		Noisy data	
		Root	Class.	Root	Class.	Root	Class.	Root	Class.
ms03a	2	1	92	7	92	2	87	18	87
ms03a_2	1	0	67	3	82	0	62	9	75
ms03a_3	2	0	82	3	89	0	89	12	89
ms14-6	n/s	1	87	7	85	2	87	3	85
ms19ab	17	17	79	32	80	20	78	24	78
ms32	98	11	73	12	77	16	77	18	78
rp05tc01	20	1	69	12	66	1	71	8	71
rp5actc1	n/s	1	64	18	52	1	60	14	60
rr10	n/s	11	91	29	109	12	91	18	111
rr15a	1	0	95	4	92	0	95	1	95
rr15a_2	1	2	91	12	99	2	91	6	92
rr15a_3	1	2	78	9	92	2	78	5	78
rr30	18	19	89	21	91	14	88	36	98
rr30_2	30	2	84	22	78	2	93	17	93
rr30_3	59	13	94	18	81	12	94	14	109
rx01	n/s	8	32	10	33	7	32	11	32
ad05	3	-	-	-	-	4	68	14	68
fw18a	8	-	-	-	-	8	61	24	66
fw18a_2	24	-	-	-	-	12	73	18	91
fw18a_3	n/s	-	-	-	-	12	68	15	71
mc01a	275	-	-	-	-	4	92	11	92
ms04a	2	-	-	-	-	1	92	6	92
ms04a_2	2	-	-	-	-	0	88	17	87
ms04a_3	2	-	-	-	-	2	96	12	104
rd13	1	-	-	-	-	1	72	3	81
rd13_2	4	-	-	-	-	1	75	4	102
tc02	5	-	-	-	-	0	63	9	64

Table 3.7: Architecture of the artificial neural networks used in Adviser 2b, and the input variables used by the networks.

Network	Associated Transient	Architecture	Input Variables (Numbers in this column correspond to the order in which the variables are listed in Table 3.4.)
Root	-	16 x 22 x 1	3, 5, 7, 8, 9, 10, 13, 14, 23, 24, 25, 26, 52, 74, 91, 95
Class. 1	cu10	17 x 26 x 1	3, 5, 7, 8, 9, 10, 13, 14, 23, 24, 25, 26, 50, 52, 74, 91, 95
Class. 2	cu10gp5	17 x 6 x 1	3, 5, 7, 8, 9, 10, 13, 14, 23, 24, 25, 26, 50, 52, 74, 91, 95
Class. 3	fw02a	14 x 18 x 1	5, 7, 8, 9, 10, 14, 23, 24, 25, 26, 52, 74, 91, 95
Class. 4	fw04a	40 x 13 x 1	3, 4, 5, 7, 8, 9, 10, 13, 14, 15, 16, 17, 21, 22, 23, 24, 25, 26, 29, 30, 50, 51, 52, 53, 54, 55, 56, 57, 60, 62, 63, 74, 87, 88, 89, 90, 91, 92, 93, 95
Class. 5	fw08-6	13 x 8 x 1	5, 7, 8, 9, 10, 23, 24, 25, 26, 52, 74, 91, 95
Class. 6	fw09a	5 x 3 x 1	7, 8, 9, 25, 52
Class. 7	fw12c0	9 x 5 x 1	5, 8, 15, 24, 26, 52, 74, 91, 94
Class. 8	fw12c1	12 x 19 x 1	5, 7, 8, 15, 24, 25, 26, 52, 55, 74, 91, 94
Class. 9	fw17a	26 x 21 x 1	3, 4, 8, 9, 10, 13, 14, 15, 17, 21, 22, 23, 24, 25, 26, 50, 51, 52, 54, 55, 74, 90, 91, 92, 93, 95
Class. 10	hp05	3 x 6 x 1	9, 25, 52
Class. 11	hp08	7 x 4 x 1	7, 8, 9, 23, 24, 25, 52
Class. 12	ia01	16 x 10 x 1	3, 5, 7, 8, 9, 10, 13, 14, 23, 24, 25, 26, 52, 74, 91, 95
Class. 13	ic14scr	18 x 25 x 1	3, 5, 7, 8, 9, 10, 13, 14, 19, 23, 24, 25, 26, 52, 74, 75, 91, 95
Class. 14	ic20scr2	12 x 8 x 1	5, 7, 15, 17, 24, 25, 26, 52, 55, 74, 91, 94
Class. 15	ic20scrm	6 x 5 x 1	8, 9, 24, 25, 26, 52
Class. 16	ic22scra	7 x 6 x 1	8, 9, 23, 24, 25, 26, 52
Class. 17	ic23scrm	24 x 14 x 1	3, 4, 8, 9, 10, 13, 14, 15, 22, 23, 24, 25, 26, 50, 51, 52, 54, 55, 74, 90, 91, 92, 93, 95
Class. 18	ic24scrm	20 x 15 x 1	3, 4, 8, 9, 10, 13, 14, 15, 23, 24, 25, 26, 50, 52, 54, 55, 74, 90, 91, 92

Table 3.7 (Continued)

Network	Associated Transient	Architecture	Input Variables (Numbers in this column correspond to the order in which the variables are listed in Table 3.4.)
Class. 19	mc04a	12 x 9 x 1	5, 7, 8, 10, 13, 14, 23, 24, 26, 50, 74, 91
Class. 20	ms02	32 x 17 x 1	3, 4, 5, 7, 8, 9, 13, 14, 15, 16, 17, 22, 23, 24, 25, 26, 29, 50, 51, 52, 53, 54, 63, 74, 87, 88, 89, 90, 91, 92, 93, 95
Class. 21	ms03a	25 x 18 x 1	3, 4, 5, 7, 8, 9, 10, 13, 14, 17, 21, 22, 23, 24, 25, 26, 50, 51, 52, 54, 74, 91, 92, 93, 95
Class. 22	ms14-6	7 x 7 x 1	7, 8, 9, 23, 24, 25, 52
Class. 23	ms19ab	16 x 15 x 1	3, 5, 7, 8, 9, 10, 13, 14, 23, 24, 25, 26, 52, 74, 91, 95
Class. 24	ms32	30 x 27 x 1	3, 4, 5, 7, 8, 9, 13, 14, 15, 16, 17, 22, 23, 24, 25, 29, 50, 51, 52, 53, 54, 63, 74, 87, 88, 89, 90, 92, 93, 95
Class. 25	rp05tc01	22 x 8 x 1	5, 7, 8, 9, 13, 15, 16, 22, 23, 25, 26, 50, 51, 54, 55, 74, 89, 90, 91, 92, 93, 95
Class. 26	rp5actc1	17 x 11 x 1	3, 5, 7, 8, 9, 10, 13, 14, 23, 24, 25, 26, 50, 52, 74, 91, 95
Class. 27	rr10	5 x 8 x 1	7, 8, 9, 25, 52
Class. 28	rr15a	25 x 58 x 1	3, 4, 5, 7, 8, 9, 10, 13, 14, 17, 21, 22, 23, 24, 25, 26, 50, 51, 52, 54, 74, 91, 92, 93, 95
Class. 29	rr30	28 x 17 x 1	4, 9, 15, 17, 21, 22, 25, 52, 50, 51, 54, 55, 90, 92, 93
Class. 30	rx01	5 x 10 x 1	7, 9, 25, 48, 52
Class. 31	ad05	12 x 8 x 1	4, 5, 8, 10, 13, 14, 23, 24, 50, 51, 74, 91
Class. 32	fw18a	20 x 18 x 1	3, 4, 8, 9, 10, 14, 15, 17, 21, 22, 24, 25, 26, 50, 51, 52, 54, 90, 91, 95
Class. 33	mc01a	7 x 12 x 1	7, 23, 24, 82, 83, 84, 95
Class. 34	ms04a	19 x 17 x 1	3, 4, 5, 7, 10, 13, 14, 17, 21, 22, 23, 24, 25, 26, 74, 91, 92, 93, 95
Class. 35	rd13	5 x 4 x 1	6, 23, 24, 25, 29
Class. 36	tc02	28 x 32 1	4, 8, 9, 13, 14, 15, 16, 17, 21, 22, 23, 24, 25, 26, 29, 30, 51, 52, 55, 56, 57, 60, 62, 63, 74, 87, 91, 95

GENERAL SUMMARY

In this study, a viable architecture selection technique has been presented. The dynamic input selection (DIS) scheme allows an ANN model to be generated without any guesswork regarding the selection of inputs and the size of the hidden layer. This scheme was used to develop a fault diagnostic adviser for a nuclear power plant. The adviser was designed such that it could be expanded easily to detect a greater number of transients. The diagnostic problem was broken into subtasks, and each of these was solved by an individual neural network. The adviser was initially developed to detect thirty distinct transients irrespective of the severity. This was achieved by a root networks that detected the onset of abnormal plant conditions, and thirty classifier networks that detected the particular transient in progress. The modular design of the adviser enabled the expansion of this adviser to detect and classify six more transients; the existing ANNs were slightly modified and six new ANNs were added to the adviser. The DIS scheme was instrumental in the speedy development of the adviser without any guesswork regarding the input variables and hidden nodes used in each of the component ANNs. Though the backpropagation scheme was used in this work, DIS can be applied to any learning scheme.

Some salient DIS features

In this section, some of the features, assumptions, and concerns about the DIS scheme are presented and discussed. These are issues that were not addressed in the three papers that form the body of this dissertation.

Correlation coefficient vs. information theory

Correlation coefficient r between two variables x and y, commonly given in statistics as

$$r = \frac{\sigma_{xy}}{\sigma_x \sigma_y} = \frac{E[xy] - E[x]E[y]}{\sqrt{E[x^2] - E^2[x]}\sqrt{E[y^2] - E^2[y]}} \tag{11}$$

measures the extent of linear correlation between x and y. If the relationship is totally linear, then the correlation coefficient will be 1. On the other extreme, if we have a function such as

$$y = sin(x)\ , \qquad 0 < x < \pi, \tag{12}$$

then the correlation coefficient between x and y will be equal to zero. This indicates that there is no linearity in the relationship between x and y.

We are interested in using a measure of association between x and y that tells us how important x is for the determination of y. In the linear case, the correlation coefficient is a very good measure. However, in the extreme non-linear case, we have seen that the correlation coefficient indicates that x is not important for the determination of y. We know that x and y are related by a sine function, so knowledge of x should be significant and sufficient to determine y. In such cases, the correlation coefficient would be an inappropriate measure of the importance of x to determine y. On the other hand, the information theoretic measure of association $U(y|x)$ would

have a very high value (almost 1), indicating a strong functional relationship between x and y. The non-linear nature of the information correlation enables it to identify the non-linearity in the relationship between x and y. Since most of problems suitable for a neural network based solution represent non-linear systems, it is desirable to use information correlation over linear correlation coefficient.

Relative ranking of random variables

The input variable ranking capabilities of DIS for a continuous function mapping problem is demonstrated in section 4.2 of Chapter 2. The output z is given by

$$z = sin(\pi x)sin(\pi y) \tag{13}$$

where x and y are the two input variables required to solve the problem. In the ranking analysis, there are three other inputs: $(x + y)/2$, xy, and a random number (see Table 2.4). The analysis and the subsequent training demonstrated that the DIS scheme could select the variables x and y to solve the problem. It is however of some concern that the random variable (Input 5 in Table 2.4) is ranked higher than $(x + y)/2$ and xy (see Table 2.5). Moreover, in Case 2 (Table 2.7), the random variable is ranked above x. This might indicate a problem with the ranking method, especially because 600 data patterns were used.

I had originally used a bin size of 0.05 (20 bins) for the information theoretic analysis of the data. Upon decreasing the bin size to 0.02 (50 bins) I noticed a change in the ranking of the variables. For Case 1, at a bin size of 0.01, the random variable had fallen to fourth place in the ranking analysis. For Case 2, the random variable was now ranked last. Further analysis with new sets of data indicated that the random number tended to be ranked third with a bin size of 0.05, and second on

rare occasions. But reduction in the bin size to 0.01 invariably relegated the random variable to the fourth or fifth spot.

The above experiment indicates that the selection of the bin size is important for the proper ranking of the variables. This might provide us with a means to determine the appropriate bin size. If we introduce a random variable to the data set, then we may select the bin size such that the random variable is ranked very low.

Validation of the DIS models

The effectiveness of the dynamic input selection scheme was demonstrated on benchmark problems where the scheme was able to select the correct input variables to arrive at a solution to the problem. This scheme was later used to develop models to solve real world problems, such as the temperature prediction problem in Paper II (page 79). The effectiveness of this model was illustrated by predicting the noon-time temperatures for 1993, while the model was based on data for 1991 and 1992. The result of this prediction can be seen in Figure II.2 (page 84). This model was accepted because it achieved the target RMS error over the training set and performed well over the novel data using a subset of the available input variables. However, we can perform a few tests to gauge the acceptability of the model. The first test is a test for white noise in the prediction error of the model. The second test is based on Mallows C'_{p^*} criterion that measures the DIS model versus a competing model that uses all the available input variables.

Test for white noise

The data for the temperature prediction problem happens to be a time series. Thus the prediction error of the model is also a time series. The SPECTRA procedure of the time series module in the SAS statistical package has a WHITETEST option that evaluates the whiteness of a time series. The test prints out the value of Bartlett's Kolmogorov-Smirnov Statistic [8, 5] which is the maximum absolute difference of the standardized partial sums of the periodogram of the time series data and the cumulative density function (CDF) of a uniform (0,1) random variable. The test statistic for the error data over the recall patterns was 0.0745, indicating that the error series is very close to a white noise distribution. This shows that the model derived by the DIS scheme does not consistently miss any particular feature during prediction using novel data.

Mallows C'_{p^*} criterion test

The C'_{p^*} criterion developed by Mallows [3] is used to compare a regression model using p^* parameters to the regression model that uses all the T available parameters [2, 4, 6]. The C'_{p^*} statistic is defined as

$$C'_{p^*} = \frac{SSE_{p^*}}{MSE_T} - (2p^* - n) \tag{14}$$

where p^* is the number of parameters included in a particular model with p input variables, SSE_{p^*} is the error sum of squares for the regression model with p^* parameters, T is the total number of parameters available for use in a regression model, MSE_T is the mean square error of a regression model containing all T parameters,

and n is the sample size. Equation 14 can also be expressed as [7]

$$C'_{p^*} = \frac{(1 - R^2_{p^*})(T - n)}{1 - R^2_T} - (2p^* - n) \tag{15}$$

where $R^2_{p^*}$ and R^2_T are the coefficients of multiple determination for the regression models with p^* and T parameters respectively. In the above equation, it is extremely important to have the values of $R^2_{p^*}$ and R^2_T to very high precision. A regression model is considered adequate if C'_{p^*} is equal to or very close to p^*.

We can modify the above definition of C'_{p^*} in the context of regression models in order to apply it to neural network models. The number of trainable parameters in a given ANN model can be taken as the equivalent of the number of parameters in a regression model. In the case of the ANNs used in this work, the weights are the only trainable parameters. Thus the number of parameters p^* in a model given by the DIS scheme is the number of weights in the neural network. We now need a value for T, the number of parameters in a model that uses all the available input variables. We use the dynamic node architecture (DNA) scheme to develop an ANN model with all the available inputs and the optimum hidden layer. The number of weights in this ANN is the value for T. For the two models, the squares of the correlation coefficients between the expected and predicted values over a set of novel test patterns gives us the values for $R^2_{p^*}$ and R^2_T. We can now calculate the C'_{p^*} criterion for the model provided by the DIS scheme.

As with the test for white noise, we take the temperature prediction problem for purposes of demonstration. The test set consisted of 92 patterns. Thus, $n = 92$. The DIS model had an architecture of 23 x 32 x 1 with 768 interconnecting weights. Thus, $p^* = 768$. The square of the correlation coefficient between the expected and

predicted output values for the 92 test patterns gives the value of $R^2_{p^*} = 0.802095$. We now develop an ANN model for the same problem using all the 48 available input variables. The DNA scheme is used to develop this model, and the final architecture was 48 x 44 x 1. We thus get $T = 2156$, the number of weights in this model. From this model we also get $R^2_T = 0.818452$.

We now have all the values required to calculate C_{p^*} for the DIS model. Upon substitution of the values into Equation 15, we get

$$C_{p^*} = \frac{(1 - 0.802095)(2156 - 92)}{(1 - 0.818452)} - (2(768) - 92) \quad (16)$$

$$= 805.96$$

$$\therefore \quad \frac{C_{p^*}}{p^*} = 1.04942 \quad . \quad (17)$$

This value of $\frac{C_{p^*}}{p^*}$ is sufficiently close to 1, and thus the model is acceptable. It is unlikely that any models with fewer input variables would be able to have a lower value for the ratio $\frac{C_{p^*}}{p^*}$. Though it is theoretically possible for C_{p^*} to be less than p^*, it is extremely rare to realize such a model.

Future work with DIS

The DIS scheme presented here uses a combination of principal component analysis (PCA) and information theoretic interdependency analysis (ITIA) to rank the available input variables. ITIA was not performed directly on the input variables because of the possibility that some input variables might be individually uncorrelated to the output, but together they might define the output to a very large extent. In the information theoretic notation, $U(\mathbf{y}|x_1) \approx 0$ and $U(\mathbf{y}|x_2) \approx 0$ but $U(\mathbf{y}|x_1, x_2) \approx 1$. In such a case, ITIA on the input variables would assign low importance to the input

variables x_1 and x_2, and some other input variable x_3 with higher information theoretic correlation to the output would be assigned a higher importance. This variable x_3, though ranked higher than x_1 and x_2, might be unable to predict the output without the presence of both x_1 and x_2 in the model. But since x_1 and x_2 together define the output to a great extent, x_3 might not be needed for predicting the output.

Principal component analysis was used to work around the problem described above. PCA transforms the input data into a set of mutually uncorrelated and orthogonal principal components (PC's). The PC's are projections of the original data points along mutually perpendicular axes. Each PC is a linear combination of the input variables. By converting the input data into PC's, we have preserved all the information in the input data while performing a transformation. Each PC contains the information in the original data set along one particular direction. Thus, each PC not only contains the information in each of the input variables along one direction, but it also contains the joint information in any combination of the input variables along that direction.

In its most desirable form, the information in the original input variables should have been distributed among the PC's such that knowledge of two PC's together is no more valuable to determining the output than knowledge of those two PC's individually. It is not possible to guarantee such a condition by transformation of the input data irrespective of its relationship with the output. PCA, however, performs transformation of the input data alone, and does not deal with the outputs. Practical experience has shown that for most problems, including the ones used in Paper II, the value of $U(\mathbf{y}|\omega_1,\omega_2)$ is very nearly equal to the value of $U(\mathbf{y}|\omega_1)+U(\mathbf{y}|\omega_2)$. This has enabled the use of the information theoretic correlation between any particular

PC and the output vector as a measure of the importance of that PC. Since the importance measure used in DIS is not exact, the scheme has an ability to correct any possible mistakes. Dynamic addition of inputs allows network growth. Once a problem is learned, an attempt is made to rerank the input variables actually used by the ANN, and delete any unnecessary inputs. The ability to detect and delete any unnecessary input variables from the trained ANN has been demonstrated in Paper II.

PCA is basically an analysis of the variance in the input data. The PC's are derived using the eigenvalues and eigenvectors of the data covariance matrix. It might be possible to take the matrix of the information theoretic correlation between the input variables instead of the covariance matrix, and construct a set of components based on the eigenvectors of this matrix. The information theoretic properties of these components might provide a better transformation than PCA. Other methods can also be investigated that take into account the correlation between the input variables and the output.

ADDITIONAL REFERENCES

[1] SAS Institute Inc., *SAS ETS User Guide, Version 6, First Edition*, Cary, NC : SAS Institute Inc., 1988.

[2] J. W. Gorman and R. J. Toman, "Selection of variables for fitting equations to data," *Technometrics*, vol. 8, no. 1, pp. 27-51, February 1966.

[3] C. L. Mallows, "Some comments on C_p," *Technometrics*, vol. 15, no. 4, pp. 661-675, November 1973.

[4] R. R. Hocking and R. N. Leslie, "Selection of the best subset in regression analysis," *Technometrics*, vol. 9, no. 4, pp. 531-540, November 1967.

[5] D. E. Barton and C. L. Mallows, "Some aspects of the random sequence," *Annals of Mathematical Statistics*, vol. 36, pp. 236-260, 1965.

[6] M. Kobayashi and S. Sakata, "Mallows' C_p criterion and unbiasedness of model selection," *Journal of Econometrics*, vol. 45, pp. 385-395, 1990.

[7] M. L. Berenson, D. M. Levine, and M. Goldstein, *Intermediate Statistical Methods and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey (1983).

[8] M. S. Bartlett, *An Introduction to Stochastic Processes*, Second Edition, Cambridge University Press, Cambridge (1966).

APPENDIX. DESCRIPTION OF TRANSIENT SCINARIOS

MALFUNCTION AD05

Spurious automatic depressurization system (ADS) actuation
A. Logic A
B. Logic B

Generic. Logic Failure. 100% power.

This malfunction will cause the selected ADS channel to actuate spuriously from a channel logic failure. The effected valves will not respond to any other actuation signals, open or close, auto. or manual.

The failed valves will cause the steam flow to increase and the reactor pressure will decrease. The reactor vessel water level will decrease, causing reactor scram on low level at 170 inches. The suppression pool parameters will respond to the increased temperature, pressure and level will react to the steam regulating valve (SRV) discharge. The emergency core cooling system (ECCS) will activate automatically and provide the system with cooling as the plant condition degrades.

The rate of depressurization and level decrease will be consistent with the mass and energy balances on the vessel.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

ad05.dat: Transient is inadvertant initiation of ADS (AD05A). IC'24, 100% power, middle of cycle. This is for the first trend file.

MALFUNCTION CU10

Coolant leakage outside the primary containment. Severity is variable: 0 - 100% = 0 - 4" diameter pipe (single-ended shear). Reactor Water Cleanup System (RWCS) expansion joint failure at 100% power.

This malfunction will cause a leak to occur at the cleanup system inlet expansion joint. The leak rate will be determined by the specified severity.

A low-severity leak will cause the ambient temperature to increase and will actuate the leak detection system isolation and annunciation at setpoint. As severity increases the leak detection system will be actuated by area temp/temp differentials. Prior to isolation, a brief decrease in pressure and flow will indicate mass loss on the inlet to the RWCS pumps. The pump discharge pressure will decrease proportional to leak severity and the cleanup system return temperature will decrease. When the reactor water cleanup system leak detection system activates, motor valves (MO-2700, 2701, 2740) will close, and the RWCU pumps will trip. The motor valve position indicating lights will indicate the valves are closed, and the RWCU pump motor breaker will indicate the breaker is open. The RWCU leak will cause the system pressure to decrease to atmospheric pressure. The system flow will decrease resulting in appropriate annunciation. The cleanup holding pumps will start automatically from the system low flow. System temperature will slowly decay to ambient, and the heat load on Reactor Building Closed Coolant Water (RBCCW) will decrease rapidly.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

cul0.dat: Accident is reactor water cleanup line break outside primary containment 100% break. IC24, 100% power, Middle of Cycle (MOC). Malfunction is YP:MCU10 at 100%. Decay heat is normal. No operator action.

cul0gp5.dat: Accident is reactor water cleanup line break outside primary containment 100% break. IC24, 100% power, Middle of Cycle (MOC). Malfunction is YP:MCU10 at 100%. Decay heat is normal. No operator action. Automatic group 5 isolation is overridden. Valves M02700, 2701, 2740 do not close feedwater pumps run out trip on delayed overload.

MALFUNCTION FW02

Malfunction is condensate pump trip.
A) Pump A B) Pump B

Generic, breaker overcurrent device (50) failure, 100% power.

This malfunction will cause the selected main condensate pump breaker to trip from a faulty overcurrent device (50). The condensate pump breaker will indicate open, motor current will decrease, and annunciation from the trip will occur.

When the condensate pump motor breaker trips, the pump will stop, and pump discharge pressure and flow will decrease. The corresponding reactor feedwater pump will trip and the recirculation system will run back low water level of 186 inches to 45% speed. Condensate header pressure will decrease, and flow will increase as the remaining condensate pump capacity is exceeded.

If both condensate pumps are tripped, the reactor feedwater pumps will trip. The recirculation pumps will start to run back at 186 inches Reactor Pressure Vessel (RPV) water level to 45% speed.

The reactor will scram when level reaches 170 inches.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

fw02a.dat: Accident is trip of condensate pump A. No loss of power. When a condensate pump trips, the associated feedwater pump trips automatically. Reactor trips on low level. The turbine then trips on reverse power. IC'24, 100% power, Middle of Cycle (MOC). Malfunction is YP:MFW02(A) the runback of recirculation pumps delays the reactor scram.

MALFUNCTION FW04

Malfunction is condensate filter demineralization resin injection.

A) Filter A B) Filter B
C) Filter C D) Filter D
E) Filter E

Generic, variable, 1 - 100% = 1 - 5% resin release, resin retention element failure, 100% power.

This malfunction will cause the release of resin from the selected demineralizer filter to the severity selected. Any release of resin into the condensate system will cause a buildup in the reactor vessel. Irradiation and carryover into the main steam system will cause the radiation monitoring system to respond to increased radiation levels and annunciation. High temperature and radiation cause a large increase in reactor vessel conductivity and a decrease in water pH.

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

fw04a.dat: Condensate filter demin resin injection causes increase in vessel conductivity and steam line radiation IC'24, 100% power, Middle of Cycle (MOC').

MALFUNCTION FW08

Feedwater heater tube leak.

A) Heater 1A B) Heater 1B C) Heater 2A D) Heater 2B
E) Heater 3A F) Heater 3B G) Heater 4A H) Heater 4B

Generic, variable. 0 - 100% = 0 - 4 inch diameter (equivalent to rupture of approx 30 tubes) tube failure 100% power.

This malfunction will cause a tube failure in the selected feedwater heater at a rate specified by the severity.

As the leak severity increases for the selected feedwater heater, water from the condensate system, for heaters #1 - #5, and water from the feedpump discharge, for heaters #6 will be induced into the shell side of the heater. The addition of mass will cause the mass within the heater to increase and the level control system will respond and modulate the drain and dump valves open. The cascading effect of the excess mass will impact the downline low pressure feedwater heaters and the level control systems.

A maximum severity or multiple leaks in the #6 heater could cause insufficient feedwater flow to the reactor vessel and result in a reactor scram on low water level. A maximum severity or multiple leaks in the #1 - #5 heaters could result in a low suction pressure trip of the reactor feedpumps. A less severe leak will result in decreased feedwater heating and decrease in condenser vacuum.

If the heater inleakage exceeds the normal drain and dump capacity the level will increase and actuate the high and hi-hi level annunciators. The level could backup

into the turbine resulting in a turbine trip from high vibration, reactor scram and recirculation RPT breakers to open. The hi and hi-hi level alarms are bypassed if HS1393 in bypass and there is a turbine trip or a reactor scram.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

Feedwater tube leak inside heaters 6A & 6B. IC'24, 100% power, middle of cycle. The following three simulations are for the first trend file:
fw08-6.dat: 100% severity corresponding to 30 tubes breaking
fw08-6_2.dat: 60% severity corresponding to 18 tubes breaking
fw08-6_3.dat: 30% severity corresponding to 9 tubes breaking

MALFUNCTION FW09

Malfunction is reactor feedwater pump trip.
A) Pump A B) Pump B

Generic, spurious trip signal, 100% power.

This malfunction will cause the selected main feedwater pump to trip instantly from a spurious trip signal. The pump motor breaker will trip open and annunciation will activate. The pump pressure and flow will decrease and the recirculation valve will close, if open.

The main feedwater pump trip will cause a partial loss of feedwater to the reactor, the level will decrease, and speed is runback to 45% at the recirculation pump. Reactor and turbine power are reduced accordingly. The feedwater control valves will modulate and maintain reactor water level in the control band at the reduced power level. The plant will stabilize at a new lower power. It is possible for low water level scram because of too high a power level or load line.

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

fw09a.dat: Accident is reactor feedwater pump 'A' trip. IC'24, 100% power, Middle of Cycle (MOC').

MALFUNCTION FW12

Feedwater regulator valve controller (FWRV) failure (auto).

A) FWRV A B) FWRV B C) Master controller

Generic, variable. 0 - 100% = 0 - 100% of valve position. Auto output signal failure. 100% power.

This malfunction will cause the selected feedwater regulator valve controller output to fail to the specified severity. The output will cause the regulator valve to modulate normally to the new position independent of the automatic control input signal. Placing the affected controller in the manual control mode will allow operator control.

If the regulator valve position is decreased, the flow provided will decrease, if the uneffected feedwater regulator valve capacity permits, it will open and compensate for the failed valve control. If the failed valve closes or closes sufficiently to decrease the total system capacity the reactor water level will decrease. A reactor scram may result.

If the output signal on the effected controller drops below 6 milliamps. The feedwater control valve lockout relay will actuate. An amber alarm light and annunciation will respond.

If the regulator valve position is increased, the flow provided will increase and the uneffected feedwater regulator valve will close to compensate for the failed control valve. The plant will remain stable in this condition. As flow is increased for either reactor feedwater pump to a value exceeding the design capacity, the motor current developed may exceed the overcurrent rating and the unit will trip on overcurrent.

The master controller failure will cause the feedwater regulator valve controllers in auto to respond in unison, the overall effect will be similar to the above failures. Exessive flow will increase reactor water level and decreased flow will lower reactor water level. A reactor high water level of 211 inches will result in a trip of both feedwater pumps and the main turbine.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

Master feedwater controller failure. Fails both feedwater regulator valves. 100% power, IC24, middle of cycle. The following two simulations are for the first trend file:

fw12c0.dat: Feedwater regulator valves fail fully closed
fw12c1.dat: Feedwater regulator valves fail fully opened

MALFUNCTION FW17

Malfunction is main feedwater line break inside primary containment.
A. Feed line A
B. Feed line B
Generic, variable, 0-100% = 0-16" diameter double-ended shear weld failure on outlet of check valve, 100% power. This malfunction will cause the selected feed line inside the primary containment to shear at the outlet of the check valve to the size specified by severity. This malfunction is unisolable from the reactor vessel through the selected feedwater line.

At 100% severity, the rupture will cause a rapid depressurization of the reactor vessel and feedwater line. The reactor water level will initially increase resulting in a high-level trip of the main turbine, reactor feed pumps, High Pressure Coolant Injection (HPCI) and Reactor Core Isolation Cooling (RCIC). Low-Pressure Coolant Injection (LPCI) and Core Spray (CS) will initiate on the resulting containment pressure of 2 psig and inject into the reactor vessel when reactor pressure decreases below the shutoff head of the pumps. (Inject valves for LPCI and CS will not open until reactor pressure decreases below 400 PSIG).

Drywell pressure and temperature will increase rapidly, and at 2 PSI group isolations 2,3,4,8,9 and reactor scram will occur. Suppression pool temperature and level will increase in response to the rupture severity. The reactor water level will decrease rapidly actuating reactor trip, and turbine reactor water low,low-low,low-low-low isolation signals for groups 1,2,3,4,5,7,8, seal purge.

The core spray, HPCI and LPCI systems will actuate and begin to flood the reactor with water. The unisoltated rupture will continue to cause mass loss from the reactor to the drywell and suppression pool. HPCI and RCIC will receive initiation signals on lo-lo reactor water level. If the reactor pressure is greater than 100 PSIG, these systems will initiate.

Depending on which feedwater line is broken, HPCI or RCIC will inject to the reactor vessel. ('A' feedwater line break, RCIC injects to vessel, portion of HPCI bay inject and rest through break, and the opposite is true of 'B' feedwater line breaks).

The reactor will cooldown in response to the Emergency Core Cooling System (ECCS),and the event will eventually stabilize. Reactor pressure and drywell

pressure will equalize on large break in very short period of time.
This malfunction is unrecoverable, and the simulator will have to be reinitialized for malfunction removal.

Datafiles:

fwl7a.dat: Accident is main feed water line break. 100% single-ended shear loop A IC24, 100% power, MOC.

fwl7a_2.dat: Accident is main feed water line break. 60% single-ended shear loop A IC24, 100% power, MOC.

fwl7a_3.dat: Accident is main feed water line break. 30% single-ended shear - loop A IC24, 100% power, MOC.

MALFUNCTION FW18

Malfunction is main feedwater line break outside primary containment.
A. Feed line A
B. Feed line B
Generic, variable, 0-100% = 0-16" diameter double-ended shear weld failure on outlet of feed reg valve 100% power. This malfunction will cause the selected feed line outside the primary containment to shear at the outlet of the feedwater regulator valve to the size specified by severity.

At 100% severity, the rupture will cause the feedwater header pressure to decrease rapidly to less than reactor pressure. Header flow will increase rapidly to maximum, and the feedwater pumps capacity will be exceeded and trip on low suction pressure of 250 PSIG. Initially the feedwater regulator valves will modulate open from steam/feedwater mismatch, then open when the decreasing reactor water level overrides control.

A reactor scram will occur when reactor water level decreases from lack of feedwater, the low water level causes High Pressure Coolant Injection (HPCI), and Reactor Core Isolation Cooling (RCIC) actuation and begin to flood the reactor with water and eventually recover the level. Group isolations will occur at the respective setpoints.

The reactor will cool down in response to the Emergency Core Coolant System (ECCS), and the event will eventually stabilize. This malfunction is unrecoverable, and the simulator will have to be reinitialized for malfunction removal.

Datafiles:

fwl8a.dat: Accident is main feedwater line break outside primary containment 100% break.

fwl8a_2.dat: Accident is main feedwater line break outside primary containment 60% break.

fwl8a_3.dat: Accident is main feedwater line break outside primary containment 30% break.

MALFUNCTION HP05

Malfunction is High Pressure Coolant Injection (HPCI) steam supply line break (HPCI room). Variable, exponential. 0J-J100% = 0J-J10" diameter single-ended shear weld failure on HPCI steam supply line any. HPCI in operation. This malfunction will cause the HPCI turbine steam supply line to break at the drain pot inlet. The break size will be specified by severity.

A low severity steam line break will cause the HPCI turbine speed to decrease. The turbine speed controller will cause the throttle valve to open and return the speed to setpoint.

As severity increases, the steam flow will increase and the turbine speed/pumping capacity will decrease. The emergency area cooler will detect a high differential temperature, high room temperature or high steam flow (300%) caused by the steam line break and actuate an auto-isolation signal, closing the steam isolation valves MO-2238, MO-2239, torus suction valves close, and tripping the turbine. The room fire suppression system may activate at high severities.

The HPCI turbine steam inlet valve HV-2201 will close, the turbine speed will decrease, and exhaust pressure will go to minimum. The HPCI pump discharge pressure and flow will decrease as pump capacity is lost. Reactor water level will not increase from the HPCI system. Without a manual reset the turbine will not attempt a restart at 119.5".

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

hp05_2.dat: Accident is High Pressure Coolant Injection (HPCI) steam line break in HPCI room. 60% single-ended shear IC20, 100% power, End of Cycle (EOC).

hp05_3.dat: Accident is High Pressure Coolant Injection (HPCI) steam line break in HPCI room. 30% single-ended shear IC20, 100% power, End of Cycle (EOC).

MALFUNCTION HP08

Malfunction is High Pressure Coolant Injection (HPCI) steam supply line break (torus room), variable, 0-100% = 0-10" diameter single-ended shear, weld failure on HPCI supply line at M0-2298 any, HPCI in operation. This malfunction will cause the HPCI turbine steam supply line to break in the torus room at the outlet of M0-2298. The break size will be specified by severity. A low-severity steam line break will cause the HPCI turbine speed to decrease, and the turbine speed controller will cause the throttle valve to open and return the speed to setpoint. HPCI isolation can result if the torus area temperature increases to the high setpoint or has a high D/T for greater than 15 minutes.

As severity increases, the steam flow will increase, and the turbine speed/pumping capacity will decrease. The excessive steam flow will cause a high steam line D/P isolation signal to be generated, closing the steam isolation valves MO-2238, MO-2239, torus suction valves close and tripping the turbine.

The HPCI turbine steam inlet valve MO-2202 will close, the turbine speed will decrease, and exhaust pressure will go to minimum. The HPCI pump discharge pressure and flow will decrease as pump capacity is lost. Reactor water level will not increase from the HPCI system. Without a manual reset the turbine will not attempt a restart at 119.5".

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

hp08_2.dat: Accident is High Pressure Coolant Injection (HPCI) steam line break in torus room. 60% single-ended shear IC'20, 100% power, End of Cycle (EOC').

hp08_3.dat: Accident is High Pressure Coolant Injection (HPCI) steam line break in torus room. 30% single-ended shear IC'20, 100% power, End of Cycle (EOC').

MALFUNCTION IA01

Loss of instrument air

Variable 0 - 100% = 0 - 200% of capacity (CFM). 100% capacity = 3000 CFM. Air reciever leak 100% power

This malfunction will cause the instrument air system to leak from the air reciever at a rate specified by severity.

At severities less than 100% the pressure will decrease to the auto start setpoint of

the standby compressors which will start and recharge the system.

At severities greater than 100% the system pressure will decrease with all three compressors running, severity will determine the rate of decrease. The following automatic functions will occur at setpoint:

1. 87 PSIG - service air low press ann.
2. 82 PSIG - CV3032 isolates service air hdr.
3. 85 PSIG - inst air dryer disch press low ann.
4. 80 PSIG - CV3034, CV3035, CV3039 isolate appropriate I.A. hdrs
5. 60 PSIG - breathing air low press ann.

Each isolated air header pressure will decrease dependent on header air usage and individual components will go to thier fail position or mode.

The plant is expected to trip and will stablize in a post shutdown condition.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

ia01.dat: Transient is complete loss of instrumentation air (IA01). 100% severity. IC24, 100% power, MOC. This is for the first trend file.

ia01_2.dat: Transient is complete loss of instrumentation air (IA01). 60% severity. IC24, 100% power, MOC. This is for the first trend file.

MALFUNCTION MC01

Malfunction is main circulating water pump trip.
A. Pump A, LP4A
B. Pump B, LP4B
Generic, upper motor bearing failure, 100% power. This malfunction will cause the selected main circulating water-pump motor upper bearing to fail, resulting in a motor breaker trip on overcurrent.

The circulating water-pump upper motor bearing will fail causing the motor speed or current to fluctuate. After approximately one minute the motor bearing will seize and cause a very high current to be drawn by the motor, and the supply breaker will trip on overcurrent. As the circulating water pump discharges pressure, flow will decrease, and the pump discharge valve will close. The cooling tower basin level will increase slightly then return to normal as the system mass rebalances.

With a circulating water pump tripped the circulating water temperatures will

increase across the condensers. Condenser vacuum will decrease, and annunciation and a turbine trip will result, causing a reactor scram and Reactor Pump Trip (RPT). The plant protection system will respond appropriately to the turbine trip, and the plant will stabilize in a post trip condition with Electro-hydraulic Control (EHC) maintaining reactor pressure with the bypass valves.

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafile:

mc0la.dat: Accident is main circulation water pump "a" trip. No loss of power IC24, 100% power, Middle of Cycle (MOC). Malfunction is YP:MMC01(A).

MALFUNCTION MC04

Main condenser air inleakage.

A) High pressure condenser B) Low pressure condenser

Generic, variable. 0 - 100% = 0 - 1000 SCFM at 29 in. Hg. Condenser boot seal failure. 100% power.

This malfunction will cause the selected condenser to have air inleakage at a CFM rate specified by severity.

At low severites the air ejector system will compensate for the air inleakage, however as severity is increased the selected condenser will lead a total vacuum decrease by both condensers. As the condenser pressure decreases the following functions occur:

1) 5.0" HgAbs - Condenser low vacuum annunciation
2) 5.0" HgAbs - Turbine 1B/1C low vacuum annunciation
3) 7.5" HgAbs - Turbine trip actuates
4) 19" HgAbs - Group 1 isolation signal
5) 22" HgAbs - Bypass valves are closed

The offgas system will be affected by this malfunction. The pressure and flow will increase and as severity increases, the loop seals will isolate on high pressure. The flow will alarm. The recombiner temperatures will decrease depending on the malfunction. Offgas should stabilize after a short period of time.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

Main condenser air inleakage. IC24, 100% power, middle of cycle.
mc04a.dat: 100% severity
mc04a_2.dat: 60% severity
mc04a_3.dat: 30% severity

MALFUNCTION MS02

Steam leak inside the primary containment. Variable 0 - 100% = 0 - 4" diameter single ended shear. Caused by RCIC steam line weld failure at elbow instrument tap (unisolable). 100% power. This malfunction will cause a main steam leak at RCIC line elbow instrument tap at a rate specified by severity. Very small severities

will cause local heating inside the drywell, a very slight pressure increase, and an increase in leakage to the drywell floor drain system.

The main steam flow will increase and rx pressure will decrease. The turbine Electro-Hydraulic Control (EHC) system will detect the pressure decrease and respond to maintain pressure. With the decreased steam flow the feed flow will decrease causing the reactor vessel water level to decrease until the level dominates and stablizes the level.

The drywell pressure and temp will respond quickly to the leak and the high drywell pressure trip at 2 psig. The HPCI, LPCI, CS and DG's will start, group 2, 3, 4, 5 isolations will activate. If the rx pressure decreases to 850 psig, group 1 isolation will occur and isolate the turbine bypass system. The rx will continue to blow down and the rx pressure and level will decrease consistant with the severity. Reactor pressure and temperature will decrease rapidly. The torus level and temperature will increase in response to the rupture.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafile:

ms02.dat: RCIC line break inside primary containment. The simulation was a 100% single ended shear in a 4" dia pipe. Initial condition is IC24, 100% power, Middle of Cycle (MOC). 100% severity.

ms02_2.dat: RCIC line break inside primary containment. The simulation was a 60% single ended shear in a 4" dia pipe. Initial condition is IC24, 100% power, Middle of Cycle (MOC). 60% severity.

ms02_3.dat: RCIC line break inside primary containment. The simulation was a 30% single ended shear in a 4" dia pipe. Initial condition is IC24, 100% power.

Middle of Cycle (MOC). 30% severity.

MALFUNCTION MS03

Malfunction is Main Steam Line (MSL) rupture inside primary containment.
A. Steam line A
B. Steam line B
C. Steam line C
D. Steam line D
Generic, variable. 0- 100% = 0 - 20" diameter double-ended shear piping on flow restricter at the high pressure instrument tap that feed CRM steam flow instrumentation, 100% power. This malfunction will cause the selected main steam line to rupture at the flow restricter inlet at a rate specified by severity.

At very low severities, local drywell temperatures will increase. Drywell pressure increase will be small. The drywell floor drain sump will fill up faster, and the drywell cooling system will indicate temperature increases. At lower severities, the main steam flow will increase, measured flow will decrease, and reactor pressure will decrease. The turbine bypass and control valves will modulate close, if open, in an attempt to increase steam pressure.

The feedwater control system will increase feedflow to control the increased demand. The hotwell level control system will begin to makeup from the condensate storage tank to maintain the hotwell level in the normal control band, compensating for the system mass loss. The drywell temperature/pressure will increase at an appropriate level consistent with severity.

At higher severities, the excessive main steam pressure decrease will cause the main turbine control valves to close, attempting to maintain steam pressure. Whenever the steam line pressure decreases to 850 PSIG, the main steam isolation signal and group I isolation will activate and the Main Steam isolation Valve (MSIV) will close, the reactor will scram, and the turbine bypass system will isolate. The reactor will continue to blow down, and the reactor pressure and level will decrease consistent with the severity. The drywell temperature/pressure increase will cause the ECCS to activate at 2 PSIG, and group isolation II,III & IV will occur. Reactor pressure and temperature will decrease rapidly.

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafile:

ms03a.dat: Accident is main steam line header double ended shear, 100%. (20"

line). IC24, 100% power, Middle of Cycle (MOC).

ms03a_2.dat: Accident is main steam line header double ended shear, 60%. (20" line). IC24, 100% power, Middle of Cycle (MOC).

ms03a_3.dat: Accident is main steam line header double ended shear, 30%. (20" line). IC24, 100% power, Middle of Cycle (MOC).

MALFUNCTION MS04

Malfunction is Main Steam Line (MSL) rupture outside primary containment.
A. Steam line A
B. Steam line B
C. Steam line C
D. Steam line D
Generic, variable, 0 - 100% = 0 - 20" diameter double-ended shear, piping failure at ms common header, 100% power. This malfunction will cause the selected main steam line to rupture outside the containment at the turbine inlet header at a rate specified by severity. At lower severities, the main steam flow will increase, and reactor pressure will decrease. The turbine bypass and control valves will modulate close, if open, in an attempt to increase steam pressure to control the increased demand. The hotwell level control system will begin to makeup from the condensate storage tank to maintain the hotwell level in the normal control band, compensating for the system mass loss. A Primary Containment Isolation System (PCIS) group I isolation is possible on steam line high temp (200 DEG F), and probability increases with severity.

At 100% severity, the PCIS group I isolation will be initiated on steam line low-pressure (850 PSIG in the run mode) with the steam line flow (140%) as a backup. The reactor will scram on MSIV closure. Because of the rapid steaming rate, the reactor water level will rapidly increase causing the main turbine and both reactor feed pumps, HPCI and RCIC, to trip. As the MSIVs close, steam flow through the break will cease, voids will collapse, and reactor water level will stabilize at some new lower level. Emergency Core Cooling System (ECCS) will respond to maintain adequate core cooling. The pressure rise in the turbine building will cause the blowout panels to function, releasing the steam cloud to the environment.

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafile:

ms04a.dat: Accident is main steam line header double ended shear, 100%. (20"

line). Outside primary containment. IC24, 100% power, Middle of Cycle (MOC).

ms04a_2.dat: Accident is main steam line header double ended shear, 60%. (20" line). Outside primary containment. IC24, 100% power, Middle of Cycle (MOC).

ms04a_3.dat: Accident is main steam line header double ended shear, 30%. (20" line). Outside primary containment. IC24, 100% power, Middle of Cycle (MOC).

MALFUNCTION MS14

Loss of extraction steam to feedwater heater

A) Heater 1A	B) Heater 2A	C) Heater 3A	D) Heater 4A
E) Heater 5A	F) Heater 6A	G) Heater 1B	H) Heater 2B
I) Heater 3B	J) Heater 4B	K) Heater 5B	L) Heater 6B

Generic, extraction line restriction, 100% power.

This malfunction will cause the selected feedwater heater to lose extraction steam.

The loss of extraction steam will cause the selected heater level to decrease and the associated heat gain across the heater will be lost. The resultant decrease in feedwater temperature will cause a power increase for the resultant load. The extent of power increase will be based on the heater selected, and/or the number of heaters selected. Worst case would produce a reactor scram on high flux. Less severe failures will produce possible long term effects for fuel failure considerations and administrative requirements.

The reduced drain flow to the next heater in the drain line will cause its level to decrease and the heater level controller will modulate to compensate and maintain the setpoint level. The reduced flow will cause a cascading effect for the remaining heaters down line. Once the mass transient is over, the drain heaters individual level control system will compensate for the loss of extraction steam and stabilize transient condition.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

ms14-6.dat: Loss of feedwater heating to both feedwater heaters 6A & 6B. IC24, 100% power, middle of cycle. No associated severity.

MALFUNCTION MS19

Malfunction is spurious group I isolation.
A. Logic 'A'
B. Logic 'B'
Generic, relay failure, 100% power. This malfunction will cause the selected group I isolation relay to fail and cause the isolation signal to be generated.

The failure of logic 'A' will cause the group I isolation annunciator to actuate without valve response.

The failure of logic 'B' will cause the group I isolation annunciator to actuate without valve response.

The failure of both logic 'A' and 'B' will cause the below listed inboard/outboard isolation valves to trip closed. The valve positions will be displayed at the hand-switches and 1C03. The valves are listed below:

MO-4423	MO-4424	CV-4639	CV-4640	CV-4412	CV-4415
CV-4418	CV-4420	CV-4413	CV-4416	CV-4419	CV-4421

The closing of the Main Steam Isolation Valve (MSIV) inboard or outboard will result in a reactor scram, turbine trip, and plant shutdown. Reactor high pressure will activate the Low Low Set (LLS), and reactor pressure will be controlled at about 900 to 1020 PSIG.

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafile:

msl9ab.dat: Accident is feedwater relay failures A and B logic causing a group I isolation. IC24, 100% power, Middle of Cycle (MOC).

MALFUNCTION MS32

Spurious group 7 isolation. Discrete. Short circuit causes relay CR4841 to energize. 100% power.

This malfunction will cause the selected group 7 isolation relay to fail and cause the isolation signal to be generated.

The failure of relay CR4841 will cause the group 7 isolation valves to trip closed. The valve positions will be displayed at the handswitches and 1C03. The isolation valves are listed below:

MO4841A MO4841B

MO4841A/B closure will cause the respective recirculation pumps to loose cooling, overheat and fail.

CV5718A CV5718B CV5704A CV5704B

CV5718A/B and CV5704A/B closure will cause the drywell to loose cooling, drywell temperature and pressure will increase. At 2 PSIG the reactor will scram, emergency core cooling system (ECCS) will be initiated and PCIS group 2,3,4, and 5 isolations will be initiated.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

ms32.dat: Spurious group 7 isolation. Isolates well water cooling to the drywell. IC24, 100% power, middle of cycle.

MALFUNCTION RD13

Loss of air pressure to control rod drive (CRD) hydraulic control units (HCUs).

Variable 0 - 100% = 0 - 1.5 inch diameter pipe break. Air header piping failure. 100% power.

This malfunction will cause the loss of air pressure to the CRD HCU'S at a rate consistant with the piping failure specified by severity.

Loss of air pressure will cause the flow control valves to fail closed eliminating the supply from the CRD pumps to the drive, exhaust and cooling water headers. The scram discharge volume will be isolated when CV'S 1867A/B and 1859A/B fail closed. Manipulation of the rod control system will not be available, however the charging header will remain pressurized. The drive/exhaust valves, CV1849,1850 are fail open valves. If they loose air pressure the charging header pressure will insert the control rods.

Malfunction removal will restore the effected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

rd13.dat: Transient is loss of air to HCU (RD13). 100% severity. IC24, 100% power, MOC. This is for the first trend file.

rd13_2.dat: Transient is loss of air to HCU (RD13). 60% severity. IC24, 100% power, MOC.

MALFUNCTION RP05

Malfunction RPS scram circuit failure (ATWS).
A. Auto-scram failure
B. Manual-scram failure
C. ARI failure
D. RPS fuse removal failure
E. All individual rod-scram switches fail
F. Hydraulic lock-scram discharge volume
Discrete, RPS scram circuit internal short circuit in wiring (A,B,C,D,E) scram discharge volume blockage (F), 100% power. This malfunction will cause the selected RPS scram circuit to fail to cause a reactor scram when actuated. (A,B,C,D,E) selection of the hydraulic lock malfunction will reduce the scram discharge volume to simulate flow blockage.

If the auto-scram is selected for failure, the plant will respond to the effects of the condition that generated the scram signal. The annunciators and indications will respond to auto-scram inputs as they are generated. However, the plant will remain operating until a protection feature or injection of sodium pentaborate causes the plant to shutdown. The reactor has the manual-scram capability functional, and the operator can utilize this mode as desired.

With an active auto-scram, the plant will scram as required by logic whenever the appropriate condition exists.

A failure of ARI to cause a scram will also cause a failure of the RPT breakers to trip on lo-lo reactor water level, high reactor pressure, or manual initiation of ARI.

The effects of the manual-scram feature failures would be the response failure of the function to respond when activated manually.

RPS fuse removal failure simulates a failure of the RPS fuse removal to work.

Failure of the rod-scram switches simulates a failure of all 89 scram switches to work.

The hydraulic lock malfunction reduces the volume and will allow the rods to partially insert, with each scram signal/reset applied.

Insertion of all (6) generic failures will result in a "ATWS" condition.

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafiles:

rp05tc0l.dat: Accident is trip of main turbine together with failure to scram (ATWS). No operator action. IC'24, 100% power, Middle of Cycle (MOC). Malfunction is YP:MTC'0l (turbine trip) followed by YP:MRP05(A) (failure to automatically scram). Scram will be delayed and slow. Scram is from alternate rod insert (ARI). Recirculation pumps also trip on ARI. ARI is triggered at 119" vessel level or 1140 PSIG vessel pressure. Recirculation pump trips at turbine control valves fast closure or stop valves less than 90% open.

rp5actcl.dat: Accident is trip of main turbine together with failure to scram (ATWS), and Alternate Rod Insert (ARI). No operator action. IC'24, 100% power, Middle of Cycle (MOC). Malfunction is YP:MTC'0l (turbine trip) followed by YP:MRP05(A) (failure to automatically scram) and YP:MRP05(C) (failure of ARI). Recirculation pump trips on turbine control valves fast closure or stop valves less than 90% open.

MALFUNCTION RR10

Malfunction recirculation pump speed feedback signal failure.
A. Pump A
B. Pump B
Generic, variable, 0 - 100% = 0 - 100% of feedback signal, speed control circuit failure, 50% power. This malfunction will cause the selected recirculation pump speed-control feedback circuit to fail to the specified severity. The pump speed indicator will fail to the specified severity. With the tacho-generator signal failing below the speed demand/manual pot position signal, the recirculation pump actual speed will increase, and the scoop tube will increase to maximum or auto lock if auto-lock conditions are met. With the tacho-generator signal failing above the speed demand/manual pot position signal, the recirculation actual speed will decrease to minimum.

The resulting effect on the plant will be the increase in power for an increased recirculation flow and a decrease in power for a decreased recirculation flow. Turbine generator power and control valve positions will respond as appropriate. Annunciator response to flow limits and control failures will actuate at setpoint.

Reactor water level will respond to the opposite of the recirculation speed initially until feedflow and steam flow can get matched at the proper water level.

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafile:

rrl0.dat: Accident is recirculation pump speed feedback signal failure caused by speed circuit control failure IC24, 100% power, Middle of Cycle (MOC).

MALFUNCTION RR15

Malfunction is recirculation loop rupture (design basis Loss of Coolant Accident (LOCA) at 100%).
A. Loop A
B. Loop B
Generic, variable, 0 - 100% = 0 - 22" diameter double-ended shear piping failure at recirc pump suction, 100% power. This malfunction will cause the selected recirculation loop, inside the primary containment, to shear at the recirculation pump suction to the size specified by severity.

At 100% severity, the rupture will cause the recirculation loop and reactor pressure to decrease rapidly. The affected loop recirculation pump will cavitate and flow will be lost. Reactor water level will decrease rapidly as the reactor blows down through the rupture into the containment. The reactor water level decrease will actuate reactor scram, and reactor water level low, low-low, low-low-low isolation signals for groups 1, 2, 3, 4, 5, 7, 8, seal purge. The Core Spray (CS), High-Pressure Coolant Injection (HPCI), Low-Pressure Coolant Injection (LPCI), DG, and ADS systems will actuate and begin to flood the reactor with water. The RR pump discharge valves will close on the non-broken loop on the LPCI loop select signal. Reactor feed pumps will trip on overcurrent.

At 100% severity, the HPCI and RCIC will receive initiation signals. However, the reactor pressure will decrease so fast that they will trip and isolate on low pressure before they will have any noticeable effect. Drywell pressure and temperature will increase rapidly and at 2 PSI group isolations 2,3,4,8,9 will occur. Suppression pool temperature and level will increase in response to the rupture severity. The reactor will cooldown in response to the Emergency Core Coolant System (ECCS), and the event will eventually stabilize.

This malfunction is unrecoverable, and the simulator will have to be reinitialized for malfunction removal.

Datafiles:

rrl5a.dat: Accident is recirculation loop rupture. 100% double-ended shear - loop a IC24, 100% power, Middle of Cycle (MOC).

rrl5a_2.dat: Accident is recirculation loop rupture. 60% double-ended shear - loop a IC24, 100% power, Middle of Cycle (MOC).

rrl5a_3.dat: Accident is recirculation loop rupture. 30% double-ended shear - loop a IC'24, 100% power, Middle of Cycle (MOC).

MALFUNCTION RR30

Malfunction is coolant leakage inside primary containment. Variable (exponential). 0 - 100% = 0 -J2" diameter pipe (double-ended shear) reactor vessel bottom drain weld failure, 100% power. This malfunction will cause reactor coolant to leak from the reactor vessel bottom drain failed weld at a rate specified by severity. As severity increases the mass loss from the reactor will easily be made up for by the hotwell level control system. In fact, at 100% severity the effects on reactor level/hotwell level will be very small. The most effective display of mass loss will be at hot standby.

At 100% severity, drywell pressure, temperature, and activity will increase. At 2 PSI, group isolations 2,3,4,5,8 will occur. Suppression pool temperature and level will increase in response to the rupture severity. The reactor scram will result from the 2 PSIG drywell pressure, and a turbine trip will result from reserve power.

The shutdown plant will cooldown in response to the Emergency Core Cooling (ECC) and will stabilize. The longterm effect of the leak will be the transfer of the Condensate Storage Tanks (CST) mass to the suppression pool via the leak in the reactor vessel.

At small severities where the drywell pressure remains below 2 PSIG, the floor drain equipment system will see a high leak (in excess of the 5 GPM tech spec limit). The drywell cooler heat load will increase as seen on the cooler temperatures on IC'25.

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafile:

rr30.dat: Accident is reactor bottom head drain 100% single-ended shear (2" line). IC'24, 100% power, Middle of Cycle (MOC).

rr30_2.dat: Accident is reactor bottom head drain 60% single-ended shear (2" line). IC'24, 100% power, Middle of Cycle (MOC).

rr30_3.dat: Accident is reactor bottom head drain 30% single-ended shear (2" line). IC'24, 100% power, Middle of Cycle (MOC).

MALFUNCTION RX0l

Malfunction is fuel cladding failure. Variable, exponential, 0 -100% = 0 - 30% fuel clad damage, fuel cladding degradation, 100% power. This malfunction will cause the fuel cladding to fail to a value specified by severity.

As the fuel failure increases, the amount of activity in the reactor recirculation and main steam system will increase. This activity will propagate throughout the plant and the radiation monitoring system will detect, indicate, and alarm as the activity increases. At low severities, offgas post-treat radiation monitors will cause offgas to isolate (without a group isolation) resulting in a loss of condenser vacuum, main turbine trips, and reactor scram. As the severity increases, the main steam line radiation monitors will cause main steam line isolation and reactor scram at setpoint. As the normal power dependent background radiation levels decrease, the additional radiation levels will be more evident on area and process monitors. At high severities the before mentioned will occur faster with more dramatic increases. Various system trips and isolations will occur, protecting the environment from excessive discharges.

The sequence of the fuel failure indication will be as follows:

1. Offgas pretreat and post-treat radiation monitors increase
2. Offgas stack release will start to increase
3. Offgas system isolates on post-treat hi-hi radiation level
4. Main steam line radiation monitors respond:
   a. MSL high radiation alarm
   b. Group I isolation
   c. Reactor scram
5. Drywell monitors increase
6. Torus radiation monitors increase from relief valve discharge or HPCI and/or RCIC exhaust.
7. Reactor building area radiation increases from Emergency Core Cooling System (ECCS) system operation. (HPCI, RCIC, LPCI, CS).

NOTE: Malfunction severity will cause some of the above items to be "passed over" or will result in a delayed response.

Datafile:

rx0l.dat: Accident is 30% fuel clad failure. Causes high radiation alarm to go off IC24, 100% power, Middle of Cycle (MOC).

MALFUNCTION TC02

Malfunction is EHC hydraulic pump trip.
A. Pump A
B. Pump B
Generic, motor failure, 100% power. This malfunction will cause the selected EHC pump motor to fail and trip on overload. The starter will open, and an annunciator will actuate.

The unaffected EHC pump will start when the low pressure annunciator actuates at 1300 PSIG EHC system pressure.

If the unaffected EHC pump is unavailable or both EHC pumps are failed, the turbine will trip at 1100 PSIG EHC system pressure. When the turbine trip is initiated, the turbine stop valves, control valves, and combined intermediate valves will rapidly close. The reactor/main steam pressure increases rapidly activating the turbine bypass system. The turbine bypass system will modulate as necessary to control the steam pressure by dumping steam to the main condenser. A reactor scram will occur as a result of the turbine trip. Recirculation pumps will trip when the RPT breakers open from a turbine trip.

4KV bus close-circuit transfer is initiated. The main generator 286/B lockout relay will actuate from the anti-motoring trip protection. The main generator output breakers 352-H and/or 352-I will trip open and lockout. The generator exciter field breaker will open. Generator indication of current, voltage, megawatts, megavars, etc. will decrease to zero. Appropriate annunciators for generator trip will actuate.

The plant will stabilize in a post shutdown condition.

Malfunction removal will restore the affected components to normal. Operator action may be required to restore the plant to normal.

Datafile:

tc02.dat: Accident is EHC hydraulic pump trip causes pump motors to fail and trip on overload. IC24, 100% power, Middle of Cycle (MOC).

SOME SELECTED INITIAL CONDITIONS (IC) DESCRIPTIONS

IC-14 FULL POWER OPERATIONS (ANSI PARAMETERS)

A) Moderator temperature = saturated conditions

B) Reactor pressure = power dependent

C) Reactor power level = 100% power

D) Reactivity = critical

E) Xenon condition = 100% equilibrium

F) Core life = beginning of life

This IC is set up to meet the ANSI criteria for 100% power.

IC-20 FULL POWER OPERATION/MAX DECAY HEAT (EOL)

A) Moderator temperature = saturated conditions

B) Reactor pressure = power dependent

C) Reactor power level = 100% power

D) Reactivity = critical

E) Xenon condition = 100% equilibrium

F) Core life = end of life

This IC is similar to IC-15 except it has 20% more decay heat than normal.

Note: *Effective Jan 27, 1993, some changes in IC-nomenclature were made.* The above initial conditions - the older "IC-20" - is now "IC-26" except that it has 20% more decay heat than normal (see below). The new initial conditions "IC-20" is similar to IC-14 except that all cooling tower fans are on.

IC-22 25% POWER (ANSI PARAMETERS)

A) Moderator temperature = saturated conditions

B) Reactor pressure = power dependent

C) Reactor power level = 25% power

D) Reactivity = critical

E) Xenon condition = equilibrium for 25% power

F) Core life = beginning of life

This IC is set up to meet the ANSI criteria for 25% power.

IC-23 75% POWER (ANSI PARAMETERS)

This IC is similar to IC-22, except at 75% power.

IC-24 FULL POWER OPERATIONS (MOL)

This IC is similar to IC-14, except the core is at middle of life (MOL).

IC-26 100% POWER (EOL)

This IC is similar to IC-20 (i.e., the older version given above), except that it has 20% more decay heat than normal.

Datafiles:

ic14scra.dat: Accident is spurious scram. No operator action. IC14, 100% power, BOC. Malfunction is YP:MRP03.

ic20scr2.dat: Accident is spurious scram with operator action. IC20, 100% power, EOC. Malfunction is YP:MRP03. Decay heat is 10% of full power. Operator action is according to Integrated Plant Operating Instruction (IPOI) NO. 5 - reactor scram. Actions include: mode switch to shutdown position, feedwater level controller to 175 inches, use reactor water cleanup to maintain RPV level, trip 'A' feed and condensate pumps, manually control feed regulator valves, insert all SRM and IRM detectors. Level was manually controlled to prevent the trip of feed pumps and turbine.

ic20scrm.dat: Accident is spurious scram. No operator action. IC20, 100% power, EOC. Malfunction is YP:MRP03.

ic22scrm.dat: Accident is spurious scram. No operator action. IC22, 25% power, BOC. Malfunction is YP:MRP03.

ic23scrm.dat: Accident is spurious scram. No operator action. IC23, 75% power, BOC. Malfunction is YP:MRP03.

ic24scrm.dat: Accident is spurious scram. No operator action. IC24, 100% power, MOC. Malfunction is YP:MRP03.