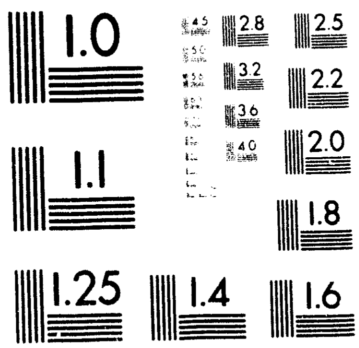
1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

1 of 1

Conf-930749--46

LA-UR-93-2604

**Title:** DATA REQUIREMENTS FOR AN ANOMALY DETECTOR IN AN AUTOMATED SAFEGUARDS SYSTEM USING NEURAL NETWORKS

**Author(s):** R. Whiteson and J. J. Britschgi

**Submitted to:** 34th Annual Meeting of the Institute of Nuclear Materials Management, Scottsdale, Arizona, July 18-21, 1993



**Los Alamos**
NATIONAL LABORATORY

# DATA REQUIREMENTS FOR AN ANOMALY DETECTOR IN AN AUTOMATED SAFEGUARDS SYSTEM USING NEURAL NETWORKS*

R. Whiteson
Safeguards Systems Group, MS E541
Los Alamos National Laboratory
Los Alamos, New Mexico 87545
(505) 667-7777

J. J. Britschgi
Westinghouse Idaho Nuclear Company
P.O.B. 4000, MS 5102
Idaho Falls, Idaho 83415-5102
(208) 526-2258

## ABSTRACT

An automated safeguards system must be able to detect and identify anomalous events in a near-real-time manner. Our approach to anomaly detection is based on the demonstrated ability of neural networks to model complex, nonlinear, real-time processes. By modeling the normal behavior of processes, we can predict how a system should behave and, thereby, detect when an abnormal state or event occurs. In this paper, we explore the computational intensity of training neural networks, and we discuss the issues involved in gathering and pre-processing the safeguards data necessary to train a neural network for anomaly detection. We explore data requirements for training neural networks and evaluate how different features of the training data affect the training and operation of the networks. We use actual process data to train our previous 3-tank model and compare the results to those achieved using simulated safeguards data. Comparisons are made on the basis of required training times in addition to correctness of prediction.

## NEURAL NETWORKS

Neural computing attempts to simulate the functions of the human brain and create models by matching that functionality. These models are based on the assumption that information processing takes place through the interactions of many simple processing units or nodes, each sending excitatory and inhibitory signals to other units.[1,2] In contrast to traditional rule-based systems, neural networks are better able to extract expert knowledge from raw data by adaptively processing large amounts of knowledge, efficiently reducing data, and robustly classifying input patterns. During an iterative training process, the network forms a model of the relationship of the outputs as a function of the inputs. It can model highly nonlinear processes, given the correct internal architecture. Figure 1 shows a typical neural network with three layers. The input layer contains five nodes. The hidden layer has three nodes and the output layer four nodes. Input to this network would be input vectors with five elements, one element going to each node in the input layer. It would produce an output vector of four elements.

Neural networks have been shown to be very versatile in modeling the input-output relationships of complex nonlinear systems. As a result, this artificial intelligence technique has recently been proposed to provide support for nuclear power plant operations.[3-5] Because of the complexity of the processes and the large and diverse amount of data, efficient automatic algorithms are necessary to interpret the data and ensure secure plant operation.[6]

Because prediction is one of the most promising applications of neural networks, it is a natural choice for modeling nuclear plant activity.[7,8] Modern safeguards systems for nuclear materials handling typically use distributed control systems that record and store large amounts of data.[9] From this data, we are using current and past information about characteristics of the physical system to predict future correlated characteristics. By comparing predicted states with those reported by a control system, anomalies can be detected.[10]

## SIMULATED DATA AND REAL DATA

In earlier work using neural networks to detect anomalies in safeguards systems, we created training and test data using a simulation.[11] This was done because real data are generally noisy and occasionally erroneous. The data that we created represented tank volumes and sensor states from M-cell processing

*This work supported by the U.S. Department of Energy, Office of Safeguards and Security.

tanks in the Idaho Chemical Processing Plant (ICPP). See Fig. 2 for a diagram of the simulated flow of materials.

For our current work, we used raw data obtained from the Process Monitoring Computer System (PMCS) at the ICPP. The PMCS is a set of data acquisition devices that transmits process data from plant instruments and speciality sensors such as remote valves, pumps, and steam jets installed on the plant equipment to a computer for data processing and storage. These data are in the form of analog and digital outputs.[12] The PMCS database is used by safeguards personnel to monitor solution movements to and from the measurement vessels. These vessels are positioned so that all input and output streams can be measured. Data from the inventory vessels M-101, M-102, M-103, and M-104 are the focus of our current work. Figure 3 shows the flow of material.[13]

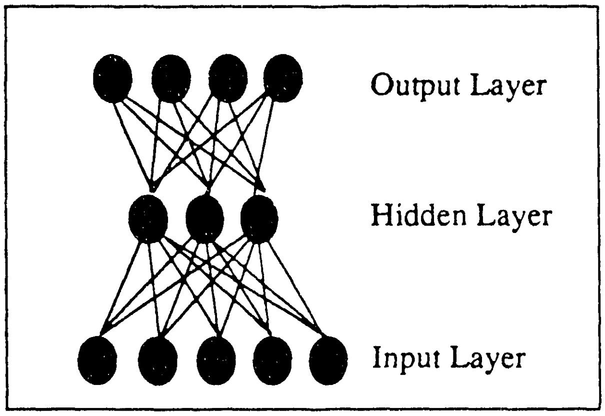


*Fig. 1. A neural network with three layers.*

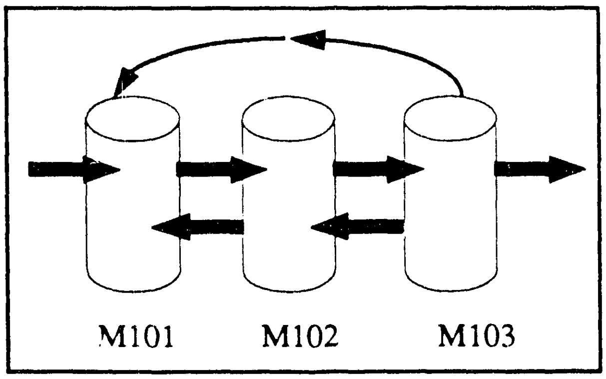


*Fig. 2. Simulated flow of materials.*

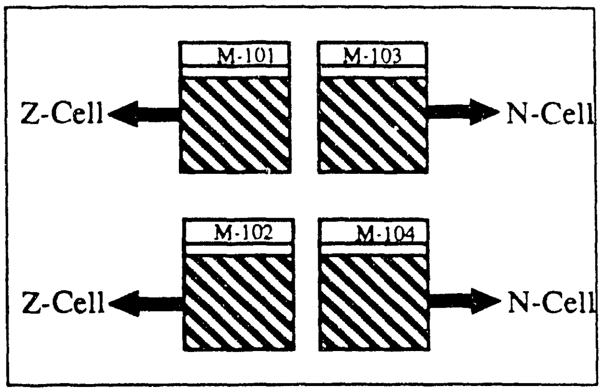


*Fig. 3. Measurement vessels in M-cell.*

The ICPP uranium fuel process operation consists of a number of headend dissolution processes, one cycle of tributyl phosphate solvent extraction, two cycles of hexone extraction, and one denitration step to end with uranium trioxide as a product.[12]

## PREPROCESSING THE DATA

Two sets of process data were available. The first is from a campaign run from November 15, 1988, to May 1, 1989. We will refer to this as campaign 1 data. The second is from a campaign run from April 1, 1991, to January 1, 1992. We will refer to this as campaign 2 data.

To compare results with those from the networks that were trained and tested with simulated data, we used the same software to preprocess the data. This software takes the raw data and outputs snapshots of the facility at four-minute intervals. As with the simulated data, we created input vectors that consisted of sensor states from three consecutive snapshots. The output, the predictions of the network, was to be predicted changes in the tank volumes. For our simulated data, we used sensor states that we knew were correlated to changes in tank volumes. With our real data, we did not initially know which sensors indicated activity in the tanks. Campaign 1 data contained volumes for 4 tanks and 48 sensors. Campaign 2 data contained volumes for 3 tanks and 21 sensors. Because our input vector was to include the three consecutive values for each sensor, it was important to include data from relevant sensors only. If we used data from all sensors, our input vector for campaign 1, for example, would contain 144 elements, resulting in a cumbersome network and increased training as well as testing times. To accomplish the necessary feature selection, we analyzed the raw data to determine which sensors indicated tank activity. Table I shows the sensors with the highest correlations and the sensors selected for the input vectors. From campaign 1 data we selected 12 sensors. Because we are using values from 3 consecutive snapshots of each sensor, a network with 36 input nodes is required. See Fig. 4 for a diagram of this network. From campaign 2 data we selected 5 sensors. The network thus has 15 input nodes, as shown in Fig. 5.

**Table 1: Sensors Correlated to Tank Volumes**

| | Campaign 1 | Campaign 2 |
|---|---|---|
| Tank | Correlated Sensors | Correlated Sensors |
| M101 | 0, 3, 32 | 7, 12, 18 |
| M102 | 4, 18, 33 | 10, 11, 18 |
| M103 | 7, 11,3 4 | 10, 11, 18 |
| M104 | 12, 16, 35 | N/A |
| Selected Sensors | 0,3,4,7,11,12,16, 18,32,33,34,35 | 7,10,11,12,18 |

Again, to make meaningful comparisons with the networks we had used on the simulated data, we designed our new networks to be similar. That is, we used feed forward, back-propagation networks with one hidden layer, a hyperbolic tangent transfer function, and the Norm-Cumulative-Delta learning rule.

## COMPARISON OF RESULTS WITH REAL DATA AND SIMULATED DATA

Using simulated data, our networks were very successful in predicting changes in tank volumes; they predicted every transaction and detected several instances of loss of material from the system.[11] However, using real data, the results were not as impressive. There were no known anomalies in these data, but there were a number of normal transactions. For campaign 1 our network correctly predicted 58 transactions where tank volumes changed. However, in five cases it predicted a transaction where in fact there was none. In campaign 2, the network correctly predicted 12 transactions but erroneously predicted 2 transactions.

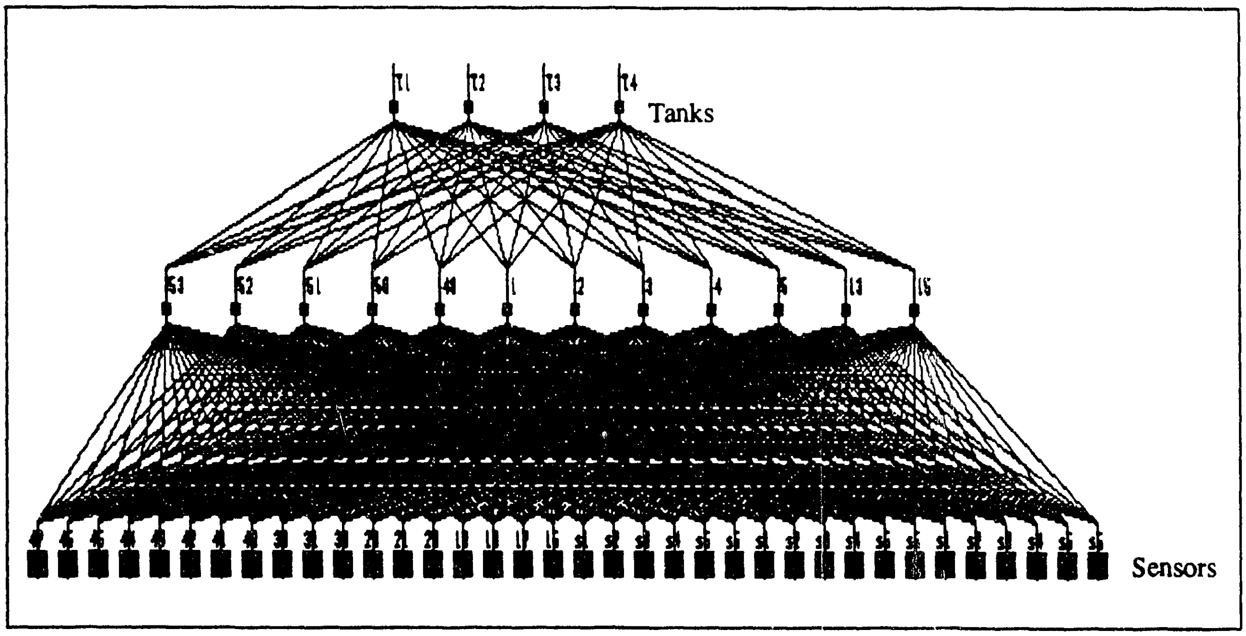


*Fig. 4. Network for campaign 1. Input data are from 12 sensors, 3 values for each sensor. Outputs represent changes involes of four tanks.*

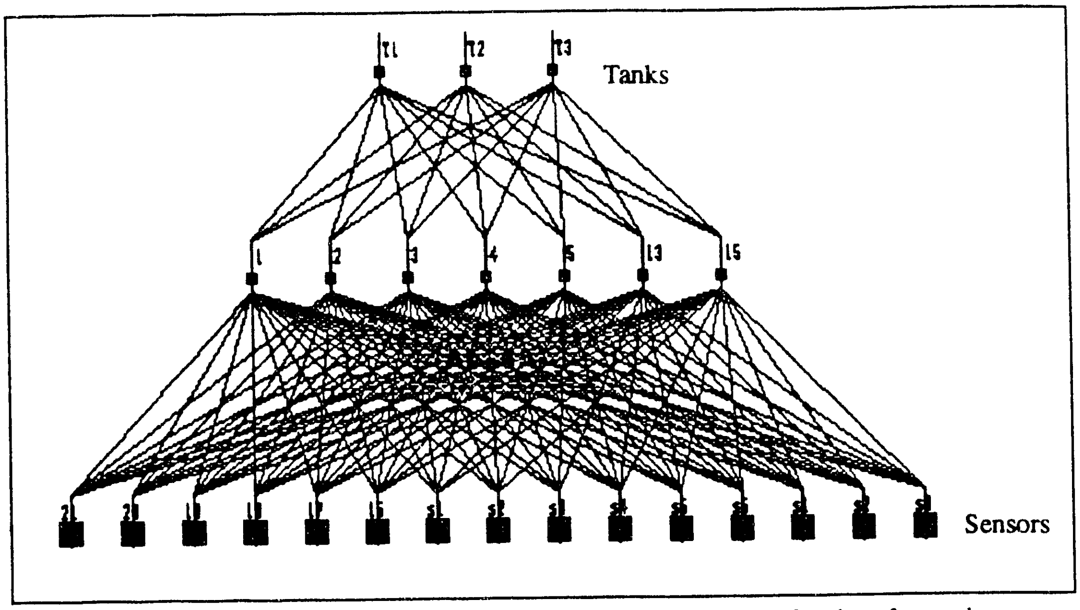


*Fig. 5. Network for campaign 2. Input data are from 5 sensors, 3 values for each sensor. Outputs represent changes involes of three tanks.*

## SELECTING VALUES FOR INPUT VECTORS

We hypothesized that using a subset of sensors with three consecutive states for each would create the most useful and effective input vector. To test this hypothesis, we utilized the data sets described above and, in addition, we created training and test sets that contained only the current value for each of the sensors. For example, for the latter set, campaign 2 data required an input vector of 21 elements, one value for each of the 21 sensors. The networks that used these data were identical in all other ways to the networks using three values for each selected sensor.

The results of testing our hypothesis using the two data sets described above show that networks using real data in input vectors containing only one value for each sensor produced much poorer results than input vectors with three consecutive values for each. This is true even though in the former, data from all sensors were included.

## COMPUTATIONAL COMPLEXITY

The computational complexity of training time for neural networks is well understood. The time required for the network to process one training vector is a function of the number of connections in the network. In other words, the complexity is $O(n)$ where $n$ is the number of connections. The training time for an entire network can be viewed as $O(n \times m)$ where $m$ is the number of training vectors. However, this is complicated by the fact that during training, networks process each training vector a number of times. The training time for an entire network is not easy to quantify as it is often difficult to determine when the network is fully trained, although this time is generally considered to be NP-Complete.[14]

## CONCLUSIONS AND FUTURE DIRECTIONS

Our experiments with neural networks on simulated and real data from a process monitoring system indicate that a neural-network-based module may be able to provide an anomaly detection system as an adjunct to a material control and accounting system. If a trained network represents a good model of normal plant operation, it can be a reliable tool for recognizing non-normal activity. The neural network will then be able to flag any anomalous facility state such as loss or diversion of material, instrument failures, or other abnormal or unusual events. This would improve the sensitivity of any existing system and reduce inspector effort.

The design of the architecture for a specific problem is somewhat of an art form, and it is clear that the choice of architecture and learning rule is critical to success. However, overall success for many applications may rely more on data representation (preprocessing and encoding) than the choice of architecture or learning rule.[15] We believe that more work needs to be done on tools for preprocessing data. One approach would be development of sophisticated tools that can aid in feature selection, data aggregation, data fusion, concept extraction, and handling of missing or erroneous data. Another approach would be to work with an expert to accomplish these same goals. Further efforts in the areas of architecture design and data preprocessing should lead to neural networks that can more accurately model processes and detect anomalies.


## ACKNOWLEDGMENTS

The authors would like to thank Frank German and Neil Liester of Westinghouse Idaho Nuclear Company for their assistance in obtaining and extracting the data.


## REFERENCES


1. D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing* (MIT Press, Cambridge, Massachusetts. 1989).

2. Yann le Cunn, "Generalization and Network Design Strategies," in *Connectionism in Perspective*, Rolf Pfeifer, Zoltan Schreter, Francoise Fogelman-Soulie, and Luc Steels, Eds. (Elsevier Science Publishers B.V., Amsterdam, Holland, 1989), pp. 143- 156.

3. M. S. Roh, S. W. Cheon, and S. H. Chang, "Power Prediction in Nuclear Power Plants Using a Back-Propagation Learning Neural Network," *Nuclear Technology* **94**, 270 (1991).

4. B. R. Upadhyaya and E. Eryurek, "Application of Neural Networks for Sensor Validation and Plant Monitoring," *Nuclear Technology* **97**, 170 (1992).

5. E. B. Bartlett and R. E. Urig, "Nuclear Power Status Diagnostics Using an Artificial Neural Network," *Nuclear Technology* **97**, 272 (1992).

6. A. Zardecki, R. Whiteson, and A. Coulter, "Pattern Recogr. ion Methods for Anomaly Detection," Los Alamos National Laboratory, Safeguards Systems Group document N-4/91-420 (April 9, 1991).

7. R. D. Jones, Y. C. Lee, C. W. Barnes, G. W. Flake, K. Lee, P. S. Lewis, and S. Qian, "Function Approximation and Time Series Prediction with Neural Networks," Los Alamos National Laboratory document LA-UR-90-21 (1990).

8. Kanad Chakraborty, Kishan Mehrotra, Chilukuri K. Mohan, and Sanjay Ranka, "Forecasting the Behavior of Multivariate Time Series Using Neural Networks," *Neural Networks* **5**(6), 961-970 (November 1992).

9. M. H. Ehinger, N. R. Zack, E. A. Hakkila, and F. Franssen, "Use of Process Monitoring For Verifying Facility Design For Large-Scale Reprocessing Plants," Los Alamos National Laboratory report LA-12149-MS (1991).

10. D. E. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering* **SE-13**(2), 222-232 (February 1987).

11. R. Whiteson and J. A. Howell, "Anomaly Detection in an Automated Safeguards System Using Neural Networks," *Nucl. Mater. Manage.* **XXI**, 411-417 (1992).

12. C. A. Dahl and N. A. Liester, "The Idaho Chemical Processing Plant Process Monitoring Computer System," in *Proceedings of the Third International Conference on Facility Operations—Safeguards Interface* (American Nuclear Society, La Grange Park, Illinois, December 1987), pp. 263-269.

13. F. O. German, "Accountability Volume Measurement at the Idaho Chemical Processing Plant," Westinghouse Idaho Nuclear Company, Inc. report WIN-333 (November 1991).

14. A. L. Blum and R. L. Rivest, "Training a 3-Node Neural Network is NP-Complete," *Neural Networks* **5**(1), 117-127 (1992).

15. Vladimir Cherkassky and Hossein Lari-Najafi, "Data Representation for Diagnostic Neural Networks," *IEEE Expert* **7**(5), 43-53 (October 1992).

DATE

FILMED

11/8/93

END