Estimating All-Terminal Network Reliability Using a Neural Network

Chat Srivaree-ratana and Alice E. Smith
Department of Industrial Engineering
University of Pittsburgh
Pittsburgh, PA 15261 USA
aesmith@engrng.pitt.edu

ABSTRACT

The exact calculation of all-terminal network reliability is an NP-hard problem, with computational effort growing exponentially with the number of nodes and links in the network. Because of the impracticality of calculating all-terminal network reliability for networks of moderate to large size, Monte Carlo simulation methods to estimate network reliability and upper and lower bounds to bound reliability have been used as alternatives. This paper puts forth another alternative to the estimation of all-terminal network reliability – that of artificial neural network predictive models. Neural networks are constructed, trained and validated using alternative network topologies, a network reliability upper bound and the exact network reliability as a target. A hierarchical approach is used: a general neural network screens all network designs for reliability followed by a specialized neural network for highly reliable network designs. Results on a ten node problem are given using a grouped cross validation approach.

1. INTRODUCTION TO THE PROBLEM

Reliability and cost are the two most important considerations when designing communications networks, especially backbone telecommunications networks, wide area networks, local area networks and data communications networks located in industrial facilities. If the *nodes* (stations, terminals or computer sites) of the network are fixed, the main design decisions are selection of the type and routing of *links* (cables or lines) of the network to ensure proper and reliable operation while meeting cost constraints. The following define the problem assumptions:

1. The location of each network node is given.
2. Nodes are perfectly reliable.
3. Link costs and reliabilities are fixed and known.
4. Each link is bi-directional.
5. There are no redundant links in the network.
6. Links are either operational or failed.
7. The failures of links are independent.
8. No repair is considered.

Mathematically, the design optimization problem can be expressed as:

$$\text{Minimize } Z(\mathbf{x}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij}\, x_{ij} + \delta\,(c_{MAX}(R(\mathbf{x})-R_o))^2$$

where:

$$\delta = \begin{cases} 0, & \text{if } R(\mathbf{x}) \geq R_o \\ 1, & \text{if } R(\mathbf{x}) < R_o \end{cases}$$

N	set of nodes
(i,j)	a link between nodes i and j
x_{ij}	decision variable, $x_{ij} \in \{0,1\}$
$\mathbf{x}$	a link topology of $\{x_{11}, x_{12}, \ldots, x_{ij}, \ldots, x_{N,N-1}\}$
$R(\mathbf{x})$	reliability of $\mathbf{x}$
R_o	network reliability requirement
Z	objective function
c_{ij}	cost of (i,j)
c_{MAX}	the maximum value of c_{ij}

The $R(\mathbf{x})$-R_o term penalizes networks that do not meet the minimum reliability constraint, and moves the search to the set of feasible networks.

The network design problem has been studied in the literature with both enumerative based methods (usually a variation of branch-and-bound) [15] and heuristic methods [1, 4, 6, 7, 8, 17]. One aspect of these methods is that network reliability must be calculated for each and every candidate network design identified, often running to thousands or millions of designs. For networks of realistic size, a computationally expedient alternative to the exact network reliability must be found.

The network design problem is especially difficult when considering *all-terminal* network reliability (also called *uniform* or *overall* network reliability), defined as the probability that all nodes can communicate with all other nodes. This is equivalent to stationary availability when a mission time is not implicitly assumed. The difficulty arises because the exact calculation of all-terminal network reliability is NP hard, that is, computational effort increases exponentially with network size [11]. Therefore, exact calculation of network reliability is not usually practical for networks of realistic size. Monte Carlo stochastic simulation methods can estimate network reliability very precisely [9, 20], however, simulation must be repeated numerous times to ensure a good estimate. Therefore, the simulation approach also incurs significant computational effort when estimating the reliability of the network (that is, of the communications system).

2. NEURAL NETWORKS FOR ESTIMATION OF NETWORK RELIABILITY

Neural networks were inspired by the power, flexibility and robustness of the biological brain. They are computational (mathematical) analogs of the basic biological components of a brain - neurons, synapses and dendrites. Artificial neural networks (hereafter referred to as ANN) consist of many simple computational elements (summing units — *neurons* — and weighted connections — *weights*) that work together in parallel and in series (Figure 1). Neural networks begin in a random state and "learn" using repeated processing of a training set, that is, a set of inputs with target outputs. Learning occurs because the error between the ANN output and the target output is calculated and used to adjust the weighted synapses of the ANN. This continues until errors are small enough or no more weight changes are occurring. The ANN is then trained and the weights are fixed. The trained ANN can be used for new inputs to perform estimation or classification tasks.

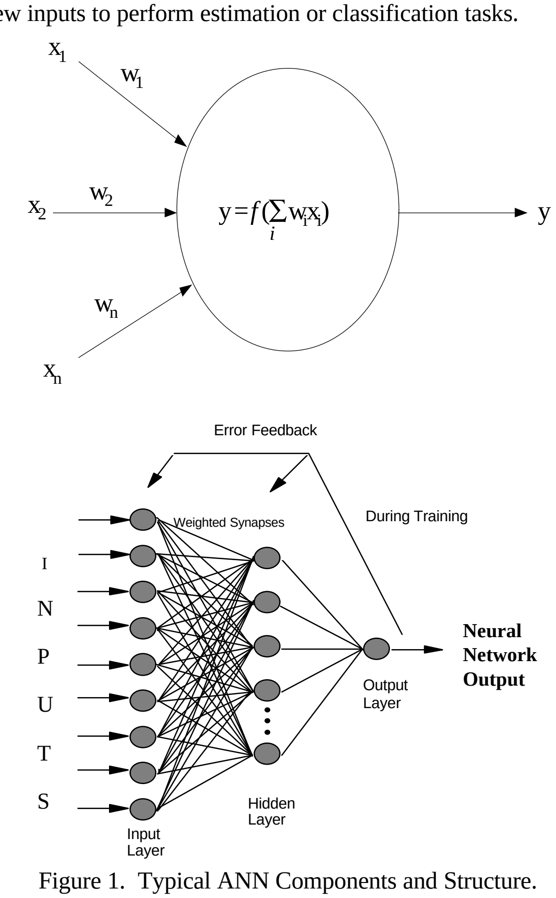

Figure 1. Typical ANN Components and Structure.

While the original inspiration was the biological brain, an ANN can also be regarded as a statistic, and there are many strong and important parallels between the field of statistics and the field of ANN [3, 12]. The process of training the ANN using a data set is an analog to computing a vector valued statistic from that data set. Just as a regression equation's coefficients (viz., slopes and intercepts) are calculated by minimizing squared error over the data set, ANN weights are determined by minimizing error over the data set. However, there are also important dissimilarities between statistics and ANN. ANN have many free parameters (i.e., weighted connections). A ANN with five inputs, an intermediate (*hidden*) layer of five neurons and a single output has 36 trainable weights, where a simple multiple linear regression would have six (five slopes and an intercept). ANN can accommodate redundant free parameters rather well, but there is significant danger in overfitting an ANN model [12]. An overfitted ANN would be extremely dependent on the data set (*sample*) used to build it, and would poorly reflect the underlying relationship (*population*). Therefore, validation of ANN using data not used in training is essential.

An important property of ANN, under certain conditions, is that they are universal approximators [10, 13, 19]. This means that the bias associated with choosing a functional form, as is done in regression analysis when a linear relationship is selected, is eliminated. This is a substantial advantage over traditional statistical prediction models, as the relationship between network topology and reliability is highly non-linear with significant, but complex, interactions among the links.

In this paper, ANN are developed, or trained, based on the all-terminal reliability of a set of possible network topologies and link reliabilities for a given number of nodes. The resulting ANN is used to estimate network reliability as a function of the link reliabilities and the design configuration during the search for optimal design. In this way, multiple estimates of network reliability are available without calculation of reliability for each new candidate design. A disadvantage of using ANN as a reliability evaluator is that the reliability prediction is only an estimate that may be subject to bias and/or variance depending on the adequacy of the ANN. A similar approach was used for design of series-parallel systems when considering cost and reliability. This is described in [5], but is fundamentally different because reliability of series-parallel systems can be exactly calculated quite easily with closed form mathematical expressions.

One important way to evaluate the efficiency of the combined ANN / optimization approach is to consider the number of reliability calculations required to train and validate the ANN compared to the number of reliability calculations saved by use of the ANN. In [5], for the first series-parallel design problem considered, 9600 reliability calculations were required to train and validate the ANN, which then saved approximately 50,000 reliability calculations during a single optimization. If a single optimization for a single design problem were the extent of the effort, then the implicit savings of approximately five times fewer reliability calculations was not dramatic considering that the saving was obtained at the expense of precision in the reliability estimation.

However, as additional problems are considered, the benefits of this approach become increasingly clear. A second problem used the same ANN so no additional reliability calculations were required, yet 50,000 more reliability calculations were saved for each single optimization. Collectively, the ANN approach now led to approximately an order of magnitude fewer reliability calculations. The benefit continued to increase as more design problems were analyzed (because the ANN was designed to be a universal approximator for series-parallel systems). Ultimately, the “cost” of the initial training set becomes inconsequential if the ANN is used repeatedly to analyze additional design problems. This reasoning also applies to the network reliability calculation considered in this paper. The ANN can estimate the reliability of any connected topology for a variety of link reliabilities for any set of a fixed number of nodes.

3. TRAINING AND VALIDATING THE NEURAL NETWORKS

A backpropagation training algorithm [18] was selected because of its powerful approximation capacity and it applicability to both binary and continuous inputs. The problem studied was where the reliability of every possible link is identical. This is a common assumption in the network design literature. It was desired to develop an ANN that could handle various values of link reliability although this complicates the estimation task. The number of nodes in the network for a given ANN was fixed. The inputs to the ANN were:

1. The architecture of the network as indicated by a series of binary variables (x_{ij}). The length of the string of 0’s and 1’s is equal to $\frac{N(N-1)}{2}$ where N is the number of nodes.

2. The link reliability.

3. The calculated upper bound using the method of [14] of the network.

The upper bound calculation, while adding some computational effort, significantly improved the estimation precision of the ANN. Without the upper bound as an input, the errors of the ANN reported in the next section were nearly doubled. The output of the ANN was the estimated all-terminal network reliability. For the training and validation sets, the target network reliability, was the exact value as calculated using the backtracking technique of [2].

A node size of ten was chosen to investigate the approach of this paper. Link reliabilities of 0.80, 0.85, 0.90, 0.95 and 0.99 were used. A set of 750 network topologies were randomly generated (ensuring each network formed at least a minimum spanning tree, i.e., R(**x**) > 0) with 150 observations of each link reliability. The upper bound of each network and the exact network reliability were calculated to use as an input and as the target output, respectively. After preliminary experiments, a network architecture of 47 (45 possible arcs, the link reliability and the network upper bound) inputs, 47 hidden neurons in one hidden layer and a single output was used. The data set was divided using a five-fold cross validation technique so that five validation ANN were trained and one final application ANN was trained. The five validation ANN used 4/5’s of the data set for training and the remaining 1/5 as testing, where the testing set changed with each validation ANN. The final application ANN was trained using all 750 members of the data set and the validation is inferred using the cross validation ANN. (For a full presentation on the cross validation approach as applied to ANN, see [16].)

A second strategy using a specialized ANN for highly reliable networks was also employed. Because most actual network topology designs will be highly reliable, it is important that the reliability estimation be precise when R(**x**) ≥ 0.90. If the first ANN (just described) estimated a reliability of 0.90 or greater, the network topology, the link reliability and the upper bound were input to the second, specialized ANN, as shown in Figure 2. This ANN was trained on 250 randomly generated topologies (using the same five link reliabilities) that had actual network reliabilities of 0.90 or greater. As in the general ANN, there were equal number (50) observations of each link reliability in the data set. Also as in the general ANN, a five-fold cross validation procedure was used for ANN training and validation. The ANN architecture was the same as the first network. Using the network reliability estimate from the first ANN as an input to the specialized ANN was tried, but this did not improve predictive performance of the specialized ANN.

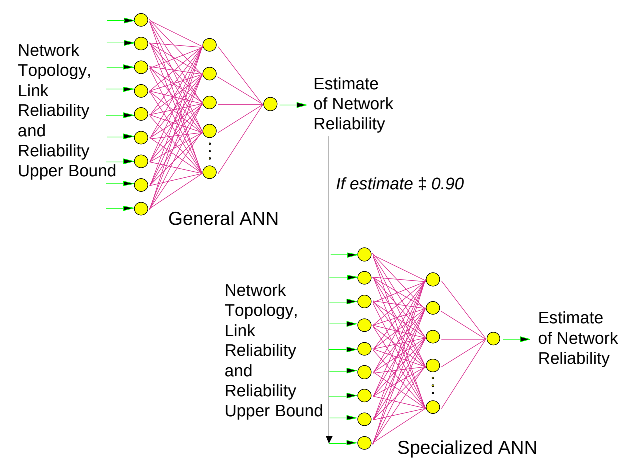

Figure 2. The Hierarchy of a General ANN and a Specialized ANN.

4. COMPUTATIONAL RESULTS

Tables 1 and 2 give the five-fold results in root mean squared error (RMSE) for the general ANN and the specialized ANN, respectively. It can be seen that the ANN estimations always improve upon the upper bound estimates, sometimes significantly. Furthermore, the errors of the specialized ANN are much less than that of the general ANN, allowing a more precise network reliability estimation for those topologies that are likely to be considered the best.

Table 1. Errors for the General Neural Network.

Fold	RMSE Training	RMSE Testing	RMSE Upperbound
Set1	0.03672	0.04260	0.08875
Set2	0.03073	0.05004	0.08954
Set3	0.03444	0.03067	0.07158
Set4	0.03123	0.05666	0.07312
Set5	0.03173	0.05131	0.08800
Average	0.03297	0.04626	0.08220

Table 2. Errors for the Specialized Neural Network.

Fold	RMSE Training	RMSE Testing	RMSE Upperbound
Set1	0.00664	0.00688	0.01232
Set2	0.00583	0.01271	0.01371
Set3	0.00630	0.00892	0.00908
Set4	0.00629	0.00795	0.00927
Set5	0.00555	0.01125	0.01598
Average	0.00612	0.00954	0.01207

Figure 3 shows an example of one of the five-fold validations comparing the estimation of the ANN on the test set with the actual reliability while Figure 4 shows the same for the specialized ANN. It can be seen that the predictions of the ANN are unbiased and are quite precise. Where the general ANN is less precise (at $R(\mathbf{x}) \geq 0.90$), the specialized ANN does a much better job. Figures 5 and 6 give the prediction of the application ANN (general and specialized, respectively) along with the exact reliability from backtracking and the upperbound. The better performance of the ANN relative to the upperbound is confirmed along with the precision of the ANN estimate relative to the actual reliability. The MAE of the application general ANN is 0.036 and the MAE of the application specialized ANN is 0.007. Of course, these errors may be positive or negative since an ANN is an unbiased estimator while the upperbound errors will always be positive.

Considering the relatively small training set used (only 150 observations of each link reliability for the general ANN and only 50 observations of each link reliability for the specialized ANN), the computational benefits of the approach become apparent. The ANN approach can now be used for any network design problem of ten nodes with these five link reliabilities. Considering that each ten node reliability design problem (i.e., those with a specified link reliability) has a search space of 3.5×10^{13} an optimization procedure that examined only a very small fraction of possible designs would still require millions of network reliability calculations. The tiny amount of network topologies needed for ANN training and validation (150 for the general and 50 for the specialized of 3.5×10^{13}) give an indication of the power of the method. Increasing the training and validation set size will almost certainly improve the estimation accuracy of the ANN while a decrease in size will worsen the estimates.

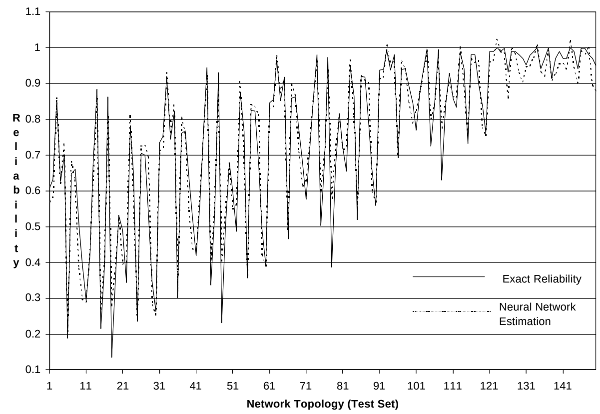

Figure 3. General Neural Network Estimation of Reliability versus Actual Reliability on the Fifth Fold.

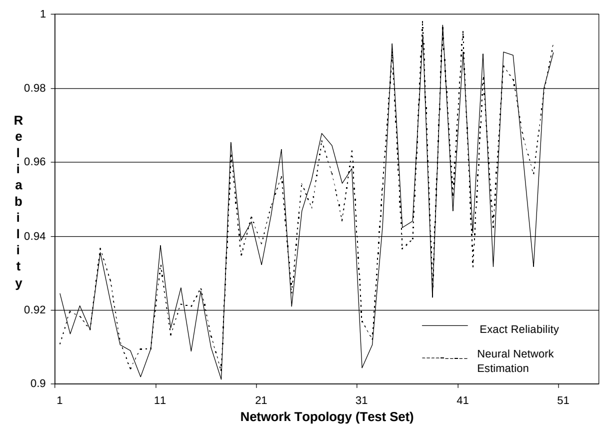

Figure 4. Specialized Neural Network Estimation of Reliability versus Actual Reliability on the First Fold.

5. CONCLUSIONS AND DISCUSSION

The ANN approach to estimating all-terminal reliability worked well. Using an extremely small fraction of the possible network topologies for a ten node problem, a general ANN and a specialized ANN are trained and validated. Subsequent use of the ANN during network design optimization will be basically computationally "free". The recommended approach is to use the ANN estimation during the design optimization for all designs considered and then exactly calculate the network reliability on only the optimal design, or the few best designs. In this way, almost all of the computational effort of reliability calculation is eliminated while maintaining a workable design optimization method.

It is likely that confining the ANN to only a single link reliability would further improve estimation precision, however this would reduce flexibility during the design phase. Likewise, expanding the ANN to consider links of differing reliability within a single design would improve design flexibility but complicate the ANN estimation task. The authors have completed studies on this last approach and will be reporting results in a subsequent publication.

ACKNOWLEDGEMENT

Alice E. Smith gratefully acknowledges support from the National Science Foundation CAREER grant DMI-9502134.

REFERENCES

[1] K. K. Aggarwal, Y. C. Chopra and J. S. Bajwa, "Topological layout of links for optimising the overall reliability in a computer communication system," *Microelectronics and Reliability*, Vol. 22, 1982, pp. 347-351.

[2] M. Ball and R. M. Van Slyke, "Backtracking algorithms for network reliability analysis," *Annals of Discrete Mathematics*, Vol. 1, 1977, pp. 49-64.

[3] B. Cheng and D. M. Titterington, "Neural networks: A review from a statistical perspective," *Statistical Science*, Vol. 9, 1994, pp. 2-54.

[4] Y. C. Chopra, B. S. Sohi, R. K. Tiwari and K. K. Aggarwal, "Network topology for maximizing the terminal reliability in a computer communication network," *Microelectonics & Reliability*, Vol. 24, 1984, pp. 911-913.

[5] D. W. Coit and A. E. Smith, "Solving the redundancy allocation problem using a combined neural network / genetic algorithm approach," *Computers and Operations Research*, Vol. 23, 1996, pp. 515-526.

[6] D. L. Deeter and A. E. Smith, "Heuristic optimization of network design considering all-terminal reliability," *Proceedings of the Reliability and Maintainability Symposium*, 1997, pp. 194-199.

[7] B. Dengiz, F. Altiparmak and A. E. Smith, "Efficient optimization of all-terminal reliable networks using an evolutionary approach," *IEEE Transactions on Reliability*, Vol. 46, 1997, pp. 18-26.

[8] B. Dengiz, F. Altiparmak and A. E. Smith, "Local search genetic algorithm for optimal design of reliable networks," *IEEE Transactions on Evolutionary Computation*, Vol. 1, 1997, pp. 179-188.

[9] G. S. Fishman, "A Monte Carlo sampling plan for estimating network reliability," *Operations Research*, Vol. 34, 1986, pp. 581-594.

[10] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, Vol. 2, 1989, pp. 183-192.

[11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco, 1979.

[12] S. Geman, E. Bienenstock and R. Doursat "Neural networks and the bias/variance dilemma," *Neural Computation*, Vol. 4, 1992, pp. 1-58.

[13] K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, Vol. 2, 1989, pp. 359-366.

[14] R.-H. Jan, "Design of reliable networks," *Computers and Operations Research*, Vol. 20, 1993, pp. 25-34.

[15] R.-H. Jan, F.-J. Hwang, and S.-T. Chen, "Topological optimization of a communication network subject to a reliability constraint," *IEEE Transactions on Reliability*, Vol. 42, 1993, pp. 63-70.

[16] J. M. Twomey and A. E. Smith, "Bias and variance of validation methods for function approximation neural networks under conditions of sparse data," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, Vol. 28, August 1998, in print.

[17] A. N. Venetsanopoulos and I. Singh, "Topological optimization of communication networks subject to reliability constraints," *Problem of Control and Information Theory*, Vol. 15, 1986, pp. 63-78.

[18] Paul J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, unpublished Ph.D. Thesis, Harvard University, 1974.

[19] H. White, "Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings," *Neural Networks*, Vol. 3, 1990, pp. 535-549.

[20] M.-S. Yeh, J.-S. Lin and W.-C. Yeh, "A new Monte Carlo method for estimating network reliability," *Proceedings of the 16th International Conference on Computers & Industrial Engineering*, 1994, pp. 723-726.

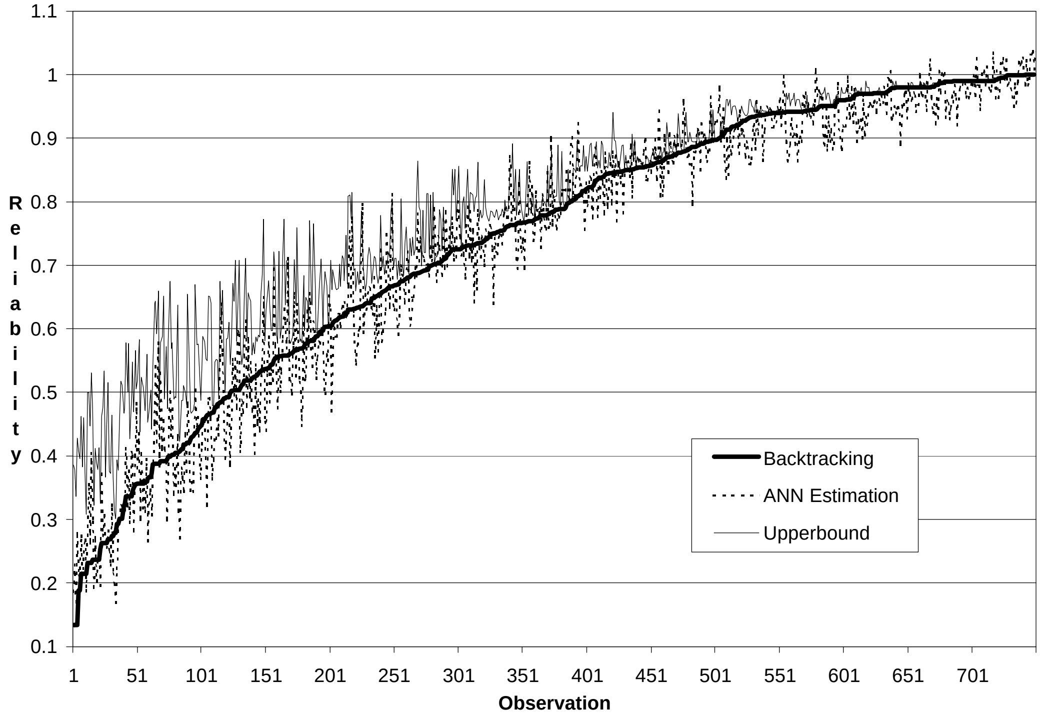

Figure 5. Comparison of Actual Reliability (Backtracking), the Upperbound and the General ANN Estimate.

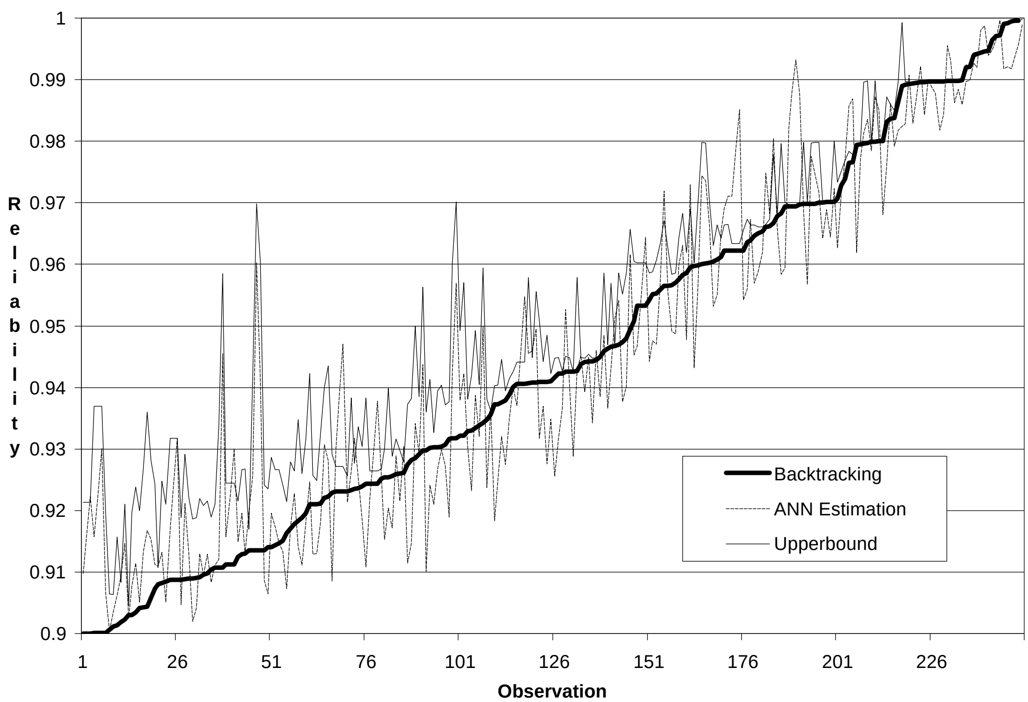

Figure 6. Comparison of Actual Reliability (Backtracking), the Upperbound and the Specialized ANN Estimate.