IJCSC International Journal of Computer Science and Communication Vol. 2, No. 1, January-June 2011, pp. 79–87

A NEW ANN METHOD FOR MEASURING OVERALL RELIABILITY AND PERFORMANCE IN GROWING COMPUTER NETWORKS WITH STATIC AND VARIABLE CONNECTIONS

Baijnath Kaushik[1], Navdeep Kaur[2], Amit Kumar Kohli[3]

[1]Department of Computer Science, Krishna Engineering College, Ghaziabad, INDIA, *E-mail: bkaushik99@gmail.com*

[2]Department of IT, Chandigarh Engineering College, Chandigarh, INDIA, *E-mail: nrwsingh@yahoo.com*

[3]Department of Electronics and Telecommunication, Thapar University, Patiala, INDIA, *E-mail: drkohli_iitr@yahoo.co.in*

ABSTRACT

This paper presents a computational study on the performance of reliability measures by using optimized ANN for computer networks with fixed and varying link reliabilities. This paper focus on the design of minimum cost reliable computer networks when a set of nodes, their topology, and links are given to connect them. A comparative study of various approaches for evaluating reliabilities has been studied such as Monte Carlo simulation methods and upper and lower bounds to bound reliability. The network design problem is difficult when overall reliability measure is considered through these methods. The objective is to design a minimum cost reliable networks that meets minimum reliability requirements. Therefore, for optimal network design, an optimized ANN is used for reliability measures. An optimal ANN is constructed, trained and validated using topologies, fixed and varying link reliabilities, and upper-bound on reliability as inputs to produce overall reliability as output.

Keyword: Computer Networks, Link reliabilities, Optimized ANN, Network reliability, Upper-bound, Lower-bound.

1. INTRODUCTION

The measurement of overall reliability in computer networks of growing size is NP—hard problem, the computational effort required is growing exponentially with growing network size [4, 10, 25] in terms of nodes and links in the network. An optimal network design is also difficult as it requires reliability calculation for each topology. The objective of this proposal is to design an optimal computer networks that have minimum costs and minimum reliability requirement and is relevant for many real world applications such as in design of telecommunication networks [9, 26, 31], computer networks [16, 17, 19], oil and gas lines [22], water systems [24]. The purpose is to construct an optimal network design when number of nodes, their topology, and links are given to connect them.

An optimized ANN [41] consists of: A general neural network scans all possible network topologies on given number of nodes for reliability measures then a specialized neural network for highly reliable network design is considered [33—45]. Both neural networks with fixed and varying link reliabilities re studied in [33—38, 41]. Results are grouped using cross-validation method show that the optimized ANN gives precise measures for reliability than the upper-bound [29, 32] and Monte-carlo simulation method [11, 20]. Results shows that the optimized ANN produces optimal network designs and reliability measures at reasonable computational cost.

1.1 Problem Definition

This paper discuss the problem of how to design growing computer networks so that cost and reliability is optimized. The design problem solved by optimized ANN [32, 33, 36, 39—45] is significant of real design problems. Cost and reliabilities (links) are two important considerations when designing a real world networks which is applicable in many industrial applications such as WAN, LAN, and data networks in industrial facility. In any network design, following are the problem assumptions must be considered.

1. Location of each network node is given.
2. Nodes are perfectly reliable.
3. Link costs and reliability 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. Failure of links is independent of network design.
8. No repair is considered.

The design optimization problem for a minimum cost networks that meets minimum reliability requirement can be expressed mathematically as follows:

$$\text{Minimize } Z(X) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} C_{ij} X_{ij}$$

$$\text{And} \qquad R(X) \geq R_0 \tag{1}$$

Where N is the number of nodes; (i, j) is the link between nodes i and j; X_{ij} is the decision variable, $X_{ij} \in \{0, 1\}$ for networks with fixed reliability; X is the link topology of $X_{12}, \ldots, X_{ij}, \ldots, X_{N-1}, \ldots, _N$; $R(X)$ is the reliability of X; R_O is the network reliability requirement; Z is the objective function and C_{ij} is the cost between (i, j).

The complexity of possible network topology in terms of space size complexity is given

$$K \frac{(|N| \times (|N|) - 1)}{2} \tag{2}$$

Where K is the choices for the links is to be connected in the growing networks. For fixed links, there are always two choices – 0 for no link present and 1 for link present – between any pair of nodes i and j. For varying link, we can choose a single link connecting two link or two nodes or more. There are several design options. For example, a 10 node network ($N = 10$) with fixed links ($k = 2$) has $3.5 * 10^{13}$ possible designs. A network with ($N = 10$) and with ($K = 5$) varying links choices has 10^{35} possible designs [32–38]. For a growing network size, it is practically difficult to calculate the exact network reliability. Therefore, an optimization procedure must be used to calculate exact reliability.

The design of network is difficult when overall reliability is considered. It is defined as the probability that all nodes communicate with every other nodes. The reliability is defined as p, and a non-zero reliability is $q = 1 - p$, at any time, only some links in a topology X may be operational. A state of a topology X is represented by a sub-graph (N, X'), where X' represents set of operational specific links such that $X' \subseteq X$. The network reliability for the state graph $X' \subseteq X$ is given by:

$$R(X) = \Sigma_{\Omega} \left[\Pi_{j \in x'} \, p(x_j) \right] \left[\Pi_{j \in (x\, x')} \, q(x_j) \right] \tag{3}$$

Where Ω = all operational states in the state graph.

Another objective is to minimize the cost of the network design problem from Eq. (1). Costs can include material costs of the cabling, installations costs such as trenching or boring, purchase of land or right way costs, and connection or terminal costs. These costs are assumed as unit costs because they depend on the length of the links. In many literature, cost is assumed as fixed weights [6, 9].

There are two main reliability measures have been studied, all-terminal (also called overall reliability) and source-sink (two-terminal reliability). The overall network reliability is concerned with the ability of each and every network node to be able to communicate with every other nodes in the network through some non-specified path. This means that network must form at least a minimum spanning tree. The two-terminal reliability is concerned with the ability of source node (pre-specified) to communicate with the sink node (also pre-specified) through some non-specified path. The problem of measuring the reliability in computer network design is an active area of research. There are four approaches have been discussed in the literature - exact calculation through analytic methods, estimation through variations of Monte-carlo simulations [11, 20], upper or lower bounds of reliability [29, 32], and easily calculated direct method for reliability[33]. The issue of measuring reliability of network is so important for optimal design of computer network.

The most common objective is to design a computer by selecting a subset of possible links so that network reliability is maximized and maximum cost constrained is met. But, in many situations, it makes more sense is to minimize cost subject to a minimum network reliability constraint which can be can measured from Eq. (3). There are some other constraints mentioned in the literature, such as minimum node degree or maximum links length allowed in the network. The objective of this proposal is to find the minimum cost network architecture that meets pre-specified minimum network reliability. The equation for the objective is given as follows:

$$Min: C(X)$$

That is

$$R(X) \geq R_0 \tag{4}$$

2. PRESENT WORK

This proposal explain the various steps for measuring reliability in growing with fixed and varying connections using optimized ANN.

2.1 Neural Networks for Reliability Measures

Artificial neural network [15, 21] is used as a function approximation or a non-linear estimation technique which takes set of input values and it produces an output value.

In this paper, ANN are developed, trained, based on the overall terminals reliability of a very small 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 topology during search for the optimal design. In this way, estimates of the reliability of numerous topologies are available without costly calculation or simulation.

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 [15, 21]. The functionality of using an ANN estimation of reliability during optimal network design is tested by comparing it to an easily calculated upper-bound and a computationally expensive exact calculation.

2.2 An Optimal Design of Neural Network

An optimal design of ANN [41] is considered here which is divided into two stages: experimental condition setup and optimization process as shown in Fig. 1. In optimized ANN, training of neural networks is iterated until a given condition is satisfied. The ANN process consists of nine steps, as described below:

Step 1: Specifies the design parameters that are same in all iterations are the number of network input and output signals, input and target patterns, and the number of network layers. In this method, the number of hidden nodes and learning rate can be defined as design parameters.
Step 2: Prepares an orthogonal array as shown in table 1 based on the number of design parameters and their levels, then each orthogonal array column is allocated to a design parameter.
Step 3: Initial values of parameters such as connection weights, learning rate (α), number of trials to achieve epochs, and the value for tolerance are assigned.
Step 4: The neural networks parameters are set as per the value of orthogonal array listed in the table 1.
Step 5: Our design methods treats training of ANN as an experiment. The neural network is trained using the back-propagation training algorithm.
Step 6: The next step is to conduct Analysis of Variance (ANOVA) for results with experimental setup of orthogonal array data.
Step 7: This step is used to choose a significant parameters for the neural networks.
Step 8: Based on the selection of significant parameter, the neural network is reconfigured and go back to step 4 for optimization process of neural network.
Step 9: If no significant parameters have been selected then stop the training process.

Fig. 1: Algorithm for Designing an Optimal Neural Network

In this paper, an experimental setup is done for optimizing the design of an ANN [41] which is described in the above algorithm. Experiment is based on fractional factorial experiments and uses orthogonal arrays efficiently. In complete experimental setup, all combinations of design parameter levels are tried, so the number of combinations increases exponentially with the number of design parameters increases.

This experiment deals with experimental results including errors due to ANOVA. ANOVA gives information on factorial effects and experimental error, i.e., error unrelated to any factor. Confidence of optimization is evaluated by the magnitude of experimental error, so Design of Optimization [41] optimizes design of ANN architecture and ensures its effectiveness. Table 1 shows an orthogonal array in which the number of design parameters (shown in column) is 5, the number of levels is 4, and the number of experiments (row) is 16. The table is denoted by $L_{16}\ (4^5)$.

Table 1
Orthogonal array $L_{16}(4^5)$

No.	Column 1	2	3	4	5
1	1	1	1	1	1
2	1	2	2	2	2
3	1	3	3	3	3
4	1	4	4	4	4
5	2	1	2	3	4
6	2	2	1	4	3
7	2	3	4	1	2
8	2	4	3	2	1
9	3	1	3	4	2
10	3	2	4	3	1
11	3	3	1	2	4
12	3	4	2	1	3
13	4	1	4	2	3
14	4	2	3	1	4
15	4	3	2	4	1
16	4	4	1	3	2

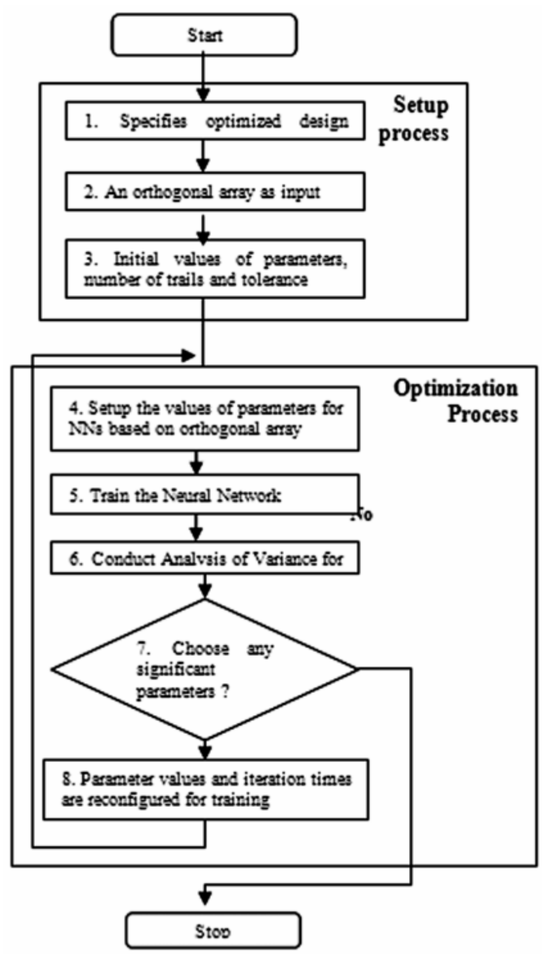

Fig. 2: The Design Approach for Optimizing ANN

The 5 columns to the right indicate combinations of levels of design parameters. In this experimental setup with orthogonal array, 16 experiments with different combinations of parameter levels are conducted. For example, consider 5 parameters be a, b, c, d, and e, corresponds to 5 columns in Table 1. Each parameter in the Table has 4 levels. For example, The levels for parameter a are defined as a_1, a_2, a_3, and a_4. The combinations of level in experiment is as follows (a_1, b_1, c_1, d_1, e_1) is (1, 1, 1, 1, 1). The experiment will be repeated with all level of combinations.

This experiment performs 64 trials, to measure an optimal levels of design. In full fractional experimental setup, the total trials taken are 1.7×10^{10} from [41] that reduces time as compared to previous method.

2.3 Training and Validation of Optimized ANN

A back-propagation training algorithm [1, 38, 39] was selected because of its powerful approximation capacity and its applicability to both binary and continuous inputs. The back-propagation algorithm minimizes the squared error between the ANN output and the target. A hyperbolic activation function was used in all neurons to set the learning rate of hidden neurons and a learning rate for output neurons.

A standard ANN software package, neural works explorer [30], was used to perform training and validation of neural networks for: Networks with fixed and varying link reliabilities. After preliminary experiments, the architecture of ANN consists of 107, 70, and 1 neurons in input, hidden and output layers, respectively. The ANN models were trained for 300000 epochs, that is 300000 passes through the training set, with the normalized cumulative delta rule (learning rule) with 0.30 and 0.15 learning coefficients for the hidden layer and output layer, respectively, using Neuralworks Explorer software package [30].

All data sets are divided into five subsets to use the five-fold cross validation technique. The five-fold validation ANN used 4/5 of the data set for training and the remaining 1/5 data set for testing, where the testing set changed with each validation of ANN. A final application ANN is trained using all members of the data sets for each network size and its validation is inferred using the five-fold cross validation of ANN. This cross validation approach provides an unbiased and quite precise measurement of ANN performance on the population of network topologies.

3. COMPUTATIONAL PROCEDURE

3.1 Networks with Fixed Link Reliabilities

The problem of measuring reliability can be simplified by limiting the links chosen in a network topology with same reliability i.e., with $K = 2$, because the number of possible topologies grows exponentially with increase in K (refer to (1)). In this case, if $X_{ij} = 1$, the link is chosen for the network topology and if $X_{ij} = 0$, no link is present. To make the ANN more applicable to a variety of design problems, five different values of link reliability were chosen to be included in a single ANN.

The inputs to the ANN were:

1. The architecture of the network as indicated by a series of binary variables (X_{ij}).
2. The length of the string of 0′*s* and 1′*s* is equal to $(N(N-1))/2$.
3. The link reliability is chosen in between 0 and 1 may be (0.80, 0.85, 0.90, 0.95, 0.99).
4. The calculated upper-bound of network reliability using the method of [29, 32-33].

The upper-bound for reliability calculation is significantly improved using the Eq. (5) from [29, 32-33] is given below:

$$R(G) \le 1 - \left[\sum_{i=1}^{N} q_i^{d} \prod_{k=1}^{m_i} \left(1 - q_{k-1}^{d}\right) \prod_{k=m_1-1}^{i-1} \left(1 - q_k^{d}\right) \right] \qquad (5)$$

Where $m_i = \min(d_i, i-1)$, $i = 1, 2, \ldots, N$, $R(G)$: Reliability of G, p, q: Reliability and unreliability of a link; $p + q \equiv 1$, and d_i : the degree of (the number of links incident on) node i.

The output of the optimized ANN will be the overall reliability measurement for the computer networks. Here, we present the comparison of the measurement of reliability of each network topologies using Monte-carlo sampling method [11, 20] with the optimized ANN method [41]. The Monte Carlo algorithm and sampling plans is divided into two algorithms called NFA and BETA procedure as discussed in [11, 20] and shown in Figures 2-4. The steps of Monte Carlo algorithm as follows: first, NFA procedure is called to generate the probability distribution of the number of failed arcs ($P[n=k]$; k=0, .., N), then, BETA procedure is called m times (m=3000) in a loop to simulate β_k values depending on whether the topology instance is fully connected or not, for every instance with k number of failed arcs from 0 to N. Next, $R_j(G)$ values are computed for *each* j=0,..m. Finally, the reliability is estimated as the mean of $R_j(G)$ values.

Where; N: the number of links, $R(G)$: the reliability estimator of $R(G)$, $R_j(G)$: the reliability estimation value for $j = 1, 2,., m$, and n : the total number of failed arcs.

Data sets were generated using two different approaches [38]; random design and experimental design, which are explained as follows.

(*a*) **Random design 1:** The training data set called D1 includes equal number of network topologies for each link reliability. A stochastic depth-first search algorithm is used to build a spanning tree. Since it is possible to obtain $n^{(n-2)}$ spanning trees from a fully connected network, a sub set of different kinds of networks can be easily generated using the stochastic depth-first search algorithm [19, 28].

(*b*) Random Design 2: In this approach, the training data set called $D2$ is generated considering a predetermined number of network topologies for intervals of each system reliability, as given in Table 2. A stochastic depth-first search algorithm was also used to generate network topologies as in $D1$.

(c) Experimental Design: To obtain more representative data sets of the problem space an experimental design approach is used considering connectivity and link reliability together as design points to generate training data sets. It is obvious that system reliability increases with increasing connectivity. Connectivity is the minimum number of links or nodes that must be removed from a network to break all paths between any pair of nodes [1-3]. Preliminary experiments showed that the network reliability is very close to 1 if the networks have five connectivity or more. Therefore, two different data sets, $D3$ and $D4$, with up to four connectivity and five, respectively, are generated. An equal number of network topologies is generated for each level of network connectivity. The sizes of $D3$ and $D4$ are as the same as $D1$ and $D2$ (1250 total). The labeling algorithm given in [17-19] is used to check the network connectivity for each generated network in these data sets.

3.2 Networks with Varying Link Reliabilities

A real world problem consideration is to allow links of varying reliability within a network topology. This greatly expand the number of possible topologies of a network, also complicates the network design problem and computation of overall reliability of network. For real world example, consider a network with links value is $K = 6$, that is, network can take any five reliability value or 0, which indicates that link is not present. For further clarification, for any network design problems, we can use any of five link reliabilities in any combination.

The inputs to ANN are:

1. The architecture of network is given by a series of real-value variables (X_{ij}).
2. The length of string is given by $(N(N-1))/2$.
3. The Konak and Smith [29, 32] method is used to calculate the upper-bound reliability.

The upper-bound for reliability calculation is significantly improved using the Eq. (6) from [29, 32] is given below:

$$R(X) \le 1 - \left[\sum_{i=1}^{N} \left(\prod_{(k,i)\in E_i} (1-p_{ki}) \right) \prod_{j=1}^{N} \left(1 - \frac{\prod_{(k,j)\in E_j} (1-p_{kj})}{(1-p_{ij})} \right) \right] \quad (6)$$

Where p is the reliability of a link and E is the set of links connected to a given node. The output of the ANN will be the measurement of overall network reliability. The target network reliability of each network is estimated using the Monte Carlo method which was explained in Section 3.1. The network designs for varying link reliabilities are generated using the same manner used for generating the fixed data sets and also named the same, $D1$, $D2$, $D3$ and $D4$. The reliability value of each link in a network is used as an input. For example, (0, 0.80, 0, 0.85, 0.95, 0.95, 0.99) can be a representation of a network topology. This representation results in 301 for a 25 node network.

4. COMPUTATIONAL RESULTS

4.1 Networks with Fixed Link Reliabilities

Table 3 gives five-fold cross validation results in root mean squared error (RMSE) for the ANN models built with the data sets with homogenous link reliabilities. The error, which is used to calculate 0.0000* difference between Monte Carlo and ANN estimations of the network reliability. When the RMSE columns of training and testing sets are examined, it can be seen that the ANN models built with $D4$ generate minimum average RMSE values of 0.02809 on the training, and 0.03639 on the testing sets. Ordering all data sets from the best to the worst according to their average RMSE values of testing sets, the sequence of $D4$, $D3$, $D1$, $D2$ is obtained. Upper-bound RMSE columns represent the RMSE of the upper-bound only (no ANN estimation) on the testing sets. It can also be seen that the ANN always improve upon the upper-bound estimates.

A statistical analysis is applied to test whether there are statistically significant differences between final ANN models of different pairs of data sets according to MAE. Since the homogeneity of variances and normality assumptions for the ANN models were not satisfied, the Kruskal-Wallis test [15], a nonparametric version of the ANOVA (Ana [38, 41]), is used. Based on the test there are significant differences between ANN models with a p value of < 0.000 at $\alpha = 0.05$ [38]. Table 4 shows pairs, mean differences and p values. As shown in this table that there are no statistically significant differences between the optimized ANN models built with $D4$ and $D3$, while other pairs are statistically significantly different [38, 41].

4.2 Networks with Varying Link Reliabilities

Similar comparisons and tests were carried out for networks with non-uniform link reliability to determine the effects of data sets for the ANN performance. Table 3 shows that the ANN models built with the $D4$ data set generate minimum average RMSE values of 0.03608 and 0.04510 on the training and testing sets, respectively. When data sets are ordered from the best to the worst according to their average RMSE values of testing sets, the sequence of $D4$, $D3$, $D1$, $D2$ is obtained. It is also shown that each ANN model estimation always improves upon the upper-bound estimates, sometimes significantly for this hard problem. It is found that there are significant differences between ANN models with a p value of < 0.000 at $\alpha = 0.05$ according to the Kruskal-Wallis test [15, 38, 41]. Because of the significant differences of the optimized ANN models according to MAE, comparisons between pairs of them were carried out using the sequence of $D4$, $D3$, $D1$, and $D2$. Table 4 shows pairs, mean differences and p-values at $\alpha = 0.01$. While there are no statistically significant difference between ANN models built on $D1$ and $D2$, other pairs are statistically significantly different [38].

According to these results the training data set generated considering up to five connectivity and link reliabilities exhibits the best performance. It can be seen that the ANN models give unbiased results very close to the Monte Carlo results. A statistical analysis based on the 7450 test observations considering the final ANN model shows that the ANN estimations are statistically closer to the Monte Carlo estimations than the upper-bound. Paired t tests between the ANN, the Monte Carlo method, the upper bound and the lower bound method is compared. The Monte Carlo method had a p-value of 0.829 and has a value with a mean difference of -2.18×10^{-4} and a p value < 0.0000 and mean difference of 0.0316, respectively [38].

Table 2
Distribution of Data Set D1

		Reliability				
Network	0.80	0.85	0.90	0.95	0.99	TNN*
$G = (25, 300)$	850	850	850	850	850	7450

Table 3
Distribution of Data Set D2

		System Reliability				
Network	0.80	0.85	0.90	0.95	0.99	TNN*
G = (25, 300)	425	425	425	425	425	7450

TNN*: Total number of networks in the data set.

Begin
1. Use procedure NFA to simulate the number of failed links, $P[n=k]$, for all $k = 0, 1, 2, \ldots, N$
2. For $j = 1$ to m do
  2.1. Use procedure BETA to simulate β_k for all $k = 0, 1, 2, \ldots, N$
  2.2. Compute $R_j(G) = \sum_{k=0}^{N} \beta_k \, P[n = k]$
3. Compute the final result; $R(G) = \sum_{j=1}^{m} R_j(G)/m$
End.

Fig. 3: Monte Carlo Algorithm

Begin
1. $n_k = 0$ for $k = 0, 1, 2, \ldots N$
2. For $j = 1 to\ m$ do
  2.1. For $I = 1\ to\ n$ do
    2.2.1. One random number u_i is generated from $U(0,1)$
    2.2.2. If u_i □ p then $X_i = 1$else $X_i = 0$.
  2.2. If the number of failed arcs is equal to k then $n_k = n_k + 1$
    For $k = 0, 1, 2, \ldots, N$.
3. $P[n = k]\ nk/m$ for $k = 0, 1, 2, \ldots, N$.
End.

Fig. 4: The NFA Procedure

Begin
1. For $k = 0$ to N do
  1.1. Arbitrarily chose k arcs from the given network topologies.
  1.2. If the network is connected after removing the chosen k arcs, then $\beta_k = 1$, else $\beta_k = 0$.
End.

Table 4
Five-fold Cross Validation Results for Static and Variable Connections

	Fixed Link Reliability			*Varying Link Reliability*		
Experiment		G = (25, 300)			G = (25, 300)	
Different		RMSE			RMSE	
Data sets	Training	Testing	Upper-bound	Training	Testing	Upper-bound
Results for *D*1						
1	0.03260	0.04201	0.07272	0.04774	0.06066	0.09277
2	0.03337	0.03937	0.07034	0.04944	0.05041	0.08848
3	0.03279	0.03995	0.06863	0.04966	0.05425	0.08439
4	0.03465	0.03935	0.07301	0.05056	0.05248	0.08362
5	0.03377	0.03603	0.06307	0.04857	0.05725	0.08916
Average	0.03343	0.03934	0.06955	0.04919	0.05501	0.08768
Results for D2						
1	0.04505	0.05279	0.07661	0.05152	0.06059	0.09947
2	0.03722	0.04815	0.08575	0.05247	0.05788	0.09095
3	0.03876	0.04036	0.07011	0.05132	0.05777	0.09979

Table Contd

Table 4 Contd.

4	0.03858	0.04001	0.07169	0.05123	0.05858	0.09379
5	0.03852	0.04776	0.07796	0.05144	0.05959	0.09813
Average	0.03963	0.04581	0.07642	0.05160	0.05888	0.09643
Results for D3						
1	0.02910	0.03853	0.06251	0.03967	0.05061	0.08505
2	0.02826	0.04213	0.06936	0.03897	0.04703	0.08184
3	0.02990	0.03279	0.07280	0.04050	0.05202	0.08030
4	0.02994	0.03340	0.07345	0.04111	0.04626	0.08473
5	0.02856	0.03940	0.07443	0.03988	0.04561	0.07621
Average	0.02915	0.03725	0.07051	0.04002	0.04831	0.08163
Results for D4						
1	0.02743	0.03595	0.05888	0.03679	0.04468	0.07657
2	0.02732	0.04129	0.07156	0.03588	0.04582	0.07606
3	0.02958	0.02966	0.06290	0.03525	0.04405	0.07132
4	0.02797	0.03713	0.06635	0.03750	0.04652	0.07388
5	0.02794	0.03790	0.06793	0.03499	0.04443	0.07613
Average	0.02809	0.03639	0.06552	0.03608	0.04510	0.07479

Table 5
Results of the Comparisons between Pairs of the Optimized ANN for Static and Variable Connections

	Fixed Link Reliability		*Varying Link Reliability*	
Pairs	*Mean Difference*	*p-value*	*Mean Difference*	*p-value*
D4-D3	-0.00105	1.22E-01	-0.00347	0.0018*
D4-D1	-0.00567	5.97E-10*	-0.01261	0.0000*
D4-D2	-0.00883	0.0000*	-0.01591	0.0000*
D3-D1	-0.00463	5.85E-07*	-0.00915	2.47E-12*
D3-D2	-0.00778	1.68E-14*	-0.01244	0.0000*
D1-D2	-0.00316	1.40E-03*	-0.00329	0.0102

*: Represents significant difference

5. CONCLUSIONS AND DISCUSSIONS

This paper proposed optimized ANN models [41] as an alternative way to measure the overall network reliability for computer networks. This model is developed and tested for 25 nodes with fixed and varying link reliabilities. The data sets used in this study in training of optimized ANN models generated with two approaches: random design and design of experiment (DOE) or experimental design [38, 41]. The results show that optimized ANN models built with the data generated by experimental design considering connectivity and link reliability produce more accurate results than those developed by random design and or by experimental design considering system reliability. The recommended approach is to use the ANN models to measure network reliability of all candidate designs during the topological optimization (network design) phase. Then, the network reliability for only the best design, or for a few good designs can be exactly calculated. In this way, the computational efforts of exact reliability calculation using Monte Carlo estimation can be reduced.

The neural network approach, an upper-bound approach and an exact backtracking calculation are compared for network design using simulated annealing for optimization and can be shown that the neural network approach gives superior designs at manageable computational cost. As future research, ANN models built in this work will be used in the genetic algorithm and other meta-heuristic algorithms [35, 37, 39] to obtain a more computational efficient design optimization method and will be applied to larger size networks.

REFERENCES

[1] P.J. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences", *Unpublished Ph.D thesis*, Harvard University, 1974.

[2] J.K. Cavers, "Cutset Manipulations for Communication Network Reliability Estimation, *IEEE Transactions on Communications*, **23**, 1975, pp. 569-575.

[3] M. Ball, R.M. Van Slyke, "Backtracking Algorithms for Network Reliability Analysis", *Annals of Discrete Mathematics*, **1**, 1977, pp. 49-64.

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

[5] K.K. Aggarwal, S. Rai, "Reliability Evaluation in Computer-communication Networks", *IEEE Transactions on Reliability*, **5**, 1981, pp. 30-32.

[6] K.K. Aggarwal, Y.C. Chopra, J.S. Bajwa, "Topological Layout of Links for Optimizing the Overall Reliability in a Computer Communication System", *Microelectronics and Reliability*, **22**, 1982, pp. 347-351.

[7] S. Rai, "A Cutset Approach to Reliability Evaluation in Communication Networks", *IEEE Transactions on Reliability*, **31,** 1982, pp. 428-431.

[8] J.S. Provan, M.O. Ball, "The Complexity of Counting Cuts and of Computing the Probability that Graph is Connected", *SIAM Journal of Computing*, **12**, 1983, pp. 777-788.

[9] 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", *Microelectronics and Reliability*, **3**, 1984, pp. 911-924.

[10] A.N. Venetsanopoulos, I. Singh, "Topological Optimization of Communication Networks Subject to Reliability Constraints", *Problem of Control and Information Theory*, **15**, 1986, pp. 63-78.

[11] G.S. Fishman, "A Monte Carlo Sampling Plan for Estimating Network Reliability", *Operations Research*, **34**, 1986, pp. 581-594.

[12] D.E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", *MA: Addison-Wesley Publishing*, 1989.

[13] K. Funahashi, "On the Approximate Realization of Continuous Mappings by Neural Networks", *Neural Networks*, **2**, 1989, pp. 183-192.

[14] K. Hornik, M. Stinchcombe, H. White, "Multilayer Feedforward Networks are Universal Approximators", *Neural Networks*, **2**, 1989, pp. 359-366.

[15] H. White, "Connectionist Nonparametric Regression: Multilayer Feedforward Networks can Learn Arbitrary Mappings", *Neural Networks*, **3**, 1990, pp. 535-549.

[16] S. Geman, E. Bienenstock, and R. Doursat, "Neural Networks and the Bias/variance Dilemma", *Neural Computation*, **4**, 1992, pp. 41-58.

[17] R.H. Jan, F.J. Hwang, S.T. Chen, "Topological Optimization of a Communication Network Subject to a Reliability Constraint", *IEEE Transactions on Reliability*, **42**, 1993, pp. 63-70.

[18] R.H. Jan, "Design of Reliable Networks", *Computers and Operations Research*, **20**, 1993, pp. 25-34.

[19] F.N. Abuali, D.A. Schoenefeld, and R.L. Wainwright, "Designing Telecommunications Networks using Genetic Algorithms and Probabilistic Minimum Spanning Trees", *Proceedings of the 1994 ACM Symposium on Applied Computing*, 1994, pp. 242-246.

[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 and Industrial Engineering*, 1994, pp. 723-736.

[21] B. Cheng, D.M. Titterington, "Neural Networks: a Review from a Statistical Perspective", *Statistical Science*, **9**, 1994, pp. 42-54.

[22] K. Ida, M. Gen, and T. Yokota, "System Reliability Optimization of Series-parallel Systems using a Genetic Algorithm", *Proceedings of the 16th International Conference on Computers and Industrial Engineering*, 1994, pp. 349-352.

[23] D.A. Savic, and G. A. Walters, "An Evolution Program for Pressure Regulation in Water Distribution Networks", *Engineering Optimization*, **24**, 1995, pp. 197-219.

[24] S. Pierre, M.A. Hyppolite, J.M. Bourjolly, and O. Dioume, "Topological Design of Computer Communication Networks using Simulated Annealing", *Engineering Applications of Artificial Intelligence*, **8**, 1995, pp. 61-69.

[25] D.W. Coit, and A.E. Smith, "Solving the Redundancy Allocation Problem using a Combined Neural Network/genetic Algorithm Approach", *Computers and Operations Research*, **23**, 1996, pp. 515-526.

[26] 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.

[27] B. Dengiz, F. Altiparmak, and A.E. Smith, "Efficient Optimization of All-terminal Reliable Networks using an Evolutionary Approach", *IEEE Transactions on Reliability*, **46**, 1997, pp. 18 - 26.

[28] B. Dengiz, F. Altiparmak, and A.E. Smith, "Local Search Genetic Algorithm for Optimal Design of Reliable Networks", *IEEE Transactions on Evolutionary Computation*, **1**, 1997, pp. 179-188.

[29] A. Konak, and A.E. Smith, "A General Upper-bound for All-terminal Network Reliability and its uses", *Proceedings of the Industrial Engineering Research Conference*, Banff, Canada, May 1998, CD Rom Format.

[30] Neuralworks: Reference Guide and Software from http://www.neuralware.com, NeuralWare, Pittsburgh, PA, Various Years.

[31] D.L. Deter, and A.E. Smith, "Economic Design of Reliable Networks", *IIE Transactions*, **30**, 1998, pp. 1161-1174.

[32] C.S. Ratana, and A.E. Smith, *Estimation of All-Terminal Network Reliability using an Artificial Neural Network*, *Computers and Operations Research*, June 1999.

[33] C.S. Ratna, A. Konak, and A.E. Smith, "Estimation of All Terminal Reliability using an Artificial Neural Network", *Computers and Operations Research*, **29** **(7)**, 2002, pp. 849-868.

[34] A. Fulya, D. Berna, and A.E. Smith, "A Comparison of Memetic Algorithms and Other Metaheuristic Hybrids for Reliability Optimization Problem", *Poster Proceedings of the Fifth International Conference Adaptive Computing in Design and Manufacture (ACDM)*, University of Exeter, UK, 2002, pp. 5-9.

[35] F. Altiparmak, B. Dengiz, and A.E. Smith, "Optimal Design of Reliable Computer Networks: A Comparison of Meta-heuristics", *Journal of Heuristics*, **9** **(6)**, 2003, pp. 471-487.

[36] B. Dengiz, F. Altiparmak, A.E. Smith, "Performance Evaluation of Generalized Neural Network For All-Terminal Network Reliability Estimation", *Informs,* October, 18-22, 2003.

[37] A. Konak, and A.E. Smith, "Designing Resilient Networks using a Hybrid Genetic Algorithm Approach", *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, June 25-29, 2005.

[38] B. Dengiz, F. Altiparmak, "A Computational Study on the Performance of All-terminal Reliability Estimation of Neural Networks for Communication Networks with Homogeneous and Heterogeneous Link Reliabilities", *Submitted to Microelectronics and Reliability*, 2005.

[39] L.M.L. Gen, "A Self-controlled Genetic Algorithm for Reliable Communication Network Design", *Evolutionary Computation IEEE Congress*, **0-00**, 2006, pp. 640-647.

[40] B. Dengiz, C. Alabas, and O. Dengiz, "A Tabu Search Algorithm for Neural Networks Training", *Journal of Operation Research*, In Press, 2007.

[41] E. Inohira, and H. Yokoi, "An Optimal Design Method for Artificial NN by the DOE Method", *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **11 (6)**, 2007, pp. 593-594.

[42] B. Dengiz, C. Alabas, O. Dengiz, "Optimization of Manufacturing Systems using Neural Network Metamodel with a New Training Approach", *Journal of Operation Research Society*, In Press, 2007.

[43] Springer, "A Chapter on Network Reliability Optimization: Handbook of Optimization in Telecommunications", *Mathematics and Statistics*, Issue 10, 1007, pp. 735-760, 2008.

[44] F. Altiparmak, B. Dengiz, and A.E. Smith, "Performance Evaluation of Generalized Neural Network For All-Terminal Network Reliability Estimation", *IEEE Transactions on Reliability*, 2008.

[45] F. Altiparmak, B. Dengiz, "Cross Antrophy Approach to Design of Reliable Networks", *European Journal of Operation Research (EJOR)*, In Press, 2008.