PREDICTION OF HIGHWAY BRIDGE PERFORMANCE BY ARTIFICIAL NEURAL NETWORKS AND GENETIC ALGORITHMS

Onur B. Tokdemir[1], Cemal Ayvalik[2], and Jamshid Mohammadi[3]

Project Manager, The Rise Group International, 120 S. LaSalle, Suite 1750, Chicago, IL, 60603 USA, Otokdemir@risegroup.com.
[2] *Research Assistant, Department of Civil and Architectural Engineering, Illinois Institute of Technology, Chicago IL 60616 USA*
[3] *Professor and Chairman, Department of Civil and Architectural Engineering, Illinois Institute of Technology, Chicago IL 60616 USA*

Abstract: Bridge management systems (BMS) comprise various techniques need to help make decisions on the type of works that need to be performed to maintain the serviceability of a bridge and to extend its useful life. These decisions rely on current and future bridge conditions therefore it is essential for a BMS to accurately predict the future bridge performance, or in other words to assess the extent of bridge deterioration. Numerous deterioration models are reported in the literature. Most of these methods were developed using probabilistic approaches ranging from Markovian methods to regression techniques with various levels of detail. While offering mostly marginal improvements, such methods increase the complexity of the procedures and level of expertise needed. Besides, high reliance of these methods on historical data, which are likely to contain missing information, reduces the chances for a reliable model. The ability of learning in Artificial Intelligence (AI) methods provides promising results in modeling and forecasting even in the existence of non-linear complex relationships. Furthermore, easier use of AI tools provided by today's software makes AI methods even more attractive. In this study two AI tools, artificial neural networks (ANN) and genetic algorithms (GA), are utilized to develop models to predict bridge sufficiency ratings using current geometrical, age, traffic, and structural attributes as explanatory variables. Data is acquired from California Department of Transportation through the Internet and it includes 19120 structural bridge components owned and maintained by the State of California. The models developed by both ANN and GA provided promising and interpretable results. ANN models performed better when different models are constructed for different levels of sufficiency ratings. GA models outperformed ANN models while achieving a better goodness of fit even when using the whole data. However, remarkably prolonged training times for GA models might be considered as the only disadvantage for this type of application.

Keywords: Bridge Management System, Artificial Neural Networks, Genetic Algorithms

1. INTRODUCTION

According to the National Bridge Inventory (NBI), there are approximately 589,815 bridges in the US. Of these bridges, 104,612 (23.8%) are categorized as substandard [1]. About 78,000 have inadequate (poor or worse) condition ratings, 53,300 have inadequate structural appraisal ratings, and 3,100 have inadequate waterway ratings. Further categorization reveals that, of the 78,000 structures with poor or worse condition ratings, 36,000 have poor deck conditions, 40,000 have poor superstructure conditions, and 50,000 have poor substructure conditions [2].

The implementation of bridge management systems (BMSs) is required by the Intermodal Transportation Efficiency Act (ISTEA) of 1991 and its predecessor Transportation Equity Act for 21st Century (TEA-21) of 1998. The bridge management system assists in determining the optimal time for an agency to execute improvement actions on a bridge, given the funds available. A BMS comprises various techniques need to help make decisions on the type of works that must be performed to maintain the serviceability of a bridge before lapsing into a unsafe state. The Federal Highway Administration (FHWA) defines a BMS as "an integrated set of formal procedures for directing or controlling all activities related to bridges" [3]. A BMS includes four basic components: data storage, cost and deterioration models, optimization models for analyzing, and updating mechanisms. The database connected to a BMS stores data from periodic field inspections. The bridge is divided into individual elements or sections of the bridge which are comprised of the same material and can be expected to deteriorate in the same manner. The condition of each element is reported according a condition state, which is a quantitative measure of deterioration on a scale from 1 to 5, or 1 to 10. Information stored in the database is used as input into deterioration models. Deterioration models predict the condition of bridge elements at any given point in future. These can be deterministic or probabilistic in nature. Deterministic models assume that deterioration will take place at a known rate; however, probabilistic models treat deterioration rate as random variable and estimate the probability that a bridge will be at a certain condition in the future. Most probabilistic models implement Markov process, in which the future condition of a bridge can be estimated when the initial or present conditions are known [4]. Costs and expected benefits, in terms of user cost savings and serviceability improvements, are estimated and fed into optimization models for identifying the least-cost maintenance, repair, or, rehabilitation alternatives using a life-cycle cost analysis or a comparable procedure. A BMS also performs a multi-year network analysis which allows an agency to estimate the impacts of implementing or deferring repairs in the future considering all the bridges under its jurisdiction.

2. PERFORMANCE PREDICTION

Many bridge rating methods have been developed to aid in priority setting. Most such methods develop a composite index or indices for each project. Bridges or projects are then ranked according to the values of these indices, which determines the priority of each project [5]. The FHWA sufficiency rating is computed using the structural-condition rating from the inspection reports of bridge components and other related information [6]. The sufficiency rating procedure is a method of evaluating highway bridge condition data by calculating four separate factors – structural adequacy and safety, serviceability and functional obsolescence, essentiality for public use, and special reductions – to obtain a numeric value which is indicative of bridge sufficiency to remain in the service. The result of this method is a percentage in which 100 percent would represent an entirely sufficient bridge and zero percent would indicate completely insufficient or a deficient bridge [7].

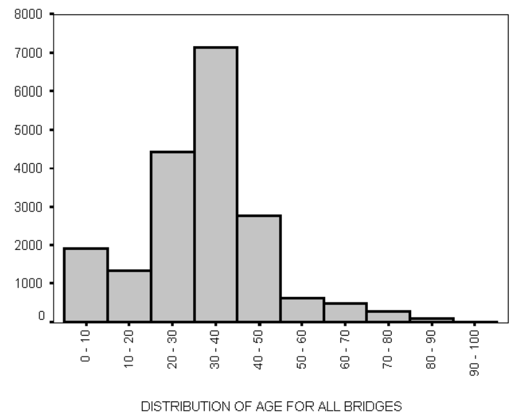
The prediction of the impact of different strategies on the system objectives is an important purpose of bridge data analysis. The decision making, either at the network level or at the project level, is based on current and future bridge conditions. Therefore, it is essential for a bridge management system to be capable of accurately predicting future bridge conditions. This involves predicting the future conditions of bridge elements, agency costs of different projects and activities, and user and non-user consequences in terms of user costs, user time, accident rates and other impacts [5]. In order to predict the future condition of a bridge element, one should know how this structure deteriorates in time. Numerous deterioration models are reported in the literature. Most of these methods were developed using probabilistic approaches ranging from Markovian methods to regression techniques with various levels of detail. However, estimation of transition probabilities, necessary in Markovian approaches, depends on subjective engineering judgement and requires constant updating. Madanat et al. [8] introduced an ordered probit model for estimating transition probabilities for condition ratings. While offering mostly marginal improvements, such methods increase the complexity of the procedures and level of expertise needed. Besides, high reliance of these methods on historical data, which are likely to contain missing information, reduces the chances for a reliable model. The ability of learning in artificial intelligence (AI) methods provides promising results in modeling and forecasting even in the existence of non-linear complex relationships [9],[10].

In this study two AI tools, artificial neural networks and genetic algorithms, are utilized to develop models to predict bridge sufficiency ratings using current geometrical, age, traffic, and structural attributes as explanatory variables without considering a deterioration model.

3. DATA

The data used in this study was acquired from the California Department of Transportation's web-site (http://www.dot.ca.gov/hq/structur/strmaint). This site provides data for the conditions of bridges that are owned both by state and local governments. This study investigated state owned bridges. The data was compiled in 1999; therefore, it is assumed that the condition

of the bridges are of the year 1999. There are 19120 cases in the data although the California State Highway System contains 12126 bridges, third highest in the US. The data contains more than one entry for some bridges since these contain more than one major structural element rated by the officials. Of these 12126 bridges, 1756 (14%) were reported as substandard by Better Roads 1999 which is approximately 10% below the national average. Figure 1 and Figure 2 illustrate the age and



sufficiency rating distribution of the structural elements used in the study.

Figure 1. Distribution of Age of California Bridges

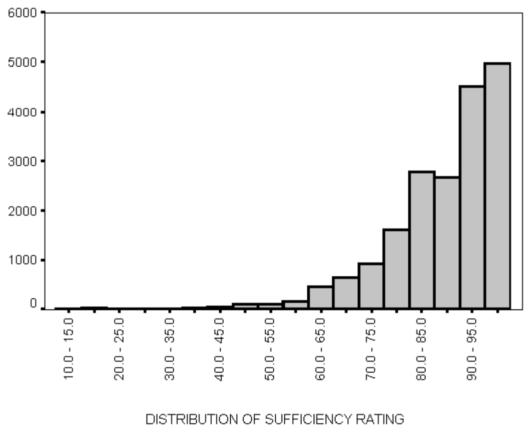


Figure 2. Distribution of Sufficiency Ratings of California Bridges

The mean of the bridge age was 31.6 years and mean sufficiency rating was found as 86.5 reflecting fairly well condition of the overall bridge inventory.

Although California Department of Transportation has already developed a Markovian deterioration model, this study intends to demonstrate the power of AI methods and their applicability for BMS.

4. ARTIFICIAL NEURAL NETWORKS

An ANN is defined as a type of information processing system whose architecture is inspired by the structure of biological [11]. The ANN emulates the network of neurons in human brain and therefore its capability of processing complex, non-linear relationships. The artificial neuron is an approximately simulated model of a biological neuron. A typical neuron receives input either by excitation or inhibition from many other neurons. When its excitation reaches a certain level it reacts or fires. The firing is propagated through a link to other neurons where it in turn acts as input to those neurons. The firing can be thought as its output in which case the neuron becomes a binary device: it either fires or not. For some purposes it is useful to think the neurons output as a continuous quantity such as: its level of activation, its net excitation, or its firing frequency. Regardless of its nature of output it is possible to describe the relationship between input and output mathematically. This function and treatment of the weights, addition or multiplication etc., is one of the major components of a neural network that controls its behavior. Although a neuron computes the same function, within the limits of its accuracy, variations in the strengths of the connections among neurons enable a network to learn. The learning may increase or decrease over time in a way that depends on experience of the connection between two neurons. These artificial neurons are used to develop an artificial neural net with many interconnections among different neurons. The connectivity of neurons in a net is a second component that determines the network behavior. The most commonly used structure has three distinctive layers:

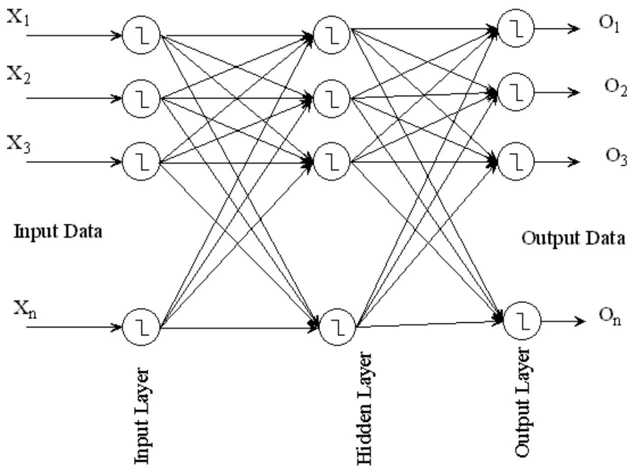


Figure 3. Multi-layer and Back-Propagation ANN

an input, a processing or hidden, and an output layer as illustrated in Figure 3. The third component is the treatment of error to produce targeted output in the network. The most widely applied learning scheme is the supervised learning. Learning is expected to be achieved when the patterns of changes in the weights reach some stability. Each node accommodates a function triggered by the input in the input layer, and by the weights in other layers at each of links that connect the node to the others. The resultant output is compared to targeted solution providing a measure of total error. The weights are then adjusted such that the previously introduced output can be produced at the output layer with highest accuracy possible [12]. Therefore the error is back propagated to the network. Processing some number of different input patterns, it is anticipated that the network is able to generalize what it has practiced, in other words the network has been trained. However, there is no guarantee or a measure that indicates the level of learning attained by the network, sufficiency of number of different input-output patterns or cases in the training set is subjective. Such a network has been found to be capable of carrying out parallel computations for different tasks such as pattern recognition, linear optimization, speech recognition, and prediction [13].

Back-propagation networks are known for their ability to solve a wide variety of prediction and classification problems; and it is used in different areas from predicting the outcome of construction litigation [14] to predicting changes in construction cost indexes [15]. Back-propagation networks are preferred to predict the bridge rating items in this study for its simplicity and proven success.

5. GENETIC ALGORITHMS

The Genetic Algorithm is a search and optimization technique based on the mechanism of natural evolution [16], [17]. Evolutionary computation is the name given to a collection of algorithms based on the evolution of a population toward a solution of a certain problem. These algorithms can be used successfully in many applications requiring the optimization of a certain multi-dimensional function. The population of possible solutions evolves from one generation to next, ultimately arriving at a satisfactory solution to the problem. These algorithms differ in the way a new population is generated from the present one, and in the way the members are represented within the algorithm [18].

The first step to employ a GA is to encode any possible solution to the optimization problem as a set of strings (chromosome). Each chromosome represents one solution to the problem, and a set of chromosomes is referred to as a population. The next step is to derive an initial population. A random set of chromosomes is often used as the initial population. Some specified chromosomes can also be included. The initial population is the first generation from which the evolution starts. The third step is to evaluate the quality of each chromosome. Each chromosome is associated with a fitness value. The objective of the GA search is to find a chromosome that has the optimal fitness value. The selection process is the next step. In this step, each chromosome is eliminated or duplicated (one or more times) based on its relative quality. The population size is typically kept constant. Selection is followed by the crossover step. With some probability, some pairs of chromosomes are selected from the current population and some of their corresponding components are exchanged to form two valid chromosomes, which may or may not already be in the current population. After crossover, each string in the population may be mutated with some probability. The mutation process transforms a chromosome into another valid one that may or may not already be in the current population. The new populations are then evaluated. If the stopping criteria have not been met, the new population goes through another cycle (iteration) of selection, crossover, mutation, and evaluation. These cycles continue until one of the stopping criteria is met [19].

There are several applications of the genetic algorithms in civil engineering problems. For example Al-Tabtabai and Alex [20] proposed a genetic algorithm model to solve optimization problems in the construction, Hegazy [21] used in resource allocation and leveling, Feng et al. [22] developed a system to solve construction time-cost trade-off problems, Navon and McRea [23] used to select optimal construction robot and Yang and Soh [24] applied GAs in structural optimization.

6. TRAINING AND TESTING NETWORKS

The NeuroShell® Predictor is a software program designed to simplify the creation of a neural network and genetic algorithm applications to solve the forecasting and pattern recognition problems. It builds back-propagation neural networks with one hidden layer and tests networks with different numbers of nodes in the layers, allowing user to identify the best performing configuration. For

Table 1. Variable Descriptions

Feature Name (1)	Description (2)
(1) Kind of Highway	1 = Interstate 2 = U.S. Numbered Highway 3 = State Highway 4 = County Highway 5 = City Street 6 = Federal Lands Road 7 = State Lands Road 8 = Other
(2) Designated LOS	D = Designated Level of Service (NBI Item 5C) 0 = None of the below 2 = Mainline 3 = Alternate 4 = Bypass 6 = Spur 7 = Ramp, Wye, Connector, etc. 8 = Service and/or unclassified frontage road
(3) District	
(4) Bypass Length	Bypass detour length in kilometers
(5) Lanes On Str.	Number of lanes on the structure
(6) Lanes UnderStr.	Number of lanes under the structure
(7) AADT	Annual Average Daily Traffic
(8) Approach Width	Width of the approach roadway, shoulders included, in meters.
(9) Bridge Rail Rate	0 = Does not meet currently accepted standards 1 = Meets currently acceptable standards N = Not applicable or not required
(10) Transition Rate	0 = Does not meet currently accepted standards. 1 = Meets currently acceptable standards N = Not applicable or not required
(11) Approach Guard Rail	0 = Does not meet currently accepted standards. 1 = Meets currently acceptable standards N = Not applicable or not required
(12) Approach Guard Rail End	0 = Does not meet currently accepted standards. 1 = Meets currently acceptable standards N = Not applicable or not required
(13) Material Type	1 = Concrete 2 = Concrete continuous 3 = Steel 4 = Steel continuous 5 = Prestressed concrete 6 = Prestressed concrete continuous 7 = Wood or timber 8 = Masonry 9 = Aluminum, wrought iron, or cast iron 0 = Other
(14) Design Type	01 = Slab 02 = Stringer/Multi-beam or Girder 03 = Girder and Floor beam System 04 = Tee Beam 05 = Box Beam or Girders - Multiple 06 = Box Beam or Girders - Single or Spread 07 = Frame (except frame culverts) 08 = Orthotropic 09 = Truss - Deck 10 = Truss - Thru 11 = Arch - Deck 12 = Arch - Thru 13 = Suspension 14 = Stayed Girder 15 = Movable - Lift 16 = Movable - Bascule 17 = Movable - Swing 18 = Tunnel 19 = Culvert 21 = Segmental Box Girder 22 = Channel Beam 00 = Other
(15) Roadway Width	Bridge roadway width curb-to-curb (meters)

Table 1. (Cont.d) Variable Descriptions

Feature Name (1)	Description (2)
(16) Vertical Clearance	Minimum vertical clearance over bridge roadway (meters). 99.99 means no impaired vertical clearance. 0.00 means not applicable.
(17a) Year Built	Year of Construction
(17b) Age	
(18) Inventory Rate	
(19) Operational Rate	
(20) Deck Rating	
(21) Sup-Structural Rating	
(22) Sub-Structural Rating	
(23) Culvert Rating	
(24) Overall Structural Rating	
(25) Deck Geometry Rating	
(26) Under Clearance Rating	
(27) Waterway Rating	
(28) Approach Roadway Alignment Sufficiency Rate	

genetic training it uses a modified General Regression Neural Net [25]. This program was used for training a network using the data downloaded. The data was organized in a MS Excel spreadsheet that was later input into NeuroShell® Predictor. Different variations of the data, the cases, and the parameters of the program were experimented with in order to attain the network that performed best.

The data is randomly divided into two groups, 75% of the data used in the training and remaining used in the testing of the networks. The networks comprised of 28 input features and one output feature, Table 1 displays the variable descriptions. 16 different cases, in 4 sets, were created for neural network application by altering time variable, maximum number of hidden neurons, and data ranges based on sufficiency ratings. In this section the performances of network training for these 18 networks are presented. It was hypothesized before any computation that age variable must be the most important variable that affects deterioration. In order to observe the scale effect of age variable on the results all sets included two types of network: one with nominal age, and one with 'year built' as age variable. In the first set the effect of increasing maximum number of hidden neurons from 80 to 150 is studied. Table 2 presents the training performance of networks in this set. The impact of increasing maximum number of hidden neurons was not conclusive, increasing number of hidden neurons improved R^2 and decreased average error for the networks with year built as age, however, changes were in opposite direction with networks with nominal age. In both cases the changes in the magnitude were trivial. For the importance ranking of variables, 'year built' ranked the first with networks when it is used. 'Sub-structure rate' ranked second for these networks while, 'sub-structure rate' ranked first and 'number of lanes on structure' ranked the second with networks with nominal age. The results of the first set imply that overall, sub-structure rating is more important. However, time was important for Network 1 and 3, but it is suspected that this result was arbitrary. Since data does not provide any information on maintenance history, age or year-built variables can not be considered as better identifiers than actual ratings (which are more likely to reflect current levels of deterioration).

In the second set the number of input features decreased from 28 to 23 by eliminating highly insignificant variables after the first

Table 2. Effect of Number of Hidden Neurons to the Training Performance

Name	Time Variable	Training Set	No of Inputs	No of Hidden Neurons	Training Time (min.)	R^2	Average Error	Most Important Variable	2nd Most Important Variable	3rd Most Important Variable
Network 1	Year Built	0-100	28	78	2:59	0.559	5.8	Year Built (0.340)	Sub. Str. Rate (0.151)	No of Lanes on Str. (0.062)
Network 2	Age	0-100	28	80	3:39	0.600	5.6	Sub. Str. Rate (0.206)	No of Lanes on Str. (0.095)	Overall Str. Rate (0.089)
Network 3	Year Built	0-100	28	147	11:26	0.607	5.6	Year Built (0.291)	Sub. Str. Rate (0.099)	Sup. Str. Rate (0.081)
Network 4	Age	0-100	28	145	11:57	0.594	5.7	Sub. Str. Rate (0.240)	No of Lanes on Str. (0.094)	Overall Str. Rate (0.085)

Table 3. Effect of Removing Some Variables to the Training Performance with Different Time Variables

Name	Time Variable	Training Set	No of Inputs	No of Hidden Neurons	Training Time (min.)	R^2	Average Error	Most Important Variable	2nd Most Important Variable	3rd Most Important Variable
Network 5	Year Built	0-100	23	137	15:11	0.593	5.8	Year Built (0.295)	Sub. Str. Rate (0.197)	No of Lanes on Str. (0.075)
Network 6	Age	0-100	23	144	14:01	0.582	5.8	Sup. Str. Rate (0.159)	No of Lanes on Str. (0.103)	Deck Geo. Rate (0.103)
Network 7	Cum. Traffic	0-100	23	144	14:48	0.575	5.9	Sup. Str. Rate (0.205)	Sub. Str. Rate (0.134)	Deck Geo. Rate (0.097)
Network 8	Stan. Year Built	0-100	23	149	16:05	0.610	4.8	Sub. Str. Rate (0.098)	No of Lanes on Str. (0.090)	Overall Str. Rate (0.087)

set and the effects of representing age with additional measures are investigated. The maximum number of hidden neurons was kept as 150 since there was no significant adverse effect and in order to avoid poor results due to such limitation. Cumulative traffic that a structure carried over its life is not only a measure of time but also an indicator of a total physical load exerted on the structure. Network 7 used cumulative traffic as age variable. Cumulative traffic values are the product of the Annual Average Daily Traffic (AADT) by the structure's age. In Network 8 'year built' variable was statistically standardized, the difference between the mean and each individual value is divided by the standard deviation. The results of this set are given in Table 3 and they confirm that 'year built' variable's importance is arbitrary. The structural ratings ranked as the most important variables in the networks other than Network 5, as expected.

In the third set, the training data was grouped according to different sufficiency ratings after realizing that the distribution of sufficiency ratings as in Figure 2 is not uniform. The performance of the training can be improved by working with more uniformly distributed output variable. The original data is divided into three groups. The first group included the structures in very good condition, sufficiency ratings between 100-90, the second group had structures in good to fair condition with sufficiency rating between 89-70, and the third group covered structures in fair to poor condition. The results of this set are illustrated in Table 4. These results show that the Network 12 with fair to poor condition ratings outperformed the other networks significantly with a R^2 value of 0.831 and highest rankings in 'overall structure', 'approach guard rail', and 'sub-structure' ratings. The inferior performance for networks with better structural conditions can be explained with the fact that there may not be enough variation in input and output features to capture. However, if this is not the case then it may indicate that the network training is insufficient under current circumstances, and it is subject to a more detailed investigation.

Finally in the fourth set of training, the training performances of networks with different output ranges were tested. The amount of variation is increased by widening the range of output for Network 12, while trying not create biases towards structures in good condition. Two new networks are then created with sufficiency rating ranges of 0-75, and 0-79. Training performances are provided in

Table 4. Training Performance for Structures That Grouped According to Their Sufficiency Ratings

Name	Time Variable	Training Set	No of Inputs	No of Hidden Neurons	Training Time (min.)	R^2	Average Error	Most Important Variable	2nd Most Important Variable	3rd Most Important Variable
Network 9	Year Built	0-69	28	149	1:54	0.826	3.9	Appr. Guard Rail End (0.216)	Overall Str. Rate (0.122)	Deck Rate (0.090)
Network 10	Year Built	70-89	28	147	6:41	0.273	3.5	Culvert Rating (0.641)	Year Built (0.121)	Appr. Guard Rail (0.040)
Network 11	Year Built	90-100	28	149	7:44	0.303	1.9	Sub. Str. Rate (0.123)	Deck Rate (0.107)	Roadway Width. (0.107)
Network 12	Age	0-69	28	150	1:32	0.831	3.8	Overall Str. Rate (0.144)	Appr. Guard Rail (0.120)	Sub. Str. Rate (0.113)
Network 13	Age	70-89	28	148	4:53	0.287	3.5	Culvert Rating (0.309)	Sup. Str. Rate (0.088)	Appr. Guard Rail (0.082)
Network 14	Age	90-100	28	150	5:41	0.324	1.9	Roadway Width. (0.149	Approach Width (0.096)	Sub. Str. Rate (0.070)

Table 5. Training Performance Of Data Sets With Different Sufficiency Rating Ranges

Name	Time Variable	Training Set	No of Inputs	No of Hidden Neurons	Training Time (min.)	R^2	Average Error	Most Important Variable	2nd Most Important Variable	3rd Most Important Variable
Network 15	Age	0-79	28	147	2:46	0.710	4.7	Sub. Str. Rate (0.165)	Sup. Str. Rate (0.134)	Overall Str. Rate (0.099)
Network 16	Age	75	28	146	2:02	0.779	4.2	Overall Str. Rate (0.257)	Sup. Str. Rate (0.110)	Sub. Str. Rate (0.097)

Table 6. Training Performance For Genetic Algorithms

Name	Time Variable	Training Set	No of Inputs	Training Time	R^2	Average Error	Most Important Variable	2nd Most Important Variable	3rd Most Important Variable
Network 17	Age	0-100	28	43:09:04"	0.69	3.48	District (0.062)	Vertical Clear (0.062)	App. Rd. All. (0.062)

Table 5. The results indicate that although variation in the output is increased, the performances were affected adversely. This outcome rejects the conclusion reached previously. Therefore, the need for more detailed study and probably for a more advanced ANN method or another method is inevitable.

Within this context Network 4 is trained with a genetic algorithm. The training performance is presented in Table 6 and Network 17 shows higher R^2 of 0.69 vs. 0.6, and 3.48 of an average error vs. 5.6. The variables in the highest importance rankings reflect geographical, geometrical characteristics with equal weights and close to those of structural ratings. While providing overall better results Network 17's training time was 43 hours on a 333MHz, Pentium Workstation.

The trained networks are applied to the second group of data reserved for testing. The results of the tests are given in Table 6. It shows that overall performance of testing is slightly lower than of training for all network types except Network 17, this can be result of a better model or can be incidental and requires more work. The ranges of data (column 3) must be taken into consideration when evaluating the percentage of cases with error for a specified range (columns 4-7).

7. CONCLUSIONS

The preliminary findings of an effort to model bridge performance with using AI techniques instead of conventional deterministic and/or probabilistic techniques are presented. The experimentation with different configurations of input and output patterns yielded some understanding of the performance of the neural network models. However, the levels of improvements are still at marginal levels. There exist numerous parameters and options to be experimented with. Among there are: adding variables on maintenance history, concentrating on a specific type of structure, input transformations, more elaborate network architectures and learning schemes. The problem of training time with genetic algorithms is of concern, therefore, it may be preferable to work with smaller data for GA applications and to limit the number of experiments with GAs such as comparing GA vs. the best ANN model.

Table 7. Results for Testing Performance

	R^2	Input-Output Range	Percentage of Error Between 0-3	Percentage of Error Between 3-6	Percentage of Error Between 6-9	Percentage of Error Greater than 10
Network 1	0.564	0-100	34.8	28.9	17.1	19.2
Network 2	0.595	0-100	34.6	28.9	17.9	18.5
Network 3	0.558	0-100	34.2	29.1	17.9	18.8
Network 4	0.578	0-100	33.9	29.4	18.1	18.6
Network 5	0.567	0-100	32.1	29.0	19.1	19.8
Network 6	0.560	0-100	32.2	29.1	18.9	19.8
Network 7	0.563	0-100	31.9	28.9	18.2	20.1
Network 8	0.615	0-100	34.5	29.1	17.8	18.6
Network 9	0.751	0-69	42.1	29.1	16.0	12.8
Network 10	0.170	70-89	47.6	32.1	15.5	4.8
Network 11	0.215	90-100	78.1	20.8	0.8	0.3
Network 12	0.707	0-69	43.0	26.6	13.3	17.2
Network 13	0.175	70-89	48.1	31.7	15.6	4.6
Network 14	0.235	90-100	79.7	19.3	0.6	0.4
Network 15	0.626	0-79	38.6	29.4	16.3	15.7
Network 16	0.683	0-75	42.2	29.5	14.4	13.9
Network 17 GA	0.729	0-100	65.5	14.2	6.72	13.6

REFERENCES

[1] Better Roads, *1999 Bridge Inventory*, http://www.betterroads.com/better_roads/bridgeinv99.htm, 1999.

[2] Small, E. P. and Cooper, J., "Condition of the Nation's Highway Bridges a Look at the Past, Present, and Future", *TR News*, No. 194, pp. 3-9, 1998.

[3] Federal Highway Administration, "*The Status of Nation's Highways and Bridges: Conditions and Performance, and Highway Bridge Replacement and Rehabilitation Program 1989*" Report of the Secretary of Transportation to the United States Congress, 101st Congress, 1st Session, U.S. Government Printing Office, Washington, D.C. , 1989.

[4] Jiang, Y., Saito, M. and Sinha, K. C., "Bridge Performance Prediction Model Using the Markov Chain," *Transportation Research Record*, No. 1180, pp. 25-32, 1988.

[5] Kleywegt, A. J. and Sinha, K. C. "Tools for Bridge Management Data Analysis", *7th TRB Conference on Bridge Management, Austin, Texas*, pp.1-17, 1993.

[6] Gralund, M. S. and Puckett, J. A., "System for Bridge Management in a Rural

Environment", *Journal of Computing in Civil Engineering*, Vol. 10, No. 2, pp. 97-105, 1996.

[7] Federal Highway Administration, *Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges*, U.S. Department of Transportation, Federal Highway Administration, 1995.

[8] Madanat, S., Mishalani, R. and Ibrahim, W. H. W., "Estimation of Infrastructure Transition Probabilities from Condition Rating Data", *Journal of Infrastructure Systems*, Vol. 1, No. 2, pp. 120-125, 1994.

[9] Cattan, J. and Mohammadi, J., "Analysis of Bridge Condition Rating Data Using Neural Networks", *Microcomputers in Civil Engineering*, Vol. 12, pp. 419-429, 1997.

[10] Liu, C., A. Hammad, and Y. Itoh, "Maintenance Strategy Optimization of Bridge Decks Using Genetic Algorithm", *Journal of Transportation Engineering*, ASCE, 123 (2), 91-100, 1997.

[11] Caudill, M. and Butler, C., *Naturally Intelligent Systems*, MIT Press, Cambridge, Mass. , 1990.

[12] Flood I. and N. Kartam, "Neural Networks in Civil Engineering I: Principles and Understanding", *Journal of Computing in Civil Engineering*, ASCE, 8 (2), 131-148, 1994.

[13] Mukherjee, A. and Deshpande, J. M., "Modeling Initial Design Process Using Artificial Neural Networks", *Journal of Computing in Civil Engineering*, 9(3), pp. 194-200, 1995.

[14] Arditi, D., Oksay, F. E. and Tokdemir, O. B. "Predicting the Outcome of Construction Litigation Using Neural Networks", *Computer-Aided Civil and Infrastructure Engineering*, No.13, pp. 75-81, 1998.

[15] Williams, T.P., "Predicting Changes in Construction Cost Indexes Using Neural Networks", *Journal of Construction Engineering and Management*, ASCE, 120 (2), 306-319, 1994.

[16] Goldberg, D. H., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[17] Holland, J. H., *Adaptation in Natural and Artificial Systems*, MIT Press, 1975

[18] Martin, N. M. and Jain, L. C., *Introduction to Neural Networks, Fuzzy Systems, Genetic Algorithms, and Their Fusion, Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications*, CRC Press, New York, 1999.

[19] Wang L., Siegel H. J., Roychowdhury, V. P. and Maciejewski, A. A., "Task Matching and Scheduling in Heterogeneous Computing Environments Using a Genetic-Algorithm-Based Approach", *Journal of Parallel and Distributed Computing*, 47, pp. 8-22, 1997.

[20] Al-Tabtabai, H., and A.P. Alex. "Using Genetic Algorithms to Solve Optimization Problems in Construction", *Engineering, Construction and Architectural Management*, 6 (2), 121-132, 1999.

[21] Hegazy, T., "Optimization of Resource Allocation and Leveling Using Genetic Algorithms", *Journal of Construction Engineering and Management*, ASCE, 125 (3), 167-175, 1999.

[22] Feng, C., L. Liu, S.A. Burns, "Using Genetic Algorithms to Solve Construction Time-Cost Trade-Off Problems", *Journal of Computing in Civil Engineering*, ASCE, 11 (3), 184-189, 1997.

[23] Navon, R., and A. M. McCrea, "Selection of Optimal Construction Robot Using Genetic Algorithm", *Journal of Computing in Civil Engineering*, ASCE, 11 (3), 175-183, 1997.

[24] Yang, J., and C.K. Soh, "Structural Optimization by Genetic Algorithms with Tournament Selection", *Journal of Computing in Civil Engineering*, ASCE, 11 (3), 195-200, 1997.

[25] Chen, C.H., *Fuzzy Logic and Neural Network Handbook*. New York, NY, McGraw-Hill, Inc., l996.