



NEURAL NETWORKS SURROGATE MODELS FOR SIMULATING PAYMENT RISK IN PAVEMENT CONSTRUCTION

Anshu Manik¹, Kasthurirangan Gopalakrishnan²,
Abhishek Singh³, Shengquan Yan⁴

¹Dept of Civil Engineering, Michigan State University, East Lansing, MI 48824, USA

²Dept of Civil Engineering, Iowa State University, Ames, IA 50011, USA

^{3,4}Dept of Civil Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

E-mail: ¹manik@msu.edu; ²rangan@iastate.edu; ³asingh8@uiuc.edu; ⁴smyan@uiuc.edu

Received 9 Apr 2008; accepted 1 Sept 2008

Abstract. A common provision in quality control/quality assurance (QC/QA) highway pavement construction contracts is the adjustment of the pay that a contractor receives on the basis of the quality of the construction. It is important to both the contractor and the contracting agency to examine the amount of pay that the contractor can expect to receive for a given level of construction quality. Previous studies have shown that computer simulations can provide a better, more detailed examination of the pay schedule than is possible by simply determining the expected pay. In particular, the simulation process can provide an indication of the variability of pay at various quality levels and can identify the factors most responsible for pay adjustments. Stochastic simulation models are very useful in estimating and analyzing payment risk in highway pavement construction. However, such models are constrained by their computational requirements, and it is often necessary to couple them with simpler models to speed up the process of decision-making. This paper investigates the use of Neural Networks (NN) to build surrogate models for a pavement construction payment-risk prediction model. The results show that although the average error associated with the NN predictions are acceptable; in some particular cases the errors may be unacceptably high.

Keywords: QC/QA, artificial neural networks, pavement construction, payment risk, stochastic simulation.

1. Introduction

Quality control (QC) is a procedure or set of procedures intended to ensure that a manufactured product adheres to a defined set of quality criteria or meets the requirements of the client or customer, whereas quality assurance (QA) is intended to ensure that a product under development (before work is complete, as opposed to afterwards) meets specified requirements. QA specifications are an important component of an organization's commitment to overall quality management, and consist of several activities, including process control, acceptance, and sometimes – independent assurance of a product (Buttler and Harrel 2000).

To promote the construction of better-quality asphalt pavements, over the years, a number of state highway agencies have moved from using traditional method specifications to using statistically based QC/QA specifications (Patel *et al.* 1997). In many QC/QA specifications, the contractor is primarily responsible for quality control of the process, whereas the highway agency is responsible for testing the acceptance of the product. These specifications are typically statistics based, in which methods such as stratified random sampling and lot-by-lot testing are used, allowing contractors to ensure that their operations are producing an acceptable product

(TRR 1996). There is considerable literature discussing the various steps involved in developing a new QA specification for asphalt pavements (Burati and Patrick 2000; NCHRP 38; NCHRP 65; NCHRP 212; Weed 1995; Burati 2006).

Specifications for the construction of asphalt pavements can generally be classified into method-related specifications (MRS), end-result specifications (ERS), performance-related specifications (PRS), or combinations thereof. Method specifications give a set of procedures, that if followed by the contractor, will result in full payment for the constructed facility. This places a great deal of responsibility and testing burden on the agency rather than the contractor. End-result and performance-related specifications, as their names imply, require a contractor to achieve specified as-produced or as-constructed quality levels, which are ideally linked to the attainment of good future performance. These types of specifications shift most or all of the responsibility for producing a high quality product to the contractor, and should ideally offer the contractor complete freedom in the methods used to arrive at these quality levels (Buttler and Harrell 2000).

Quality is estimated by measuring certain quality characteristics like density, air voids, asphalt content etc. But the measurement process has some uncertainties

involved because of measurement errors, operator is error, instrument bias etc (Buttler and Harrell 2000). Fig. 1 shows some real data from Illinois highway construction project. Voids in the gyratory (a laboratory compaction equipment) asphalt concrete specimen were measured by the contractor and the district. For the same material two parties got different values. From the plot it can be noted that:

- (1) There is a clear shift in the two measurements. This is called as bias, i.e. consistently measuring lesser or more than the real value or other party.
- (2) In addition, it can be seen that the difference in measurement varies from one pair of readings to the other. This is called as measurement variability.

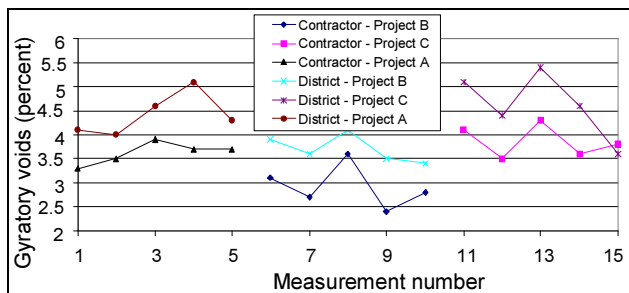


Fig. 1. Contrast in air void measurements in 3 projects by 2 parties

This means that if the payment to the contractor is based on these measurements, there is a probability that he either gets more pay than he deserved or is paid less than that. Payment risk is defined as the difference between the payment made to the contractor based on the above-mentioned quality characteristic measurements. The ideal pay is based on the actual quality of the pavement constructed. It is desirable to determine the magnitude of risk involved with such construction processes and relate them to the factors that may be affecting them. This knowledge can be used to reduce this risk.

The parameters that have been identified to significantly affect risk are: 1) production variability, 2) measurement variability, 3) targeted mean quality, 4) sample size, 5) bias in measurements of contractor, agency and third party, 6) upper and lower specification limits for the quality characteristic and 7) tolerance allowed in comparing the measurements of the contractor, agency and the third party. It is desirable to have a simulation that would take the above-mentioned parameters as input and would predict risk in the form of a risk plot showing magnitude of risk versus the mean quality characteristic and confidence intervals for the risk.

2. Objective

Although Monte Carlo-based simulations are generally used for these purposes, such simulations are often computationally intensive and therefore not very useful for batch processing because of prohibitive time requirements. This paper explores the feasibility of using computational intelligence-based systems such as neural networks for developing surrogate simulation models for

risk analysis in highway construction processes. In this problem, although the simulations work with generation of normally distributed random numbers, the calculations and conditional processing of data make the output non-linear and harder to map and therefore the use of computational intelligence-based techniques was investigated.

3. Neural networks-based surrogate model

Background

Neural networks (NN) have been found to be powerful and versatile computational tools for organizing and correlating information in ways that have proved useful for solving certain types of problems too complex, too poorly understood, or too resource-intensive to tackle using more traditional computational methods. NN have been successfully used for tasks involving pattern recognition, function approximation, optimization, forecasting, data retrieval, and automatic control, to name just a few (Haykin 1999).

There are several different types of artificial neural networks such as back-propagation neural networks (BPNN), radial basis function networks (RBFNN), probabilistic neural networks (PNN), and generalized regression neural networks (GRNN). Computing abilities of neural networks have been proven in the fields of prediction and estimation, pattern recognition, and optimization (Haykin 1999). The best-known example of a neural network training algorithm is back-propagation (Rumelhart *et al.* 1986; Haykin 1994; Fausett 1994; Patterson 1996), which is based on a gradient-descent optimization technique. The back-propagation neural networks have been described in many sources (Hegazy *et al.* 1994; Adeli and Hung 1995; Mehrotra *et al.* 1997; Haykin 1999).

Feed forward neural networks

First, the possibility of using a *fully connected, feed forward neural network model* as a surrogate model for simulating payment risk in highway construction was investigated. NNs are parallel connectionist structures constructed to simulate the working network of neurons in human brain. They attempt to achieve superior performance via dense interconnection of non-linear computational elements operating in parallel and arranged in a pattern reminiscent of a biological neural network. The perceptrons or processing elements and interconnections are two primary elements which make up a neural network. A single perceptron is mathematically represented as (Haykin 1999):

$$y_k = \varphi(v_j) = \varphi\left(\sum_{i=1}^n x_i w_{ij} - b_j\right), \quad (1)$$

where x_i is input signal, w_{ij} – synaptic weight, b_j – bias value, v_j – activation potential, $\varphi()$ – activation function, y_k – output signal, n – the number of neurons for previous layer, and k – the index of processing neuron.

Multilayer perceptrons (MLPs), frequently referred to as multi-layer feedforward neural networks, consist of an input layer, one or more hidden layers, and an output

layer. Training in a MLP is an unconstrained optimization problem, which is subject to the minimization of a global error function depending on the synaptic weights of the network (Goktepe *et al.* 2004). For a given training data consisting of input-output vectors, values of synaptic weights in a MLP are iteratively updated by a training algorithm to approximate the target behaviour. This update process is usually performed by backpropagating the error signal layer by a layer and adapting synaptic weights with respect to the magnitude of error signal (Goktepe *et al.* 2004). Rumelhart *et al.* (1986) presented the first backpropagation training algorithm for use with MLP structures.

Backpropagation algorithm

The backpropagation training algorithm for a simple three-layer MLP structure (one input layer, one hidden layer, and one output layer) is described as follows. The network is initially presented with an input vector ($x_1, x_2, x_3, \dots, x_N$), augmented by a bias $x_0 = 1$. The net activations of the hidden neurons and the outputs from the hidden layer are calculated as follows:

$$I_j = \varphi(\text{neth}_j) = \varphi\left(\sum_{i=0}^N v_{ji}x_i\right), \quad (2)$$

where i varies from 0 to N and j – from 1 to L hidden neurons. The synaptic weights of the interconnections between the inputs and the hidden neurons are represented by v_{ji} . Among the non-linear activation functions, the *sigmoid* (logistic) function is the most usually employed in ANN application. The presence of a non-linear activation function, $\varphi()$, is important because, otherwise, the input-output relation of the network could be reduced to that of a single-layer perceptron. The computation of the local gradient for each neuron of the multilayer perceptron requires that the function $\varphi()$ be continuous. In other words, differentiability is the only requirement that an activation function would have to satisfy. The *sigmoid* function is a bound, monotonic, non-decreasing function that provides graded, non-linear response within a specified range, 0 to 1. The *sigmoid* non-linear activation function is given by:

$$\varphi(\text{neth}_j) = \frac{1}{1 + \exp(-\beta \text{neth}_j)}, \quad (3)$$

where β is a parameter defining the function slope. The net activations for the neurons in the output layer and the outputs are calculated as follows:

$$y_k = \varphi(\text{neto}_k) = \varphi\left(\sum_{j=0}^L w_{kj}I_j\right), \quad (4)$$

where k varies from 1 to M output neurons. The synaptic weights of the interconnections between the hidden neurons and the output neurons are represented by w_{kj} . The system error is then computed by comparing the actual outputs (y_k) with the desired outputs (d_k). The error terms for the output neurons (δ_k^o) and the hidden neurons (δ_j^h) are given by:

$$\delta_k^o = (d_k - y_k)\varphi'(\text{neto}_k), \quad (5)$$

$$\delta_j^h = \varphi'(\text{neth}_j) \sum_{k=1}^M \delta_k^o w_{kj}, \quad (6)$$

where the sigmoid activation function is differentiated as follows:

$$\varphi'(\text{neto}_k) = \varphi(\text{neto}_k)(1 - \varphi(\text{neto}_k)) = y_k(1 - y_k), \quad (7)$$

$$\varphi'(\text{neth}_j) = \varphi(\text{neth}_j)(1 - \varphi(\text{neth}_j)) = I_j(1 - I_j). \quad (8)$$

Then, the synaptic weights are updated for each neuron in the hidden layer and the output layer. The backpropagation algorithm essentially changes synaptic weights along the negative gradient of error energy function; thus, weight changes are proportional to the magnitude of error energy. The formulations for weight updates in the output layer and the hidden layer are given as:

$$w_{kj}(t+1) = w_{kj}(t) + \eta \delta_k^o I_j + \alpha[w_{kj}(t) - w_{kj}(t-1)], \quad (9)$$

$$v_{ji}(t+1) = v_{ji}(t) + \eta \delta_j^h x_i + \alpha[v_{ji}(t) - v_{ji}(t-1)], \quad (10)$$

where η is the learning rate parameter that can be selected from the range $[0,1]$ and α indicates momentum term varying within $[0,1]$.

In this algorithm, the error energy used for monitoring the progress toward convergence is the generalized value of all errors that is calculated by the least-squares formulation and represented by a Mean Squared Error (MSE) as follows (Haykin 1999):

$$MSE = \frac{1}{MP} \sum_{k=1}^M \sum_{p=1}^P (d_k - y_k)^2, \quad (11)$$

where M is the number of neurons in the output layer and P represents the total number of training patterns.

4. Achieving optimum network architecture

For this problem, the input layer will have 11 nodes, corresponding to the risk parameters identified previously: (1) asphalt concrete production variability, (2) measurement variability, (3) targeted mean quality, (4) sample size, (5) bias in measurements of contractor, (6) bias in measurements of agency, (7) bias in measurements of the third party, (8) upper specification limits for the quality characteristic, (9) lower specification limits for the quality characteristic, (10) tolerance allowed in comparing the measurements of the contractor, agency and third party (for $N = 1$ comparison), and (11) tolerance allowed in comparing the measurements of the contractor, agency and third party (for $N = 3$ comparison). The last two input parameters are based on standard Illinois Department of Transportation (IDOT) procedure. The output parameters are (1) the mean risk magnitude, (2) higher confidence limits, and (3) lower confidence limits corresponding to one value of chosen quality characteristic. The output is desirable in the form of a risk plot showing magnitude of risk versus the mean quality characteristic and confidence intervals for the risk. Fig. 2 shows a typical risk plot with mean risk and 90% confidence intervals for payment based on voids measurements.

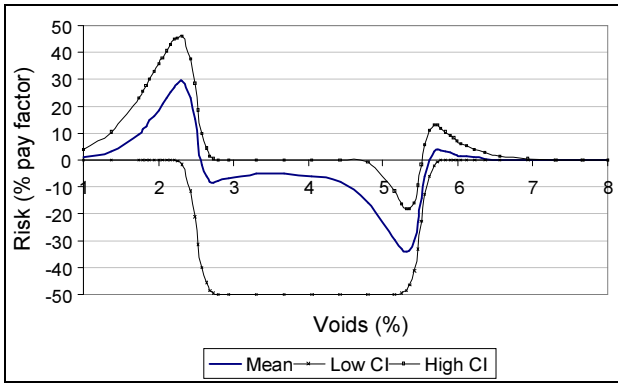


Fig. 2. Typical risk plot

A Monte Carlo-based simulation was developed to generate 60 000 data points for training the NN model and 25 000 data points were used for testing. The input values were chosen randomly from within their feasible ranges. The data sets were normalized to fall between 0.0 and 1.0. In this case, although the simulations works with generation of normally distributed random numbers the calculations and conditional processing of data makes the output to be often non-normally distributed. This is evident in Fig. 1 as well. The distance of the lower confidence limit from the mean is much greater than that between the upper confidence limit and the mean, especially in the middle part of the plot. This makes the simulation non-linear and harder to map.

One of the most important issues in the development of an NN model is the architecture. Determination of the input and output variables, number of hidden layers, and number of hidden neurons in each hidden layers is crucial in the development part of the NN models. The architecture of the NN model has significant effect on the success of the developed model. Usually, a neural network with too few hidden neurons is unable to learn sufficiently from the training data set, whereas a neural network with too many hidden neurons will allow the network to memorize the training set instead of generalizing the acquired knowledge for unseen patterns (Rumelhart *et al.* 1986). Haykin (1999) recommends using two hidden layers; the first one for extracting local features and the second one for extracting global features.

Initially, a network with only one hidden layer was tried in this study. The MATLAB[®] software was used for developing the NN models. But the errors were higher than acceptable. Fig. 3 shows the MSE for network with one hidden layer with respect to number of neurons in the hidden layer for two different learning rates (denoted as LR in Fig. 3). The MSE were determined with 50 000 random cases in the training set and 25 000 cases in the test set. Number of cycles run in training was fixed at 150. Some preliminary runs indicate that after 150 cycles generally the change in MSE with cycles is very small, even if not minimal or zero. In this Fig., MSEs are shown for scaled data. However, the best possible accuracy with one hidden layer also may be not considered acceptable because, when unscaled, they will translate to 16% to 22% error in payment.

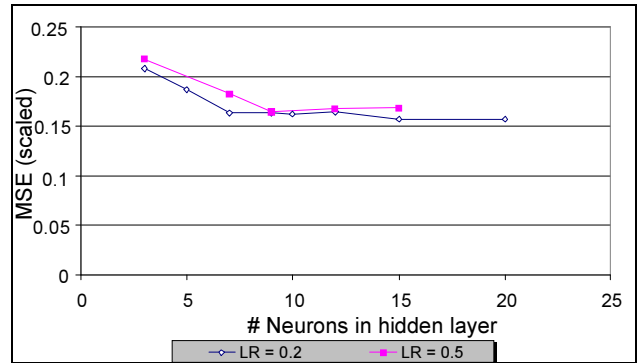


Fig. 3. Mean squared error (MSE) for one hidden layer network

Two or more hidden layers were expected to capture the non-linearities of the phenomena better, so several two hidden layer architectures were tested. The MSEs for all the cases, as shown in Fig. 4, remained between 0.115 and 0.16. This shows that a network architecture with two hidden layers is more suited for this problem compared to one hidden layer architecture. In most cases, the MSE increased as the learning rate increased from 0.2 to 0.3. When the learning rate was increased above 0.3, the magnitude of increase in MSE values was lower. With a learning rate of 0.5, the MSE decreased slightly for the same cases, thus indicating unstable performance for learning rates above 0.3. Similar analyses were also carried out with 3 hidden layers, but the MSE values were found to be much higher than those from 2 hidden layers. Therefore, the final optimal architecture had two hidden layers. The momentum parameter was set to zero during training.

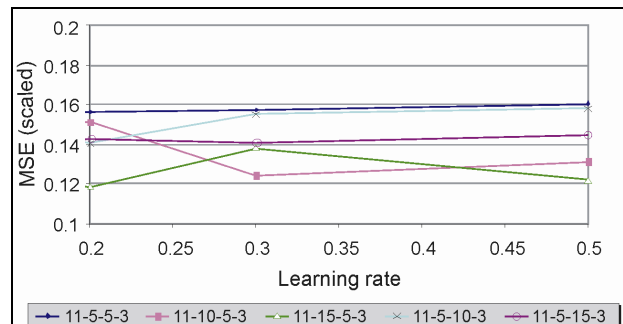


Fig. 4. Mean squared error (MSE) for two hidden layer network

Also, parametric analyses were conducted to determine the optimum number of hidden neurons. The results showed that network with 15 neurons in each of the hidden layers had the best performance. The chosen neural network (11-15-15-3) was run for 500 epochs (cycles) and the errors against the number of epochs are plotted in Fig. 5. Considering constraints on the processing time, it was decided that the network could be trained for 250 epochs without compromising on solution quality.

An important factor in training neural network is to have sufficient number of training cases while avoiding over-fitting. In all the cases, 50 000 training cases were

used. To determine the optimal number of training cases required, the chosen network was trained with 60 000 cases and then with 85 000 cases. The MSEs in both the cases were almost identical. However, the decrease in error when using 60 000 cases was significant compared to that with 50 000 cases. This shows that 60 000 cases may be a good number for the training set. Moreover, it was also found that training errors were comparable to the errors in independent test set cases. This indicated that the network was trained to adequate precision without over-fitting.

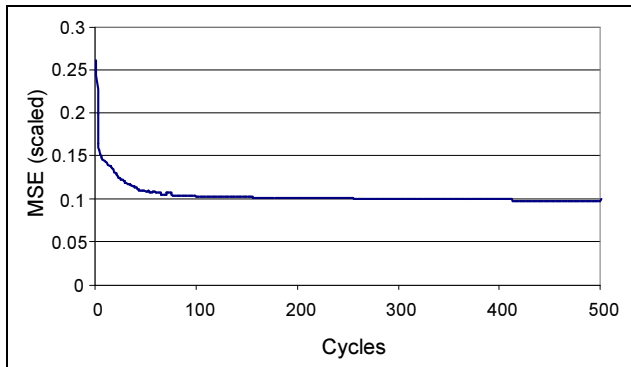


Fig. 5. Training curve for 11-15-15-3 network

5. Results and discussion

Fig. 6 contrasts the predicted values of payment risk with the expected values. In this plot, the dark solid line represents the expected value of mean risk for a particular set of input parameter values. The lighter band around it is formed by plotting the corresponding predicted values. The thickness of the band shows the spread in prediction. Smaller spread would mean that the predicted mean risks are closer to the expected mean risks and hence errors are small and vice-versa. In the sample result, the spread in mean risk is almost $\pm 7.5\%$ for the first part of the plot.

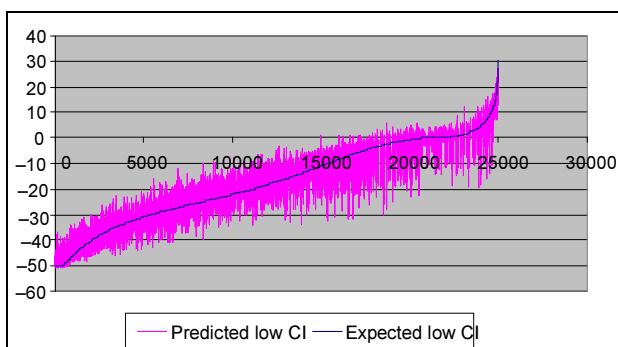


Fig. 6. Expected versus NN predicted mean risk magnitude

Figs 7 and 8 show the results for two standard risk plots for air voids in constructed asphalt pavements. The entire plot is obtained by running the neural network simulation for each data point. Fig. 7 shows that for the cases analyzed the predicted values are very close to the expected values. If the network does predict risk with this

much accuracy, then the chosen network would be considered as a good surrogate simulation model.

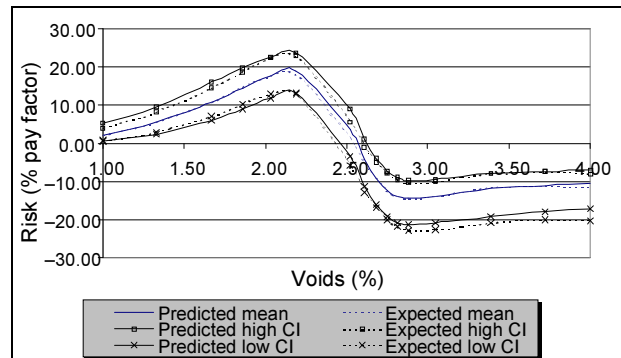


Fig. 7. NN predicted payment risk plot (typical example)

Fig. 8 shows another representative case, where the predicted values are significantly away from the expected ones, especially for higher air voids. In this case, however, the predicted values are appreciably away from the expected ones, especially for higher voids values. It should be noted that the most important part of risk plots are the peak risks. This represents one of the extreme cases.

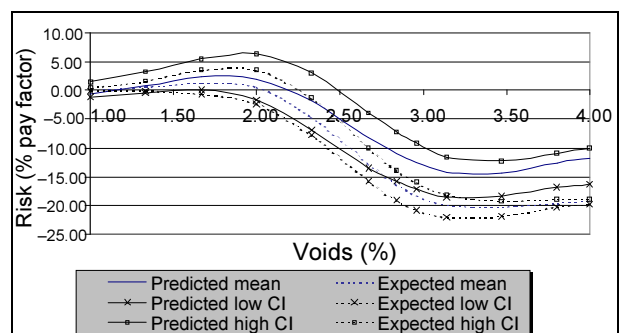


Fig. 8. NN predicted payment risk plot (extreme example)

The results show that although the average error associated with the NN predictions are acceptable, in some particular cases the errors may be unacceptably high (especially around very high or very low input values). This provides motivation for trying some improvements in the surrogate model so that such errors can be reduced. Future research efforts will focus on investigating other types of computational intelligence paradigms such as hierarchical approach, decision trees, and support vector machines (SVM), as surrogate models, for payment risk analysis in highway pavement construction. It is noted that the data used in this paper is purely synthetic and future research will also focus on the applicability of the developed neural networks models to actual field data.

References

Adeli, H.; Hung, S. L. 1995. *Machine learning: neural networks, genetic algorithms, and fuzzy systems*. Wiley, New York.
 Burati, J. L.; Patrick, T. M. 2000. *Development of a quality assurance program for asphalt paving mixtures in South*

- Carolina. Report FHWASC-00-05, South Carolina Dept of Transportation, Columbia, 2000.
- Burati, J. L. 2006. Evaluating specification limits, *Transportation Res. Rec.* 1946: 92–98.
- Buttlar, W. G.; Harrell, M. 2000. Development of end-result and performance-related specifications for asphalt pavement construction in Illinois, in *Proceedings of Transportation Conference, Crossroads*, 2000, Iowa State University, Ames, IA, 195–202.
- Fausett, L. 1994. *Fundamentals of neural networks*. Prentice Hall, New York.
- Goktepe, A. B.; Agar, E.; Lav, A. H. 2004. Comparison of multilayer perceptron and adaptive neuro-fuzzy system on backcalculating the mechanical properties of flexible pavements, *ARI: The Bulletin of the Istanbul Tech. Univ.* 54(3): 65–77.
- Haykin, S. 1999. *Neural networks: A comprehensive foundation*. Prentice-Hall, Inc., NJ, USA.
- Hegazy, T.; Fazio, P.; Moselhi, O. 1994. Developing practical neural network applications using backpropagation, *Microcomputers in Civil Engrg.* 9(2): 145–159.
- Mehrotra, K.; Mohan, C. K.; Ranka, S. 1997. *Elements of artificial neural networks*. MIT Press, Cambridge, MA.
- NCHRP Synthesis of Highway Practice 38: Statistically Oriented End-Result Specifications. TRB, National Research Council, Washington, D.C., 1976.
- NCHRP Synthesis of Highway Practice 65: Quality Assurance. TRB, National Research Council, Washington, D.C., 1979.
- NCHRP Synthesis of Highway Practice 212: Performance-Related Specifications for Highway Construction and Rehabilitation. TRB, National Research Council, Washington, DC, 1995.
- Patel, A.; Thompson, M. R.; Harm, E.; Sheftick, W. 1997. Developing QC/QA specifications for hot mix asphalt concrete in Illinois, *Transportation Res. Rec.* 1575: 66–74.
- Patterson, D. 1996. *Artificial neural networks*. Prentice Hall, Singapore.
- Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. 1986. Learning integral representation by error propagation, in Rumelhart, D. E. et al. (Eds.). *Parallel Distributed Processing*, MIT Press, Cambridge, MA, 318–362.
- TR, 1996. Transportation Research Circular 457: Glossary of Highway Quality Assurance Terms. TRB, National Research Council, Washington, DC.
- Weed, R. M. 1995. Development of air voids specification for bituminous concrete, *Transportation Res. Rec.* 1491: 33–39.

NEURONINIŲ TINKLŲ PAKAITALO MODELIS MOKĖJIMO RIZIKAI KELIŲ STATYBOJE NUSTATYTI

A. Manik, K. Gopalakrishnan, A. Singh, S. Yan

S a n t r a u k a

Bendroji sąlyga kokybės kontrolės užtikrinimo (QC/QA) kelių tiesimo sutartyse yra užmokesčio nustatymas. Jį rangovas gauna atsižvelgdamas į statybos kokybę. Svarbu rangovui ir agentūrai išnagrinėti užmokesčio kiekį, kurio rangovas gali tikėtis gauti už tam tikrą statybų kokybę. Ankstesni tyrimai parodė, kad pasitelkiant kompiuterinį modeliavimą galima gauti geresnį, daug išsamesnį apmokėjimo vaizdą. Tai galima padaryti paprasčiausiai nustatant tikėtiną užmokesčių. Modeliavimo procesas rodo užmokesčio kitimą paisant kokybės ir gali pateikti veiksnus, nuo kurių priklauso kainos nustatymas. Tikimybinis modelis yra labai naudingas apskaičiuojant ir analizuojant užmokesčio riziką tiesiant kelių dangas. Tačiau tokie modeliai yra suvaržyti kompiuterinių reikalavimų, ir dažnai juos reikia susieti su paprastesniais modeliais norint greitinti sprendimų priėmimo procesą. Šiame straipsnyje tyrinėjamas neuroninių tinklų naudojimas pakeičiantiems modeliams sukurti norint teisingai nustatyti kelių dangos tiesimo užmokesčių. Rezultatai parodė, kad vidutinė paklaida, susijusi su neuroninių tinklų spėjimu, yra priimtina, tačiau kai kuriais atvejais paklaidos gali būti neleistinai didelės.

Reikšminiai žodžiai: kokybės kontrolės užtikrinimas, dirbtiniai neuroniniai tinklai, kelių statyba, mokėjimo rizika, stochastinis modeliavimas.

Dr. **Anshu MANIK** is currently working as Visiting Research Associate in the Department of Civil and Environmental Engineering at Michigan State University. Prior to this position he worked for Wolfram Research Inc. He completed his MS and Ph.D. in Civil Engineering from the University of Illinois at Urbana-Champaign (UIUC). He also holds the Certificate for Specialization in Computational Science and Engineering from UIUC.

Dr. **Kasthurirangan (Rangan) GOPALAKRISHNAN**. Research Assistant Professor in the Department of Civil Construction, and Environmental Engineering at Iowa State University, USA. Member of the American Society of Civil Engineers as well as an Editorial Board Member of the Journal of Artificial Intelligence, an International Scientific Committee Member of the International Journal of Computational Intelligence. Received the *Dwight D. Eisenhower Transportation Fellowship* award in 1999 and he completed his PhD in 2004 at the University of Illinois at Urbana-Champaign, USA. His research interests include the use of bio-inspired computing paradigms in civil engineering applications, simulation, imaging, and mechanics of transportation infrastructure systems.

Abhishek SINGH. A graduate student in the Department of Civil and Environmental Engineering at the University of Illinois at Urbana-Champaign. Member of the Environmental & Water Resources Institute of the American Society of Civil Engineers. His research interests lie in the analysis, modelling and optimization of complex environmental systems, particularly water resources related problems.

Shengquan YAN was a graduate student in the Department of Civil and Environmental Engineering at the University of Illinois at Urbana-Champaign. His research interests include environmental systems analysis using machine learning methods such as genetic algorithms, decision trees, support vector machines, and artificial neural networks.