Paper No. 03-3441

Forecasting Overall Pavement Condition with Neural Networks

Application on Florida Highway Network

Jidong Yang, Jian John Lu, Manjriker Gunaratne, and Qiaojun Xiang

Timely identification of undesirable crack, ride, and rut conditions is a critical issue in pavement management systems at the network level. The overall pavement surface condition is determined by these individual pavement surface conditions. A research project was carried out to implement an overall methodology for pavement condition prediction that uses artificial neural networks (ANNs). In the research, three ANN models were developed to predict the three key indices—crack rating, ride rating, and rut rating—used by the Florida Department of Transportation (FDOT) for pavement evaluation. The ANN models for each index were trained and tested by using the FDOT pavement condition database. In addition to the three key indices, FDOT uses a composite index called pavement condition rating (PCR), which is the minimum of the three key indices, to summarize overall pavement surface condition for pavement management. PCR is forecast with a combination of the three ANN models. Results of the research suggest that the ANN models are more accurate than the traditional regression models. These ANN models can be expected to have a significant effect on FDOT's pavement management system.

Pavement condition prediction models, which simulate the deterioration process of pavement condition and forecast the pavement condition over time, play a pivotal role in pavement management systems (PMSs). Much research has been done on pavement performance modeling. However, the pavement deterioration process is so complex that finding an appropriate functional form with which to model it, as in traditional modeling, is tedious. Hence, a new approach, which is biologically inspired, is taking over from its traditional counterpart for modeling complex processes. Typical models in this category are neural networks and genetic algorithms. Neural networks abstract the underlying relationships between dependent and independent variables from the exemplar data pairs and express them as weight matrices. A literature review shows the increasing popularity of this approach in pavement performance modeling. Lou et al. (*1*) developed a neural network model for forecasting pavement crack condition. Shekharan (*2*) developed neural network models for predicting the conditions of five families of pavement: original flexible, overlaid flexible, composite, jointed, and continuously reinforced concrete (*2*). Owusu-Ababio (*3*) applied neural networks to model the performance of thick asphalt pavement (≥152.4 mm). Attoh-Okine (*4*) applied a neural network to develop a pavement roughness progression model. The main objective for the current research is to develop a pavement performance prediction model that applies a neural network algorithm and implements the model in the Florida Department of Transportation's (FDOT's) PMS.

J. Yang, J. J. Lu, and M. Gunaratne, Department of Civil and Environmental Engineering, University of South Florida, 4202 East Fowler, ENB118, Tampa, FL 33620. Q. Xiang, Transportation College, Southeast University, 2 Si Pai Lou, Nanjing, People's Republic of China, 210096.

FDOT uses three key indices—crack index, ride index, and rut index—to capture the different attributes of pavement surface condition. In addition, FDOT uses a composite index called the pavement condition rating (PCR), the minimum of the three key indices, to represent the overall pavement condition. This representation implies that the three indices are equally important, and the lowest one represents the overall pavement condition. The cracks, ride, and ruts are rated on a scale of 0 to 10, where 10 indicates the best condition and 0 the worst. If PCR is greater than the decision threshold of 6.4, the pavement is considered to be in a sound condition; if PCR is equal to 6 and determined by ride rating, the pavement is not considered to be deficient when the speed limit of the pavement segment is less than 45 mph.

To forecast the overall pavement condition, FDOT uses two mathematical methods for each roadway segment: mean deterioration rate and simple linear regression. In practice, the method that best fits the prior trend of the condition data usually is chosen. However, since pavement performance is a nonlinear phenomenon, neither of the two methods has comprehensive application. Hence, alternative modeling methods based on sophisticated forecasting methodologies are sought for FDOT's PMS.

METHODOLOGY

An artificial neural network (ANN) is a parallel information processing system that has certain performance characteristics similar to biological neural networks. A neural net consists of a large number of simple processing elements called neurons. Each neuron is connected to other neurons by means of directed links, and each directed link has an associated weight. The weights acquired through the training process represent abstracted information from the data set, which is used by the net to solve a particular problem. To construct a neural network for solving a particular problem, three key components first must be determined: architecture, neuron activation function, and learning method.

Architecture

Significant effort is involved in the selection of ANN architecture. The efforts include determination of input and output variables, number of hidden layers, and number of hidden neurons in each hidden layer. 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 (*5*). Haykin (*6*) recommended using two hidden layers, one for extracting local features and one for extracting global features. However, with two hidden layers, the authors experienced a significant increase in the training time and a corresponding decrease in the efficiency of the training process. Funahashi (*7*) and Hornik et al. (*8*) separately proved that any continuous function can be approximated with an arbitrary accuracy by using the three-layered network. Thus, from a theoretical viewpoint, the three-layered network is adequate for function approximation. In practice, most neural network applications use only one hidden layer. Because of a still vague understanding of the impact of the variation of ANN architecture, the trial and error method is conventionally employed to select the appropriate number of neurons in the hidden layer.

Neuron Activation Function

Each neuron in an ANN is an independent processing element, having its own inputs and output. Its function is that of a distributed parallel computation (Figure 1). The output is calculated by the equation

$$O_j = f\left(\sum_{i=1}^{n} x_i w_i\right) \tag{1}$$

where

x_i = the *i*th input,
w_i = the connection weight associated with the *i*th input,
O_j = output of the *j*th neuron, and
f = transfer function.

The processing of each neuron is simply a weighted summation plus a function transfer. Five typical transfer functions generally are used as neuron activation functions, depending on the characteristics of the problem under investigation. These activation functions can be of different types: linear, linear threshold, step, sigmoid, and Gaussian. Among these, the sigmoid function is the one commonly used, because of its concise form and differentiability. Therefore, the sigmoid function is used as a neuron activation function in this research. Provided the input vector, $[x_1, x_2, \ldots, x_n]$, the output of each neuron calculated by the sigmoid transfer function can be expressed as

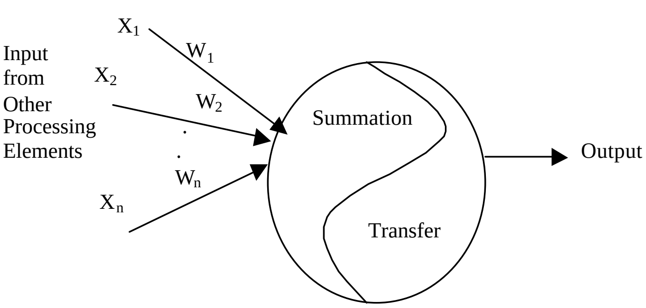

FIGURE 1 Artificial neuron function.

$$z = f(y) = \frac{1}{1 + e^{-a(y)}} \tag{2}$$

$$y = \sum_{i=1}^{n} w_i x_i \tag{3}$$

where

z = neuron output,
y = input to the transfer function,
a = gain of the sigmoid function, and
n = number of elements in the input vector.

Learning Method

The learning capability of ANNs is achieved by adjusting the signs and magnitudes of their weights according to learning rules that seek to minimize a cost or an error function. All learning methods can be classified into two categories: supervised learning and unsupervised learning. Supervised learning is a process that uses an external teacher or global information or both. Unsupervised learning relies on local information during the entire learning process by organizing presented data and discovering its emergent collective properties.

The backpropagation method, which is used in this research, falls into the category of supervised learning. It is the most widely used method for network learning. It provides a great opportunity for multidimension vector mapping. Because of its generality, the backpropagation neural network can be used to tackle a wide array of problems. Moreover, the backpropagation method offers a clear mathematical concept and ease of programming. These conveniences make backpropagation a versatile and pragmatic mechanism with which to implement neural networks. Many neural network software applications use backpropagation as the embedded training law. BrainMaker, used in this research effort, is one of these.

Following description of the architecture, neuron activation function, and learning method, a neuron network needs to be trained by using sample data to obtain the network parameters required for application. The training process consists of two steps. In the first step, the training patterns (a set of known input and output pairs) obtained from the database are fed into the input layer of the network. These inputs are then propagated through the network until the output layer is reached. The output of each neuron is computed by the transfer function, as shown in Equation 2, which "squashes" the range of input to be between 0 and 1.0. Then, a forward preprocessing error is calculated by using the following equation:

$$E_{\text{total}} = \frac{1}{2} \sum_{r=1}^{p} \sum_{k=1}^{m} \left(T_k^{(r)} - Y_k^{(r)}\right)^2 \tag{4}$$

where

E_{total} = square of the output error for all the patterns in the data sample,
p = number of patterns in the data sample,
m = number of neurons in the output layer,
$T_k^{(r)}$ = target value of neuron *k* for pattern *r*, and
$Y_k^{(r)}$ = output of neuron *k* for pattern *r* based on the sigmoid function $f(y)$.

In the second step, the error is minimized by backpropagation through the neural network. During this process, the individual error contribution caused by each layer is computed and distributed backward, and the corresponding weight adjustments are made to minimize the error. By using a gradient descend method, the back-propagation weight adjustment for the connections between the hidden layer and the output layer can be expressed as

$$w_{jk}(l+1) = w_{jk}(l) - \eta(l)\frac{\partial E_{\text{total}}}{\partial w_{jk}} + \alpha(l)\left[w_{jk}(l) - w_{jk}(l-1)\right] \qquad (5)$$

Similarly, weight adjustment for the connections between the input layer and the hidden layer can be written as

$$w_{ij}(l+1) = w_{ij}(l) - \eta(l)\frac{\partial E_{\text{total}}}{\partial w_{ij}} + \alpha(l)\left[w_{ij}(l) - w_{ij}(l-1)\right] \qquad (6)$$

where

$w_{ij}^{(1)}(l+1)$ = weight of link for training iteration $l+1$ between neuron i in the input layer and neuron j in the hidden layer,
$w_{ij}^{(1)}(l)$ = weight of link for training iteration l between neuron i in the input layer and neuron j in the hidden layer,
$w_{ij}^{(1)}(l-1)$ = weight of link for training iteration $l-1$ between neuron i in the input layer and neuron j in the hidden layer,
$\eta(l)$ = positive constant termed the learning coefficient at iteration l, and
$\alpha(l)$ = momentum term used to achieve rapid convergence and avoid numerical vibration during training.

The discussed training approach is called batch training. In batch training, the weights are adjusted after all samples are processed. Batch training can guarantee E_{total} to decrease gradually and can speed up convergence as well. Training is considered complete when overall error E_{total} is lowered to an acceptable level.

MODEL DEVELOPMENT

Data Preprocessing

Preprocessing of the database for neural network modeling is done through a sequence of steps. First, missing data points in the database are supplemented by linear interpolation. Then, moving averaging is carried out to smooth the time series of index data. Smoothing is done to reduce or eliminate the illogical signals of the time series data. For example, in the case of the crack index, the moving average is computed by the following equation, using a three-step moving range:

$$\text{crack}(t) = \frac{\text{crack}(t-1) + \text{crack}(t) + \text{crack}(t+1)}{3} \qquad (7)$$

where crack(t) is the measured crack index at year t.

Figure 2 illustrates the usefulness of the moving average technique for improving the time series of index data. A SAS program was developed to automatically preprocess the database and generate the data sets for training, testing, and validation of the ANN models for the crack, ride, and rut indices. The variables corresponding to the model for cracks are

- Crack($t-4$) = crack index for year ($t-4$),
- Crack($t-3$) = crack index for year ($t-3$),

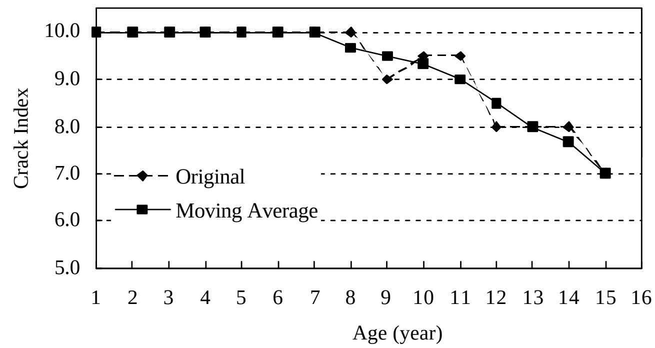

FIGURE 2 Comparison of original series and moving average series.

- Crack($t-2$) = crack index for year ($t-2$),
- Crack($t-1$) = crack index for year ($t-1$),
- Crack(t) = crack index for year (t),
- Crack($t+N$) = crack index for year ($t+N$),
- Age = section age since the last major maintenance activity, and
- Cycle = pavement maintenance cycle of roadway section,

where N(1,2,3,4,5) represents the forecasting interval.

Similar data sets were developed for rut and ride indices as well.

Neural Network Architecture Design

Similar to the traditional modeling process, where the objective is to estimate a set of coefficients for a particular function form of specification, the objective of ANN modeling is to attain a set of weight matrices, which is the abstracted underlying knowledge from the example data after many loops of training.

However, to use the neural network to solve a particular real-life problem, a framework must be designed first, according to the characteristics of the problem under study. The objective of the framework design is to determine whether a nested ANN architecture is necessary for the problem and how the components of the architecture should communicate with each other. The next stage is to design the architecture of each ANN submodel. The ANN architecture design process is actually a decision-making process, which includes determining the number of layers, the number of neurons in each layer, the variables that should be included in the input layer and the output layer, and so forth. Once ANN architecture is determined, the ANN model is trained, tested, and evaluated.

In this research effort, the data used for training and testing did not include the latest-year data (2001) in the database. The 2001 data are reserved for model evaluation.

Model Framework Design

To enhance the performance of the ANN models, two families of models were developed for two types of pavement—flexible and rigid. Three submodels were developed for flexible pavements for the three key indices associated with flexible pavement condition: crack, ride, and rut. Two submodels were developed for rigid pavements for the two key indices associated with rigid pavement condition: crack and ride. The PCR model is a de facto combination of these submodels, as shown in Figure 3 and Figure 4.

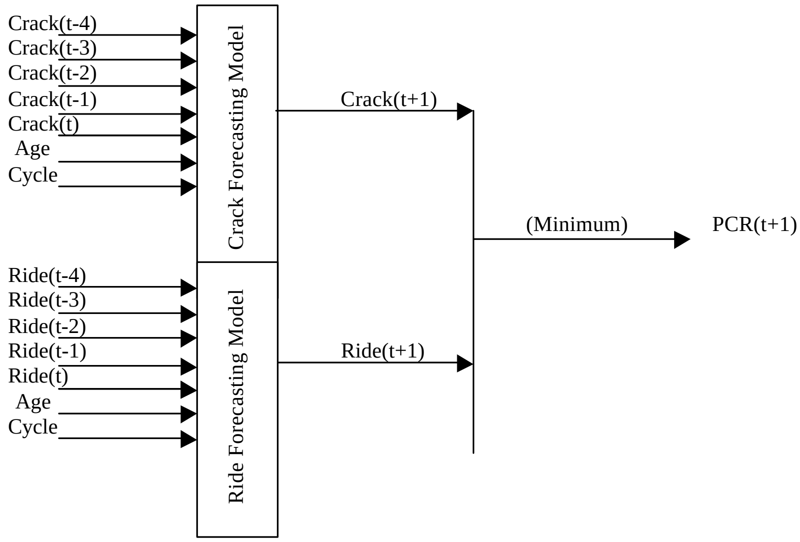

FIGURE 3 Architecture of PCR forecasting model (rigid pavements).

Architectures of Neural Network Models

The architectures selected after a process of trial and error are detailed in Table 1 (where CI = crack index, RT = rut index, and RD = ride index). The schematic architectures of a 1-year forecasting model for the crack index are illustrated in Figure 5, where *t* represents the current year and *n* represents the number of future years for which forecasts are made.

Model Training and Testing

Given the architecture of ANNs, the weights of links among the neurons are resolved through the training process. The training process presents all example pattern pairs in the training data set to the network and adjusts the weights of the connections according to the weight-adjustment rule as defined by Equations 5 and 6. The training process is considered complete when the total error, as defined in

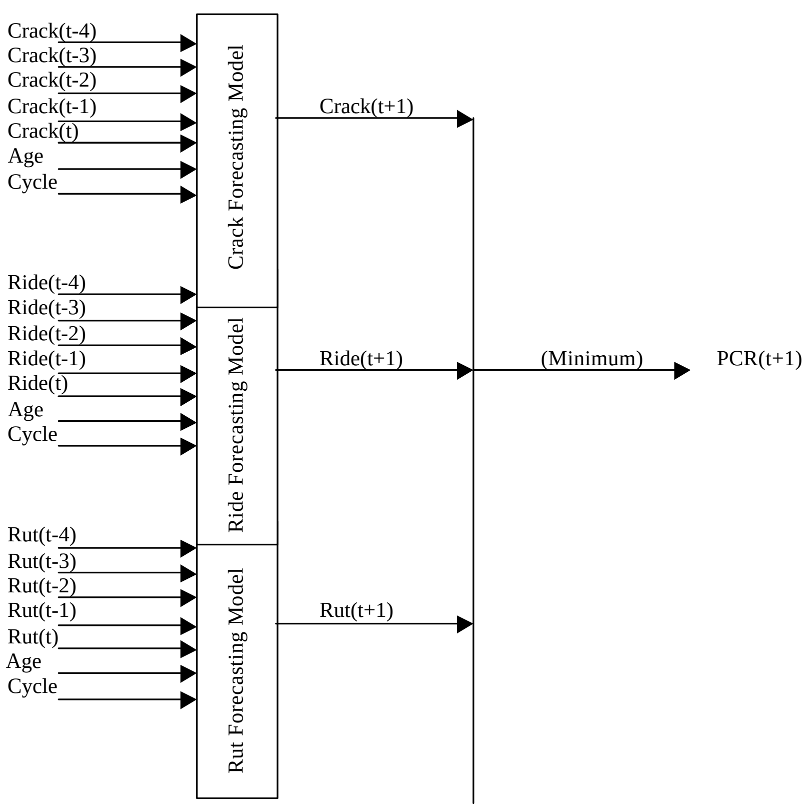

FIGURE 4 Architecture of PCR forecasting model (flexible pavements).

TABLE 1 Selected Network Architectures

Model			Inputs	Output	Architecture (*l*-*m*-*n*)
Flexible Pavements	Crack	1 year	CI(t-4), CI(t-3), CI(t-2), CI(t-1), CI(t), Cycle, Age	CI(t+1)	11-11-1
		2 year		CI(t+2)	11-14-1
		3 year		CI(t+3)	11-15-1
		4 year		CI(t+4)	11-18-1
		5 year		CI(t+5)	11-22-1
	Ride	1 year	RD(t-4), RD(t-3), RD(t-2), RD(t-1), RD(t), Cycle, Age	CI(t+1)	11-11-1
		2 year		CI(t+2)	11-14-1
		3 year		CI(t+3)	11-15-1
		4 year		CI(t+4)	11-18-1
		5 year		CI(t+5)	11-22-1
	Rut	1 year	RT(t-4), RT(t-3), RT(t-2), RT(t-1), RT(t), Cycle, Age	CI(t+1)	11-11-1
		2 year		CI(t+2)	11-14-1
		3 year		CI(t+3)	11-15-1
		4 year		CI(t+4)	11-18-1
		5 year		CI(t+5)	11-22-1
Rigid Pavements	Crack	1 year	CI(t-4), CI(t-3), CI(t-2), CI(t-1), CI(t), Cycle, Age	CI(t+1)	9-10-1
		2 year		CI(t+2)	9-12-1
		3 year		CI(t+3)	9-14-1
		4 year		CI(t+4)	9-16-1
		5 year		CI(t+5)	9-18-1
	Ride	1 year	RD(t-4), RD(t-3), RD(t-2), RD(t-1), RD(t), Cycle, Age	CI(t+1)	9-10-1
		2 year		CI(t+2)	9-12-1
		3 year		CI(t+3)	9-14-1
		4 year		CI(t+4)	9-16-1
		5 year		CI(t+5)	9-18-1

NOTE: "*l*-*m*-*n*" for neural network architecture denotes the neural network structure with *l* input neurons, *m* hidden neurons, and *n* output neurons.

Equation 4, reduces to an acceptable level, which is predefined according to characteristics of the problem under study. After completion of the training process, the trained network is exposed to the testing data set to ensure the success of the training. The testing data pairs are fed into the trained ANN, and the testing error is calculated accordingly. If the testing error is calculated to be within the acceptable level defined for testing, the ANN model is considered a reasonable model.

PERFORMANCE EVALUATION OF ANN MODELS

With completion of the training and testing processes, the neural network attains the capabilities of simulating the pavement condition deterioration mechanism and forecasting the future pavement condition. Then, the subsequent step is to evaluate the performance of the developed neural network models. A reserved data set including only these pavement data for 2001 was used for the evaluation. To ensure unbiased evaluations, irrational data that showed improved pavement condition with time were discarded.

Comparison of Forecasting Errors with Autoregressive Model

To evaluate the performance of the ANN models, three autoregressive (AR) models were developed to forecast the three key indices for comparison purposes. In the AR models, the three distress indices after n years—$\mathrm{CI}(t+n)$, $\mathrm{RT}(t+n)$, $\mathrm{RD}(t+n)$—were forecast by linearly extrapolating corresponding crack, rut, and ride indices in the previous 3 years by using Equation 8. Then, the PCR value was forecast as the minimum of the three individual indices. Table 2 compares forecasting errors of the ANN model and the AR model. It can be seen that the ANN model was more accurate than the AR model for average error and root-mean-square error (RMSE).

$$\mathrm{Index}(t+n) = a + b(t+n) \tag{8}$$

where

$$b = \frac{\sum_{i=0}^{2}(t-i)\times \mathrm{Index}(t-i) - \dfrac{\left(\sum_{i=0}^{2}(t-i)\right)\left(\sum_{i=0}^{2}\mathrm{Index}(t-i)\right)}{3}}{\sum_{i=1}^{3}(t-i)^2 - \dfrac{\left(\sum_{i=0}^{2}(t-i)\right)^2}{3}}$$

$$a = \frac{\sum_{i=0}^{2}\mathrm{Index}(t-i)}{3} - b\times\frac{\sum_{i=0}^{2}(t-i)}{3}$$

$\mathrm{Index}(t-i) = \mathrm{CI}(t-i)$, $\mathrm{RT}(t-i)$, or $\mathrm{RD}(t-i)$;
$\mathrm{Index}(t+n) = \mathrm{CI}(t+n)$, $\mathrm{RT}(t+n)$, or $\mathrm{RD}(t+n)$; and
t = pavement age.

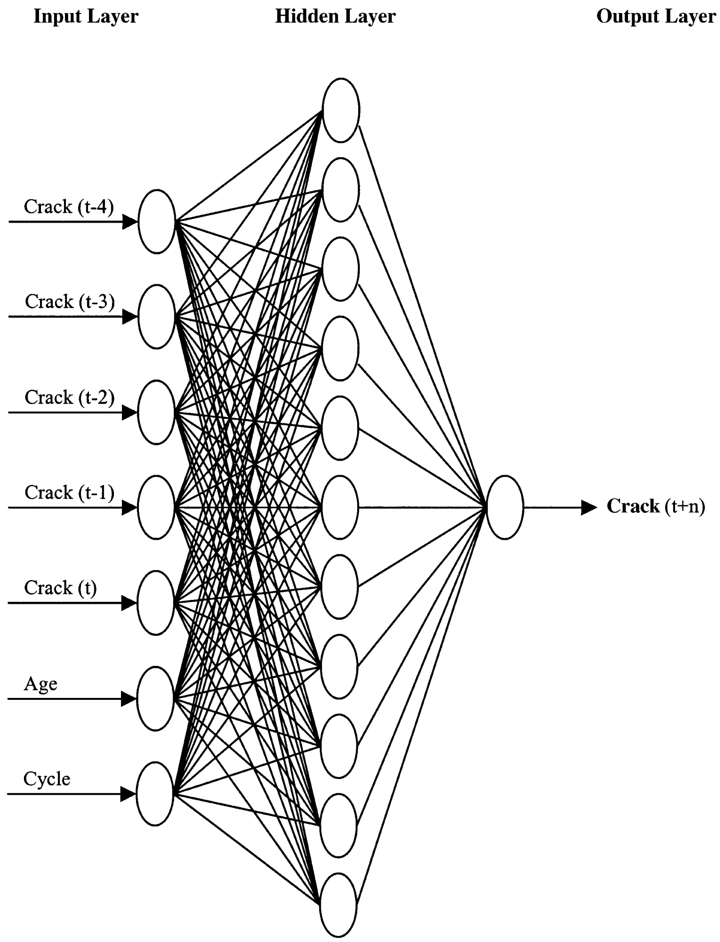

FIGURE 5 Architecture of 1-year crack index forecasting model.

Goodness of Fit

Goodness of fit is a common approach used to evaluate performance of models. In this research, the performance of the PCR forecasting models was further evaluated by comparing the goodness of fit of the ANN models and the corresponding AR models. For comparison purposes, R^2 was calculated against the ideal correlation line, $\text{PCR}_{\text{act}} = \text{PCR}_{\text{pred}}$, by using the following equation:

$$R^2 = 1 - \frac{\sum (\text{PCR}_{\text{act}} - \text{PCR}_{\text{pred}})^2}{\sum (\text{PCR}_{\text{act}} - \text{PCR}_{\text{avg}})^2} \qquad (9)$$

where

PCR_{act} = actual value of PCR,
PCR_{pred} = value of PCR predicted by neural network model, and
PCR_{avg} = average value of PCR.

This shows that the ANN models have higher R^2 values than those of corresponding AR models. Moreover, the AR models become useless when the forecasting interval exceeds 2 years, as seen by negative R^2. Table 3 summarizes goodness of fit for the models in R^2. Correlation graphs for 1-year forecasting are shown in Figure 6. It can be seen from Figure 6 that the data points of the ANN model are more closely clustered around the correlation line

TABLE 2 PCR Forecasting Errors of Models

Years and Model of Forecast		Average Absolute Error	RMSE
1 year	ANN	0.0479	0.0664
	AR	0.1268	0.1969
2 year	ANN	0.0644	0.0913
	AR	0.1699	0.2436
3 year	ANN	0.0791	0.1118
	AR	0.1892	0.2723
4 year	ANN	0.0889	0.1284
	AR	0.2127	0.2977
5 year	ANN	0.1054	0.1496
	AR	0.2312	0.3215

TABLE 3 R^2 **Comparisons of Models**

Years and Model of Forecast		Goodness of Fit (R^2)	
		Flexible Pavements	Rigid Pavements
1 year	ANN	0.88	0.79
	AR	0.58	0.39
2 year	ANN	0.76	0.55
	AR	0.29	0.2
3 year	ANN	0.59	0.52
	AR	-0.22	-0.15
4 year	ANN	0.48	0.4
	AR	-0.49	-0.28
5 year	ANN	0.38	0.2
	AR	-0.74	-0.14

than they are in the AR model, and the ANN model exhibits much higher R^2 values (0.88 for flexible pavement and 0.79 for rigid pavement) than does the AR model (0.55 for flexible pavement and 0.39 for rigid pavement).

Case Study of Individual Sections

Several typical sections that were not included in the training process were initially set aside for the case study. The PCR forecasts of the ANN model and the AR model are plotted on the same graph, as shown in Figure 7. It can be seen that the ANN model and the AR model have comparable forecasting accuracy for 1- and 2-year forecasting. However, when the forecasting interval increases to 3, 4, and 5 years, the ANN model outperforms the AR model. This pattern becomes more pronounced when pavements tend to deteriorate at a higher rate. It can be seen that the forecasts of the AR model tend to lag behind the observed values, which often occurs in the conventional time series models.

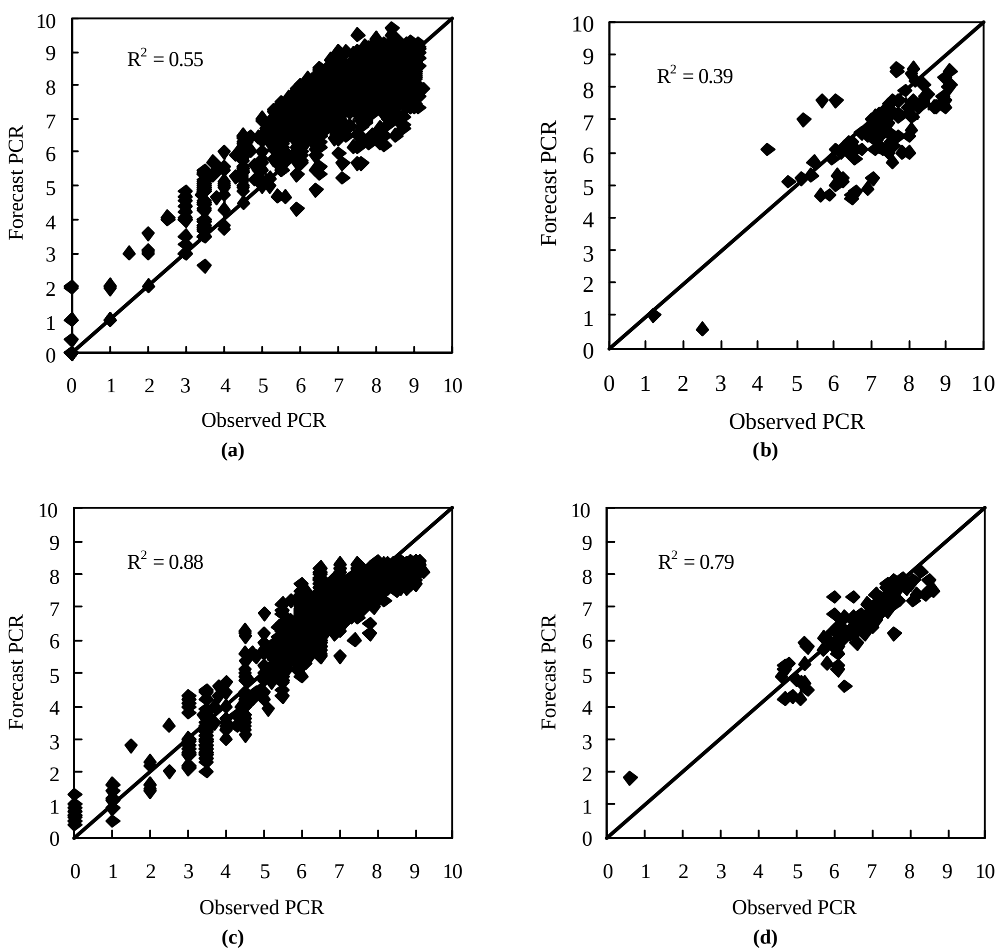

FIGURE 6 Goodness of fit: (*a*) AR model, flexible pavements; (*b*) AR model, rigid pavements; (*c*) ANN model, flexible pavements; (*d*) ANN model, rigid pavements.

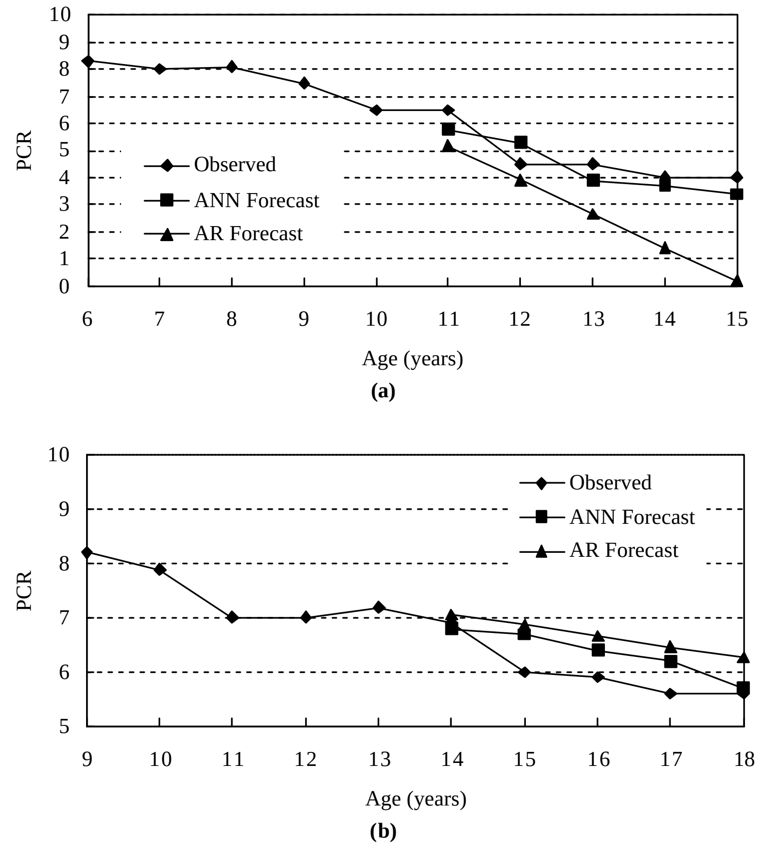

FIGURE 7 Case study: (*a*) Section 1, flexible pavement, Cycle 2; (*b*) Section 2, rigid pavement, Cycle 1.

IMPLEMENTATION

To implement the ANN models in the FDOT PMS, the pavement performance forecasting system (PPFS) was developed. PPFS software contains three major operational modules: data preprocessing, pavement condition forecasting by ANN models, and data postprocessing. The software architecture is shown in Figure 8. Provided with the FDOT original database, the first module is designed to preprocess the database, which includes adding missing data and use of the moving average. Then, the processed database is fed into the second module, which uses the developed ANN models to forecast the future pavement condition. Following forecasting, the postprocessing module summarizes the forecast results in formats that can be easily understood and directly used by FDOT for budget planning at the network level and rehabilitation activities at the project level. To integrate the three major modules in the forecasting system, a main interface program is needed. Because of its popularity and convenience of implementation, the widely used programming language Visual Basic is used to code the main interface program.

Data Preprocessing Module

The original FDOT pavement condition survey database, which is in SAS format, is the original input to the software. A SAS program was developed to preprocess the database, which is called the main interface program. The SAS program carries out the three data preprocessing tasks: patching, filtering, and conversion. The data-patching step adds missing data points, if needed, to the database. The data-filtering step filters out high-frequency noise in the time series of pavement index data. The data-conversion step converts the data format of the original database into a format that can be directly used for neural network modeling and forecasting.

The output of this module is the preprocessed database, which is used as input to the next module, the ANN forecasting module.

ANN Forecasting Module

The ANN forecasting module consists of 25 ANN forecasting models, shown in Table 1. To facilitate forecasting, a user-friendly interface is provided that allows users to select the forecasting detail of interest. Specifically, users can select the forecasting interval from 1 to 5 years and the forecasting domain of state or district. According to the user's selection, the main program calls the corresponding ANN models to perform the desired forecasting.

Data Postprocessing Module

In the data postprocessing module, the forecasted results are summarized at two levels: state and district. At the state level, lane-mile deficiency forecasts for seven districts are provided. At the district level, lane-mile deficiency forecasts for three systems—arterial, Interstate,

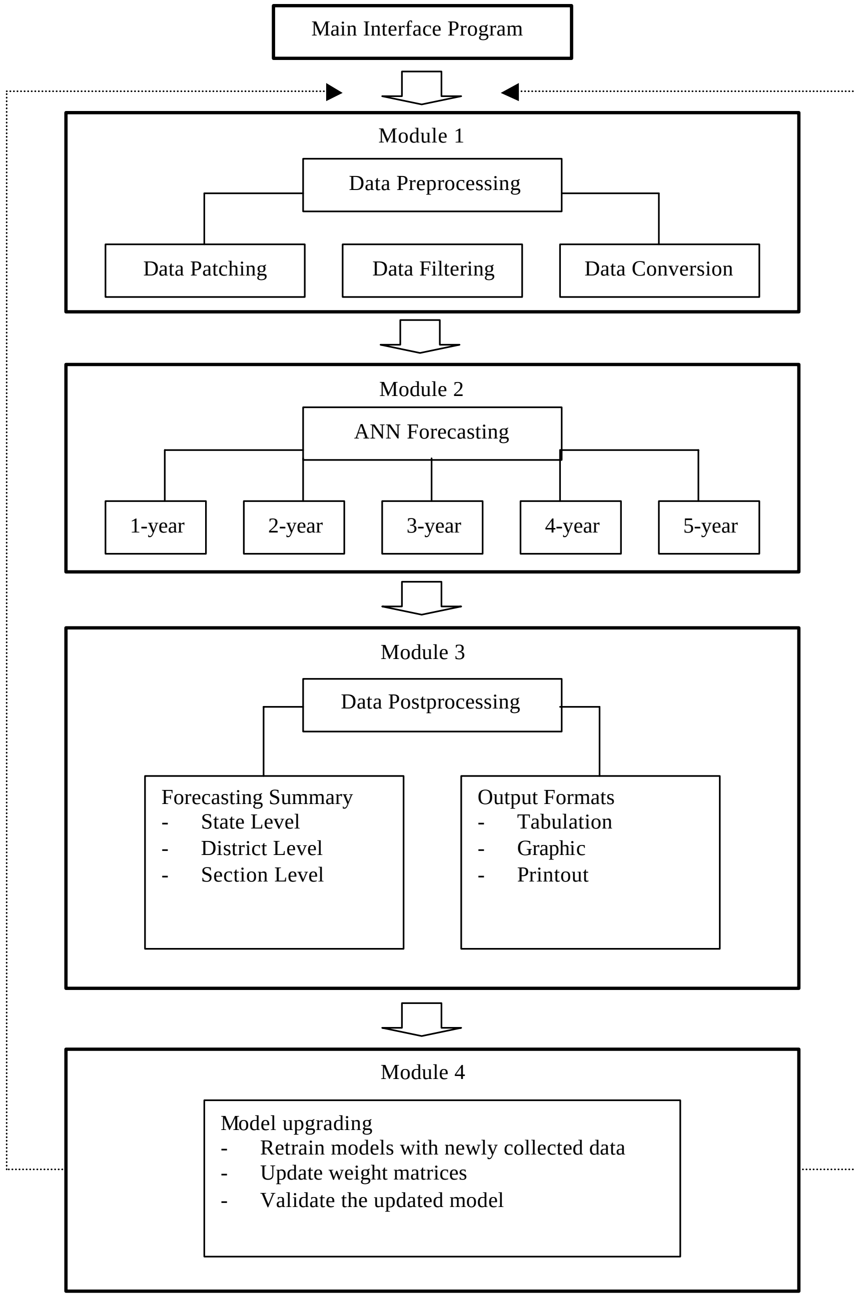

FIGURE 8 PPFS software architecture.

and turnpike—are provided. For the user's convenience, a detailed list of section-by-section forecasting is provided in addition to the two levels of forecasts. As the output of the module, the forecast results are presented in three formats: tabular, graphic, and listing. Users can either print the output or save the outputs into files for later assessment.

CONCLUSIONS

This research was conducted to develop appropriate pavement performance forecasting models based on ANNs and the corresponding software that can implement these models in FDOT's PMS. It was shown that the ANN model provided an effective alternative to the current pavement performance forecasting models. By undergoing training with historical pavement condition data, the trained ANN models were able to extract underlying information contained within the historical database and then make reasonable forecasts of pavement condition.

The original database typically is not suitable to be used directly for the ANN model development. Accordingly, a data preprocessing procedure was used to rectify the database and transform the database into formats that can be directly used for ANN modeling purpose. A software package was developed to directly implement the ANN models in FDOT's PMS, and Visual Basic was used for the code development.

A commercial professional software package was used for the ANN model training. Use of external software does not cause any difficulty, because separate text files are used to store the trained network weight matrices. In practice, the highway agency that uses this software needs to update the network weight matrix at regular intervals by retraining the ANN models with newly available data and replacing these text files storing weight matrices.

As a key component of PMS, pavement performance models play a crucial role in PMS at the network level, where forecasting results provide key information to highway agencies for decision making on the overall maintenance and budget plan. Therefore, improved accuracy of pavement performance models could make a considerable difference in expenditures for pavement maintenance and rehabilitation.

ACKNOWLEDGMENTS

This paper is based on research sponsored by the Florida Department of Transportation. The authors thank Bruce Dietrich for his suggestions and assistance and Sandra Kang for her help with database processing.

REFERENCES

1. Lou, Z., M. Gunaratne, J. J. Lu, and B. Dietrich. Application of Neural Network Model to Forecast Short-Term Pavement Crack Condition: Florida Case Study. *Journal of Infrastructure Systems,* Vol. 7, No. 4, 2001, pp. 166–171.
2. Shekharan, A. R. Pavement Performance Prediction by Artificial Neural Networks. *Computational Intelligence Applications in Pavement and Geomechanical Systems,* A. A. Balkema, Rotterdam, Netherlands, 2000, pp. 89–98.
3. Owusu-Ababio, S. Application of Neural Networks to Modeling Thick Asphalt Pavement Performance. *Artificial Intelligence and Mathematical Methods in Pavement and Geomechanical Systems,* A. A. Balkema, Rotterdam, Netherlands, 2000, pp. 23–30.
4. Attoh-Okine, N. O. Predicting Roughness Progression in Flexible Pavements Using Artificial Neural Networks. *Proc., Third International Conference on Managing Pavements,* Vol. 1, 1994, pp. 55–62.
5. Lawrence, J., and J. Fredrickson. *BrainMaker User's Guide and Reference Manual,* 7th ed. California Scientific Software, Nevada City, Calif., 1993.
6. Haykin, S. *Neural Networks—A Comprehensive Foundation.* MacMillan College Publishing, New York, 1994.
7. Funahashi, K. On the Approximate Realization of Continuous Mappings by Neural Networks. *Neural Networks,* Vol. 2, No. 3, 1989, pp. 183–192.
8. Hornik, K., M. Stinchcombe, and H. White. Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks,* Vol. 2, No. 5, 1989, pp. 359–366.

Publication of this paper sponsored by Committee on Pavement Management Systems.