ARI The Bulletin of the Istanbul Technical University VOLUME 54, NUMBER 3
Communicated by Sondan Durukanoğlu Feyiz

Comparison of Multilayer Perceptron and Adaptive Neuro-Fuzzy System on Backcalculating the Mechanical Properties of Flexible Pavements

A. Burak Göktepe
Department of Civil Engineering, Ege University, 35100 Bornova, Izmir, Turkey
Emine Ağar, and A. Hilmi Lav*
Faculty of Civil Engineering, Istanbul Technical University, 34469 Maslak, Istanbul, Turkey
(Received 31 December 2004)

Nondestructive testing (NDT) is the integral part of the performance evaluation of flexible pavements. In all NDT methods, Falling Weight Deflectometer (FWD) is probably the most popular technique. Basically, it measures time-domain deflections from numerous road sections emerging by the applied impulse load. In order to characterize the structural integrity of considered pavement system, it is required to make an inversion for the calculation of mechanical pavement properties using a backcalculation tool covering both a forward pavement response model and an optimization algorithm. On the other hand, backcalculation problem can also be solved by an adaptive system using a supervised learning algorithm. In this manner, multilayer perceptron (MLP) and adaptive neuro-fuzzy system (ANFIS) methodologies, popular universal functional approximating techniques of Artificial Intelligence (AI), are appropriate for pavement backcalculation problem. Therefore, two-phased (forward and backward) structure of traditional backcalculation approaches is reduced into one step with the help of the supervised learning mechanisms of MLP and ANFIS. In this study, these methodologies are both employed to backcalculate mechanical properties of flexible pavements and compared in terms of modeling precision, uncertainty handling, computational expense, and data requirements. Results indicated that, both techniques are valid and have certain advantages over each other and should be preferred with respect to quantity and quality of the data at hand. In addition, AI-based supervised nonlinear mapping techniques not only exhibit precise backcalculation results, but also enable real-time pavement analyzing abilities.

Keywords: FWD, NDT, backcalculation, flexible pavements, MLP, ANFIS

1. Introduction

Performance evaluation of flexible highway pavements is fundamentally carried out for two purposes, which are the structural evaluation of existing pavement system and the quality control of a new pavement construction. Nondestructive testing (NDT) methods are the common way of evaluating the pavement's structural condition since they do not damage the structure and applied more rapidly. Among all NDT methods, Falling Weight Deflectometer (FWD) is the most widely used technique because of its ability to successfully simulate traffic loadings and capacity to produce larger amount of deflection data in unit time [1-3]. Basically, FWD measures time-domain deflection values from numerous road sections emerging by the applied impulse load. In traditional methods, deflections obtained from FWD test are commonly utilized for the backcalculation of mechanical pavement properties using specific software tools [4-7]. In this context, there are two calculation directions in these codes, namely forward and backward. In forward process, deflections are calculated for considered traffic loading, pavement structure, and initial mechanical parameters. Therefore, an appropriate pavement stress-strain analysis method, such as layered elastic theory, elastodynamic Green function solution, and Finite Element Method (FEM), should be employed in forward direction of calculation. Through backward direction, calculated deflections are compared with deflections measured by FWD and new mechanical properties are estimated by a parameter identification routine. Consequently, this optimization steps are performed until the discrepancy between calculated and measured deflections stays under a certain value. The fact is that this iterative process may take considerable time and require extensive computational power.

Multilayer perceptrons (MLPs) are a class of artificial neural network (ANN) structures, fo-

*Corresponding author: A. Hilmi Lav

cuses on building intelligent codes that mimic the learning mechanism of a human brain by constituting a parallel-connected network model. In a MLP model, once the system is trained, network can calculate outputs as a functional mapper using last updated network parameters. This is also the reason why MLPs are called by "universal functional approximaters"[8]. Under the way of this, when using ANN methodology for pavement backcalculation problems, it is possible to simulate the considered inverse mapping in real-time [9-11].

The idea of ANN-based backcalculation was first brought up by Meier and Rix [12], who considered MLP ethodology for the SASW test data inversion and for the backcalculation of flexible pavement layer properties. Later, Meier and Rix [9] published the keynote study on the suitability of ANN methodology on flexible pavement backcalculation process. Successively, Meier and Rix [10] presented another ANN-based backcalculation model involving dynamic aspects and rigid bottom depth concepts. In addition to these initial attempts, several researchers also focused on ANN-based pavement backcalculation models and implemented similar results [13-16]. It should be noted that, MLP is an adaptive system utilizing supervised learning algorithms, such as gradient descent, Lavenberg-Marquardt, and scaled conjugate gradient. For that reason, it can solely learn the behavior given by input/output data pairs; thus, there is no underlying material model and mechanical analysis methodology. As a result, the performance of an ANN-based backcalculation model is sensitive to quality and quantity of training data [11].

Adaptive neuro-fuzzy inference system (ANFIS) is another AI-based supervised learning methodology, making inferences by fuzzy logic and shaping fuzzy membership functions using neural network learning. Basically, ANFIS model iteratively determines membership functions to produce correct outputs, and simulates nonlinear input-output mapping. Under the view of this, ANFIS can also be employed for analogous adaptive backcalculation process and the technique is quite accurate and fast with respect to FEM and layered elastic approaches.

In this study, for the flexible pavement backcalculation problem, the susceptibility and relative performances of MLP and ANFIS methodologies are comprehensively compared in terms of modeling precision, uncertainty handling, computational expense, and data requirements. Training and testing data sets were synthetically obtained by Finite Element Methodology (FEM), and a total of 1500 data patterns were generated. In addition, the advantages and drawbacks of these AI methods are also evaluated with the consideration of the quantity and appropriateness of data.

2. Nondestructive Testing and Backcalculation of Flexible Pavements

The determination of mechanical properties of pavement layers by NDT methods is crucial for the structural evaluation of existing flexible pavements. The philosophy of the NDT methodology for the structural performance of the pavement system is that the structural integrity is inversely proportional with the amount of surface deflections utilized by applied load. Fundamental differences among deflection based NDT methods come from the fact that loading type and deflection measurement locations are different for each method. In general, applied loads are divided into three categories, namely, static, steady-state vibratory, and time domain impulse. Static loading is the simplest case, which cannot behave like the actual traffic loads.

In the time domain impulse loading, an impulse load is applied on pavement surface and deflection data is recorded in time domain. Generally, there are several sensors to measure the deflection values on different points of pavement surface. Falling Weight Deflectometer is an impulse loading device. In FWD test, a falling mass (an impulse load within the range of 6.7kN-156kN) is dropped on pavement surface, and transient deflections are recorded at each geophone. The impulse load is applied by a circular plate and a rubber seal is placed between plate and pavement surface in order to reduce the instant impact effect. In addition, transient surface deflections are measured at different locations (usually at 7 locations) by geophones. Consequently, peak values for each geophone are used to plot deflection basin curve [17, 18].

Backcalculation process in pavement system is the numerical analysis of measured surface deflections, which is performed for the estimation of layer stiffness parameters (namely, moduli). In order to accomplish this, measured deflections are iteratively matched with calculated deflections obtained by equivalent pavement response model. Iterations are continued until a close match between measured and calculated deflection values are satisfied. In this context, numerous backcalculation techniques were developed for the backcalculation of pavement layer moduli so far. The fundamental discrepancies among developed backcalculation models are the type of forward response model and the optimization procedure carried out for the determination of appropriate layer modulus values [1, 4, 5, 7, 19].

Basically, pavement response analyses can be considered as either static or dynamic. Static approaches are based on the layered elastic theory or finite element method (FEM) for linear or nonlinear material behaviors. In dynamic response analysis, loading can be either impulsive or vibratory. Eventually, deflection data are obtained in frequency domain or in time domain for impulse loads, and in steady state for vibratory loads. Additionally, in dynamic analyses, elastodynamic methods (such as Green function solution) or dynamic FEMs are employed to calculate surface deflections. For static and dynamic backcalculation analyses, optimization process can be performed using parameter identification routine (such as linear and nonlinear least squares), database search method, and genetic algorithm [3, 5, 7, 15, 17, 20, 21].

Besides existing advantages of dynamic approach, it has several obstacles coming from the complexity and computational expense of dynamic analyses. Furthermore, in many problems, it is hard to get all necessary data required for a dynamic analysis. For these reasons, static approaches are preferred in the majority of pavement backcalculation studies, because of their simplicity and acceptable error ranges.

3. Multilayer Perceptron (MLP) and Adaptive Neuro-Fuzzy Inference System

Artificial neural networks (ANNs) are parallel connectionist structures, which simulate the working network of neurons in human brain. Basically, as in human brain, artificial neural networks (ANNs) consist of neurons, which are parallel connected to each other via synapses. The term perceptron is equated with a processing unit including a single neuron, synaptic weights, and bias term. In a perceptron, input signals are accumulated after incorporating with synaptic weights. Successively, total impulse is compared with bias term and activation potential (ν_j) is calculated. Consequently, the output signal is produced by the normalization of activation potential to a certain range. Mathematical representation of a perceptron is given below [8]

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

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

Multilayer perceptrons (MLPs), also referred as multi layer feedforward neural networks, comprise an input layer, one or more hidden layer, and an output layer. Learning 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. For a given training data consists of input-output patterns, values of synaptic weights in a MLP are iteratively updated by a learning algorithm to approximate the target behavior. This update process is usually performed by backpropagating the error signal layer by layer and adapting synaptic weights with respect to the magnitude of error signal. The first backpropagation learning algorithm for use with MLP structures was presented by Rumelhart [22]. In this algorithm, error energy is the generalized value of all errors that is calculated by the least-squares formulation as follows [8]:

$$E = \frac{1}{mN} \sum_{k=1}^{N} \sum_{j=1}^{m} \left(y_j^k - t_j^k\right)^2 , \qquad (2)$$

where m is the number of neurons in output layer, N is number of training patterns, t_j^k is the target value of processing neuron. Essentially, this algorithm changes synaptic weights along the negative gradient of error energy function; thus, weight changes are proportional with the magnitude of error energy. The local error gradient is defined by:

$$\delta_n = \frac{\partial E_n}{\partial w_n} \qquad (3)$$

and, the formulation of weight update is given as:

$$\Delta w_n = \alpha \Delta w_{n-1} + \eta \delta_n y_n \qquad (4)$$

where Δw is weight update, η is learning rate parameter that can be selected from the range $[0, 1]$, δ is local error gradient, y is output signal, α indicates momentum term varying within $[0, 1)$, and n represents the processing neuron. [8, 22].

Standard backpropagation learning algorithm usually exhibits poor performance for large-scale problems, and its success is related with learning rate and momentum term parameters. Conjugate gradient algorithms are good choices to handle the optimization of large-scale problems. Virtually, conjugate gradient algorithms use an adaptive learning rate parameter that determines the step size of weight update to reach the global minimum of the performance function. The step size is adjusted separately in each step with accordance to the conjugate direction utilizing the following expression [23]:

$$\Delta \mathbf{w}_n = \mathbf{w}_{n+1} - \mathbf{w}_n = \alpha_n \mathbf{p}_n \,, \tag{5}$$

where p is search direction vector, and α_n is step size calculated by:

$$\alpha_n = \frac{\mathbf{p}_n^T \mathbf{r}_n}{\mathbf{p}_n^T \mathbf{H}_n} \,, \tag{6}$$

where $\mathbf{H}$ is Hessian matrix, and $\mathbf{r}$ is the negative error gradient vector. Algorithmically, conjugate direction is determined by a recursive process by the following expression:

$$\mathbf{p}_{n+1} = \mathbf{r}_{n+1} + \beta_n \mathbf{p}_n \tag{7}$$

in which β_n is scaling factor. That is, conjugate gradient methods approximate the step size using a line search routine to determine optimal step size minimizing error energy along the line $\mathbf{w}_n + \alpha \mathbf{p}_n$. However, this is the major drawback of these methods results in computational inefficiency. For that reason, scaled conjugate algorithm that does not require any line search routine was developed by [23]. In this new algorithm, in order to avoid a line searching routine, the Hessian matrix is described as follows:

$$\mathbf{H}_n = \frac{E'(\mathbf{w}_n + \sigma_n \mathbf{p}_n) - E'(\mathbf{w}_n)}{\sigma_n} \,, \tag{8}$$

where σ_n is the convergence factor used for the determination of second order derivatives. However, Eq. 5 can be used for positive and definite Hessian matrix, but this is not valid for every situation. In order to avoid this danger, a scalar parameter is set into this equation and the resulting equation is as follows:

$$\mathbf{H}_n = \frac{E'(\mathbf{w}_n + \sigma_n \mathbf{p}_n) - E'(\mathbf{w}_n)}{\sigma_n} + \lambda_n \mathbf{p}_n \,, \tag{9}$$

in which λ_n is Lagrange multiplier named by Marquardt parameter, which is updated iteratively. This algorithm is only reliable within the territories of a small region around the searching point. The extent of reliable region is controlled by Marquardt parameter, and this scalar is adjusted gradually to regulate the indefiniteness of the Hessian matrix. Consequently, in scaled conjugate gradient algorithm, the success of local quadratic convergence is determined by the following equation:

$$\chi_n = \frac{2\delta_n (E(\mathbf{w}_n) - E(\mathbf{w}_n + \sigma_n \mathbf{p}_n))}{(\mathbf{p}_n^T \mathbf{r}_n)^2} \,, \tag{10}$$

where χ_n is stopping parameter varying within $[0; 1]$, and if χ_n approaches to 1 then the convergence is successful [23].

Fuzzy inference systems (FIS) are powerful tools for the simulation of nonlinear behaviors utilizing fuzzy logic and linguistic fuzzy rules. In the literature, there are several inference techniques developed for fuzzy rule-based systems, such as Mamdani [24] and Sugeno [25]. Mamdani FIS is the first inference methodology, in which inputs and outputs are represented by fuzzy relational equations in canonical rule-based form. In Sugeno FIS, output of the fuzzy rule is characterized by a crisp function. Typical representation of a fuzzy rule in a Sugeno FIS is given by:

$$IF\, x\, is\, \underline{A}_1\, AND\, y\, is\, \underline{B}_1\, THEN\, z = f(x, y) \,, \tag{11}$$

where $\underline{A}$ and $\underline{B}$ are fuzzy sets and z is a crisp function. In Sugeno FIS, the outcome of each rule is a crisp value, and the result of all rules is calculated by weighted average. Mathematical definition of the nonlinear mapping of a Sugeno FIS (f_{FS}) can be written as follows:

$$f_{FS} = \frac{\sum_{i=1}^{m} w_i \prod_{j=1}^{n} \mu_{A_j^i}(x_j)}{\sum_{i=1}^{m} \prod_{j=1}^{n} \mu_{A_j^i}(x_j)} \,, \tag{12}$$

in which m is the number of rules, n defines the number of data points, and μ_A is the membership function of fuzzy set $\underline{A}$.

It is possible to simulate the nonlinear mapping, which is defined by known input-output data, using FIS methodology. Obviously, this is an unconstrained parameter identification problem based on the searching for optimal model parameters that can simulate target behavior. Jang [26] presented an adaptive network approach to solve this unconstrained optimization problem, namely the adaptive neuro-fuzzy inference system (ANFIS). Learning process in ANFIS methodology, namely adaptation of membership functions, is commonly performed by two techniques, i.e. backpropagation and hybrid learning algorithms. In hybrid learning algorithm, consequent parameters are identified in forward computation by LSE algorithm, and premise parameters are adjusted in backward computation using backpropagation algorithm.

In LSE methodology, the output of a linear model is expressed by:

$$y = \theta_1 f_1(u) + \theta_2 f_2(u) + \cdots + \theta_n f_n(u) + \epsilon \,, \quad (13)$$

where $u(u_1 \cdots u_n)$ is input vector, $f(f_1 \cdots f_n)$ are known functions, $y(y_1 \cdots y_m)$ is output vector, and $\theta(\theta_1 \cdots \theta_n)$ is unknown parameter vector. Using matrix notation, Eq. 13 can also be rewritten as:

$$\mathbf{A}\theta + \epsilon = Y \,, \quad (14)$$

where error is denoted by ϵ, and $\mathbf{A}$ is design matrix defined by:

$$A = \begin{bmatrix} f_1(u_1) & \cdots & f_n(u_1) \\ \vdots & \ddots & \vdots \\ f_1(u_m) & \cdots & f_n(u_m) \end{bmatrix} . \quad (15)$$

The objective is to find LSE (θ^*) that minimizes the sum of squared error, which is calculated by $||\mathbf{A}\theta - Y||^2$. In order to realize this, Eq. 13 is updated by:

$$\theta^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T Y + \epsilon \,. \quad (16)$$

Consequently, θ^* is obtained by an iterative procedure depending on following expression:

$$\theta_{n+1} = \theta_n + S_{n+1} a_{n+1} (y^T_{n+1} - a^T_{n+1}\theta_n) \,, \quad (17)$$

in which a^T_n is the n^{th} row vector of $\mathbf{A}$ matrix, y^T_n is the n^{th} element of Y vector, $\theta_0 = 0$, and parameter set (S_{n+1}) is calculated as follows:

$$S_{n+1} = S_n - \frac{S_n a_{n+1} a^T_{n+1} S_n}{1 + a^T_{n+1} S_n a_{n+1}} \,, \quad (18)$$

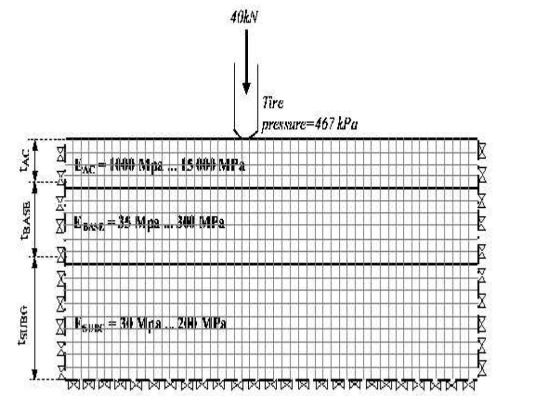

Figure 1. Illustration of FEM model for flexible pavement analysis

where S_0 is determined by identity matrix ($\mathbf{I}$), and a positive large number (γ) as given below:

$$S_0 = \gamma \mathbf{I} \quad (19)$$

4. Comparison of MLP and ANFIS Methodologies for Pavement Backcalculation Problem

From the modeling point of view, MLP and ANFIS methodologies can be utilized in the solution of this inverse problem based on the backcalculation of mechanical properties using measured surface deflection values. Basically, considered adaptive methodologies may be employed to learn the inverse mapping between known input (layer thicknesses, pavement moduli, and Poisson ratio) and output (surface deflections) patterns in a supervised manner. In this study, varying model parameters to evaluate the different backcalculation abilities of MLP and ANFIS methodologies were carried out.

Firstly, synthetic training and testing databases were generated by Finite Element Method (FEM). Then, MLP and ANFIS models were trained and tested using these synthetic databases. Then, small and poorly distributed synthetic databases were generated by the same manner, in order to investigate the role of number, scatting and uncertainty of training patterns in both methodologies. Besides, results from both methodologies were also compared with nonlinear least square based method, namely conventional backcalculation algorithm.

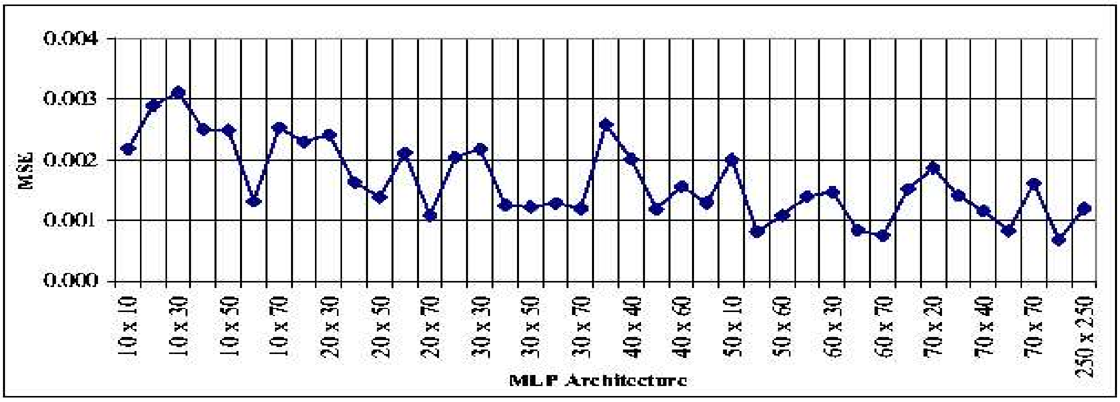

Figure 2. Effect of network architecture on MLP's performance

Table 1
Properties of synthetic training and testing model parameters.

Layer	Thickness m	Young's Modulus MPa	Poisson's Ratio
Surface	0.05-0.20	1000 - 15000	0.35 (fixed)
Base	0.15-0.50	35-300	0.40 (fixed)
Subgrade	15.00 (fixed)	30-200	0.45 (fixed)
FWD deflections(δ_0,δ_1,δ_2,δ_3,δ_4,δ_5,δ_6)			

4.1. Backcalculation with Sufficient Flexible Pavement Data

In this analysis, synthetic training and testing data sets, involving 1250 and 250 patterns are utilized, respectively. The data were generated by FEM. That is, inverse FEM inference is formulated in the first analysis using adaptive methods. The three-layered flexible pavement model is illustrated in Fig. 1. In order to prepare input data for FEM model, parameters were randomly selected from Gauss distribution in accordance with predetermined ranges given in Table 1. The reason of selecting Gauss distribution is the central limit theorem and the need for generating appropriate data within the predefined range. Furthermore, 80kN of Equivalent Single Axle Load (ESAL) for dual wheels is applied in the model. Pavement layers were assumed linear elastic, and computations were carried out by MATLAB 7.0. Details of FEM method and the software are beyond the scope of this study, and can be found elsewhere [27].

For MLP-based backcalculation session, training data were normalized to the range of $[-1, 1]$ and synaptic weights were selected randomly from normal distribution. This preprocessing technique is widely accepted method to increase the MLP's performance [8, 10]. It should be noted that, the scaling was performed in accordance with the hyperbolic tangential activation function as follows:

$$X_{new} = \frac{X - x_{min}}{x_{max} - x_{min}}\ , \qquad (20)$$

where X_{new} is normalized value, X is original value, and x_{max}, x_{min} are the maximum and the minimum values in the dataset, respectively. Error energy is measured by mean squared error (MSE) based formulation as given in Eq. 2. Scaled conjugate gradient learning algorithm was employed for 10000 epochs (network pass) throughout learning process. In order to determine optimal MLP architecture, a parametric study is carried out with trial-and-error basis. In Fig. 2, results for varying network architectures are shown, and 9x50x40x3 was decided to be the optimal structure.

There are two primary training parameters (λ and σ) to be determined at the beginning of scaled conjugate gradient learning process. The parameter "σ" determines the change in the weight for the second derivative approximation, and the parameter "λ" regulates the indefiniteness of the Hessian matrix. In this study, these parameters are selected by 0.00005 and 0.00007 for λ and σ, respectively [23, 28]. The training process

Table 2
Result of training and testing sessions.

Method	Session	Duration (hour)	Asphalt Coef. of det. R^2	Base Coef. of det. R^2	Subgrade Coef. of det. R^2
MLP	Training	0.92	0.97	0.92	0.96
(9x50x40x3)	Testing	real-time	0.91	0.88	0.92
ANFIS	Training	73	0.71	0.62	0.69
	Testing	9	0.64	0.58	0.60
MICHBACK	Training	0.27	0.94	0.90	0.91
	Testing	0.14	0.92	0.87	0.90

took between 3 min to 4 hours with a P4 3.0GHz and 1GB RAM PC.

On the other hand, for ANFIS-based backcalculation session, input parameters were partitioned using grid partitioning technique and Gaussian membership functions (Eq. 21).

$$f(x) = \mathrm{e}^{\frac{-(x-c)^2}{2\sigma^2}}, \tag{21}$$

in which c and σ are the functional parameters describing the shape of the curve. Input variables were fuzzified by dividing them into 3 partitions. Additionally, the first order Sugeno FIS with linear output function was selected as the inference system. ANFIS structure was completed by the selection of hybrid learning algorithm. Since grid partitioning was selected, $3^9 = 19683$ fuzzy rules were established for the inference system. Obviously, this size of rule-base is extremely large and computationally inefficient; so, it is required to reduce input space partitioning to limit the rule-base size. Nevertheless, it is not possible to decrease the partitioning to a smaller value than 3 for modeling precision. Actually, ANFIS methodology and fuzzy partitioning are not appropriate for a multivariate nonlinear approximation problem comprising 9 input variables. In the rule-base, fuzzy variables were connected with T-norm (fuzzy AND) operators and rules were associated using max-min decomposition technique. Furthermore, training continued for over 1000 epochs and process terminated by the observation of the stability in error decrement. As a result of the size of rule base and the number of training patterns, training process lasted 73 hours with the same PC mentioned previously. As a result, neither model precision nor computational requirements found appropriate for the solution of such a problem using ANFIS.

In order to asses the performance of conventional backcalculation techniques, MICHBACK computer program was also employed for same data. MICHBACK is a backcalculation software, using nonlinear least-square optimization technique, developed by University of Michigan Ann Arbor [19]. Comparative results of MLP, ANFIS, and MICHBACK based backcalculation analyses are given in Table 2. As can be seen from this table, MLP exhibited superior performance over other methods. Although MICHBACK is less precise than MLP, it produced satisfactory results. Apart from that, ANFIS was unsuccessful on considered backcalculation problem in terms of modeling ability and extremely high computational expense. In other words, 1250 training pattern and 9 input variables are not feasible for ANFIS based backcalculation process.

Also, same backcalculation models were employed for unseen testing data. Similar to training session, the architecture of MLP was chosen as 9x50x40x3 in testing session. Testing data was generated in the same way and consists of 250 different test patterns. The reason of testing was to investigate the performance of MLP on unrecognized data. Details of this session are also given in Table 2. Comparing testing and the training results, MLP exhibited poorer performance, yet showed good precision. MLP proved to be the most successful backcalculation methodology as well as faster real-time backcalculation capability where there is a large number of training data. It should be emphasized in assessing the performance of adaptive methods that, these methods are limited by the ranges and behavior of training patterns and do not guarantee meaningful outcomes beyond these ranges. Consequently, in order to illustrate the modeling precision for asphalt concrete, base, and sub-

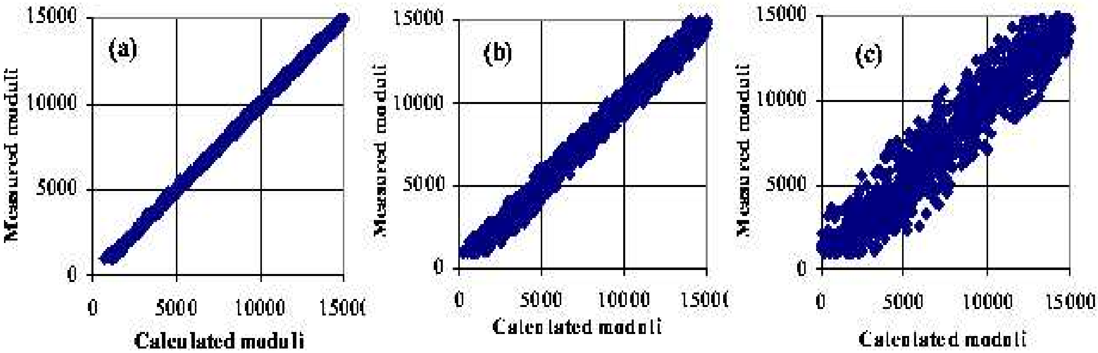

Figure 3. Scatter plots of AC layer for training data (a) MLP, (b) MICHBACK, (c) ANFIS

Table 3
Ranges of training and testing variable for the second analysis.

Layer	Thickness m	Young's Modulus MPa	Poisson's Ratio
Surface	0.10(fixed)	1000 - 15000	0.35 (fixed)
Base	0.30(fixed)	35-300	0.40 (fixed)
Subgrade	15.00(fixed)	30-200	0.45 (fixed)

FWD deflections (δ_0,δ_1,δ_2,δ_3,δ_4)

grade layers, scatting graphs between calculated and measured deflections for training and testing sessions are shown from Fig. 3 to Fig. 8, respectively.

4.2. Backcalculation of Incomplete Pavement Data

In the second phase of this study, another comparative analysis was carried out to evaluate the performances of considered methodologies, namely MLP, ANFIS, and conventional backcalculation (MICHBACK), on incomplete and/or small amount of flexible pavement data. It is likely when performing a backcalculation analysis that, there may not have large amount of data or the distribution of data may not be uniform. Furthermore, layer thicknesses can be the only output of the system, or it may be focused on one layer instead of whole structure. As a result, it may be misleading to prefer MLP over other methods (as implemented in the first analysis) for backcalculation under undesirable conditions. Especially, MLP is quite sensitive to the quality and the quantity of data, and network cannot produce meaningful outcomes for unrecognized inputs eventhough training session is successful [11].

In order to prove this, another hypothetical flexible pavement system was designed, and input-output data were limited. The ranges of model parameters for the second analysis are given in Table 3. As can be seen from Table 3, five surface deflections were selected as input variable and three pavement moduli were chosen as output parameter to develop pavement model. The numbers of training and testing patterns were also limited to 76 and 24, respectively. The scatting behavior of training data was not homogeneous, and there are unrecognized patterns in testing data. In ANFIS model, input variables were divided into 6 partitions; thus, there are $5^6 = 3125$ IF-THEN rules in the rule-base. Other attributes of MLP and ANFIS models were kept same with the first analysis.

In Table 4, results of training and testing sessions of second analysis are given. As can be seen from the table, durations of all training sessions were reduced with reference to the first analysis. Contrary to the first analysis, ANFIS model characterized the outcome better than MLP method. Especially for testing session, MLP based backcalculation exhibited unacceptable performance on unrecognized data patterns. Consequently,

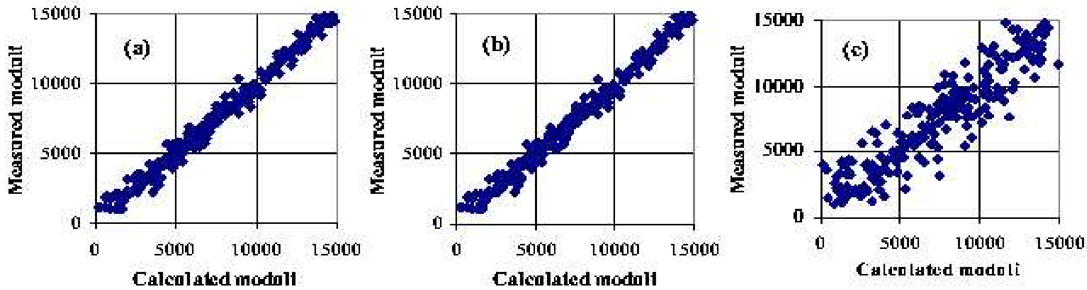

Figure 4. Scatter plots of AC layer for testing data (a) MLP, (b) MICHBACK, (c) ANFIS

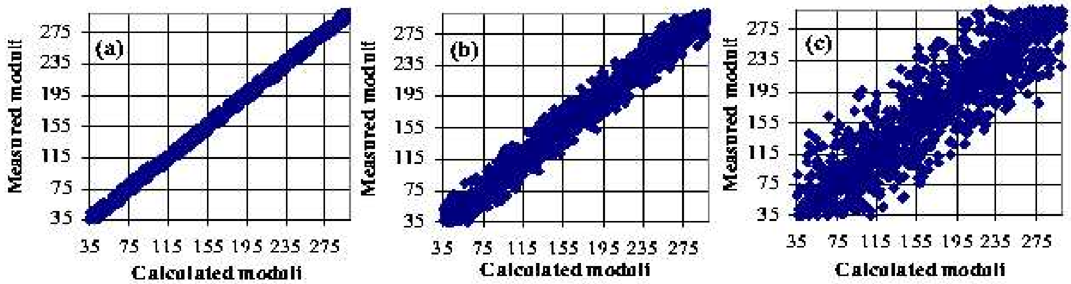

Figure 5. Scatter plots of base layer for training data (a) MLP, (b) MICHBACK, (c) ANFIS

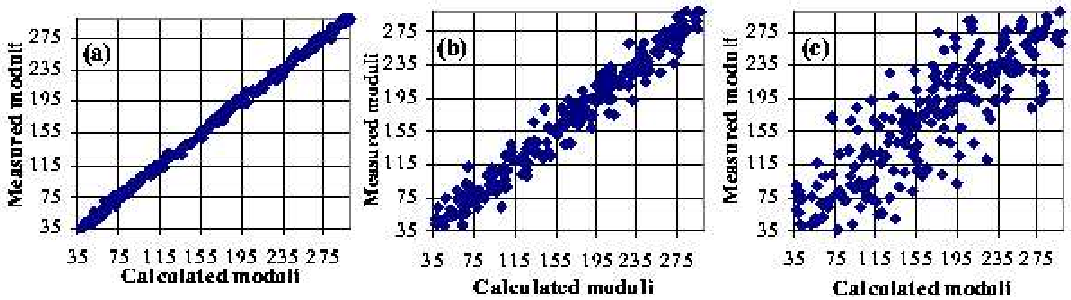

Figure 6. Scatter plots of base layer for testing data (a) MLP, (b) MICHBACK, (c) ANFIS

Table 4
Result of second training and testing sessions.

Method	Session	Duration (hour)	Asphalt Coef. of det. R^2	Base Coef. of det. R^2	Subgrade Coef. of det. R^2
MLP	Training	0.12	0.89	0.85	0.88
(9x50x40x3)	Testing	real-time	0.68	0.61	0.71
ANFIS	Training	0.68	0.90	0.88	0.90
	Testing	0.02	0.81	0.80	0.82
MICHBACK	Training	0.09	0.92	0.87	0.89
	Testing	0.04	0.91	0.85	0.87

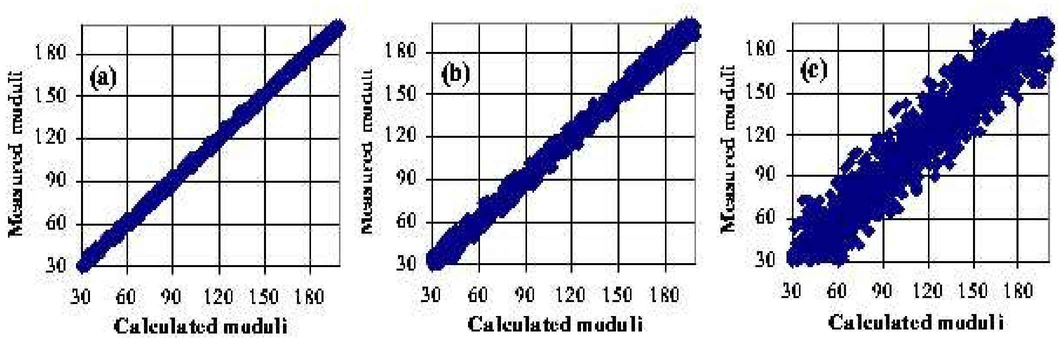

Figure 7. Scatter plots of subgrade for training data (a) MLP, (b) MICHBACK, (c) ANFIS

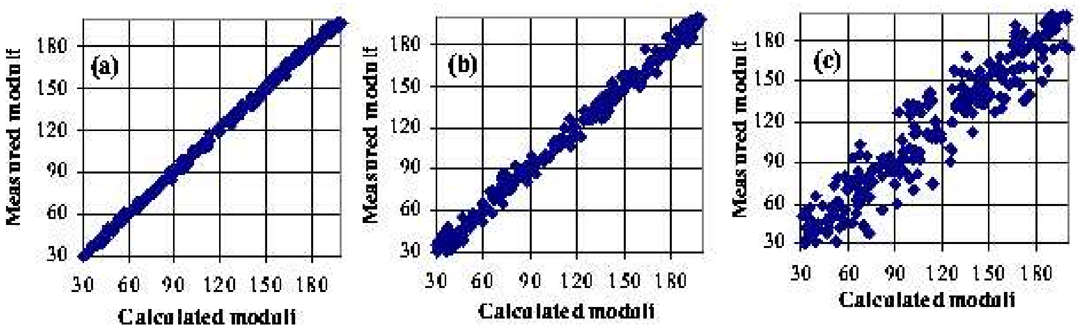

Figure 8. Scatter plots of subgrade for testing data (a) MLP, (b) MICHBACK, (c) ANFIS

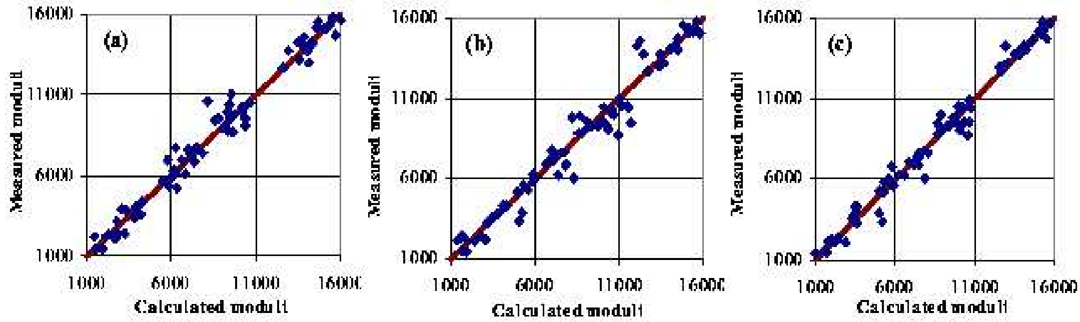

Figure 9. Scatter plots of AC layer for second training data (a) MLP, (b) MICHBACK, (c) ANFIS

ANFIS seems more successful over MLP if lack of data exists.

Scatter graphs between calculated and measured deflections were only plotted for AC layer. In Fig. 9 and Fig. 10, training and testing sessions' scatter plots are given, respectively. Scatter plots for base and subgrade are omitted in order to save space; yet, same conclusion can be drawn for these layers.

In summary, for the second analysis, MLP exhibited the poorest performance. Although training results are almost the same for all methods, outcomes of testing session are different. It should be noted that, MLP is unsuccessful for testing session, and failed for unrecognized patterns. ANFIS model exhibited quite good performance than MLP, but slightly poorer performance compared with MICHBACK. The reason of drastic performance increment and processing time decrement in ANFIS model is due to the increase in the input space partitioning, and significant reduction in the number of training patterns during the second analysis.

5. Conclusions

In this study, multi-layer perceptron (MLP) and adaptive neuro-fuzzy inference system (ANFIS) methodologies were employed for backcalculation of flexible pavements. Results of these adaptive techniques were also compared with a conventional backcalculation program (MICHBACK) using nonlinear least-square estimator. In order to investigate the effect of the quality and the quantity of model data, two different analyses were performed. This study of comparison of multilayer perceptron and adaptive neuro-fuzzy system on backcalculating the mechanical properties of flexible pavements lead to the following conclusions.

- Adaptive backcalculation methodologies, namely MLP and ANFIS, are fundamentally different from conventional backcalculation techniques and cannot be replaced completely since there is no physical principle, mechanical background, and material behavior utilized in these techniques. However, they give precise results comparing to classical methods. Furthermore, results are obtained faster; hence the methods are suitable for real time calculations.
- MLP is the best choice if sufficient amount of data exists to characterize the target behavior. Otherwise, ANFIS should be preferred due to its ability of fuzzy logic which manages uncertainty.
- ANFIS methodology can be employed for backcalculation problems involving considerable amount of uncertainty or having incomplete data.
- In ANFIS methodology, input space partitioning and the size of rule-base are crucial for computational expense. For this reason,

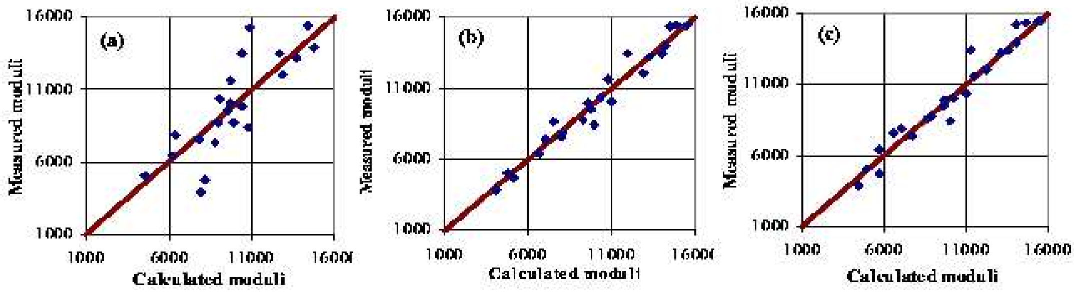

Figure 10. Scatter plots of AC layer for second testing data (a) MLP, (b) MICHBACK, (c) ANFIS

this method is appropriate for problems having relatively small number of input variables and/or involving small to medium number of training patterns.

References

[1] M. S. Hoffman and M. R. Thompson, Backcalculating nonlinear resilient moduli from deflection data, Transportation Research Record, 852, TRB, National Research Council, Washington, DC, 42 (1982).

[2] O. Tholen, J. Sharma, and R. L. Terrel, Comparison of FWD with other deflection testing devices, Transportation Research Record 1007, TRB, National Research Council, Washington, DC, 20 (1985).

[3] K. Tawfiq, J. Armaghani, and J. Sobanjo, Seismic pavement analyzer vs. FWD for pavement evaluation: comparative study, NDT of Pavements and Backcalculation of Moduli, 3, Special Technical Pub., STP 1375, 327-345, Eds. Tayabji, S. D. and Lukanen, E.O., ASTM Pub., PA. (2000).

[4] R. L. Lytton, Backcalculation of layer moduli, state of the art, NDT of Pavements and Backcalculation of Moduli, 1, Special Technical Pub., STP 1026, 7, Eds. Bush, A. J. and Baladi, G. Y., ASTM Pub., PA. (1989).

[5] J. Uzan, Advanced backcalculation techniques, NDT of Pavements and Backcalculation of Moduli, 2, Special Tech. Pub, STP 1198, 3, Eds. Von Quintus, H. L., Bush, A. J. and Baladi, G. Y., ASTM Pub, PA. (1994).

[6] A. C. Callop and D. Cebon, ASCE Journal of Transportation Engineering **122**, 131 (1996).

[7] P. Ullidtz, Will nonlinear backcalculation help, NDT of Pavements and Backcalculation of Moduli, 3, Special Technical Pub., STP 1375, 14, Eds. Tayabji, S.D. and Lukanen, E.O., ASTM Pub., PA. (2000).

[8] S. Haykin, Neural Networks: A Comprehensive Foundation (Prentice-Hall Inc., New Jersey, 1999).

[9] R. W. Meier and G. J. Rix, Backcalculation of flexible pavement moduli using artificial neural networks, Transportation Research Record, 1448, TRB, National Research Council, Washington, DC, 75 (1994).

[10] R. W. Meier and G. J. Rix, Backcalculation of flexible pavement moduli from dynamic deflection basins using artificial neural networks, Transportation Research Record, 1473, TRB, National Research Council, Washington, DC, 72 (1995).

[11] E. Tutumluer and R. W. Meier, Attempt at resilient modulus modeling using artificial neural networks, Transportation Research Record, 1560, TRB, Washington, DC, 1 (1996).

[12] R. W. Meier and G. J. Rix, ASTM Geotechnical Testing Journal **16**, 425 (1993).

[13] E. Tutumluer and U. Seyhan, Neural network modeling of anisotropic aggregate behavior from repeated load triaxial tests, Transportation Research Record, 1615, National Research Council, Washington, DC, 86 (1998).

[14] Y. R. Kim, B. Xu, and Y. Kim, A new backcalculation procedure based on dispersion analysis of FWD time history deflections and surface wave measurements using ANNs, NDT of

Pavements and Backcalculation of Moduli, 3, STP 1375, 297, Eds. Tayabji, S.D. and Lukanen, E.O., ASTM Pub., PA. (2000).

[15] S. Terzi, M. Saltan, and T. Yildirim, Optimization of deflection basin by genetic algorithm and neural network approach. Int. Conf. on Neural Networks, ICANN/ICONIP 2003, LNCS 2714, 662 (2003).

[16] A. B. Goktepe, E. Agar, and A. H. Lav, Backcalculation of mechanical properties of flexible pavements using neural networks, ITU Engineering Journal D, in press (2004).

[17] B. Sebaaly, T. G. Davies, and M. S. Mamlouk, ASCE Journal of Transportation Engineering **111**, 618 (1985).

[18] D. F. E. Stolle, Computers and Geotechnics **11**, 83 (1991).

[19] R. S. Harichandran, M. S. Yeh, and G. Y. Baladi, MICH-PAVE: A nonlinear finite element program for analysis of flexible pavements, Transportation Research Record, 1286, TRB, National Research Council, Washington, DC, 123 (1990).

[20] T. F. Fwa, C. Y. Tan, and W. T. Chan, Backcalculation analysis of pavement-layer moduli using genetic algorithms, Transportation Res. Record 1570, TRB, National Res. Council, Washington, DC, 134 (1997).

[21] M. A. Reddy, K. S. Reddy, and B. B. Pandey, Journal of Highway Research Board **66**, 1 (2002).

[22] D. E. Rumelhart and J. L. McClelland, Parallel distributed processing: explorations in the microstructure of cognition (MIT Press, Cambridge, Massachusetts, 1986).

[23] M. F. Moller, Neural Networks **6**, 525 1993.

[24] E. H. Mamdani and S. Assilian, Int. Journal of Man-Machine Studies **7**, 1 (1975).

[25] T. Takagi and M. Sugeno, IEEE Trans. Sys. Man and Cybernetics. **15**, 116 (1985).

[26] J. S. R. Jang, IEEE Trans. Systems Man and Cybernetics. **23**, 665 (1993).

[27] K-J. Bathe, Finite Element Procedures in Engineering Analysis (Prentice Hall Inc., New Jersey, 1982).

[28] M. T. Hagan, H. Demuth, and M. Beale, Neural Network Design (PWS Publishing Co., New York, 1996).