Acta Sci. Pol.
Architectura 24 (2025), 123–141
aspa.sggw.edu.pl ISSN 1644-0633 (suspended) eISSN 2544-1760 DOI: 10.22630/ASPA.2025.24.10

ORIGINAL PAPER

Received: 02.04.2025
Accepted: 25.04.2025

JOINT APPLICATION OF THE FINITE ELEMENT METHOD AND ARTIFICIAL NEURAL NETWORK IN THE IDENTIFICATION OF PARAMETERS OF LAYERED PAVEMENT

Artur Sławomir Góral[1]✉, Marek Wojciechowski[2]

[1] Institute of Civil Engineering, Warsaw University of Life Sciences – SGGW, Warsaw, Poland
[2] Department of Concrete Structures, Division of Geotechnics and Engineering Structures, Lodz University of Technology, Łódź, Poland

ABSTRACT

In this paper, an inverse analysis of the mechanical parameters of existing layered pavements was performed. An artificial neural network (ANN) was used to approximate the response of the numerical pavement model to the input parameters. Two methods were presented. In the first method, the ANN directly maps the inverse relationship between the pavement surface deflection and mechanical parameters. In the second method, the approximated model was used in the classical back-calculation of pavements by a procedure of minimising the differences between the model response (now a neural network) and field measurements obtained from the falling weight deflectometer (FWD) test. Only one parameter of each pavement layer, the Young's modulus, was identified. It was found that the identification is not unambiguous. This means a given pavement deflection can be observed with different sets of layer stiffness moduli. However, the average stiffness of the layers is always identified with high accuracy.

Keywords: artificial neural network, inverse problems, numerical model, parameter identification, falling weight deflectometer, field measurement

INTRODUCTION

The road pavement structure is made up of a set of layers whose task is to transfer the loads from the wheels of vehicles to the ground, while ensuring an adequate level of convenience and traffic safety. The most important performance characteristic in the Polish evaluation system is the bearing capacity of the pavement, i.e. it is the ability to bear loads from traffic. Assessment of the mechanical properties of the existing multi-layer pavements based on the non-destructive load test of the top surface is currently the most widely used method. They are determined on the basis of measured, by static or dynamic methods, vertical displacements (deflections) of the pavement structure under a known load. A standardised device called a falling weight deflectometer (FWD) is used to perform such a test. Floor deflections are then caused by the impulse load from the falling weight. They are then measured at several points using geophones (a maximum of nine geophones) mounted on a rigid support, as part of a special-purpose vehicle (Figs 1 and 2). This process has an impact character and, as a result, the recorded pavement displacement represents the dynamic response to the load. On the other hand, in design practice, static pavement models in the form of layered elastic half-spaces are

Artur Sławomir Góral https://orcid.org/0000-0002-6862-1394; Marek Wojciechowski https://orcid.org/0000-0003-3766-2801
✉artur_goral@sggw.edu.pl

most often used. In current practice, researchers use methods of transforming the displacements obtained from the dynamic test into their static substitutes (Ruta, Krawczyk & Szydło, 2015). This paper emphasises the evaluation of the use of ANNs to identify pavement parameters. The FWD test was the inspiration for the conducted numerical experiments. In this paper, theoretically calculated (for the assigned known elasticity moduli) displacements were used as 'experimental' pseudostatic data from the FWD test.

Fig. 1. Dynatest® – falling weight deflectometer

Source: own work.

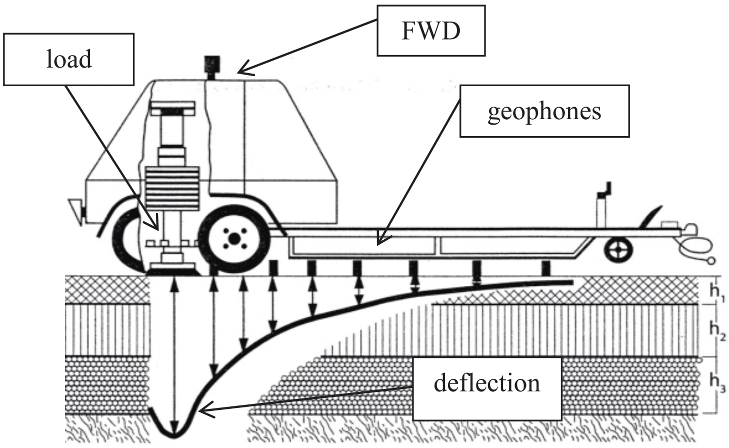

Fig. 2. Deflection bowl from FWD test

Source: Wesołowski and Blacha (2014).

A set of measured deflections is usually named the deflection bowl. These bowls are used in back analysis to identify parameters of computational models of road pavements. This process is called back calculation. The reaction of the pavement to the applied load, i.e. the measured deflections, determines the values of material properties that should be assigned to specified layers separately in the theoretical model of the pavement. As mentioned above, the theoretical model for the FWD test is usually an elastic layered half-space, statically loaded over a small area at the point of the falling mass (Yoder & Witczak, 1975; Ruta et al., 2015). In the case of rigid pavements, the theoretical model may also be a layered plate supported on an elastic subgrade, e.g. the Winkler–Pasternak model (Boso & Lefik, 2012). Another method for the theoretical model of the pavement is a numerical formulation using, for example, the finite element method (FEM). The latter approach gives the greatest freedom of use in selecting the model parameters and has been used for the purposes of this article.

Classic backward calculations leading to the determination of the mechanical parameters of pavement layers consist in minimising the sum-of-squares differences between the measured and theoretical pavement deflections. Which can be written as follows:

$$\min_{p_i}\left[\sum_{k=1}^{m}\left(u_k\left(p_i\right)-d_k\right)^2\right], \tag{1}$$

where:
p_i – assumed parameters of the theoretical model,
i – layer index,
m – number of reading points,
$u_k(p_i)$ – calculated theoretical deflection at the k-th surface point,
d_k – measured deflection at the k-th surface point.

Any variables that have an influence on the solution can be model parameters, such as: elastic modules (Young's moduli) of layers E_j, Poisson's ratios v_j, layer thicknesses h_j, interface conditions between layers $k_{s,j}$ (j – layer index), and others. With the elastic formulation of the problem, assuming the continuity of the material between the layers and the assumed thickness of the pavement layers, only the moduli E_j have a significant influence on the deformation of the pavement system. Then, problem (1) can be written as follows:

$$\min_{E_j}\left[\sum_{k=1}^{m}\left(u_k\left(E_j\right)-d_k\right)^2\right]. \tag{2}$$

The optimisation problems mentioned above require frequent model evaluation when searching for the minimum of the objective function (calculation of deflections u_k). Depending on the complexity of the mechanical model, it can be a time-consuming process. Artificial neural networks (ANN) can be used as an input-output algorithm where the first set of data is the mechanical parameters of the pavement structure, and a second set of data is the mechanical response of the pavement under a known load, including measured deflections at given points on the pavement. When the first set is used as input data and the second set as output data, an ANN is being used as a numerical approximator that learns to simulate and solve the engineering problem. When the input data is the second dataset, and the output is the first set of data, the ANN is being used as a universal approximator of the inverse problem.

In this research, two alternative methods of searching for the optimal parameters of the FEM numerical model of the layered pavement will be presented and tested for the problem (2). Feed-forward artificial neural networks were used for both algorithms (Wojciechowski, 2011; Boso & Lefik, 2012; Lefik & Boso, 2016). This approach allows for a massive speed boost in solving the parameter identification problem and leads to better recognition of the inverse problem under consideration. ANNs have been previously used to solve the inverse problem of pavement mechanical properties identification (Meier, 1995; Bredenhann & Ven, 2004; Saltan & Terzi, 2005; Shahnazari, Tutunchian, Mashayekhi & Amini, 2012; Elbagalati, Elseifi, Gaspard & Zhang, 2017; Li & Wang, 2017; Li & Wang, 2018; Aubdulnibe & Jassim, 2019). In the literature on the subject, mostly examination of the first method is observed. The application and comparison of the two alternative approaches using the same ANN training and testing data is considered a main contribution of this paper.

ARTIFICIAL NEURAL NETWORKS

Artificial neural networks were based on the anatomical, physiological and biochemical knowledge of the human brain that was available at the time. However, the developers of neural networks treated this knowledge as a source of inspiration, not as a model to be copied exactly. Therefore, the construction and operating principles of artificial neural networks used in practice are not a faithful reflection of the state of biological knowledge, even that somewhat outdated knowledge from a dozen years ago. The first widely known example of a built and interestingly functioning neuro-like network is the 'Perceptron' (Minsky & Papert, 1969). The performance of the perceptron was not satisfactory from the point of view of the fundamental purpose of the test: although the system learned the task of recognition, it could not cope with more complex signs, and it showed sensitivity to changes in the scale of the learned objects. Despite the diverse architectures used by artificial neural networks, the essence of their operation is still based on the concept of the Perceptron algorithm, composed of a single artificial neuron.

The artificial neuron building blocks used in networks architecture are, of course, highly simplified models of nerve cells found in nature. The construction of an artificial neuron is best illustrated by the diagram shown in Figure 3.

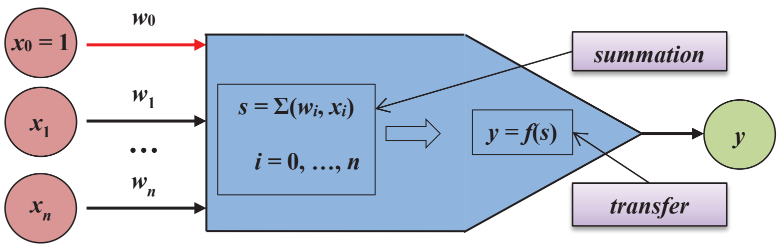

Fig. 3. Overall structure of a simplified neuron: basic signals, weighting coefficients, aggregation of data in the first internal function of the neuron, additional parameter – bias, transfer function

Source: own work.

Artificial neurons are characterised by the presence of multiple inputs and a single output. The input signals x_i $(i = 0, 1, 2, ..., n)$ and the output signal y can take only numerical values, most often in the range from 0 to 1, while the fact that they correspond to certain unit of information in the tasks solved by the network is the result of a special procedure. Artificial neurons perform certain actions on the signals they receive at the inputs, as a result of which they produce signals (only one for each individual neuron), which are present at their outputs and are transmitted further (either to other neurons or to the output of the network, as a solution to the problem presented). Thus, the task of the network, reduced to the functioning of its basic element, which is the neuron, is that it transforms input information x_i into output y using rules derived from how it is built and what it has been taught. With each separate input of the neuron is an associated parameter called weight w_i. This name means that it expresses the degree of importance of the information reaching the neuron with this particular input. These weights reflect the rather complex biochemical and bioelectrical processes occurring in the real synapses of biological neurons. The most important thing is that synaptic weights can be modified (i.e. have their values changed), which is the basis of network learning. The signal coming in with a particular input is first modified using the weight of the input. The most common modification is that the signal is simply multiplied by the weight of the given input, so that it already participates in further calculations in a modified form. Sometimes (not in all types of networks, but generally often), some additional component independent of the input signals is added, called a bias in the literature. Bias is also subject to a learning process. The bias signal is of a constant value of 1. The role of bias is that, thanks to its presence, the properties of the neuron can be shaped much more freely during the learning process. The addition of a bias value allows you to change the 0 point of the function. The input signals are aggregated inside the neuron, which most often means that the corresponding signals are simply summed. The result of this process gives some auxiliary internal signal called a net value. Such a sum can sometimes be directly transferred to the output and treated as the neuron's output signal. In more sophisticated networks, on the other hand, the output signal of a neuron is calculated using some nonlinear function, which is called a transfer function and usually is defined as $f(s)$ or $\varphi(s)$. The transfer function is commonly called the characteristics of the neuron. Various characteristics of the used activation functions are known: Sigmoid, threshold, linear, piecewise linear, threshold linear, Gaussian, tanh, and rectifying linear unit. The behaviour of the artificial neuron maximally resembles the behaviour of a real biological neuron (sigmoidal function), but they can be selected in such a way as to ensure maximum efficiency of the calculations carried out by the neural network.

So far, a single artificial neuron has been described. A neural algorithm is a network of artificial neurons organised in interconnected layers called an ANN. Such networks are built to eliminate the major drawback of the single neuron 'Perceptron' algorithm – the ability to classify only linear data. An example of a classical ANN is presented in Figure 4. At the beginning of the network is the input layer (LI), which, after receiving input signals x^0_i ($i = 0, 1, 2, ..., n$), generally does not process them but distributes them – which means that it distributes them to all the neurons x^j_i ($i = 0, 1, 2, ..., n$; $j = 0, 1, 2, ..., k$) of the hidden layers [with deep learning, the internal hidden layers (L1–LK) are observed]. Hidden layer neurons use the properties of a single artificial neuron described above. Thus, they process the input signal into an output signal y_i ($i = 0, 1, 2, ..., n$) and transmit it to the output layer (LO). In the classical ANN architecture, each neuron of the selected layer is connected to all neurons of the previous and next layers. This network architecture is often described as a feedforward structure. Feedforward networks are structures in which there is a well-defined direction of signal flow, from input to output on which the final result is given. Only during network training is the data flow reversed. Such networks are the most widely used and useful (Tadeusiewicz, 1993; Lefik & Schrefler, 2001; Shahin, Jaksa & Maier, 2001; Wojciechowski, Lefik, 2004; Boso, Lefik & Schrefler, 2012; Momeni, Nazir, Jahed Armaghani & Maizir, 2015; Alwattar & Mian, 2019; Vahab, Shahbodagh, Haghighat & Khalili, 2023). This type of network will also be used in this paper.

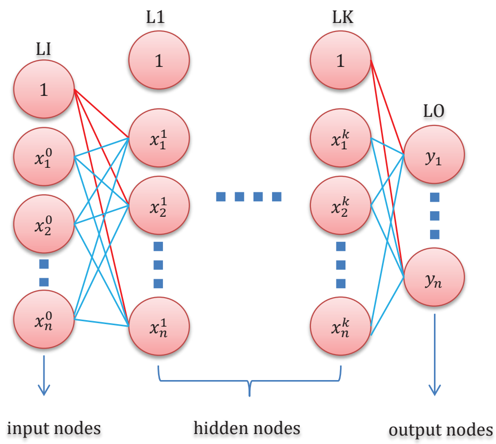

Fig. 4. Overall classical ANN architecture

Source: own work.

Training of the network is required so that it can correctly perform the operations intended for it. Training the ANN consists in finding the optimal values of the weights, so that the inputs belonging to the training data are processed and transmitted through the layers, resulting in outputs that fit well-known outputs (the ground truth) (Galbusera, Casaroli & Bassani, 2019). For training in the case of supervised learning, a set of labelled numerical data is used, divided into training, validation and testing subsets. The literature offers several proposals for dividing the dataset into the aforementioned subsets in different proportions. It is important that one data point can belong to one of the subsets. The training subset is used strictly to calculate weights with the use of backpropagation algorithms. The validation subset is used to adjust the net hyperparameters (e.g. learning rates, momentum). The test subset is used after the training process is completed, not for training the weights or adjusting the hyperparameters, but for assessing the model's accuracy and efficiency.

For many years, no effective method for training multilayer networks was known, which is why the layers for which the error signal could not be determined were known in the literature as 'hidden layers'. For this problem, the backpropagation algorithm turned out to be revolutionary. It consists of the fact that for the determined error $\delta^{(m)(j)}$ occurring during the j-th step of the learning process in an m neuron, we can project this error backwards to all neurons whose signal were inputs to the m neuron. The number of comparisons carried out depends on the size of the training set, the assumed number of iterations, and the training process of the network. This is because there are no specific guidelines to determine the weight parameters without testing the algorithm each time for a given problem.

NUMERICAL MODEL OF PAVEMENT

Formulation

The classical theory of an elastic half-space of a domain, e.g. a soil medium, describes the half-space as a homogeneous, infinite structure, limited on one side by a plane – the terrain surface. The layered half-space is a structure composed of many layers with different mechanical properties and separated by planes parallel to the surface of the terrain. For a pavement structure, the typical direction is horizontal. Typical examples of layered structures are the pavements of airports, roads, highways, and parking lots, supported on a subgrade. The general model of the pavement supported on the ground is shown in Figure 5. Each layer has an infinite size in the horizontal direction and is defined by a finite thickness h_j, Young's modulus E_j and Poisson's ratio v_j, where $j = 1, ..., n$ defines the number of the next layer. The last layer is described by the modulus of elasticity E_n and Poisson's ratio v_n and it has an infinite size in both the horizontal and vertical directions ($h_n \to \infty$).

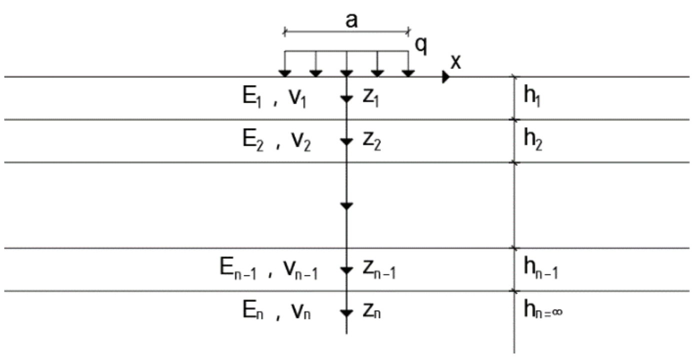

Fig. 5. Illustration of a flexible layered half-space model

Source: own work.

Analytical solutions for a homogeneous and layered half-space are known and described in the literature on the subject for both cases of static and dynamic load (Boussinesq, 1885; Burmister, 1945; Lamb, 1997; Lin, Zhang, Liu & Wang, 2016). However, from the applied side, the analytical models are very simplified. They do not make it possible to take into account physical factors that can significantly affect the measured values of displacements, e.g. imperfect layer adhesion, material anisotropy, temperature gradient, non-linearity in the behaviour of the subgrade.

In this article, the pavement model was described numerically using the finite element method. It allows for advanced deformation analysis. However, the main aim of the study is to present and evaluate methods of identifying model parameters with the use of artificial neural networks. The aim is not to precisely reproduce the structure of the layered pavement, so the FEM model was limited to an elastic isotropic domain in a plane strain state. It was also assumed that only Young's modulus would be identified due to the significant impact on deformation. Other important mechanical parameters, e.g. Poisson's ratios and thicknesses for all layers, will be known, e.g. based on structural design.

Figure 6 shows a diagram of the considered model. The model's layers correspond to the structure of the actual airport pavement (Nita, 2008). Therefore, the model consists of three structural concrete layers from the top, each with different stiffnesses. The fourth layer is natural soil. The extract of the data for the model is summarised in Table 1.

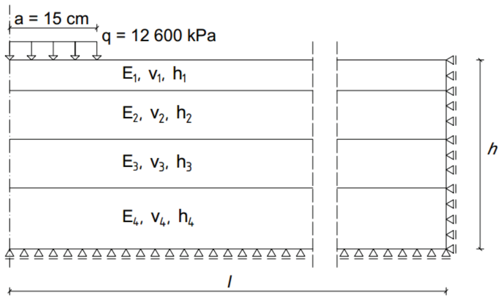

Fig. 6. Diagram of the considered layered pavement

Source: own work.

Table 1. Summary of data for the applied model of the layered pavement

Layer	h_j [cm]	v_j [-]	E_j [GPa]
L1	20	0.2	15.0–45.0
L2	20	0.2	12.5–37.5
L3	15	0.2	7.5–22.5
L4	145	0.3	0.025–0.075

Source: own work.

Due to the symmetry of the pavement model, only half of the geometry model was considered. The dimensions of the analysed area are $l \times h = 5 \times 2$ m. A static constant load was assumed on the top surface with a width of 30 cm and a value of 12,600 kPa. The value and range of the load correspond to the static vertical contact stresses apparent under the wheel of the Boeing 747-300 airplane (Wikipedia, 2023). As boundary condition horizontal displacements are blocked on the axis of symmetry and on the right edge of the model, a full displacement lock is used at the bottom of the model.

Figure 7 shows the finite element mesh adopted for the analysed boundary problem. Six-node triangular elements were used. The edge size of an equilateral triangle is 5 cm at the top surface of the model, evenly increasing to 15 cm at the bottom surface. The FEM model and the calculations described below were performed using the *fempy* software (Wojciechowski, 2018).

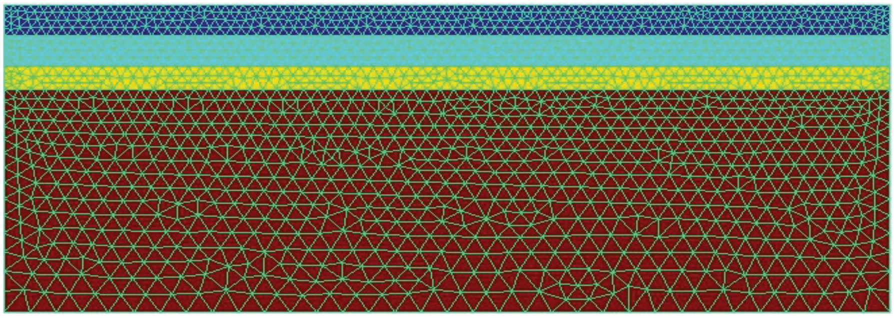

Fig. 7. Finite element mesh of the model

Source: own work.

Generation of data for ANN training

For the model defined above, 100 sets of deformation modules E_j for the four layers were randomly generated. It was assumed that all other parameters of the model are invariant. The sampling was carried out using the Latin hypercube sampling (LHS) method, which is a statistical method of generating samples with a multivariate distribution (Mckay, Beckman & Conover, 2000). It ensures that the set of samples is representative of the given ranges of parameter variability (while the usual random sampling is just a collection of random numbers with no guarantees whatsoever). Then, calculations were performed for all the generated samples, registering u_k deflections of the top surface at nine points 0, 30, 60, 90, 120, 150, 180, 210 and 240 cm from the axis of symmetry. In this way, data was obtained in sequence ($E_j^{(p)}$, $u_k^{(p)}$), where $E_j^{(p)}$ – elasticity modules of the layers, $u_k^{(p)}$ – deflections of the top surface, $p = 1, ..., 100$; $j = 1, ..., 4$; $k = 1, ..., 9$. Selected calculation results are presented in Figures 8–10.

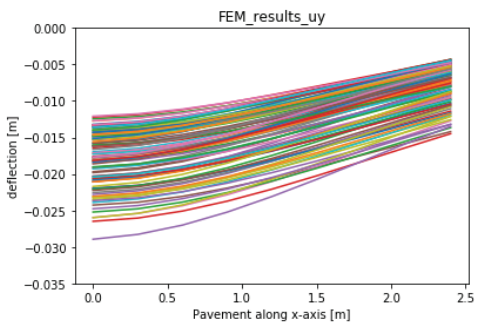

Fig. 8. Deflections of the pavement recorded in reading points for 100 random sets of E_j modules of the layers

Source: own work.

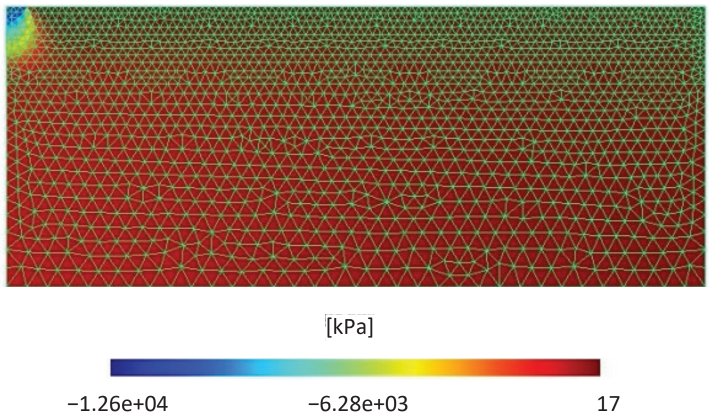

Fig. 9. Vertical stresses for the selected set of modules

Source: own work.

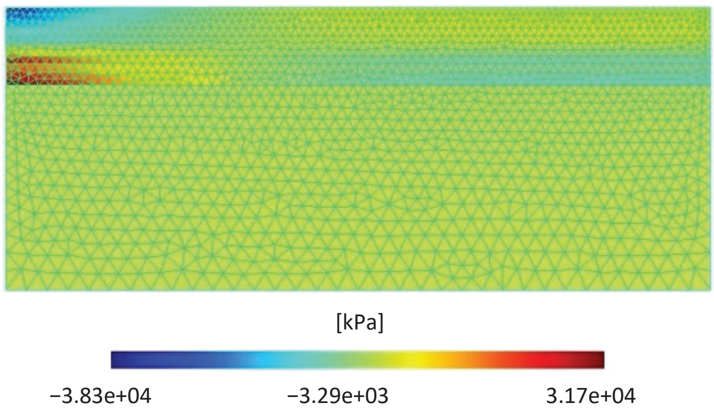

Fig. 10. Horizontal stresses for the selected set of modules. In the area of the concrete pavement, the stress distribution characteristic for the slab resting on the elastic subgrade is visible

Source: own work.

IDENTIFICATION OF MODEL PARAMETERS USING ANN

Training of artificial neural network

The fast feed-forward artificial neural network is a universal approximator of multivalued functions of several variables (Burczyński, Beluch, Długosz, Orantek & Nowakowski, 2000; Burczynski, Beluch, Długosz, Orantek & Skrobol, 2006). If the input and output of the network is interpreted as an argument and image of a certain operator or function that the ANN approximates, respectively, the consistency of the output values with the expected values should be satisfied for an infinite number of all possible sets of arguments. To achieve this, the network parameters (weights) are selected by procedure of successive corrections in an iterative process called training or teaching the network. The training of the network minimises the difference between the ANN output signal and the target signal expected at the ANN output. A popular technique that can be found in the literature is the minimisation of the sum of squared errors calculated for all output neurons and for all input patterns, and such a formulation was used in this article. The ANN weight optimisation procedure gives the desired effect when there is a clear correlation between the input and output data. At the same time, the ANN must be designed with an appropriate number of hidden neurons, adjusted to the complexity of the objective function being approximated and to the size of the training set. Having data from the FEM model evaluation for random parameter sets, i.e. having ($E_j^{(p)}$, $u_k^{(p)}$) pairs, it is possible to train a feed-forward artificial neural network both in order to approximate the inverse relations (deflections – modules) and direct relations (modules – deflections). In this article, the *ffnet* software (Wojciechowski, 2016) was used for this purpose, which enables ANN training with the use of advanced optimisation tools, with automatic selection of the number of epochs, learning parameters and criteria for the completion of the weight selection process. In this paper, sigmoidal functions were used in the neural network models, both in the hidden layers and in the output layer. Despite the possibility of using other activation functions, this function was deliberately chosen. The *ffnet* program makes it possible to use higher-order derivatives for calculations, which is guaranteed by the sigmoid function. For example, the ReLU function does not have a second-order derivative. For all examples in the paper, the root mean squared error (RMSE) cost function was used. The cost function characterises the effectiveness of the model on the training dataset. Loss functions express the discrepancy between the predictions of the trained model and the actual occurrences of the problem. If the deviation between the predicted and actual results is too large, then the loss function will have a very high value. To accelerate the training of the network, supporting algorithms – optimisation functions – are also used. Optimisers are algorithms or methods used to change neural network attributes, such as weights and learning speed, to reduce losses. Gradient descent is the simplest but most commonly used optimisation algorithm. In this paper, the gradient descent optimiser was used.

ANN as direct approximator of the inverse relation

Method description

In this approach, an artificial neural network is defined and trained, directly mapping the inverse relationship between surface deflections and stiffness modules, which can be written as follows:

$$\tilde{E}_j = N_j^{-1}\left(u_k, \vec{w}\right), \tag{3}$$

where:

$\tilde{E}_j$ – network output, i.e. approximated elastic modules,

N_j^{-1} – operator performed by ANN,

u_k – network input in the form of pavement deflections,

$\vec{w}$ – network weights.

The trained ANN, therefore, has nine input neurons and four output neurons. The network architecture also consists of 10 hidden neurons in one layer. Network training was carried out on 95 sets of training data (training and validation sets), and the remaining five were used for network testing (test set). An example of the input and output data are presented in Table 2. The training therefore consisted in solving the following minimisation problem:

$$\min_{\vec{w}}\left[\sum_{p=1}^{95}\sum_{j=1}^{4}\left(N_j^{-1}\left(u_k^{(p)}, \vec{w}\right) - E_j^{(p)}\right)^2\right]. \tag{4}$$

Table 2. Example of inputs and outputs data for trained 9-10-4 ANN

Dataset	Input [cm]									Output [GPa]			
	D0	D30	D60	D90	D120	D150	D180	D210	D240	E_1	E_2	E_3	E_4
Training	2.15	2.12	2.03	1.92	1.78	1.63	1.47	1.30	1.14	33.75	29.48	19.10	0.03
	1.42	1.3	1.29	1.18	1.05	0.91	0.76	0.62	0.49	19.98	31.91	15.32	0.07
Testing	1.37	1.32	1.23	1.12	0.99	0.85	0.71	0.57	0.44	16.75	26.86	17.28	0.07

Note: D0, D30, etc. denotes the deflection at the geophone at 0 mm, 30 mm, etc. from the load centreline.

Source: own work.

After training the network, the identified values of the layer modules are obtained directly from the relation (3) by entering the values of the measured (test) deflection d_k on the surface. Thus, the module-matching procedure described by the formula (2) is not performed. The advantage of this approach is the very fast parameter identification process. However, their optimality, depending only on the quality of the learning process (4), is not guaranteed. Usually, however, it is possible to obtain parameters close to optimal, with an accuracy sufficient for engineering applications.

Learning and identification results

The network with the 9-10-4 architecture was trained as described above for 95 training samples. Figures 11 and 12 show the convergence results of the pattern modules and modules determined by the trained ANN for all 100 samples. The best fit is observed for the last layer – the subgrade (bottom right graph in Figure 11). Slightly poorer results are obtained for the remaining layers. It is worth noting that the fit of the test results does not deviate from the fit of the training data, which indicates that the ANN has been trained properly.

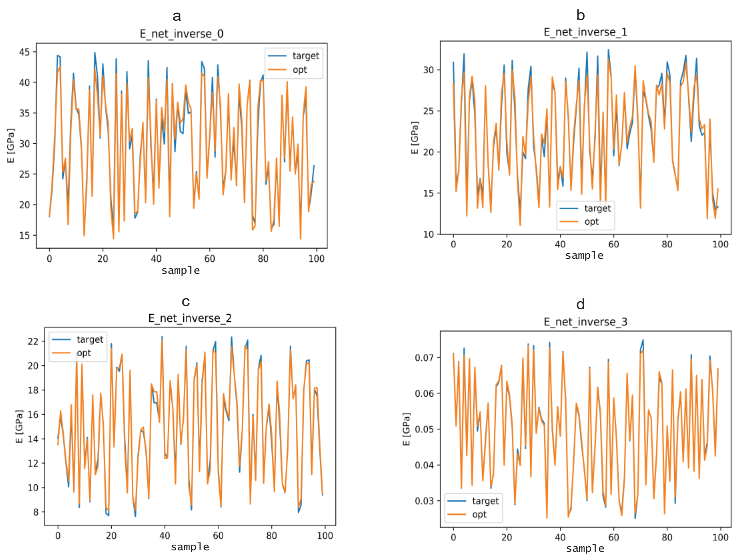

Fig. 11. Results of training the 'inverse' network N_j^{-1}: (a) 1st layer, (b) 2nd layer, (c) 3rd layer, (d) 4th layer: target – reference values of modules, opt – module values identified by the trained ANN. The last five points of the graphs refer to the testing data

Source: own work.

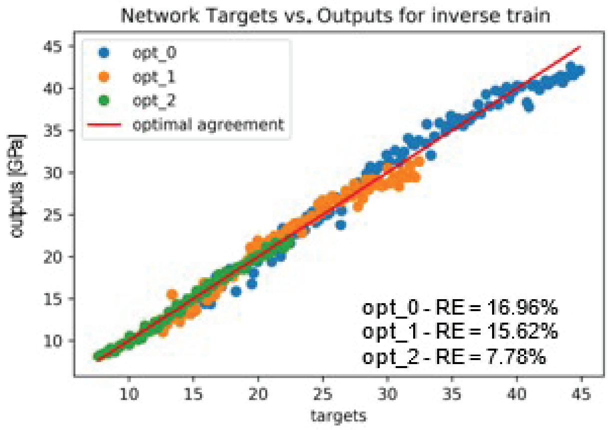

Fig. 12. Response of the 'inverse' network versus reference of the modules: E_1 (opt_0), E_2 (opt_1) and E_3 (opt_2). The points on the graph also contain test data

Source: own work.

Table 3 summarises the numerical identification results for the test samples. The average values of the modules for the model layers are also given. The differences in the computed and pattern modules do not exceed 15%, and the differences in the mean values do not exceed 5%. This indicates the ability of the network to correctly reproduce the averaged, effective properties of the layered pavement, even if the modules of the individual layers are determined with significant errors.

Table 3. Numerical compilation of the reference (target) and modules of successive layers of the model determined by the ANN (opt) for five test samples

Test	Young's moduli [GPa]	E_1	E_2	E_3	E_4	Arithmetic mean	Harmonic mean	Weighted arithmetic mean	Weighted harmonic mean
1	opt	23.75889	15.49612	9.50824	0.06691	12.20754	0.26389	4.68713	0.09213
	target	26.37433	13.31490	9.39255	0.06586	12.28691	0.25970	4.72112	0.09069
2	opt	23.32254	11.91731	13.42858	0.04248	12.17773	0.16849	4.56193	0.05854
	target	21.91036	12.93881	13.07886	0.04399	11.99300	0.17443	4.49772	0.06061
3	opt	18.87684	14.21434	18.17562	0.06081	12.83190	0.24065	4.71638	0.08377
	target	19.01810	15.04311	17.49291	0.06060	12.90368	0.23986	4.76203	0.08348
4	opt	39.26647	23.98959	18.17301	0.06916	20.37456	0.27432	7.73872	0.09529
	target	37.61994	22.84971	18.06242	0.07037	19.65061	0.27902	7.45266	0.09696
5	opt	34.83376	11.86434	11.06079	0.04532	14.45105	0.17961	5.53222	0.06243
	target	34.45835	12.87345	11.56119	0.04626	14.73481	0.18338	5.63380	0.06373

Source: own work.

In addition, the identified (opt) modules listed in Table 3 were used to perform calculations using the FEM model. The results are shown in Figure 13. The alignment of the reference (target) and identified (opt) deflections is qualitatively very good, with slight differences visible quantitatively, e.g. for Samples 2 and 5.

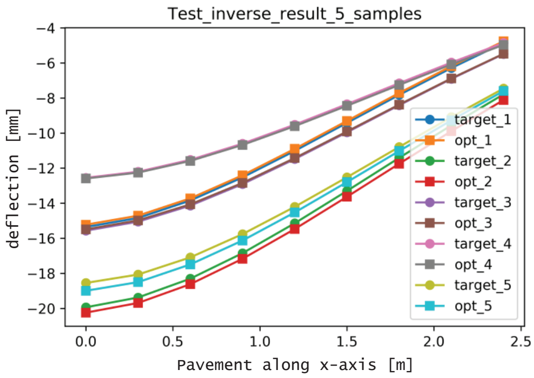

Fig. 13. Reference (target) and ANN identified (opt) deflections of the pavement for five test specimens

Source: own work.

ANN as an approximator of the model response

Method description

This approach defines and trains an artificial neural network, mapping the relationship between the layer modules and pavement deflections. In other words, the ANN directly approximates the response of the FEM numerical model to the load for given material properties. It can be written as follows:

$$\tilde{u}_k = N_k\left(E_j, \vec{w}\right), \tag{5}$$

where:

$\tilde{u}_k$ – network output, i.e. approximate pavement deflections,
N_k – operator performed by ANN,
E_j – network input in the form of layer modules,
$\vec{w}$ – net weight.

The trained ANN has four input and nine output neurons. The network architecture also consists of 10 hidden neurons in one layer. Network training was carried out on 95 training datasets (training and validation sets); the remaining five were used for network testing (test set). An example of the input and output data is presented in Table 4.

Table 4. Example of inputs and outputs data for trained 4-10-9 ANN

Dataset	Input [cm]				Output [GPa]								
	E_1	E_2	E_3	E_4	D0	D30	D60	D90	D120	D150	D180	D210	D240
Training	41.61	27.98	7.55	0.06	1.45	1.41	1.33	1.22	1.09	0.95	0.80	0.66	0.53
	17.91	29.52	15.22	0.04	2.12	2.06	1.96	1.82	1.65	1.46	1.27	1.08	8.96
Testing	39.81	27.51	9.45	0.04	2.00	1.96	1.87	1.75	1.59	1.43	1.25	1.08	0.91

Note: D0, D30, etc. denote the deflection at the geophone at 0 mm, 30 mm, etc. from the load centreline.

Source: own work.

The training, therefore, consisted of solving the following minimisation problem:

$$\min_{\vec{w}}\left[\sum_{p=1}^{95}\sum_{k=1}^{9}\left(N_k\left(E_j^{(p)},\vec{w}\right)-u_k^{(p)}\right)^2\right]. \tag{6}$$

After training the ANN, parameter identification consists of solving problem (2), in which the model response u_k is replaced with an approximated ANN response, i.e. the problem is solved:

$$\min_{\vec{w}}\left[\sum_{k=1}^{9}\left(N_k\left(E_j\right)-d_k\right)^2\right]. \tag{7}$$

Another advantage of this approach is the quick identification of parameters, because calling the ANN during optimisation is many times faster than calculating the FEM model. Moreover, the determined values of the modules certainly meet the optimality criterion defined by (7). Nevertheless, as understood in the original formulation (2), the solution's optimality is still only 'approximated'. An additional advantage of the approach is the ability to easily verify whether the problem (7) has one or more solutions. From this, one can conclude the uniqueness (or lack thereof) of the inverse problem.

Learning and identification results

The network training of the 4-10-9 architecture was performed as described above for 95 training samples. Figures 14 and 15 show the results of the pattern fits and deflections determined by the trained ANN for all 100 samples. The training results are very good; significantly better than in the case of the inverse relation approximation. A perfect match of the deflection values was achieved even though ANNs with an unfavourable architecture were trained (more outputs than inputs). It is worth noting that the quality of the fit of the test results does not differ from that of the training data, which indicates that the ANN has been trained correctly.

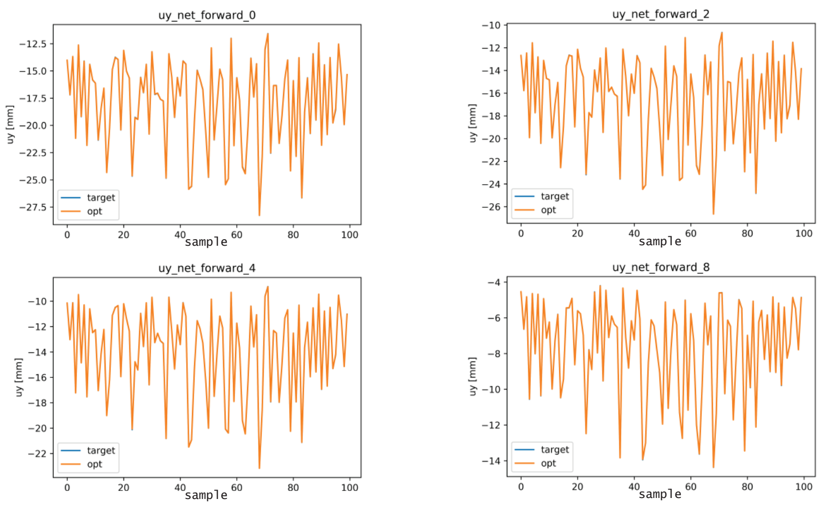

Fig. 14. Direct N_k network learning results for the 1st, 3rd, 5th and 9th reading point of pavement deflection: target – reference deflection values, opt – deflection values identified by the trained ANN. The last five points of the graphs refer to the testing data

Source: own work.

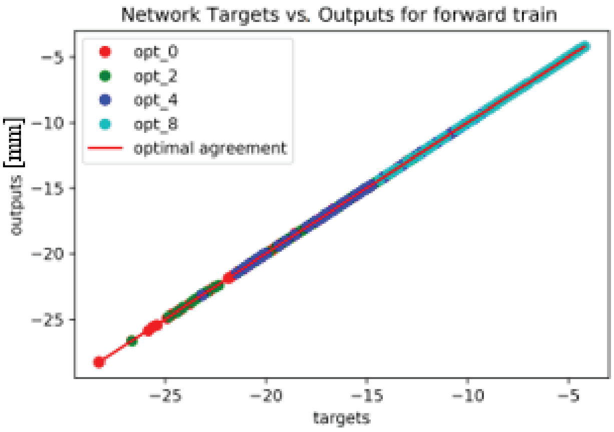

Fig. 15. Direct response of the network versus the pattern deflection values: u_1 (opt_0), u_3 (opt_2), u_5 (opt_4) and u_9 (opt_8). The points on the graph also contain test data

Source: own work.

After the ANN has been trained, the module parameters are identified by solving relation (2), in which the model response is replaced by the approximated ANN response. Therefore, the optimisation procedure (7) was used for the five test sets of deflections to identify the layer modules. This minimisation was done using the 'minimise' function from the Python language library scipy.optimize. Table 5 presents the numerical results of this identification and the averaged values of the modules. The differences in the determined and model modules are significant and may exceed even 40% in the case of concrete pavement layers, while the differences in the harmonic mean values are not greater than 5% (slightly larger differences are observed in the arithmetic means). This indicates the correct identification of the averaged, effective properties of the layered substrate, even if the modules of individual layers are determined with very significant errors in relation to the expected values.

Table 5. Numerical compilation of the reference (target) and the modules of successive layers of the model determined by optimisation defined by Formula (7) (opt) for five test samples

Test	Young's moduli [GPa]	E_1	E_2	E_3	E_4	Arithmetic mean	Harmonic mean	Weighted arithmetic mean	Weighted harmonic mean
1	opt	18.58321	18.33018	12.04275	0.06590	12.25551	0.26032	4.64233	0.09076
	target	26.37433	13.31490	9.39255	0.06586	12.28691	0.25970	4.72112	0.09069
2	opt	26.84213	21.27434	8.67345	0.04417	14.20852	0.17512	5.49418	0.06085
	target	21.91036	12.93881	13.07886	0.04399	11.99300	0.17443	4.49772	0.06061
3	opt	27.80757	21.70412	10.58812	0.06081	15.04016	0.24064	5.78936	0.08376
	target	19.01810	15.04311	17.49291	0.06060	12.90368	0.23986	4.76203	0.08348
4	opt	32.68377	23.56009	20.43075	0.07038	19.18625	0.27912	7.20772	0.09697
	target	37.61994	22.84971	18.06242	0.07037	19.65061	0.27902	7.45266	0.09696
5	opt	28.07469	21.76108	11.15155	0.04644	15.25844	0.18429	5.85362	0.06399
	target	34.45835	12.87345	11.56119	0.04626	14.73481	0.18338	5.63380	0.06373

Source: own work.

In addition, the identified (opt) modules listed in Table 5 were used to perform calculations using the FEM model to verify the fit of the reference (target) and identified pavement deflections. The results are shown in Figure 16. The alignment of the deflection lines is ideal for all test samples, despite significant discrepancies between the identified (opt) and reference (target) modules. This fact indicates the ambiguity of the considered inverse task, i.e. for different sets of modules, indistinguishable lines of pavement deflections are obtained from the FEM model.

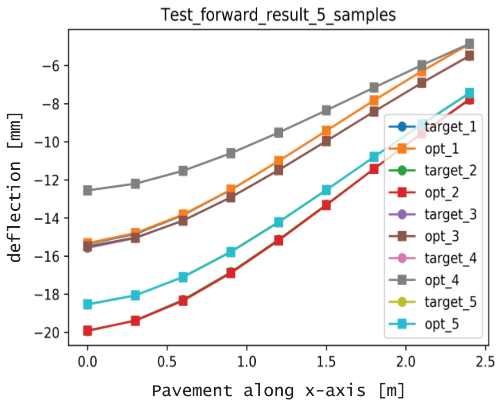

Fig. 16. Reference (target) and ANN identified (opt) deflections of the pavement for five test specimens

Source: own work.

DISCUSSION OF RESULTS AND CONCLUSIONS

The performed calculations of the identification of the parameters of the numerical model of the multi--layer pavement indicate that artificial neural networks can be successfully used for this purpose. A trained inverse network N^{-1} identifies layer modules directly, generating results with relative error not exceeding 15% (Method 1). The identification procedure using an artificial neural network N approximating the model, on the other hand, generates discrepancies in the identified modules up to 65% (Method 2). The percentages quoted only relate to rigid concrete pavement layers. In the case of the last, relatively soft layer of the subsoil, the identification results are much better; the discrepancies usually do not exceed 1% for both methods. Both methods also generate very good results in identifying the averaged, effective modules for the layers. With the mean harmonics, the divergences do not exceed 5% (see Tables 3 and 5).

The large module identification errors generated by Method 2 require a broader comment. As this method guarantees a very good adjustment of pavement deflections during the identification process, which is shown in Figure 16, obtaining very divergent modules of layers is somewhat surprising. The analysis of the variability of the minimised function while solving the problem (7) shows, however, that the solved problem of matching pavement deflections does indeed have many solutions, i.e. there are many sets of concrete pavement layer modules that generate indistinguishably close deflection lines. It is also an important observation from the point of view of Method 1, where significant network learning errors are observed (Figs 11 and 12) along with errors in deflection identification (Fig. 13). These errors are precisely the result of the reverse mapping ambiguity.

As a potential source of the ambiguity described above, we can point to the accepted ranges of variability of pavement layer modules, whose ranges partially overlap. In this context, it is worth noting the very good results of identification for the subsoil layer module, which differs from the pavement layer modules by three orders of magnitude. The reason for the ambiguity of the inverse relation may also be the assumptions related to the FEM numerical model used (adopted boundary conditions, plane deformation state) and insufficient representation of the model response (vertical deflections on the surface at several points). This issue requires further research, especially since the problems related to the unequivocal identification of material parameters of layered pavements are also reported by other authors, including in the case of identification based on the results of actual dynamic FWD tests. One of the possible approaches for obtaining better identification results is using the sophisticated ANN learning algorithms, which can include information about the derivatives of the pavement deflection line (Wojciechowski, 2011; Wojciechowski, 2012; Raissi, Perdikaris & Karniadakis, 2017).

An important conclusion resulting from the performed calculations is that a relatively small training dataset and small neural networks are sufficient to approximate both direct and inverse relations for the FWD test. The results of training and testing of the neural networks indicate that the method of drawing training data (LHS method), the applied sizes of the neural networks and the training methods are correct. Nevertheless, further testing of the presented methods is necessary, in particular using in situ test data.

Acknowledgements

This paper was completed while the first author was a doctoral candidate at the Interdisciplinary Doctoral School at the Lodz University of Technology.

The content contained in this paper was included in the first author's doctoral dissertation, which was published prior to this article.

REFERENCES

Alwattar, T. A. & Mian, A. (2019). Development of an Elastic Material Model for BCC Lattice Cell Structures Using Finite Element Analysis and Neural Networks Approaches. *Journal of Composites Science*, *3* (2), 33. https://doi.org/10.3390/jcs3020033

Aubdulnibe, F. F. & Jassim, k. A. (2019). An Application of Artificial Neural Networks (ANNs) To the Backcalculation of Flexible Pavement Moduli. *Journal of Physics: Conference Series*, *1362* (1), 012146. https://doi.org/10.1088/1742-6596/1362/1/012146

Boso, D. P. & Lefik, M. (2012). Inverse problem – Soft solution. In *Bytes and Science*. Barcelona: CIMNE.

Boso, D. P., Lefik, M. & Schrefler, B. (2012). Multiscale numerical modeling of composite material: A combined FE-ANN approach. *ECCOMAS 2012 – European Congress on Computational Methods in Applied Sciences and Engineering, e-Book Full Papers*, 8912–8926.

Boussinesq, J. (1885). *Application des potentiels à l'étude de l'équilibre et du mouvement des solides élastiques: Principalement au calcul des déformations et des pressions que produisent, dans ces solides, des efforts quelconques exercés sur une petite partie de leur surface ou de leur intérieur: Mémoire suivi de notes étendues sur divers points de physique, mathematique et d'analyse*. Paris: Gauthier-Villars.

Bredenhann, S. & Ven, M. (2004, 12 September). *Application of artificial neural networks in the back-calculation of flexible pavement layer moduli from deflection measurements*. Presented at the 8th Conference on Asphalt Pavements for Southern Africa.

Burczyński, T., Beluch, W., Długosz, A., Orantek, P. & Nowakowski, M. (2000). Evolutionary methods in inverse problems of engineering mechanics. In *Inverse Problems in Engineering Mechanics II International Symposium on Inverse Problems in Engineering Mechanics 2000 (ISIP 2000) Nagano, Japan* (553–562). https://doi.org/10.1016/B978-008043693-7/50131-8

Burczyński, T., Beluch, W., Długosz, A., Orantek, P. & Skrobol, A. (2006). Intelligent computing in inverse problem. *Computer Assisted Mechanics and Engineering Sciences*, *13* (1), 161–206.

Burmister, D. M. (1945). The General Theory of Stresses and Displacements in Layered Soil Systems. III. *Journal of Applied Physics*, *16* (5), 296–302. https://doi.org/10.1063/1.1707590

Elbagalati, O., Elseifi, M., Gaspard, K. & Zhang, Z. (2017). Development of an artificial neural network model to predict subgrade resilient modulus from continuous deflection measurements. *Canadian Journal of Civil Engineering*, *44* (9), 700–706. https://doi.org/10.1139/cjce-2017-0132

Galbusera, F., Casaroli, G. & Bassani, T. (2019). Artificial intelligence and machine learning in spine research. *JOR SPINE*, *2* (1), e1044. https://doi.org/10.1002/jsp2.1044

Gawin, D., Lefik, M. & Schrefler, B. A. (2001). ANN approach to sorption hysteresis within a coupled hygro-thermo-mechanical FE analysis. *International Journal for Numerical Methods in Engineering*, *50* (2), 299–323. https://doi.org/10.1002/1097-0207(20010120)50:2<299::AID-NME20>3.0.CO;2-Y

Lamb, H. (1997). I. On the propagation of tremors over the surface of an elastic solid. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, *203* (359–371), 1–42. https://doi.org/10.1098/rsta.1904.0013

Lefik, M. & Boso, D. P. (2016). *Identification of parameters of adsorption by ap-proximation of inverse relation and using artificial neural networks*. Presented at the Proceedings Solmech2016 Conference. Retrieved from: http://www.solmech2016.ippt.pan.pl/Abstracts/0177.pdf [accessed: 14.06.2023].

Lefik, M. (2004). Hybrid, finite element-artificial neural network model for composite materials. *Journal of Theoretical and Applied Mechanics*, *42* (3), 539–563.

Li, M. & Wang, H. (2017). Development of ANN-GA program for backcalculation of pavement moduli under FWD testing with viscoelastic and nonlinear parameters. *International Journal of Pavement Engineering*, *20*, 490–498. https://doi.org/10.1080/10298436.2017.1309197

Li, M. & Wang, H. (2018). Prediction of Asphalt Pavement Responses from FWD Surface Deflections Using Soft Computing Methods. *Journal of Transportation Engineering*, *144*. https://doi.org/10.1061/JPEODX.0000044

Lin, G., Zhang, P., Liu, J. & Wang, W. (2016). Elastic solutions due to a time-harmonic point load in isotropic multi-layered media. *Structural Engineering and Mechanics*, *57*, 327–355. https://doi.org/10.12989/sem.2016.57.2.327

Mckay, M. D., Beckman, R. J. & Conover, W. J. (2000). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, *42* (1), 55–61. https://doi.org/10.1080/00401706.2000.10485979

Meier, R. W. (1995). *Backcalculation of flexible pavement moduli from falling weight deflectometer data using artificial neural networks*. Atlanta: Georgia Institute of Technology.

Minsky, M. & Papert, S. (1969). *Perceptrons; an Introduction to Computational Geometry*. Cambridge, MA: MIT Press.

Momeni, E., Nazir, R., Jahed Armaghani, D. & Maizir, H. (2015). Application of Artificial Neural Network for Predicting Shaft and Tip Resistances of Concrete Piles. *Earth Sciences Research Journal*, *19* (1), 85–93. https://doi.org/10.15446/esrj.v19n1.38712

Nita, P. (2008). *Budowa i utrzymanie nawierzchni lotniskowych*. Sulejówek: Wydawnictwa Komunikacji i Łączności.

Raissi, M., Perdikaris, P. & Karniadakis, G. E. (2017, 28 November). Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. *arXiv*, 1711.10561. https://doi.org/10.48550/arXiv.1711.10561

Ruta, P., Krawczyk, B. & Szydło, A. (2015). Identification of pavement elastic moduli by means of impact test. *Engineering Structures*, *100*, 201–211. https://doi.org/10.1016/j.engstruct.2015.05.038

Saltan, M. & Terzi, S. (2005). Comparative analysis of using artificial neural networks (ANN) and gene expression programming (GEP) in backcalculation of pavement layer thickness. *Indian Journal of Engineering and Materials Sciences*, *12* (1).

Shahin, M., Jaksa, M. & Maier, H. (2001). Artificial Neural Network Applications in Geotechnical Engineering. *Australian Geomechanics*, *36*, 49–62.

Shahnazari, H., Tutunchian, M., Mashayekhi, M. & Amini, A. (2012). Application of Soft Computing for Prediction of Pavement Condition Index. *Journal of Transportation Engineering*, *138* (12), 1495–1506. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000454

Tadeusiewicz, R. (1993). *Sieci neuronowe*. Warszawa: Akademicka Oficyna Wydawnicza RM.

Vahab, M., Shahbodagh, B., Haghighat, E. & Khalili, N. (2023). Application of Physics-Informed Neural Networks for forward and inverse analysis of pile–soil interaction. *International Journal of Solids and Structures*, *277–278*, 112319. https://doi.org/10.1016/j.ijsolstr.2023.112319

Wesołowski, M. & Blacha, K. (2014). Assessment of Load Capacity of the Airport Pavement Structure with the Use of the ACN-PCN Method. *Research Works of Air Force Institute of Technology*, *35* (1). https://doi.org/10.1515/afit-2015-0002

Wikipedia (2023, 12 March). *Boeing 747*. Wikipedia. Retrieved from: https://pl.wikipedia.org/w/index.php?title=Boeing_747&oldid=69845987 [accessed: 14.06.2023].

Wojciechowski, M. (2011). Application of artificial neural network in soil parameter identification for deep excavation numerical model. *Computer Assisted Mechanics and Engineering Sciences*, *18*.

Wojciechowski, M. (2012). Solving Differential Equations by Means of Feed-Forward Artificial Neural Networks. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh & J. M. Zurada (Eds), *Artificial Intelligence and Soft Computing* (pp. 187–195). Berlin Heidelberg: Springer. https://doi.org/10.1007/978-3-642-29347-4_22

Wojciechowski, M. (2016). *ffnet: Feed-forward neural network for Python* [Python]. Retrieved from: https://github.com/mrkwjc/ffnet [accessed 15.06.2023].

Wojciechowski, M. (2018). *Fempy – Finite element method in Python* [Python]. Retrieved from: https://fempy.org/ [accessed: 15.03.2021].

Yoder, E. J. & Witczak, M. W. (1975). *Principles of Pavement Design*. Hoboken, NJ, USA: John Wiley & Sons. https://doi.org/10.1002/9780470172919

ZASTOSOWANIE METODY ELEMENTÓW SKOŃCZONYCH I SZTUCZNYCH SIECI NEURONOWYCH W IDENTYFIKACJI PARAMETRÓW NAWIERZCHNI WARSTWOWEJ

STRESZCZENIE

W artykule przeprowadzono analizę odwrotną parametrów mechanicznych istniejących nawierzchni warstwowych. Sztuczna sieć neuronowa (SSN) została wykorzystana do aproksymacji odpowiedzi modelu numerycznego nawierzchni na parametry wejściowe. Przedstawiono dwie metody. W pierwszej metodzie SSN bezpośrednio odwzorowuje odwrotną zależność między ugięciem nawierzchni a parametrami mechanicznymi. W drugiej metodzie aproksymowany model jest wykorzystywany w klasycznych obliczeniach wstecznych nawierzchni poprzez procedurę minimalizacji różnic między odpowiedzią modelu (obecnie sieci neuronowej) a pomiarami terenowymi uzyskanymi z testu ugięciomierzem dynamicznym (FWD). Identyfikowano tylko jeden parametr każdej warstwy nawierzchni – moduł Younga. Stwierdzono, że identyfikacja ta nie jest jednoznaczna. Oznacza to, że dane ugięcie można zaobserwować dla różnych zestawów modułów sztywności warstw nawierzchni. Średnia sztywność warstw jest jednak zawsze identyfikowana z dużą dokładnością.

Słowa kluczowe: sztuczna sieć neuronowa, problemy odwrotne, model numeryczny, identyfikacja parametrów, ugięciomierz dynamiczny, pomiary terenowe