

PAPER • OPEN ACCESS

## Artificial Neural Network Prediction of Airport Pavement Moduli Using Interpolated Surface Deflection Data

To cite this article: Nicola Baldo *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1203** 022112

View the [article online](#) for updates and enhancements.

### You may also like

- [An Estimation of the Maximum Take-off Weight Commuter Category Aircraft Based on Airstrip Condition on Remote High Terrain Area](#)  
L L Sunara and W Nirbito
- [Research on Risk Assessment with Uncontrollable High Thrust for Civil Airplane](#)  
Zou Candong and Xu Nanbo
- [Research on the Causes and Prevention of Airport Pavement Diseases](#)  
Xinpu Feng, Lei Guo, Geng Ren et al.



The Electrochemical Society  
Advancing solid state & electrochemical science & technology

242nd ECS Meeting

Oct 9 – 13, 2022 • Atlanta, GA, US

Abstract submission deadline: **April 8, 2022**

Connect. Engage. Champion. Empower. Accelerate.

**MOVE SCIENCE FORWARD**



Submit your abstract



# Artificial Neural Network Prediction of Airport Pavement Moduli Using Interpolated Surface Deflection Data

Nicola Baldo<sup>1</sup>, Matteo Miani<sup>1</sup>, Fabio Rondinella<sup>1</sup>, Clara Celauro<sup>2</sup>

<sup>1</sup>Polytechnic Department of Engineering and Architecture (DPIA), University of Udine, Via del Cottonificio 114, 33100 Udine, Italy

<sup>2</sup>Department of Civil, Environmental, Aerospace, and Materials Engineering (DICAM), University of Palermo, Via delle Scienze - Parco D'Orleans – Building n°8, 90128 Palermo, Italy

matteo.miani@phd.units.it

**Abstract.** Establishing the structural integrity of an airport pavement is crucial to assess its remaining life and implement strategies or priorities for action. In this context, the elastic modulus represents an effective indicator of the condition of the pavement which can be calculated through back-calculation procedures starting from surface deflections, obtained from a non-destructive test (such as the Heavy Weight Deflectometer). Nevertheless, the conventional inverse engineering analysis involves the use of an axial-symmetric pavement finite-element program able to evaluate stiffness values exclusively at the deflection measuring points. This study presents an alternative methodology for spatial modelling of the load-bearing capacity of the runway surface pavement layer from deflection data measured at specific points, using Shallow Artificial Neural Networks. The search of the optimal neural model hyperparameters has been addressed through a Bayesian Optimization procedure and a 5-fold cross-validation has been implemented for a fair performance evaluation, given the limited number of deflection measures available. Once the optimal model has been defined, the measured surface deflection data were linearly interpolated and resampled gridding data were used as a new input matrix of the neural model to predict the expected value of elastic moduli at non-sampled points on the runway. The optimal BO model has returned very satisfactory results with a value of Pearson Coefficient R averaged over 5-fold equal to 0.96597 and of Mean Squared Error averaged over 5-fold equal to 0.01849. In such a way, a contour map of the runway stiffness has been drawn, to provide a support tool for the planning of intervention priorities.

## 1. Introduction

Airport infrastructures are indeed economic activities capable of generating profits and attracting private investors. However, they have to meet increasing mobility and territorial development needs. In order to guarantee a constant level of service, transportation agencies and airport companies need increasingly reliable methods and tools for pavement management. In this context, non-destructive analyses represent an essential practice for the acquisition of data and information on the conditions of asphalt concrete and, therefore, for the definition of intervention priorities in management systems. The Heavy Weight Deflectometer (HWD) is the most widely used test to determine the structural integrity of a runway in a non-destructive manner and differs from the Falling Weight Deflectometer (FWD) due to the higher intensity of the loads. The HWD test allows to measure the deflection basin



generated by the application of an impulsive load. This load simulates the effects impressed by a moving wheel under dynamic conditions and is generated by dropping a suspended mass on a plate that rests on the pavement surface. The deflection basin, measured by means of geophones placed at different radial offsets, represents an overall pavement response to an applied load. For this reason, it is common practice to use deflections or a combination of them, through Deflection Basin Parameters (DBPs), as predictors of the pavement mechanical behavior [1]. In the literature about this topic, the back-calculation method is probably the best known and long-established in practice. This procedure is based on the multilayer elastic theory (MLE) in which stresses and strains are characterized by differential equations of the 4th order [2] and the solution of the inverse problem is sought. The first step involves the calculation of the deflection basin of the pavement according to the thickness of the layers, to the elastic modulus of the single layers (assuming values based on experience and best engineering practices) and to the magnitude of load. Once the theoretical deflection surface is calculated, the features are varied until the best match between the computed pavement deflections and the F/HWD measured pavement deflections is achieved [3]. However, some studies have observed discrepancies between back-calculated moduli and those obtained from laboratory tests. In particular, the back-calculated moduli of the surface layers are often within the accepted tolerance limits while those for base and subgrade layers are either underestimated or overestimated [4]. This is mainly due to the simplifications related to the MLE theory which does not take into account the effect of temperature and load frequency on the deformation of the different layers. In other words, traditional back-calculation methods neglect the dynamic effects of F/HWD loadings. Implementing these factors in conventional back-calculation software would lead to an excessive complexity of the model, with considerable computational cost [5]. Moreover, there is the problem that different back-calculation software provide solutions with different accuracies, depending on the pavement structures [6].

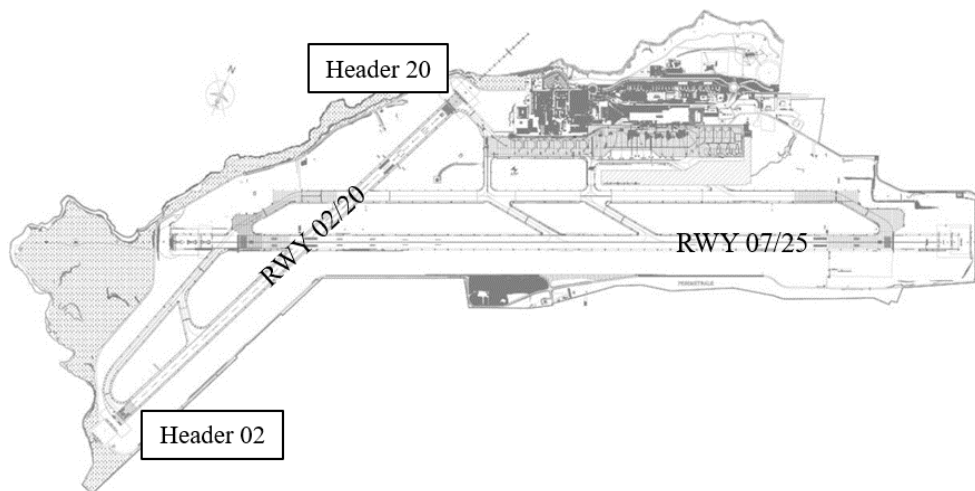
In recent years, soft computing techniques such as Artificial Neural Networks (ANNs) have been used in the pavement moduli back-calculation because they are able to identify the relationship between the variables of interest regardless of the physical nature of the problem. ANNs are highly recommended due to three merits over traditional back-calculation such as: less error, high efficiency, and output uniqueness [7]. Another important benefit of applying an ANN based back-calculation technique in routine F/HWD evaluations comes from the very high-speed data processing and analyses that can even be performed in the field [8]. Meier [9] experimented with an ANNs-based approach to back-calculate modulus pavement-layers through deflections obtained by FWD analysis: a feedforward back-propagation neural network was used and its performance was validated by field measurements (the Mean Squared Error was about 0.005). A few years later, Lee et al. [10] developed a neural model that used DBPs to back-calculate elastic moduli in asphalt and unbound material layers (the Root Mean Squared Error, or RMSE, was about 6.6 percent for the unbound layer and about 11.6 percent for the asphalt layer). However, an extensive experimental data set is necessary to successfully train a neural network and, for this reason, synthetic databases generated using finite element software are often used. In fact, traditionally ANN models are trained based on data obtained through multiple FE simulations using different combinations of material property, layer thickness, HWD loading magnitude and pavement temperature [11].

In this study, the objective was to identify a neural model able to predict the elastic modulus of the surface layer of the runway of Palermo airport as a function of the deflection measured below the point of application of the load as well as the spatial distribution of the measurement points. Moreover, in order to calculate the modulus value at an unsampled point (and therefore for the use of the model itself) the input vector of the neural model has been defined by means of a spatial interpolation. In particular, during the model definition phase, a procedure has been developed that takes into account the limited number of measurement points in common experimental campaigns (the k-fold cross-validation) and the difficult definition of the ANN model's structure (Bayesian optimization, BO). This model could therefore be an effective tool in airport infrastructure management systems.

## 2. Measurement Campaign

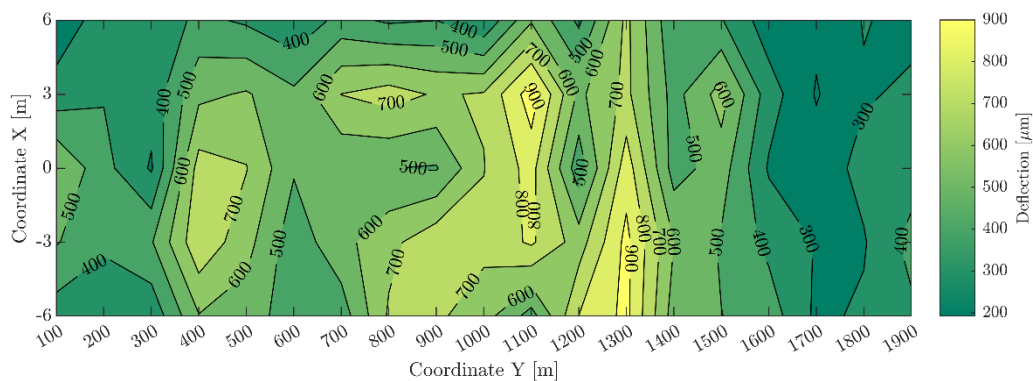
### 2.1. In situ Investigations

The international airport "Falcone e Borsellino" of Palermo-Punta Raisi is an Italian airport located 35 km west of the city of Palermo. The airport has two intersecting runways: the main one "RWY 07/25" and the secondary one "RWY 02/20".



**Figure 1.** Layout of Palermo Airport.

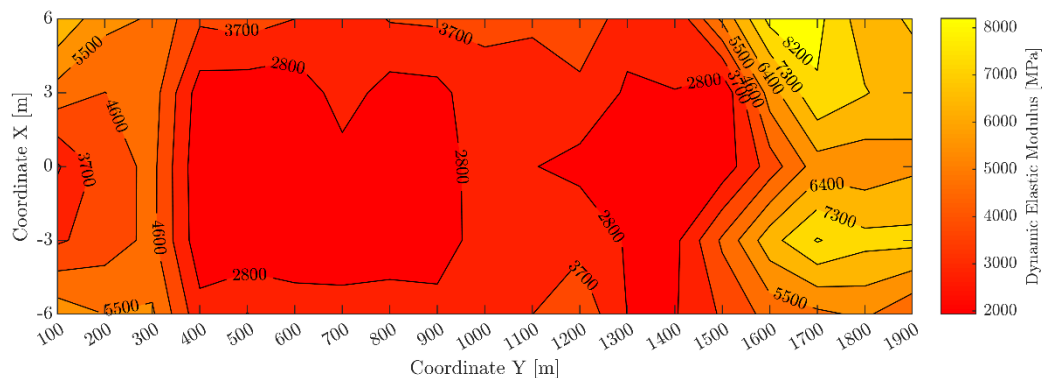
The HWD tests were performed on the RWY 02/20 with the Dynatest 8000 from header 02 to header 20 (figure 1) on five measuring lines (represented on the  $x$ -axis): runway central axis (0 m),  $\pm 3$  m-and  $\pm 6$  m-from the center of the runway. Each measure was taken by using a 100 m pitch between two tests along the runway. The portion of runway considered is 1800 m long (represented on the  $y$ -axis) for a total of 19 beating points for each measurement line. These data correspond to deflections measured automatically by means of accelerometric transducers, one of which is positioned in the center of the load plate ( $d_0$ ) and the others aligned radially at different distances – in mm – from the load axis (i.e.,  $d_{200}$  represents the deflection at 200 mm from the load axis,  $d_{300}$  at 300 mm, etc.). The following figure shows the deflections  $d_0$  expressed in  $\mu\text{m}$  obtained through tests performed on the above-mentioned measuring lines. The  $y$ -axis was appropriately scaled for representativity reasons. As shown in figure 2, the highest deflections are concentrated near the central axis of the runway. Moving away toward the ends of the runway, instead, the deflections tend to decrease in absolute value. This allows us to identify the "Touch-down zones" (TDZs) which are the impact areas between the aircraft's tires and the runway pavement during the landing phase.



**Figure 2.** Contour map of the deflections  $d_0$  on the runway measurement sections.

## 2.2. Back-calculation Process

The RO.M.E. (Road Moduli Evaluation) method was considered for the determination of the pavement dynamic elastic moduli. Such calculation procedure allows the different surface layers' moduli to be calculated starting from the tests carried out with the H.W.D. and from the thickness measurements of the pavement layers. The RO.M.E. relies on the elastic multilayer theory, the Boussinesq-Odemark equation and the Equivalent Thickness (ET) method to determine the stress/deformation state at each point of the bituminous layer, outlined as an infinite, isotropic, and homogeneous material. Once estimated the pavement layers' moduli, the RO.M.E. makes the theoretical deflection basin congruent with the one measured in the experimental investigation, by means of an iterative procedure. An exhaustive description of the RO.M.E. back-calculation method can be found on the work of Battiato et al. [12]. It was then possible to draw the back-calculated moduli contour map shown in figure 3. A clear evidence is that the zones characterized by the highest deflections (TDZs) have lower modulus values. Similarly, lower deflections are associated with higher moduli.



**Figure 3.** Contour map of the back-calculated moduli.

## 3. Theory and Calculations

### 3.1. Neural Modeling

ANNs are mathematical models that try to simulate the information processing and learning processes that occur in biological neural systems. Just like the human nervous system, an artificial neural network consists of multiple neurons connected by more or less reinforced connections. Through these artificial synapses, neurons are able to "communicate" by sending signals each other, processing the information and producing an output. They can be interpreted as logistic regression models provided with nonlinear activation functions. Typically, neurons are arranged in layers and connected through weighted and biased connections that progressively evolve according to a learning algorithm. In fact, during the usual training phase, the connections are gradually adjusted allowing the network to replicate the outputs associated with the inputs. As the complexity of the problem to analyze increases, it is advisable to add one or more layers between the input and the output ones. These additional layers are called "Hidden Layers". However, one hidden layer with a sufficient number of neurons allows the network to solve many multi-dimensional input-output fitting problems [13]. These structures are called "Shallow Neural Networks" (SNNs).

In the proposed SNN, the input layer consists of 3 neurons representative of the input features; the hidden layer is equipped with  $N$  neurons and 4 hypothetical activation units (AU): the hyperbolic tangent sigmoid (*tansig*), the exponential linear unit (*ELU*), the positive linear (*poslin*) and the linear (*purelin*) [14]; the output layer is realized with 1 neuron and the *purelin* function has been taken as activation. The input parameters considered were the deflection immediately below the loading plate ( $d_0$ ) and the coordinates ( $x, y$ ) of the correspondent measuring point along the runway. The output

evaluated was the calculated elastic moduli of the surface course. Before being inputted to the SNN both inputs and outputs were standardized, i.e., subtracted from the respective average and divided by the respective standard deviation. The SNN learns its weights and biases  $W$  through a supervised training phase. It is composed by a "forward pass" followed by a "backward pass" [15]. The former consists in using the features vector as input layer of the network and therefore computing, through the elaborations of the hidden layer, the output  $\hat{\mathbf{y}}$  of the network. The latter consists in comparing the target vector  $\mathbf{y}$  with the output just calculated through a loss function  $L(\hat{\mathbf{y}}, \mathbf{y})$ . In the current study, the training process uses the Mean Squared Error (MSE) as loss function, along with the Levenberg-Marquardt (LM) algorithm which is one of the fastest optimization methods for training shallow neural networks [16]. The LM algorithm (hereafter referred to *trainlm*) can be expressed as follows:

$$W^{(e)} = W^{(e-1)} - [J^T(W^{(e-1)})J(W^{(e-1)}) + \mu_{(e-1)}I]^{-1}J^T(W^{(e-1)})q(W^{(e-1)}) \quad (1)$$

where  $e \in \{0, \dots, E - 1\}$ ,  $J$  is the Jacobian matrix of the training loss  $L(\cdot)$  with respect to  $W^{(e-1)}$ ,  $I$  is the identity matrix and  $q = \widehat{\mathbf{y}}^{(d)} - \mathbf{y}^{(d)}$  is the vector of network errors. The scalar  $\mu$  (or learning rate) determines the algorithm's rate of convergence. However, when  $\mu$  increases, the LM algorithm takes a small step in the steeper direction of the loss function gradient and, on the contrary, the convergence becomes faster when  $\mu$  decreases. As a result, the LM algorithm needs an initial value of  $\mu$  at the first step, set in the current study at 0.001, and then increases (or decreases) the  $\mu$  value multiplying it by the factor  $\mu_{inc} > 1$  (or  $\mu_{dec} < 1$ ) if the previous step did not lead to a smaller MSE (or if it led to a smaller MSE). In this way, the value of the loss function tends to progressively decrease step by step. It is also necessary to set a  $\mu$  maximum value ( $\mu_{max}$ ) in order to stop the training if it is exceeded. Once the best weights and biases are found, they are fixed and the training loss function can finally be calculated:

$$L(\widehat{\mathbf{y}}^{(d)}, \mathbf{y}^{(d)}) = \|\widehat{\mathbf{y}}^{(d)} - \mathbf{y}^{(d)}\|_2^2 \quad (2)$$

### 3.2. SNN Regularization

When a neural model adapts too much to the learning dataset during the training phase, it loses its generalization ability with following poor performance in the testing phase. This situation is called overfitting. To avoid such a problem, the *early stopping* procedure has been implemented in the current study. It consists in dividing the dataset into three groups (the training set, the validation set, and the testing set). In this way, in fact, the first set allows to calculate the loss function as shown above. The second is instead used to control the model's generalization capabilities during the training process. In fact, in overfitting situations, the validation error increases and if this happens for a certain number of consecutive iterations  $\delta$  (assumed as model parameter), the learning phase stops. The parameter of the neural model is therefore set up referring to the minimum validation error. Finally, the testing set is used to evaluate the model's predictive capabilities on data it has never seen before.

### 3.3. k-fold Cross-Validation

Even before training the net, the k-fold Cross-Validation (CV) was planned: it is a resampling technique often used to elaborate an actual model on a limited data sample [17], such as the one in the current study. This procedure has only one parameter, k, which refers to the number of partitions into which a given data sample set must be divided. The k-fold CV is a commonly used method because it is easy to understand and it generally gives a better estimate rather than the one obtained by other models that divide the data by performing a simple "train/test" subdivision. The procedure is as follows: first mix the data randomly; then divide the dataset into k-partitions. Next, for each individual partition: take the individual group as a validation dataset; take the remaining partitions as a training dataset; create a model on the training set and evaluate it on the validation set; keep a record of the

validation score and then discard the model; summarize the skill of the models using the previously determined score sample. The average validation score is given as general performance of the model [18]. It is important to note that each observation in the data sample is assigned to a single group and remains in that group for the duration of the procedure. This means that it has been given to each sample the opportunity to be used once as a "validation dataset" and  $k-1$  times as a "training dataset". It was, finally, decided to give a  $k$ -value equal to 5, consistently with the relevant literature [19]. As a consequence of the regularization and  $k$ -fold CV, the dataset is split as follows: 64% of the available data for the training set, 16% for the validation set and 20% for the testing set.

### 3.4. Bayesian Hyperparameters Optimization

The definition of hyperparameters represents a crucial phase of modeling. The standard methodologies involve a random search or, starting from pre-established and well-known intervals of variation, a grid search. Recently, some automatic systems of hyperparameters research (based on Bayesian methods) have been introduced [20]. In Bayesian optimization processes, the main objective is to minimize one function  $f(\mathbf{x})$ , where  $\mathbf{x}$  belongs to some bounded set  $\mathcal{X} \subset \mathbb{R}$ . To determine which set of hyperparameters  $\mathbf{x}_{next} \in \mathcal{X}$  should be investigated next during the optimization, a function  $a : \mathcal{X} \rightarrow \mathbb{R}^+$  is used [21]. It is usually called Expected Improvement (EI) and it solves the equation  $\mathbf{x}_{next} = \text{argmax}_{\mathbf{x}} a(\mathbf{x})$ . Consequently, the function  $f(\cdot)$  is defined as  $f : X_N \times X_{AU} \times X_{\delta} \times X_{\mu} \times X_E \times X_{\mu_{inc}} \times X_{\mu_{dec}} \times X_{\mu_{max}} \rightarrow [0, \infty]$  and it has to be minimized when the *trainlm* algorithm is used with  $N, AU, \delta, \mu, E, \mu_{inc}, \mu_{dec}, \mu_{max}$  as hyperparameters.  $f(\cdot)$  will return a scalar that expresses the average MSE obtained by the SNN on the 5 test folds set before. The number of iterations performed by the Bayesian optimization algorithm has been fixed at 300. Iteration by iteration, the model learns which are the best areas of the given hyperparameters ranges to sample from.

In the current study, the hyperparameters to be optimized through Bayesian procedures are varied within the following ranges: the integer range  $X_N = \{4, \dots, 40\}$  for the number of neurons in the hidden layer;  $X_{AU} = \{tansig, ELU, poslin, purelin\}$  is the set of activation functions that can be applied after the hidden layer; the integer range  $X_{\delta} = \{5, \dots, 10\}$  for the maximum number  $\delta$  of validation failures; the range  $X_{\mu} = [10^{-4}, 10^{-2}]$  for the scalar  $\mu$ ; the integer range  $X_E = \{500, \dots, 5,000\}$  for the number of learning iterations; the range  $X_{\mu_{inc}} = [10^1, 10^3]$  for the increase factor  $\mu_{inc}$ ; the range  $X_{\mu_{dec}} = [10^{-3}, 10^{-1}]$  for the decrease factor  $\mu_{dec}$ ; the range  $X_{\mu_{max}} = [10^8, 10^{10}]$  for the maximum  $\mu$ . It should be pointed out that all the features outlined above have been implemented in MATLAB<sup>®</sup> software [14].

### 3.5. Linear Interpolation

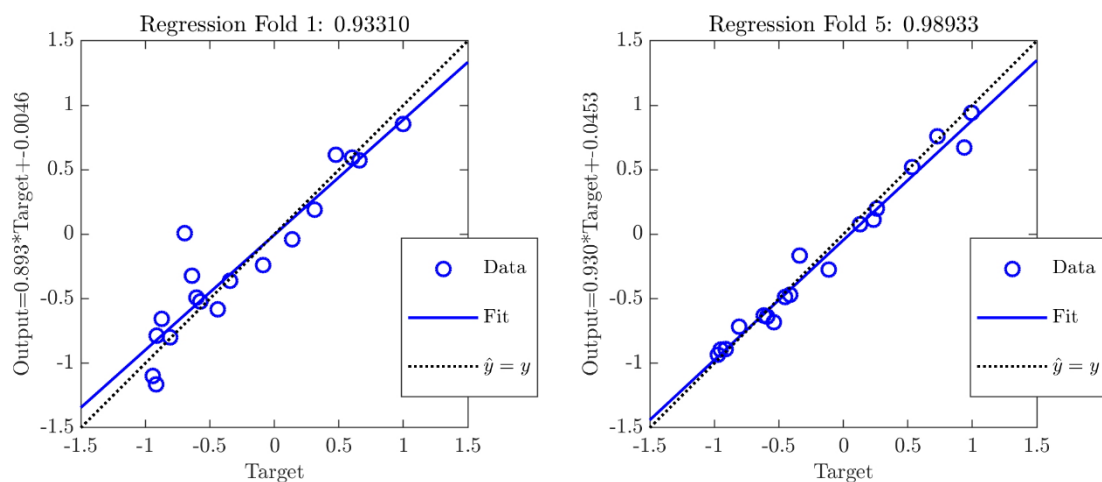
To get the information about the modulus value in non-sampled points of the runway, the neural model needs in input the deflection value at the same point and his  $x, y$  coordinates. For this reason, a grid of 13x361 nodes with a spacing of 1 m along the  $x$ -axis and 5 m along the  $y$ -axis was generated. Consequently, it was decided to connect the deflections through a linear structure. This methodology is among the most used ones for the development of digital surface models: to connect two or more sampled points, it would be necessary to introduce a statistical function, a priori unknown (because it depends also on the non-sampled points), which can be concave or convex, simple (e.g., linear) or complex (e.g., polynomial). The linear interpolation allows the uncertainty on the definition of the most suitable statistical function to be reduced (i.e., the accuracy of the interpolating map to be increased), which in most cases is chosen in a purely subjective way. Therefore, the deflections linear interpolation allows the presented neural model to be validated with respect to a dataset never been considered before.

#### 4. Results and discussions

Although there was great variability in the data set, justified by the different mechanical behaviors along the runway, the optimal BO model returns satisfactory results on all the 5 folds, here expressed in terms of Pearson coefficient ( $R$ ) scores. It expresses (if any) a linearity relationship between two statistical variables (in this case the back-calculated moduli and those predicted by the SNN model) and  $R$  values that exceed 0.8 are typical of a satisfactory correlation [22]. The actual model gives a  $R$ -score of 0.93310 in the worst case (fold 1), while in the best case (fold 5) it gives a score of 0.98933 (figure 4). Averaging the results over the 5 folds, the predictive capabilities of the proposed SNN model can be correctly evaluated:

$$R_{k-fold} = (0.93310 + 0.96887 + 0.95456 + 0.98399 + 0.98933)/5 = 0.96597 \quad (3)$$

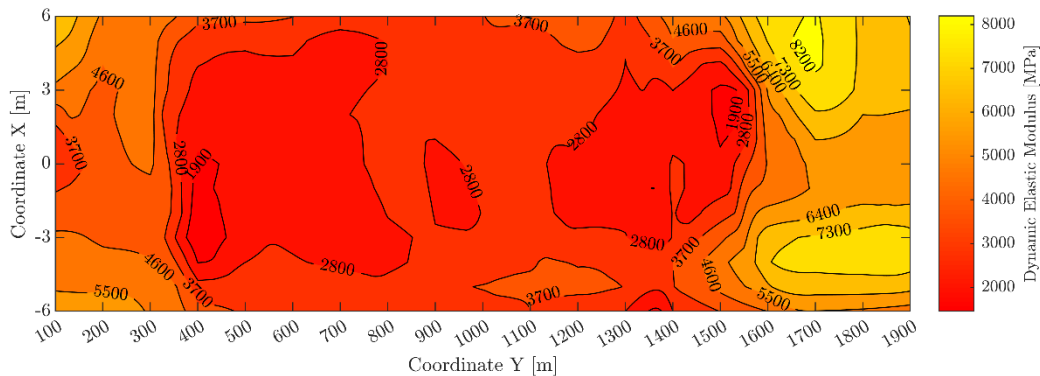
These results were obtained over 300 iterations of the BO process that, starting from a random hyperparameters set within the aforementioned variability ranges, have identified the best set for the current study problem. The optimal SNN is characterized by  $N = 30$  neurons in the hidden layer and an  $ELU$  activation function. Such neural network is trained according to the LM algorithm for  $E = 1051$  iterations with an initial learning step size  $\mu = 7.8e - 4$ . The parameter  $\mu$  is modified at each iteration by  $\mu_{inc} = 297.14$  or  $\mu_{dec} = 0.02$  to reduce the convergence time and to avoid local minima. The training process is stopped when the maximum number of iterations is reached or  $\mu_{max} = 1.4e9$  which denotes the convergence of the regularization process. However, the network properties set out above may be different if the proposed procedure is applied to a different experimental data set. The average MSE-score performed by using the LM algorithm over the 5 folds was 0.01849.



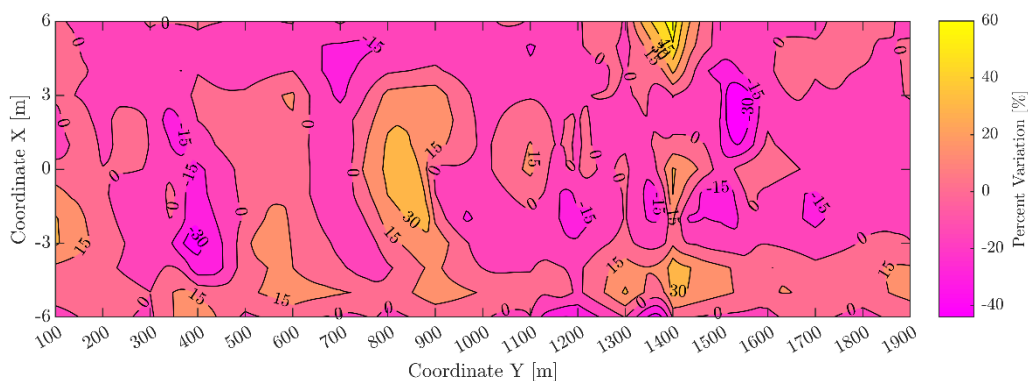
**Figure 4.** Summary of SNN performance.

The optimized neural model allowed to predict the modulus values at each point of the runway (figure 5). Therefore, it has been possible to compute the percentage variation between the back-calculated moduli and those predicted by the SNN model, which has been represented by means of the contour map shown in figure 6.





**Figure 5.** Contour map of the Dynamic Elastic Moduli predicted by the optimized SNN.



**Figure 6.** Contour map of Percent Variation between back-calculated and predicted moduli.

Although the neural model was trained on the back-calculated moduli and has achieved a  $R$ -score equal to 0.96597, the comparison between the contour maps of figure 3 and 5 shows the existence of runway areas where the percentage variation (figure 6) can be even higher than 50%. However, to confirm the results shown in figure 6 and validate the proposed SNN model, it would be necessary to carry out a further investigation in those points of the runway that show the major differences between the load-bearing capacity obtained from the interpolation of the back-calculated data and those predicted by SNN. In fact, differences pointed out in figure 6 could be related to the different modeling approach offered by the two presented methods: although it is possible to obtain a contour map of the moduli starting from the back-calculated data by means of a spatial interpolation technique, the result will be purely geometric and cannot account for the deflections variation along any direction of the runway, while the presented neural model provides in a generic point of the runway the value of the elastic modulus as a function of the deflection at that point (which in the current study was calculated by means of the linear interpolation method). Therefore, the results of a further experimental investigation are necessary to assess if the differences between the maps are due to methodological deficiencies of the spatial interpolation or to a biased prediction of the SNN model.

## 5. Conclusions

The Experimental data from Heavy Weight Deflectometer investigation has been analyzed. More attention was paid for the deflection at the center of the HWD loading plate ( $d_0$ ). A neural modeling approach was adopted using three-layer SNNs. Bayesian Optimization procedures and k-fold cross validation were implemented in order to enrich the model and to guarantee the best predictive performance. There have been many advantages derived from the integration of SNNs with traditional backcalculation methodologies. Although the SNNs models represent a not physically based "black

box" method, this main disadvantage has been largely compensated by the possibility to obtain satisfactory predictions of the mechanical parameter in a relatively simple way. Compared to a traditional backcalculation procedure that allows the evaluation of a single beat point, the proposed methodology is able to manage a grid of points simultaneously. It has very short processing times while maintaining a similar reliability in terms of predictions (as confirmed by the very high R-values). A simple interpolation would estimate the modulus value through other modulus data: it would represent a purely mathematical method that estimates a parameter starting from the same parameter and does not keep minimally account of other factors. The neural model proposed, instead, provides at a generic point on the runway the value of the elastic modulus as a function of the deflection at that point, which can be derived from spatial interpolation (or through field measurement). It establishes a relationship between a measured parameter (the deflection) and an estimated one (the elastic modulus), trying to approximate the constitutive law with an analytical relationship. The map of the moduli that would be obtained by simple interpolation would certainly allow data to be extrapolated from any location, but it would be blind to the deflection basin. The neural model proposed instead, predicts the modulus value but requires the deflection data in input. It is a bit more onerous, but it is definitely sounder from an engineering point of view because it maintains the logic of a phenomenological relationship between modulus and deflection basin. It is worth pointing out that the proposed model has been applied to the case of Palermo runway only, but the procedure presented can be applied to any runway. At this stage, the current study does not consider the relationship between a combination of the deflections measured at each step and the dynamic elastic modulus. For future developments, it is recommended to study this effect by integrating new input variables such as the Deflection Basin Parameters (DBPs) that may be more closely related to the structural conditions of the pavement. It would be also relevant to analyze runway deflections historical series, obtained by the surveys periodically carried out for maintenance purposes. In this way, neural models could be used to predict the future trend of pavement deflections and the corresponding moduli, thus establishing intervention priorities over time.

## References

- [1] K. Gopalakrishnan, and M.R. Thompson. "Use of nondestructive test deflection data for predicting airport pavement performance." *Journal of transportation engineering*, 133(6), 389-395, 2007.
- [2] D.M. Burmister. "The general theory of stresses and displacements in layered systems." *I. Journal of Applied Physics*, vol. 16(2), pp. 89-94, 1945.
- [3] K. Chatti, Y. Ji, and R. Harichandran. "Dynamic time domain backcalculation of layer moduli, damping, and thicknesses in flexible pavements." *Transportation research record*, vol. 1869(1), pp. 106-116, 2004.
- [4] R. Siddharthan, , G.M. Norris, and J.A. Epps. "Use of FWD data for pavement material characterization and performance." *Journal of Transportation Engineering*, vol. 117(6), pp. 660-678, 1991.
- [5] A. Goel, and A. Das. "Nondestructive testing of asphalt pavements for structural condition evaluation: a state of the art." *Nondestructive Testing and Evaluation*, vol. 23(2), pp. 121-140, 2008.
- [6] G.R. Rada, C.A. Richter, and P.J. Stephanos. "Layer moduli from deflection measurements: software selection and development of strategic highway research program's procedure for flexible pavements." *Transportation Research Record*, vol. 1377(1), pp. 77-87, 1992.
- [7] M. Li, and H. Wang. "Development of ANN-GA program for backcalculation of pavement moduli under FWD testing with viscoelastic and nonlinear parameters." *International Journal of Pavement Engineering*, vol. 20(4), pp. 490-498, 2019.
- [8] K. Gopalakrishnan, H. Ceylan, and A. Guclu. "Airfield pavement deterioration assessment using stress-dependent neural network models." *Structure and Infrastructure Engineering*, vol. 5(6), pp. 487-496, 2009.

- [9] R.W. Meier. "Backcalculation of flexible pavement moduli from Falling Weight Deflectometer data using Artificial Neural Networks." *US Army Corps of Engineers, Waterways Experiment Station, Vicksburg, Technical Report GL-95-3*, 1995.
- [10] Y.C. Lee, Y.R. Kim, and S.R. Ranjithan. "Dynamic analysis-based approach to determine flexible pavement layer moduli using deflection basin parameters." *Transportation Research Record*, vol. 1639(1), pp. 36-42, 1998.
- [11] H. Wang, P. Xie, R. Ji, and J. Gagnon. "Prediction of airfield pavement responses from surface deflections: comparison between the traditional backcalculation approach and the ANN model." *Road Materials and Pavement Design*, article in press, 2020.
- [12] G. Battiato, E. Ame, and T. Wagner. "Description and implementation of RO. MA. for urban road and highway network maintenance." In *Conference Proceedings 1* (Vol. 2), 1994.
- [13] M.T. Hagan, H.B. Demuth, M.H. Beale, and O. De Jesús. "Neural Network Design", *PWS Publishing Co., Boston*, pp. (11)4-7, (12)9-14, 2014.
- [14] Math Works. "MATLAB: The Language of Technical Computing from Math Works"; *Math Works: Natick, MA, USA*, 2018.
- [15] N. Baldo, E. Manthos, and M. Pasetto. "Analysis of the mechanical behaviour of asphalt concretes using artificial neural networks." *Advances in Civil Engineering*, 2018.
- [16] M.T. Hagan, and M. Menhaj. "Training feed-forward networks with the Marquardt algorithm." *IEEE Transactions on Neural Networks*, vol. 5(6), pp. 989–993, 1994.
- [17] J.D. Rodriguez, A. Perez, and J.A. Lozano. "Sensitivity analysis of k-fold cross validation in prediction error estimation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32(3), pp. 569-575, 2010.
- [18] N. Baldo, E. Manthos, and M. Miani. "Stiffness modulus and marshall parameters of hot mix asphalts: Laboratory data modeling by artificial neural networks characterized by cross-validation." *Applied Sciences*, 9(17), 3502, 2019.
- [19] M. Kuhn, and K. Johnson. "Over-Fitting and Model Tuning." In: *Applied predictive modeling*, Springer, New York, pp. 61-92, 2013.
- [20] J. Snoek, H. Larochelle, and R.P. Adams. "Practical Bayesian Optimization of Machine Learning Algorithms." *Advances in Neural Information Processing Systems*, vol. 4, pp. 2951-2959, 2012.
- [21] N. Baldo, J. Valentin, E. Manthos, and M. Miani. "Numerical Characterization of High Modulus Asphalt Concrete Containing RAP: A Comparison among Optimized Shallow Neural Models." In *IOP Conference Series: Materials Science and Engineering* (Vol. 960, No. 2, p. 022083). IOP Publishing, 2020.
- [22] J. Benesty, J. Chen, Y. Huang, and I. Cohen. "Pearson correlation coefficient." In: *Noise reduction in speech processing*, Springer, Berlin, pp. 37-40, 2009.