PAPER • **OPEN ACCESS**

Numerical Characterization of High Modulus Asphalt Concrete Containing RAP: A Comparison among Optimized Shallow Neural Models

To cite this article: Nicola Baldo *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **960** 022083

IOP Conf. Series: Materials Science and Engineering **960** (2020) 022083 doi:10.1088/1757-899X/960/2/022083

Numerical Characterization of High Modulus Asphalt Concrete Containing RAP: A Comparison among Optimized Shallow Neural Models

Nicola Baldo[1], Jan Valentin[2], Evangelos Manthos[3], Matteo Miani[1]

[1]Polytechnic Department of Engineering and Architecture (DPIA), University of Udine, Via del Cotonificio 114, 33100 Udine, Italy
[2]Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7, 166 29 Prague, Czech Republic
[3]Department of Civil Engineering, Aristotle University of Thessaloniki, University Campus, 54124 Thessaloniki, Greece

nicola.baldo@uniud.it

Abstract. Knowing the relationship between the stiffness modulus and the empirical mechanical characteristics of asphalt concrete, road engineers may predict the expected results of costly laboratory tests and save both time and financial resources in the mix design phase. In fact, such a model would make it possible to assess a priori whether the stiffness of a specific mixture, characterised in the laboratory only by the common Marshall test, is suitable for the level of service required by the road pavement under analysis. In this study, 54 Marshall test specimens of high modulus asphalt concrete were prepared and tested in the laboratory to determine an empirical relationship between the stiffness modulus and Marshall stability by means of shallow artificial neural networks. Part out of these mixtures was characterised by different types of bitumen (20/30 or 50/70 penetration grade) and percentages of used reclaimed asphalt (RAP at 20% or 30%); a polymer modified bitumen was used in the preparation of the remaining Marshall test specimens, which do not contain RAP. For the complex and laborious identification of the neural model hyperparameters, which define its architecture and algorithmic functioning, the Bayesian optimization approach has been adopted. Although the results of this methodology depend on the predefined hyperparameters variability ranges, it allows an unbiased definition of the optimal neural model characteristics to be performed by minimizing (or maximizing) a loss function. In this study, the mean square error on 5 validation folds was used as a loss function, in order to avoid a poor performance evaluation due to the small number of samples. In addition, 3 different neural training algorithms were applied to compare results and convergence times. The procedure presented in this study is a valuable guide for the development of predictive models of asphalt concretes' behaviour, even for different types of bitumen and aggregates considered here.

1. Introduction

In the Czech Republic, high-modulus asphalt concrete mixtures (specified in the Czech Republic under the abbreviation “VMT”) have been in use since 2001. Their initial introduction was promoted by an extensive research project commissioned by the Ministry of Transportation, known under the project title “New Generation of Asphalt Pavements” (AVNG) which was realized by a large expert team 20 years ago. The starting point was a collection of practical findings and experience with the

application of such type of mixture in France starting from the early 1990s where the mixtures were originally referred to as EME (*Enrobés à Module Élevé*). The initial development of the mixtures in France was followed by a number of studies and practical implementations in other countries, where the mixtures gradually established themselves under the term HMAC (*High Modulus Asphalt Concrete*).

In the first stage, the original EME concept was based primarily on high stiffness values and excellent resistance to rutting while, at the time, the French approach distinguished several classes of mixtures: a higher stiffness modulus was required for base layers and a lower stiffness (S_{min} = 11,000 MPa) applied to binder layers. It shall be pointed out that the French modulus characteristics have been and still are, in compliance with the EN 12697-26 test standard, most often determined by the 4-point beam test or by direct tensile stress test. This can be a factor complicating a comparison of the original French limits to the limits introduced in the Czech Republic by technical specifications TP 151 (Technical Specifications of the Ministry of Transportation for High-Modulus Asphalt Mixtures). The minimum required values there are tested either by 2-point test on trapezoidal test specimens or by indirect tensile test on cylindrical specimens. Even in this case, both tests do not provide on one mixture at the same temperature identical stiffness value.

In the Czech republic the HMAC mixtures are distinguished from common asphalt concrete mixtures primarily by a more restricted grading curve range in comparison to AC_{bin} or AC_{base} of same maximum particle size, more stringent requirements for air voids content (3-5%-vol. for type testing and 2.5-6.0%-vol. for control testing), different requirements for the content of soluble bituminous binder (4.2-5.4% by mass for HMAC 22 and 4.4-5.6% by mass for HMAC 16) and, first and foremost, by the required minimum stiffness determined at 15°C; i.e. criterion S_{min} = 9,000 MPa. For long-life pavement concepts additionally fatigue criterion is set with min. ε_6 = 125. Recently the discussion was raised to introduce the fatigue parameter also to HMAC variants if their stiffness at 15°C is larger than 13,500 MPa and there is a potential risk of limited fatigue life. The performance related behaviour is further defined by the minimum required water susceptibility ratio, ITSR ≥ 80%, and by resistance to permanent deformation which is tested on a small device in air bath at 50°C according to EN 12697-22 (wheel tracking test). In this case, the key parameters and their required maximum values are PRD_{AIR} = 3.0% and WTS_{AIR} = 0.05 mm/103 cycles. For HMAC mixtures different types if bituminous binders can be used. The most common are paving grades and hard paving grades like 20/30 or 15/25 as well as modified binders PMB 25/55-60 (-65) or PMB 10/45-60. It is allowed to use also multi-grade bituminous binders as defined by EN 13924-2. Similarly, concepts for warm mix asphalt can be used, whereas the so far most commonly applied solution was by doping either the bitumen or the asphalt mixture by fatty acid amides (e.g. Licomont) since such solution improves the resistance to permanent deformation and increases stiffness. Experimental tests were done with using crumb rubber modified bitumen as well, nevertheless, this option is still under extensive expert discussion. With respect to reclaimed asphalt (RA), it is allowed according to TP 151 to substitute up to 30% aggregates by RA. Regularly up to 15% are used and introduced to the mixture by cold batching. If higher content is targeted double coated drum or parallel drum needs to be used on a mixing plant. For higher RA contents, rejuvenators are usually used in recent years or hard paving grade is replaced by 50/70 bitumen. Such solutions have been successfully introduced to some motorway projects in the last two years.

In order to optimize the composition of a bituminous mixture, computational approaches could be useful, especially to avoid further expensive tests, with respect to the minimum necessary for the mix design. The possibility of using constitutive models to numerically simulate the mechanical behaviour of materials has been already investigated [1] and some attempts have been made by means of artificial intelligence, mainly with respect to conventional asphalt concretes [2].

This article fits into the latter context and tries to identify an efficient procedure for modelling the stiffness of asphalt mixes, even of the high modulus type, with and without RAP, by means of the Marshall stability and some categorical variables (necessary to identify the type of mix).

2. Description of tested asphalt mixtures

In the present study, a set of 18 variants of HMAC mixtures was used. Many of these mixtures were produced as a regular product by mixing plants, some are part of further developments or optimizations. Mixtures containing either paving grade or modified binders are represented, in two cases WMA concept was used and, in many cases, HMAC contained 20-30% reclaimed asphalt. This covers most of the options which are technically allowed and are used in practice. Test specimens used for further assessments were compacted by 2x75 blows using Marshall hammer.

Table 1. Volumetric properties and mechanical characteristic of HMAC variants (mean values on 3 specimens)

Asphalt mix VMT (ID)	Bitumen	Bulk density g/cm^3	Maximum density g/cm^3	Voids content %	Marshall Stability kN	Marshall deformation 0.1 mm	Marshall stiffness kN/mm	Stiffness at 15°C MPa
Standard asphalt concretes:								
30RA_F-A	20/30	2.447	2.640	7.3%	21.9	32	0.69	15,474
30RA_F-B	20/30	2.456	2.647	7.2%	20.9	45	0.47	14,944
30RA_F-C	20/30	2.478	2.663	7.0%	24.3	24	1.00	15,654
20RA_F-1	20/30	2.464	2.676	7.9%	20.4	43	0.51	13,157
20RA_F-2	20/30	2.476	2.682	7.7%	20.5	49	0.42	12,601
20RA_F-4	20/30	2.454	2.678	8.4%	24.0	43	0.60	15,371
20RA_F-6	20/30	2.418	2.667	9.3%	23.1	32	0.73	12,344
30RA_F-D	50/70	2.546	2.617	2.7%	20.5	57	0.38	12,691
30RA_F-E1	50/70	2.537	2.607	2.7%	20.8	55	0.37	13,081
30RA_F-E2	50/70	2.545	2.602	2.2%	19.0	61	0.32	13,539
30RA_F-F	50/70	2.550	2.626	2.9%	21.7	52	0.42	16,081
20RA_F-5	50/70	2.482	2.639	6.0%	21.0	38	0.56	13,037
20RA_P-A	50/70	2.410	2.496	3.4%	13.1	50	0.26	8,622
NT_Lic_1	20/30	2.394	2.490	3.9%	19.0	51	0.37	14,580
NT_Lic_2	20/30	2.302	2.490	7.6%	17.2	76	0.23	14,842
PMB-modified asphalt concretes:								
SK-1	25/55-60	2.361	2.436	3.1%	20.5	68	0.30	10,675
SK-2	25/55-60	2.364	2.436	3.0%	17.9	30	0.60	6,444
NT_Lic_1_M	25/55-60	2.358	2.465	4.3%	19.2	101	0.19	9,875

For each variant bulk density and maximum density was determined according to EN 12697-5 and -6. Based on this air voids content was calculated. Soluble bitumen content was determined as well to define basic empirical characteristics which could be used for neural network analysis and learning. From the perspective of mechanical and deformation characteristics, it was decided to focus on Marshall test parameters (Marshall stability, Marshall Flow and Marshall stiffness) as defined by EN 12697-34. The test was done after conditioning the test specimens in a water bath of 60°C. Mainly Marshall stability and Marshall stiffness were compared to stiffness determined according to EN 12697-26, method C (indirect tensile test on cylindrical test specimens), performed at 15°C as a standard test temperature required in the Czech Republic. For this test, a common universal testing machine was used and the resulting stiffness is based on 5 test cycles applied for two directions on test specimen with a rising time of 124 ms.

Volumetric properties and Marshall test results are summarized in Table 1. With respect to voids content variants produced on asphalt mixing plants where samples were taken for control testing fulfil the limits given by the interval for control testing. Only the sample VMT NT_Lic_2 which test specimens were compacted at 130°C as an extreme check for a potential drop in working temperature since it was a WMA variant, it showed that the limiting temperature is most probably rather between 140-150°C. With respect to used hard paving grade (20/30) and the typical potentials of the used fatty acid amid, this result is not surprising. For many options of VMT 30RA F or VMT 20RA F, the voids content levels were either too high (> 6,0%-vol.) or too low (< 2,5%-vol.). Since for defining the possible functionality between stiffness and Marshall test parameters not only the ideal values are important but dissonant results (with respect to voids content) may help to identify the sensitivity of such functionality, these variants were tested and included in the study as well.

Results for HMAC mixtures containing RA show the challenge to find appropriate balance between the grading curve limits, bitumen content, possible variations of degraded bitumen content in RAP and voids content. In several cases, it was shown that higher voids content results in high stiffness values. In some cases, the bitumen content played in this respect nearly no role (compare e.g. options VMT 30RA F-A to F-C). On the other hand, if the voids content was very low (e.g. options VMT 30RA F-D to F-F) the impact of receiving significantly lower stiffness was not true. These both sets of experimentally designed HMACs are characterized by higher reclaimed asphalt content which seems to have a dominant impact. The difference in stiffness values can be given by the used paving grades (hard grade for the first set and typical 50/70 paving grade for the second set). In the first set additionally, the reason for higher voids content can be due to the compaction temperature which was 160-165°C. In the case of combining RA and hard paving grade, it can be even possible that the compaction temperature needs to be a bit higher. The results for VMT 20RA F-1 to F-6 provide a similar resume. Of course, there might be another aspect which needs to be considered, especially if comparing VMT 20RA F-5 and VMT 20RA P-A. In this case, the same concept for mix composition was used with the only difference in bitumen content which was by 0.4% lower for the latter mixture. The voids content in these two cases fulfils the criteria given in TP 151, but the stiffness values have a difference of 4,500 MPa. One aspect might be the used aggregates, since the first mix contains granitic porphyry and the latter mix uses spilite aggregate which might be a little bit softer than the first one. Heterogeneity of reclaimed asphalt is another aspect since it is received from various pavements and their wearing and binder courses (or even asphalt based courses). For the neuronal network analysis aspect of aggregate type and reclaimed asphalt origin were not considered, for the future maybe they will need to be reflected as well.

If looking on the bitumen content for all 18 assessed HMAC variants, the received values are within the range as required by the specification TP 151. Most of the variants are even in a range of 4.5 to 5.0% by mass resulting in a very small difference in bitumen content. Marshall test characteristics do not provide any clear trends. Since this test and its characteristics are not required as a standard parameter for HMAC it is even not possible to compare it with some limiting threshold values. In specifications TP 151, it is only stated that Marshall stability and Marshall flow can be taken as the first indication before stiffness according to EN 12697-26 is determined. In this case and only for paving grades, multi-grade bitumen and hard paving grades the specifications state that if Marshall stability is > 14 kN and Marshall flow is 20-50 0.1mm than it is highly probable that required stiffness will be reached as well. Therefore, in this study we focused on this functionality.

Lastly, Table 1 provides data about stiffness. Only two mixtures showed lower values than the minimum requirement from the specification TP 151 – variant VMT 20RA_P-A and VMT SK-2. For the later, the value is very low. On the other hand, usually for HMACs with PMB 25/55-60, it is challenging to exceed the limit of 9,000 MPa due to the elastic behaviour of this polymer modified binder and several optimizations and modifications of the grading curve or even used aggregate are

needed. On the contrary for base layers especially if extended lifetime would be expected such behaviour and limited stiffness might be more positive.

3. Theory and calculations

3.1. Artificial Neural Networks (ANNs)

ANNs are mathematical models that aim to explain the information processing scheme happening in biological systems [3,4]. Towards the replication of the brain's structure, an ANN is composed of sets of neurons which are connected together by weighted connections. Through these, neurons are capable of sending signals to each other in order to process information and ultimately produce an outcome. From the point of view of pattern recognition problems, ANNs are a general class of nonlinear parametric functions. In their simplest form, which is also referred as multi-layer perceptrons, neurons are arranged in multiple sequential layers. These, called hidden layers, are organized in a stack that lays between an input and an output layer. Artificial neurons are implemented as logistic regression models with nonlinear activation functions. Each of them is chained, through weighted and biased connections, to every neuron present in the previous layer of the network. During a training process, the connections are adjusted by means of a learning rule, in order to replicate the ground-truth output (target) associated with the input pattern (supervised learning). By increasing the number of hidden layers and/or the number of neurons, it is possible to control the complexity and the richness of the relations between input and output patterns. However, Shallow Neural Networks (SNNs), i.e. two-layer perceptron networks, have been shown to solve arbitrarily well any multi-dimensional input-output fitting problem by providing a sufficient number of neurons in its only hidden layer [5].

In the proposed SNN, the input layer consists of 2 neurons corresponding to the number of input features; the hidden layer is provided with N neurons and passed to an exponential linear (ELU), hyperbolic tangent ($tanh$) or rectified linear ($ReLU$) activation unit; the output layer is realized with 1 neuron and the identity function is considered as activation. The input parameters considered were the bitumen type, the percentage of RAP used and the Marshall Stability. The output evaluated was the stiffness modulus of the asphalt concrete. Before being inputted to the ANN, each feature contained in the feature vectors was standardized, i.e. the respective mean was subtracted and division by the respective standard deviation was applied. The same procedure was performed for the target feature vectors, where each target variable was subtracted by its mean and divided by its standard deviation computed.

3.2. ANN optimization

The ANNs' weights and biases W are learned with a supervised training phase. First, a forward pass of the network is performed, i.e. the feature vector is presented as an input layer of the network. The activation is obtained by the hidden layer and then the output of the network $\hat{\boldsymbol{y}}$ is computed. At this point, the backward pass is performed. $\hat{\boldsymbol{y}}$ is compared to the ground-truth vector $\boldsymbol{y}$ by means of a loss function $L(\hat{\boldsymbol{y}}, \boldsymbol{y})$. Then, the backpropagation algorithm [3] is used to compute the gradients of the loss with respect to the parameters W. These are used in the learning rule to update the weights and biases of the network, i.e.

$$W^{(e)} = W^{(e-1)} - \alpha \nabla E[W^{(e-1)}], e \in \{0, \cdots, E-1\} \quad (1)$$

where $W^{(e-1)}$ are the parameters values at iteration $e-1$. $\nabla E[W^{(e-1)}]$ is the first order approximation of the gradients of the weights at time step t and it is estimated by computing the derivative of $L(\cdot)$ with respect to $W^{(e-1)}$. The process is repeated for a fixed number of iterations E, until the loss value is reduced to a minimum. α is the learning rate and it is used to control the step size of the movement towards such point. After that the ANN is trained, the obtained weights and

biases are kept fixed while test patterns are processed just in the forward manner. In this work the Mean Squared Error (MSE) is considered as training loss L:

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

The aforementioned training process, with the update rule introduced in Equation 1, is known as Gradient Descent (GD).

It is worth pointing out that the selection of the learning rate affects the performance of the GD algorithm: too small α values determine very long convergence times, while too high values can make the algorithm unstable, deviating the solution from the absolute minimum. In order to improve convergence speed and quality of ANNs, different improvements have been proposed to this algorithm. Two heuristic methods have been shown to be effective for increased performance. The first referred as Variable Learning Rate Backpropagation (VLBP) [5,6], addresses the issue of choosing an appropriate learning rate for the steepest descent algorithm when the curvature of the error surface varies drastically over the parameters' space. In fact, a large learning rate may be taken in flat regions of the error surface, while a small learning rate may be appropriate when the curvature in a certain region is high. Therefore, convergence can be accelerated by varying the learning rate during the training course, according to the performance of the algorithm. The application of this method requires some changes in the training process: at each epoch, a comparison is performed between the squared error produced by the learning rate at the iteration $e-1$ and that at the current iteration e. If such error increases by more than a set percentage L_{max}, then the new weights and biases are discarded. In addition, the learning rate is multiplied by a reducing factor α_{dec}. Otherwise (the error increases by less than L_{max}), the weight update is kept and the learning rate is unchanged. If the network error decreases at the iteration e, then the new weights and biases are accepted and the learning rate is multiplied by a factor of increase α_{inc}. Obviously, the adoption of this method involves an increase in the number of the neural model's parameters, compared to the only parameter required by the GD algorithm. The second advancement, Momentum Backpropagation (MOBP) [5,6], is a method to reduce the number of oscillations in the algorithm's trajectory to the minimum error. Basically, it is a low-pass filter that allows a network to overcome a shallow local minimum and to respond to recent trends in the error surface. Adding a first-order filter to Equation 1 leads to the following modification of the GD algorithm:

$$W^{(e)} = W^{(e-1)} + \mu\Delta W^{(e-2)} - \alpha(1-\mu)\nabla E[W^{(e-1)}], e \in \{0, \cdots, E-1\} \quad (3)$$

where $\mu\Delta W^{(e-2)}$ is the previous $(e-2)$ weight update and μ the constant that defines the amount of momentum.

The aforementioned strategies can be combined to produce a gradient descent algorithm with momentum and variable learning rate backpropagation. In this study, three optimization algorithms were employed to compute the optimal weights of the implemented ANN, that is: a standard gradient descent (hereafter referred to as *traingd*), a GD with adaptive learning rate (*traingda*) and a GD with momentum and variable learning rate (*traingdx*).

3.3. ANN regularization

Overfitting is the situation where a machine learning model fits too much the training data, with subsequent poor performance on testing data. To overcome such issues, different solutions have been introduced which are known as regularization techniques. In the current study setup, the *early stopping* procedure has been implemented. Such technique requires a random partition of the dataset into three subsets with predefined tasks: the training, validation and test. The first one allows the training loss to be computed, while the second one is used for monitoring the generalization error during the training phase. Since the validation error has to increase when the model overfits the data, the training process

stops when the validation error always increases during a number of iterations δ (assumed as network parameter), with respect to the training error. As a result, the ANN parameters are set at the minimum of the validation error. Finally, the test set is used on data never seen before to assess the model predictive skill. In this study, the validation and test datasets were composed of 16% and 20% of the original data respectively, while the remaining 64% makes the training dataset up. This dataset subdivision derives from the necessary application of a 5-fold cross-validation, followed by a partition of the training fold into the actual training sub-sets of 80% and the validation sub-set of 20%.

3.4. K-fold cross validation

To have a fair evaluation of the performance of the proposed model, the K-Fold Cross Validation was employed. This method suggests splitting the dataset of interest into k equally sized partitions referred as folds. k experiments are run where each data fold is in turn considered as the test set, while the remaining $k-1$ folds are used to form the training fold. After the whole procedure is completed, k validation scores are obtained and their average value is given as the general performance of the model. In this way, the issue of having potentially biased performance results is avoided. Indeed, with a fixed training-test split, the achieved performance may not be actual due to the different distribution of training and test data. In the study setting, a K-Fold Cross Validation method with $k = 5$ fold was implemented.

3.5. Bayesian hyperparameters optimization

Many machine learning models require a careful definition of different hyperparameters. This process is often considered an art as it requires time, expert knowledge and sometimes even brute force search. Standard methodologies include random or grid search that still requires a precise definition of parameters ranges and sampling strategies. To overcome these problems, an automatic hyperparameters search based on Bayesian methods have been introduced recently [7]. In Bayesian optimization, the goal is to find the minimum of a fusnction $f(\boldsymbol{x})$, where $\boldsymbol{x}$ belongs to some bounded set $\mathcal{X} \subset \mathbb{R}$. Algorithms for this problem construct and continuously update a probabilistic model for $f(\boldsymbol{x})$, which is exploited at the same time to make decisions about where in $\mathcal{X}$ to next evaluate $f(\cdot)$. The work of Snoke et al. [7] provides a practical guide on how to use Bayesian optimization for searching machine learning models hyperparameters. The authors propose a framework based on the assumption that $f(\cdot)$ is drawn from a Gaussian Process (GP) prior [8]. During the optimization process, the framework maintains a posterior distribution for $f(\cdot)$ which is updated as the results of running the machine learning experiments with different hyperparameters are observed. To determine which hyperparameters $\boldsymbol{x_{next}} \in \mathcal{X}$ should be evaluated next during the optimization, an acquisition function $a: \mathcal{X} \rightarrow \mathbb{R}^+$ is employed by solving $\boldsymbol{x_{next}} = argmax_{\boldsymbol{x}} a(\boldsymbol{x})$. Among the different existing definitions for $a(\cdot)$, the Expected Improvement (EI) [9] is perhaps the most popular method and has been shown to be efficient in the number of function evaluations required to find the global optimum of many multimodal black-box functions [10,11]. Such EI acquisition function evaluates the expected amount of improvement in $f(\cdot)$, ignoring the values that cause an increase. To escape a local minimum, the improvement proposed by Bull [11] allows the EI acquisition function to modify its behaviour when it estimates the over-exploitation of an area of the surface $f(\cdot)$. Thanks to this enhancement, such acquisition function is called Expected-Improvement-Plus (EIP).

In the problem of interest, the function $f(\cdot)$ defined as $f: X_N \times X_{AU} \times X_\delta \times X_\alpha \times X_E \rightarrow [0, \infty]$ has to be maximized when the *traingd* algorithm is used. Therefore, given the five hyperparameters N, AU, δ, α, E, the $f(\cdot)$ is a function that constructs an SNN with N neurons in the hidden layer, AU as activation function and runs a 5-Fold Cross Validation experiment in which the SNN is trained for E iterations with learning rate α. Such training process is early stopped after δ consecutive validation fails. $f(\cdot)$ returns a single scalar that expresses the average mean squared error obtained by the SNN on the 5 test folds. The Bayesian optimization algorithm is run for 300 iterations. At each iteration, candidates are sampled by the posterior distribution with the EIP algorithm and given to $f(\cdot)$ to run an

experiment. The retrieved performance is then used to update the posterior distribution. Iteration by iteration, the underlined GP learns which are the best areas of the given hyperparameters ranges to sample from. The same procedure also applies to the *traingda* and *traingdx* algorithms. However, the number of hyperparameters to be set increases: in fact, the function $f(\cdot)$ has also to consider the L_{max}, α_{dec}, α_{inc} parameters for the adaptive variation of α when *traingda* is applied and, in addition to these, the momentum constant μ when *traingdx* is used.

In this work, the following ranges were defined for the hyperparameters to be optimized through the Bayesian methodology: the integer range $X_N = \{4, \cdots, 40\}$, for the number of neurons in the hidden layer of the network; $X_{AU} = \{tanh, ReLU, ELU\}$ is the set of activation layers to apply after the hidden layer; the integer range $X_\delta = \{5, \cdots, 10\}$, for the maximum number δ of validation failures; the range $X_\alpha = [10^{-4}, 10^{-2}]$, for the learning rate α; the integer range $X_E = \{500, \cdots, 5{,}000\}$, for the number of learning iterations; the range $X_{L_{max}} = [1.03, 1.15]$, for the maximum percentage of MSE increase (in VLBP); the range $X_{\alpha_{dec}} = [0.60, 0.90]$, for the learning rate factor to decrease (in VLBP); the range $X_{\alpha_{inc}} = [1.03, 1.15]$, for the learning rate factor to increase (in VLBP); the range $X_\mu = [0.40, 0.80]$, for the momentum constant (in MOBP).

It is important to point out that all the source code required for this study was implemented in MATLAB.

4. Results and discussions

Table 2 shows the results of the Bayesian optimization process for tuning the SNN model's hyperparameters. Firstly, the BO processes associated with the considered training algorithms led to the same network architecture, although these GD algorithms differ due to the enhancements described in Subsection 3.2: the optimal SNN is characterized by 27 neurons in the hidden layer and an activation layer with $ReLU$ function. Regarding the optimized parameters of the training algorithms (Table 2), a direct comparison between the obtained results is not feasible, due to the implementations considered to modify the standard GD algorithm. However, a few considerations can be made: the *traingdx* algorithm performs 3 times more iterations to optimize network weights and bias; the enhanced GD algorithms are characterized by initial learning rates with the same order of magnitude (10^{-4}); the adaptation parameters of alpha are different: the absolute value of the differences is 0.09 for α_{dec}, 0.08 for α_{inc} (quite large compared to its variability range) and 0.01 for L_{max}; finally, *traingdx* algorithm is sufficiently sensitive to the local gradient, thanks to the optimized μ of 0.75, to ignore small features in the error surface.

Table 2. Bayesian optimization results

GD Algorithms	N	AU	E	δ	α	α_{dec}	α_{inc}	L_{max}	μ
traingd	27	$ReLU$	1,304	6	6.40e-3	-	-	-	-
traingda	27	$ReLU$	1,273	10	8.26e-4	0.70	1.06	1.03	-
traingdx	27	$ReLU$	4,298	10	9.59e-4	0.61	1.14	1.04	0.75

The average error on the 5 folds scored differently between the three optimal models (Figure 1), as expected: the standard GD algorithm had the highest MSE-score of 0.02203, while the learning rate's adaptive variation (implemented in *traingda*) during the training course improved performance by 8.6% compared to *traingd*, scoring 0.02013. The lowest value of the mean square error, equal to 0.01906 (Figure 1), was reached by the SNN trained by means of the GD algorithm with momentum and adaptive learning rate backpropagation. In particular, *traingdx* resulted in 13.5% improvement over *traingd* and 5.3% over *traingda*. Therefore, the VLBP and MOBP heuristic methods have

improved the quality of the neural model, overcoming the limits of the standard GD algorithm, which probably ran into a local minimum.

It is worth pointing out that performance also depends on the variability ranges, fixed by the research engineer, for the hyperparameters to be optimized. However, the ranges set in this study provided a sufficiently large searching domain, as shown by the shape of the error surface. Looking at the graphs in Figure 1, the loss function has assumed, in the 300 iterations of the BO process, values ranging from 10^{-2} to 10^{0} as an order of magnitude. However, the higher number of scores below the performance level 0.1 suggests that the error surface is characterized by a wide lightly tilted region (where MSE-scores vary gradually) that becomes steeper in some areas of the solution space (typically at one or more of the domain edges), scoring values close to or above the unit. This result reveals the presence, within the domain, of hyperparameter's combinations not suitable for the case study, due to the use of wide ranges for one or more of the model parameters. Therefore, the performance of the optimal BO models in terms of an average error on the 5 folds is not affected by the limits imposed on the variability of the hyperparameters.

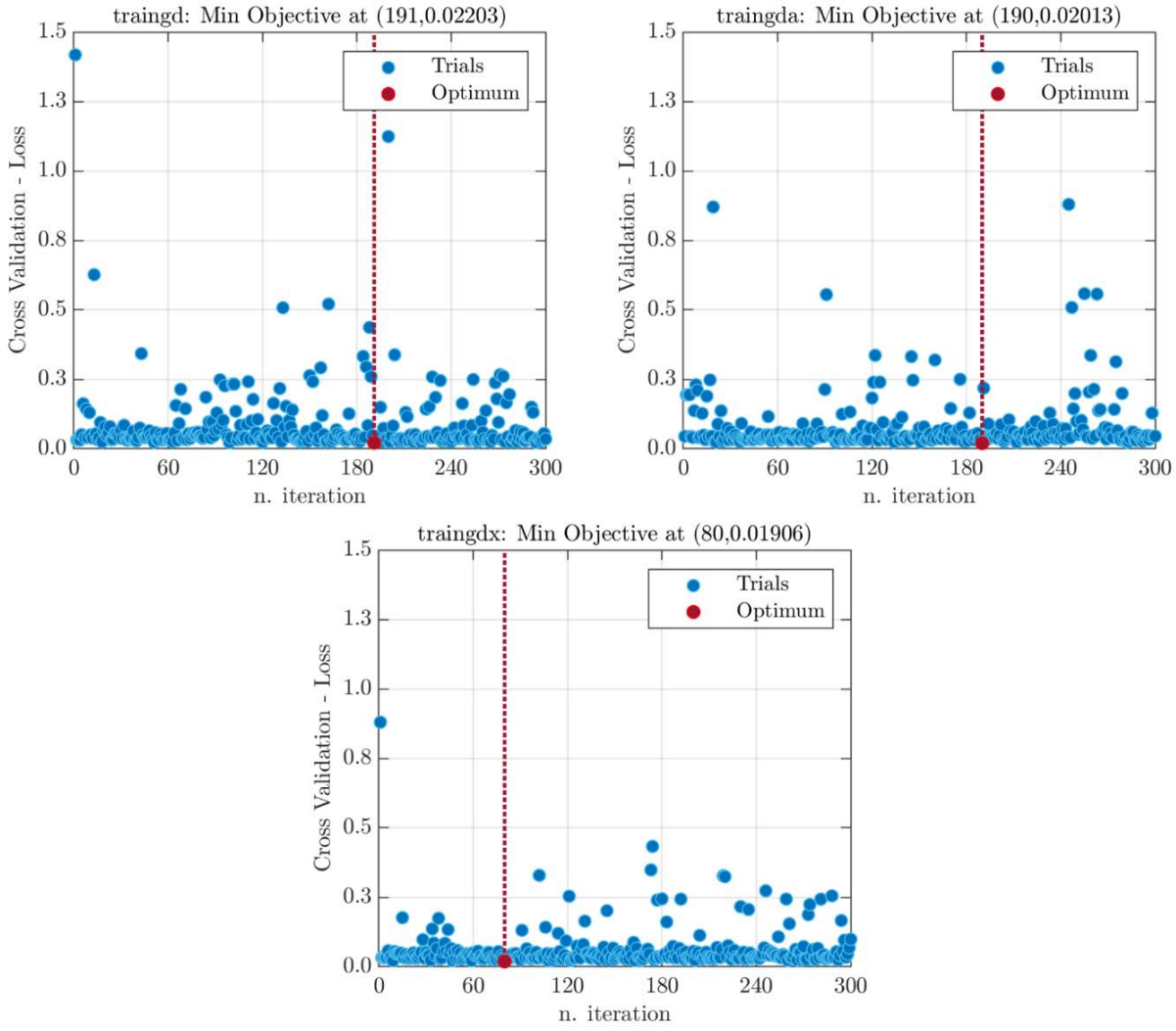

Figure 1. Average MSE-scores on the 5 test folds for the 300 iterations of the 3 algorithms

Further consideration should be made on the graphs in Figure 1: the MSE-scores take a downward trend, i.e. the Bayesian optimizer found network configurations with progressively better performance.

However, when the trend seems to stabilize and no new minimum is found (the Bayesian optimizer identified a region of the searching space, near the optimum, where the test error does not change significantly), the squared error suddenly increases. This happens due to the improvement proposed by Bull [11] to the EI acquisition function: the EIP function estimated an over-exploration of a specific area of the error surface and moved elsewhere, randomly selecting a new combination of hyperparameters

With regard to computational times, it is important to report that the first attempt values of the hyperparameters were randomly chosen by the Bayesian optimizer within their variability ranges, as a suitable combination for all considered models was impossible to identify (due to the different number of hyperparameters that characterize them). Therefore, the three BO processes reached the optimum at different iterations, as shown in Figure 1.

Despite the variability of the data set, justified by the use of mixtures with very different characteristics, the optimal BO model gives satisfactory results on all the 5 folds in terms of Pearson coefficient (R) scores. In fact, in the worst case (fold n°3), the R-score was 0.86827, while in the best case (fold n°1), the scored value was 0.97944. Averaging the results over the 5 folds, the predictive skills of the proposed SNN model can be properly evaluated:

$$R_{k-fold} = (0.97944 + 0.97698 + 0.86827 + 0.97564 + 0.90441)/5 = 0.94095 \qquad (4)$$

5. Conclusions

The main purpose of this study was to implement and apply a novel procedure, based on several well-established methodologies, for the determination of a reliable correlation between stiffness modulus and Marshall stability of asphalt concretes. In particular, the case study involved a set of 18 variants of HMAC mixtures, prepared with different types of bitumen and RAP percentages. The application of the ANNs required the evaluation of several network structures that had to be built by setting the model hyperparameters. The Bayesian optimization has provided a computationally-effective technique for solving time-consuming model selection problems. The shallow neural network that best fits the experimental data is characterized by 27 neurons in the hidden layer, $ReLU$ as activation function and is trained by the gradient descent algorithm with momentum and variable learning rate backpropagation (*traingdx*) for E = 4,298 iterations with an initial learning step size α = 9.59e-4. Such training process is early stopped after δ = 10 consecutive validation fails. The adaptation parameters of α are set to 0.61 for α_{dec}, 1.14 for α_{inc} and 1.04 for L_{max}.

The procedure presented in this study was explained in detail to give the reader an opportunity to replicate it. In fact, this procedure still stands for the development of any predictive model of the asphalt concretes' mechanical behaviour, even for mixtures different from those considered here. Although ANNs allow for satisfactory results, such approaches do not offer the possibility to follow directly the internal training process and to understand the physical meaning of the model factors. For this reason, they are sometimes referred to as "*black box*" methods. Furthermore, it is worth pointing out that this study did not consider the effect of aggregate gradation and bitumen content on the results. For future developments, it is recommended to study this effect by integrating new input variables.

References

[1] M. Pasetto, and N. Baldo, "Computational analysis of the creep behaviour of bituminous mixtures," Constr. Build. Mater., vol. 94, pp. 784–790, 2015.
[2] 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," Appl. Sci. (Basel), vol. 9(17), 3502, 2019.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” In J. A. Anderson, E. Rosenfeld (Eds.), *Neurocomputing: Foundations of Research*, MIT Press, Cambridge, pp. 696–699, 1988.
[4] W. S. McCulloch, and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” In J. A. Anderson, E. Rosenfeld (Eds.), *Neurocomputing: Foundations of Research*, MIT Press, Cambridge, pp. 15–27, 1988.
[5] 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.
[6] T. P. Vogl, J. K. Mangis, A. K. Zigler, W. T. Zink, and D. L. Alkon, “Accelerating the convergence of the backpropagation method,” *Biol. Cybern.*, vol. 59, pp. 256–264, 1988.
[7] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” *Adv. Neural. Inf. Process. Syst.*, vol. 2, pp. 2951–2959, 2012.
[8] C. E. Rasmussen, and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, pp. 105–118, 2006.
[9] J. Mockus, V. Tiešis, and A. Zilinskas, “The application of Bayesian methods for seeking the extremum,” In L. C. W. Dixon, G. P. Szego (Eds.), *Towards Global Optimization 2*, North Holland Publishing Co., Amsterdam, pp. 117–129, 1978.
[10] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design," ICLM 27th Proc. Int. Conf. Mach. Learn., pp. 1015–1022, 2010.
[11] A. D. Bull, "Convergence rates of efficient global optimization algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2879-2904, 2011.