ORIGINAL ARTICLE

Machine learning techniques to estimate the degree of binder activity of reclaimed asphalt pavement

Ramon Botella · Davide Lo Presti [ORCID] **· Kamilla Vasconcelos · Kinga Bernatowicz · Adriana H. Martínez · Rodrigo Miró · Luciano Specht · Edith Arámbula Mercado · Gustavo Menegusso Pires · Emiliano Pasquini · Chibuike Ogbo · Francesco Preti · Marco Pasetto · Ana Jiménez del Barco Carrión · Antonio Roberto · Marko Orešković · Kranthi K. Kuna · Gurunath Guduru · Amy Epps Martin · Alan Carter · Gaspare Giancontieri · Ahmed Abed · Eshan Dave · Gabrielle Tebaldi**

Abstract This paper describes the development of novel/state-of-art computational framework to accurately predict the degree of binder activity of a reclaimed asphalt pavement sample as a percentage of the indirect tensile strength (ITS) using a reduced number of input variables that are relatively easy to obtain, namely compaction temperature, air voids and ITS. Different machine learning (ML) techniques were applied to obtain the most accurate data representation model. Specifically, three ML techniques were applied: 6th-degree multivariate polynomial regression with regularization, artificial neural network and random forest regression. The three techniques produced models with very similar precision, reporting a mean absolute error ranging from 12.2 to 12.8% of maximum ITS on the test data set. The work

R. Botella · A. H. Martínez · R. Miró
Universitat Politècnica de Catalunya – BarcelonaTech,
Barcelona, Spain
e-mail: ramon.botella@upc.edu

A. H. Martínez
e-mail: adriana.martinez@upc.edu

R. Miró
e-mail: r.miro@upc.edu

D. Lo Presti · G. Giancontieri
University of Palermo, Palermo, Italy
e-mail: gaspare.giancontieri@unipa.it

D. Lo Presti (✉) · G. M. Pires · A. Abed
University of Nottingham, Nottingham, UK
e-mail: davide.lopresti@unipa.it

A. Abed
e-mail: ahmed.abed2@nottingham.ac.uk

K. Vasconcelos
University of São Paulo – Polytechnic School,
São Paulo, Brazil
e-mail: kamilla.vasconcelos@usp.br

K. Bernatowicz
Vall d'Hebron Institute of Oncology (VHIO), Barcelona,
Spain
e-mail: kbernatowicz@vhio.net

L. Specht
Federal University of Santa Maria, Santa Maria, Brazil
e-mail: luspecht@ufsm.com

E. A. Mercado · A. E. Martin
Texas A&M Transportation Institute, Bryan, USA
e-mail: e-arambula@tti.tamu.edu

A. E. Martin
e-mail: a-eppsmartin@tamu.edu

G. M. Pires
Dynatest Latam, São Paulo, Brazil

presented in this paper is an evolution in terms of data analysis of the results obtained within the interlaboratory tests conducted by Task Group 5 of the RILEM Technical Committee 264 on Reclaimed Asphalt Pavement. Hence, despite it has strong bonds with this framework, this work was developed independently and can be considered as a natural follow-up.

1 Introduction

1.1 Degree of activation

During the manufacture of a bituminous mix, the mixing operation aims to reduce segregation of the component materials to achieve homogeneity. In the case of a recycled mix, the mixing process must break the pre-existing bonds between the RAP particles and relocate these particles within the mix to avoid segregation and consequent poor quality of the final mix.

It is difficult to know to what extent an amount of RAP can contribute with its binder to the mix in such a way that it combines with the virgin bitumen to increase cohesion and adhesion. Thus, studies have been carried out to assess the performance of recycled mixes with different RAP contents [1–5]. When manufacturing a recycled mix, it cannot be guaranteed that the degree of mixing of both binders (i.e., virgin and recycled) will be 100%. In order to analyze this phenomenon, two possible scenarios are usually defined that represent the extreme mixing situations that could occur and that are accepted as ideal and far from what happens in reality: "full blending" and "black rock". This issue is of particular relevance when analyzing the response of recycled mixtures with high amounts of RAP [2].

Full blending represents what happens during the design process when it is assumed that both virgin and RAP binder are fully blended. At the other extreme is the so-called "black rock" scenario, which symbolizes that the RAP binder is unable to combine with the new binder, and in this case, it is the virgin binder that does all the wrapping and adhesive work. Finally, what is often accepted is what is called "partial blending", which, in the opinion of many researchers, could summarize what happens in reality, accepting that a certain percent of the aged RAP binder is blended with

E. Pasquini · M. Pasetto
University of Padua, Padua, Italy
e-mail: emiliano.pasquini@unipd.it

M. Pasetto
e-mail: marco.pasetto@unipd.it

C. Ogbo · F. Preti · E. Dave
University of New Hampshire, Durham, USA
e-mail: cuo1001@wildcats.unh.edu

F. Preti
e-mail: francesco.preti1992@icloud.com

E. Dave
e-mail: eshan.dave@unh.edu

F. Preti · A. Roberto · G. Tebaldi
University of Parma, Parma, Italy
e-mail: antonio.roberto@unipr.it

G. Tebaldi
e-mail: gtebaldi@unipr.it

A. J. del Barco Carrión
University of Granada, Granada, Spain
e-mail: ajbc@ugr.es

M. Orešković
University of Belgrade, Belgrade, Serbia
e-mail: moreskovic@grf.bg.ac.rs

K. K. Kuna · G. Guduru
Indian Institute of Technology Kharagpur, Kharagpur, India
e-mail: kranthi@iitkgp.ac.in

A. Carter
École de Technologie Supérieure, Montreal, Canada
e-mail: alan.carter@etsmtl.ca

the virgin binder, while another part remains adhered to the RAP aggregates and behaves in a similar way to them [6].

It is important to note that the behavior of recycled mixes is not only affected by the type and effectiveness of the mixing system, but also by the interaction between RAP and virgin binders. Mechanical mixing only attempts to get the virgin binder to coat the RAP particles, but the objective pursued when manufacturing a recycled mix is that the virgin binder (or the recycling agent) penetrates or diffuses through the RAP, reducing the viscosity of the RAP binder and recovering, to some extent, its original properties.

At the end of the last century, several researchers began to study the diffusion of virgin binder (or recycling agent) into recycled binder from RAP [7, 8]. These investigations confirmed that there was a relationship between the properties of the recycled mixes and the degree of mixing of the binders in the mix. Despite numerous studies, the lack of understanding of some of the mechanisms involved in mixing continues to be recognized today. Lo Presti et al. presented a methodology in 2020 (adapted from what was initially proposed by Menegusso et al. [9]) together with a nomenclature to differentiate certain properties that are key to this process [10].

They introduced a promising parameter called "degree of activity" (DoA), which is the minimum amount of active RAP binder that can be considered at the design stage of the recycled mix. Other blending parameters have been developed and analyzed such as: effective RA binder [11], transferred binder [12], mobilized binder [13–16], reactivated binder [17], etc., which were collected and described by Orešković et al. [18].

1.2 Machine learning

Machine learning (ML) employs different algorithms to fit data to a mathematical function. Linear regression would be the simplest example of an ML technique. A programming code that performs a linear regression systematically in any kind of data set (x_i, y_i) makes the machine learn by itself to predict y as a function of x. However, modelling physical systems and their outputs often requires more complex techniques, allowing to incorporate multiple input variables/features and allowing for non-linear, multi-dimensional fitting. In this study, three different techniques were applied, sorted from simpler to the more complex, i.e. multivariate polynomial regression (MPR), artificial neural network (ANN) and random forest regression (RFR). The first two were implemented using MATLAB 2021a, while the third was implemented using Python 3.9.5. From now on, the input variables x_i may be referred to as *features* while the output variables y_i as *labels*.

1.2.1 Multivariate polynomial regression background

In a multivariate linear regression, the main objective is to fit $h(\theta)$ such as Eq. 1:

$$h_\theta = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n, \tag{1}$$

to a data set of $(x_1^{(i)}, \ldots, x_n^{(i)}, y^{(i)})$ with the objective of making $h(\theta) \cong y$. Performing a linear regression means finding the parameters θ_i that minimize $h(\theta) - y$. To that end a $J(\theta)$ function is defined as Eq. 2:

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} h_\theta(x^{(i)}) - (y^{(i)})^2, \tag{2}$$

where m is the number of data samples and $x_0 \equiv 1$, is an independent term added to simplify vectorization, called the "bias" term. The $J(\theta_0, \theta_1, \ldots, \theta_n)$ is called the cost function and is representing the squared error of the fitting on predicting the actual outputs (or labels) $y^{(i)}$. Given a set of initial parameters $(\theta_0, \theta_1, \ldots, \theta_n)$, if the partial derivatives $\frac{\partial J(\theta_0, \theta_1, \ldots, \theta_n)}{\partial \theta_i}$ are computed and subtracted from the initial θ_j, a new θ_j that reduces $h(\theta) - y$ is obtained. This step-wise process is known as *Gradient Descent* and can be represented as Eq. 3:

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta_0, \theta_1, \ldots, \theta_n)}{\partial \theta_i}, \tag{3}$$

$$\frac{\partial J(\theta_0, \theta_1, \ldots, \theta_n)}{\partial \theta_i} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)})x_j^{(i)}, \tag{4}$$

where := means simultaneously update for every $j = 1, \ldots, n$. In other words, that every θ_j has to be updated only after the last one has been updated. This optimization process is called Gradient Descent and its convergence depends mostly on the shape of $J(\theta_0, \theta_1, \ldots, \theta_n)$ and the step parameter chosen α. Sometimes a linear regression function $h_\theta(x)$ is not the best fit to represent data complexity. In those cases, a

higher degree polynomial expression can help. One way of introducing higher exponential terms on $h-\theta(x)$ is by generating a new set of features x_j' by developing higher degree polynomial terms between the initial features x_j, as in Eq. 5:

$$1, x_1, x_2 = > 1, x_1, x_2, x_1^2, x_1 x_2, x_2^2 \qquad (5)$$

where the superscripts represent exponents, subscripts the different features and $x_0 \equiv 1$ by definition. In Eq. 5 the number of features was increased in 3 terms which are quadratic terms of x_1 and x_2. If Gradient Descent is applied using these six new features a quadratic multivariate regression would be obtained. The main concern when introducing higher degree polynomials on the features is the fact that given enough exponential terms and features, gradient descent would usually be able to find a complex enough function that fits the training data points. However, this hypothetical function would fail to predict new samples, because the relation found is tailored exclusively to the training set and not based on real correlation. This problem is known as *Overfitting*. To compensate for that, a weight can be added to the higher degree terms in features to ensure that the main fitting is done by the lower degree terms and the higher degree terms only add small corrections to the model. This is called Regularization, Eq. 6.

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}\Big(h_\theta(x^{(i)} - y^{(i)}\Big)^2 + \lambda \sum_{j=1}^{n}\theta_j^2, \qquad (6)$$

By adding a regularization parameter λ multiplied by the sum of squares of $\theta_1, \ldots, \theta_n$ (note that θ_0 is deliberately excluded from that summation since its feature is $x_0 \equiv 1$), overfitting can be prevented. The square of the higher degree terms is going to be a high magnitude, by choosing a high valued λ, gradient descent is going to select a θ_j that keeps the cost function $J(\theta_0, \theta_1, \ldots, \theta_n)$ low, thus giving more weight to lower degree terms over those of a higher degree.

1.2.2 Artificial neural networks background

ANN are an ML technique that replicate, in a very simplified manner, the way neurons in a human brain interconnect to each other to produce solutions to given problems [19–23]. Although they can be used for a wide spectrum of problems, normally they are used to produce a logistic output, i.e., the labels are a set of discrete numbers (binary, integers from 1 to 10, etc.). For that reason, the output function to fit $h(\theta)$ normally takes the form of a sigmoid that returns values between 0 and 1 ($g(z) \in [0,1]$). In ANN the goal is to interconnect the features as much as possible to produce an output function $h(\theta)$ with strong non-linearities capable of describing and predicting complex behaviors. Specifically, features are combined using parameters $\theta_i^{(j)}$ to produce several intermediate functions, called activation functions. Each time this process is performed a new layer is created. This process is repeated iteratively as much as desired until one final function is obtained. This final function is the output function $h_\theta(x)$ that provides the predictions on the labels. Therefore, an ANN is defined by the:

- Number of features or input variables
- Number of layers
- Number of units in each layer
- Propagation function (typically, a sigmoid function $g(z)$).

The optimization of the parameters of an ANN follows the same principles explained in the previous section, with the following key variants:

- The set of parameters θ_i that convert features into activation functions and the activation functions $a_i^{(j)}$ in the final output function $h(\theta)$ form matrixes $\Theta^{(j)}$, instead of vectors as in the previous example for MPR.
- The cost function $J(\Theta)$ and its partial derivatives are different due to the application of the sigmoid function $g(z)$.
- Due to the increased complexity of the computation of the partial derivatives of the cost function and the number of computations needed, advanced optimization methods are required. Those methods are normally implemented in built-in functions in most common math coding languages, like MATLAB 2021 and Python 3.9.5.

1.2.3 Random forest regression background

Random forests are an ensemble learning method for classification and regression that operates by constructing many decision trees. For the problem at hand, the regression capabilities that consist of making a prediction based on the average of the prediction of the individual trees was employed [24, 25]. Each tree gets a random sample of the data with replacement, a process known as *bagging*, and splits it using the pre-established features until all the data is separated by classes. At each node the tree will ask: *What feature will allow me to split the observations at hand in a way that the resulting groups are as different from each other as possible (and the members of each resulting subgroup are as similar to each other as possible)?* The idea behind the concept of random forest is that a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. The built-in functions that perform RFR are quite easy to apply to a different set of problems and often provide predictions with good accuracy when the training sample is not big enough to apply other ML techniques.

1.3 Objective

The objective of this project was to develop a ML model to predict the DoA, as defined in Eq. 7, of 100% RAP samples using the input variables compaction temperature, air voids and Indirect Tensile Strength (ITS) at 25°C.

$$ \text{DoA (\% max ITS)} = 100 \cdot \frac{\text{ITS}(\text{RAP}_i, T\ ^{\circ}\text{C})}{\text{ITS}(\text{RAP}_i, \text{max})} \qquad (7) $$

where $\text{ITS}(\text{RAP}_i, T\ ^{\circ}\text{C})$ was the average ITS value for five specimens of RAP sample RAP_i compacted at temperature T and $\text{ITS}(\text{RAP}_i, \text{max})$ was the maximum average ITS of all compaction temperatures tested for the RAP sample RAP_i.

Such a model would increase the information to extract from a simple InDirect Tensile test (IDT) regarding the binder activity of a RAP sample. In order to obtain the DoA, Eq. 7, of a RAP sample, several IDT tests have to be performed at several different temperatures. A model that predicts DoA with just one value of ITS at one temperature would reduce the amount of testing required to that end.

2 Materials and methods

2.1 Testing

A total of 17 laboratories collaborated in this project, following a protocol designed by the leaders of Task Group 5 of the RILEM Technical Committee 264 on Reclaimed Asphalt Pavement. The procedure carried out by each laboratory is briefly summarized as follows. Each laboratory collected one or more samples of RAP that were used to manufacture 100% RAP cylindrical specimens by heating the source material to at least three of the five temperatures proposed (70 °C, 100 °C, 140 °C, 170 °C and 190 °C). The manufacturing and testing procedures are detailed in the following steps:

1. The RAP was dried in an oven at 40 °C for 48 h.
2. The material was properly selected using a riffle box to obtain a sample.
3. The RAP sample was pre-conditioned in the oven for 4 h (prior to mixing), at the desired temperatures (70, 100, 140, 170 and 190 °C).
4. The samples were mechanically or hand mixed, while controlling the temperature.
5. The specimens were compacted using a Marshall compactor, 50 blows each side of the sample.
6. The specimens obtained were tested for air void content and ITS.

Each laboratory had the possibility of using their common standard for those two tests. As a result, 5 different standards were used to determine air voids content [26–30] and 3 for ITS [31–33]. The sizes of these specimens were around 100 mm in diameter and 63.5 mm in height. Finally, for each RAP sample at different compaction temperatures, the DoA parameter was obtained as a function of the ITS.

The data analysis presented in this paper comprised a total of 32 RAP samples tested by 17 laboratories. Each data sample consisted of a four-dimensional vector, where the first three components were the features compaction temperature, air voids and ITS and the fourth component was the label DoA(% max. ITS). A total of 144 data points were analyzed.

2.2 Feature selection

This section describes the motivation behind the selection of compaction temperature, air voids and ITS as the features employed to predict DoA. As a first exploratory data analysis, a correlation heatmap, Fig. 1, was composed using features that were available from laboratory testing and that were considered easy to obtain for future researchers that may consider using the model.

As derived from Fig. 1, binder content showed low correlation with all other features. This might be caused by the extraction process in laboratory or due to the fact that many RAP samples lacked this data. For that reason, binder content was excluded from the model training.

Regarding density, it showed reasonable good correlation with DoA, but its high correlation with air voids and the fact that air voids showed higher correlation with DoA led to discard this variable from the model training as well.

2.3 Multivariate polinomial regression implementation

Following the theory introduced previously, an MPR with regularization was implemented in MATLAB 2021a. A 6th degree polynomial, which expanded the three initial features (compaction temperature, air voids and ITS) to a total of 82 polynomial features was employed. For that reason, it was necessary to implement regularization to avoid overfitting. The data set used to implement the model consisted of a coma-separated-values (.csv) file with four columns, where the first three corresponded to the features compaction temperature in °C, air voids in % in mixture volume and ITS in MPa, in that order. The fourth column contained the DoA in % of the maximum ITS (DoA (% max. ITS)) for that RAP sample computed using Eq. 7. The main objective was to train a model to predict the DoA (% max. ITS) using compaction temperature, air voids and ITS. The first step of the code was to randomize the rows of the data file to separate blocks of data coming from the same RAP sample. The next step was to split the data into three sets, namely training, validation and test. The training set was established as 60% first data points of
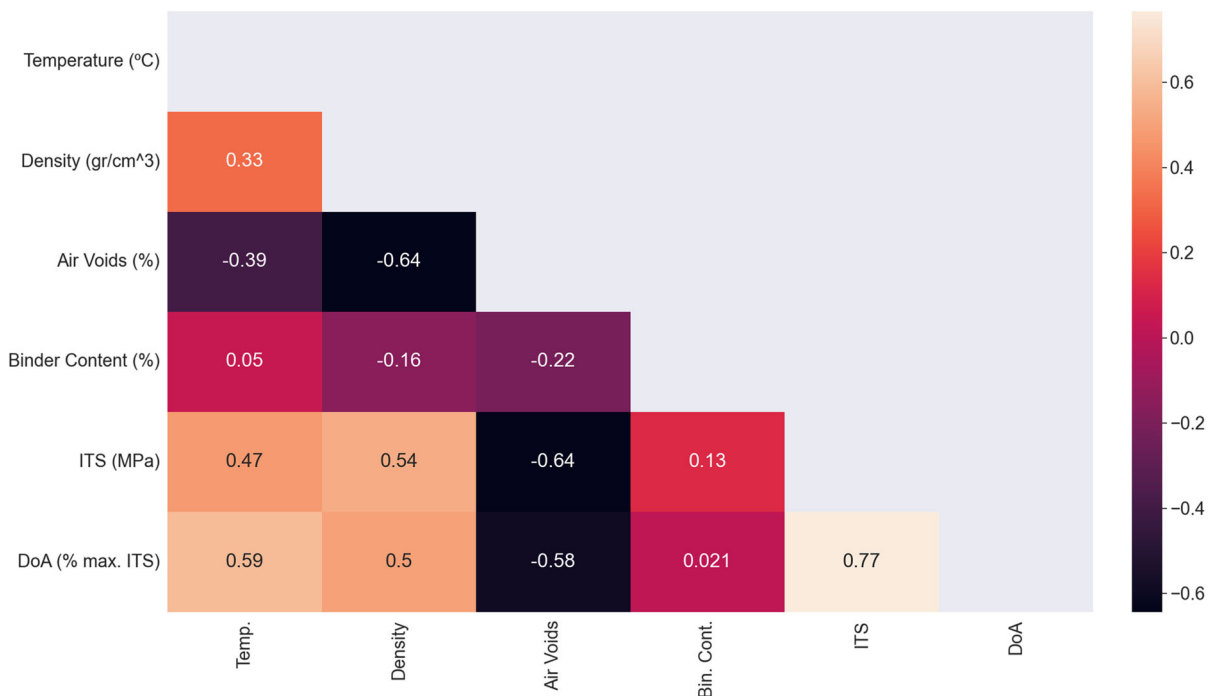

Fig. 1 Correlation heatmap between available features

the randomized data file and the test and validation set 40% (20% each). The training set was used to train the model of the validation set to find the optimum value for the regularization parameter, and the test set to measure the precision of the model. Precisely, the randomized data file contained 144 samples, the first 86 composed the train set, the next 29 the validation set and the final 29 the test set. The next step consisted of expanding the three features into a 6th degree polynomial. A tailored function was written in MATLAB 2021 code to perform this operation. The three original features resulted in 82 new features containing terms up to the 6th power. The main objective of this operation was to build an output function $h_\theta(x_i)$ complex large enough to fit complex data relations. In order to apply Gradient Descent, it is helpful to have features of similar magnitude, having expanded the original features to high degree polynomials that was not the case. Therefore, the 82 features were normalized using a tailored function to that end. Precisely, the normalization was done using the average x_i and the standard deviation σ_i for each of the 82 features, using only the training set.

$$\bar{x}_i^{(m)} = \frac{x_i^{(m)} - \bar{x}_i}{\sigma_i} \tag{8}$$

where $\bar{x}_i^{(m)}$ is the normalized value on the data sample m for feature i, and $x_i^{(m)}$ is the unnormalized value on the data sample m for feature i. The following step was to write an iterative loop to compute the cost $J(\theta)$ with regularization (Eq. 6), the gradient of the cost for each θ_i (Eq. 4) and update accordingly the θ_i parameters using Eq. 3. Regularization was mandatory since using a high degree polynomial function implies a high risk of overfitting for the reasons explained in Sect. 1.2.1. The step parameter for Gradient Descent α was set to 1.0 and the number of iterations was limited to 500,000. However, the amount of regularization depends on the value of the regularization parameter λ, which is completely arbitrary. If lambda is too low, or 0, the optimized $h_\theta(x_i)$ overfits the training set and fails to predict accurately the test set (high variance). If lambda is too high, all terms on $\theta_{(1..i)}$ are minimized by the gradient descent algorithm. As a consequence, the only non-zero term remaining is θ_0, which multiplies the bias term $x_{(0)} \equiv 1$, producing a constant output function $h_\theta(x_i) = \theta_0$ and, therefore, underfitting the

data (high bias). The model's performance was measured on the test set using Eq. 9:

$$h_\theta(X_{\text{test}}) = \theta^T X_{\text{test}} \tag{9}$$

where θ^T is the transpose vector of 82 fitting parameters and X_{test} is a matrix of 29 rows (data points) and 82 columns (features). Since the data was split randomly into three sets, different distributions of data points on training, validation and test sets may yield different models with different precision. For that reason, a last cross-validation exercise was necessary, which consisted of training 50 different models using 50 different random data splits. The average precision score of those 50 different models was the final expected precision of the model. The definitive model was obtained by training the MPR with the whole data set.

2.4 Artificial neural network implementation

The architecture of the ANN designed to predict the DoA (% max. ITS) is shown in Fig. 2.

The selection of this architecture was based on previous trials that consisted of training ANNs with few iterations (short computing time) using different quantities of hidden layers and hidden units on all the data available. The architecture that obtained the least mean absolute error (MAE) on the predictions was composed of 4 layers: the input layer had 3 features (compaction temperature, air voids and ITS), the 2 hidden layers had 5 hidden units each and the final layer produced 1 final output function. The activation function employed to propagate the features into the output function was a sigmoid, Eq. 10:

$$g(z) = \frac{1}{1 + e^{-z}} \tag{10}$$

The output of $g(z)$ is a value between 0 and 1, which fitted perfectly the behavior of the DoA (% max. ITS) which comprised values between 0 and 100, therefore a simple escalation of the final output function gave the predictions on the labels directly. The procedure to train, tune and validate the results of the model was the same followed for the MPR model calibration. The data set was split into three sets, namely training, test and validation set. In this case, it was not necessary to normalize the features, since the sigmoidal activation function performed that process implicitly. The
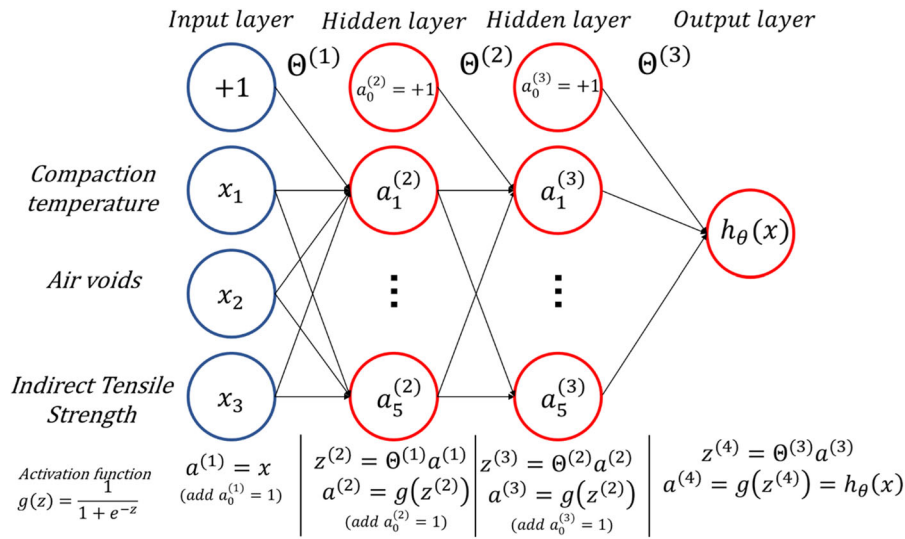
Fig. 2 ANN architecture chosen for the current problem

optimization of the parameters was performed using the "fmincg" built-in function from the MATLAB 2021a library. This built-in function only requires the cost function and its gradients with respect to θ_i to perform the optimization. The expression for the cost function of the ANN architecture chosen for this problem is shown in Eq. 11.

$$
\begin{aligned}
J(\Theta) = \frac{1}{m} \sum_{i=1}^{m} & \left[-y^{(i)} \log \left(h_\theta \left(x^{(i)} \right) \right) \right. \\
& \left. - (1 - y^{(i)}) \log \left(1 - h_\theta \left(x^{(i)} \right) \right) \right] \cdots \\
& + \frac{\lambda}{2m} \left[\sum_{j=1}^{5} \sum_{k=1}^{3} \left(\Theta_{jk}^{(1)} \right)^2 + \sum_{j=1}^{5} \sum_{k=1}^{5} \left(\Theta_{jk}^{(2)} \right)^2 \right. \\
& \left. + \sum_{j=1}^{1} \sum_{k=1}^{5} \left(\Theta_{jk}^{(3)} \right)^2 \right],
\end{aligned}
$$

(11)

where m is the number of training examples and $\Theta_{jk}^{(l)}$ are matrixes of the parameters that propagate the features into the output function $h_\theta(x)$. Given one training example, the forward propagation is done by following Eqs. 12–18:

$$
a^{(1)} = x \tag{12}
$$

$$
z^{(2)} = \Theta^{(1)} a^{(1)} \tag{13}
$$

$$
a^{(2)} = g\left(z^{(2)} \right) \left(\text{add } a_0^{(2)} = 1 \right) \tag{14}
$$

$$
z^{(3)} = \Theta^{(2)} a^{(2)} \tag{15}
$$

$$
a^{(3)} = g\left(z^{(3)} \right) \left(\text{add } a_0^{(3)} = 1 \right) \tag{16}
$$

$$
z^{(4)} = \Theta^{(3)} a^{(3)} \tag{17}
$$

$$
a^{(4)} = h_\Theta(x) = g\left(z^{(4)} \right), \tag{18}
$$

where the superscripts in parentheses refer to the layers of the ANN and Θ^l are matrixes of the $s_{l+1} \times (s_l + 1)$ dimension where s_l is the number of hidden units in layer l.

This process provides the expression for $h_\Theta(x)$, Eq. 18, that has to be inserted in Eq. 11 to compute the cost $J(\Theta)$. The next thing required to apply built-in optimization methods are the gradients $\frac{\partial}{\partial \Theta_{ij}} J(\Theta)$. It can be proven mathematically that, if regularization is ignored, the gradients satisfy Eq. 19:

$$
\frac{\partial}{\partial \Theta_{ij}^l} J(\Theta) = a_j^{(l)} \delta_i^{(l+1)} \; for \; \lambda = 0, \tag{19}
$$

where $\delta_i^{(l+1)}$ is the error of node i in layer $l + 1$. These errors are obtained by applying what is called a back

propagation algorithm, described in the next set of Eqs. (20–23).

$$\delta^{(4)} = a^{(4)} - y \tag{20}$$

$$\delta^{(3)} = \left(\Theta^{(3)}\right)^T \delta^{(4)} \cdot *g'\left(z^{(3)}\right) \tag{21}$$

$$\delta^{(2)} = \left(\Theta^{(2)}\right)^T \delta^{(3)} \cdot *g'\left(z^{(2)}\right) \tag{22}$$

$$g'\left(z^{(l)}\right) = a^{(l)} \cdot *(1 - a^{(l)}), \tag{23}$$

where .* means element-wise multiplication of the two vectors. Then, in machine code, $\Delta_{ij}^{(l)}$ matrixes are defined initially as $\Delta_{ij}^{(l)} = 0$ for all l, i, j, and the next quantities are computed to finally obtain the gradients (24–27).

$$\Delta^{(l)} := \Delta^{(l)} + \delta^{(l+1)}\left(a^{(l)}\right)^T \tag{24}$$

$$D_{ij}^{(l)} := \frac{1}{m}\Delta_{ij}^{(l)} + \lambda\Theta_{ij}^{(l)} \; if \; j \neq 0 \tag{25}$$

$$D_{ij}^{(l)} := \frac{1}{m}\Delta_{ij}^{(l)} \; if \; j = 0 \tag{26}$$

$$\frac{\partial}{\partial\Theta_{ij}^{(l)}}J(\Theta) = D_{ij}^{(l)}, \tag{27}$$

where := means update the variable with the previous value of the variable plus a new computation. It turns out that adding regularization to the gradients is as simple as adding the matrixes $\Theta_{ij}^{(l)}$ multiplied by the regularization parameter lambda λ in all terms but the ones corresponding to the bias terms ($j = 0$).

As in the case of the MPR model, regularization is needed to avoid overfitting of the model to the train set. The second term in Eq. 11 is responsible for regularizing the model and prevents overfitting. Again, the value of the regularization parameter λ determines the bias or variance of the model. If λ is too high the model outputs a very simple (or even constant) function that fails to fit the data. Conversely, if λ too low the model overfits the training set but fails to predict successfully test data. For that reason, it was necessary to split the data into three sets: the training set was used to find the optimum $\Theta_{ij}^{(l)}$ parameters that fit the model, the validation set was used to measure the precision of the model with different λ values on a data set different from the training set, and finally the test set was used to measure the precision of the model with the definitive λ on a third different data set.

However, the optimum λ value, i.e. the one that led to the lowest MAE on the validation set, also depended on how the data was split. For that reason, the whole process was written in one script that nested an iterative loop that ran through the 12 selected λ values inside another iterative loop that ran through 30 different random data splits. Therefore, a total of $30 \times 12 = 360$ ANN's were trained. The workflow of the script is shown the following scheme:

1. Do the following steps:

    a. Split data into three sets randomly
    b. Take the following steps: @@@
    c. Find optimum lambda that provides the lowest error on the validation set
    d. Compute error of ANN with optimum λ on the test set

2. Return to step **1** until 30 iterations are completed

This script returned 30 different optimum values of λ as well as 30 MAE values for training, validation and test sets. The definitive model was trained using the whole data set and the optimum value of λ.

2.5 Random forest regression implementation

The RFR was implemented using the built-in *RandomForestRegressor* function from the scikit-learn Python package. The function automatically fitted the regression to the supplied data. However, the *RandomForestRegressor* function allows the modification of several parameters that control the fitting. These model fit parameters are known as hyperparameters, to distinguish them from the parameters that describe the model itself. The hyperparameters of the *RandomForestRegressor* function that were considered in the optimization were the following:

- bootstrap: whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.
- max_depth: the maximum depth of the tree.
- max_features: the number of features to consider when looking for the best split.

- min_samples_leaf: the minimum number of samples required to be at a leaf node.
- min_samples_split: the minimum number of samples required to split an internal node.
- n_estimators: the number of trees in the forest.

Each of these hyperparameters must be optimized, which implies the generation of multiple models, each with a different partition of the data and a different combination of hyperparameters. The standard procedure to validate the model was as follows:

1. Split data into three sets, namely training, validation and test sets.
2. Train several models using the training set and a different set of values of the hyperparameters considered.
3. Evaluate the precision of each model on the validation set.
4. Find the values of the hyperparameters that maximize precision on the validation set.
5. Evaluate the precision of the model with the best hyperparameters on the test set.

The *RandomizedSearchCV* function from the scikit-learn package was used to adjust hyperparameters automatically. Given a data set and ranges for all parameters, multiple fits were made by modifying the training and validation data sets, as shown in Fig. 3, as well as the hyperparameter combinations in a random manner. Since *RandomizedSearchCV* automatically performed the cross-validation data split on training/validation sets, the data fed to the function was 80% of all data, saving 20% (test set) for final evaluation of the definitive model. Hence, the set of hyperparameters that produced the model with the best precision on the validation sets were determined.

In this case, $K = 3$ was used, that is, the *RandomizedSearchCV* function fitted models with different hyperparameters using 3 different splits in training and validation data sets. In total, 729 models were fitted. This function narrowed down the values of the hyperparameters that obtained the best precision in the prediction of the labels of the different validation sets. However, this function does not analyze all possible hyperparameter combinations, but rather randomly combines the hyperparameters. To fine-tune the final hyperparameters, the *GridSearchCV* function of the scikit-learn package was used which, given a set of hyperparameter ranges, trains models with all possible combinations of these. The values of the hyperparameters in this second search were limited to those close to the optimal hyperparameters obtained in the previous search carried out with *RandomizedSearchCV*.

Once the optimum hyperparameters were defined, the precision of the model was evaluated on the test set. Then 50 different models were trained with 50 different random data splits and their precision score was averaged to obtain an estimate of the final precision of the definitive model. The definitive model was obtained by training the RFR with all the data set and the optimum hyperparameters.

3 Results

This section is divided into three subsections, each one describing and discussing the results of each of the three ML techniques employed to analyze the data.

Fig. 3 Graphic example of cross-validation using $K = 3$ data splits

3.1 Multivariate polynomial regression results

Figure 4 shows the evolution of the cost function on training and validation sets with the change in the regularization parameter λ.

The regularization analysis shown in Fig. 4, led to an optimum regularization parameter value of $\lambda = 0.08$. Since this value was chosen by minimizing error on the validation labels prediction, the performance of the model could not be measured on this same set.

Therefore, the model's performance was measured on the test set using Eq. 9. For this concrete random data split, the model predicted the DoA with a 9.7% of max ITS MAE.

However, the model was not yet validated, since another issue regarding data split should be addressed. Since the data was split randomly into three sets, different distributions of data points on training, validation and test sets may yield different models with different precisions. For that reason, it was necessary to perform a cross-validation exercise which consisted of training 50 different models using 50 different random data splits.

The average value of the MAE on the predictions of the test set labels was 12.2% max. ITS, taking all 50 different models, with a standard deviation of 1.1% max. ITS. The standard deviation of all DoA (% max. ITS) values on the original data set was 31%.

To obtain the definitive model parameters (namely θ_i, $\overline{x}_i$ and σ_i) that should be used for future predictions using new data, a new MPR was trained using the whole data set available (144 data points). The resulting model fitted the whole data set with a MAE of 9% max. ITS.

3.2 Artificial neural network results

Table 1 shows the MAE on train, validation and test sets for 8 different values of the regularization parameter λ and 30 different data splits. Figure 5 shows the evolution of the average value of the MAE for each lambda tested in each of the 30 random data split iterations.

Values of λ over 0.64 did not get optimum precision in any of the 30 iterations, meaning that over this value regularization is too high for the function $h_\Theta(x)$ to fit the data. The most occurring optimum λ was 0.01 with 7 occurrences, but the value with the lowest average MAE on the test set among the 30 random data split iterations occurred for 0.08. For that reason, the chosen λ value for the definitive fitting was 0.08.

The average MAE on the 30 random train sets with $\lambda = 0.08$ was 9.4% max.ITS with a standard deviation of 0.9% max.ITS, while the average MAE on the test set was 12.8% max. ITS with a standard deviation of 2.6% max.ITS. Therefore, the precision obtained for this model was sligthy worse than the one obtained on
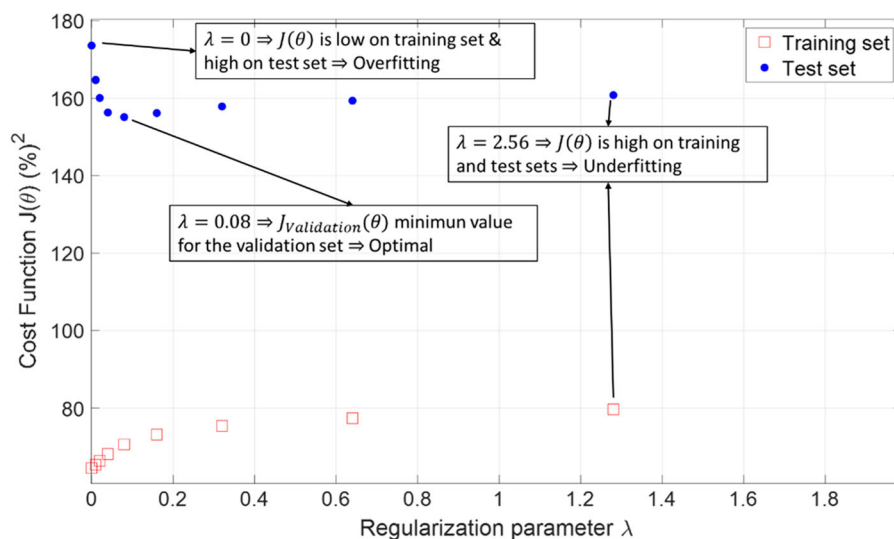

Fig. 4 Cost function for training and validation sets as a function of the regularization parameter λ

Table 1 MAE in DoA for the different random data splits and λ values

λ	Times optimum*	Average MAE in DoA (% max. ITS)					
		Train set	SD	Validation set	SD	Test set	SD
0	3	6.8	1	14.9	3.2	13.4	3.5
0.01	7	7.5	1	13.7	2.4	12.3	2.4
0.02	5	8.4	0.9	13.5	2.4	12.3	2.7
0.04	6	8.8	0.9	13.6	2	12	2.3
0.08	5	9.4	0.9	13.5	1.9	12.8	2.6
0.16	0	10.4	0.8	14.1	2	12.4	2.2
0.32	3	11.7	0.8	14.7	1.8	13.2	1.9
0.64	1	13.4	1	15.7	1.8	14.3	1.8

*Times the value was optimum (out of 30)
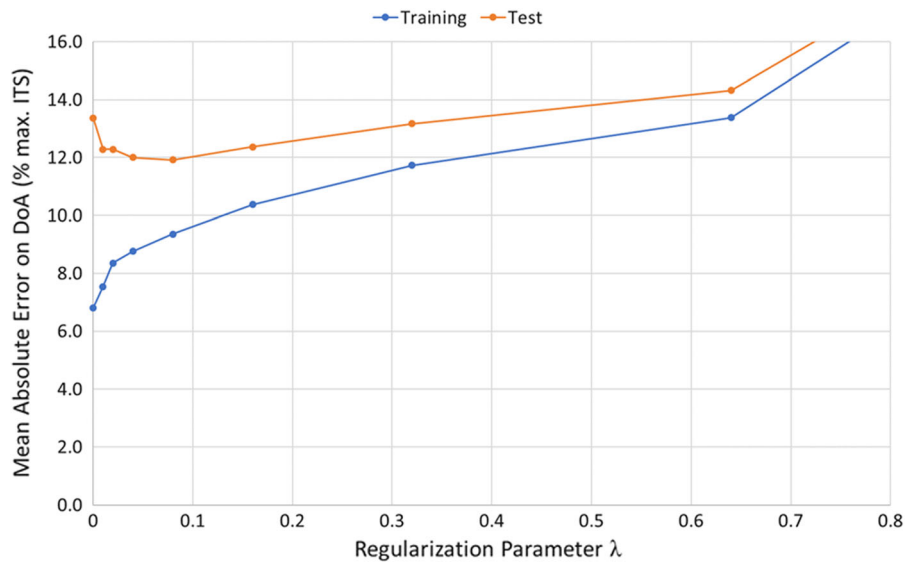
Fig. 5 Evolution of the MAE on the training and test set with the value of the regularization parameter λ

the MPR model, although the difference was not statistically significant. Finally, the definitive parameters for this model to be applied in the future to new data sets were obtained by training the ANN with the whole data set available.

The ANN required a total of 56 parameters to perform similarly to the MPR which needed 82 parameters plus 164 values of average and standard deviation on each of the features to perform normalization. The computational time needed to train the ANN was much higher than the one needed to fit the MPR, but once the ANN was trained its application to a data set was immediate. The parameters of the definitive ANN are shown below.

$$\Theta^{(1)} = \begin{pmatrix} -14.6903 & 0.0862 & 0.0455 & -2.2005 \\ -3.9353 & 0.1243 & -0.4212 & 0.4260 \\ -3.2587 & -0.0279 & 0.2689 & -0.8007 \\ 13.1888 & -0.1614 & 0.0601 & 0.8350 \\ -10.1158 & 0.0703 & 0.1225 & -0.2690 \end{pmatrix}$$

$$\Theta^{(2)} = \begin{pmatrix} -0.29972 & -1.44842 & 0.461003 & -0.57908 & -0.65696 & 0.599678 \\ -0.29956 & -1.44815 & 0.460914 & -0.57903 & -0.65684 & 0.599496 \\ -0.29810 & -1.44564 & 0.460096 & -0.57859 & -0.65574 & 0.597836 \\ -0.28162 & -1.4154 & 0.450492 & -0.57301 & -0.6426 & 0.578534 \\ -0.29944 & -1.44794 & 0.460847 & -0.57899 & -0.65675 & 0.599361 \end{pmatrix}$$

$$\Theta^{(3)} = \begin{pmatrix} -3.75572 & 1.944483 & 1.944088 & 1.940485 & 1.897315 & 1.943793 \end{pmatrix}$$

3.3 Random forest regression results

The hyperparameters that obtained the best precision were those shown in Table 2.

Once the optimal hyperparameters were obtained, a model was trained using these optimized values and 80% of total data set previously fed to *RandomizedSearchCV* and *GridSearchCV* functions. The precision of the model was assessed using the test data set, which until now had not come into contact with model optimization, to ensure the validity of the predictions.

Figure 6 compares graphically the real DoA values with the DoA predictions on the test set, for the first model trained using the optimum hyperparameters from Table 2. The DoA MAE, precision and the correlation coefficient on the test were 11.7% max.-ITS, 81% and 0.77, respectively.

As mentioned in previous sections, different splits of the data produce different models with different accuracies. Therefore, as the last validation step, 50 models were trained with 50 different training / test data splits.

After cross-validation with 50 different data splits, the scoring metrics shown in Table 3 were obtained.

3.4 Models' score comparison summary

This section sumarizes and compares the precision of the three types of models trained, in terms of the average MAE, precission and correlation coeficient on DoA predicton, Table 4.

The average precision of the three models were very similar, almost indistinguishable. This may indicate the consistency of the models and also set a minimum value for the error on the predictions. That limit appears to be just over 12% max. ITS on DoA predicted values. In terms of accuracy, the models produced predictions of DoA with around 72% precision.

In addition of the accuracy, these models provided also metrics on the importance of each of the features on the labels computation. Regarding the RFR model,

the feature importance was extracted using a built-in method of the function *RandomForestRegressor*, while for the ANN and the MPR feature importance was measured as the relative increase in MAE on the train set when the given feature was shuffled randomly along the data set (Table 5).

The compaction temperature was the variable that most influenced the predictions. A higher compaction temperature favors the mobility and activity of the RAP binder by reducing its viscosity. Also, the method to obtain DoA, Eq. 7, might have influenced this result. The second most influential variable was the ITS value, this may be due to the fact that the calculation of the DoA (% max. ITS) depends directly on this variable. Finally, the importance of the air void content was estimated around 10%, according to the models, however this feature was the one with the highest correlation with ITS, Fig. 1.

4 Conclusion

Three different machine learning (ML) techniques were applied to predict the degree of binder activity (DoA) in terms of the maximum value of indirect tensile strength (ITS) for a given sample of Reclaimed Asphalt Pavement (RAP), using the compaction temperature, the air voids and the ITS of specimens manufactured with 100% RAP.

The three ML methodologies evaluated were multivariate polynomial regression (MPR), artificial neural networks (ANN) and random forest regression (RFR). Out of the three techniques, the simplest and easiest to understand was the MPR, followed by the ANN. However, the easiest to implement and the one requiring less computational time was the RFR. Precision-wise, the three methodologies provided models with similar accuracies.

The definitive models, based on an MPR, an ANN and an RFR, produced predictions on the DoA with $\pm 12.2\%$ max. ITS, $\pm 12.8\%$ max. ITS and $\pm 12.3\%$ max. ITS, respectively. Using any of these models, anyone who needs to evaluate a RAP sample can perform a fairly good prediction of the quantity of binder that is going to activate in the RAP when compacted at a certain temperature, just by measuring air voids and ITS at 25 °C on 100% RAP specimens at only one compaction temperature. Since the source data used to train the model came from different

Table 2 Optimal values for the hyperparameters of the random forest model

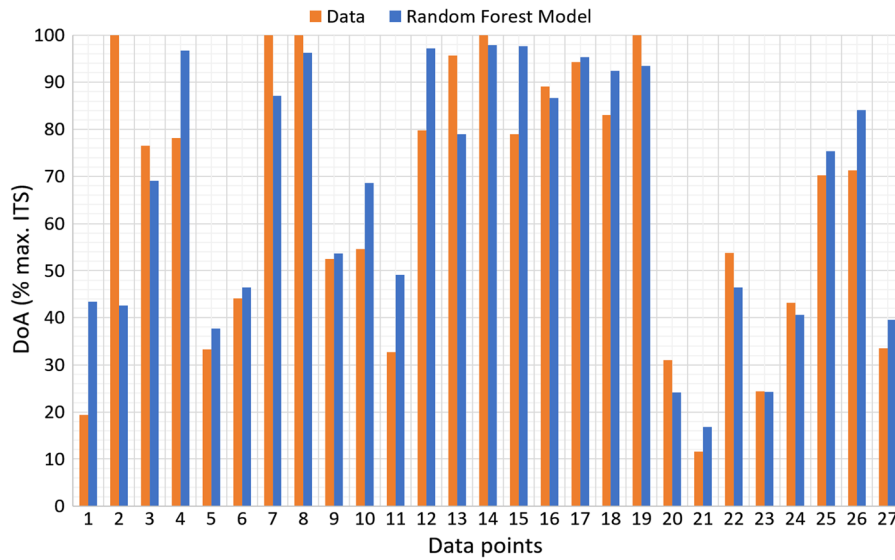| Hyperparameter	Value
bootstrap	TRUE
max_depth	40
max_features	sqrt
min_samples_leaf	2
min_samples_split	7
n_estimators	1000

Fig. 6 Predictions and real values for the 27 test points

Table 3 Average evaluation metrics for 50 models with different data splits in training and test sets

Metric	Mean	SD
Avg. MAE on DoA	12.3% max. ITS	2.6% max. ITS
Avg. precision	73.8%	11.8%
Avg. correlation coefficient	0.71	0.08

Table 4 Scores for each kind of model

Metric	RFR	ANN	MPR
Avg. MAE on DoA	12.3% max. ITS	12.8% max. ITS	12.2% max. ITS
Avg. precision	73.8%	71.0%	72.2%
Avg. correlation coef.	0.71	0.71	0.71

Table 5 Importance of each feature on each of the definitive models

Feature	Importance (%)		
	RFR	ANN	MPR
Compaction temperature (°C)	66	72	68
ITS at 25 °C (MPa)	23	19	22
Air voids (%)	10	9	10

laboratory equipment, different air void determination standards and different ITS test standards, these models should be able to generalize. These models can also be used to reduce the amount of testing required to find the optimum compaction temperature for a RAP sample. Running these models on the compaction temperature, air voids and ITS can help in narrowing down the compaction temperature that provides the maximum ITS for the material.

Feature importance analysis showed that compaction temperature was the feature that most influenced the predictions of the models. However, air voids it is known to influence strongly on ITS, hence this result might be caused by the way DoA was defined in Eq. 7.

Finally, these models were trained using exclusively RAP samples and their validity is limited to that

material. However, the concept of finding the compaction temperature that provides the maximum ITS, or the expected percentage of maximum ITS for a given compaction temperature, can be useful for any kind of asphalt mixture.

5 Supplementary materials

The following files are available online at https://github.com/ramonbotella/RAP-data-RILEM-TCRAP-TG5: MPR and ANN Matlab live scripts, RFR Python scripts and the RAP data set.

Declarations

References

1. Kennedy TW, Tam WO, Solaimanian M (1998) Optimizing use of reclaimed asphalt pavement with the Superpave system. J Assoc Asphalt Paving Technol 67:311–333
2. McDaniel R, Anderson RM (2001) Recommended use of reclaimed asphalt pavement in the superpave mix design method: technician's manual. Recomm use reclaimed asph pavement superpave mix des method tech man
3. Oliver JWH (2001) The influence of the binder in RAP on recycled asphalt properties. Road Mater Pavement Des 2(3):311–325. https://doi.org/10.1080/14680629.2001.9689906
4. Stephens JE, Mahoney J, Dippold C (2001) Determination of the PG binder grade to use in a rap mix. Connecticut Department of Transportation
5. Huang B, Li G, Vukosavljevic D, Shu X, Egan B (2005) Laboratory investigation of mixing hot-mix asphalt with reclaimed asphalt pavement. Transp Res Rec J Transp Res Board 1929:37–45. https://doi.org/10.3141/1929-05
6. Al-Qadi IL, Elseifi M, Carpenter SH (2007) Reclaimed asphalt pavement—a literature review (FHWA-ICT-07-001). Tech. rep. http://hdl.handle.net/2142/46007
7. Carpenter SH, Wolosick JR (1980) Modifier influence in the characterization of hot-mix recycled material. Transp Res Rec 777:15–22
8. Noureldin AS, Wood LE (1987) Rejuvenator diffusion in binder film for hot-mix recycled asphalt pavement. Transp Res Rec 1115:51–61
9. Menegusso Pires G, Lo Presti D, Airey GD (2021) A practical approach to estimate the degree of binder activity

of reclaimed asphalt materials. Road Mater Pavement Des 22(5):1093–1116. https://doi.org/10.1080/14680629.2019.1663244

10. Lo Presti D, Vasconcelos K, Orešković M, Pires GM, Bressi S (2020) On the degree of binder activity of reclaimed asphalt and degree of blending with recycling agents. Road Mater Pavement Des 21(8):2071–2090. https://doi.org/10.1080/14680629.2019.1607537

11. Kaseer F, Arámbula-Mercado E, Martin AE (2019) A method to quantify reclaimed asphalt pavement binder availability (effective RAP binder) in recycled asphalt mixes. Transp Res Rec J Transp Res Board 2673(1):205–216. https://doi.org/10.1177/0361198118821366

12. Zhang K, Wen H, Hobbs A (2015) Laboratory tests and numerical simulations of mixing superheated virgin aggregate with reclaimed asphalt pavement materials. Transp Res Rec J Transp Res Board 2506(1):62–71. https://doi.org/10.3141/2506-07

13. Bressi S, Cavalli MC, Partl MN, Tebaldi G, Dumont AG, Poulikakos LD (2015) Particle clustering phenomena in hot asphalt mixtures with high content of reclaimed asphalt pavements. Constr Build Mater 100:207–217. https://doi.org/10.1016/j.conbuildmat.2015.09.052

14. Ding Y, Huang B, Shu X (2018) Blending efficiency evaluation of plant asphalt mixtures using fluorescence microscopy. Constr Build Mater 161:461–467. https://doi.org/10.1016/j.conbuildmat.2017.11.138

15. Vassaux S, Gaudefroy V, Boulangé L, Soro LJ, Pévère A, Michelet A, Barragan-Montero V, Mouillet V (2018) Study of remobilization phenomena at reclaimed asphalt binder/virgin binder interphases for recycled asphalt mixtures using novel microscopic methodologies. Constr Build Mater 165:846–858. https://doi.org/10.1016/j.conbuildmat.2018.01.055

16. Zhao S, Huang B, Shu X, Woods ME (2015) Quantitative characterization of binder blending. Transp Res Rec J Transp Res Board 2506(1):72–80. https://doi.org/10.3141/2506-08

17. Stimilli A, Virgili A, Canestrari F (2015) New method to estimate the "re-activated'' binder amount in recycled hot-mix asphalt. Road Mater Pavement Des 16(sup1):442–459. https://doi.org/10.1080/14680629.2015.1029678

18. Orešković M, Menegusso Pires G, Bressi S, Vasconcelos K, Lo Presti D (2020) Quantitative assessment of the parameters linked to the blending between reclaimed asphalt binder and recycling agent: a literature review. Constr Build Mater 234:117323. https://doi.org/10.1016/j.conbuildmat.2019.117323

19. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5(4):115–133. https://doi.org/10.1007/BF02478259

20. Ivakhnenko AG, Lapa VG (1967) Cybernetics and forecasting techniques. Elsevier, Amsterdam

21. Ivakhnenko AG, Lapa VG (1973) Cybernetic predicting devices. New York

22. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117. https://doi.org/10.1016/j.neunet.2014.09.003

23. Specht L, Khatchatourian O (2014) Application of artificial intelligence to modelling asphalt-rubber viscosity. Int J Pavement Eng 15(9):799–809. https://doi.org/10.1080/10298436.2014.893316

24. Kleinberg EM (1990) Stochastic discrimination. Ann Math Artif Intell 1(1–4):207–239. https://doi.org/10.1007/BF01531079

25. Breiman L (2001) Using iterated bagging to debias regressions. Mach Learn. https://doi.org/10.1023/A:1017934522171

26. EN:12697–6:2020 (2020) Bituminous mixtures. Test methods, Determination of bulk density of bituminous specimens

27. UNE-EN:12697-8:2018 (2018) Bituminous mixtures—test methods—part 8: Determination of void characteristics of bituminous specimens

28. ASTM:D1188-07(2015) (2015) Standard test method for bulk specific gravity and density of compacted bituminous mixtures using coated samples

29. ASTM:D3203/D3203M-17 (2014) Standard test method for percent air voids in compacted asphalt mixtures. In: 2017 30 "AASHTO T 269 2014". Standard method of test for percent air voids in compacted dense and open asphalt mixtures

30. AASHTO:T269:2014 (2014) Standard method of test for percent air voids in compacted dense and open asphalt mixtures

31. UNE-EN:12697-23:2017 (2017) Bituminous mixtures—test methods—part 23: determination of the indirect tensile strength of bituminous specimens

32. ASTM:D6931-17 (2017) Standard test method for indirect tensile (IDT) strength of asphalt mixtures

33. DNIT:135/2018-ME (2018) Pavimentação asfáltica - Misturas asfálticas Determinação do módulo de resiliência—Método de ensaio