



Development of a Machine Learning (ML)-Based Computational Model to Estimate the Engineering Properties of Portland Cement Concrete (PCC)

Rodrigo Polo-Mendoza^{1,2} · Gilberto Martinez-Arguelles² · Rita Peñabaena-Niebles³ · Jose Duque^{1,4}

Received: 7 December 2023 / Accepted: 23 January 2024
© The Author(s) 2024

Abstract

Portland cement concrete (PCC) is the construction material most used worldwide. Hence, its proper characterization is fundamental for the daily-basis engineering practice. Nonetheless, the experimental measurements of the PCC's engineering properties (i.e., Poisson's Ratio ν -, Elastic Modulus E -, Compressive Strength $ComS$ -, and Tensile Strength $TenS$ -) consume considerable amounts of time and financial resources. Therefore, the development of high-precision indirect methods is fundamental. Accordingly, this research proposes a computational model based on deep neural networks (DNNs) to simultaneously predict the ν , E , $ComS$, and $TenS$. For this purpose, the Long-Term Pavement Performance database was employed as the data source. In this regard, the mix design parameters of the PCC are adopted as input variables. The performance of the DNN model was evaluated with 1:1 lines, goodness-of-fit parameters, Shapley additive explanations assessments, and running time analysis. The results demonstrated that the proposed DNN model exhibited an exactitude higher than 99.8%, with forecasting errors close to zero (0). Consequently, the machine learning-based computational model designed in this investigation is a helpful tool for estimating the PCC's engineering properties when laboratory tests are not attainable. Thus, the main novelty of this study is creating a robust model to determine the ν , E , $ComS$, and $TenS$ by solely considering the mix design parameters. Likewise, the central contribution to the state-of-the-art achieved by the present research effort is the public launch of the developed computational tool through an open-access GitHub repository, which can be utilized by engineers, designers, agencies, and other stakeholders.

Keywords Computational modelling · Concrete structures · Construction materials · Deep neural networks · Machine learning · Portland cement concrete

1 Introduction

Portland Cement Concrete (PCC) is the composite material produced by mixing coarse aggregates (i.e., gravel), fine aggregates (i.e., sand), Portland Cement (PC), water, and optionally admixtures (e.g., chemical additives, fly ash,

and steel slag) [1–3]. PCC is the world's most widely used construction material, mainly employed for constructing buildings and pavements [4–6]. Notoriously, the high consumption of time and financial resources for its proper experimental characterization implies that designers must resort to indirect methods [7–9]. However, the traditional mathematical models used for these purposes have been confirmed to be inaccurate and unreliable [10–12]. The preceding has led the estimation of PCC's engineering properties to focus on developing advanced computational models [13–15]. Within this field of research, Artificial Intelligence (AI)-based models stand out for their high precision and ability to adapt to new types of mixture designs not considered during the model creation stage [8, 14, 16].

Table 1 summarizes a literature review on estimating PCC's engineering properties using AI techniques. This table

✉ Rodrigo Polo-Mendoza
rpoloe@uninorte.edu.co

¹ Faculty of Science, Charles University, Prague, Czech Republic

² Department of Civil and Environmental Engineering, Universidad del Norte, Barranquilla, Colombia

³ Department of Industrial Engineering, Universidad del Norte, Barranquilla, Colombia

⁴ Department of Civil and Environmental, Universidad de la Costa, Barranquilla, Colombia



Table 1 Literature review on estimating PCC's engineering properties using AI techniques

References	Type of PCC	Number of datasets	Estimated properties	Computational models based on
[7]	Fly-ash/slag-based geopolymer concrete	156	ComS	DT, RF, SVM
[8]	Fly ash-based geopolymer concrete	151	ComS	MLP, SVM, XGB
[9]	Fly ash-based geopolymer concrete	263	ComS	ANN, ResNet
[13]	Geopolymer concrete	147	ComS	CBR, ETR, GBR
[14]	PCC modified with high calcium fly ash	56	Chloride resistance	RGM
[15]	Waste ceramic concrete with metallic and non-metallic fiber	33	ComS, TenS	ANN
[16]	Alkaline-activated slag concretes	181	ComS	ANN
[20]	Geopolymer concrete	371	ComS	BR, DT, GEP, RF
[21]	PCC modified with metakaolin	982 for ComS, 204 for TenS, 63 for FS		GEP, ANN, DT, RF
[22]	Ordinary PCC	324	ComS	ANN

BR bagging regressor, *CBR* catboost regressors, *DT* decision tree, *ETR* extra trees regressors, *FS* flexural strength, *GBR* gradient boosting regressors, *GEP* gene expression programming, *MLP* multilayer perceptron, *ResNet* deep residual network, *RGM* rule generation methods, *RF* random forest, *SVM* support vector machine

shows that most previous investigations focused on predicting Compressive Strength (ComS) and Tensile Strength (TenS). The preceding is expected since the ComS and TenS are the most relevant properties (regarding the characterization of the PCC's mechanical behaviour) to design structural elements and rigid pavements, respectively [17–19]. On the other hand, Table 1 also reveals a critical gap in the state-of-the-art, i.e., in none of the consulted research, Poisson's Ratio (ν) and Elastic Modulus (E) are considered output variables. In this regard, the aforementioned computational models did not take into account the strong relationship that exists between the resistance-related properties with the ν and E . In other words, previous case studies do not guarantee that for each evaluated dataset (i.e., a set of experimental data), the physical associations between ν , E , ComS, and TenS would be simultaneously maintained.

The present research aims to overcome the previously explained literature gap. Consequently, a computational model is proposed using the Long-Term Pavement Performance (LTPP) database as the data source [23]. Specifically, a Machine Learning (ML) technique called Artificial Neural Network (ANN) is employed for this purpose. From the LTPP database, experimental information from multiple PCC samples was extracted. The experimental information includes design variables and laboratory testing results (i.e., ν , E , ComS, and TenS). In turn, the PCC's design variables comprise the following parameters: volumetric air content (VairC), volumetric aggregate content (VaggC), volumetric PC content (VpcC), volumetric water content (VwC), gravimetric water-to-cement ratio (GwcR), and specific gravity (Gs). Notoriously, it is expected that the proposed ANN model can be used (by agencies, designers and the scientific

community) to forecast the PCC's engineering properties (ν , E , ComS, and TenS) when experimental measurements are not feasible.

Below is described the structure of the present manuscript. Initially, Sect. 2 describes the origin of the data used to develop the ML-based model. Next, Sect. 3 introduces the computational methods employed, and the basic concepts of ANNs are also explained. Then, Sect. 4 exhibits a deep discussion about the exactness and accuracy of the proposed computational model. It is important to highlight that the discussion section includes several SHapley Additive exPlanations (SHAP) assessments and running time analyses. Finally, Sect. 5 summarises the research and offers the main findings and conclusions achieved throughout the investigation.

2 Data Source

All the data employed in this investigation was extracted from the LTPP database. The LTPP's information management system has operated since 1988 and is managed by the Federal Highway Administration (i.e., part of the US Department of Transportation) [24–27]. The LTPP database comprises exhaustive information about design parameters, laboratory characterization of materials, climate variables, in-situ performance, and life-cycle behaviour on over 2500 pavement test sections in the USA and Canada [28–31].

With the aim of filtering the relevant information (for this study) from the LTPP database, it was decided to solely consider the experimental dataset that simultaneously contains information about VairC, VaggC, VpcC, VwC, GwcR, Gs, ν ,

Table 2 Basic statistical description of the adopted variables

Variable (Unit)	Minimum	Maximum	Mean	Median	Standard deviation	Kurtosis	Skewness
VairC (%)	3.100	12.200	7.417	7.440	2.115	- 0.600	- 0.268
VaggC (%)	18.767	80.500	69.608	70.960	8.635	16.970	- 3.456
VpcC (%)	4.175	33.781	9.680	9.277	4.652	10.288	2.629
VwC (%)	5.337	36.652	13.295	12.289	4.385	11.070	2.594
GwcR (-)	0.270	0.823	0.478	0.450	0.124	- 0.324	0.587
Gs (-)	2.340	2.743	2.545	2.543	0.073	0.857	0.362
ν (-)	0.090	0.318	0.189	0.190	0.042	0.536	0.326
E (GPa)	9.825	52.314	29.028	28.060	8.980	- 0.124	0.211
ComS (MPa)	7.591	67.564	32.418	32.486	12.106	- 0.331	0.230
TenS (MPa)	2.015	5.826	4.164	4.082	0.829	- 0.160	- 0.106

E , ComS, and TenS. Furthermore, the datasets were manually scanned to discard those with missing information. Consequently, a total number of 75 datasets were finally obtained from experimental/laboratory works that were developed in the US states of Alabama, Arizona, Arkansas, California, Colorado, Delaware, Illinois, Indiana, Iowa, Kansas, Michigan, Missouri, North Carolina, North Dakota, Ohio, Oklahoma, Texas, Washington, and Wisconsin. In this regard, it is essential to highlight that all datasets used in this research are available for free download on the LTPP’s website. Table 2 presents the adopted datasets’ statistical description (i.e., each variable’s minimum, maximum, mean, median, standard deviation, kurtosis, and skewness values). Also, Fig. 1 shows a scatterplot matrix to depict the considered parameters’ variability and correlation. From Fig. 1, it is clear that there is no marked trend between the input data (i.e., the design variables comprised by the VairC, VaggC, VpcC, VwC, GwcR, and Gs) and the output data (i.e., the laboratory testing results comprised by ν , E , ComS, and TenS).

2.1 Definition of New Input Variables

According to Fig. 1, there is no strong correlation between the input and output variables (at least in their current form). Therefore, in order to generate more relationships between the considered properties, it was decided to create four new input variables, namely volumetric water-to-cement ratio (VwcR), volumetric paste content (VpasteC), aggregate-to-paste ratio (AggPasR), and paste to air ratio (PasAirR). Equations (1), (2), (3), and (4) show the mathematical formulations for calculating VwcR, VpasteC, AggPasR, and PasAirR, respectively. In Eq. (1), Gs_pc refers to the specific gravity of the PC, which was assumed as a typical value of

3.1 [32–34].

$$VwcR = GwcR * Gs_pc \tag{1}$$

$$VpasteC = VpcC + VwC \tag{2}$$

$$AggPasR = \frac{VaggC}{VpasteC} \tag{3}$$

$$PasAirR = \frac{VpasteC}{VairC} \tag{4}$$

2.2 Feature Scaling

According to Table 2, the input and output variables present different units and magnitudes, making the learning process of ANNs demanding. In order to simplify the artificial learning procedure, it was decided to apply a feature scaling technique called standardization; Eq. (5) shows its mathematical definition [35]. The standardization process transforms data arrays, obtaining a new one with three principal characteristics [36, 37]: (i) mean equal to 0, (ii) standard deviation equal to 1, (iii) and most data points are between the range [- 1, 1]. Table 3 exhibits the basic statistical description of the transformed variables; in this table, it is clear that the standardized variables present the expected (and previously explained) features.

$$Standardized\ value = \frac{Original\ value - Mean}{Standard\ deviation} \tag{5}$$

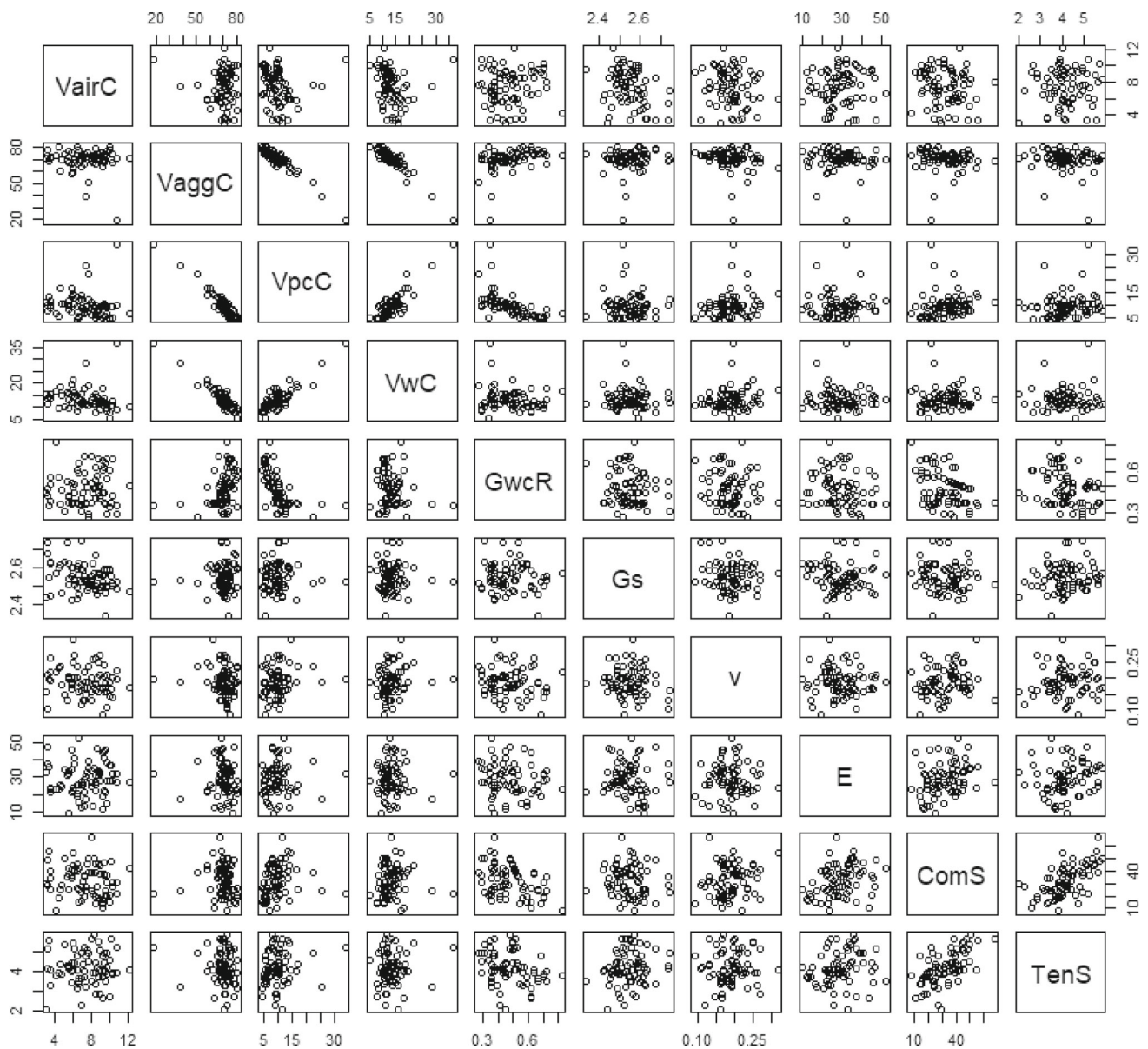


Fig. 1 Correlation and variability of the considered variables. Units: VairC (%); VaggC (%); VpcC (%); VwC (%); GwcR (-); Gs (-); v (-); E (GPa); ComS (MPa); TenS (MPa)

2.3 Data Augmentation

The main limitation of this study is that the employed database is only composed of 75 datasets. That amount of data is minimal; hence, an ML-based computational model created solely by that data could suffer from overfitting phenomena [38–40]. In order to avoid overfitting, it was decided to apply two techniques: (i) data augmentation during the data preprocessing stage and (ii) early stopping during the learning process stage. The early stopping technique is explained later in the manuscript. On the other hand, the data augmentation technique is just described below.

Data augmentation is one of the most powerful techniques to avoid overfitting [41–43]. Moreover, this technique stands out due to its simplicity [44, 45]. The data augmentation consists of conducting the model's learning process by considering several slightly altered copies of the original datasets together with the authentic ones [46–48]. Thus, for this research, it was decided to create 9 modified copies of each of the original datasets. Each modified copy was formed by affecting the authentic values with a pseudo-random distortion of between $\pm 3\%$. This low alteration ratio was selected to ensure that the physical consistency of the concrete mix designs was maintained. In this regard, 675 artificial datasets were obtained, which, added to the 75 original datasets, yields

Table 3 Statistical description of the variables after the standardization process

Variable (Unit)		%between [- 1, 1]	Minimum	Maximum	Mean	Median	Standard deviation	Kurtosis	Skewness
Input variables	VairC (-)	82.667	- 2.042	2.262	0.000	0.011	1.000	- 0.600	- 0.268
	VaggC (-)	92.000	- 5.887	1.261	0.000	0.157	1.000	16.970	- 3.456
	VpcC (-)	92.000	- 1.183	5.181	0.000	- 0.087	1.000	10.288	2.629
	VwC (-)	92.000	- 1.815	5.327	0.000	- 0.229	1.000	11.070	2.594
	GwcR (-)	88.000	- 1.678	2.781	0.000	- 0.226	1.000	- 0.324	0.587
	Gs (-)	88.000	- 1.678	2.781	0.000	- 0.226	1.000	- 0.324	0.587
	VwcR (-)	88.000	- 1.678	2.781	0.000	- 0.226	1.000	- 0.324	0.587
	VpasteC (-)	94.667	- 1.445	5.452	0.000	- 0.158	1.000	12.473	2.874
	AggPasR (-)	89.333	- 2.538	3.387	0.000	- 0.167	1.000	1.503	0.577
Output variables	PasAirR (-)	86.667	- 1.326	2.739	0.000	- 0.308	1.000	0.102	1.028
	v (-)	86.667	- 2.374	3.113	0.000	0.033	1.000	0.536	0.326
	E (-)	86.667	- 2.138	2.593	0.000	- 0.108	1.000	- 0.124	0.211
	ComS (-)	80.000	- 2.051	2.903	0.000	0.006	1.000	- 0.331	0.230
	TenS (-)	85.333	- 2.592	2.004	0.000	- 0.100	1.000	- 0.160	- 0.106

a total of 750. For the subsequent learning process (that is, the design of the ANN architecture and its performance assessments), the embraced database was separated as follows: 70% (i.e., 525 datasets) for the training process, 20% (i.e., 150 datasets) for the testing process, and 10% (i.e., the remaining 75 datasets) for the validation. It is crucial to highlight that this classification is accomplished using a random shuffle.

3 Methods

A broad set of ML-based computational techniques can be used for regression/forecasting problems, e.g., ANNs, bagging regressors, decision trees, lasso regression, random forests, and support vector machines [49–52]. In this research, it was decided to apply ANNs to estimate the PCC’s engineering properties (v , E , ComS, and TenS). The ANNs are one of the most popular deep-learning techniques because they allow for establishing correlations between variables that did not present a strong relationship [22, 45, 53]. The internal working of the ANNs is based on replicating the logic of the human brain’s neural connections [11, 54, 55]. Specifically, in this study, it was decided to apply a type of ANN called Deep Neural Networks (DNNs). The DNNs are defined as ANNs with at least two hidden layers and densely linked neurons (i.e., it is necessary to establish all the possible connections between neurons) [9, 39, 54]. Figure 2 shows the base architecture of DNNs.

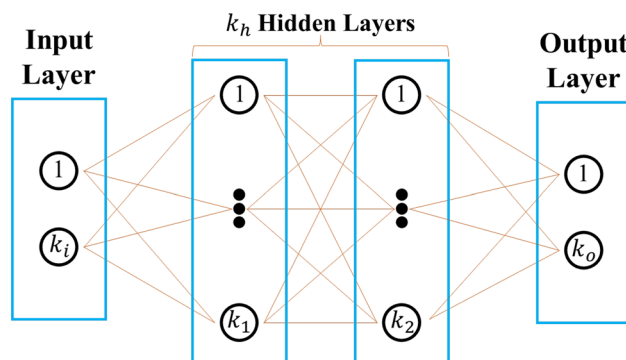


Fig. 2 Base architecture of DNNs. Adapted from: [54, 56]. Legend: k_i , k_h , k_o , k_1 , and k_2 are integer numbers equal to or greater than two

Based on the DNN canonical architecture shown in Fig. 2, the following aspects could be highlighted for this case study: (i) the input layer comprises 10 neurons (i.e., each neuron for each input variable presented in Table 3); (ii) the output layer comprises 4 neurons (i.e., each neuron for each output variable presented in Table 3); and (iii) the number of hidden layers and their number of neurons should be determined. In order to obtain a suitable configuration of the number of hidden layers and neurons, an extensive inspection of the possible combinations was carried out. In this way, the most appropriate DNN architecture was found, as indicated in Fig. 3. According to Fig. 3, there are four hidden layers. Respectively, the first, second, third and fourth hidden layers are composed of 800, 640, 160, and 40 neurons. Therefore, the proposed DNN is formed by 6 layers, 1654 neurons, and

Fig. 3 Proposed DNN

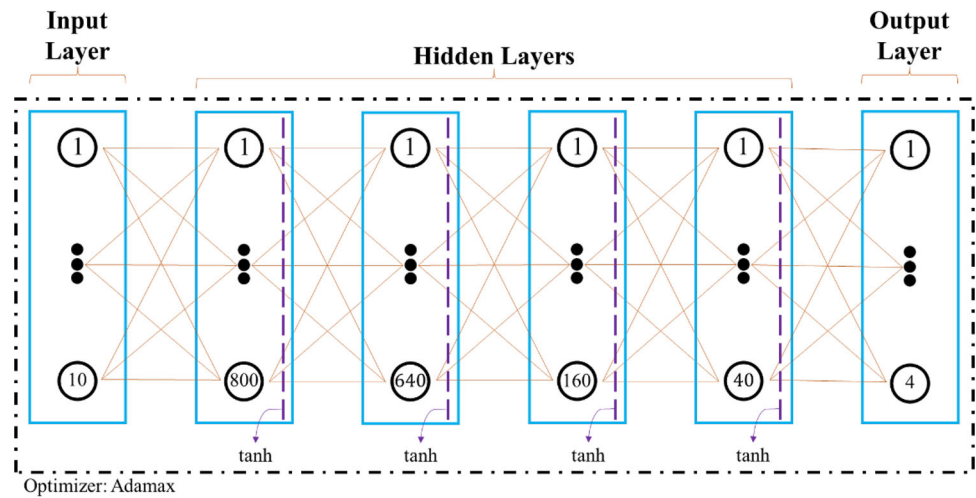


Table 4 Calculation of the total number of trainable parameters for the proposed DNN

Layer	Number of neurons received (Nr)	Number of biases (Nb)	Number of neurons contained (Nc)	Number of parameters [(Nr + Nb) * Nc]
First hidden layer	10	1	800	8 800
Second hidden layer	800	1	640	512 640
Third hidden layer	640	1	160	102 560
Fourth hidden layer	160	1	40	6 440
Output layer	40	1	4	164
Total number of trainable parameters				630 604

630,604 trainable parameters. The procedure to calculate the total number of trainable parameters is exhibited in Table 4.

In Fig. 3, two characteristics stand out: (i) all the hidden layers are affected by the hyperbolic tangent (tanh) activation function, and (ii) the proposed DNN is executed under a first-order gradient-based optimization method called Adamax. On the one hand, tanh is one of the most helpful activation functions because it causes all data coming out of a layer to be between the closed interval from -1 to 1 [57–59]. The preceding is particularly desirable in this case study because, according to Table 3, most of the data points (for both input and output variables) are comprehended within that range. In this regard, it was deliberately designed that the output layer would not have an activation function so that the few data outside the interval [− 1, 1] could be adequately predicted. On the other hand, Adamax is a variant of the traditional Adam optimization method [41, 60]. The internal functioning of Adamax is based on the infinity norm concept, which allows a high capacity to modify the learning rate according to the features of the input data [60, 61]. Algorithm 1 explains in detail the optimization procedure applied by Adamax [62, 63]. Also, Table 5 shows the hyperparameter values adopted

Table 5 Selected hyperparameter values for the Adamax optimizer

Hyperparameter	Value
β_1	0.9
β_2	0.999
lr	0.001
ϵ	1E−7

for this case study.

Algorithm 1 Pseudocode of the internal working of the Adamax optimizer. Adapted from: [62, 63].

```

t, m, u, w = 0, 0, 0, 0
for t = 1 to T do
    gt = ∇w * w
    m = β1 * m + (1 - β1) * gt
    u = max(β2 * u, abs(gt))
    lr = lr / (1 - β1 ** t)
    w = w - lr * m / (u + ε)
end for
    
```

t - iteration index; m - first moment vector; u - exponentially weighted infinity norm; w - convergence parameter (weight variable); T - number of iterations to reach the convergence; gt - gradient; β_1 - exponential decay rate for the first moment estimates; β_2 - exponential decay rate for the exponentially weighted infinity norm; lr - learning rate; ϵ - small constant for numerical stability

In order to make this research reproducible, duplicatable, and replicable, Algorithm 2 shows the simplified pseudocode of the proposed DNN. Further, the authors publicly share the computational model to the following GitHub repository (programmed in Python language): <https://github.com/rpol/oe/PCC>. It is essential to highlight that the authors bring an open license to guarantee the unrestricted use, modification, and distribution of the codes hosted in this repository. From Algorithm 2, four aspects should be discussed: (i) the selected loss function, (ii) the error metrics employed, and (iii) the required number of epochs.

The Mean Square Error (MSE) was chosen as the loss function. Equation (6) explains it [54, 56]. The MSE is one of the most common loss functions because it maximizes the

epochs (depending on the error metrics). Thus, with the aim of adopting a conservative number of epochs, this was set at 500.

$$MSE = \frac{1}{n} \sum_{i=1}^n (OD^i - ED^i)^2 \tag{6}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |OD^i - ED^i| \tag{7}$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{OD^i - ED^i}{OD^i} \right| \tag{8}$$

$$MSLE = \frac{1}{n} \sum_{i=1}^n \left(\ln(1 + OD^i) - \ln(1 + ED^i) \right)^2 \tag{9}$$

$$LHCE = \frac{1}{n} \sum_{i=1}^n \left(\frac{e^{OD^i - ED^i} + e^{ED^i - OD^i}}{2} \right) \tag{10}$$

where n—number of data points. i—data index. OD—observed/experimental data. ED—estimated data.

Algorithm 2 Simplified pseudocode to create the proposed DNN

```

Input: VairC, VaggC, VpcC, VwC, VwcR, GwC, Gs, VpasteC, AggPasR, PasAirR, v, E, ComS, TenS
input_data = dataset[:, 0:10]
output_data = dataset[:, 10:14]
def build_model():
    model = ([ input_shape = [10],
                layers.Dense(800, activation = "tanh"),
                layers.Dense(640, activation = "tanh"),
                layers.Dense(160, activation = "tanh"),
                layers.Dense(40, activation = "tanh"),
                layers.Dense(4) ])
    model.compile(loss = "MSE", optimizer = Adamax, metrics=["MAE", "MAPE", "MSLE", "LHCE"])
return model
model = build_model()
model.fit(input_data, output_data, epochs = 500)
    
```

differences between predicted and expected values [55, 64, 65]. Furthermore, four additional error metrics were considered, namely Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Logarithmic Error (MSLE), and the Logarithm of the Hyperbolic Cosine of the Error (LHCE). Respectively, Eqs. (7), (8), (9), and (10) define the MAE, MAPE, MSLE, and LHCE [54, 56]. On the other hand, the early stopping technique demarcated the required number of epochs (i.e., cycles of training and testing). This technique consists of determining the maximum number of epochs that the optimization process can reach before stopping learning, which is evidenced when, in the error functions (in this case, MSE, MAE, MAPE, MSLE, and LHCE), the assumed error changes from a decreasing trend to an ascending one [40, 54, 66]. In other words, the early stopping technique consists of locating the moment of maximum precision [54, 67, 68]. For the proposed DNN, the instant when learning stagnated varied between 600 and 1400

4 Discussion

4.1 Model's Exactness

Figure 4 compares the measured and predicted values of the PCC's engineering properties through 1:1 line plots. This figure clearly shows that the proposed DNN model can assemble forecasting with almost perfect accuracy. Regardless of the origin of the datasets (i.e., training, testing, or validation), the computational model is able to reproduce the laboratory-tested behaviour. It is paramount to emphasise that the training and testing datasets are employed along the supervised learning process, but the model has no prior knowledge of the validation datasets [69, 70]. Therefore, the validation datasets corroborate whether the DNN architecture can forecast never-before-observed scenarios [71,

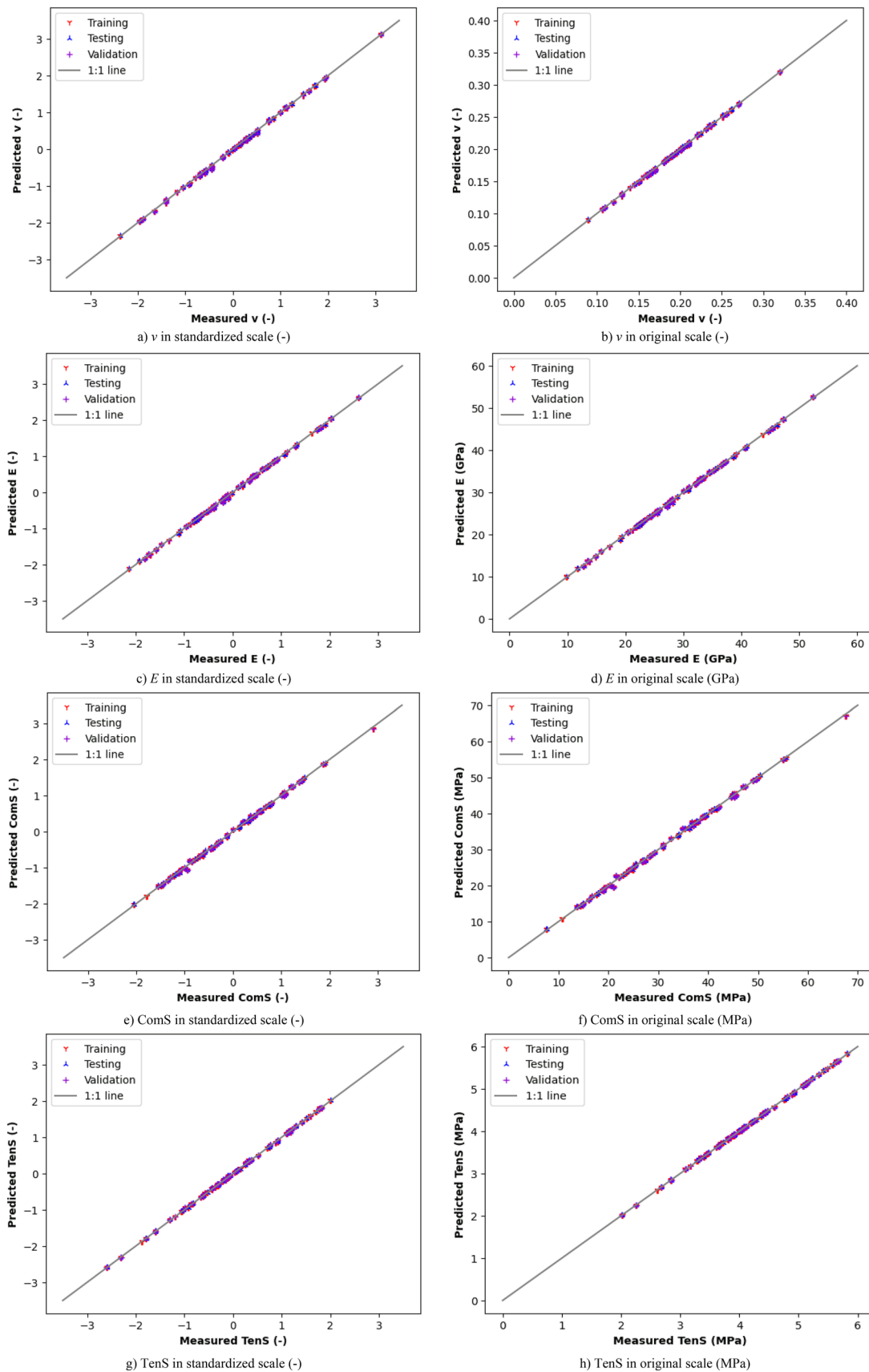


Fig. 4 Comparison of measured and predicted values (1:1 lines)

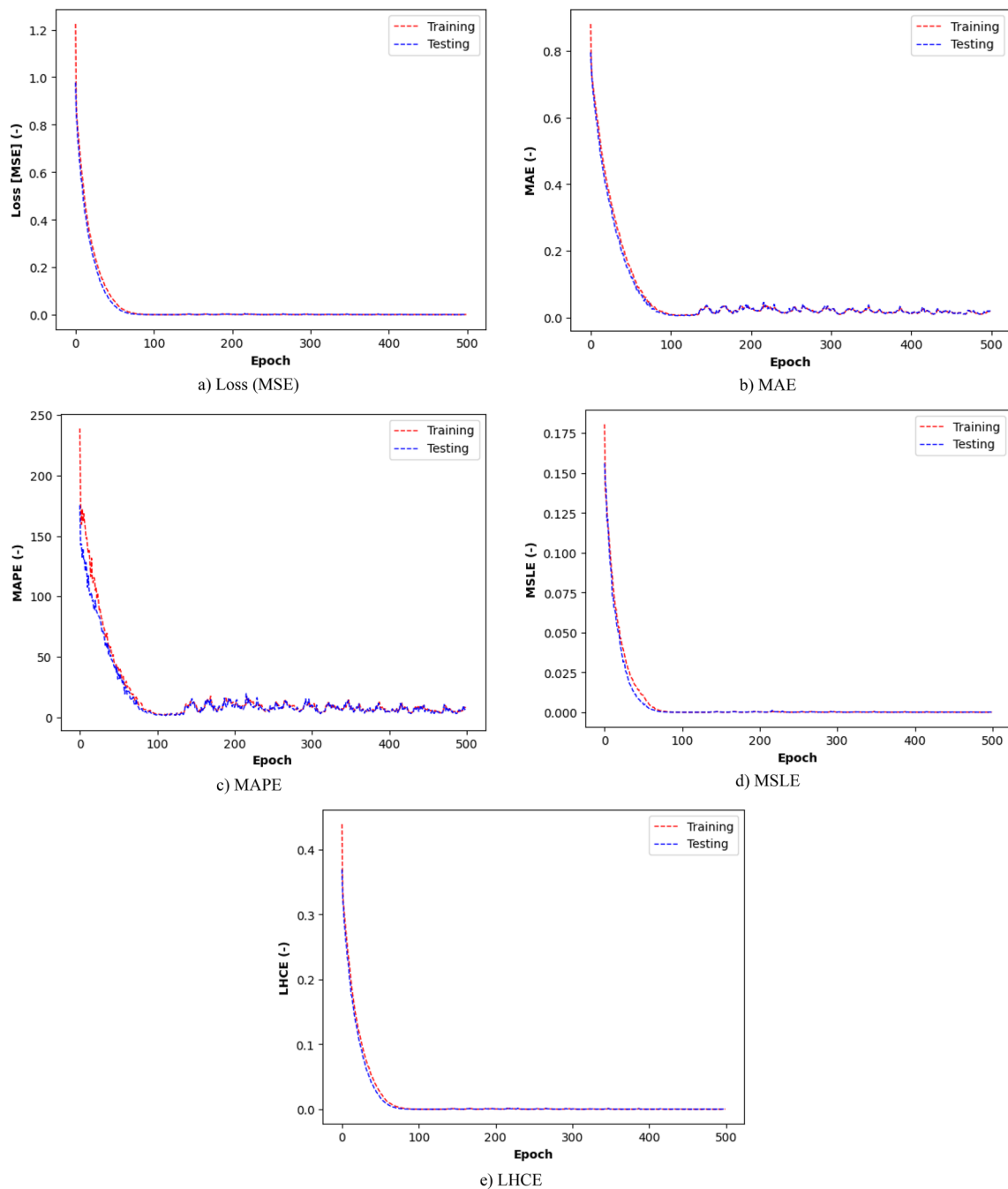


Fig. 5 Statistical evaluation of the proposed DNN

72]. Thus, based on Fig. 4, it is possible to recognise the almost perfect estimation capacity of the proposed computational model. Besides, several goodness-of-fit parameters are examined in Fig. 5 and Table 6. These parameters are the previously introduced error metrics and the coefficient of determination (R^2). Table 6 shows the goodness-of-fit parameters at the first and last epochs, as well as the “initial/final” error ratio. From this table, the following findings can be drawn. First, at the final epoch, the prediction errors

of the DNN model are close to zero (0). Second, the error metrics that best capture the improving behaviour of the DNN architecture through the epochs are MSE and MSLE. Third, at the initial epoch, the accuracy of the estimations according to the R^2 parameters was between 2.7–11.7%, 0.5–2.3%, 19.4–24.6%, and 0.4–6.0% for v , E , ComS, and TenS, respectively. Fourth, at the final epoch, the accuracy of the estimations according to the R^2 parameters was higher

Table 6 Summary of goodness-of-fit parameters determined for the proposed DNN

Estimated property	Error criteria	Training			Testing			Validation		
		Initial epoch	Final epoch	Initial/Final	Initial epoch	Final epoch	Initial/Final	Initial epoch	Final epoch	Initial/Final
<i>v</i>	Loss [MSE]	0.95307	0.00065	1473.1	0.93973	0.00071	1319.8	0.90340	0.00097	931.3
	MAE	0.76007	0.01885	40.3	0.74281	0.02015	36.9	0.77438	0.02277	34.0
	MAPE	2.67958	0.06002	44.6	2.27937	0.05926	38.5	2.93434	0.07106	41.3
	MSLE	0.98706	0.00747	132.1	0.73095	0.01249	58.5	0.74517	0.00966	77.1
	LHCE	1.60035	1.00032	1.6	1.59679	1.00036	1.6	1.54271	1.00049	1.5
	R ²	0.03903	0.99937	–	0.11653	0.99920	–	0.02655	0.99901	–
<i>E</i>	Loss [MSE]	1.01149	0.00045	2232.9	0.99377	0.00068	1459.3	0.91259	0.00071	1294.4
	MAE	0.82109	0.01655	49.6	0.79397	0.01993	39.8	0.74552	0.02037	36.6
	MAPE	1.82340	0.07119	25.6	2.34037	0.09195	25.5	1.91979	0.07520	25.5
	MSLE	0.75877	0.00180	422.5	0.76906	0.00269	285.9	0.66583	0.00408	163.4
	LHCE	1.63697	1.00023	1.6	1.63374	1.00034	1.6	1.59762	1.00035	1.6
	R ²	0.00914	0.99955	–	0.02266	0.99932	–	0.00467	0.99925	–
ComS	Loss [MSE]	0.79397	0.00083	960.1	0.82935	0.00100	831.8	0.75057	0.00156	480.8
	MAE	0.74360	0.02061	36.1	0.76523	0.02238	34.2	0.73935	0.02732	27.1
	MAPE	1.64045	0.18042	9.1	1.42121	0.10869	13.1	1.28304	0.16640	7.7
	MSLE	0.64641	0.00575	112.4	0.65638	0.00665	98.7	0.80670	0.01175	68.7
	LHCE	1.46454	1.00041	1.5	1.49329	1.00050	1.5	1.44754	1.00078	1.4
	R ²	0.19446	0.99921	–	0.21200	0.99882	–	0.24592	.99833	–
TenS	Loss [MSE]	0.95215	0.00036	2674.6	0.89210	0.00034	2631.5	1.02651	0.00025	4057.4
	MAE	0.74771	0.01484	50.4	0.70519	0.01465	48.1	0.77107	0.01210	63.7
	MAPE	1.17022	0.03830	30.6	1.36762	0.04362	31.4	1.22548	0.03255	37.7
	MSLE	0.76748	0.03229	23.8	0.61149	0.01066	57.3	0.38771	0.04121	9.4
	LHCE	1.61469	1.00018	1.6	1.59115	1.00017	1.6	1.66655	1.00013	1.7
	R ²	0.06040	.99964	–	0.04465	0.99967	–	0.00445	0.99974	–

than 99.8% for all the forecasted properties. Also, Fig. 5 provides a detailed explanation of the error evolution along the epochs, which reveals that overfitting and underfitting are not occurring.

4.2 SHAP Assessments

Figure 6 exhibits four SHAP assessments on the proposed DNN: a summary plot, a waterfall plot, a decision plot for the training datasets, and a decision plot for the testing datasets. The summary plot represents a SHAP global interpretation of the DNN model, whilst the other charts correspond to SHAP local interpretations [54, 73]. According to these plots, it is possible to draw the following findings. In the summary plot, each input variable has an associated SHAP value. Higher

SHAP values indicate more influence on the global predictive response of the DNN model [74, 75]. Therefore, the *Gs*, *VairC*, *AggPasR*, *PasAirR*, and *GwcR* are the input variables with more weight in the computational model. Thus, it is clear that creating new variables was a valuable strategy for this case study. In the waterfall plot, the input variables marked with red color (i.e., solely *Gs* and *VairC*) display a positive contribution to the output variables, whilst the blue color (i.e., all the input variables except *Gs* and *VairC*) denotes a negative one. This behaviour is helpful for the user to understand how the DNN model internally works [76, 77]. Meanwhile, the decision plots do not provide meaningful information individually since they reveal the learning trend followed by the DNN model [75, 77]. However, when comparing the decision plots for the training and testing datasets, it is notable that they follow a very similar trajectory. Hence, it is evidenced

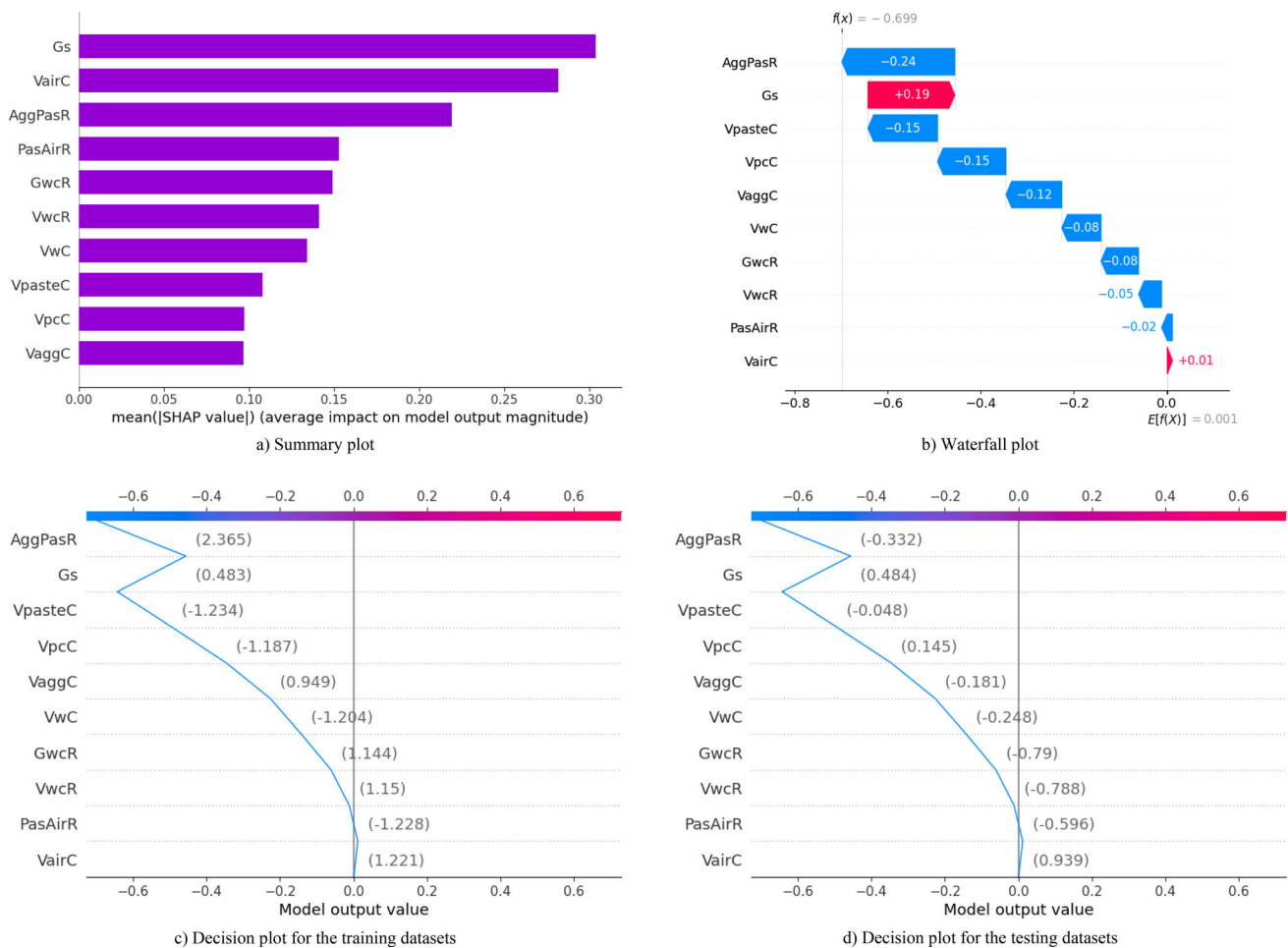


Fig. 6 SHAP assessments for the proposed DNN model

that the two phases of the learning process are congruent, which is desired [78, 79].

4.3 Running Time

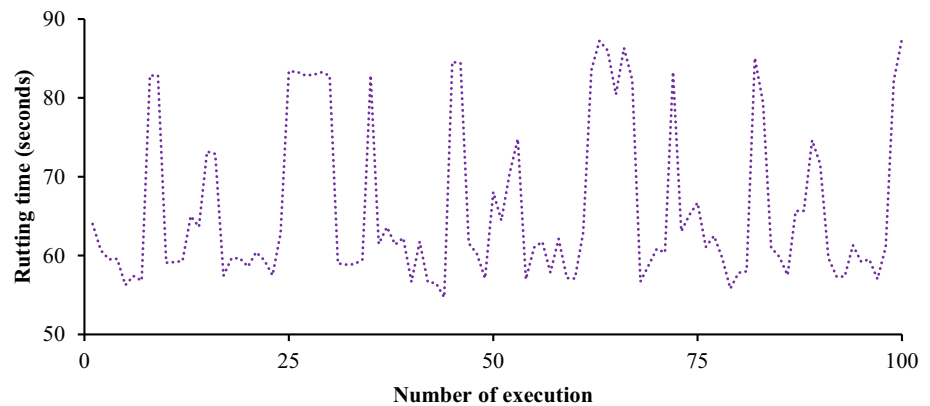
The running time of ML-based computational models is an important parameter to assess the feasibility of their implementation in the daily-basis engineering practice [52, 56]. For this purpose, the proposed DNN model was evaluated using Python’s native "TIME" module. Thus, the proposed DNN was executed on 100 independent occasions; Fig. 7 shows these results. According to this graph, the minimum, maximum, and average running times were 54.70, 87.47, and 66.01 s, respectively. Also, the standard deviation of the measurements was 10.15 s. Therefore, the proposed DNN requires approximately one minute for its execution, i.e., an acceptable magnitude within the context of civil engineering [52, 54]. It is essential to highlight that the running time depends primarily on the software and hardware used [80–82]. Hence, in order to be transparent, the ones used in this research are detailed below. On the one hand, Google

Colab (a hosted Jupyter Notebook service) was utilized as the development environment. On the other hand, the NVIDIA® V100 Tensor Core GPU was adopted as the acceleration hardware.

4.4 Model’s Limitations

The central limitation of the proposed DNN model is associated with the adopted database. The DNN architecture was trained, tested, and validated with 750 datasets. Nonetheless, only 75 datasets were obtained experimentally (from the LTPP database), and the other 675 datasets were artificially created through the data augmentation technique. Although this technique has been widely used in the literature to improve limited databases in ML-related regression problems [42–45, 83–85], the new modified database may not be broad enough to cover all physically/phenomenologically conceivable scenarios [44, 86, 87]. Therefore, the proposed DNN model may not yield highly accurate predictions of engineering properties (v , E , ComS, and TenS) for PCC mix

Fig. 7 Analysis of running time for the proposed DNN



designs that incorporate unconventional proportions of raw materials.

Another limitation of the computational model is that it is designed to forecast ordinary PCC's engineering properties (v , E , ComS, and TenS). In other words, the proposed DNN model is not able to predict the performance-related properties of mix designs that contain additives or supplementary cementitious materials (e.g., blast furnace slag, coal bottom ash, coal fly ash, metakaolin, palm oil ash, rice husk ash, silica fume, and steel slag [4, 88–90]). The preceding is particularly important considering that modified PCCs are increasingly used to construct buildings and rigid pavement.

5 Conclusions

In this investigation, an ML-based computational model was developed to estimate the engineering properties of the PCC, namely v , E , ComS, and TenS. Specifically, the ANN technique was employed as the primary computational method. In this regard, the LTPP database was utilized as the data source for the experimental/laboratory information. Eventually, a DNN architecture was proposed and evaluated with 1:1 lines, goodness-of-fit parameters, SHAP assessments, and running time analyses. Thus, the main conclusions that can be drawn from this study are presented below:

- The data preprocessing techniques (i.e., the definition of new input variables, feature scaling, and data augmentation) were necessary to obtain a proper ML-based computational model.
- The most suitable DNN architecture comprised one 10-neuron input layer (each neuron for each input variable), four hidden layers, and one 4-neuron output layer (each neuron for each output variable). The hidden layers subsequently had the following number of neurons: 800, 640, 160 and 40. Also, the tanh activation function was added for all the hidden layers.

- Notoriously, the proposed DNN shows an accuracy higher than 99.8%, which makes it a valuable tool for estimating the engineering properties of PCC when there is no possibility of experimental measurements.
- The SHAP assessments demonstrated that the input variables more critical for the proposed DNN model were the Gs, VairC, AggPasR, PasAirR, and Gwcr.
- The average running time of the proposed DNN model is approximately one minute, which implies a relatively low time quantity. In this regard, the model's speed is expected to be an attractive feature for potential users.
- Although the proposed DNN model is limited by the few original datasets, its most significant advantage is that it can be easily used for transfer learning. Accordingly, other researchers and designers can adjust/fine-tune the model for their contexts. Hence, the authors publicly share the computational model assembled in this study through an open-access GitHub repository.

Acknowledgements The authors thank the Department of Science, Technology, and Innovation (COLCIENCIAS) and the Universidad del Norte for funding this investigation through the "Research Project 745/2016, Contract 037-2017, No. 1215-745-59105".

Funding Open Access funding provided by Colombia Consortium.

Data availability The authors publicly share the computational model produced in this research through the following GitHub repository: <https://github.com/rpoloe/PCC>.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Liu, Y.; Du, P.; Tan, K.H.; Du, Y.; Su, J.; Shi, C.: Experimental and analytical studies on residual flexural behaviour of reinforced alkali-activated slag-based concrete beams after exposure to fire. *Eng. Struct.* **298**, 1–14 (2024). <https://doi.org/10.1016/j.engstruct.2023.117035>
- Singh, A.; Bhadauria, S.S.; Thakare, A.A.; Kumar, A.; Mudgal, M.; Chaudhary, S.: Durability assessment of mechanochemically activated geopolymer concrete with a low molarity alkali solution. *Case Stud. Constr. Mater.* **20**, 1–19 (2024). <https://doi.org/10.1016/j.cscm.2023.e02715>
- Singh, P.R.; Vanapalli, K.R.; Jadda, K.: Durability assessment of fly ash, GGBS, and silica fume based geopolymer concrete with recycled aggregates against acid and sulfate attack. *J. Build. Eng.* **82**, 1–17 (2024). <https://doi.org/10.1016/j.jobe.2023.108354>
- Polo-Mendoza, R.; Mora, O.; Duque, J.; Turbay, E.; Martinez-Arguelles, G.; Fuentes, L.; Guerrero, O.; Perez, S.: Environmental and economic feasibility of implementing perpetual pavements (PPs) against conventional pavements: a case study of Barranquilla city, Colombia. *Case Stud. Constr. Mater.* **18**, 1–21 (2023). <https://doi.org/10.1016/j.cscm.2023.e02112>
- Yuanliang, X.; Zhongshuai, H.; Chao, L.; Chao, Z.; Yamei, Z.: Unveiling the role of Portland cement and fly ash in pore formation and its influence on properties of hybrid alkali-activated foamed concrete. *Constr. Build. Mater.* **411**, 1–10 (2024). <https://doi.org/10.1016/j.conbuildmat.2023.134336>
- Yang, Y.; Yao, J.; Liu, J.; Kong, D.; Gu, C.; Wang, L.: Evaluation of the thermal and shrinkage stresses in restrained concrete: new method of investigation. *Constr. Build. Mater.* **411**, 1–14 (2024). <https://doi.org/10.1016/j.conbuildmat.2023.134493>
- Amin, M.N.; Khan, K.; Javed, M.F.; Aslam, F.; Qadir, M.G.; Faraz, M.I.: Prediction of mechanical properties of fly-ash/slag-based geopolymer concrete using ensemble and non-ensemble machine-learning techniques. *Materials* **15**, 1–20 (2022). <https://doi.org/10.3390/ma15103478>
- Cao, R.; Fang, Z.; Jin, M.; Shang, Y.: Application of machine learning approaches to predict the strength property of geopolymer concrete. *Materials* **15**, 1–15 (2022). <https://doi.org/10.3390/ma15072400>
- Huynh, A.T.; Nguyen, Q.D.; Xuan, Q.L.; Magee, B.; Chung, T.; Tran, K.T.; Nguyen, K.T.: A machine learning-assisted numerical predictor for compressive strength of geopolymer concrete based on experimental data and sensitivity analysis. *Appl. Sci.* **10**, 1–16 (2020). <https://doi.org/10.3390/app10217726>
- Nithurshan, M.; Elakneswaran, Y.: A systematic review and assessment of concrete strength prediction models. *Case Stud. Constr. Mater.* **18**, 1–15 (2023). <https://doi.org/10.1016/j.cscm.2023.e01830>
- Moein, M.M.; Saradar, A.; Rahmati, K.; Ghasemzadeh Mousavinejad, S.H.; Bristow, J.; Aramali, V.; Karakouziyan, M.: Predictive models for concrete properties using machine learning and deep learning approaches: a review. *J. Build. Eng.* **63**, 1–41 (2023). <https://doi.org/10.1016/j.jobe.2022.105444>
- Ahmed, H.U.; Mohammed, A.S.; Qaidi, S.M.A.; Faraj, R.H.; Hamah Sor, N.; Mohammed, A.A.: Compressive strength of geopolymer concrete composites: a systematic comprehensive review, analysis and modeling. *Eur. J. Environ. Civ. Eng.* **27**, 1383–1428 (2023). <https://doi.org/10.1080/19648189.2022.2083022>
- Mansouri, E.; Manfredi, M.; Hu, J.-W.: Environmentally friendly concrete compressive strength prediction using hybrid machine learning. *Sustainability* **14**, 1–17 (2022). <https://doi.org/10.3390/su142012990>
- Marks, M.; Glinicki, M.A.; Gibas, K.: Prediction of the chloride resistance of concrete modified with high calcium fly ash using machine learning. *Materials* **8**, 8714–8727 (2015). <https://doi.org/10.3390/ma8125483>
- Najm, H.M.; Nanayakkara, O.; Ahmad, M.; Sabri Sabri, M.M.: Mechanical properties, crack width, and propagation of waste ceramic concrete subjected to elevated temperatures: a comprehensive study. *Materials* **15**, 1–32 (2022). <https://doi.org/10.3390/ma15072371>
- Tang, Y.X.; Lee, Y.H.; Amran, M.; Fediuk, R.; Vatin, N.; Kueh, A.B.H.; Lee, Y.Y.: Artificial neural network-forecasted compression strength of alkaline-activated slag concretes. *Sustainability* **14**, 1–20 (2022). <https://doi.org/10.3390/su14095214>
- Shafiqh, P.; Asadi, I.; Mahyuddin, N.B.: Concrete as a thermal mass material for building applications—A review. *J. Build. Eng.* **19**, 14–25 (2018). <https://doi.org/10.1016/j.jobe.2018.04.021>
- Rocha Segundo, I.; Silva, L.; Palha, C.; Freitas, E.; Silva, H.: Surface rehabilitation of Portland cement concrete (PCC) pavements using single or double surface dressings with soft bitumen, conventional or modified emulsions. *Constr. Build. Mater.* **281**, 1–15 (2021). <https://doi.org/10.1016/j.conbuildmat.2021.122611>
- Aquino Rocha, J.H.; Toledo Filho, R.D.: Microstructure, hydration process, and compressive strength assessment of ternary mixtures containing Portland cement, recycled concrete powder, and metakaolin. *J. Clean. Prod.* **434**, 1–24 (2024). <https://doi.org/10.1016/j.jclepro.2023.140085>
- Zou, Y.; Zheng, C.; Alzahrani, A.M.; Ahmad, W.; Ahmad, A.; Mohamed, A.M.; Khallaf, R.; Elattar, S.: Evaluation of artificial intelligence methods to estimate the compressive strength of geopolymers. *Gels* **8**, 1–23 (2022). <https://doi.org/10.3390/gels8050271>
- Shah, H.A.; Yuan, Q.; Akmal, U.; Shah, S.A.; Salmi, A.; Awad, Y.A.; Shah, L.A.; Iftikhar, Y.; Javed, M.H.; Khan, M.I.: Application of machine learning techniques for predicting compressive, splitting tensile, and flexural strengths of concrete with metakaolin. *Materials* **15**, 1–36 (2022). <https://doi.org/10.3390/ma15155435>
- Silva, V.P.; Carvalho, R.D.; Rêgo, J.H.; Evangelista, F., Jr.: Machine learning-based prediction of the compressive strength of Brazilian concretes: a dual-dataset study. *Materials* **16**, 1–16 (2023). <https://doi.org/10.3390/ma16144977>
- FHWA. Long-Term Pavement Performance Information Management System User Guide. Fed. Highw. Adm. FHWA-HRT-2, 1–208 (2021)
- Karalafitis, A.G.; Badr, A.: Predicting asphalt pavement crack initiation following rehabilitation treatments. *Transp. Res. Part C Emerg. Technol.* **55**, 510–517 (2015). <https://doi.org/10.1016/j.trc.2015.03.031>
- Jia, Y.; Wang, S.; Huang, A.; Gao, Y.; Wang, J.; Zhou, W.: A comparative long-term effectiveness assessment of preventive maintenance treatments under various environmental conditions. *Constr. Build. Mater.* **273**, 1–10 (2021). <https://doi.org/10.1016/j.conbuildmat.2020.121717>
- Yu, Y.; Sun, L.: Effect of overlay thickness, overlay material, and pre-overlay treatment on evolution of asphalt concrete overlay roughness in LTPP SPS-5 experiment: a multilevel model approach. *Constr. Build. Mater.* **162**, 192–201 (2018). <https://doi.org/10.1016/j.conbuildmat.2017.12.039>
- Sollazzo, G.; Fwa, T.F.; Bosurgi, G.: An ANN model to correlate roughness and structural performance in asphalt pavements. *Constr. Build. Mater.* **134**, 684–693 (2017). <https://doi.org/10.1016/j.conbuildmat.2016.12.186>
- Gong, H.; Sun, Y.; Shu, X.; Huang, B.: Use of random forests regression for predicting IRI of asphalt pavements. *Constr. Build. Mater.* **189**, 890–897 (2018). <https://doi.org/10.1016/j.conbuildmat.2018.09.017>



29. Gong, H.; Huang, B.; Shu, X.: Field performance evaluation of asphalt mixtures containing high percentage of RAP using LTPP data. *Constr. Build. Mater.* **176**, 118–128 (2018). <https://doi.org/10.1016/j.conbuildmat.2018.05.007>
30. Chen, X.; Dong, Q.; Zhu, H.; Huang, B.: Development of distress condition index of asphalt pavements using LTPP data through structural equation modeling. *Transp. Res. Part C Emerg. Technol.* **68**, 58–69 (2016). <https://doi.org/10.1016/j.trc.2016.03.011>
31. Gong, H.; Sun, Y.; Hu, W.; Polaczyk, P.A.; Huang, B.: Investigating impacts of asphalt mixture properties on pavement performance using LTPP data through random forests. *Constr. Build. Mater.* **204**, 203–212 (2019). <https://doi.org/10.1016/j.conbuildmat.2019.01.198>
32. Rashidian-Dezfouli, H.; Rangaraju, P.R.: Evaluation of selected durability properties of portland cement concretes containing ground glass fiber as a pozzolan. *Transp. Res. Rec.* **2672**, 88–98 (2018). <https://doi.org/10.1177/0361198118773198>
33. Ogbodo, M.C.; Akpabot, A.I.: An assessment of some physical properties of different brands of cement in Nigeria. In: *IOP Conference Series: Materials Science and Engineering*, vol. 1048, pp. 1–5 (2021). <https://doi.org/10.1088/1757-899X/1048/1/012013>
34. Zemri, C.; Bachir Bouiadjra, M.: Comparison between physical–mechanical properties of mortar made with Portland cement (CEMI) and slag cement (CEMIII) subjected to elevated temperature. *Case Stud. Constr. Mater.* **12**, 1–12 (2020). <https://doi.org/10.1016/j.cscm.2020.e00339>
35. Latifoglu, L.; Ozger, M.: A novel approach for high-performance estimation of SPI data in drought prediction. *Sustainability* **15**, 1–29 (2023). <https://doi.org/10.3390/su151914046>
36. Anysz, H.; Zbiciak, A.; Ibadov, N.: The influence of input data standardization method on prediction accuracy of artificial neural networks. *Procedia Eng.* **153**, 66–70 (2016). <https://doi.org/10.1016/j.proeng.2016.08.081>
37. Deo, R.C.; Şahin, M.: Application of the Artificial Neural Network model for prediction of monthly Standardized Precipitation and Evapotranspiration Index using hydrometeorological parameters and climate indices in eastern Australia. *Atmos. Res.* **161–162**, 65–81 (2015). <https://doi.org/10.1016/j.atmosres.2015.03.018>
38. Chollet Ramampandra, E.; Scheidegger, A.; Wydler, J.; Schuwirth, N.: A comparison of machine learning and statistical species distribution models: quantifying overfitting supports model interpretation. *Ecol. Model.* **481**, 1–11 (2023). <https://doi.org/10.1016/j.ecolmodel.2023.110353>
39. Ookura, S.; Mori, H.: An efficient method for wind power generation forecasting by LSTM in consideration of overfitting prevention. *IFAC Pap.* **53**, 12169–12174 (2020). <https://doi.org/10.1016/j.ifacol.2020.12.1008>
40. Piotrowski, A.P.; Napiorkowski, J.J.: A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling. *J. Hydrol.* **476**, 97–111 (2013). <https://doi.org/10.1016/j.jhydrol.2012.10.019>
41. Bahtiyar, H.; Soydaner, D.; Yüksel, E.: Application of multilayer perceptron with data augmentation in nuclear physics. *Appl. Soft Comput.* **128**, 1–9 (2022). <https://doi.org/10.1016/j.asoc.2022.10.9470>
42. Min, R.; Wang, Z.; Zhuang, Y.; Yi, X.: Application of semi-supervised convolutional neural network regression model based on data augmentation and process spectral labeling in Raman predictive modeling of cell culture processes. *Biochem. Eng. J.* **191**, 1–9 (2023). <https://doi.org/10.1016/j.bej.2022.108774>
43. Hao, R.; Zheng, H.; Yang, X.: Data augmentation based estimation for the censored composite quantile regression neural network model. *Appl. Soft Comput.* **127**, 1–11 (2022). <https://doi.org/10.1016/j.asoc.2022.109381>
44. Demir, S.; Mincev, K.; Kok, K.; Paterakis, N.G.: Data augmentation for time series regression: applying transformations, autoencoders and adversarial networks to electricity price forecasting. *Appl. Energy* **304**, 1–19 (2021). <https://doi.org/10.1016/j.apenergy.2021.117695>
45. Hao, R.; Weng, C.; Liu, X.; Yang, X.: Data augmentation based estimation for the censored quantile regression neural network model. *Expert Syst. Appl.* **214**, 1–15 (2023). <https://doi.org/10.1016/j.eswa.2022.119097>
46. Shorten, C.; Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *J. Big Data* **6**, 1–48 (2019). <https://doi.org/10.1186/s40537-019-0197-0>
47. Berrett, C.; Calder, C.A.: Data augmentation strategies for the Bayesian spatial probit regression model. *Comput. Stat. Data Anal.* **56**, 478–490 (2012). <https://doi.org/10.1016/j.csda.2011.08.020>
48. Mazzoleni, M.; Breschi, V.; Formentin, S.: Piecewise nonlinear regression with data augmentation. *IFAC Pap.* **54**, 421–426 (2021). <https://doi.org/10.1016/j.ifacol.2021.08.396>
49. Polo-Mendoza, R.; Martínez-Arguelles, G.; Peñabaena-Niebles, R.; Covilla-Valera, E.: Neural networks implementation for the environmental optimisation of the recycled concrete aggregate inclusion in warm mix asphalt. *Road Mater. Pavement Des.* (2023). <https://doi.org/10.1080/14680629.2023.2230298>
50. Himmetoğlu, S.; Delice, Y.; Aydoğan, E.K.; Uzal, B.: Green building envelope designs in different climate and seismic zones: multi-objective ANN-based genetic algorithm. *Sustain. Energy Technol. Assess.* **53**, 1–17 (2022). <https://doi.org/10.1016/j.seta.2022.102505>
51. Minh, D.; Wang, H.X.; Li, Y.F.; Nguyen, T.N.: Explainable artificial intelligence: a comprehensive review. *Artif. Intell. Rev.* **55**, 3503–3568 (2022). <https://doi.org/10.1007/s10462-021-10088-y>
52. Polo-Mendoza, R.; Martínez-Arguelles, G.; Peñabaena-Niebles, R.: A multi-objective optimization based on genetic algorithms for the sustainable design of Warm Mix Asphalt (WMA). *Int. J. Pavement Eng.* **24**, 2074417 (2023). <https://doi.org/10.1080/1029.8436.2022.2074417>
53. Bialek, J.; Bujalski, W.; Wojdan, K.; Guzek, M.; Kurek, T.: Dataset level explanation of heat demand forecasting ANN with SHAP. *Energy* **261**, 1–12 (2022). <https://doi.org/10.1016/j.energy.2022.12.5075>
54. Polo-Mendoza, R.; Duque, J.; Maşín, D.; Turbay, E.; Acosta, C.: Implementation of deep neural networks and statistical methods to predict the resilient modulus of soils. *Int. J. Pavement Eng.* **24**, 2257852 (2023). <https://doi.org/10.1080/10298436.2023.2257852>
55. Hamim, A.; Yusoff, N.I.M.; Omar, H.A.; Jamaludin, N.A.A.; Hassan, N.A.; El-Shafie, A.; Ceylan, H.: Integrated finite element and artificial neural network methods for constructing asphalt concrete dynamic modulus master curve using deflection time-history data. *Constr. Build. Mater.* **257**, 1–14 (2020). <https://doi.org/10.1016/j.conbuildmat.2020.119549>
56. Polo-Mendoza, R.; Martínez-Arguelles, G.; Peñabaena-Niebles, R.: Environmental optimization of warm mix asphalt (WMA) design with recycled concrete aggregates (RCA) inclusion through artificial intelligence (AI) techniques. *Res. Eng.* **17**, 1–15 (2023). <https://doi.org/10.1016/j.rineng.2023.100984>
57. Zdravković, S.; Kavitha, L.; Satrić, M.V.; Zeković, S.; Petrović, J.: Modified extended tanh-function method and nonlinear dynamics of microtubules. *Chaos Solitons Fractals* **45**, 1378–1386 (2012). <https://doi.org/10.1016/j.chaos.2012.07.009>
58. Wuraola, A.; Patel, N.: Resource efficient activation functions for neural network accelerators. *Neurocomputing* **482**, 163–185 (2022)
59. Liu, K.; Shi, W.; Huang, C.; Zeng, D.: Cost effective Tanh activation function circuits based on fast piecewise linear logic. *Microelectron. J.* **138**, 1–9 (2023). <https://doi.org/10.1016/j.mejo.2023.10.5821>

60. Kingma, D.P.; Ba, J.L.: Amax: a method for stochastic optimization. In: 3rd International Conference on Learning Representations (ICLR), pp. 1–15 (2015)
61. Pandi Chandran, P.; Hema Rajini, N.; Jeyakarthic, M.: Optimal deep belief network enabled malware detection and classification model. *Intell. Autom. Soft Comput.* (2023). <https://doi.org/10.32604/iasc.2023.029946>
62. Obayya, M.; Maashi, M.S.; Nemri, N.; Mohsen, H.; Motwakel, A.; Osman, A.E.; Alneil, A.A.; Alsaid, M.I.: Hyperparameter optimizer with deep learning-based decision-support systems for histopathological breast cancer diagnosis. *Cancers* **15**, 1–19 (2023). <https://doi.org/10.3390/cancers15030885>
63. Sadykov, M.; Haines, S.; Broadmeadow, M.; Walker, G.; Holmes, D.W.: Practical evaluation of lithium-ion battery state-of-charge estimation using time-series machine learning for electric vehicles. *Energies* **16**, 1–34 (2023). <https://doi.org/10.3390/en16041628>
64. Zhang, S.; Lei, H.; Zhou, Z.; Wang, G.; Qiu, B.: Fatigue life analysis of high-strength bolts based on machine learning method and SHapley Additive exPlanations (SHAP) approach. *Structures* **51**, 275–287 (2023). <https://doi.org/10.1016/j.istruc.2023.03.060>
65. Nicolson, A.; Paliwal, K.K.: Deep learning for minimum mean-square error approaches to speech enhancement. *Speech Commun.* **111**, 44–55 (2019). <https://doi.org/10.1016/j.specom.2019.06.002>
66. Koya, B.P.; Aneja, S.; Gupta, R.; Valeo, C.: Comparative analysis of different machine learning algorithms to predict mechanical properties of concrete. *Mech. Adv. Mater. Struct.* **29**, 4032–4043 (2022). <https://doi.org/10.1080/15376494.2021.1917021>
67. Vilares Ferro, M.; Doval Mosquera, Y.; Ribadas Pena, F.J.; Darriba Bilbao, V.M.: Early stopping by correlating online indicators in neural networks. *Neural Netw.* **159**, 109–124 (2023). <https://doi.org/10.1016/j.neunet.2022.11.035>
68. Zeng, J.; Zhang, M.; Lin, S.-B.: Fully corrective gradient boosting with squared hinge: fast learning rates and early stopping. *Neural Netw.* **147**, 136–151 (2022). <https://doi.org/10.1016/j.neunet.2021.12.016>
69. Singh, V.; Pencina, M.; Einstein, A.J.; Liang, J.X.; Berman, D.S.; Slomka, P.: Impact of train/test sample regimen on performance estimate stability of machine learning in cardiovascular imaging. *Sci. Rep.* **11**, 1–8 (2021). <https://doi.org/10.1038/s41598-021-93651-5>
70. Xu, Y.; Goodacre, R.: On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *J. Anal. Test.* **2**, 249–262 (2018). <https://doi.org/10.1007/s41664-018-0068-2>
71. Quinn, T.P.; Le, V.; Cardilini, A.P.A.: Test set verification is an essential step in model building. *Methods Ecol. Evol.* **12**, 127–129 (2021). <https://doi.org/10.1111/2041-210X.13495>
72. Straub, J.: Machine learning performance validation and training using a ‘perfect’ expert system. *MethodsX* **8**, 1–6 (2021). <https://doi.org/10.1016/j.mex.2021.101477>
73. Polo-Mendoza, R.; Duque, J.; Mašín, D.: Prediction of California bearing ratio and modified proctor parameters using deep neural networks and multiple linear regression: a case study of granular soils. *Case Stud. Constr. Mater.* **20**, 1–17 (2024). <https://doi.org/10.1016/j.cscm.2023.e02800>
74. Li, Z.: Extracting spatial effects from machine learning model using local interpretation method: an example of SHAP and XGBoost. *Comput. Environ. Urban Syst.* **96**, 1–18 (2022). <https://doi.org/10.1016/j.compenvurbysys.2022.101845>
75. Kim, Y.; Kim, Y.: Explainable heat-related mortality with random forest and SHapley Additive exPlanations (SHAP) models. *Sustain. Cities Soc.* **79**, 1–15 (2022). <https://doi.org/10.1016/j.scs.2022.103677>
76. Lin, K.; Gao, Y.: Model interpretability of financial fraud detection by group SHAP. *Expert Syst. Appl.* **210**, 1–9 (2022). <https://doi.org/10.1016/j.eswa.2022.118354>
77. Kashifi, M.T.: Investigating two-wheelers risk factors for severe crashes using an interpretable machine learning approach and SHAP analysis. *IATSS Res.* **47**, 357–371 (2023). <https://doi.org/10.1016/j.iatssr.2023.07.005>
78. Meng, Y.; Yang, N.; Qian, Z.; Zhang, G.: What makes an online review more helpful: an interpretation framework using XGBoost and SHAP values. *J. Theor. Appl. Electron. Commer. Res.* **16**, 466–490 (2021). <https://doi.org/10.3390/jtaer16030029>
79. Scavuzzo, C.M.; Scavuzzo, J.M.; Campero, M.N.; Anegagrie, M.; Aramendia, A.A.; Benito, A.; Periago, V.: Feature importance: opening a soil-transmitted helminth machine learning model via SHAP. *Infect. Dis. Model.* **7**, 262–276 (2022). <https://doi.org/10.1016/j.idm.2022.01.004>
80. Tang, Y.; Wang, C.: Performance modeling on DaVinci AI core. *J. Parallel Distrib. Comput.* **175**, 134–149 (2023). <https://doi.org/10.1016/j.jpdc.2023.01.008>
81. Dube, P.; Suk, T.; Wang, C.: AI gauge: runtime estimation for deep learning in the cloud. In: 31st International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), pp. 160–167 (2019)
82. Aghapour, Z.; Sharifian, S.; Taheri, H.: Task offloading and resource allocation algorithm based on deep reinforcement learning for distributed AI execution tasks in IoT edge computing environments. *Comput. Netw.* **223**, 1–17 (2023). <https://doi.org/10.1016/j.comnet.2023.109577>
83. Assaf, A.M.; Haron, H.; Hamed, H.N.A.H.; Ghaleb, F.A.; Dalam, M.E.; Eisa, T.A.E.: Improving solar radiation forecasting utilizing data augmentation model generative adversarial networks with convolutional support vector machine (GAN-CSVR). *Appl. Sci.* **13**, 1–23 (2023). <https://doi.org/10.3390/app132312768>
84. Harrou, F.; Dairi, A.; Dorbane, A.; Sun, Y.: Energy consumption prediction in water treatment plants using deep learning with data augmentation. *Res. Eng.* **20**, 1–14 (2023). <https://doi.org/10.1016/j.rineng.2023.101428>
85. Liu, K.-H.; Xie, T.-Y.; Cai, Z.-K.; Chen, G.-M.; Zhao, X.-Y.: Data-driven prediction and optimization of axial compressive strength for FRP-reinforced CFST columns using synthetic data augmentation. *Eng. Struct.* **300**, 1–16 (2024). <https://doi.org/10.1016/j.engstruct.2023.117225>
86. Mumuni, A.; Mumuni, F.: Data augmentation: a comprehensive survey of modern approaches. *Array* **16**, 1–27 (2022). <https://doi.org/10.1016/j.array.2022.100258>
87. Maharana, K.; Mondal, S.; Nemade, B.: A review: data pre-processing and data augmentation techniques. *Glob. Transitions Proc.* **3**, 91–99 (2022). <https://doi.org/10.1016/j.gltpr.2022.04.020>
88. Walubita, L.F.; Martinez-Arguelles, G.; Polo-Mendoza, R.; Ick-Lee, S.; Fuentes, L.: Comparative environmental assessment of rigid, flexible, and perpetual pavement: a case study of Texas. *Sustainability* **14**, 1–22 (2022). <https://doi.org/10.3390/su14169983>
89. Gupta, S.; Chaudhary, S.: State of the art review on supplementary cementitious materials in India—II: characteristics of SCMs, effect on concrete and environmental impact. *J. Clean. Prod.* **357**, 1–19 (2022). <https://doi.org/10.1016/j.jclepro.2022.131945>
90. Sharma, R.K.; Singh, D.; Dasaka, S.M.: Investigating supplementary cementitious materials’ effects on stabilized aggregate performance, behaviour, and design aspects. *Constr. Build. Mater.* **411**, 1–16 (2024). <https://doi.org/10.1016/j.conbuildmat.2023.134564>

