Journal of Building Engineering 82 (2024) 108370

Contents lists available at ScienceDirect

Journal of Building Engineering

journal homepage: www.elsevier.com/locate/jobe

Unveiling non-steady chloride migration insights through explainable machine learning

Woubishet Zewdu Taffese *, Leonardo Espinosa-Leal

School of Research and Graduate Studies, Arcada University of Applied Sciences, Helsinki, Finland

ARTICLE INFO

Keywords:
Model-agnostic explanations
SHAP
Chloride diffusion
Concrete
Durability
Explainable machine learning
Chloride migration coefficient

ABSTRACT

This study explores the influence of concrete mix ingredients on the non-steady chloride migration coefficient (D_{nssm}) using an explainable machine learning (XML) approach that integrates Extreme Gradient Boosting (XGBoost) and Shapley Additive Explanations (SHAP). The dataset, comprising 204 observations from literature, is utilized to train the XGBoost algorithm for predicting D_{nssm}. The model demonstrates notable performance metrics with (MAE = 1.61×10^{-12} m^2/s, RMSE = 2.38×10^{-12} m^2/s, and $R^2 = 0.95$) in the training set and (MAE = 2.22×10^{-12} m^2/s, RMSE = 3.18×10^{-12} m^2/s, and $R^2 = 0.87$) and the test set. The SHAP method provides comprehensive insights into feature importance, offering valuable information about the relationships and dependencies among various features. The top five features identified as significant contributors include coarse aggregate, superplasticizer, concrete age, cement, and water. Visualization of SHAP values through diverse plots proves essential for obtaining a thorough understanding of feature influence. The explainability of the model's results contributes new insights, aiding in the development of optimal and sustainable concrete with enhanced resistance to chloride penetration. Furthermore, the model's explainability fosters trust in its predictions, facilitating seamless integration into real-world applications.

1. Introduction

Concrete is an essential component in the field of civil engineering construction, but its durability is affected by a range of factors as time passes. Among these factors, chloride attack emerges as prominent threat to the durability of reinforced concrete (RC) structures, especially in marine environments or regions subjected to chloride-containing de-icing salts in cold climates [2]. While chloride penetration itself doesn't harm concrete, once the concentration of chloride ions surpasses a certain threshold at the steel reinforcement bars, it triggers depassivation and subsequent corrosion [3]. The corrosion of reinforcement bars caused by chloride has a detrimental impact on the functionality and safety of RC structures worldwide, leading to significant economic losses stemming from the premature need for rehabilitation and repair. In fact, some developed countries allocate a substantial portion of their gross domestic product (GDP), ranging from 3.5% to 4.5%, towards combating corrosion-related damage and implementing effective control measures [4].

Gaining a comprehensive understanding of the chloride transport process is essential for ensuring the durability and extended lifespan of structures exposed to chloride-rich environments. The transportation of chloride ions within concrete involves intricate physicochemical phenomena, encompassing multiple mechanisms such as diffusion, capillary suction, and permeation [5–7]. In order

* Corresponding author.
E-mail address: woubishet.taffese@arcada.fi (W.Z. Taffese).

https://doi.org/10.1016/j.jobe.2023.108370
Received 12 September 2023; Received in revised form 9 December 2023; Accepted 19 December 2023
Available online 24 December 2023

to simplify the assessment of chloride penetration into concrete, diffusion is commonly regarded as the primary mechanism considered by service-life prediction models like Life-365 [8] and DuraCrete [9]. These models rely on the concrete chloride diffusion coefficient, which is typically determined through laboratory testing using procedures like ASTM C1556 [10], NT Build 443 [11], and NT Build 492 [12]. However, these laboratory testing procedures are time-consuming and resource intensive.

To overcome this limitation, significant efforts have been made in recent decades to develop prediction models for the chloride diffusion coefficient based on phenomenological and physically-based approaches [13–15]. These models take into account factors related to the concrete mixture components. While traditional approaches have been successful in establishing correlations between parameters such as cement dosage, water-to-cement ratio, and exposure time, it remains challenging to effectively consider the combined effects of these factors. Furthermore, conventional methods often overlook the impact of secondary factors like the dosage of different types of chemical admixtures and Supplementary cementitious materials (SCMs). Consequently, developing a robust and universally applicable model for predicting concrete strength using conventional approaches is a daunting task [1].

In recent years, there has been a growing interest in implementing data-driven approaches to establish the relationships between inputs and outputs. These approaches involve the use of machine learning (ML) techniques to construct models that can predict outputs based on learned features from the data to solve complex problems. This inclination has garnered widespread acceptance across diverse fields. Within the realm of civil engineering, ML models are applied to address complex issues in concrete science [16–19], geotechnics [20–23], structural engineering, and other relevant areas. The use of ML in these contexts highlights its versatility and effectiveness in offering innovative solutions to complex problems within the field of civil engineering.

Regarding the prediction of concrete chloride diffusion coefficients, there are noteworthy ML-based models documented in Refs. [1, 24–27]. These studies employ various machine learning algorithms. For example, Hoang et al. [24] utilized multivariate adaptive regression splines (MARS) and multi-gene genetic programming (MGGP) to predict chloride diffusion in cement mortar, achieving a coefficient of determination (R^2) of 0.91. Yao et al. [25] harnessed the combined power of the Particle Swarm Optimization (PSO) algorithm and a Backpropagation (BP)-based Artificial Neural Network (ANN) to predict chloride penetration in concrete. In their study, they introduced various mineral admixtures into their concrete specimens and found that the PSO-BP neural network outperformed the BP neural network, yielding a superior estimate with an R^2 value of 0.967 as opposed to the BP neural network's R^2 value of 0.957. Delgado et al. [26] employed ANN to determine the diffusion coefficient of concrete specimens subjected to drying-wetting cycles. They used predictors such as cement type, water-to-cement ratio, mineral additives, curing age, and the number of drying-wetting cycles. The developed model adequately predicted the chloride diffusion coefficient with an R^2 value of 0.954. Tran [27] utilized a diverse array of algorithms, encompassing Support Vector Machine (SVM), Extreme Learning Machine (ELM), K-Nearest Neighbors (KNN), and five distinct types of decision tree-based ensemble models—amounting to eight algorithms in total. The aim was to predict the chloride diffusion coefficient of concrete incorporating SCMs like silica fume, ground granulated blast furnace slag, and fly ash. Notably, the author achieved a comparable R^2 value to that reported in Ref. [26] when employing the Gradient Boosting ensemble model. Taffese and Espinosa-Leal [1] utilized the eXtreme Gradient Boosting (XGBoost) algorithm to predict the chloride diffusion coefficient, achieving higher accuracy than previous works with an R^2 value of 0.963. They considered thirteen features describing the concrete mixture components, as well as basic fresh and hardened tests.

While the machine learning-based predictions mentioned above have shown high accuracy in predicting the chloride diffusion coefficient, their lack of interpretability is a significant drawback. These models are often treated as black boxes, providing little understanding of the actual relationships between the different features involved. Although some previous studies have attempted to determine feature importance for global interpretation, such efforts fall short in instilling confidence among practitioners. In any engineering application, it is essential for models to be transparent, explainable, and accountable, allowing engineers and stakeholders, even those without technical expertise, to comprehend how the algorithms predict the chloride diffusion coefficient. This level of understanding is crucial for building trust and effectively implementing these models in practical scenarios. Consequently, there is a need to seamlessly integrate a robust data interpretation algorithm with machine learning-based models for predicting the chloride diffusion coefficient. This integration aims to create a synergistic and interpretable ML-based predictive tool that addresses the current limitations.

The objective of this study is to create an explainable machine learning-based model for predicting the chloride diffusion coefficient. This model aims to go beyond conventional approaches by not only providing a clear understanding of its overall functioning (global explanation) but also offering detailed explanations for individual predictions (instance-based explanations). By achieving this level of explainability, the model will provide insights into its decision-making process for each prediction. Based on previous research indicating the superior accuracy of XGBoost [1], we have chosen to employ this algorithm in our study for developing the chloride diffusion coefficient prediction model. However, we have made adjustments to the types of features utilized. Further details regarding these modifications are provided in Section 3.1.

The structure of the paper is as follows: Section 2 provides an introduction to the concept of explainable machine learning (XML), with a particular emphasis on model-agnostic approaches. Section 3 outlines the data utilized for developing the chloride diffusion coefficient prediction models, along with the model development process. Section 4 examines the research findings. Finally, Section 5 presents the concluding remarks.

2. Explainable machine learning

Explainable machine learning approaches can be categorized into two main types: model-agnostic methods and model-specific methods. Model-specific methods are tailored to the characteristics of specific machine learning models or families of models. These methods leverage the internal structure and properties of the model to generate explanations. Model-agnostic methods are

techniques that can be applied to any machine learning model, regardless of its architecture or underlying algorithm. These post-hoc techniques explain predictions of these models by treating the models as black boxes and then generating explanations without inspecting the internal model parameters. Since these methods are not tied to specific model architectures, they can be applied to a wide range of ML algorithms.

There are various model-agnostic methods available for enhancing the interpretability of ML models, including popular techniques like SHAP (SHapley Additive exPlanations) [28] and LIME (Local Interpretable Model-Agnostic Explanations) [29]. They both serve the purpose of providing insights and understanding into the decision-making process of complex models. SHAP offers a more comprehensive approach, providing both global and local explanations, while LIME focuses primarily on local explanations. Thus, in this instance, we have employed the SHAP explainable model, which is founded on the fundamental principle outlined in Section 2.2. To construct a prediction model for the chloride diffusion coefficient of concrete, we employed the tree-based ensemble method known as XGBoost. The subsequent section provides a comprehensive explanation of its operational principles.

2.1. XGBoost

XGBoost, an influential ML algorithm, adheres to the fundamental principles of gradient boosting, where weak learners are combined to form a powerful learner. Originally developed by Tianqi Chen in 2016 [30], XGBoost has since undergone extensive contributions from numerous scholars. Unlike traditional gradient boosted trees that are constructed sequentially, XGBoost generates trees in parallel. By effectively controlling model complexity and mitigating overfitting through its inherent regulation techniques, XGBoost enhances prediction accuracy in each iteration. Fig. 1 presents the general architecture of XGBoost. Its remarkable speed and performance have established it as the premier ML algorithm for addressing regression problems in diverse construction engineering applications. Notably, XGBoost has been effectively employed in predicting porosity [17], prediction of pile bearing capacity [31] shear strength of concrete-to-concrete interfaces [32], chloride resistance of concrete [33], and forecasting the residual value of construction equipment [34].

Equation (1) defines the ultimate robust XGBoost model denoted as $F(.)$ [30,35].

$$\widehat{y}_i = F(X_i) = \sum_{k=1}^{K} f_k(X_i), \tag{1}$$

where $\widehat{y}_i$ represents the predicted value for the *ith* sample; $f_k(.)$ denotes the *kth* weak learner of the robust model, which corresponds to the *kth* decision tree; K indicates the total count of weak learners in the model.

Equation (2) presents the objective function used during the training phase.

$$Obj\ (\theta) = \sum_{i=1}^{N} l(\widehat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{2}$$

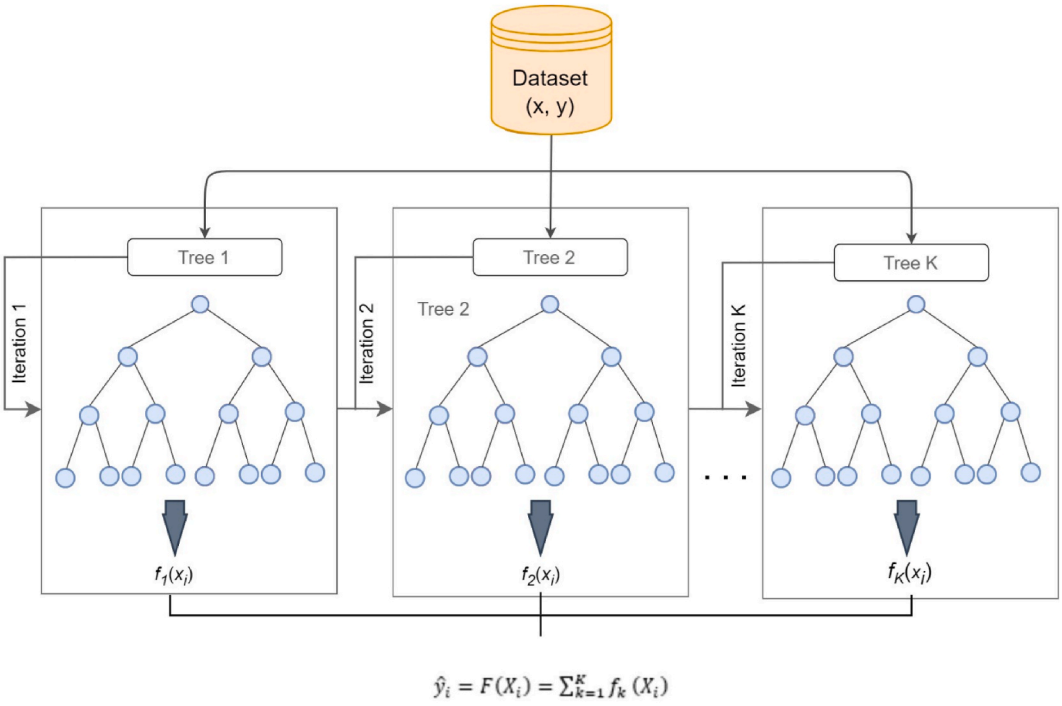

Fig. 1. The general architecture of XGBoost.

where $l(\cdot)$ denotes the loss function; y_i represents the true value of the *ith* sample; $\Omega(\cdot)$ corresponds to the regularization term.

The prediction for the *ith* sample at the *kth* iteration is given by Equation (3).

$$\hat{y}_i^k = \hat{y}_i^{k-1} + f_k(X_i), \tag{3}$$

where $\hat{y}_i^{k-1}$ represents the prediction from the previous cumulative model after the $(k-1)th$ iteration.

Thus, the objective function can be formulated as depicted in Equation (4).

$$Obj = \sum_{i=1}^{N} l\left(\hat{y}_i^{k-1} + f_k(X_i), y_i\right) + \Omega(f_k). \tag{4}$$

Due to XGBoost's utilization of a second-order Taylor approximation of the loss function, the objective function can be approximated as demonstrated in Equation (5).

$$Obj \cong \sum_{i=1}^{N} l\left(\hat{y}_i^{k-1}, y_i\right) + g_i f_k(X_i) + \frac{1}{2} h_i f_k^2(X_i) + \Omega(f_k), \tag{5}$$

where $g_i = \partial_{\hat{y}_i^{(k-1)}} l(\hat{y}_i^{k-1}, y_i)$, and $h_i = \partial^2_{\hat{y}_i^{(k-1)}} l(\hat{y}_i^{k-1}, y_i)$.

Equation (6) serves as a representation for the decision tree used as a weak learner.

$$f_k(X_i) = w_{q(X_i)}, \tag{6}$$

where q represents the tree's structure, while w corresponds to the leaf weights of the decision tree.

The regularization term $\Omega(f_k)$ can be expressed as depicted in Equation (7).

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{7}$$

where T denotes the total number of leaves in the decision tree, while γ and λ represent the penalty coefficients.

Equation (8) offers a more concise form of the objective function.

$$\begin{aligned} Obj &\cong \sum_{i=1}^{N}\left[g_i f_k(X_i) + \frac{1}{2} h_i f_k^2(X_i)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T}\left[\left(\sum_{i \in I_j} g_i\right) w_j + \frac{1}{2}\left(\sum_{i \in I_j} h_i + \lambda\right) w_j^2\right] + \gamma T. \end{aligned} \tag{8}$$

Hence, the optimal weights w_j^* for the decision tree and the optimal value of the objective function can be computed using Equation (9) and Equation (10), respectively.

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}. \tag{9}$$

$$Obj = -\frac{1}{2}\sum_{j=1}^{T} \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T. \tag{10}$$

2.2. SHAP

SHAP is based on cooperative game theory and utilizes Shapley values to determine the importance of features [17]. The Shapley value for a specific feature $\varnothing_j(val)$ is calculated by taking the weighted sum of its marginal contributions to the model's output across all possible subsets of features. Equation (11) [36] describes this computation process.

$$\varnothing_j(val) = \sum_{S \subseteq \{1,\ldots,p\} \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!}\left(val(S \cup \{j\}) - val(S)\right), \tag{11}$$

where S represent a subset of the features utilized in the model, x denote the feature value vector of the instance to be explained, and p represent the number of features. The term $val_x(S)$ refers to the prediction made for the feature values within set S. This prediction is obtained by marginalizing over the features not included in set S, as illustrated in Equation (12):

$$val_x(S) = \int \widehat{f}(x_1, \ldots, x_p) d\mathbb{P}_{x \notin S} - E_x(\widehat{f}(x)). \tag{12}$$

By calculating the average of the absolute Shapley values across multiple instances, as described in Equation (13), a more robust measure of feature importance (I_j) can be derived. This aggregation method provides a comprehensive assessment of the influence each feature has on the model's predictions. Features with higher absolute Shapley values are considered more significant in the model's prediction process.

$$I_j = \frac{1}{n}\sum_{i=1}^{n}\left|\varphi_j^{(i)}\right|. \tag{13}$$

3. Dataset and modelling

In this section, the primary focus lies on the utilization of experimental data to predict the chloride diffusion coefficient of concrete. It comprehensively presents the step-by-step process involved in creating the predictive model, encompassing data preprocessing, model training, and evaluation. The initial part of the section is dedicated to introducing and describing the experimental dataset, shedding light on its key characteristics and relevant information. Following that, a detailed account of the model development process is provided.

3.1. Experimental dataset

This study focuses on the chloride diffusion coefficients obtained through the NT Build 492 method, specifically referred to as the "non-steady-state migration coefficient" or D_{nssm}. Considering their comprehensiveness and appropriateness, the selected features for this study amount to 10 and encompass 204 observations sourced from the original dataset. This dataset was compiled from various research projects and internationally published journal articles [5–7,15,37–53]. While feature engineering, which involves creating new features to simplify the model and improve predictive performance, is a common practice, this study only utilizes features that describe the concrete constituents without any transformations. The primary objective is to investigate the impact of concrete constituents on D_{nssm}. The features considered in this study are water content (in kg/m^3), binder contents (cement, slag, fly ash, and silica fume, in kg/m^3), fine and coarse aggregate amounts (in kg/m^3), superplasticizer content (as a percentage of binder weight), migration test age (in days) when the diffusion coefficient test was conducted, and D_{nssm} (in units of x10^{-12} m^2/s). The dataset used in this study encompasses various types of concrete, including regular strength, lightweight, high-strength, high-performance, and self-consolidating concrete, without discriminating based on their strength class. Table 1 provides a summary of the features included in the database.

Histogram plots with distribution fits, as illustrated in Fig. 2, visualize the distribution of input features influencing the target feature, D_{nssm}, of the raw dataset. The distribution curves clearly indicate that the features do not conform to a normal distribution and display distinct shapes. Notably, water, cement, fine aggregate, and coarse aggregate exhibit a discernible bimodal distribution. As depicted in the figure, the number of concrete samples containing slag, fly ash, and silica fume is limited.

3.2. Data preprocessing

Data processing stands as a pivotal phase in the development of machine learning-based models, encompassing crucial tasks such as handling missing data, detecting and addressing outliers, data encoding, data scaling, and data partitioning. In the context of this study, all of these tasks are thoroughly executed, with the exception of data encoding and scaling. Data encoding is omitted due to the absence of categorical features in the dataset, eliminating the need for such processing. As for data scaling, while many machine learning algorithms benefit from having features on the same scale for optimal performance, the chosen algorithm for model training in this study doesn't necessitate data scaling. This is because the algorithm inherently incorporates scaling as an integral aspect of its model development process.

3.2.1. Missing data processing

Missing data processing refers to the techniques employed to handle and analyze datasets that contain incomplete values, which may arise from various causes. Addressing missing data is crucial as it can impact the reliability and validity of prediction models and the conclusions derived from the data. Several approaches can be utilized, including: i) excluding observations with any missing values, ii) relying on the learning algorithm to handle missing values during training, and iii) imputing all missing values before

Table 1
Explanation of features utilized in the original dataset.

Feature category	No.	Feature subcategory	Description	Unit
Concrete mix ingredients	1	Water content		[kg/m^3]
	2	Cement content		[kg/m^3]
	3	Mineral admixtures content	Slag	[kg/m^3]
	4		Fly ash	[kg/m^3]
	5		Silica fume	[kg/m^3]
	6	Aggregates content	Fine aggregate	[kg/m^3]
	7		Coarse aggregate	[kg/m^3]
	8	Chemical admixtures content	Superplasticizer	[% by binder wt.]
Chloride migration	9	Migration properties	Concrete age at migration test	[Days]
	10		Migration coefficient (D_{nssm})	[x10^{-12} m^2/s]

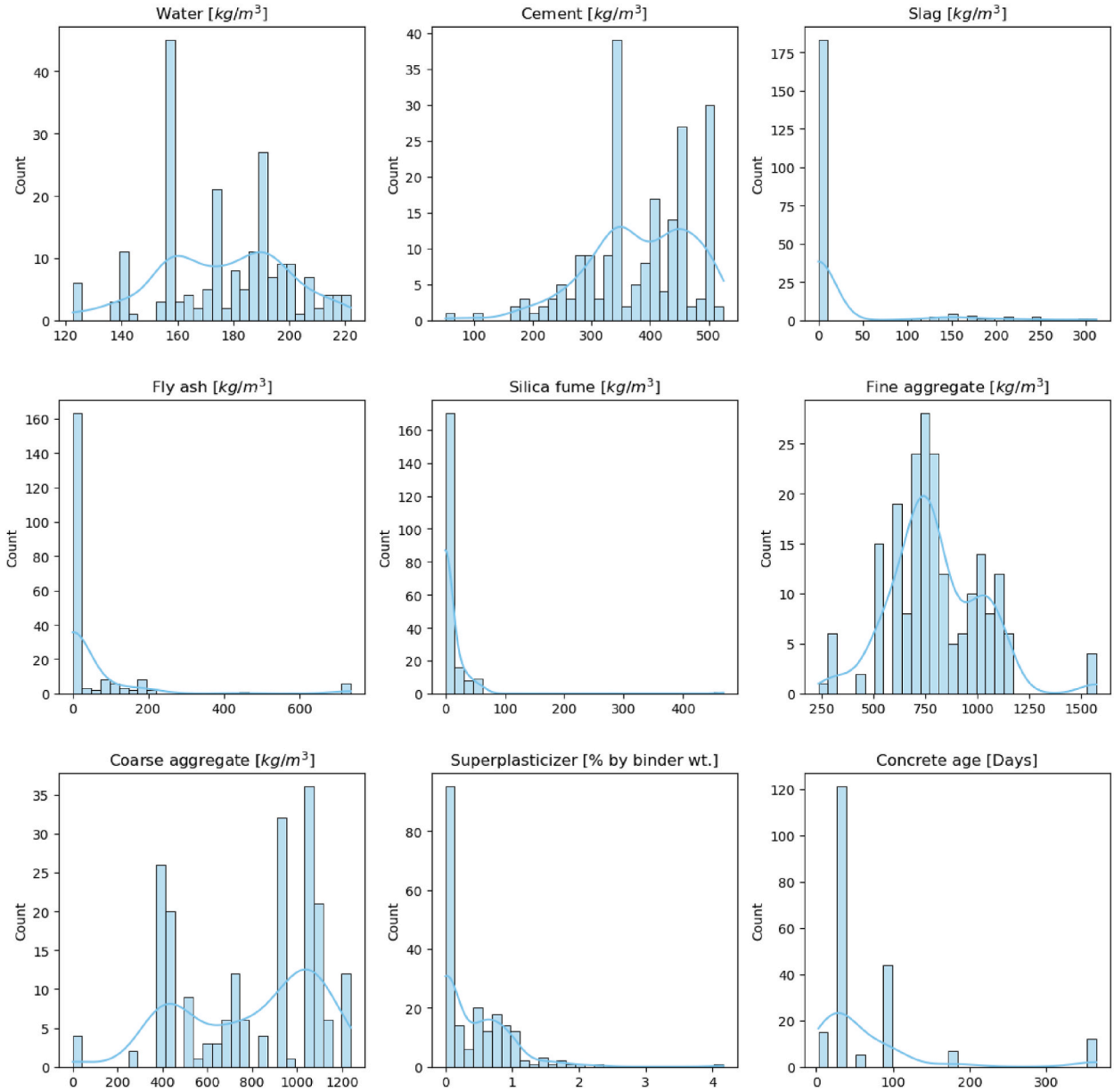

Fig. 2. Distribution of the input features of the raw dataset that controls the D_{nssm}..

training. The choice of the method for handling missing data depends on the type of missingness, the extent of missing data, and the specific research question. Although the selected algorithm, XGBoost, utilized to train the D_{nssm} prediction model incorporates mechanisms to handle missing data, the decision was made to eliminate observations with missing values. This decision was based on the fact that experimental tests for certain feature types were never conducted, rather than the missing data occurring randomly. In such cases, imputing missing values could introduce bias and lead to erroneous predictions and conclusions.

3.2.2. Detecting and treating outliers

Outliers refer to unusual observations that significantly differ from the majority of the dataset. It is crucial to identify and handle outliers in any data-driven model development process, as the quality of the data directly affects the performance of the model. While not all ML algorithms are sensitive to outliers, the chosen algorithm, XGBoost, is highly affected by them. Common approaches for outlier detection focus on each variable individually, identifying extreme observations based on the distribution of that specific variable. However, this method fails to recognize outliers that arise from relationships between multiple features. Therefore, it is important to adopt a multivariate outlier detection method that can identify uncommon combinations of scores across two or more features. There are two main types of multivariate outlier detection techniques: distance-based approaches and lower-dimensional

projection-based methods [54]. The Mahalanobis Distance (MD) is a commonly used distance measure in multivariate space, which considers both the mean and covariance of the data. It assigns larger distances to observations that deviate from the mean in directions with lower covariance. The MD between two objects, X_A and X_B, is defined by Equation (14) [43].

$$d = \left[(X_B - X_A)^T . C^{-1} . (X_B - X_A)\right]^{0.5}, \tag{14}$$

where C represents the covariance matrix of the sample.

Equation (15) in Ref. [55] demonstrates how the Mahalanobis distance can be computed from each observation to the center of the data.

$$d_i = \left[(X_i - \overline{X})^T . C^{-1} . (X_i - \overline{X})\right]^{0.5}, \tag{15}$$

where X_i refers to an object vector, and $\overline{X}$ represents an arithmetic mean vector.

In the process of identifying multivariate outliers using Mahalanobis Distance (MD), a comparison is made between the Mahalanobis distance of each instance and a threshold value derived from the chi-square distribution. In this study, any instance is categorized as a multivariate outlier if the probability associated with its Mahalanobis distance is 0.001 or less. Consequently, twelve instances meeting this criterion were identified as outliers and subsequently removed from the dataset.

3.2.3. Data partitioning

Data partitioning in ML involves dividing a dataset into separate subsets for training, validation, and testing, aiming to accurately assess the model's performance and generalization ability. The training set is utilized to train the model and validate it using *K*-fold cross-validation. On the other hand, the test set, which remains unseen during model training and validation, is employed to evaluate the model's generalization performance.

In this study, the dataset undergoes a random division, with 80% allocated for training with validation and the remaining 20% set aside for testing. His strategic allocation aims to provide the model with a substantial volume of data for learning, ensuring it can effectively capture patterns and relationships within the dataset. Simultaneously, reserving 20% of the data for testing enables the evaluation of the model's generalization capabilities on new and unseen data, offering a robust assessment of its performance during the training process.

3.3. Data after preprocessing

The preprocessed data's descriptive statistics are presented in Table 2. It is worth noting that the total number of observations has decreased to 192 from the original 204 due to the removal of 12 outlier instances. The cement content ranges from 104 to 525 kg/m^3. The SCMs, including slag, fly ash, and silica fume, have maximum values of 244 kg/m^3, 216 kg/m^3, and 60 kg/m^3, respectively. These values have relatively small standard deviations (49.28, 51.98, and 14.84 kg/m^3), indicating that they are observed only in a few instances. There are instances where the minimum amount of coarse aggregates is zero, indicating that some tests were conducted on mortar instead of concrete. These instances have been retained in the dataset to investigate their impact during the feature dependency analysis in Section 4.2. Additionally, the age of the concrete during the migration test varies from 3 to 365 days, with a standard deviation of 83.70 days. The D_{nssm} values also exhibit a wide range, spanning from 0.74 to 80.60 $\times$ 10^{-12} m^2/s, with a mean value of 10.37 $\times$ 10^{-12} m^2/s. These observations highlight the diverse range of concrete types present in the dataset.

Examining the dependencies among features is crucial because partial dependency analysis assumes independence between features when evaluating the partial effect of a specific feature on the target variable. To verify the absence of dependencies, the Pearson correlation coefficients are calculated between all possible feature pairs. The formula for calculating the Pearson correlation coefficient is given in Equation (16). The Pearson correlation coefficient, denoted by " r" and ranging from −1 to +1, measures the strength and direction of the linear relationship between two continuous variables. A value of +1 indicates a complete positive correlation, −1 represents a complete negative correlation, and 0 indicates no correlation [56]. Fig. 3 presents the Pearson correlation coefficient

Table 2
Descriptive statistics of the preprocessed data.

Units	Input features									Target feature
	Water	Cement	Slag	Fly ash	Silica fume	Fine aggregate	Coarse aggregate	Superplasticizer	Concrete age	D_{nssm}
	[kg/m^3]	[kg/m^3]	[kg/m^3]	[kg/m^3]	[kg/m^3]	[kg/m^3]	[kg/m^3]	[% by binder wt.]	[Days]	[x10^{-12} m^2/s]
Count	192	192	192	192	192	192	192	192	192	192
Mean	176.49	383.60	15.34	21.67	6.44	788.92	799.58	0.41	64.10	10.37
Std	22.72	86.65	49.28	51.98	14.84	215.64	300.78	0.48	79.48	10.50
Min	122.50	104.00	0.00	0.00	0.00	235.00	0.00	0.00	3	0.74
25%	158.00	330.00	0.00	0.00	0.00	645.00	451.50	0.00	28	5.11
50%	178.20	391.00	0.00	0.00	0.00	760.50	915.53	0.20	28	7.50
75%	191.25	450.00	0.00	0.00	0.00	959.00	1061.00	0.70	90	10.50
Max	222.00	525.00	244.00	216.00	60.00	1574.10	1240.00	2.27	365	80.60

values depicting the relationships between the considered features in the dataset. It is apparent that most features display minimal correlation. However, certain features, such as fine and coarse aggregate with $r = -0.43$, exhibit a moderate negative correlation. This coefficient suggests that the variables tend to move in opposite directions in a linear manner, although the relationship may not be very strong. In other words, an increase in the content of fine aggregate is associated with a decrease in the content of coarse aggregate, and vice versa, but not necessarily in a perfectly proportional manner. It is important to note that the moderate correlation between these two features is due to specific concrete design specifications rather than an inherent relationship between the aggregates themselves, as both are distinct components used in concrete to achieve desired properties. The same holds true for other features that show moderate correlation, such as silica fume and slag, as well as cement and slag.

$$r = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}, \tag{16}$$

where x_i and y_i represent the values of the two variables being analyzed, and $\bar{x}$ and $\bar{y}$ represent the means of the respective variables.

3.4. Model training and evaluation

The D_{nssm} prediction model was implemented by training the XGBoost algorithm using Python's scikit-learn library. This achievement was realized by leveraging input and target attributes from preprocessed data. The training dataset, which accounted for 80% of the data, was employed to train the model. To enhance the model's performance, a combination of random search and K-fold cross-validation was utilized to tune the hyperparameters. In this method, the dataset was divided into K equally-sized subsets or folds, and the hyperparameters were randomly sampled from a predefined range. For each set of hyperparameters, the model was trained on $K - 1$ folds and evaluated on the remaining fold. This process was repeated K times, with each fold serving as the validation set once. The results were then averaged to estimate the model's performance, reducing variance and providing a more reliable evaluation of its generalization capability. The adopted value of K in this study is 10, which is a standard practice in machine learning [57]. Table 3 presents an overview of all the hyperparameters considered during the training process, along with their respective ranges. For a comprehensive understanding of each hyperparameter, detailed information is available in the Python scikit-learn library's documentation [58].

After the algorithm is trained, its ability to predict D_{nssm} using a test dataset, which was not previously seen during the training phase, needs to be assessed using statistical metrics. The commonly used metrics for evaluating the accuracy of regression models include mean-square error (MSE), root-mean-square error (RMSE), mean-absolute error (MAE), and coefficient of determination (R^2) [59].

The MSE computes the average of the squared disparities between predicted and actual values, as shown in Equation (17). This metric quantifies the overall variability or dispersion of errors, with smaller values indicating superior predictive performance.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{17}$$

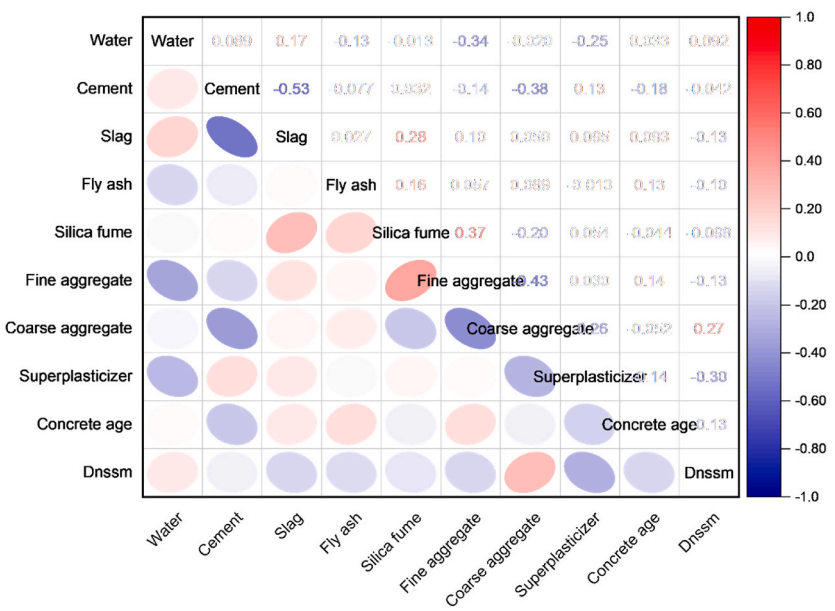

Fig. 3. The Pearson correlation coefficients of all the features.

Table 3
The hyperparameters taken into account in the XGBoost algorithm.

Hyperparameters	Search space	Description
n_estimators	[10, 20, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 600, 700, 800, 900, 1000]	Number of gradient boosted trees
max_depth	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]	Maximum tree depth for base learners
min_split_loss	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20]	Minimum loss reduction required to make a further partition on a leaf node of the tree.
learning_rate	[1e-05, 0.0001, 0.001, 0.01, 0.1, 1.0]	Boosting learning rate.
booster	['gbtree','gblinear','dart']	Booster types.

The RMSE is the square root of the mean of the squared disparities between predicted and actual values, as described in Equation (18). The RMSE has the advantage of being expressed in the same unit as the dependent variable, which facilitates interpretation. Like the mean square error (MSE), lower RMSE values indicate superior predictive performance.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} = \sqrt{MSE}, \tag{18}$$

The MAE computes the average of the absolute disparities between predicted and actual values, as stated in Equation (19). The MAE offers an assessment of the average magnitude of errors, regardless of their direction. Lower MAE values indicate superior predictive performance. Unlike other error metrics that involve squaring the differences (e.g., MSE), MAE treats all errors equally, assigning equal importance to both overestimation and underestimation.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{19}$$

The R^2 value signifies the proportion of variance in the response feature that can be explained by the regression model. It is regarded as a standardized version of MSE, providing enhanced interpretability regarding the model's performance. Ranging from 0 to 1, a value of 0 indicates no explanatory power, while a value of 1 represents a perfect fit. The R^2 value is computed using Equation (20).

$$R^2 = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{MSE}{Var\ (y)}, \tag{20}$$

In these equations, n represents the number of observations. The variable y_i denotes the actual target value, while $\hat{y}_i$ represents the predicted output value. $\bar{y}$ represents the mean value of the actual target, and Var represents the variance.

4. Results and discussion

In this section, the emphasis is on presenting and analyzing the performance of the D_{nssm} prediction model. A comprehensive grasp of the prediction process for the algorithms is attained through the application of SHAP, a Python library that utilizes Shapley values to clarify the results of any machine learning model. This interpretation encompasses two main aspects: the global explanation, which offers an understanding of the model's overall behavior, and the instance-based explanations, which explore specific and detailed

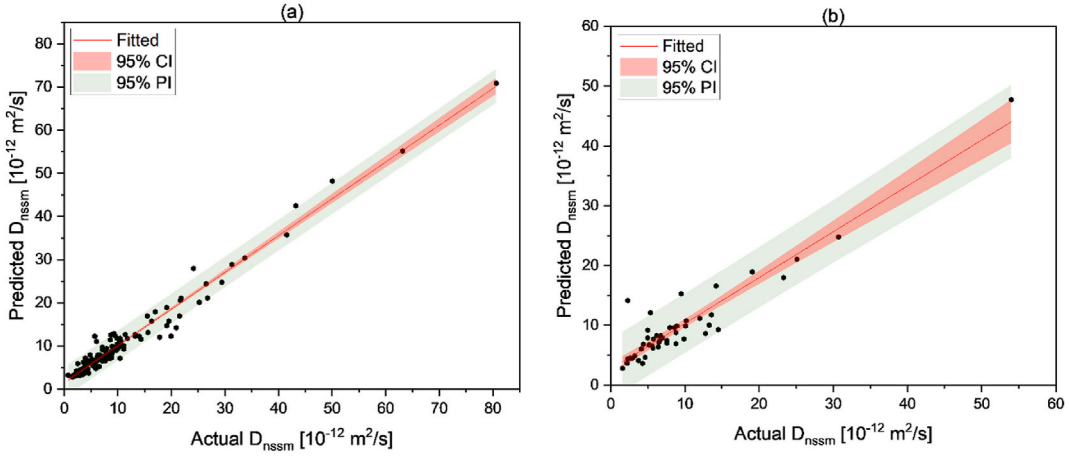

Fig. 4. Regression plots of the D_{nssm} prediction model (a)training phase, and (b) test phase.

reasons for individual predictions. Additionally, the evaluation involves assessing the partial effect of specific features on the target variable, thereby enhancing the understanding of the model's predictive capabilities.

4.1. Model performance evaluation

Fig. 4 illustrates the training and test performance of the model using a regression plot, which showcases the linear relationship between the actual and predicted D_{nssm} values. The plot reveals that the predicted values closely align with the regression line, which represents the best-fit line through the data points during the training phase. This indicates that the model successfully captured the patterns in the data during training. During the test phase, the model continues to perform well, although not as well as during training. The R-square value, a measure of the model's goodness of fit, is 0.95 during training and 0.87 during the test phase. These values indicate a strong correlation between the predicted and actual D_{nssm} values, affirming the model's effectiveness in both phases.

The model's performance on both the training and test datasets is evaluated using statistical measures, including MSE, RMSE, and MAE. In the training phase, the model achieved an MAE of 1.62, MSE of 6.65, and RMSE of 2.38. In the test phase, the corresponding values were MAE = 2.22, MSE = 10.12, and RMSE = 3.18. These metrics provide insights into the accuracy and performance of the model on both training and test data. To attain this level of accuracy, optimal hyperparameters were identified from the hyperparameter search space presented in Table 3. The identified optimal hyperparameters are as follows: n_estimators = 600, max_depth = 4, min_split_loss = 20, learning_rate = 0.01, and booster = dart.

Fig. 5 provides a comparison of the performance between the current model and an earlier model developed by the authors using the same algorithm and dataset. The primary differences lie in the choice of input features and the hyperparameters, as well as the hyperoptimization method employed. The current work utilized only the constituents used to produce the dataset without transforming them into new features, aiming to eliminate feature interdependence. The prior work utilized grid search with *K*-fold cross-validation, while this work employed random search with *K*-fold cross-validation techniques. Grid search systematically explores all possible combinations of predefined hyperparameter values within a specified search space, ensuring a thorough examination but potentially becoming computationally expensive, especially for larger search spaces. On the other hand, random search takes a more flexible approach by randomly sampling a specified number of hyperparameter combinations from the search space. This randomness can be more efficient, particularly when only a subset of hyperparameters significantly influences model performance, and computational resources are limited. The shift to random search seeks to strike a balance between exploration and efficiency. The figure clearly depicts that the MSE, RMSE, and MAE of the earlier work that closely correspond to the considered feature types are slightly lower than those of this work. However, the R^2 of this work is considerably higher than the previous work. These results affirm that the model developed in this study performs favorably despite employing fewer features and abstaining from feature engineering. Notably, the adoption of random search for hyperparameter optimization, in lieu of the exhaustive grid search, attests to the model's robustness and efficiency in achieving competitive performance.

4.2. Model explainability

Fig. 6 illustrates the visualization of the mean absolute SHAP values for the input features. The significance of a feature in predicting the D_{nssm} can be determined by its corresponding mean absolute SHAP value, with higher values indicating greater importance. Upon examining the figure, it becomes evident that the amount of coarse aggregate feature holds the highest importance, followed by superplasticizer, and concrete age. The mean absolute SHAP value for coarse aggregate is 2.34, which accounts for about 25% of the cumulative mean absolute SHAP value for all features. Following the top three, the features cement and water are ranked fourth and

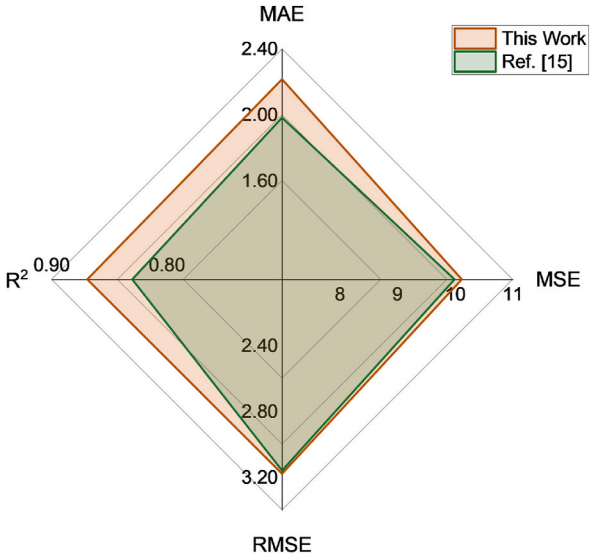

Fig. 5. A performance comparison between the developed D_{nssm} prediction model with another model.

fifth in importance, respectively. The influence of cement on concrete properties is typically interdependent on the amount of water used, and thus, considering them individually may not be fully indicative. Indeed, the cumulative mean absolute SHAP resulting from their collective contributions represents a noteworthy portion, approximately 23%, of the total mean absolute SHAP values. When considering the five key features - coarse aggregate, superplasticizer, concrete age, cement, and water - together, they account for approximately 83% of the total mean absolute SHAP values. This cumulative importance underscores their substantial role in explaining the model's output and capturing the intricate relationships between these features and the predicted outcome. On the other hand, the features slag and fly ash exhibit the least influence in predicting the D_{nssm}. By excluding these two features, the remaining features account for 98% of the total mean absolute SHAP values, indicating that the model can make accurate predictions even without considering them. However, it is important to acknowledge that the insignificant mean absolute SHAP values of slag and fly ash do not necessarily imply their complete uselessness in predicting D_{nssm}. This assertion holds true only for this particular dataset, in which the number of concrete instances containing slag and fly ash is limited, as discussed in Section 3.3. To gain a more comprehensive understanding of their influence, it is crucial to analyze the explainability of the model's predictions through diverse plots.

To gain insights into the D_{nssm} model's predictions on an individual basis, a summary plot, depicted in Fig. 7, is created. This plot combines feature importance with feature effects for each instance. Each point on the plot represents a SHAP value associated with a feature and an instance. The y-axis denotes the feature, while the x-axis represents the SHAP value. The color of the points represents the corresponding feature value, ranging from low to high. To present the distribution of SHAP values per feature, overlapping points are slightly adjusted in the y-axis direction, ensuring better visibility and clarity.

Similar to Fig. 6, the arrangement of features in Fig. 7 is based on their importance. However, Fig. 7 offers additional insights that are not readily apparent in Fig. 6's feature importance plot. For example, concrete age, at lower ages, the SHAP value is high and positive. However, as the age increases beyond a certain point, the SHAP value becomes negative, indicating a gradual increase in D_{nssm} that is not as significant as observed at earlier ages. Regarding coarse aggregate, there are three instances with exceptionally high SHAP values at around 27, 30, and 32. The average of these three values is four times higher than the maximum SHAP value of the next lower value in this group. Consequently, it becomes evident that these three extreme SHAP values contribute significantly, making the mean absolute SHAP value of the coarse aggregate feature higher compared to all other features. These observations are derived from a study [37]. Examining the mix ingredients reported in their study reveals no unusual ingredients or proportions. However, the existence of such extreme values prompts further scrutiny to validate the accuracy of these specific observations, as there may be errors in the experiment or in reporting the D_{nssm} values. If these values are omitted, the feature "coarse aggregate" no longer maintains the highest importance. Therefore, the utilization of the summary plot underscores that depending solely on the feature importance plot is inadequate. Hence, such a summary plot is vital for obtaining a more thorough understanding of the influence of features.

The SHAP value analysis reveals that water has a positive impact on D_{nssm} when its magnitude is high and a negative impact when its magnitude is low. On the other hand, the opposite pattern can be observed with cement, where a medium to lower amount of cement leads to a higher SHAP value and a reduction in D_{nssm}. These trends are as anticipated since higher cement content and lower water content reduce porosity and enhance impermeability, resulting in a decrease in D_{nssm}. Conversely, when examining the fine aggregate, there is no distinct pattern observed in many cases. However, in several instances, a higher magnitude of fine aggregate corresponds to negative SHAP values, indicating a reduction in D_{nssm}. This phenomenon might arise due to the increase in fine aggregate, which in turn reduces the amount of used coarse aggregate and affects the interfacial zone. Nevertheless, it is important to note that this feature is typically not the primary factor controlling chloride ingress.

It is also important to note that the use of silica fume, slag, and superplasticizer results in negative SHAP values, impacting the D_{nssm}

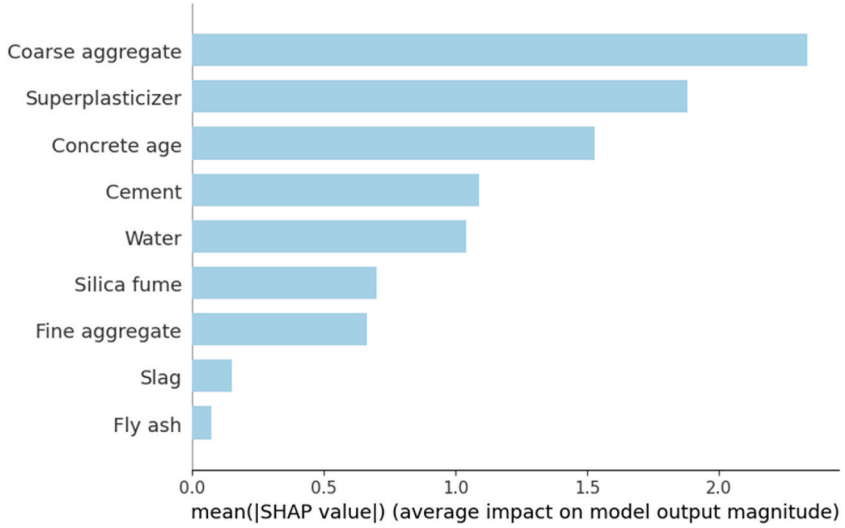

Fig. 6. Feature importance plot of the best performing model.

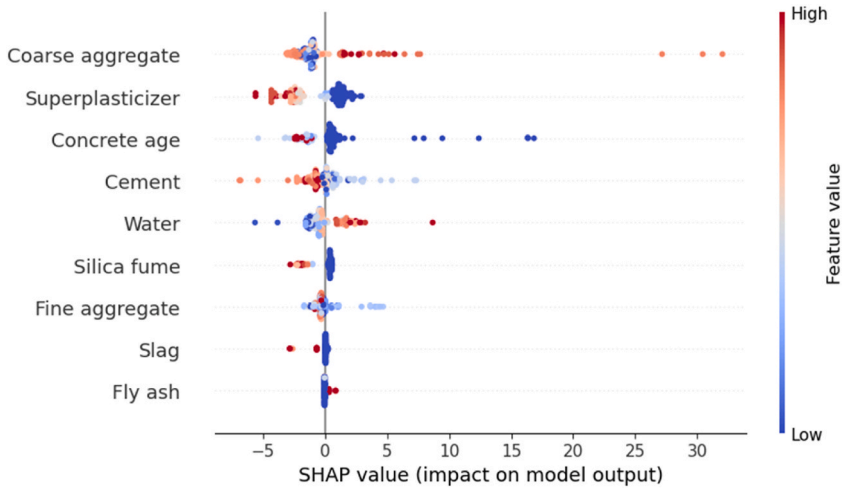

Fig. 7. SHAP summary plot of the best performing model.

of concrete to decrease. This aligns with expectations since these admixtures increase the density of the concrete matrix, reduce its permeability, and improve resistance to D_{nssm}. Although the influence of fly ash and slag may appear insignificant in impacting the D_{nssm} of concrete according to Fig. 7, it is clear that the impact of slag is significantly higher than that of fly ash. The reason why slag and fly ash appear negligible in Fig. 6 could be attributed to the limited number of concrete mixtures employing them compared to other SCMs, resulting in a lower mean absolute SHAP value and potentially downplaying its importance.

The SHAP waterfall plots, presented in Fig. 8, offer a detailed examination of how individual D_{nssm} predictions are derived by the model. On the Y-axis, features are encoded, showcasing the values observed for a particular observation. Meanwhile, the X-axis represents the range of D_{nssm} in $\times 10^{-12} m/s^2$. The baseline of the waterfall plot commences with the expected value of the model output $E[f(x)] = 11.014 \times 10^{-12} m/s^2$, representing the global predicted average D_{nssm}. Each row in the waterfall plot illustrates how

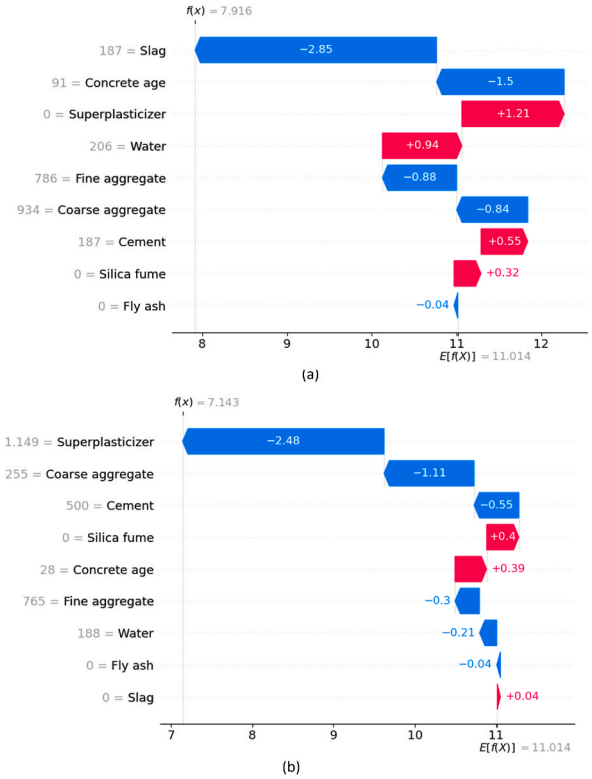

Fig. 8. SHAP waterfall plot of selected instances.

the positive contributions (depicted in red) or negative contributions (depicted in blue) of each feature influence the value, transitioning from the expected model output over the background dataset to the model output for the specific prediction under consideration. The top of the charts displays the model prediction, $f(x)$, for the observed values in the particular instances being analyzed.

The selected observations in Fig. 8(a) and (b) are aimed at enhancing our understanding of the influence of SCMs and superplasticizers, respectively, on D_{nssm} predictions, since their impact were not adequately observed in other plots. In Fig. 8(a), the chosen observation utilizes slag as an SCM, and its impact on D_{nssm} prediction stands out significantly compared to intermediate values observed for other features. The impact is notably negative, indicating a substantial contribution to reducing the D_{nssm} of concrete. Since this instance exclusively employs only slag as the SCM, the impact of other SCMs (silica fume and fly ash) appears insignificant. Similarly, in Fig. 8(b), where the considered concrete employs superplasticizers, the impact on D_{nssm} prediction is notably high, exceeding intermediate values observed for other features. Again, the impact is negative, signifying a significant contribution to reducing the D_{nssm} of concrete. This visual representation offers a clear and insightful portrayal of how each feature contributes to the model's predictions. It highlights the limitation of relying solely on global-level explanations and underscores the importance of

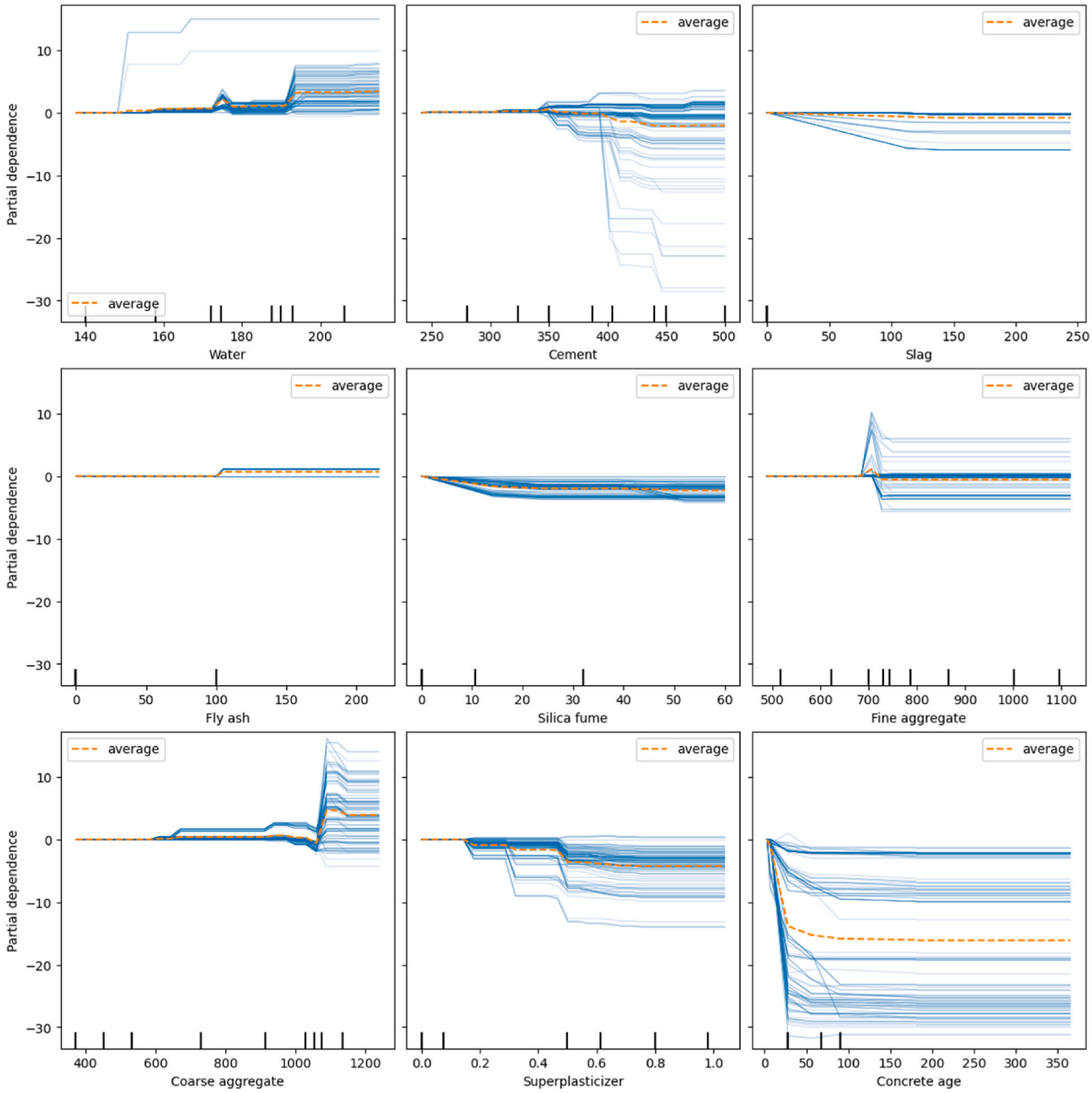

Fig. 9. ICE plots of all the input features utilized in the D_{nssm} prediction model.

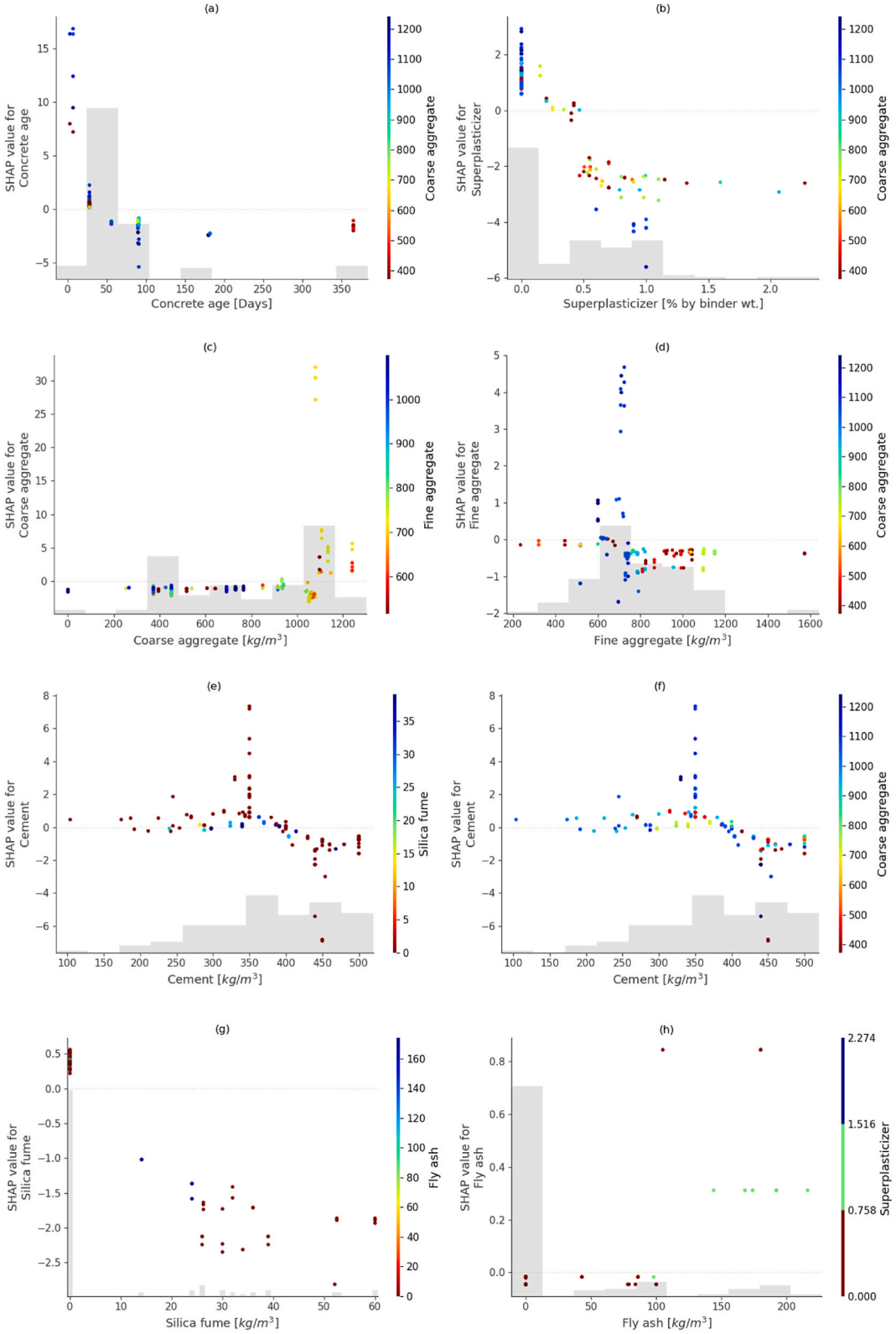

Fig. 10. SHAP dependence plots for the input features.

conducting local-level analyses to gain more nuanced and insightful results.

4.3. Partial dependency of features

Fig. 9 displays the individual conditional expectation (ICE) plots, offering a visual representation of the output variation of the D_{nssm} prediction model concerning all nine input features. These plots demonstrate the model's prediction dependency on the features for each sample individually, with each sample depicted by a separate line. In each ICE plot, the values of the remaining input features are held constant, while the variation of one input feature is depicted. Moreover, all the subplots incorporate the corresponding partial dependence line, which overlays the average effect of the input feature on the ICE lines.

Upon examining the graphs, it becomes evident that the partial dependencies (averages of the ICE lines) for the features "cement" and "superplasticizers" exhibit a decrease within specific ranges of their values. The impact of these features appears to be insignificant when their values fall below a certain threshold (350 kg/m^3 for "cement" and 0.1% by binder wt. for "superplasticizers") or exceed certain values (450 kg/m^3 for "cement" and 0.6% by binder wt. for "superplasticizers") in predicting D_{nssm}. Additionally, the study reveals that the age of the concrete significantly influences D_{nssm}, particularly during its early stages. However, as the concrete ages, this impact gradually diminishes, stabilizing after reaching 90 days. In the case of the features "water" and "coarse aggregate" exhibit a similar trend. Beyond certain threshold values, the partial dependencies increase, indicating that higher magnitudes of these features correspond to higher D_{nssm} values in the concrete. However, the feature "slag" exhibits a nearly horizontal trend around a partial dependency values of zero. This can be attributed to many cases in the dataset having a value of zero for slag. As a result, the overall influence of "slag" on the model's output is considered insignificant, despite its significant contribution to decreasing the D_{nssm} of concrete as demonstrated in Fig. 8(a). All these analysis highlights the varying impacts of different features on D_{nssm} prediction, with certain features showing clear trends while others have limited influence or only apply within specific value ranges. Understanding these relationships is crucial for optimizing concrete mixtures for desired performance.

4.4. SHAP dependency with feature interaction

Fig. 10 depicts the SHAP dependence plots for the input features across the entire dataset. These plots resemble partial dependence plots but consider the presence of interaction effects within the features. The vertical spread of SHAP values at a specific feature value is influenced by interaction effects, and another feature is selected for coloring to emphasize potential interactions.

Fig. 10(a) provides clear evidence that the SHAP value for concrete ages between 2 and 28 days is negative, whereas for ages above 28 days, it becomes positive. Additionally, the SHAP value experiences a sharp decrease from 3 days until about 90 days, followed by a more gradual decline until reaching a negative value at 28 days. This phenomenon supports the notion that the D_{nssm} of concrete is initially high at early ages but decreases with time, regardless of the concrete's composition, throughout the considered period of 365 days. This could be due to the fact that, at early age concrete is still in the initial stages of hydration, and the microstructure may contain more interconnected and larger pores. This increased porosity allows chloride ions to move more freely through the concrete, resulting in higher migration coefficients. As the concrete matures and continues to hydrate over time, the pore structure becomes more refined and the cementitious matrix denser. This reduction in pore connectivity and size leads to a decrease in the D_{nssm}.

As shown in Fig. 10(b), when the superplasticizer's quantity exceeds about 0.45% of the binder weight till the max amount considered in the training dataset (about 2.3% of the binder weight till), it leads to negative SHAP values. Nonetheless, incorporation of small amounts of superplasticizer (less than about 0.45% of the binder weight) in a concrete mix with a medium proportion of coarse aggregate increase the D_{nssm}. The effect of superplasticizer on the D_{nssm} is not always straightforward and can depend on its dosage and the specific circumstances of each concrete mixture.

As demonstrated in Fig. 10(c), the amount of coarse aggregate has a significant impact on the D_{nssm} of concrete. Specifically, within the range of 1080 kg/m^3 to 1240 kg/m^3, positive SHAP values for coarse aggregate indicate that a higher quantity of coarse aggregate is associated with a higher D_{nssm}. This phenomenon arises because a greater proportion of coarse aggregate leads to increased porosity within the concrete, creating more interconnected voids or pathways for chloride ions to migrate through. Furthermore, the interface between the coarse aggregate and the cement paste can serve as potential paths for chloride penetration. These factors collectively contribute to a higher D_{nssm} in concrete with a higher proportion of coarse aggregate. This discovery regarding the significance of coarse aggregate content aligns with the findings of [60,61]. However, it is worth noting that the degree of importance of aggregate content may diminish with prolonged concrete age, as demonstrated by Refs. [62,63]. Regarding the fine aggregate, the SHAP values predominantly cluster around zero, but there are a few exceptions. Notably, Fig. 10(d) shows instances with SHAP values higher than three, which stand out significantly compared to the rest of the data. The reason behind these elevated SHAP values is not immediately evident. However, as presented in Section 4.2, upon conducting further investigation, it was discovered that all these observations are derived from a single source [37]. Despite this finding, a clear explanation for these observations still remains elusive and requires more in-depth analysis.

As depicted in Fig. 10(e), when the cement content is higher (400 kg/m^3), the SHAP values become negative. This outcome is as expected since higher cement content in concrete leads to lower porosity and reduced permeability. Consequently, the movement of chloride ions through the concrete is impeded, resulting in a lower chloride migration coefficient. Interestingly, there are instances where the cement content is lower than 400 kg/m^3, yet negative SHAP values with small magnitude were observed. In contrast to the previous case (cement content above 400 kg/m^3), many of these instances involve the SCMs like slag, fly ash, or silica fume in the concrete mix. The probable explanation for this observation is that the combination of cement with these SCMs contributes to the formation of additional calcium silicate hydrate (C–S–H) gel and reduces the calcium hydroxide content. This results in a denser microstructure with reduced pore size and improved interconnectivity, leading to lower D_{nssm}. The interaction effect between cement and coarse aggregate fluctuates at different content levels. Notably, when the coarse aggregate content is around 730 kg/m^3, the SHAP

values exhibit a pattern of being small yet consistently positive within the cement range of 300–370 kg/m^3, as illustrated in Fig. 10(f).

Though SCMs generally have a positive effect on the D_{nssm}, the exact impact may vary depending on the specific mix proportions. For instance, all the SHAP value of silica fume values are negative, indicating their dosage in the concrete mix considered in the dataset reduces the D_{nssm} values as shown in Fig. 10(g). Nevertheless, the SHAP value for fly ash is negative when its quantity is less than 100 kg/m^3, but it becomes positive when this value exceeds 100 kg/m^3 as revealed in Fig. 10(h). It can also be seen that fly ash containing concrete is also impacted with the amount of the employed superplasticizer amount particularly when its dosage become above exceeds 100 kg/m^3.

The employed XML technique (XGBoost coupled with SHAP) offers comprehensive explanations regarding the influence of individual concrete mix ingredients on D_{nssm}, both at a global and local level. Moreover, it reveals valuable insights into the relationships and dependencies between different features. The dataset used in this research comprises various concrete types, such as lightweight, normal-weight, high-strength, high-performance, and self-compacting concretes. Despite this diversity, the adopted method enables the identification of common patterns, enhancing the robustness of the insights compared to those derived from experimental data, which are typically limited by a small number of experiments. It's worth noting that the dataset includes experiments investigating D_{nssm} only up to a one-year period and is specific to certain concrete types, which means that all the reported findings are constrained by the considered age and concrete varieties.

The inherent explainability of the XML allows for the acquisition of novel insights when supplemented with extensive data encompassing diverse concrete types and long-term D_{nssm} performance. While a large dataset is valuable, a more comprehensive understanding could be achieved by considering the detailed chemical composition of both cement and SCMs types. In this study, cement and SCMs were considered based on their general types, recognizing that even within the same types, the chemical composition may vary depending on their sources. The next phase of exploration should prioritize incorporating more specific chemical details. Despite numerous experimental studies, there remains a scarcity of open-access data repositories focused on concrete science. To leverage the benefits of rapidly evolving AI technologies in the concrete industry, there is a suggestion to develop an open data exchange platform. Such a platform would enable the scientific and concrete communities to share data in a standardized format. Looking ahead, initiatives on an international scale, such as those led by technical committees like ACI [64] and RILEM [65], may pave the way for the establishment of repositories dedicated to this purpose. Such repositories would significantly enhance the availability of broader and more diverse datasets, ultimately fostering the development of more robust and accurate models for generalized knowledge inference. This, in turn, could facilitate the creation of optimal and sustainable concrete with improved resistance to chloride penetration and other crucial characteristics. The potential impact of this data-sharing initiative extends beyond chloride resistance, potentially leading to advancements in various aspects of concrete research and development.

The proposed method possesses the capability to dynamically improve its performance by assimilating new relevant data. This adaptability ensures that the model remains current and can make accurate and precise predictions as new information becomes accessible. Additionally, the model's explainability empowers engineers and other stakeholders by offering a transparent view of the underlying algorithm and the reasoning behind its predictions. This transparency fosters trust in the model's predictions and facilitates its seamless integration into real-world applications. Additionally, there are economic implications as it allows for the determination of D_{nssm} of various concrete materials without the need for labor and resource-intensive advanced laboratory testing. In fact, this approach can be extended to develop XML based models for predicting the chloride diffusivity of concrete performed by different procedures and other time-consuming and resource-intensive tests.

5. Conclusions

The impact of concrete mix ingredients on the non-steady chloride migration coefficient was performed using explainable machine learning method that integrated XGBoost and SHAP. The conclusions are as follows:

- Performance: The XGBoost model utilized in training exhibited excellent accuracy in predicting the D_{nssm} of concrete. In the training phase, the performance metrics showed an R^2 score of 0.95, MAE of 1.61×10^{-12} m^2/s, and RMSE of 2.38×10^{-12} m^2/s. In the test phase, the model achieved an R^2 score of 0.87, MAE of 2.22×10^{-12} m^2/s, and RMSE of 3.18×10^{-12} m^2/s.
- Impacts of features: The SHAP analyses underscore the varied impacts of different features on D_{nssm} prediction, with some features showing clear trends and others having limited influence or applying only within specific value ranges. For example, within the range of 1080 kg/m^3 to 1240 kg/m^3, D_{nssm} remains consistently high. Higher cement quantities (above 400 kg/m^3) result in negative SHAP values, but even below 400 kg/m^3, negative SHAP values are evident due to the impact of SCMs in reducing D_{nssm}. The age of the concrete also significantly affects D_{nssm}, with a sharp decrease from 3 days to approximately 90 days, followed by a more gradual decline until reaching a negative value at 28 days. Fly ash's SHAP value becomes negative when used below 100 kg/m^3, with the effect magnitude influenced by the amount of superplasticizers employed.
- Explanaibility: The visualization of SHAP values via diverse plots is crucial for gaining a comprehensive understanding of the overall feature importance in the model's predictions. Additionally, it enables a detailed breakdown of how each feature contributes to specific predictions for individual samples, providing transparency and insight into the model's decision-making process.
- Limitation: The data utilized to train the XGBoost algorithm for predicting D_{nssm} of concrete is limited to a duration of one year. Additionally, there are only a few observations that involved SCMs. As a result, all the reported findings are applicable solely to the concrete age and the range of concrete mix ingredients considered in this study, and they cannot be generalized to other scenario.

CRediT authorship contribution statement

Woubishet Zewdu Taffese: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Leonardo Espinosa-Leal:** Funding acquisition, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset used in this work is provided as a supplementary file in the article referenced [1].

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jobe.2023.108370.

References

[1] W.Z. Taffese, L. Espinosa-Leal, A machine learning method for predicting the chloride migration coefficient of concrete, Construct. Build. Mater. 348 (2022), 128566, https://doi.org/10.1016/j.conbuildmat.2022.128566.
[2] H. Ye, X. Jin, C. Fu, N. Jin, Y. Xu, T. Huang, Chloride penetration in concrete exposed to cyclic drying-wetting and carbonation, Construct. Build. Mater. 112 (2016) 457–463, https://doi.org/10.1016/j.conbuildmat.2016.02.194.
[3] C. Andrade, D. Izquierdo, Statistical treatments of chloride threshold and corrosion propagation rate, Corros. Mater. Degrad. 3 (2022) 598–611, https://doi.org/10.3390/cmd3040032.
[4] R. Singh, Corrosion Control for Offshore Structures: Cathodic Protection and High Efficiency Coating, first ed., Gulf Professional Publishing, Oxford, UK, 2015 https://doi.org/10.1016/C2012-0-01231-8.
[5] V. Elfmarkova, P. Spiesz, H.J.H. Brouwers, Determination of the chloride diffusion coefficient in blended cement mortars, Cement Concr. Res. 78 (2015) 190–199, https://doi.org/10.1016/j.cemconres.2015.06.014.
[6] J. Pontes, J.A. Bogas, S. Real, A. Silva, The rapid chloride migration test in assessing the chloride penetration resistance of normal and lightweight concrete, Appl. Sci. 11 (2021) 7251, https://doi.org/10.3390/app11167251.
[7] K. Audenaert, Q. Yuan, G. De Schutter, On the time dependency of the chloride migration coefficient in concrete, Construct. Build. Mater. 24 (2010) 396–402, https://doi.org/10.1016/j.conbuildmat.2009.07.003.
[8] Life-365™ Consortium III, Life-365™ Service Life Prediction Model™ Version and Computer Program for Predicting the Service Life and Life-Cycle Cost of Reinforced Concrete Exposed to Chlorides, 2020.
[9] DuraCrete, DuraCrete Final Technical Report: Probabilistic Performance Based Durability Design of Concrete Structures, 2000.
[10] ASTM C1556 - 11a, Standard Test Method for Determining the Apparent Chloride Diffusion Coefficient of Cementitious Mixtures by Bulk Diffusion, ASTM, West Conshohocken, PA, 2016.
[11] NT BUILD 443, Concrete, Hardened: Accelerated Chloride Penetration, NORDTEST, 2010.
[12] NT BUILD 492, Concrete, Mortar and Cement-Based Repair Materials: Chloride Migration Coefficient from Non-steady-state Migration Experiments, NORDTEST, 1999.
[13] S.E. Chidiac, M. Shafikhani, Phenomenological model for quantifying concrete chloride diffusion coefficient, Construct. Build. Mater. 224 (2019) 773–784, https://doi.org/10.1016/j.conbuildmat.2019.07.006.
[14] G. Sun, W. Sun, Y. Zhang, Z. Liu, Multi-scale modeling of the effective chloride ion diffusion coefficient in cement-based composite materials, J. Wuhan Univ. Technol.-Materials Sci. Ed. 27 (2012) 364–373, https://doi.org/10.1007/s11595-012-0467-6.
[15] J.A. Bogas, A. Gomes, Non-steady-state accelerated chloride penetration resistance of structural lightweight aggregate concrete, Cem. Concr. Compos. 60 (2015) 111–122, https://doi.org/10.1016/j.cemconcomp.2015.04.001.
[16] W.Z. Taffese, F. Al-Neshawy, E. Sistonen, M. Ferreira, Optimized neural network based carbonation prediction model, in: International Symposium Non-Destructive Testing in Civil Engineering, 2015, pp. 1074–1083.
[17] S. Pan, Z. Zheng, Z. Guo, H. Luo, An optimized XGBoost method for predicting reservoir porosity using petrophysical logs, J. Pet. Sci. Eng. 208 (2022), 109520, https://doi.org/10.1016/j.petrol.2021.109520.
[18] A. Lavercombe, X. Huang, S. Kaewunruen, Machine learning application to eco-friendly concrete design for decarbonisation, Sustainability 13 (2021), 13663, https://doi.org/10.3390/su132413663.
[19] W.Z. Taffese, L. Espinosa-Leal, Prediction of chloride resistance level of concrete using machine learning for durability and service life assessment of building structures, J. Build. Eng. 60 (2022), 105146, https://doi.org/10.1016/j.jobe.2022.105146.
[20] T.A. Pham, V.Q. Tran, H.-L.T. Vu, H.-B. Ly, Design deep neural network architecture using a genetic algorithm for estimation of pile bearing capacity, PLoS One 15 (2020), e0243030, https://doi.org/10.1371/journal.pone.0243030.
[21] W.Z. Taffese, K.A. Abegaz, Artificial intelligence for prediction of physical and mechanical properties of stabilized soil for affordable housing, Appl. Sci. 11 (2021) 7503, https://doi.org/10.3390/app11167503.
[22] T.-A. Nguyen, H.-B. Ly, B.T. Pham, Backpropagation neural network-based machine learning model for prediction of soil friction angle, Math. Probl Eng. 2020 (2020) 1–11, https://doi.org/10.1155/2020/8845768.
[23] W.Z. Taffese, K.A. Abegaz, Prediction of compaction and strength properties of amended soil using machine learning, Buildings 12 (2022) 613, https://doi.org/10.3390/buildings12050613.
[24] N.-D. Hoang, C.-T. Chen, K.-W. Liao, Prediction of chloride diffusion in cement mortar using multi-gene genetic programming and multivariate adaptive regression splines, Measurement 112 (2017) 141–149, https://doi.org/10.1016/j.measurement.2017.08.031.
[25] L. Yao, L. Ren, G. Gong, Evaluation of chloride diffusion in concrete using PSO-BP and BP neural network, IOP Conf. Ser. Earth Environ. Sci. 687 (2021), 012037, https://doi.org/10.1088/1755-1315/687/1/012037.
[26] J.M.P.Q. Delgado, F.A.N. Silva, A.C. Azevedo, D.F. Silva, R.L.B. Campello, R.L. Santos, Artificial neural networks to assess the useful life of reinforced concrete elements deteriorated by accelerated chloride tests, J. Build. Eng. 31 (2020), 101445, https://doi.org/10.1016/j.jobe.2020.101445.

[27] V. Quan Tran, Machine learning approach for investigating chloride diffusion coefficient of concrete containing supplementary cementitious materials, Construct. Build. Mater. 328 (2022), 127103, https://doi.org/10.1016/j.conbuildmat.2022.127103.
[28] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 4768–4777. https://github.com/slundberg/shap.
[29] M. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, Association for Computational Linguistics, Stroudsburg, PA, USA, 2016, pp. 97–101, https://doi.org/10.18653/v1/N16-3020.
[30] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.
[31] M. Amjad, I. Ahmad, M. Ahmad, P. Wróblewski, P. Kamiński, U. Amjad, Prediction of pile bearing capacity using XGBoost algorithm: modeling and performance evaluation, Appl. Sci. 12 (2022) 2126, https://doi.org/10.3390/app12042126.
[32] J.G. Xu, S.Z. Chen, W.J. Xu, Z. Sen Shen, Concrete-to-concrete interface shear strength prediction based on explainable extreme gradient boosting approach, Construct. Build. Mater. 308 (2021), 125088, https://doi.org/10.1016/j.conbuildmat.2021.125088.
[33] W.Z. Taffese, L. Espinosa-Leal, Multitarget regression models for predicting compressive strength and chloride resistance of concrete, J. Build. Eng. 72 (2023), 106523, https://doi.org/10.1016/j.jobe.2023.106523.
[34] A. Shehadeh, O. Alshboul, R.E. Al Mamlook, O. Hamedat, Machine learning models for predicting the residual value of heavy construction equipment: an evaluation of modified decision tree, LightGBM, and XGBoost regression, Autom. ConStruct. 129 (2021), 103827, https://doi.org/10.1016/j.autcon.2021.103827.
[35] W. Dong, Y. Huang, B. Lehane, G. Ma, XGBoost algorithm-based prediction of concrete electrical resistivity for structural health monitoring, Autom. ConStruct. 114 (2020), 103155, https://doi.org/10.1016/j.autcon.2020.103155.
[36] Christoph Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, Leanpub, 2020.
[37] Y.C. Choi, B. Park, G.S. Pang, K.M. Lee, S. Choi, Modelling of chloride diffusivity in concrete considering effect of aggregates, Construct. Build. Mater. 136 (2017) 81–87, https://doi.org/10.1016/j.conbuildmat.2017.01.041.
[38] M. Marks, M.A. Glinicki, K. Gibas, Prediction of the chloride resistance of concrete modified with high calcium fly ash using machine learning, Materials 8 (2015) 8714–8727, https://doi.org/10.3390/ma8125483.
[39] M. Marks, D. Jóźwiak-NiedzWiedzka, M.A. Glinicki, Automatic categorization of chloride migration into concrete modified with CFBC ash, Comput. Concr. 9 (2012) 375–387, https://doi.org/10.12989/cac.2012.9.5.375.
[40] F.K. Sell Junior, G.B. Wally, F.R. Teixeira, F.C. Magalhães, Experimental assessment of accelerated test methods for determining chloride diffusion coefficient in concrete, Rev. IBRACON Estrut. 14 (2021), https://doi.org/10.1590/s1983-41952021000400007.
[41] H.B. Hou, G.Z. Zhang, Assessment on chloride contaminated resistance of concrete with non-steady-state migration method, J. Wuhan Univ. Technol.-Materials Sci. Ed. 19 (2004) 6, https://doi.org/10.1007/bf02841355.
[42] R.W. Shiu, C.C. Yang, Evaluation of migration characteristics of opc and slag concrete from the rapid chloride migration test, J. Mar. Sci. Technol. 28 (2020) 69–79, https://doi.org/10.6119/JMST.202004_28(2).0001.
[43] M. Maes, E. Gruyaert, N. De Belie, Resistance of concrete with blast-furnace slag against chlorides, investigated by comparing chloride profiles after migration and diffusion, Mater. Struct. 46 (2013) 89–103, https://doi.org/10.1617/s11527-012-9885-3.
[44] J. Jain, N. Neithalath, Electrical impedance analysis based quantification of microstructural changes in concretes due to non-steady state chloride migration, Mater. Chem. Phys. 129 (2011) 569–579, https://doi.org/10.1016/j.matchemphys.2011.04.057.
[45] X. Liu, K.S. Chia, M.H. Zhang, Water absorption, permeability, and resistance to chloride-ion penetration of lightweight aggregate concrete, Construct. Build. Mater. 25 (2011) 335–343, https://doi.org/10.1016/j.conbuildmat.2010.06.020.
[46] S. Real, J.A. Bogas, J. Pontes, Chloride migration in structural lightweight aggregate concrete produced with different binders, Construct. Build. Mater. 98 (2015) 425–436, https://doi.org/10.1016/j.conbuildmat.2015.08.080.
[47] C. Naito, J. Fox, P. Bocchini, M. Khazaali, Chloride migration characteristics and reliability of reinforced concrete highway structures in Pennsylvania, Construct. Build. Mater. 231 (2020), 117045, https://doi.org/10.1016/j.conbuildmat.2019.117045.
[48] J.-I. Park, K.-M. Lee, S.-O. Kwon, S.-H. Bae, S.-H. Jung, S.-W. Yoo, Diffusion decay coefficient for chloride ions of concrete containing mineral admixtures, Adv. Mater. Sci. Eng. 2016 (2016) 11, https://doi.org/10.1155/2016/2042918.
[49] X. Liu, H. Du, M.H. Zhang, A model to estimate the durability performance of both normal and light-weight concrete, Construct. Build. Mater. 80 (2015) 255–261, https://doi.org/10.1016/j.conbuildmat.2014.11.033.
[50] R. Van Noort, M. Hunger, P. Spiesz, Long-term chloride migration coefficient in slag cement-based concrete and resistivity as an alternative test method, Construct. Build. Mater. 115 (2016) 746–759, https://doi.org/10.1016/j.conbuildmat.2016.04.054.
[51] R.M. Ferreira, J.P. Castro-Gomes, P. Costa, R. Malheiro, Effect of metakaolin on the chloride ingress properties of concrete, KSCE J. Civ. Eng. 20 (2016) 1375–1384, https://doi.org/10.1007/s12205-015-0131-8.
[52] A. Pilvar, A.A. Ramezanianpour, H. Rajaie, S.M.M. Karein, Practical evaluation of rapid tests for assessing the Chloride resistance of concretes containing Silica Fume, Comput. Concr. 18 (2016) 793–806, https://doi.org/10.12989/cac.2016.18.6.793.
[53] J. Liu, X. Wang, Q. Qiu, G. Ou, F. Xing, Understanding the effect of curing age on the chloride resistance of fly ash blended concrete by rapid chloride migration test, Mater. Chem. Phys. 196 (2017) 315–323, https://doi.org/10.1016/j.matchemphys.2017.05.011.
[54] K.M. Sunderland, D. Beaton, J. Fraser, D. Kwan, P.M. McLaughlin, M. Montero-Odasso, A.J. Peltsch, F. Pieruccini-Faria, D.J. Sahlas, R.H. Swartz, R. Bartha, S. E. Black, M. Borrie, D. Corbett, E. Finger, M. Freedman, B. Greenberg, D.A. Grimes, R.A. Hegele, C. Hudson, A.E. Lang, M. Masellis, W.E. McIlroy, D.G. Munoz, D.P. Munoz, J.B. Orange, M.J. Strong, S. Symons, M.C. Tartaglia, A. Troyer, L. Zinman, S.C. Strother, M.A. Binns, The utility of multivariate outlier detection techniques for data quality evaluation in large studies: an application within the ONDRI project, BMC Med. Res. Methodol. 19 (2019) 1–16, https://doi.org/10.1186/s12874-019-0737-5.
[55] K. Varmuza, P. Filzmoser, Introduction to Multivariate Statistical Analysis in Chemometrics, CRC Press, 2016.
[56] D. Nettleton, Selection of variables and factor derivation, in: Commercial Data Mining: Processing, Analysis and Modeling for Predictive Analytics Projects, Morgan Kaufmann, 2014, pp. 79–104, https://doi.org/10.1016/B978-0-12-416602-8.00006-6.
[57] J. Brownlee, Statistical Methods for Machine Learning: Discover How to Transform Data into Knowledge with Python, 2019.
[58] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Thirion Bertrand, Olivier Grisel, Mathieu Blondel, Prettenhofer Peter, Ron Weiss, Dubourg Vincent, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[59] S. Vieira, W.H. Lopez Pinaya, A. Mechelli, Main concepts in machine learning, in: A. Mechelli, S. Vieira (Eds.), Machine Learning: Methods and Applications to Brain Disorders, Elsevier, 2020, pp. 21–44, https://doi.org/10.1016/B978-0-12-815739-8.00002-X.
[60] J. Zheng, H.S. Wong, N.R. Buenfeld, Assessing the influence of ITZ on the steady-state chloride diffusivity of concrete using a numerical model, Cement Concr. Res. 39 (2009) 805–813, https://doi.org/10.1016/j.cemconres.2009.06.002.
[61] S.D. Abyaneh, H.S. Wong, N.R. Buenfeld, Modelling the diffusivity of mortar and concrete using a three-dimensional mesostructure with several aggregate shapes, Comput. Mater. Sci. 78 (2013) 63–73, https://doi.org/10.1016/j.commatsci.2013.05.024.
[62] W.Z. Taffese, Data-driven Method for Enhanced Corrosion Assessment of Reinforced Concrete Structures, University of Turku, 2020. https://www.utupub.fi/handle/10024/149752.

[63] W.Z. Taffese, E. Sistonen, Significance of chloride penetration controlling parameters in concrete: ensemble methods, Construct. Build. Mater. 139 (2017) 9–23, https://doi.org/10.1016/j.conbuildmat.2017.02.014.
[64] ACI Committee 135, Machine Learning-Informed Construction and Design, 2023, p. 2023. https://www.concrete.org/committees/directoryofcommittees/acommitteehome.aspx?Committee_Code=C0013500. (Accessed 8 December 2023).
[65] RILEM TC DCS, DCS: Data-driven Concrete Science, 2022. https://www.rilem.net/groupe/dcs-data-driven-concrete-science-444. (Accessed 8 December 2023).