Materials Today Communications 40 (2024) 109527

Contents lists available at ScienceDirect

Materials Today Communications

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

Concrete aging factor prediction using machine learning

Woubishet Zewdu Taffese [a,*], Gustavo Bosel Wally [b,c], Fábio Costa Magalhães [c], Leonardo Espinosa-Leal [a]

[a] School of Research and Graduate Studies, Arcada University of Applied Sciences, Helsinki, Finland
[b] Catholic University of Pelotas, Pelotas, RS, Brazil
[c] Structures and Building Materials Laboratory (LEMCC), Federal Institute of Rio Grande do Sul, Rio Grande, RS, Brazil

ARTICLE INFO

Keywords:
Concrete aging factor
Chloride diffusion coefficient
Machine learning
Ensemble Methods
Concrete durability design

ABSTRACT

Accurate prediction of concrete aging factor is pivotal for performance-based durability reinforced concrete design. This study introduces an innovative method leveraging machine learning techniques, employing seven algorithms: Bagging, Random Forest, AdaBoost, Gradient Boosting, XGBoost, CatBoost, and LightGBM. The dataset comprises 130 instances with seven input features describing cement type, cement content, pozzolan type, pozzolan content, w/b ratio, exposure condition, and age of concrete. Seventy models were trained across five scenarios, categorized into two groups: Group I using all features of the raw dataset, and Group II incorporating engineered features. Model performance, assessed by mean-absolute error (MAE), mean-square error (MSE), root-mean-square error (RMSE), and coefficient of determination (R^2), reveals superior performance in Group II compared to Group I. Notably, the LightGBM algorithm in Scenario III outperforms all models with a remarkable MAE of 0.110, MSE of 0.018, RMSE of 0.133, and R^2 of 0.818. Subsequently, models from Scenarios V and IV exhibit strong performance. The implemented machine learning models demonstrate notable generalizability, effectively capturing feature interrelations without the need for resource-intensive experimental testing.

1. Introduction

Reinforced concrete is essential in civil infrastructure development, but its durability is compromised over time due to various factors, leading to inadequate performance in many structures. This raises significant environmental, economic, social, and safety concerns [1,2]. Consequently, the design of reinforced concrete structures is increasingly emphasizes durability analyses and service-life predictions [1,2]. Current standards for concrete durability is [3–5] largely prescriptive, setting limit values for parameters such as water-to-binder ratio (w/b), binder content, and compressive strength [3–5]. However, these prescriptive approaches have proven inadequate for accurately assessing concrete behavior, particularly concerning chloride penetration [6,7]. As a result, concrete durability specifications are transitioning from prescriptive to performance-based approaches [8,9].

In the context of the performance-based durability design of concrete structures exposed to chloride penetration, it is essential not only to assess concrete properties based on chloride diffusivity but also model chloride penetration and predict the service life of the concrete. Considering that concrete structures are designed to last for at least 50 years, understanding the concrete long-term behavior is crucial for making service-life predictions.

To consider the behavior of concrete diffusivity over time, several service-life prediction models introduce the concept of concrete aging factor (α) [10–14]. This parameter accounts for the variation in chloride diffusivity of concrete over time, primarily arise from the refinement of the concrete pore structure during the cement hydration process and potential pozzolanic reactions. Its value is influenced by factors such as the w/b, the cement type and content, and the type and content of mineral admixtures [15]. Nevertheless, measuring the concrete aging factor is challenging, mainly because it requires analysing concrete diffusivity at advanced ages. Therefore, estimating α values based on concrete mix design parameters and exposure conditions becomes crucial in the performance-based durability design of reinforced concrete structures.

The increasing utilization of supplementary cementitious materials (SCMs) and chemical admixtures, along with other limiting factors, hinders the feasibility of developing accurate mathematical models for

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

https://doi.org/10.1016/j.mtcomm.2024.109527
Received 25 March 2024; Received in revised form 3 June 2024; Accepted 9 June 2024
Available online 10 June 2024

predicting concrete aging factors. The intricate interplay of various parameters controlling the concrete aging factor necessitates an advanced approach capable of capturing the patterns arising from these complex interactions for precise predictions.

Within the array of methods available for predicting concrete properties, machine learning (ML) techniques, a subset of artificial intelligence, stand out. These techniques involve the development and application of algorithms aimed at discerning intricate patterns in data and making informed decisions [16] and have been widely used to infer the mechanical characteristics of concrete [17–20]. Recently, ML methods have also been applied to assess concrete durability-related properties, specifically focusing on the penetration of aggressive gases and ions that impact the concrete's performance [21–27]. For a comprehensive exploration of the utilization of ML in addressing concrete durability challenges, refer to [28].

This study provides a dual contribution. Firstly, it involves the development of ML-based concrete aging factor models for concrete of different ages, incorporating various cement and pozzolan types as well as exposure conditions. Secondly, it explores the influence of input parameters on the concrete aging factor by developing 70 models using seven different algorithms across five scenarios categorized into two groups. To the best of the authors' knowledge, there is no existing research that has investigated the use of ML methods for predicting the aging factor of concrete.

The remainder of the paper is structured as follows: Section 2 provides an overview of ensemble methods. Section 3 encompasses details on the experimental dataset, including its description, data preprocessing, the state of the data after preprocessing, and the procedures for model training and evaluation. In Section 4, results and discussion are presented, covering aspects such as model performance, the effectiveness of feature engineering, and the impact of input features in impacting the concrete aging factor. Finally, Section 5 concludes the study, addressing limitations and offering insights into future prospects.

2. Ensemble methods

In this study, ensemble techniques are leveraged to develop predictive models for estimating the concrete aging factor. This approach entails the fusion of multiple ML models to mitigate the constraints of individual models by harnessing the strengths of diverse models, resulting in enhanced overall performance and model robustness. Given the limited size of the dataset in use, employing ensemble methods to predict the concrete aging factor is a prudent choice.

The ensemble strategy adopted in this study centers around the utilization of Decision Trees (DTs) as the foundational model. DTs have gained substantial recognition and have seen extensive application in tackling complex engineering challenges, as explained in Section 2.1, where their operational principles are outlined. Within the realm of ensemble techniques rooted in DTs, two primary branches emerge: bagging and boosting. These categories' fundamental concepts for addressing regression problems are explained in Sections 2.2 and 2.3, respectively. Subsequently, in Section 2.4, we furnish a concise description of the specific bagging and boosting ensemble methods employed in this study.

2.1. Decision trees

A DT is a graphical structure with nodes, branches, and leaves, as shown in Fig. 1. In the figure's left part, data points are divided, while the right part displays the tree structure. Decision nodes mark specific domain areas to split further, while leaf nodes represent areas without further division. The root node is the highest node, and branches connect to descendant or leaf nodes based on split results. A split is defined by a test function, $t : X \rightarrow R_t$, mapping instances to split outcomes. Each split outcome has its branch. When split results are known for reachable instances, the domain is divided into subsets along the branches. As per Eq. (1), each DT node corresponds to a specific domain area, determined by a sequence of splits $t_1, t_2, \ldots, t_k$ and their results $r_1, r_2, \ldots, r_k$ from the root to the leaf nodes [29]. DTs, noted for their quick learning and interpretability, excel in handling complex nonlinear problems with a substantial amount of data and input features.

$$X_n = \{x \in X | t_1(x) = r_1 \wedge t_2(x) = r_2 \wedge \cdots \wedge t_k(x) = r_k\}. \quad (1)$$

2.2. Bagging

In ensemble learning using a bagging approach, the underlying models are created by randomly selecting bootstrapped samples from the dataset. This procedure is iterated several times, resulting in a substantial subset of training datasets in which some data points may appear multiple times. Each bootstrapped sample comprises approximately 63.2 % of the total number of training dataset samples. The instances not included in these bootstrapped samples are used to assess the model's performance. Every base model is trained on the training dataset, denoted as $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, to produce the prediction $\hat{f}(x)$ for a given input vector x. For each bootstrapped sample D^{*t}, where t ranges from 1 to T, the model makes a prediction denoted as $\hat{f}^{*t}(x)$. The aggregated estimate is obtained by computing the average prediction for input vector x across T models, as illustrated in Eq. (2). This averaging process helps reduce variance and enhances the overall stability of the ensemble [30].

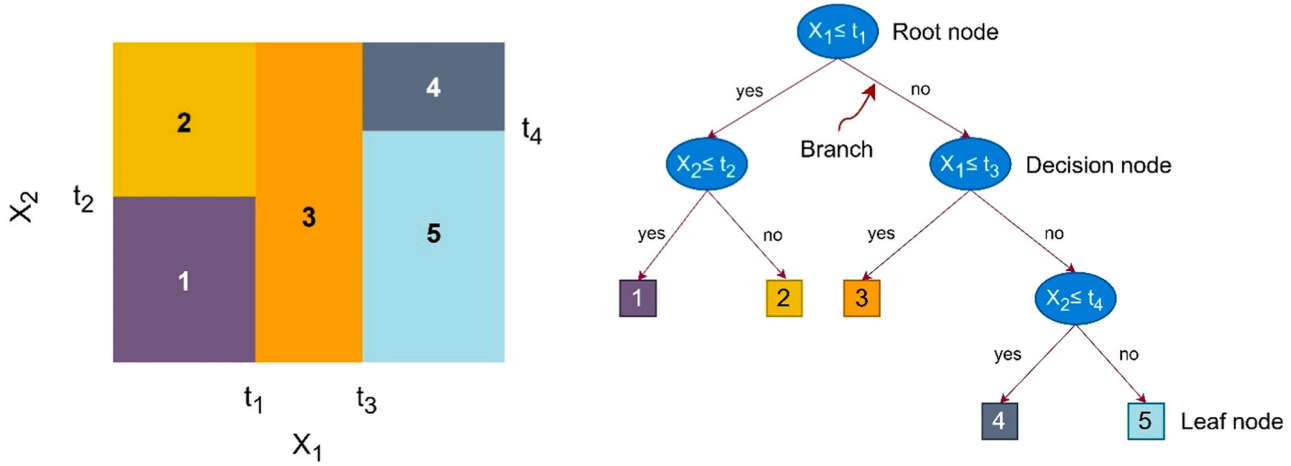

Fig. 1. Visualization of a dataset and its corresponding decision tree.

$$\widehat{f}_{bag}(x) = \frac{1}{T}\sum_{t=1}^{T}\widehat{f}^{*t}(x). \quad (2)$$

2.3. Boosting

Boosting represents a sequential procedure where it builds simple base models and integrates improvements from one model to the next, consequently enhancing the ensemble method's performance. Each base model is crafted using a training dataset denoted as $D = \{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$, and it leverages the insights gained from previously constructed base models. To achieve this, an appropriate algorithm is utilized to train the datasets $D^{(t)}$, where t varies from 1 to T, using a sequence of distinct weights $w^{(1)}, \quad w^{(2)}, \quad \ldots, \quad w^{(T)}$. This leads to the creation of predictions $\widehat{f}^{(1)}(x), \quad \widehat{f}^{(2)}(x), \quad \ldots, \quad \widehat{f}^{(T)}(x)$ for each input vector x, along with their corresponding weight vector w. Typically, the weight vector commences with an initial weight $w^{(1)}$ and is subsequently adjusted in each base model based on observed errors. The ultimate model output is acquired by combining the individual base model outputs with a weighted sum, as demonstrated in Eq. (3).

$$\widehat{f}_{boost}(x) = \sum_{t=1}^{T}\widehat{f}^{(t)}(x)w^{t}. \quad (3)$$

2.4. Brief overview of employed ML algorithms

In this study, seven well-established ensemble techniques that harness both bagging and boosting are employed. These methods include Bagging, Random Forest, Adaptive Boosting, Gradient Boosting, Extreme Gradient Boosting, Categorical Boosting, and Light Gradient Boosting Machine. While Bagging is commonly known for enhancing the stability and accuracy of machine learning algorithms, as explained in Section 2.2 above, an ensemble learning technique that exclusively relies on this approach, using DTs as base models, is employed and referred to as Bagging. The subsequent section offers concise descriptions of all the adopted algorithms.

2.4.1. Bagging

It is ab ensemble learning method where several separate DT models are trained using random subsets of the training data with replacement. Each model is trained independently of the others. The ultimate prediction is often determined by averaging the predictions generated by each individual model.

2.4.2. Random Forest (RF)

It is an enhanced iteration of bagging, involving an ensemble of base models developed through a bagging approach. Breiman [31] introduced the notion of feature randomness into the bagging process, resulting in an uncorrelated assembly of DTs, often referred to as a "forest". RF represents a substantial improvement over bagged DT.

2.4.3. Adaptive Boosting (AdaBoost)

It operates based on the core concept of fitting a series of weak learners to iteratively adjusted versions of the data, simultaneously updating their weights to minimize training error. This sequential optimization process persists until the best predictor is identified. Freund and Schapire [32] originally introduced this algorithm, initially designed for resolving classification tasks and was later extended to tackle regression problems.

2.4.4. Gradient Boosting (GB)

GB is a methodology that melds the gradient descent algorithm with the boosting technique. Conceived by Friedman [33], it functions by progressively introducing predictors into an ensemble, with each predictor striving to rectify the errors made by its forerunner.

2.4.5. Extreme Gradient Boosting (XGBoost)

Initially crafted by Chen [34], XGBoost has since garnered contributions from various individuals. It applies the foundational principles of GB and is tailored for streamlined computation and scalability. Through the utilization of multiple central processing unit (CPU) cores, XGBoost facilitates parallel learning throughout the training phase.

2.4.6. Categorical Boosting (CatBoost)

It is a GB implementation devised by the Russian IT company, Yandex [35]. It brings about two pivotal algorithmic innovations. Firstly, it introduces ordered boosting, a permutation-driven departure from the conventional approach. Secondly, it incorporates an inventive method for handling categorical features. These advancements were designed to combat a prediction bias stemming from a unique type of target leakage found in all existing GB algorithm implementations, while efficiently handling categorical data.

2.4.7. Light Gradient Boosting Machine (LightGBM)

It is based on the GB framework developed by Microsoft [36]. It excels at efficiently training on large-scale datasets while minimizing memory usage. LightGBM employs two innovative techniques, Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS enables training each tree with only a small fraction of the complete dataset, while EFB enhances the efficient handling of high-dimensional sparse features.

3. Dataset and modelling

This section delves into the details of the dataset employed in the study, offering a thorough exploration. It systematically outlines the procedural stages of developing predictive models for concrete aging factors, covering the primary phases inherent in any ML advancement including data preprocessing, model training, and model evaluation. Fig. 2 illustrates the steps involved in each model development process. The model development process begins with importing the collected experimental data, which includes parameters describing the binder types and content, water-to-binder ratio, curing and field exposure conditions, and concrete aging factor. Next, data preprocessing is performed to prepare the dataset for ML modeling. This step involves handling missing data, data encoding, feature engineering, managing outliers, and partitioning the data into training and testing sets. Subsequently, several models are trained using the training dataset and seven decision tree-based ensemble algorithms to predict the concrete aging factor under different scenarios. The performance of all trained models is then evaluated using the test dataset to identify the best-performing models. Each activity is discussed in the following subsections.

3.1. Experimental dataset

The study utilized data sourced from scholarly journal articles published globally. A total of nine scientific articles were carefully selected to construct a dataset for predicting and characterizing the concrete aging factor [15,37−44]. Subsequently, a rigorous initial evaluation was conducted to ensure data completeness and suitability, leading to the choice of eight distinct features encompassing 130 instances. Table 1 displays the feature types, their corresponding units, and data types. These features were categorized into three groups: Category 1, Category 2, and Category 3. Category 1 encompasses five features that provide comprehensive information on the concrete composition which are cement type, cement content, pozzolan type, pozzolan content, and w/b. The cement types comprise 11 variations conforming to different standards, including Brazilian, Canadian, European, and ASTM. Among them, European cement types are five in number, while each of the other standards has two types. Regarding pozzolan types, a total of seven distinct types were considered in the dataset. Although most concrete instances utilized a single pozzolan type, there were cases where a

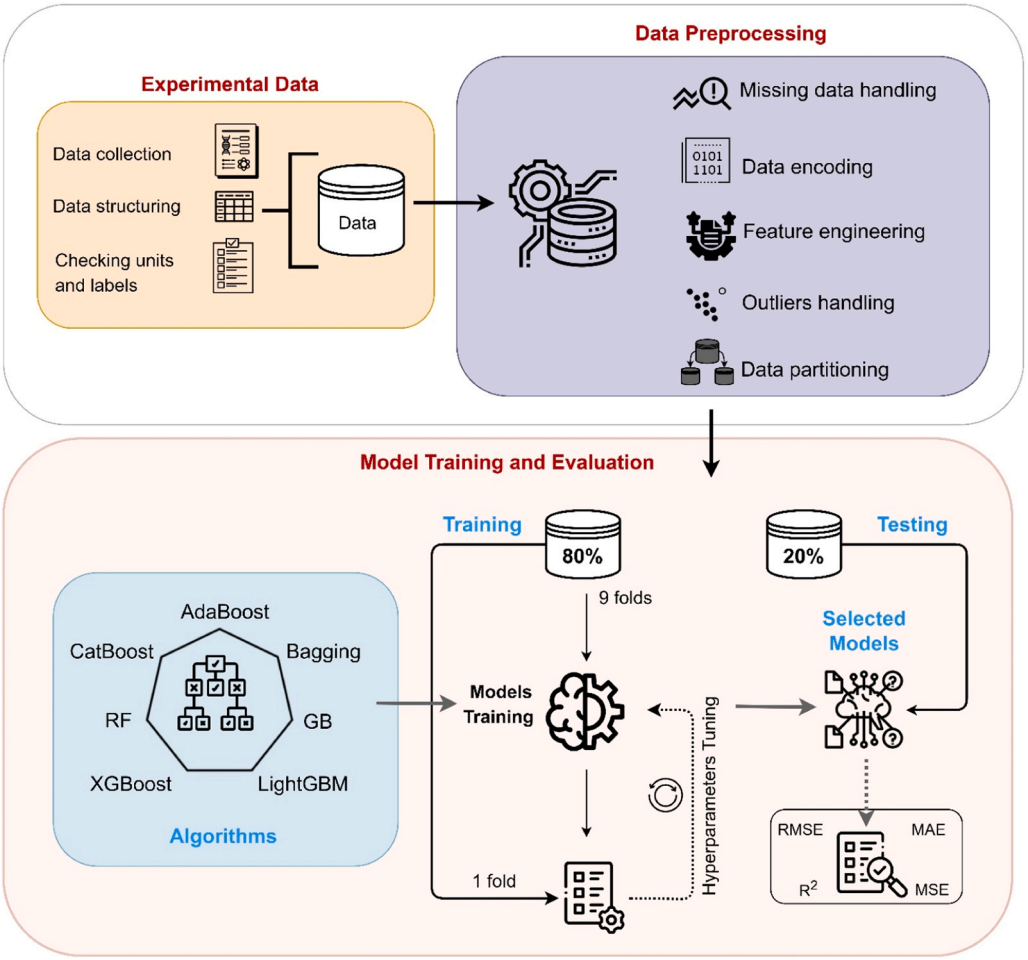

Fig. 2. Workflow for developing the concrete aging factor prediction models.

combination of two pozzolan types was employed, which are "Silica fume & fly ash", "Silica fume & Slag". Category 2 includes two features: exposure conditions and the age of the concrete. The exposure conditions of the concrete to chloride vary and are categorized into two primary groups: Field and Lab. Field exposure is further divided into splash or tidal zone, while lab exposure involves five distinct test types. The age of the concrete is interpreted differently based on the exposure condition; for lab tests, it indicates the age during testing, and for field conditions, it refers to the duration of field exposure. Category 3 comprises a single feature describing the concrete aging factor. The dataset is composed of both numerical and categorical data types, as outlined in Table 1. Numerical data types represent numeric values, while categorical data types categorize data into discrete, non-numeric categories. All features in the dataset, except for "cement type", "pozzolan type", and "exposure conditions", are of numerical data types.

Fig. 3 illustrates the distribution of numerical features within the dataset. The diagonal presents the kernel density estimate (KDE) for each feature's distribution, while the lower section exhibits a contour plot representing the probability distribution of one feature in relation to the others. Notably, the cement content feature displays a relatively normal distribution compared to the other features, which deviate from normality, presenting bimodal or trimodal distributions. This suggests the presence of two or three distinct groups or subpopulations for these features. In terms of ranges, all the concrete-related features exhibit a broad spectrum, indicating a diverse range of concrete types encompassed in the data. Additionally, the dataset covers concrete exposed to chloride for an extended duration, up to eight years.

Fig. 4 demonstrates the breakdown of categorical features within the dataset, encompassing cement types, pozzolan types, and exposure conditions. The central portion of the figure presents the distribution of standard types that the cement adheres to. For pozzolan types, it presents the count of concrete samples that utilized pozzolan in their composition. In terms of exposure conditions, it displays the distribution of concrete exposed to chloride environments under both lab and field exposure conditions. The outer circle displays the names of all cements, pozzolans, and test types with a representation exceeding 3 %, while those below 3 % are depicted in the plot without labeling. Significantly, ASTM cement types (specifically ASTM Type I and II) collectively account for 96 instances, making up 72 % of the cement types. European standard-compliant cement types make up a substantial portion (18 %), following ASTM types, with Brazilian and Canadian standards contributing equally. As for pozzolan types, 103 instances of concrete incorporate various pozzolan categories. The majority, at 37 %, utilized metakaolin, comprising 49 instances, followed by silica fume and limestone filler, constituting 16 % and 11 %, respectively. Regarding exposure conditions, 109 instances, making up 81 %, were exposed to a chloride environment in a laboratory setting. The remaining 19 % of concretes experienced field conditions, primarily in the tidal zone, with the splash zone being less common. Among the concrete chloride penetration tests conducted in a lab environment, ASTM C1556 accounts for the largest portion, with 40 %, followed by NT Build 443 and NT Build 492 at approximately 20 % and 12 %, respectively. Due to the distinct amounts of chloride used and the unique procedures associated with each lab test, they were utilized as is to model the concrete aging factor. Overall, the distributions of the three categorical features exhibited highly imbalanced data, with a prominent representation of

Table 1
Description of considered features.

Category	No	Feature	Description		Unit
Category 1	1	Cement type	ASTM standard	"ASTM Type I", "ASTM Type II"	[-]
			Brazilian standard	"CP IV", "CP V"	
			European standard	"CEM I 42,5 R", "CEM I 52,5", "CEM I HSR", "CEM I-42,5 R/SR", "CEM III/A 42,5 LA"	
			Canadian standard	"CSA Type 10SF", "Type I/II low-alkali",	
	2	Cement content			[kg/m^3]
	3	Pozzolan type	"Fly ash", "Limestone filler", "Metakaolin", "Rice husk ash", "Silica fume", "Slag"		[-]
	4	Pozzolan content			[kg/m^3]
	5	w/b			[-]
Category 2	6	Exposure condition	Field	"Splash zone", "Tidal zone"	[-]
			Lab	"ASTM C1556", "Immersion tank", "McGrath and Hooton", "NT Build 443", "NT Build 492"	
	7	Age of concrete			[Year]
Category 3	8	Concrete aging factor			[-]

one type from each feature covering the largest portion.

3.2. Data preprocessing

Data preprocessing plays a pivotal role in the development of ML models, acting as the transformative bridge that turns raw data into a more analytically friendly and model-ready format. This multifaceted procedure involves a sequence of critical tasks, encompassing handling missing data, encoding data, identifying and addressing outliers, engaging in feature engineering, and partitioning the data. The subsequent sections provide in-depth exploration of each of these critical data preprocessing steps, offering a comprehensive understanding of their application.

3.2.1. Missing data processing

The quality of input data significantly impacts the efficacy of ML models. When specific features have missing values, it can greatly affect the model's performance and introduce biases in the outcomes. As a result, accurate prediction and effective generalization to new data may become challenging for the model. Several approaches exist to address missing data, including removing instances with missing values, letting the algorithm handle the missing values, or imputing the missing data. In this study, the choice was made to exclude observations with missing values. This decision was driven by the fact that, given the small number of missing values, their removal from the dataset would negligibly affect the model's performance. Furthermore, as highlighted in Section 3.2, the number of observations is relatively low, and they encompass imbalanced categorical features. In this scenario, with a combination of limited observations and a wide feature distribution, imputing missing values could introduce biases and lead to inaccurate predictions and inferences.

3.2.2. Data encoding

numerical input. Categorical features within the dataset, such as cement type, pozzolan type, and exposure condition, have a high number of categories, as shown in Fig. 4. To address this, binary encoding is utilized instead of one-hot encoding because it encodes data in fewer dimensions. This technique uses binary code, employing sequences of zeros and ones to represent distinct categories. Initially, the categories is converted into ordinal numbers using an ordinal encoder, as illustrated in the intermediate step in Fig. 5. These ordinal numbers are then transformed into binary representations. Generally, the number of binary features needed to encode a feature as $\log_2$(number of distnict categories)[]; with 11 categories, 4 binary features are required. For instance, the integer 1 is represented as 0001 and the integer 2 as 0010, with similar binary sequences applied to all ordinal numbers defined in the intermediate step. These binary values are then divided into separate columns, creating the encoded representation of the original categories. This transformation facilitates the shift from non-numerical to numerical data, enabling further processing and utilization in ML algorithms. Fig. 5 provides a practical example, visually showcases the transformation of cement type before and after binary encoding.

3.2.3. Feature engineering and selection

Feature engineering holds a pivotal role in the domain of ML. It is the process of using domain expertise to transform existing features into a new set of features. This transformation aims to leverage the newly generated features, reduce the number of predictors, and ultimately improve prediction performance while decreasing model complexity. In this study, the focal point was on reducing input features by amalgamating descriptors of binder, specifically cement content and pozzolan content, into a singular feature which is the ratio of cement content to sum of cement and pozzolan content ("c/c+p"). This consolidation was achieved through a division operation, simplifying the representation of binder contents into a concise and informative feature.

Feature selection involves choosing the most pertinent features from the dataset. Pearson correlation coefficients, denoted by "r", are computed for all possible numerical feature pairs using the formula in Eq. (4), providing insights into their interdependence. These coefficients range from +1 to −1, where +1 signifies a complete positive correlation, −1 indicates a complete negative correlation, and 0 suggests no correlation [45]. A high correlation between two features suggests they are closely related and may have similar impacts on the dependent variable, potentially allowing for the omission of one feature. It is important to note that correlation does not imply causation, so domain knowledge is necessary to validate findings. Fig. 6 illustrates the Pearson correlation coefficient values, showcasing the relationships among the numerical features in the dataset. It is evident that there is no strong correlation between the input numerical features, allowing for the utilization of all these features in modeling the concrete aging factor.

$$r = \frac{\sum (x_i - \overline{x}) * (y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2 \sum (y_i - \overline{y})^2}} \quad , \tag{4}$$

where x_i and y_i symbolize the values of the two features under examination, and $\overline{x}$ and $\overline{y}$ denote the means of their respective features.

3.2.4. Detecting and treating outliers

Multivariate outliers refer to observations or data points in a dataset that deviate significantly from the overall pattern observed in the data across multiple features or dimensions. These outliers can impact ML-based models, making it important to identify and handle them appropriately to safeguard the accuracy and significance of results. In this study, we employ the isolation forests algorithm, which is effective in identifying multivariate outliers in datasets containing both numerical and categorical data, to isolate unusual observations within the dataset [46]. Several studies have demonstrated its effectiveness in detecting outliers in various research domains[47–50].

Contrary to many conventional outlier detection methods that focus on identifying typical data points, isolation forests explicitly aim to pinpoint actual outliers. This unique approach enables the algorithm to

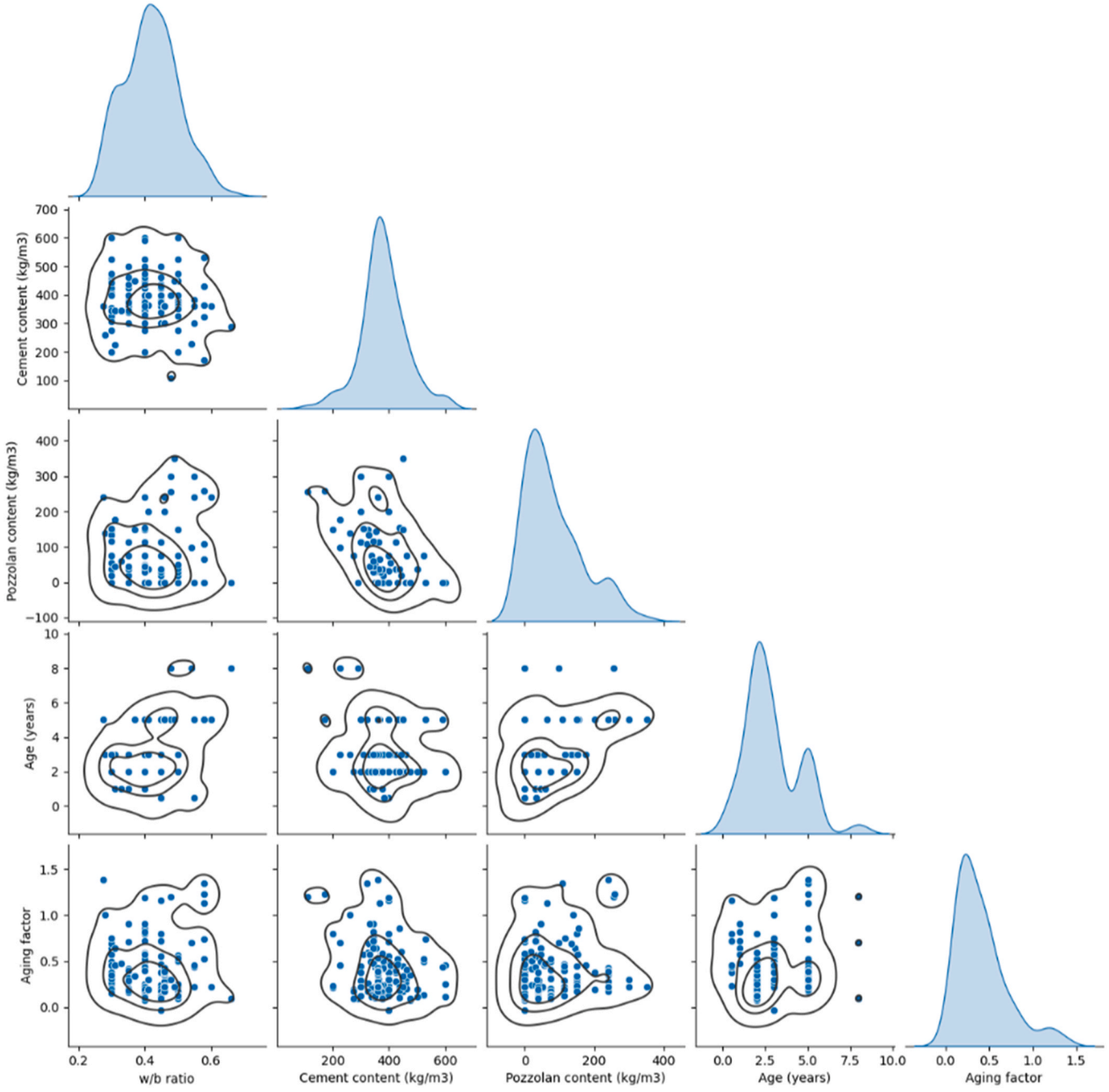

Fig. 3. Distribution of numerical features in the dataset.

be more efficient, requiring fewer conditions to segregate outliers from the normal data points. Similar to other tree-based ensemble methods, the isolation forest is constructed using a set of DTs referred to as "isolation trees" or "iTrees". Each tree within this ensemble covers a subset extracted from the complete dataset. The process begins by randomly selecting n samples of size m from the dataset. For each of these random samples, an "iTree" is created. This "iTree" is constructed through a series of splits on instances within the subsample, guided by the split value of a randomly chosen feature. Instances with feature values lower than the split value are directed to the left, while the others proceed to the right. This recursive process continues until the tree is fully grown. The split value is selected randomly within the range defined by the minimum and maximum values of the chosen feature. Outliers are identified as data points with the shortest path length, represented as $h(x)$, which indicates the distance from the root to the leaf node in each of the "iTrees," as illustrated in Fig. 7.

An instance's outlier score can be calculated by leveraging the observation that the structure of "iTrees" closely resembles that of Binary Search Trees (BST). The point where a leaf node terminates in "iTrees" corresponds to the result of an unsuccessful search in a BST. As a result, estimating the mean value $h(x)$ for the leaf node termination is akin to an unsuccessful search in a BST [51,52], as explicitly demonstrated in Equation (5).

$$c(m) = \begin{cases} 2H(m-1) - \dfrac{2(m-1)}{m} & \text{for} m > 2 \\ 1 & \text{for} m = 2 \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

where $H(i)$ stands for the harmonic number, which is estimated using $\ln(i) + 0.5772156649$, (Euler's constant) n represents the dimension of

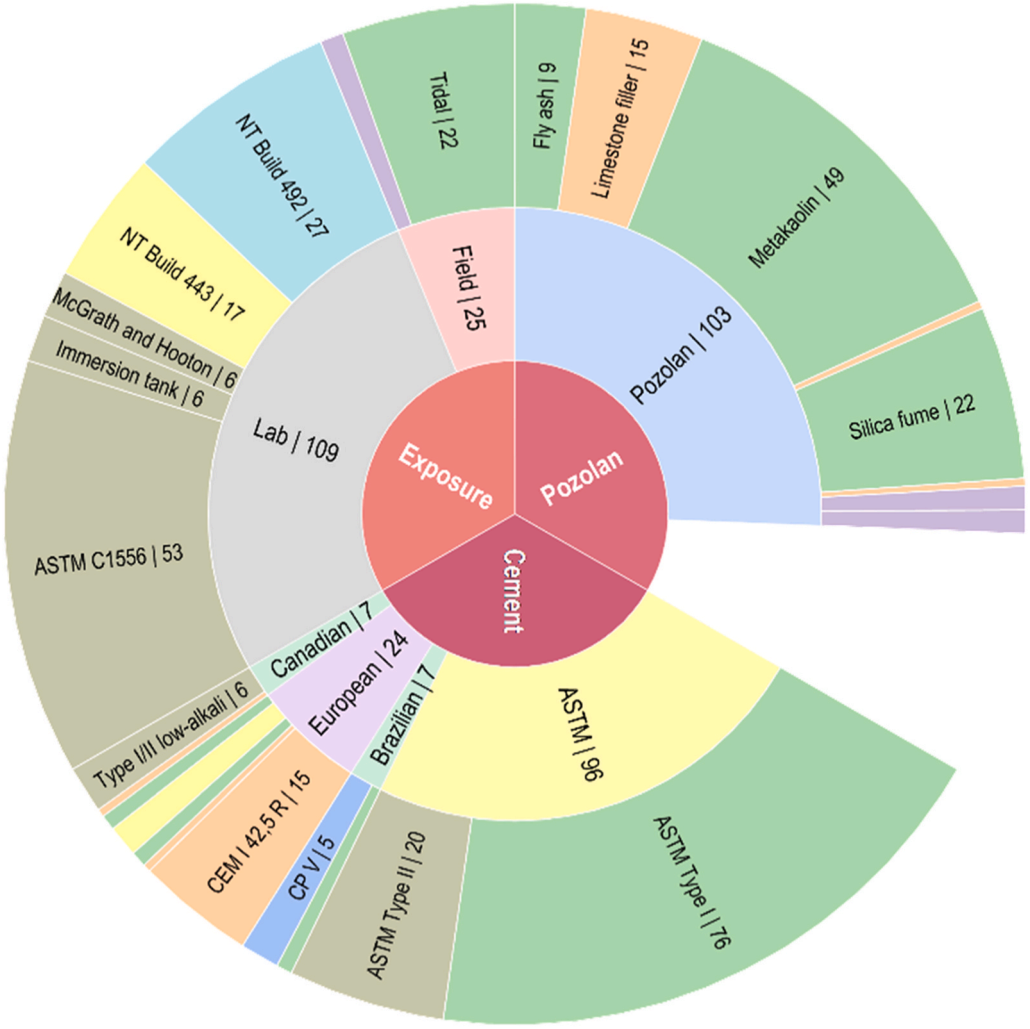

Fig. 4. Distribution of categorical features.

Cement type	Order	Binary	Cement type_1	Cement type_2	Cement type_3	Cement type_4
CP V	1	0001	0	0	0	1
CP IV	2	0010	0	0	1	0
ASTM Type II	3	0011	0	0	1	1
ASTM Type I	4	0100	0	1	0	0
Type I/II low-alkali	5	0101	0	1	0	1
CEM I-42,5 R/SR	6	0110	0	1	1	0
CSA Type 10SF	7	0111	0	1	1	1
CEM I 42,5 R	8	1000	1	0	0	0
CEM I 52,5	9	1001	1	0	0	1
CEM II/A 42,5 LA	10	1010	1	0	1	0
CEM I HSR	11	1001	1	0	0	1

Fig. 5. Categorical feature pre- and post-encoding transformation.

the test set, and m denotes the size of the sample set.

The parameter of $c(m)$ in Eq. (4) represents the mean $h(x)$ for a particular m. This value is then used to normalize $h(x)$, providing an estimated outlier score for a given instance x, as elaborated in Eq. (6).

$$s(x,m) = 2^{\frac{-E(h(x))}{c(m)}} \tag{6}$$

where $E(h(x))$ denotes the mean $h(x)$ value obtained from a set of isolation tree. When the value of s is close to 1, the instance x is classified as an outlier; conversely, if it is substantially less than 0.5, the instance x is considered normal. If all instances yield $s \approx 0.5$, the entire sample shows no notable outliers.

Fig. 8 illustrates the outliers identified by the employed isolation

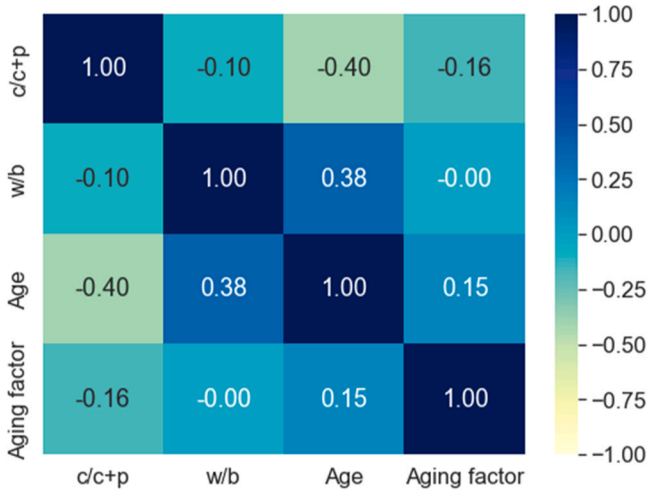

Fig. 6. The Pearson correlation coefficients among features.

forest algorithm following the reduction of feature dimensions to three using Principal Component Analysis (PCA). It is apparent that these outlier instances are largely distinct from the cluster of normal instances.

3.2.5. *Data partitioning*

In ML, partitioning data is a pivotal step involving the division of a dataset into distinct subsets—typically training, validation, and testing sets. The objective is to accurately evaluate the model's performance and its ability to generalize to unseen data. The training set is employed to train the model, often utilizing techniques like *K*-fold cross-validation for validation. Conversely, the test set, kept separate and concealed during training and validation, is crucial for assessing the model's ability to generalize to new, unseen data.

In this study, the dataset undergoes a random partition, with 80 % dedicated to training, including validation, and the remaining 20 % reserved for testing. This deliberate allocation strategy aims to furnish the model with a significant amount of data for learning, ensuring its ability to effectively capture patterns and relationships within the dataset. Concurrently, setting aside 20 % of the data for testing allows for the evaluation of the model's generalization capabilities on new and unseen data.

3.3. *Data after preprocessing*

Table 2 presents the descriptive statistics for all numerical features, including the engineered feature. Notably, the total number of observations has been reduced from the original 130 to 122 due to the exclusion of outliers and instances with missing values. The cement content varies from 172 to 600 kg/m^3, while the pozzolan content ranges from 0 to 350 kg/m^3. The mean w/b ratio stands at 0.42, with a standard deviation of 0.08, indicating that most instances cluster around this mean value. The engineered feature, c/c+p, spans from 0.4 to 1. Furthermore, the age of the concrete during the chloride test conducted in both lab and field environments exhibits a broad range, extending from 0.5 to 8 years, with a standard deviation of 1.54 years. It's worth noting the presence of a case with a negative concrete aging factor, as reported in the work [41]. The occurrence of a negative concrete aging factor, however, may be related to small variations in results arising from the variability of test methods or exposure conditions in cases of

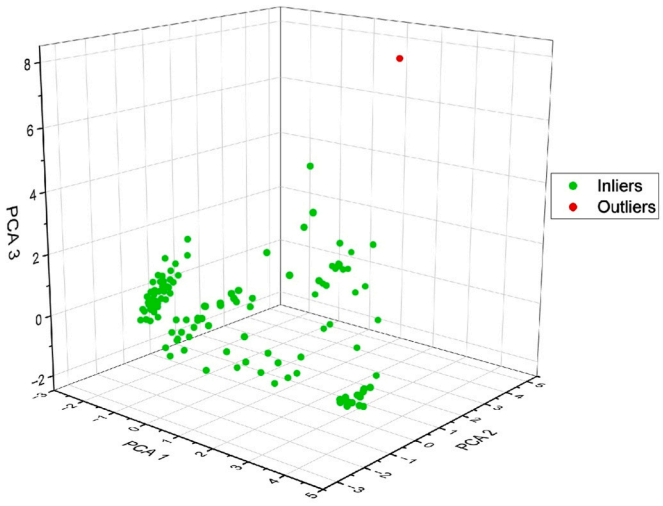

Fig. 8. Visual depiction of detected outliers.

Table 2
Descriptive statistics of the preprocessed data.

	w/b	Cement content	Pozzolan content	c/c+p	Age	Concrete aging factor
Units	[-]	[kg/m^3]	[kg/m^3]	[-]	[year]	[-]
count	122	122	122	122	122	122
mean	0.42	385.05	79.13	0.84	2.83	0.39
std	0.08	77.54	82.08	0.15	1.54	0.28
min	0.28	172.00	0.00	0.40	0.50	-0.03
25 %	0.38	350.00	18.50	0.75	2.00	0.19
50 %	0.40	375.00	48.00	0.88	2.00	0.30
75 %	0.46	425.00	112.50	0.96	3.00	0.50
max	0.66	600.00	350.00	1.00	8.00	1.38

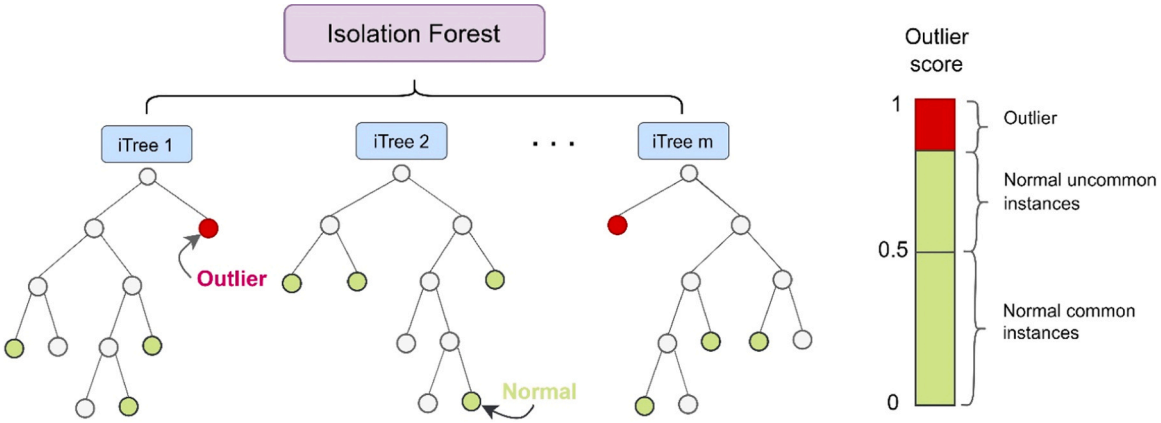

Fig. 7. Demonstration of isolation forest for detecting outliers.

field exposure. It is also noteworthy that the negative concrete aging factor observed is a very small value (0.03) and should not be understood as indicating an increase in the diffusivity of the concrete. These findings underscore the dataset's diverse representation of various concrete types, despite the limited number of instances.

3.4. Model training and evaluation

The model training process in this study encompasses five distinct scenarios organized into two groups, as outlined in Table 3. Group I involves considering the features without incorporating engineered features, while Group II incorporates the engineered feature. Within these scenarios: Scenario I of Group I includes all eight features presented in Table 1. Scenario II comprises all the features from Scenario I, with the exception of cement types. Scenario III is similar to Scenario II, but it focuses on instances involving concrete utilizing ASTM cement types, which account for a relatively higher number of cases (73 instances). Scenario IV mirrors Scenario I but concentrates on instances where the chloride penetration test was conducted in a lab environment, given its higher representation compared to field tests. Scenario V mirrors Scenario I but, instead of considering individual test types, it clusters into either lab or field categories.

All the scenarios within Group II are essentially identical to the corresponding scenarios in Group I, with the distinction that they utilize the engineered features "c/c+p" instead of the features cement and pozzolan types. With the application of seven different types of DT-based ensemble algorithms in each scenario within the two groups, a total of 70 models are developed. The primary objective of employing multiple algorithms is to identify the best models for accurately predicting the concrete aging factor. This is particularly important because the dataset comprises a wide variety of materials, varying material proportions, exposure conditions, a broad range of ages, concrete aging factors, all within the constraints of a limited number of observations.

The prediction models for concrete aging factors were developed by training seven ensemble methods, specifically Bagging, RF, AdaBoost, GB, XGBoost, CatBoost, and LightGBM algorithms, all of which were implemented using Python's scikit-learn library [53]. This achievement was realized by utilizing input and target attributes extracted from preprocessed data. The training dataset, which comprised 80 % of the available data, was used to train the models. To enhance the model's performance, the hyperparameters of these algorithms were thoroughly fine-tuned using a combination of grid search and K-fold cross-validation. The grid search systematically explored the hyperparameter space to identify the best combination of hyperparameters. Due to the computational intensity of grid search, the model development was conducted on high-performance computing infrastructure. In K-fold cross-validation, the training dataset was randomly partitioned into K approximately equal-sized subsets. Each of the K subsets functioned as a validation set to evaluate the model's performance, while the remaining $(K-1)$ subsets served as the training set. This process led to the creation of K models and the acquisition of K validation statistics. The average score across the K folds was used to assess the overall performance of the model. Various values of K were experimented with, and it was determined that a K value of 10 struck a favorable balance between bias and variance in performance evaluation. The optimal hyperparameters were subsequently incorporated to train the algorithms. An overview of all the hyperparameters considered during the training process is provided in Table 4.

Once the concrete aging prediction models have been trained using the datasets encompassing the five scenarios, it is imperative to evaluate their predictive capabilities on both training dataset and a separate test dataset that was not part of the training phase. This assessment is crucial to ascertain the models' accuracy and generalization ability. To gauge the accuracy of these regression models, several commonly used statistical metrics come into play. These metrics include mean-square error (MSE), root-mean-square error (RMSE), mean-absolute error (MAE),

Table 3
Feature details for each scenario and group.

Scenario	Group	Number of features		Feature types		Number of instances
		Inputs	Target	Inputs	Target	
Scenario I	Group I	7	1	Cement type, Cement content, Pozzolan type, Pozzolan content, w/b, Exposure condition, Age of concrete	Concrete aging factor	122
	Group II	6		Cement type, Pozzolan type, c/c+p, w/b, Exposure condition, Age of concrete		
Scenario II	Group I	6	1	Cement content, Pozzolan type, Pozzolan content, w/b, Exposure condition, Age of concrete	Concrete aging factor	122
	Group II	5		Pozzolan type, c/c+p, w/b, Exposure condition, Age of concrete		
Scenario III	Group I	6	1	Cement content, Pozzolan type, Pozzolan content, w/b, Exposure condition, Age of concrete	Concrete aging factor	73
	Group II	5		Pozzolan type, c/c+p, w/b, Exposure condition, Age of concrete		
Scenario IV	Group I	6	1	Cement type, Cement content, Pozzolan type, Pozzolan content, w/b, Age of concrete	Concrete aging factor	103
	Group II	5		Cement type, Pozzolan type, c/c+p, w/b, Age of concrete		

(continued on next page)

Table 3 (*continued*)

Scenario	Group	Number of features		Feature types		Number of instances
		Inputs	Target	Inputs	Target	
Scenario V	Group I	7	1	Cement type, Cement content, Pozzolan type, Pozzolan content, w/b, Exposure condition, Age of concrete	Concrete aging factor	122
	Group II	6		Cement type, Pozzolan type, c/c+p, w/b, Exposure condition, Age of concrete		

Table 4
Hyperparameters examined in all utilized ensemble models.

Algorithm	Hyperparameters	Ranges
Bagging	n_estimators	[20, 50, 100, 200, 300, 500, 600, 700, 800, 900, 1000]
	max_features	[0.5, 0.7, 0.8, 0.9, 1.0]
	max_samples	[0.5, 0.7, 0.8, 0.9, 1.0]
	bootstrap	[True, False]
	bootstrap_features	[True, False]
RF	n_estimators	[20, 50, 100, 200, 300, 500, 600, 700, 800, 900, 1000]
	max_features	['None', 'sqrt', 'log2']
	max_depth	[None, 10, 20, 30]
	min_samples_split	[2,3,5,7,10]
	min_samples_leaf	[1,2,4,6,8]
	bootstrap	[True, False]
AdaBoost	n_estimators	[20, 50, 100, 200, 300, 500, 600, 700, 800, 900, 1000]
	learning_rate	[0.00001,0.0001, 0.001, 0.01, 0.05, 0.1]
	loss	['linear', 'square', 'exponential']
GB	n_estimators	[20, 50, 100, 200, 300, 500, 600, 700, 800, 900, 1000]
	learning_rate	[0.00001,0.0001, 0.001, 0.01, 0.05, 0.1]
	min_samples_split	[2,5,7,10]
	min_samples_leaf	[1,2,4,8]
	max_depth	[3–6]
	max_features	['auto', 'sqrt', 'log2']
XGBoost	n_estimators	[20, 50, 100, 200, 300, 500, 600, 700, 800, 900, 1000]
	max_depth	[2–8]
	min_split_loss	[0, 0.1, 0.2, 0.3]
	learning_rate	[0.00001,0.0001, 0.001, 0.01, 0.05, 0.1]
	booster	['gbtree','gblinear','dart']
CatBoost	iterations	[100, 200, 300,400,500]
	learning_rate	[0.00001,0.0001, 0.001, 0.01, 0.05, 0.1]
	depth	[4,6,8,10]
	l2_leaf_reg	[1,3,5]
	bagging_temperature	[0.5, 1.0, 1.5]
	border_count	[32, 64, 128]
	rsm	[0.8, 0.9, 1.0]
LightGBM	n_estimators	[20, 50, 100, 200, 300, 500, 600, 700, 800, 900, 1000]
	max_depth	[4,6,8,10]
	min_child_samples	[2,5,10]
	num_leaves	[20,30,40]
	learning_rate	[0.00001,0.0001, 0.001, 0.01, 0.05, 0.1]
	subsample	[0.8, 0.9, 1.0]

and coefficient of determination (R^2) [54]. MSE, as defined in Eq. (7), quantifies the mean squared discrepancy between the predicted values and the actual values. This particular metric assesses the general variability or spread of errors, with diminished values signifying a more superior predictive performance. It is a valuable indicator of the overall model performance. RMSE, denoted by Eq. (8), is the square root of the mean of the squared differences between predicted and actual values. One notable advantage of RMSE is that it is expressed in the same units as the dependent variable, making it easier to understand. Similar to MSE, lower RMSE values are indicative of better predictive performance. MAE, as defined in Eq. (9), calculates the mean of the absolute differences between predicted and actual values. This metric provides an evaluation of the average magnitude of errors, without regard to their direction. Reduced MAE values signify enhanced predictive performance. In contrast to some other error metrics that involve squaring the disparities (e.g., MSE), MAE treats all errors evenly, assigning equal significance to both overestimation and underestimation. The R^2 value denotes the fraction of variance in the response feature that can be elucidated by the regression model. It serves as a standardized form of MSE, affording improved clarity regarding the model's effectiveness. With a range from 0 to 1, a score of 0 implies no explanatory capability, while a score of 1 indicates a flawless fit. The computation of the R^2 value is outlined in Eq. (10).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{7}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} = \sqrt{MSE}, \tag{8}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{9}$$

$$R^2 = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{MSE}{Var(y)}, \tag{10}$$

where, n signifies the total count of observations, y_i corresponds to the actual target value, $\hat{y}_i$ signifies the predicted output value, $\bar{y}$ denotes the mean value of the actual target, and Var represents the variance.

4. Results and discussion

The performance assessment of all the models involves testing them with a dataset which was not considered during the model training phase in order to determine the superior outcomes between Group types (Group I and II). Group I involves features without including engineered features, while Group II incorporates the engineered feature. Fig. 9 illustrates the MSE of all the models across the five scenarios presented as radar plots. The features considered in each scenario are detailed in Table 3 in Section 3.4. Notably, the MSE of the majority of algorithms, when based on Group II datasets, is significantly smaller than that of Group I datasets in all scenarios, except for Scenario III. Even in the cases of Scenario I and II, the MSE error of Group II datasets is consistently smaller across all algorithms. These results collectively affirm that the applied feature engineering enhances the predictive performance of the concrete aging factor. Consequently, models exclusively based on Group II datasets are considered the best-performing models in this study.

Table 5 displays the MAE, MSE, RMSE, and R^2 values for seven machine learning models across five scenarios using the Group II dataset. Fig. 10 visually represents these statistical validation metrics for enhanced comprehension. It is noteworthy that, in Scenario II, the majority of algorithms exhibit smaller values for the statistical validation metrics compared to other scenarios. However, despite lower MAE, MSE, and RMSE indicating better prediction performance, the associated

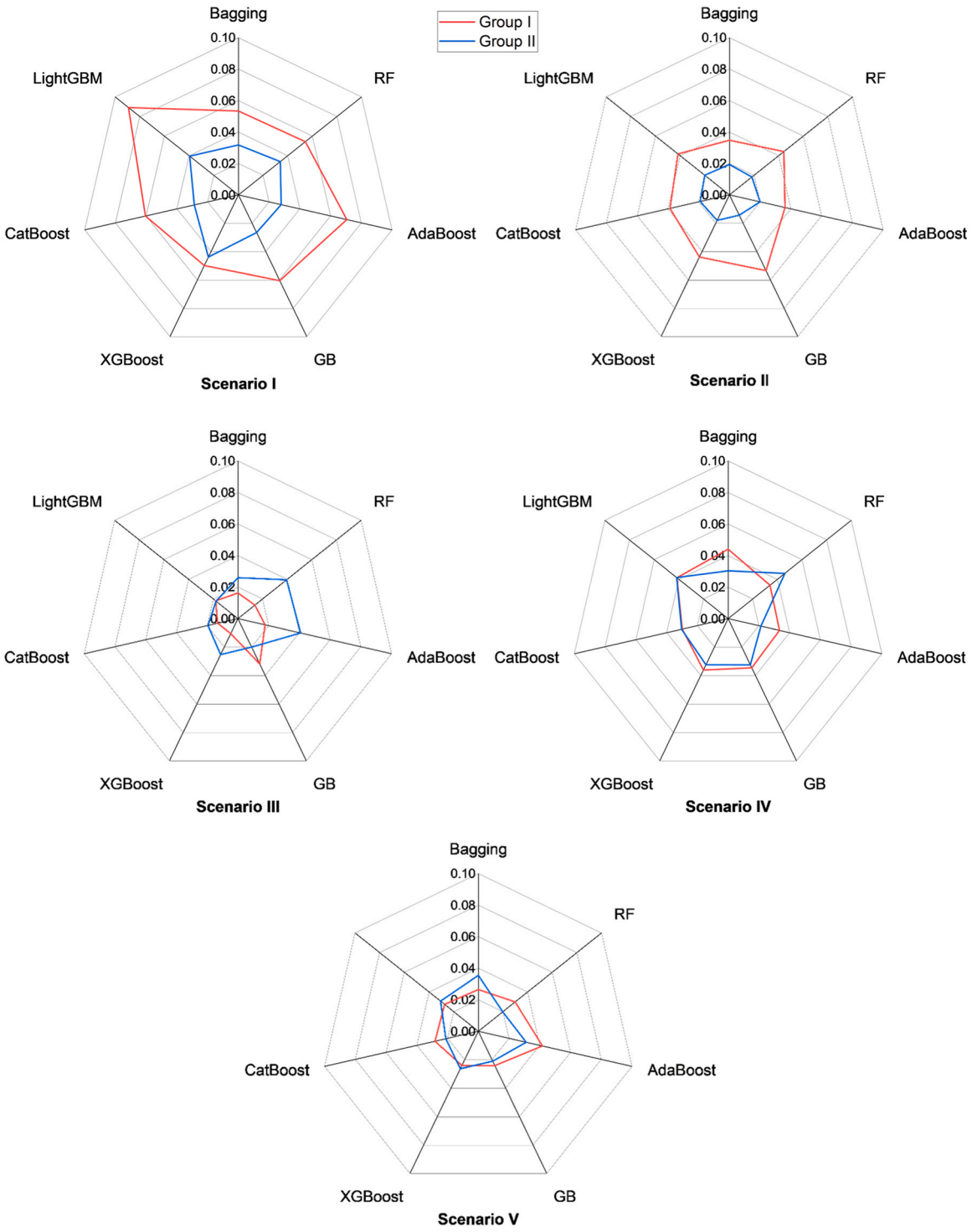

Fig. 9. The MSE of all the models across the five scenarios.

low R^2 values suggest these models do not explain a significant portion of the variance in the target feature. Consequently, models in Scenario II cannot be considered among the top-performing ones. Following Scenario II, Scenario III demonstrates the next lowest error metrics (MAE, MSE, RMSE) and the highest R^2. This scenario outperforms all others. Notably, the LightGBM algorithm excels in Scenario III, achieving values of MAE= 0.110, MSE= 0.018, RMSE= 0.133, and R^2= 0.818. Following Scenario III, Scenario V exhibits commendable performance. Here, the RF algorithm surpasses the other six algorithms, yielding values of MAE= 0.103, MSE= 0.020, RMSE= 0.141, and R^2= 0.797. Subsequent to Scenario V, Scenario IV ranks as the third-best performing scenario with MAE= 0.118, MSE= 0.021, RMSE= 0.146, and R^2= 0.802 with AdaBoost. Fig. 11 features regression plots comparing the actual and predicted concrete aging factor values for the best models from Scenarios III, V, and IV during both the training and testing phases. These plots illustrate that the predicted values closely align with the actual values, confirming that these models were well-fitted to the data during training. Table 6 details the optimal hyperparameters identified for the three top-performing models, providing insights into the tuning parameters that contributed to their superior performance.

Table 5
Statistical validation metrics for all the models across the five scenarios.

Metrics	Algorithm	Scenario I	Scenario II	Scenario III	Scenario IV	Scenario V
MAE	Bagging	0.144	0.114	0.119	0.136	0.154
	RF	0.146	0.101	0.154	0.153	0.103
	AdaBoost	0.136	0.123	0.160	0.118	0.152
	GB	0.124	0.096	0.113	0.134	0.111
	XGBoost	0.179	0.111	0.120	0.142	0.129
	CatBoost	0.132	0.111	0.116	0.134	0.105
	LightGBM	0.170	0.122	0.110	0.142	0.143
MSE	Bagging	0.032	0.019	0.026	0.030	0.035
	RF	0.034	0.018	0.040	0.046	0.020
	AdaBoost	0.028	0.020	0.041	0.021	0.031
	GB	0.027	0.014	0.020	0.032	0.021
	XGBoost	0.044	0.018	0.025	0.032	0.026
	CatBoost	0.029	0.019	0.019	0.030	0.021
	LightGBM	0.040	0.020	0.018	0.042	0.031
RMSE	Bagging	0.178	0.139	0.161	0.174	0.188
	RF	0.184	0.135	0.199	0.215	0.141
	AdaBoost	0.167	0.141	0.202	0.146	0.176
	GB	0.163	0.120	0.141	0.180	0.145
	XGBoost	0.210	0.134	0.159	0.180	0.162
	CatBoost	0.170	0.138	0.140	0.174	0.146
	LightGBM	0.199	0.142	0.133	0.204	0.175
R^2	Bagging	0.647	0.522	0.771	0.770	0.622
	RF	0.666	0.577	0.597	0.546	0.797
	AdaBoost	0.691	0.481	0.593	0.802	0.671
	GB	0.689	0.633	0.791	0.692	0.709
	XGBoost	0.590	0.545	0.742	0.737	0.767
	CatBoost	0.730	0.522	0.797	0.757	0.828
	LightGBM	0.750	0.484	0.818	0.660	0.646

In Scenario III, where the models demonstrate optimal performance, the considered input features include "Pozzolan type", "c/c+p", "w/b", "Exposure condition", and "Age of concrete". Interestingly, Scenario II, where the models perform poorly compared to all scenarios, also involves the same type of inputs. Despite both scenarios excluding the "Cement types" feature, all concrete samples in Scenario II utilized all cement types listed in Table 1. In contrast, the concrete samples in Scenario III exclusively employed ASTM I and II cement types, given their significantly higher representation compared to other types. The superior performance of models in Scenario III and the inferior performance in Scenario II affirm the significance of cement types in influencing the aging factor of concrete.

Similar to the situations in Scenario II and III, the input features in Scenario IV and V remain consistent, comprising "Cement type", "Pozzolan type", "c/c+p", "w/b", and "Age of concrete." Despite both scenarios excluding the "Exposure condition" feature, the aging factor evaluation for concrete specimens in Scenario IV was conducted solely in a lab environment, while in Scenario V, it was performed in both lab and field environments. Scenario VI involved 103 instances, whereas Scenario V had 122 instances. The superior performance of Scenario V over Scenario VI underscores that factors other than "Exposure condition" play a more crucial role in describing the concrete aging factor. Additionally, Scenario I and V share the same input features, with the only difference being that Scenario I comprises all seven exposure conditions presented in Table 1, while Scenario V presents them as either lab or field. The grouping of exposure conditions enhances the model's performance, indicating that there is not much difference among different tests in impacting the aging factor of concrete, especially laboratory ones, as this group constitutes the majority in the dataset.

As previously discussed, Scenario III stands out for its exceptional performance, leveraging the LightGBM algorithm. Noteworthy performances are also observed in Scenario V with RF and Scenario IV with AdaBoost, underscoring the significance of exploring diverse algorithms to pinpoint the most effective models. The R^2 values of the top-performing models in these scenarios exceed 0.8. The dataset used encompasses a broad spectrum of cement and pozzolan types, as well as diverse exposure conditions. The implemented ML models demonstrate a reasonable level of generalizability, successfully capturing the interrelations among the features. To further enhance the model, increased data volume and a more detailed chemical and physical characteristics of the cement and pozzolan types are recommended. In this study, we categorize cement and pozzolan types based on their general types, but it's important to note that their characteristics may considerably vary depending on their sources. Future exploration should focus on incorporating more specific chemical details.

Despite numerous experimental investigations, a shortage persists in open-access data repositories dedicated to concrete science. To harness the advantages of rapidly advancing AI technologies in the concrete industry, it is crucial to have an open data exchange platform. This platform would facilitate the sharing of data within the scientific and concrete communities. Anticipating the future, global initiatives, particularly those championed by technical committees like ACI [55] and RILEM [56], could potentially lay the groundwork for creating repositories specifically tailored to this objective. This would significantly contribute to broader and more diverse datasets, ultimately fostering the development of more robust and accurate models. These accurate models are crucial for effectively inferring knowledge about the impact of each feature on the concrete aging factor, guiding efficient concrete durability design. Additionally, the capacity of machine learning methods to dynamically enhance their performance by assimilating new relevant data ensures that the model stays current and can make reasonably accurate and precise predictions as new information becomes available. Furthermore, the proposed machine learning-based models for predicting concrete aging factors have economic implications, as they enable the determination of the aging factor in concrete samples employing different types of ingredients without the need for labor- and resource-intensive laboratory testing.

5. Conclusions

This research aimed to formulate prediction models for concrete aging factors by employing seven ensemble learning algorithms based on decision trees, namely Bagging, RF, AdaBoost, GB, XGBoost, CatBoost and LightGBM. A total of seventy models were trained under five sce-

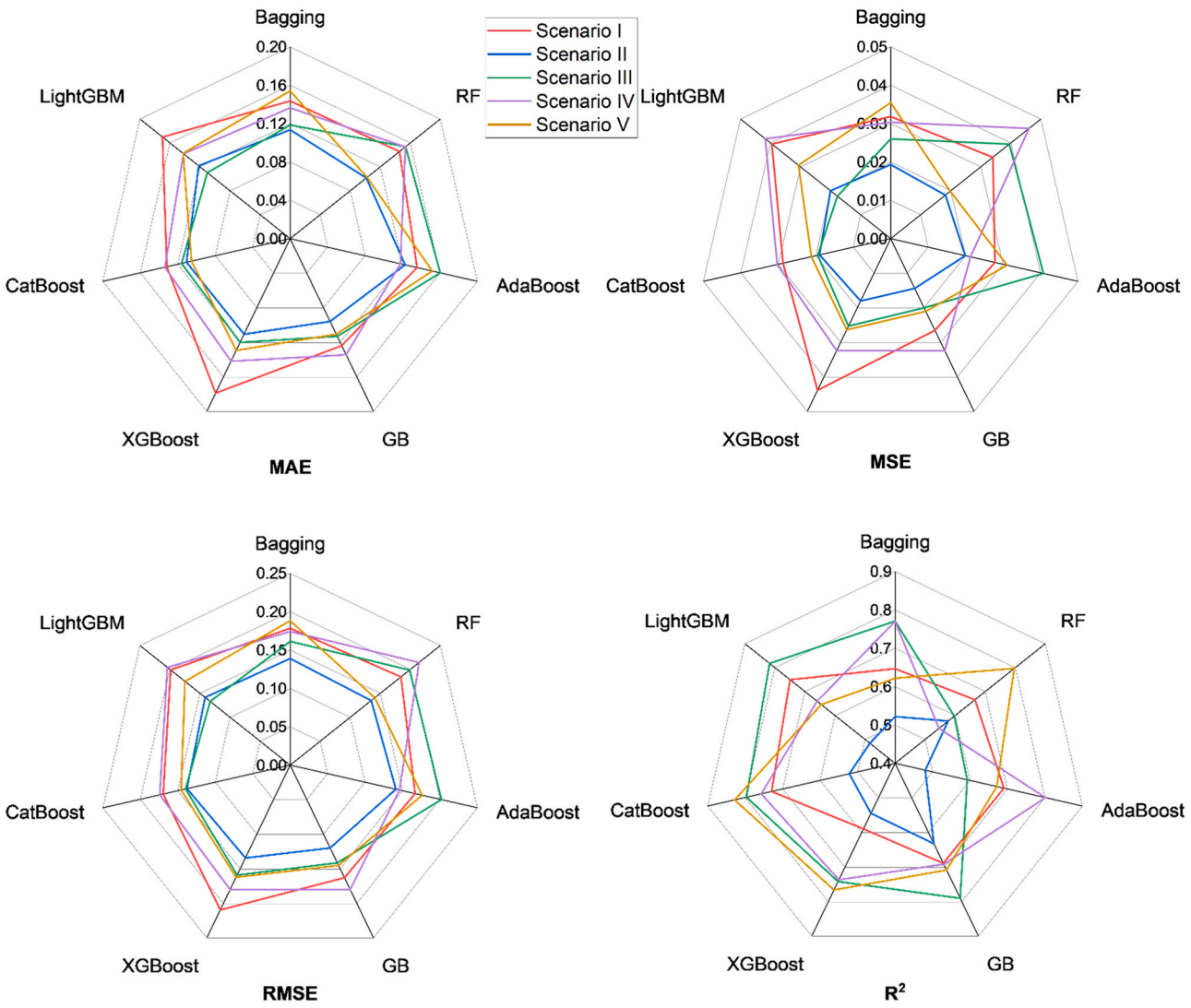

Fig. 10. MAE, MSE, RMSE, and R-Square for all the models across the five scenarios.

narios, each featuring distinct input feature conditions categorized into two groups: Group I incorporating all eight features from the raw dataset and Group II integrating engineered features. Rigorous fine-tuning of algorithm hyperparameters was conducted using a combination of grid search and 10-fold cross-validation to optimize model performance. The key findings distilled from this study are concisely outlined as follows:

- Application of feature engineering, translating the features "Cement content" and "Pozzolan content" to a single feature which is the ratio of cement content to sum of cement and pozzolan content ("c/c+p") enhance the prediction accuracy of the concrete aging factor prediction models.
- Scenario III, utilizing the LightGBM algorithm, emerges as the top-performing scenario with notable metrics including MAE= 0.110, MSE= 0.018, RMSE= 0.133, and R^2= 0.818. Following closely, Scenario V demonstrates commendable performance with RF, while Scenario IV ranks as the third-best performer with AdaBoost. The superiority of Scenario III is particularly noteworthy as it exclusively considers cement of ASTM I and II, reaffirming the significance of cement types in influencing the concrete aging factor.
- The best-performing algorithm varies across scenarios, affirming the significance of exploring diverse algorithms to identify the most effective models among the available choices.
- Considering a broad spectrum of cement and pozzolan types, as well as diverse exposure conditions, the implemented ML models demonstrate a reasonable level of generalizability, successfully capturing the interrelations among the features without the need for resource-intensive experimental testing.
- Examining the ML models with representing the cement and pozzolan types with their characteristics is highly suggested as the same types of cement or pozzolans may have different characteristics, for better representation and thus enhanced prediction accuracy.

Overall, this paper demonstrates the potential use of machine learning techniques in estimating the long-term behavior of concrete diffusivity based on concrete mix design parameters and exposure conditions. Therefore, the application of these techniques emerges as an alternative to time-consuming experimental analyses aimed at determining the concrete aging factor, contributing significantly to the service-life design of reinforced concrete structures.

CRediT authorship contribution statement

Leonardo Espinosa-Leal: Writing – review & editing, Project administration. **Fábio Costa Magalhães:** Writing – review & editing, Project administration, Conceptualization. **Woubishet Zewdu Taffese:**

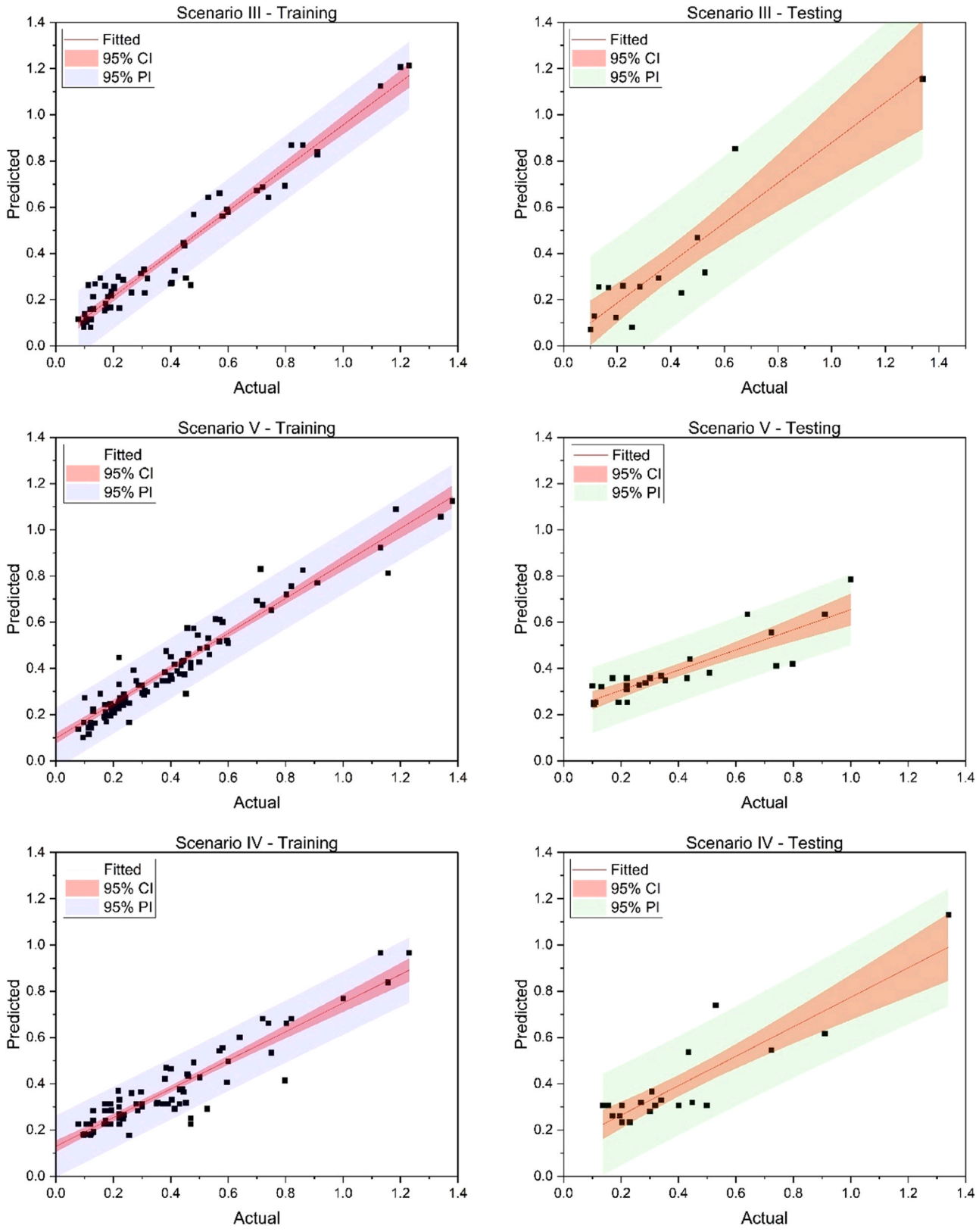

Fig. 11. Regression plots for the top three concrete aging prediction models during training and testing phases.

Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis. **Gustavo Bosel Wally:** Writing – original draft, Data curation, Conceptualization.

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.

Table 6
Optimal hyperparameters of the best three performing models.

Scenario	Algorithms	Optimal hyperparameters
Scenario III	LightGBM	{n_estimators = 600, max_depth = 4, min_split_loss = 20, learning_rate = 0.01, and booster = dart}
Scenario V	RF	{'bootstrap': True, 'max_depth': 20, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 20}
Scenario IV	AdaBoost	{'learning_rate': 0.1, 'loss': 'linear', 'n_estimators': 20}

Data Availability

Data will be made available on request.

References

[1] H. Beushausen, R. Torrent, M.G. Alexander, Performance-based approaches for concrete durability: state of the art and future research needs, Cem. Concr. Res. 119 (2019) 11–20, https://doi.org/10.1016/j.cemconres.2019.01.003.

[2] U.M. Angst, Steel corrosion in concrete – Achilles' heel for sustainable concrete? Cem. Concr. Res. 172 (2023) 107239 https://doi.org/10.1016/j.cemconres.2023.107239.

[3] ACI 318, Building Code Requirements for Structural Concrete, American Concrete Institute, 2019.

[4] EN 206, Specification, Performance, Production and Conformity, European Committee for Standardization, 2013.

[5] NBR 6118, Design of Concrete Structures – Procedure, (in Portuguese), Associação Brasileira de Normas Técnicas, 2014.

[6] V. Baroghel-Bouny, T.Q. Nguyen, P. Dangla, Assessment and prediction of RC structure service life by means of durability indicators and physical/chemical models, Cem. Concr. Compos. 31 (2009) 522–534, https://doi.org/10.1016/j.cemconcomp.2009.01.009.

[7] G.B. Wally, F.C. Magalhães, F.K. Sell Junior, F.R. Teixeira, M. de Vasconcellos Real, Estimating service life of reinforced concrete structures with binders containing silica fume and metakaolin under chloride environment: durability indicators and probabilistic assessment, Mater. Struct. 54 (2021) 98, https://doi.org/10.1617/s11527-021-01698-7.

[8] R.J. Torrent, Bridge durability design after EN standards: present and future, Struct. Infrastruct. Eng. 15 (2019) 886–898, https://doi.org/10.1080/15732479.2017.1414859.

[9] G.B. Wally, F.C. Magalhães, L.C. Pinto da Silva Filho, From prescriptive to performance-based: an overview of international trends in specifying durable concretes, J. Build. Eng. 52 (2022) 104359, https://doi.org/10.1016/j.jobe.2022.104359.

[10] F. Presuel-Moreno, Y.-Y. Wu, Y. Liu, Effect of curing regime on concrete resistivity and aging factor over time, Constr. Build. Mater. 48 (2013) 874–882, https://doi.org/10.1016/j.conbuildmat.2013.07.094.

[11] K. Stanish, M. Thomas, The use of bulk diffusion tests to establish time-dependent concrete chloride diffusion coefficients, Cem. Concr. Res. 33 (2003) 55–62, https://doi.org/10.1016/S0008-8846(02)00925-0.

[12] F. Favretto, F.C. Magalhães, A.T. da C. Guimarães, M.Á. Climent, M. de V. Real, Modelos de estimativa do grau de saturação do concreto a partir das variáveis ambientais aplicados à análise de confiabilidade de estruturas de concreto armado atacadas por íons cloreto, Mat.éria (Rio De. Jan.) 26 (2021), https://doi.org/10.1590/s1517-707620210003.13001.

[13] DuraCrete, DuraCrete final technical report: probabilistic performance based durability design of concrete structures, 2000.

[14] fib (International Federation for Structural Concrete), fib Model Code for Concrete Structures (2020), 2023.

[15] G.B. Wally, M. da C. Larrossa, L.C. de L. Pinheiro, M. de V. Real, F.C. Magalhães, 6-month evaluation of concrete aging factor using chloride migration test: effects of binder type and w/b ratio, Materials 30 (2023) 101841, https://doi.org/10.1016/j.mtla.2023.101841.

[16] S. Marsland. Machine learning: An algorithmic perspective, 2nd ed., CRC Press, Boca Raton, FL, USA, 2015.

[17] R. Kumar Tipu, V.R. Panchal, K.S. Pandya, An ensemble approach to improve BPNN model precision for predicting compressive strength of high-performance concrete, Structures 45 (2022) 500–508, https://doi.org/10.1016/j.istruc.2022.09.046.

[18] W.Z. Taffese, Y. Zhu, G. Chen, Ensemble-learning model based ultimate moment prediction of reinforced concrete members strengthened by UHPC, Eng. Struct. 305 (2024) 117705, https://doi.org/10.1016/j.engstruct.2024.117705.

[19] R.K. Tipu, V. Batra, Suman, K.S. Pandya, V.R. Panchal, Enhancing load capacity prediction of column using eReLU-activated BPNN model, Structures 58 (2023) 105600, https://doi.org/10.1016/j.istruc.2023.105600.

[20] W.Z. Taffese, Y. Zhu, G. Chen, Utilizing ensemble learning in the classifications of ductile and brittle failure modes of UHPC strengthened RC members, Arch. Civ. Mech. Eng. 24 (2024) 86, https://doi.org/10.1007/s43452-024-00897-7.

[21] W.Z. Taffese, L. Espinosa-Leal, Unveiling non-steady chloride migration insights through explainable machine learning, J. Build. Eng. 82 (2024) 108370, https://doi.org/10.1016/j.jobe.2023.108370.

[22] 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.

[23] 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.

[24] W.Z. Taffese, E. Sistonen, Significance of chloride penetration controlling parameters in concrete: ensemble methods, Constr. Build. Mater. 139 (2017) 9–23, https://doi.org/10.1016/j.conbuildmat.2017.02.014.

[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] W.Z. Taffese, L. Espinosa-Leal, A machine learning method for predicting the chloride migration coefficient of concrete, Constr. Build. Mater. 348 (2022) 128566, https://doi.org/10.1016/j.conbuildmat.2022.128566.

[28] W.Z. Taffese, E. Sistonen, Machine learning for durability and service-life assessment of reinforced concrete structures, Recent Adv. Future Dir., Autom. Constr. 77 (2017) 1–14, https://doi.org/10.1016/j.autcon.2017.01.016.

[29] P. Cichosz, Data Mining Algorithms: Explained Using R, John, Wiley & Sons, Ltd, Chichester, United Kingdom, 2015, https://doi.org/10.1002/9781118950951.

[30] E. Alpaydin. Introduction to machine learning, 2nd ed., MIT press, Cambridge, MA, USA, 2020 https://doi.org/10.1017/S0269888910000056.

[31] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140, https://doi.org/10.1007/BF00058655.

[32] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1997) 119–139, https://doi.org/10.1006/jcss.1997.1504.

[33] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (2001) 1189–1232, https://doi.org/10.1214/aos/1013203451.

[34] 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.

[35] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, A. Gulin, CatBoost: Unbiased Boosting with Categorical Features, In: Advances in Neural Information Processing Systems 31 (NeurIPS 2018), 2018.

[36] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, in: Advances in Neural Information Processing Systems 30 (NIPS 2017), 2017.

[37] M. Shekarchi, A. Rafiee, H. Layssi, Long-term chloride diffusion in silica fume concrete in harsh marine climates, Cem. Concr. Compos. 31 (2009) 769–775, https://doi.org/10.1016/j.cemconcomp.2009.08.005.

[38] M.D.A. Thomas, P.B. Bamforth, Modelling chloride diffusion in concrete, Cem. Concr. Res. 29 (1999) 487–495, https://doi.org/10.1016/S0008-8846(98)00192-6.

[39] A. Boddy, R.D. Hooton, K.A. Gruber, Long-term testing of the chloride-penetration resistance of concrete containing high-reactivity metakaolin, Cem. Concr. Res. 31 (2001) 759–765, https://doi.org/10.1016/S0008-8846(01)00492-6.

[40] H.S. Al-alaily, A.A.A. Hassan, Time-dependence of chloride diffusion for concrete containing metakaolin, J. Build. Eng. 7 (2016) 159–169, https://doi.org/10.1016/j.jobe.2016.06.003.

[41] C. Andrade, M. Castellote, R. d'Andrea, Measurement of ageing effect on chloride diffusion coefficients in cementitious matrices, J. Nucl. Mater. 412 (2011) 209–216, https://doi.org/10.1016/j.jnucmat.2010.12.236.

[42] P.S. Mangat, B.T. Molloy, Prediction of long term chloride concentration in concrete, Mater. Struct. 27 (1994) 338–346, https://doi.org/10.1007/BF02473426.

[43] M. Nokken, A. Boddy, R.D. Hooton, M.D.A. Thomas, Time dependent diffusion in concrete—three laboratory studies, Cem. Concr. Res. 36 (2006) 200–207, https://doi.org/10.1016/j.cemconres.2004.03.030.

[44] K. Audenaert, Q. Yuan, G. De Schutter, On the time dependency of the chloride migration coefficient in concrete, Constr. Build. Mater. 24 (2010) 396–402, https://doi.org/10.1016/j.conbuildmat.2009.07.003.

[45] 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/https://doi.org/10.1016/B978-0-12-416602-8.00006-6.

[46] D. Cortes, Isolation forests: looking beyond tree depth, ArXiv:2111.11639v1 11639 (2021).

[47] W. Zhao, Y. Zhang, Y. Zhu, P. Xu, Anomaly detection of aircraft lead-acid battery, Qual. Reliab Eng. Int. 37 (2021) 1186–1197, https://doi.org/10.1002/qre.2789.

[48] 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.

[49] C. Cakiroglu, F. Batool, K. Islam, M.L. Nehdi, Explainable ensemble learning predictive model for thermal conductivity of cement-based foam, Constr. Build. Mater. 421 (2024) 135663, https://doi.org/10.1016/j.conbuildmat.2024.135663.

[50] J. Wang, J. Cao, Z. Liu, Unsupervised machine learning-based multi-attributes fusion dim spot subtle sandstone reservoirs identification utilizing isolation forest, Geoenergy Sci. Eng. 234 (2024) 212626, https://doi.org/10.1016/j.geoen.2023.212626.

[51] F.T. Liu, K.M. Ting, Z.H. Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 413–422, https://doi.org/10.1109/ICDM.2008.17.

[52] R.C. Ripan, I.H. Sarker, M.M. Anwar, Md.H. Furhad, F. Rahat, M.M. Hoque, M. Sarfraz, An isolation forest learning based outlier detection approach for effectively classifying cyber anomalies, in: A. Abraham, T. Hanne, O. Castillo, N. Gandhi, T. Nogueira Rios, T.P. Hong (Eds.), Hybrid Intelligent Systems, Springer, Cham, 2021, pp. 270–279, https://doi.org/10.1007/978-3-030-73050-5_27.

[53] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, 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.

[54] 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.

[55] ACI Committee 135, Machine Learning-Informed Construction and Design, 2023 (2023). ⟨https://www.concrete.org/committees/directoryofcommittees/acommitteehome.aspx?Committee_Code=C0013500⟩ (accessed December 8, 2023).

[56] RILEM TC DCS, DCS: Data-driven concrete science, (2022). ⟨https://www.rilem.net/groupe/dcs-data-driven-concrete-science-444⟩ (accessed December 8, 2023).