Automation in Construction

Machine Learning-based Prediction of Porosity for Concrete Containing Supplementary Cementitious Materials

--Manuscript Draft--

Manuscript Number:	AUTCON-D-21-00636
Article Type:	Original article
Keywords:	Machine Learning; Decision trees; Ensemble learning; Concrete durability; Porosity
Corresponding Author:	Chong Cao University of California Los Angeles Los Angeles, UNITED STATES
First Author:	Chong Cao
Order of Authors:	Chong Cao
Abstract:	Porosity has been identified as the key indicator of the durability properties of concrete exposed to aggressive environments. This paper applies ensemble learning to predict porosity of concrete containing supplementary cementitious materials. The concrete samples are characterized by eight composition features including w/b ratio, binder content, fly ash, GGBS, superplasticizer, coarse/fine aggregate ratio, curing condition and curing days. The assembled database consists of 240 data records, featuring 74 unique concrete mixture designs. The proposed machine learning algorithms are trained on 180 observations (75%) chosen randomly from the data set and then tested on the remaining 60 observations (25%). The numerical experiments suggest that the regression tree ensembles can accurately predict the porosity of concrete from its mixture compositions. Gradient boosting trees generally outperforms random forests in terms of prediction accuracy. For random forests, the out-of-bag error based hyperparameter tuning strategy is found to be much more efficient than k-Fold Cross-Validation.
Suggested Reviewers:	Woubishet Zewdu Taffese, PhD Aalto University woubishet.taffese@aalto.fi Dr. Woubishet Zewdu Taffese is an expert in machine learning methods. He has written a review article about machine learning applications in concrete durability.
	Hongyan Ma, PhD Assistant Professor, Missouri University of Science and Technology mahon@mst.edu Dr. Ma is an expert in concrete durability. His research studies are mainly concerned with measuring and multi-scale modeling of transport properties of concrete. He has applied machine learning algorithms for empirical prediction of concrete durability properties.
	Nele De Belie, PhD Professor, Ghent University: Universiteit Gent Nele.Debelie@UGent.be Dr. De Belie is an expert in durability of cement-bond materials. She has published several paper addressing concrete porosity measurements, which have been cited and used as database in this research.
	Mohammad Iqbal Khan, PhD Professor, King Saud University miqbal@ksu.edu.sa Dr. Khan has contributed many papers by applying machine learning methods in Civil Engineering and construction materials. Particularly, he is an expert in the use of supplementary cementitious materials in concrete durability.
	Guowei Ma, PhD Professor, The University of Western Australia guowei.ma@uwa.edu.au Dr. Ma has contributed many papers by applying machine learning methods in Civil Engineering and construction materials.

	Website: https://research-repository.uwa.edu.au/en/persons/guowei-ma

Manuscript Revision

Round 2: Reviewer #1 Comments		Authors' Responses
The author does not address the provided comments fully. The introduction and discussion sections still need major improvement.		(a) Introduction Section We have added some literature review regarding the existing approaches as well as challenges in predicting porosity of concrete with SCMs. Besides, we've added a new Chapter 5 to compare the traditional chemo-mechanical model developed by Papadakis (1999) with Random Forests. The modeling results show that the machine learning method easily outperforms the traditional analytical approach. This shows the motivation of applying data-driven approach to predict high performance concrete properties. (b) Discussion Section We've enriched our discussion section by adding a new Discussion Section 4.3. The purpose is to explain the following three aspects that have been brought up by reviewer in the 1st and 2nd rounds of reviews: ▪ Algorithm convergence for regression tree ensembles ▪ Out-of-bag error vs. testing error in Random Forests ▪ Number of trees in Random Forests It is important to mention that we offer no objection to the reviewer's comments. For example, we've shown that it's not appropriate to compare *out-of-bag* error in Random Forest with testing error in Gradient Boosting Trees. Also, we've clearly mentioned that the number of trees should be properly adjusted according to the computational cost of Random Forests optimization in the case of large data set.
1	There are also flaws. For instance, the author claims 75/25 random partitions for both Gradient Boosting Trees and Random Forest. In another part, he described the data split for Random Forest is 63/37. The latter sound correct as training and test subset is formed on the embedded sampling procedure that underline in the bagging method. I alerted the author about this issue in the first review but it is not corrected in the revised version. I am afraid that the	In this study, the 75/25 random partition of dataset is utilized for both Gradient Boosting Trees (GBT) and Random Forests (RF). The purpose is to compare the prediction accuracy of the two methods on the same testing subset, i.e. testing error. On the training data set, RF is optimized with OOB error, while GBT is optimized with *k*-fold Cross-Validation error. This means that the bootstrap 63/37 partition in RF is only performed on the training subset (the 75% training data) for training/optimization purpose. To support our methodology, we've plotted the evolution of

	author does not have solid knowledge about machine learning.	testing error along with OOB error for RF. The testing error settles down together with OOB error as the number of trees increases, which clearly indicates the convergence of the proposed optimization algorithm. However, the OOB error is relatively higher than the testing error for RF. Similar for GBT, the *k*-fold CV error is higher than testing error. Therefore, to ensure a fair comparison between RF and GBT, we use testing error on the same testing subset. To be clear, we never compare the *out-of-bag* error of RF with the testing error of GBT. We've added a new discussion paragraph to explain this aspect. (red text on p.22) Moreover, the Matlab code is available online at GitHub if the reviewer wants to check details. (https://github.com/againstlaw/Porosity)
2	There is no explanation of how importance of variables is determined.	As per reviewer's suggestion, we've added Fig. 8 to explain the algorithm for computing *out-of-bag* predictor importance by permutation. (red text on p.19)
3	Moreover, the language still needs revision.	The paper has been thoroughly reviewed in terms of English language.

Round 1: Reviewer #1 Comments		Authors' Responses
The author adopted machine learning algorithms (random forest and gradient boosted tree) to predict porosity of high-performance concrete containing supplementary cementitious materials. The introduction section motivates well why the author focus on eight parameters for predicting the porosity of concrete. The major drawback of the paper is that it only compares the performance of the adopted algorithms in predicting the porosity of concrete. It must be compared with the traditional model. The paper does not motivate the need for machine learning algorithms to predict porosity. The discussion section is not in-depth. All the facts mentioned in Section 4.2 "sensitivity analysis" can be deduced without the use of the learning algorithms. I do not see any connection with the developed models. The author should also take the following comments into consideration and improve the work accordingly.		(a) Traditional Model We have added some literature review regarding the existing approaches as well as challenges in predicting porosity of concrete with SCMs. Besides, we've added a new Chapter 5 to compare the traditional chemo-mechanical model developed by Papadakis (1999) with Random Forests. The modeling results show that the machine learning method easily outperforms the analytical approach. This clearly demonstrates the motivation of applying data-driven approach to predict high performance concrete properties. (b) Discussion Section We've enriched our discussion section by adding more interpretation of the numerical results. After the most relevant variables have been identified, the next step is to understand the nature of the dependence of porosity on the joint values of critical predictors. One good thing with machine learning algorithm is that partial dependent plot (PDP) could show the relationship between a predictor and the response of regression in the trained model. Therefore, we've added PDPs (Fig. 10) on the four most relevant predictors in Section 4.1. In Section 4.2 "Sensitivity Analysis", we've added 2-Dimensional PDPs (Fig. 12 and Fig. 14) to illustrate the interaction between key features in fly ash concrete and GGBS concrete. This is the unique feature of data-driven approach, which may marginalize out the effect of the other variables. We've also added a new Discussion Section 4.3 to explain the following three aspects that have been brought up by the reviewer: ▪ Algorithm convergence for regression tree ensembles ▪ Out-of-bag error vs. testing error in Random Forests ▪ Number of trees in Random Forests
1	In the highlights "Established ensemble learning framework for empirically predicting porosity of concrete containing supplementary cementitious materials." I do not think it is a right claim. Rewriting is	As per reviewer's suggestion, the bullet point in the highlights has been revised as "Applied ensemble learning to predict porosity of concrete containing supplementary cementitious materials".

	required e.g. "Developed concrete porosity predictor using ensemble method".	
2	In the abstract it is mentioned that "Compositions of concrete are characterized by 8 features including w/b ratio, binder content, fly ash, GGBS, superplasticizer, coarse/fine aggregate, curing condition and curing days.". This is a misleading sentence as it is not all the time. You should specifically mention that in this study concrete characterized by eight parameters is utilized…	As per reviewer's suggestion, the description in Abstract has been revised as "The concrete samples utilized in this study are characterized by eight composition features including …".
3	In the introduction section, it is worth to mention other studies which followed the same approached (ensemble methods) to characterize concrete properties. e.g. http://dx.doi.org/10.1016/j.conbuildmat.2017.02.014 https://doi.org/10.1016/j.engappai.2013.03.014	We have written a new literature review paragraph regarding Ensemble Trees in the Introduction section. And we have added more references that apply ensemble learning to predict concrete properties.
4	It is mentioned that "The out-of-bag error based hyperparameter tuning strategy for random forests is found to be much more efficient than k-Fold Cross-Validation." In what criteria? Specify the detail in the body part of the manuscript. As far as I understood, K-fold cross validation is used for gradient boosted tree and OBB for random forest. So how you could compare two optimizations in two different algorithms? It is incorrect approach.	Random Forests outperforms Gradient Boosting in terms of computational efficiency. First, it has fewer hyperparameters to tune. Second, *out-of-bag* error can be conveniently used to replace cross-validation error for tuning purpose. We've added detailed explanation of computational efficiency of Random Forests.
5	Lines 129-130 "the full dataset is randomly divided into two groups: training dataset (75%) and testing dataset (25%)." This is misleading as it seems for both learning algorithms. Explicitly mention for which algorithm is this portioning is valid. Revise the abstract and the conclusion sections, accordingly.	In this study, the 75/25 random partition of dataset is utilized for both Gradient Boosting Trees (GBT) and Random Forests (RF). The purpose is to compare the prediction accuracy of the two methods on the same testing subset, i.e. testing error. On the training data set, RF is optimized with OOB error, while GBT is optimized with *k*-fold Cross-Validation error. This means that the bootstrap 63/37 partition in RF is only performed on the training subset (the 75% training data) for training/optimization purpose. To support our methodology, we've plotted the evolution of testing error along with OOB error for RF. The testing error settles down together with OOB error as the number of trees increases, which clearly indicates the convergence of the proposed optimization algorithm. However, the OOB error is relatively higher than the testing error for RF. Similar for GBT, the *k*-fold CV error is higher than testing error. Therefore, to ensure a fair comparison between RF and GBT, we use testing error on the same testing subset.

		To be clear, we never compare the *out-of-bag* error of RF with the testing error of GBT. We've added a new discussion paragraph to explain this aspect. (red text on p.22)
6	Lines 132-135. Long sentence and not clear enough. Rewrite it concisely.	The long sentence has been split into three short sentences. The purpose is to provide an accurate description of stratified sampling.
7	Lines 138-139 "Categorical predictor "curing condition" is transformed into numeric values by replacing "air curing" and "water curing" with 1 and 2, respectively." Tree based models can categorical data. Indeed, it is common practice to change categorical predictor to ordinal/numerical. However, the data type shall be set as a categorical, otherwise the algorithm interprets 2 has higher value that 1, which is not the case.	We agree that transforming categorical predictors into numeric values will affect how ML algorithms interpret the variable. After transform, the data type should be set as categorical. Actually, in our MATLAB program, the Categorical Predictors are clearly flagged in both Random Forest and Gradient Boosting. And Ensemble Trees in MATLAB can directly handle categorical variables without the need of transforming into numerical values. We have added clear description of the special treatment of categorical variable in the ensemble trees regression algorithm.
8	I do not think the word "framework" is an appropriate terminology for the title of section three. I suggest the author to change it to a more appropriate one.	We have revised the title of Section 3 as "Machine Learning Method".
9	Line 161 "th" should appear as superscript. Do the same for others.	We have revised *b*th into b^{th} throughout the manuscript.
10	The claim presented in lines 167-170 should be supported by relevant references.	We have added reference here.
11	In section 3.2, several hyperparameters are mentioned in ranges. Present all values considered in each hyperparameters it is not many. E.g. number of trees mentioned in range is [10, 500], I would prefer to see all values, such as [10, 50, 100, 150, 300, 150].	The values considered for each integer variable include the entire grid within the tuning range. For example, we set the candidate values for "Number of Trees" to be all integers within the whole range of [10, 500]. This will not lead to high computational cost, because Bayesian optimization could automatically search for the next point to evaluate in the bounded domain of the variables. The algorithm usually finds the optimal setting for hyperparameters within a few iterations. We've added clear statement about this in the main body text.
12	Lines 204-205 "sufficiently large number of trees (B = 300) is chosen for the regression forest." The best approach is test different size of tree and pick the one yields less MSE with small tree size. Taking large tree size make computationally expensive in large size of dataset.	Using "sufficiently large number of trees" for Random Forest has been recommended by its inventor (Breiman 2001) and the classic textbook (Hastie *et al*. 2009). As shown in Fig. 16, the variance of OOB MSE significantly reduces as B increases. However, there is modest prediction improvement after growing the first 300 trees. Therefore, we choose $B = 300$ in this study.

		Certainly, for large size of dataset, the number of trees should be properly adjusted according to the computational cost of Random Forests optimization. We've clearly mentioned this aspect in Section 4.3 Discussion.
13	In the manuscript, it is mentioned that Bayesian optimization algorithm is applied. But no further explanation how and why it is adopted in paper. Provide sufficient info about it.	The Bayesian optimization algorithm has been widely used in machine learning practices to search for the best hyperparameters. Compared with grid search and randomized search, Bayesian optimization is considerably more efficient as it can detect the optimal hyperparameter combinations by analyzing the previously-tested values. It can usually find the near-optimal hyperparameters within a few iterations. We've added detailed description of Bayesian optimization algorithm in Section 3.3.
14	Lines 307-312 shall be supported by references.	To support our arguments about Random Forest, we've added relevant references, e.g. Probst *et al*. (2019) and Hastie *et al*. (2009).
15	Lines 317 and 318 "Curing days, binder content and w/b ratio are considered to be the most critical factors in predicting concrete porosity". It is a well-known fact. What is new there?	After the most relevant variables have been identified, the next step is to understand the nature of the dependence of porosity on the joint values of critical predictors. One good thing with machine learning algorithm is that partial dependent plot (PDP) could show the relationship between a predictor and the response of regression in the trained model. We've added PDPs (Fig. 10) on the four most relevant predictors in Section 4.1. In Section 4.2, we've also added 2-D PDP to illustrate the interaction between key features in fly ash concrete and GGBS concrete. (Fig. 12 and Fig. 14)
16	Figure 1 - the unit for the permeability do not read correct.	The term "permeability" in Figure 1 refers to the "intrinsic permeability" (K), which has a unit of m^2. The "coefficient of permeability" (k) is usually measured from experiment and has a unit of m/s. The relationship between the two permeabilities can be written as $$\mathrm{K} = \frac{\mu}{\rho g} k$$ where μ ($kg \cdot m^{-1} \cdot s^{-1}$) is the dynamic viscosity of the fluid, ρ (kg/m^3) gives the density of fluid and g (m/s^2) denotes the standard gravity. We have added clear statement of "intrinsic permeability" in the text.
17	Enhance the quality of Figures 2, 5-10.	Figure 2 has been improved by adding 2-dimensional plot to clearly show the dependence of porosity on *w/b* ratio and curing days.

18	I do not think equal line presented in figures 5 and 7 is a right terminology. I recommend the author to rename as regression line or line of best fit.	The figures have been revised by changing “equal line” into “line of best fit”.
19	Avoid unnecessary visual elements from figures. For instance, the legend in Figure 8 is unnecessary. Even the secondary x and y axes are also redundant.	The figure has been improved by removing the unnecessary legend and the redundant x and y axes.
20	All the parameters should be mentioned consistently through the paper. For instance, Days and Aggregate mentioned as parameters in Figure 8, but these parameters read as curing days and Aggregate (CA/FA) in Table 1.	As per reviewer’s suggestion, efforts have been made to ensure that the predictor names are used consistently throughout the paper, e.g. Curing days and Aggregate (CA/FA). The figure has been carefully revised.

Highlights

- Applied ensemble learning to predict porosity of concrete containing supplementary cementitious materials.
- Compared gradient boosting trees with random forests in terms of prediction accuracy and computational efficiency.
- Estimated variable importance based on permutation of *out-of-bag* predictor observations in random forest of regression trees.
- Trained random forests with efficient tuning strategy based on *out-of-bag* error.

Machine Learning-based Prediction of Porosity for Concrete Containing Supplementary Cementitious Materials

Chong CAO*

University of California, Los Angeles, 110 Westwood Plaza, Los Angeles, CA 90095

*Corresponding author at:

UCLA Anderson School of Management, 110 Westwood Plaza, Los Angeles, CA 90095

Tel: +1 (310) 779 7537

Email address: jasongo@ucla.edu

Abstract

Porosity has been identified as the key indicator of the durability properties of concrete exposed to aggressive environments. This paper applies ensemble learning to predict porosity of high-performance concrete containing supplementary cementitious materials. The concrete samples utilized in this study are characterized by eight composition features including *w/b* ratio, binder content, fly ash, GGBS, superplasticizer, coarse/fine aggregate ratio, curing condition and curing days. The assembled database consists of 240 data records, featuring 74 unique concrete mixture designs. The proposed machine learning algorithms are trained on 180 observations (75%) chosen randomly from the data set and then tested on the remaining 60 observations (25%). The numerical experiments suggest that the regression tree ensembles can accurately predict the porosity of concrete from its mixture compositions. Gradient boosting trees generally outperforms random forests in terms of prediction accuracy. For random forests, the *out-of-bag* error based hyperparameter tuning strategy is found to be much more efficient than *k*-Fold Cross-Validation.

Keywords: Machine learning; Decision trees; Ensemble learning; Concrete durability; Porosity

1 Introduction

Concrete is a composite material made of aggregates and hydrated cement paste that each may contribute to the formation of interconnected pore structure. There is considerable interest in the relationship between porosity characteristics and transport properties of concrete, such as diffusion coefficient of oxygen and carbon dioxide, gas/water permeability, chloride ions migration, and electrical resistivity [1-11]. The empirical relationship between the compressive strength and the transport properties of concrete (expressed as intrinsic permeability) as a function of the capillary porosity is shown in Fig. 1. Generally, the increase in porosity will increase the permeability of concrete and reduce the mechanical strength. Therefore, porosity has been identified as one of the key parameters in determining the durability and serviceability of concrete structures subjected to aggressive environments [12].

Among the many factors that may affect the porosity of concrete, water/cement ratio (*w/c*) plays an essential role in facilitating the hydration reactions of cement paste. The volume of the capillary pores in the hydrated cement paste increases with the *w/c* ratio. As the curing days increase and the hydration proceeds, the porosity decreases as a result of the reduction in large-dimension pores that have been filled or connected by calcium-silicate-hydrate (C-S-H) gel pores. Powers proposed the classical model to calculate the volumetric composition of hardened cement paste from *w/c* ratio and degree of hydration of the cement [13]. Concrete can become more heterogeneous due to the presence of aggregates and the interfacial transition zone (ITZ). Previous experimental results show that the increase in size and proportion of coarse aggregate will lead to the increase in the local porosity at the

ITZ and thus the reduction in the overall durability properties of ordinary concrete [14]. The ratio of coarse aggregate to fine aggregate by weight (CA/FA) is found to be a critical factor influencing porosity, tortuosity and permeability of concrete [15].

Supplementary cementitious materials (SCMs) have been utilized to partially replace Portland cement for the purpose of enhancing durability and strength properties of concrete [16-19]. The use of SCMs such as fly ash, a byproduct of the combustion of coal powder in thermoelectric power plants, or ground granulated blast-furnace slag (GGBS), a byproduct of pig iron production, may also promote cleaner production by significantly reducing CO_2 emission. The beneficial effect of GGBS in concrete lies in the latent hydraulic reaction that contributes to the cement hydration process by densifying the concrete matrix and refining the pore structure. This will lead to reduction in porosity and increase in compressive strength of concrete at later ages. The change in mineralogy of the cement hydrates may also improve the chloride binding capabilities and increase the electrical resistivity of concrete [20]. The reduction in the permeability of fly ash concrete has been attributed to a combination of the reduced water content for a given workability and the refinement of pore structure due to pozzolanic reaction. Because of the long-term nature of the pozzolanic reaction, the beneficial effects associated with it become more evident in well-cured concrete, and therefore the curing condition (air curing or water curing) will be another crucial factor influencing porosity of high-performance concrete [21]. Furthermore, the addition of superplasticizers (SP) may allow for substantial reduction in the mixing water, which will facilitate the formation of denser pore structure [22].

Due to the various mixtures of concrete and the time-dependent hydration process of cement, the development of pore structure within concrete becomes very complex, which may defy analytical

78 modeling. The major difficulty lies in the uncertainties associated with the pozzolanic and hydraulic
79 reactivity of fly ash and slag. The chemical composition of SCMs can vary significantly and the
80 estimation of the reactive portion of the materials is quite challenging. Papadakis [23,24] has proposed
81 a theoretical model to predict the chemical and volumetric composition of fly ash concrete. The model
82 considers the stoichiometry of Portland cement hydration and pozzolanic reactions of fly ash as well
83 as the molar weights of reactants and products. However, this model assumes full hydration of Portland
84 cement and the complete pozzolanic reactions of fly ash and therefore can't consider the time-
85 dependent evolution of porosity. This motivates the data-driven approach to be widely adopted in
86 modeling high-performance concrete properties.
87 In order to make empirical prediction of the permeation properties of high-performance concrete, Khan
88 [25] applied multivariate regression to predict porosity based on concrete mixtures such as fly ash
89 proportion, microsilica content, and water/binder ratio (*w/b*) at different ages of 28, 90 and 180 days.
90 An alternative statistical approach for modelling concrete with complex mixture compositions would
91 be machine learning, which has been widely used to predict the mechanical strength and durability
92 properties of high-performance concrete with highly desirable accuracy [26-33]. However, little
93 literature deals with the machine learning-based prediction of concrete porosity. Boukhatem *et al*. [34]
94 presented a Neural Network modeling framework to predict compressive strength, porosity and
95 transport tortuosity of fly ash concrete using mix design parameters (water, binder, aggregates,
96 superplasticizer), fly ash content and age as input. The modeling results showed excellent correlation
97 between predicted values and experimentally obtained porosity, which suggests that machine learning
98 is a promising technique for predicting concrete porosity.

Among all the well-known machine learning methods, decision trees have emerged as the most popular supervised learning approach for data mining. Decision trees can naturally incorporate mixtures of numeric and categorical predictor variables and missing values. They are insensitive to the monotone transformation of the individual predictors and immune to the effects of predictor outliers. Moreover, decision trees are able to handle many irrelevant inputs because they perform internal feature selection as an integral part of the algorithm. Furthermore, ensemble learning can dramatically improve the prediction accuracy of decision trees by aggregating multiple weak learners [35]. Therefore, with their robust predictive performance and high interpretability, ensemble trees have been widely used to characterize concrete properties [36-38].

The objective of this paper is to apply regression tree ensembles for empirically predicting the porosity of high-performance concrete containing SCMs. First, a reliable database consisting of 240 data records for concrete porosity is assembled from published literature. The full dataset is randomly divided into 75% training dataset and 25% testing dataset. Then, both gradient boosting trees and random forests are trained to tune key hyperparameters by minimizing either *k*-Fold Cross-Validation error or *out-of-bag* error through Bayesian optimization algorithm. Finally, the optimized models will be tested on the testing data set in order to gain a measure of the prediction accuracy. Special attention will be given to the estimation of predictor importance based on ensemble trees. The proposed case study is utilized to compare the data-driven approach with classical chemo-mechanical model for the prediction of concrete porosity.

2 Experimental Database

In order to establish a reliable database for the development of the machine learning-based prediction model, experimental data of concrete porosity have been collected from published literature with certain selection criterion. First, the concrete is mixed with ordinary Portland cement (OPC), which can be partially replaced by either fly ash or GGBS. Second, all concrete specimens contain coarse aggregate and fine aggregate. Third, the curing regime falls into two categories: air curing and water curing. Fourth, the effect of carbonation on pore structure refinement is not introduced in the concrete specimens. Also, it has been determined to retain a balanced proportion for each type of concrete, i.e. ordinary Portland cement concrete (OPC), fly ash concrete and GGBS concrete.

The assembled dataset consists of 240 data records, featuring 74 unique concrete mixture designs [39-44]. Selected experimental data of concrete composition and porosity are shown in Table 1. There are eight input features: *w/b* ratio, binder content (kg/m^3), fly ash content (%), GGBS content (%), superplasticizer content (%), CA/FA ratio, curing condition (categorical predictor) and curing days. The fly ash and GGBS contents are recorded as replacement fraction of the binder (cement plus SCMs). To ensure the consistency of the database, the applied dosage of superplasticizers is reported as weight proportion of the binder used in the concrete mixture. If the SP content is recorded as volume values in the original literature, a density of 1.2 kg/L is assumed for SP in order to translate volume numbers into weight proportion. There are no missing values for all 8 predictors in the established database. The statistics pertaining to each of the continuous variables are summarized in Table 2.

Curing days and *w/b* ratio are generally considered as the critical factors influencing concrete porosity.

The visualization of porosity data against the two key parameters for different types of concrete (75 data records for OPC, 45 data for GGBS and 120 data for Fly ash) is shown in Fig. 2. There is no clear trend observed for both parameters in relation to porosity. This highlights the difficulty of predicting the porosity properties of high-performance concrete.

In the original experimental work, the concrete porosity was measured using three distinct methods: water saturation under vacuum, mercury intrusion porosimetry (MIP) and helium porosimeter. These approaches, however, are based on different theoretical background and therefore may not give the same porosity interpretations. A further discussion of the possible discrepancy resulting from different measuring techniques is beyond the scope of this study.

For the purpose of testing the prediction accuracy of the machine learning algorithm, the full dataset is randomly divided into two groups: training dataset (75%) and testing dataset (25%). An identifier column is created (column name “Training” in Table 1) to indicate which subset the data instances belong to. Stratified sampling has been employed to reduce the sampling bias associated with the random partition of the training dataset and the holdout testing dataset. The training dataset should contain the various categories of concrete in similar proportions to the overall dataset. For each concrete type, the distribution of curing days in the training dataset should be representative of that in the whole dataset. Also, it is necessary to make sure that the input variables (e.g., *w/b* ratio and binder content) of the training group contain values that span the entire range of the overall dataset.

The categorical predictor “curing condition” is not transformed into numeric values, because decision trees can directly handle combinations of numeric and categorical predictor variables. The standardization of datasets is commonly required for many machine learning algorithms if the values

of predictors vary on significantly different scales. Note that decision trees (including random forests and gradient boosting) are not sensitive to the magnitude of variables. Therefore, standardization is not needed before fitting ensemble trees.

3 Machine Learning Method

3.1 Regression tree ensembles

Regression trees is a top-down, greedy approach that performs recursive binary splitting to grow a large tree on the training data set, stopping only when the terminal node has reached certain minimum number of observations. At each split, the best partition of predictor space is found by minimizing the sum of squared residuals (RSS) for the resulting predictions. The optimum number of terminal nodes can be obtained by applying cost complexity pruning to the large tree for a trade-off between the subtree's variance and bias. For a given test observation, the prediction can be made by using the mean response for all the training observations within the terminal node to which that test observation belongs [45].

Bootstrap aggregating (Bagging) is an important ensemble learning technique to reduce the variance of decision trees [46]. To apply bagging to regression trees, many bootstrapped replicas of the original training data set are first generated by repeatedly random sampling with replacement. Then, separate predictions are made by constructing B independent regression trees using B different bootstrapped training data sets. The final prediction can be obtained by averaging all the resulting predictions as

follows

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x) \quad (1)$$

where $\hat{f}^{*b}(x)$ represents the prediction obtained from the b^{th} bootstrapped training data set, and B is the total number of bootstrapped training data sets. Using a sufficiently large number of bagged trees (B) can significantly reduce the prediction error without worrying about overfitting.

In addition, every tree in the ensemble can randomly select a subset of predictors for each decision split, a technique called Random Forests known to improve the accuracy of bagged trees when applied to high-dimensional data set consisting of highly correlated predictors [47]. As per the inventors' recommendations, the default value for the number of predictors to select at random for each split is approximately equal to one third of the total number of predictors for regression (minimum node size of 5) or square root of the total number of predictors for classification (minimum node size of 1) [35]. In practice, the best value for the number of predictors to select will depend on the problem and therefore should be treated as a tuning parameter. The simple example of bagging two regression trees in Random Forests is visualized in Fig. 3.

Boosting is another popular approach for improving the prediction accuracy of decision trees. Unlike bagging, gradient boosting does not involve bootstrap sampling or growing independent regression trees, but constructs new tree to the residual errors resulting from the aggregated prediction of all trees grown previously. The new decision trees, usually small with just a few terminal nodes, are sequentially added into the fitted function in order to update the residuals. This vanilla gradient boosting algorithm is often referred to as Least-squares Boosting (LSBoost), which applies gradient boosting on squared-error loss function so that the negative gradient is just the ordinary residual [48].

The LSBoost algorithm for regression is briefly illustrated in Fig. 4. The final output of the boosted regression tree ensembles can be obtained as the weighted sum of the predictions from all trees

$$\hat{f}_{\text{boost}}(x) = \sum_{b=1}^{B} \lambda \hat{f}^{b}(x) \tag{2}$$

where $\hat{f}^{b}(x)$ represents the prediction resulting from the b^{th} residual correcting tree, B is the total number of boosted trees, and λ denotes the shrinkage parameter that controls the residual updating process. Unlike Bagging and Random Forests, Boosting can overfit if the total number of boosted trees (B) is too large. In the case of overfitting for gradient boosting, typical remedial measures may include reducing the number of learning cycles, limiting the tree depth and decreasing the learning rate (λ).

3.2 Hyperparameter tuning

In order to achieve best prediction performance for the machine learning process, hyperparameters should be tuned by optimization algorithms, such as grid search, randomized search and Bayesian optimization. The hyperparameters for ensemble trees include number of trees, maximum number of decision splits per tree, minimum number of terminal node observations, learning rate for shrinkage (for LSBoost), and number of predictors to select at random for each split (for Random Forests). The typical hyperparameters for both Random Forests and Gradient Boosting and their tuning range considered in the present study are listed in Table 3. Note that the values considered for each integer variable include the entire grid within the tuning range.

Learning rate (λ) is a critical regularization parameter for gradient boosting to shrink the contribution of each individual tree grown in the ensemble. Small learning rate (more shrinkage) will typically

require a large number of learning cycles in order to achieve good prediction performance. However, overfitting may occur if the number of trees (B) is too large. The computation time increases linearly with the number of learning cycles. In this study, the number of trees is set to be tunable in the range of [10, 500] for LSBoosting and the learning rate can be chosen between 0.001 and 1. In the case of Random Forests, the number of trees is typically not tunable but should be set large enough for convergence. The prediction performance measures, e.g. mean squared error, generally show monotonously decreasing pattern with the increasing number of trees [49]. Moreover, the more trees are trained, the more stable the predictions should be for the variable importance [50]. Thus, a sufficiently large number of trees (B = 300) is chosen for the regression forest.

The maximum number of decision splits per tree is often referred to as the maximum tree depth, which determines the complexity of the individual trees. The deeper a tree, the more likely it overfits the training data set. But a too shallow tree might not allow for enough feature interactions. Another factor that controls the tree depth is the minimum number of terminal node observations. The smaller the minimum terminal node size, the deeper the trees can grow. The default values of the tree depth controllers for boosting regression trees are 10 for maximum number of splits per tree and 5 for minimum terminal node size in MATLAB [51], which indicate that shallow decision trees are grown. In this study, the maximum number of splits per tree is limited to be in the range of [1, 20] for LSBoosting. Meanwhile, the minimum size of terminal node observations is set to be tunable between 1 and 90 by default.

Random Forests, on the other hand, can achieve good prediction performance by simply using full-grown trees, according to its inventor' arguments [35]. The default maximum number of decision splits

241 is *n*-1 for bagging tree ensembles in MATLAB, where *n* is the number of observations in the training
242 data set. The minimum number of observations per tree leaf is set as 5 by default. The experimental
243 study by Segal [52] suggests that prediction performance gains can be realized by controlling the depth
244 of the individual trees grown in the regression forest, especially for certain data sets with large number
245 of noise variables. In this study, to find the optimal tree depth for Random Forests, the minimum
246 terminal node size is treated as a tunable hyperparameter and specified to be at most 20. The maximum
247 number of splits per tree is using the default parameter, which is 179 for the training data set of concrete
248 porosity.
249 The number of predictors to select at random for each split is a critical parameter for Random Forests.
250 Smaller number of randomly drawn candidate variables leads to less correlated trees, which may yield
251 better stability when aggregating. This works particularly well when there are a large number of
252 correlated predictors. However, for high dimensional data with only a small fraction of relevant
253 variables (e.g. genetic datasets [53]), Random Forests is likely to perform poorly with a small subset
254 of predictors. This is because the subtrees with selected groups of irrelevant variables may add
255 additional noise into the trees and therefore reduce the ensemble prediction accuracy. For the concrete
256 porosity data set considered in this study, the number of predictors to select at random for each split is
257 set to be tunable in the range of [1, 8]. For boosted tree, all predictors should be selected at each split
258 in order to precisely analyze the predictor importance [51].
259

3.3 Optimization algorithm

To optimize the hyperparameters for the machine learning algorithm, *k*-Fold Cross-Validation (CV) is commonly employed to train the dataset and estimate the prediction error. This approach involves randomly dividing the entire training data set into *k* distinct groups (folds) of approximately equal size. With *k*-1 folds of the observations treated as the training data set, the mean squared error (MSE) of prediction can be computed on the observations in the hold-out fold (testing data set). After repeating the procedure for *k* times, the *k*-fold CV loss estimate is computed by averaging these values

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i \tag{3}$$

where *k* is typically chosen as 5 or 10 for the bias-variance trade-off [45]. The hyperparameters for ensemble trees can be optimized by minimizing the *k*-fold CV loss. By default, the optimization objective function for regression is log(1 + 10-fold CV loss) in MATLAB [51]. In the present study, the gradient boosted trees (LSBoost) are optimized through 10-fold CV.

An important feature of bagging is that it offers a computationally efficient way to estimate the test error, without the need to perform cross-validation. For each of the bagging iterations, approximately 63.2% of the original training data set is selected as the bootstrapped sample [35].

$$\Pr\{\text{observation } i \in \text{bootstrap sample } b\} = 1 - \left(1 - \frac{1}{n}\right)^n \approx 1 - e^{-1} = 0.632 \tag{4}$$

The remaining one-third of the observations that are not used to fit a given bagged tree, aka *out-of-bag* (OOB) observations, can be used as a test set to get a measure of the prediction error [54]. First, by running the entire bagging cycles, roughly *B*/3 predictions (on average) can be made for the *i*th observation using each of the trees in which that observation is OOB. Then, for each of the *n*

observations in the training data set, the corresponding predicted responses can be averaged to obtain the OOB predictions. Finally, the overall OOB mean squared error can be conveniently computed as [35]

$$\text{OOB MSE} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\left|C^{-i}\right|}\sum_{b\in C^{-i}}\left[y_i - \hat{f}^{*b}(x_i)\right]^2 \tag{5}$$

where C^{-i} is the subset of indices of the bootstrapped sample *b* that do not contain observation *i*, and $\left|C^{-i}\right|$ is the total number of such samples. To ensure that $\left|C^{-i}\right|$ is greater than zero, only observations that are out-of-bag for at least one tree are considered.

In this study, the hyperparameters for a random forest of regression trees (i.e., minimum size of terminal nodes and number of features to select at each node) are tuned by minimizing the OOB mean squared error. A custom objective function that accepts the tuning parameters as inputs is defined in MATLAB to compute the ensemble OOB MSE on the training data set. Similar hyperparameter tuning strategy based on *out-of-bag* predictions has been employed by Probst *et al*. [50] to achieve faster computing.

The Bayesian optimization algorithm has been adopted in this study to search for the best combination of hyperparameters for regression tree ensembles. Bayesian optimization typically works by assuming Gaussian process (surrogate model) for the objective function and maintains a posterior distribution for this function as the results of running machine learning algorithm experiments with different hyperparameters are observed [55]. One unique feature of Bayesian optimization is the acquisition function, which the algorithm uses to determine the point to evaluate in the next iteration. The acquisition function estimates the expected amount of improvement in the objective function over the currently available best result. It can also balance the tradeoff between exploration of new instances in

298 the areas that have not yet been sampled and exploitation of the already examined area based on the
299 current posterior distribution.
300 The basic procedures for Bayesian optimization are summarized as follows:
301 (a) Start with initial point of hyperparameter setting to evaluate the objective function by running
302 machine learning algorithm experiment.
303 (b) Update the Gaussian process (surrogate model) to obtain a posterior distribution over the target
304 objective function.
305 (c) Pick the next point of hyperparameter setting for evaluation by maximizing the acquisition
306 function of expected improvement over the current best result.
307 (d) The procedure is repeated and the algorithm stops after a certain number of iterations (default
308 30 in MATLAB).
309 Bayesian optimization offers a natural framework for model-based global optimization of noisy,
310 expensive black-box machine learning algorithms [56]. Compared with grid search and randomized
311 search, Bayesian optimization is considerably more efficient as it can detect the optimal
312 hyperparameter combinations by analyzing the previously-tested values, and running surrogate model
313 is much cheaper than optimizing the objective function.
314

315 *3.4 Prediction performance evaluation*

316 Once the optimum hyperparameters have been obtained for the machine learning models, the
317 prediction performance can be evaluated on the hold-out testing dataset. To compare the performance

of gradient boosted trees and random forest, three statistical parameters are used to measure the prediction accuracy.

The root mean squared error (RMSE) for prediction is calculated as

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left[y_i - \hat{f}(x_i)\right]^2} \tag{6}$$

where m is the number of observations in the testing data set and $\hat{f}(x_i)$ gives the prediction for the ith observation.

The mean absolute percentage error (MAPE) is given by the following equation

$$\mathrm{MAPE} = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{y_i - \hat{f}(x_i)}{y_i}\right| \times 100\% \tag{7}$$

The R^2 statistic measures the proportion of variability in the response that can be explained by performing the regression

$$R^2 = 1 - \frac{\sum_{i=1}^{m}\left[y_i - \hat{f}(x_i)\right]^2}{\sum_{i=1}^{m}\left[y_i - \bar{y}\right]^2} \tag{8}$$

Generally, an R^2 statistic that is close to 1 indicates good performance for the regression.

4 Results and Discussion

4.1 Experimental results

For Random Forest, the observed concrete porosities are compared with predicted values for both

training and testing data sets in Fig. 5a and b. The optimal minimum size of terminal nodes is 1, which indicates that full-depth trees have been grown for Random Forest without the worry of overfitting. On the other hand, the best number of features to select at random for each split is 8, which means that all predictors should be selected at each split and Random Forest is reduced to Bagging in this case. To confirm that the number of trees grown in the regression forest is sufficient for achieving optimal prediction accuracy, the *out-of-bag* error is plotted against the number of trees in Fig. 6. It can be clearly observed that the OOB MSE is monotonically decreasing with number of trees and has settled down after growing 100 trees.

In the case of gradient boosted trees, the observed concrete porosities are compared with predicted values for both training and testing data sets in Fig. 7a and b. The best learning rate is obtained as 0.1404 and the corresponding number of learning cycles is 486. The optimal maximum number of splits is 7, which confirms that shallow trees are grown in the ensemble. This highlights one difference between boosting and bagging: in boosting, because the growth of a particular tree takes into account the other trees that have already been grown, smaller trees are typically sufficient. The optimal minimum number of observations on terminal nodes is 5.

For the purpose of comparing random forest and gradient boosting on the concrete porosity data set, the prediction performance statistics based on 100 simulations are summarized in Table 4. It can be clearly seen that gradient boosting outperforms random forest in terms of average and best prediction accuracy (RMSE, MAPE and R^2 statistics). Similar observations have been reported by Chou *et al.* [28], who have compared the performance of different data-mining techniques for predicting concrete compressive strength from mixture compositions. The experimental results show that the optimized

353 Multiple Additive Regression Tree (MART) based on boosting algorithm and decision stump provides
354 the best accuracy.
355 However, random forest produces very stable prediction accuracy, while gradient boosting may suffer
356 from high variance in the random simulations. This is because random forest contains only two
357 candidate hyperparameters that need to be tuned and the optimized parameters are quite similar across
358 multiple runs. Random forest is also known to be a robust machine learning algorithm that performs
359 well with its default settings [50]. Another advantage of random forest is that the *out-of-bag* (OOB)
360 error based tuning strategy is much faster than *k*-Fold Cross-Validation. Because the average number
361 of distinct observations in each bootstrapped sample is $0.632 \cdot n$ from Eq. (4), the OOB error will
362 roughly behave like 2-Fold CV error [35]. Hence, unlike Gradient Boosting, Random Forests can be
363 fitted in one sequence, with Cross-Validation being performed along the way.
364 One good thing with random forests is that the predictor importance can be estimated by permutation
365 of *out-of-bag* (OOB) predictor observations for the ensemble of trees [35,51]. As briefly summarized
366 in Fig. 8, the increase in OOB error as a result of randomly permutating the observations of *j*th predictor
367 variable in the OOB sample is computed for each tree and then averaged over all trees to indicate the
368 importance of variable *j* in the random forests. The influence of a variable in predicting the response
369 increases with the value of this importance measure. For the concrete porosity data set studied in the
370 present work, the variable importance plots are shown in Fig. 9. Curing days, binder content and *w/b*
371 ratio are considered to be the most critical factors in predicting concrete porosity. These data-driven
372 perspectives agree quite well with the published experimental works [13-15].
373 The partial dependent plot (PDP) shows the relationships between a predictor and the response of

regression in the trained model. The partial dependence on the selected predictor is defined as the averaged prediction obtained by marginalizing out the effect of the other variables [35]. Fig. 10 displays the single-variable partial dependence plots on the four most relevant predictors including curing days, binder content, *w/b* ratio and aggregate (CA/FA ratio). The vertical scales of the plots are the same, and give a visual comparison of the relative importance of the different variables. Porosity is generally monotonic decreasing with increasing curing days, but gradually reaching steady after 150 curing days. Porosity has a non-monotonic partial dependence on binder content. The PDP shows a large change near binder = 405.6 ~ 408.6 kg/m^3, which indicates many node splits are based on this critical value of binder. The partial dependence of porosity on *w/b* ratio is monotonic increasing. The influence of aggregate composition on porosity is on a relatively small scale. It is important to note that there may exist some interaction between different predictors, which can't be illustrated through the single-variable PDPs.

4.2 Sensitivity analysis

To further demonstrate the predicting capability of the proposed machine learning algorithm, sensitivity analysis has been performed on fly ash concrete and GGBS concrete. The predictions are made by employing the optimized gradient boosting tree. The artificial concrete mixture compositions are listed in Table 5. The varying parameters include fly ash or slag replacement portions and curing days.

As shown in Fig. 11 for fly ash concrete, the porosity increases with increasing fly ash portion in the

394 artificial specimens at early ages. The addition of fly ash in concrete can reduce water content
395 requirement for a given workability and provide refinement of pore structure. Due to the long-term
396 nature of the pozzolanic reaction, the beneficial effect of fly ash become evident after 100 days of
397 curing. The well-cured fly ash concrete has much lower porosity than OPC when other compositions
398 are the same. This observation is also consistent with the long-term strength developments in fly ash
399 concrete [57]. This phenomenon can also be demonstrated through the two-variable partial dependence
400 plot of the fitted model on joint values of fly ash content and curing days, as shown in Fig. 12.
401 The porosities of the artificial GGBS concrete specimens as a function of curing age are shown in Fig.
402 13. As the level of slag replacement increases, the early-age porosity increases. However, the long-
403 term pore structure of concrete continues to refine as a result of the cement hydration process, which
404 is contributed by the latent hydraulic reaction of GGBS. Given sufficient curing time, the concrete
405 porosity generally decreases as the slag replacement ratio increases. Similar results have been observed
406 by Choi *et al*. [58] in their experimental study of high-strength cement pastes incorporating high
407 volume GGBS. The two-variable partial dependence of the regression model on joint values of GGBS
408 content and curing days is shown in Fig. 14, which clearly demonstrate the strong interaction between
409 these two variables.
410

411 ## *4.3 Discussion*

412 In this study, the 75/25 random partition of the porosity dataset is utilized for both Gradient Boosting
413 Trees and Random Forests. The purpose is to compare the prediction performance of the two methods

414 on the same testing subset, which is independent of the training data. On the training data set, Random
415 Forests is optimized with *out-of-bag* error, while Gradient Boosting Trees is optimized with *k*-fold CV
416 error. The evolution of training error (on the 75% training subset) and testing error (on the 25% testing
417 subset) during the training process is plotted together with the *out-of-bag* error for Random Forests in
418 Fig. 15a. The testing MSE settles down together with the OOB MSE as the number of trees grows,
419 though the OOB error is relatively higher than the testing error. Similarly for Gradient Boosting Trees
420 in Fig. 15b, the testing error and the *k*-fold CV error have converged with the latter relatively higher
421 than the former. The simultaneous convergence of testing error and *out-of-bag* error or *k*-fold CV error
422 clearly demonstrates that the proposed optimization algorithm could produce the best combination of
423 hyperparameters.
424 Unlike Gradient Boosting Trees, Random Forests doesn't necessarily require dividing the whole data
425 set into 75/25 training/testing subsets because the 63/37 random partition is naturally embedded in the
426 bootstrap sampling procedure [37]. The *out-of-bag* error can be conveniently used as a measure of
427 prediction accuracy for the ensemble trees algorithm. However, there are some arguments among
428 practitioners and researchers on whether *out-of-bag* error is an unbiased estimate of the generalized
429 error for all datasets, though minimization of OOB MSE can always be used for tuning Random Forests
430 to find the optimal hyperparameters [50,53,59]. For the concrete porosity training dataset, the *out-of-*
431 *bag* prediction performance statistics based on the 100 simulations are summarized in Table 6. As
432 compared with the testing performance metrics for Random Forests in Table 4, the *out-of-bag* error
433 obtained from the training process is relatively higher than the prediction error on a new testing data
434 set. Similar observation has been reported by Breiman [47], who has concluded that the average *out-*

435 *of-bag* error is consistently higher than the average testing error for regression forests. Note that the
436 sample instances as well as the number of samples are different for the two performance metrics. Hence,
437 to ensure a fair comparison between Random Forests and Gradient Boosting Trees, testing error on the
438 same testing data set is being used in this research.
439 In this research, the number of trees grown (B) is set to be not tunable, but sufficiently large, for
440 Random Forests. Breiman [47] shows that the mean squared generalization error of random forest
441 regression converges as the number of aggregated trees increases. This may suggest that the larger
442 number of trees grown in the forests, the more accurate the ensemble prediction is. For the concrete
443 porosity dataset, the performance gains in terms of *out-of-bag* error by increasing the number of trees
444 from 50 to 1000 are illustrated in Fig. 16. The results are based on 100 simulations for each setting and
445 each run involves hyperparameter optimization. The variance of OOB MSE significantly reduces as B
446 increases. The average value of OOB MSE over 100 simulations also shows a monotonously
447 descending trend with the increasing B. However, there is only modest prediction improvement after
448 growing the first 300 trees. Therefore, the choice of $B = 300$ would be sufficient. Moreover, in the case
449 of a large dataset, the number of trees should be properly adjusted according to the computational cost
450 of Random Forests optimization.
451
452

453 5 Comparison with Chemo-Mechanical Model

454 Based on the chemo-mechanical modeling of Portland cement hydration and fly ash pozzolanic

455 reactions, Papadakis [23] has proposed a theoretical model to predict the chemical and volumetric
456 composition of fly ash concrete. The fly ash concrete porosity (ε) can be calculated as

$$\varepsilon = \varepsilon_{\text{air}} + W/\rho_w - \Delta\varepsilon_h - \Delta\varepsilon_p - \Delta\varepsilon_c \quad (9)$$

457 where ε_{air} denotes the volume fraction of entrapped or entrained air in concrete, *W* gives the water
458 content in concrete mixture (kg/m^3), $\rho_w = 1000\ \text{kg/m}^3$ is the density of water, and $\Delta\varepsilon_h$, $\Delta\varepsilon_p$, $\Delta\varepsilon_c$
459 represent the porosity reductions due to cement hydration, pozzolanic activity and carbonation,
460 respectively.
461 By assuming the full hydration of cement and the complete pozzolanic reactions of fly ash, the final
462 value of the porosity of a noncarbonated concrete can be predicted based on the physical and chemical
463 properties of cement and fly ash. Denote as $f_{i,c}$ and $f_{i,p}$ ($i = C, S, A, F, \bar{S}$) the weight fractions of
464 oxides CaO (C), SiO_2 (S), Al_2O_3 (A), Fe_2O_3 (F), SO_3 ($\bar{S}$) in cement and fly ash, respectively. The
465 glassy phases constitute the reactive portion of fly ash, particularly in low-calcium fly ash. The active
466 fractions of SiO_2 and Al_2O_3 in fly ash that contribute to the pozzolanic reactions are represented by
467 γ_S and γ_A (by weight). The cement and fly ash content in concrete mixture are given by *C* and *P*
468 (kg/m^3).
469 If the gypsum content is *higher* than that required for the full hydration of cement and the complete
470 pozzolanic reaction of fly ash alumina, i.e.

$$f_{\bar{S},c} > 0.785 f_{A,c} - 0.501 f_{F,c} + (0.785\gamma_A f_{A,p})(P/C) \quad (10)$$

471 The final value of the porosity of a noncarbonated concrete can be determined by the following
472 equation [23]

$$\varepsilon = \varepsilon_{\text{air}} + W/\rho_w - \left\{0.249(f_{C,c} - 0.7 f_{\bar{S},c}) + 0.191 f_{S,c} + 1.118 f_{A,c} - 0.357 f_{F,c}\right\} \times (C/1000) - (1.18\gamma_A f_{A,p})(P/1000) \quad (11)$$

where the maximum fly ash content ($P_{\max}$) that can participate in the pozzolanic reactions is specified as

$$P_{\max} = \frac{\left\{1.321(f_{C,c} - 0.7 f_{\bar{S},c}) - 1.851 f_{S,c} - 2.182 f_{A,c} - 1.392 f_{F,c}\right\} \times C}{1.851\gamma_S f_{S,p} + 2.182\gamma_A f_{A,p}} \quad (12)$$

If the gypsum content is *lower* than that required for the full hydration of cement and the complete pozzolanic reaction of fly ash alumina, i.e.

$$f_{\bar{S},c} < 0.785 f_{A,c} - 0.501 f_{F,c} + (0.785\gamma_A f_{A,p})(P/C) \quad (13)$$

The final value of the porosity of a noncarbonated concrete can be determined by the following equation [23]

$$\varepsilon = \varepsilon_{\text{air}} + W/\rho_w - (0.249 f_{C,c} - 0.1 f_{\bar{S},c} + 0.191 f_{S,c} + 1.059 f_{A,c} - 0.319 f_{F,c}) \times (C/1000) - (1.121\gamma_A f_{A,p})(P/1000) \quad (14)$$

where the maximum fly ash content ($P_{\max}$) that can participate in the pozzolanic reactions is specified as

$$P_{\max} = \frac{(1.321 f_{C,c} - 1.851 f_{S,c} - 2.907 f_{A,c} - 0.928 f_{F,c}) \times C}{1.851\gamma_S f_{S,p} + 2.907\gamma_A f_{A,p}} \quad (15)$$

Here, assumption has been made that the chemo-mechanical theory developed for pastes and mortars could also be applied to concrete.

The proposed case study is utilized to compare the conventional chemo-mechanical model with the machine learning method. To comply with the theoretical assumption of "complete" hydration and pozzolanic activities for Papadakis's model, we select 25 concrete specimens with at least 1-year hydration (365 curing days) from the assembled dataset. It is assumed that the entrained air in concrete

is negligible ($\varepsilon_{\text{air}} = 0$) and that there is no carbonation. The chemical compositions of the cement and fly ash used in the original experimental work [40] are shown in Table 7. Since no information has been provided about the glass phase content of the fly ash constituents, the active fractions ($\gamma_S = \gamma_A = 0.82$) reported by Papadakis [23] are adopted in this study. In literature, the experimentally measured glassy phase compositions of fly ash vary a lot at both bulk scale and oxide level [60-62]. For example, Cho *et al*. [61] have reported amorphous phases in the range of 68.1% to 77.6% with an average of 73.0% for bulk fly ashes, $\gamma_S \in (66.6\%, 77.9\%)$ for SiO_2 and $\gamma_A \in (57.4\%, 77.2\%)$ for Al_2O_3.

The comparison between the analytical prediction from Papadakis model and the empirical prediction from Random Forest is shown in Fig. 17. There are 18 instances chosen from the training data set, for which *out-of-bag* predictions are used. This is to reduce the possible overfitting bias in the error estimation for the training data. It can be clearly observed that Random Forest easily outperforms the conventional chemo-mechanical model in terms of RMSE and MAPE. This suggests that the proposed data-driven approach could be applied for practical estimation of concrete porosity.

6 Conclusions

This paper applies ensemble trees to predict the porosity of high-performance concrete containing supplementary cementitious materials. A reliable database for concrete porosity, featuring 74 unique concrete mixtures, is assembled from published literature. Compositions of concrete are characterized

by 8 features including *w/b* ratio, binder content, fly ash, GGBS, superplasticizer, coarse/fine aggregate ratio, curing condition and curing days. The full dataset is randomly divided into 75% training dataset and 25% testing dataset through stratified sampling.

The complexity (depth) of the individual trees grown in the ensemble can be regulated via limiting the maximum number of splits and/or the minimum size of terminal nodes. For boosted trees, the number of learning cycles can be controlled by the learning rate. In the case of random forest, training samples can be bootstrapped from the entire training data set and the predictors can be selected randomly at each split. The complexity level of boosted regression trees is tuned using *k*-Fold Cross-Validation, while the hyperparameters for random forest are optimized by minimizing *out-of-bag* (OOB) error. Bayesian optimization has been employed to search for the best combination of hyperparameters for regression tree ensembles.

Experimental tests show that ensemble trees can accurately predict the porosity of concrete from mixture compositions. Gradient boosting trees generally outperforms random forests in terms of prediction accuracy. Shallow trees are typically grown for gradient boosting, while full-depth trees work well for random forests. The OOB error based tuning strategy for random forest is found to be much faster than *k*-Fold Cross-Validation. The variable importance plot shows that curing days, binder content and *w/b* ratio are the most important predictors for concrete porosity. Sensitivity analysis further demonstrates the long-term beneficial effects of fly ash and GGBS in reducing concrete porosity.

Compared with conventional statistical regression or classical chemo-mechanical hydration model, the proposed ensemble learning algorithm is able to take into consideration the complex concrete

compositions and achieve high prediction accuracy. Potential applications of this method may include optimizing concrete mixture compositions for performance-based concrete structure design and for reducing environmental impact of concrete [63-65]. Future work should continue to build a reliable and balanced database to train the ensemble trees model. The prediction performance can be further improved by combining different machine learning algorithms. Moreover, for the practical implementation of Machine Learning algorithm in concrete mix design, further study is needed to explore the effect of prediction error on the safety factor required.

Data Availability

The training data and machine learning codes for this study are available in GitHub (https://github.com/againstlaw/Porosity).

References

[1] V.G. Papadakis, C.G. Vayenas, M.N. Fardis, Physical and chemical characteristics affecting the durability of concrete, ACI Materials Journal 88(2) (1991) pp. 186 – 196. https://doi.org/10.14359/1993.

[2] P. Linares-Alemparte, C. Andrade, D. Baza, Porosity and electrical resistivity-based empirical calculation of the oxygen diffusion coefficient in concrete, Construction and Building Materials 198(20) (2019) pp. 710 – 717. https://doi.org/10.1016/j.conbuildmat.2018.11.269.

[3] N. Shafiq, J.G. Cabrera, Effects of initial curing condition on the fluid transport properties in OPC and fly ash blended cement concrete, Cement and Concrete Composites 26(4) (2004) pp. 381 – 387. https://doi.org/10.1016/S0958-9465(03)00033-7.

[4] H.W. Song, S.J. Kwon, Permeability characteristics of carbonated concrete considering capillary pore structure, Cement and Concrete Research 37(6) (2007) pp. 909 – 915. https://doi.org/10.1016/j.cemconres.2007.03.011.

[5] S. Lammertign, N. De Belie, Porosity, gas permeability, carbonation and their interaction in high-volume fly ash concrete, Magazine of Concrete Research 60(7) (2008) pp. 535 – 545. https://doi.org/10.1680/macr.2008.60.7.535.

[6] M.R. Nokken, R.D. Hooton, Using pore parameters to estimate permeability or conductivity of concrete, Materials and Structures 41(1) (2008) pp. 186 – 196. https://doi.org/10.1617/s11527-006-9212-y.

[7] Q.T. Phung, N. Maes, G. De Schutter, D. Jacques, G. Ye, Determination of water permeability of cementitious materials using a controlled constant flow method, Construction and Building Materials 47 (2013) pp. 1488 – 1496. https://doi.org/10.1016/j.conbuildmat.2013.06.074.

[8] C.S. Poon, S.C. Kou, L. Lam, Compressive strength, chloride diffusivity and pore structure of high performance metakaolin and silica fume concrete, Construction and Building Materials 20(10) (2006) pp. 858 – 865. https://doi.org/10.1016/j.conbuildmat.2005.07.001.

[9] T. Simčič, S. Pejovnik, G. De Schutter, V.B. Bosiljkov, Chloride ion penetration into fly ash modified concrete during wetting-drying cycles, Construction and Building Materials 93(15) (2015) pp. 1216 – 1223. https://doi.org/10.1016/j.conbuildmat.2015.04.033.

[10] P.A. Claisse, J.G. Cabrera, D.N. Hunt, Measurement of porosity as a predictor of the durability performance of concrete with and without condensed silica fume, Advances in Cement Research 13(4) (2001) pp. 165 – 174. https://doi.org/10.1680/adcr.2001.13.4.165.

[11] L.Z. Xiao, Z.J. Li, Early-age hydration of fresh concrete monitored by non-contact electrical resistivity measurement, Cement and Concrete Research 38(3) (2008) pp. 312 – 319. https://doi.org/10.1016/j.cemconres.2007.09.027.

[12] L. Bertolini, B. Elsener, P. Pedeferri, E. Redaelli, R. Polder, Corrosion of Steel in Concrete: Prevention, Diagnosis, Repair, second ed., WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2013. Print ISBN: 9783527308002.

[13] T.C. Hansen, Physical structure of hardened cement paste: A classical approach, Materials and Structures 19 (1986) pp. 423 – 436. https://doi.org/10.1007/BF02472146.

[14] L. Basheer, P.A.M. Basheer, A.E. Long, Influence of coarse aggregate on the permeation durability and the microstructure characteristics of ordinary Portland cement concrete, Construction and Building Materials 19(9) (2005) pp. 682 – 690. https://doi.org/10.1016/j.conbuildmat.2005.02.022.

[15] S. Ahmad, A.K. Azad, K.F. Loughlin, Effect of the key mixture parameters on tortuosity and permeability of concrete, Journal of Advanced Concrete Technology 10(3) (2012) pp. 86 – 94. https://doi.org/10.3151/jact.10.86.

[16] V.G. Papadakis, Effect of supplementary cementing materials on concrete resistance against carbonation and chloride ingress, Cement and Concrete Research 30(2) (2000) pp. 291 – 299. https://doi.org/10.1016/S0008-8846(99)00249-5.

[17] ACI Committee 233, Guide to the use of slag cement in concrete and mortar, ACI 233R-17, American Concrete Institute, Farmington Hills, MI, 2017. ISBN: 9781945487804.

[18] ACI Committee 232, Report on the use of fly ash in concrete, ACI 232.2R-18, American Concrete Institute, Farmington Hills, MI, 2018. ISBN: 9781641950060.

[19] M.D.A. Thomas, P.B. Bamforth, Modeling chloride diffusion in concrete: effect of fly ash and slag, Cement and Concrete Research 29(4) (1999) pp. 487 – 495. https://doi.org/10.1016/S0008-8846(98)00192-6.

598 [20] H.W. Song, V. Saraswathy, Studies on the corrosion resistance of reinforced steel in concrete with
599 ground granulated blast-furnace slag – An overview, Journal of Hazardous Materials 138(2) (2006)
600 pp. 226 – 233. https://doi.org/10.1016/j.jhazmat.2006.07.022.

601 [21] M.D.A. Thomas, J.D. Matthews, The permeability of fly ash concrete, Materials and Structures
602 25 (1992) pp. 388 – 396. https://doi.org/10.1007/BF02472254.

603 [22] K.E. Hassan, J.G. Cabrera, R.S. Maliehe, The effect of mineral admixtures on the properties of
604 high-performance concrete, Cement and Concrete Composites 22(4) (2000) pp. 267 – 271.
605 https://doi.org/10.1016/S0958-9465(00)00031-7.

606 [23] V.G. Papadakis, Effect of fly ash on Portland cement systems Part I: Low-calcium fly ash, Cement
607 and Concrete Research 29(11) (1999) pp. 1727 – 1736. https://doi.org/10.1016/S0008-
608 8846(99)00153-2.

609 [24] V.G. Papadakis, Effect of fly ash on Portland cement systems Part II: High-calcium fly ash,
610 Cement and Concrete Research 30(10) (2000) pp. 1647 – 1654. https://doi.org/10.1016/S0008-
611 8846(00)00388-4.

612 [25] M.I. Khan, Permeation of high performance concrete, ASCE Journal of Materials in Civil
613 Engineering 15(1) (2003) pp. 84 – 92. https://doi.org/10.1061/(ASCE)0899-1561(2003)15:1(84).

614 [26] I.C. Yeh, Modeling of strength of high-performance concrete using artificial neural networks,
615 Cement and Concrete Research 28(12) (1998) pp. 1797 – 1808. https://doi.org/10.1016/S0008-
616 8846(98)00165-3.

617 [27] I.C. Yeh, Modeling slump flow of concrete using second-order regressions and artificial neural
618 networks, Cement and Concrete Composites 29(6) (2007) pp. 474 – 480.
619 https://doi.org/10.1016/j.cemconcomp.2007.02.001.

620 [28] J.S. Chou, C.K. Chiu, M. Farfoura, I. Al-Taharwa, Optimizing the prediction accuracy of concrete
621 compressive strength based on a comparison of data-mining techniques, ASCE Journal of
622 Computing in Civil Engineering 25(3) (2011) pp. 242 – 253.
623 https://doi.org/10.1061/(ASCE)CP.1943-5487.0000088.

624 [29] R. Cook, J. Lapeyre, H. Ma, A. Kumar, 2019. Prediction of compressive strength of concrete:
625 critical comparison of performance of a hybrid machine learning model with standalone models.
626 ASCE Journal of Materials in Civil Engineering 31, 04019255.
627 https://doi.org/10.1061/(ASCE)MT.1943-5533.0002902.

628 [30] H.W. Song, S.J. Kwon, Evaluation of chloride penetration in high performance concrete using
629 neural network algorithm and micro pore structure, Cement and Concrete Research 39(9) (2009)
630 pp. 814 – 824. https://doi.org/10.1016/j.cemconres.2009.05.013.

631 [31] M.I. Khan, Predicting properties of high performance concrete containing composite cementitious
632 materials using Artificial Neural Networks, Automation in Construction 22 (2012) pp. 516 – 524.
633 https://doi.org/10.1016/j.autcon.2011.11.011.

634 [32] W. Dong, Y. Huang, B. Lehane, G. Ma, 2020. XGBoost algorithm-based prediction of concrete
635 electrical resistivity for structural health monitoring. Automation in Construction 114, 103155.
636 https://doi.org/10.1016/j.autcon.2020.103155.

637 [33] W.Z. Taffese, E. Sistonen, Machine learning for durability and service-life assessment of
638 reinforced concrete structures: Recent advances and future directions, Automation in Construction
639 77 (2017) pp. 1 – 14. https://doi.org/10.1016/j.autcon.2017.01.016.

640 [34] B. Boukhatem, R. Rebouh, A. Zidol, M. Chekired, A. Tagnit-Hamou, An intelligent hybrid system
641 for predicting the tortuosity of the pore system of fly ash concrete, Construction and Building
642 Materials 205 (2019) pp. 274 – 284. https://doi.org/10.1016/j.conbuildmat.2019.02.005.

643 [35] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining,
644 Inference, and Prediction, second ed., Springer, New York, 2009. https://doi.org/10.1007/978-0-
645 387-84858-7.

646 [36] H.I. Erdal, Two-level and hybrid ensembles of decision trees for high performance concrete
647 compressive strength prediction, Engineering Applications of Artificial Intelligence 26(7) (2013)
648 pp. 1689 – 1697. https://doi.org/10.1016/j.engappai.2013.03.014.

649 [37] W.Z. Taffese, E. Sistonen, Significance of chloride penetration controlling parameters in concrete:
650 Ensemble methods, Construction and Building Materials 139(15) (2017) pp. 9 – 23.
651 https://doi.org/10.1016/j.conbuildmat.2017.02.014.

652 [38] V. Nilsen, L.T. Pham, M. Hibbard, A. Klager, S.M. Cramer, D. Morgan, Prediction of concrete
653 coefficient of thermal expansion and other properties using machine learning, Construction and
654 Building Materials 220(30) (2019) pp. 587 – 595.
655 https://doi.org/10.1016/j.conbuildmat.2019.05.006.

656 [39] A.S. Cheng, T. Yen, Y.W. Liu, Y.N. Sheen, Relation between porosity and compressive strength
657 of slag concrete, ASCE Structures Congress 2008. https://doi.org/10.1061/41016(314)310.

[40] O. Al-Amoudi, M. Maslehuddin, I. Asi, Performance and correlation of the properties of fly ash cement concrete, Cem. Concr. Aggregates 18 (1996) pp. 71 – 77. https://doi.org/10.1520/CCA10153J.

[41] N. Shafiq, M.F. Nuruddin, I. Kamaruddin, Comparison of engineering and durability properties of fly ash blended cement concrete made in UK and Malaysia, Advances in Applied Ceramics 106(6) (2007) pp. 314 – 318. https://doi.org/10.1179/174367607X228089.

[42] P. Van den Heede, E. Gruyaert, N. De Belie, Transport properties of high-volume fly ash concrete: Capillary water sorption, water sorption under vacuum and gas permeability, Cement and Concrete Composites 32(10) (2010) pp. 749 – 756. https://doi.org/10.1016/j.cemconcomp.2010.08.006.

[43] A. Younsi, P. Turcry, E. Rozière, A. Aït-Mokhtar, A. Loukili, Performance-based design and carbonation of concrete with high fly ash content, Cement and Concrete Composites 33(10) (2011) pp. 993 – 1000. https://doi.org/10.1016/j.cemconcomp.2011.07.005.

[44] S. Ahmad, A.K. Azad, An exploratory study on correlating the permeability of concrete with its porosity and tortuosity, Advances in Cement Research 25(5) (2013) pp. 288 – 294. https://doi.org/10.1680/adcr.12.00052.

[45] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, Springer, New York, 2013. https://doi.org/10.1007/978-1-4614-7138-7.

[46] L. Breiman, Bagging predictors, Machine Learning 24 (1996) pp. 123 – 140. https://doi.org/10.1007/BF00058655.

[47] L. Breiman, Random forests, Machine Learning 45 (2001) pp. 5 – 23. https://doi.org/10.1023/A:1010933404324.

[48] J.H. Friedman, Greedy function approximation: A gradient boosting machine, The Annals of Statistics 29(5) (2001) pp. 1189 – 1232. https://doi.org/10.1214/aos/1013203451.

[49] P. Probst, A.-L. Boulesteix, To tune or not to tune the number of trees in random forest, Journal of Machine Learning Research 18(181) (2018) pp. 1 – 18. https://jmlr.org/papers/v18/17-269.html.

[50] P. Probst, M.N. Wright, A.-L. Boulesteix, 2019. Hyperparameters and tuning strategies for random forest. WIREs Data Mining and Knowledge Discovery 9, e1301. https://doi.org/10.1002/widm.1301.

[51] Mathworks, Statistics and machine learning toolbox: User's guide (R2020b). Retrieved October 31, 2020 from https://www.mathworks.com/help/stats.

[52] M.R. Segal, Machine learning benchmarks and random forest regression, Technical report, eScholarship Repository, University of California, 2004. https://escholarship.org/uc/item/35x3v9t4.

[53] B.A. Goldstein, E.C. Polley, F.B.S. Briggs, Random forests for genetic association studies, Statistical Applications in Genetics and Molecular Biology 10(1) (2011) Article 32. https://doi.org/10.2202/1544-6115.1691.

[54] L. Breiman, Out-of-bag estimation, Technical report, University of California, 1996. https://www.stat.berkeley.edu/pub/users/breiman/OOBestimation.pdf.

[55] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems, Volume 2 (2012) pp. 2951 – 2959. https://dl.acm.org/doi/abs/10.5555/2999325.2999464.

[56] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, R. Adams, Scalable Bayesian optimization using deep neural networks, ICML'15: Proceedings of the 32nd International Conference on Machine learning, Volume 37 (2015) pp. 2171 – 2180. https://dl.acm.org/doi/abs/10.5555/3045118.3045349.

[57] M.D.A. Thomas, Optimizing the use of fly ash in concrete, Portland Cement Association, Skokie, IL, 2007. Retrieved March 19, 2021 from https://www.cement.org/.

[58] Y.C. Choi, J. Kim, S. Choi, Mercury intrusion porosimetry characterization of micropore structures of high-strength cement pastes incorporating high volume ground granulated blast-furnace slag, Construction and Building Materials 137(15) (2017) pp. 96 – 103. https://doi.org/10.1016/j.conbuildmat.2017.01.076.

[59] S. Janitza and R. Hornung, 2018. On the overestimation of random forest's out-of-bag error, Plos One 13(8), e0201904. https://doi.org/10.1371/journal.pone.0201904.

[60] R.T. Chaney, P. Stutzman, M.C.G. Juenger, D.W. Fowler, Comprehensive phase characterization of crystalline and amorphous phases of a Class F fly ash, Cement and Concrete Research 40(1) (2010) pp. 146 – 156. https://doi.org/10.1016/j.cemconres.2009.08.029.

[61] Y.K. Cho, S.H. Jung, Y.C. Choi, Effects of chemical composition of fly ash on compressive strength of fly ash cement mortar, Construction and Building Materials 204 (2019) pp. 255 – 264. https://doi.org/10.1016/j.conbuildmat.2019.01.208.

719 [62] D. Glosser, P. Suraneni, O.B. Isgor, W.J. Weiss, 2021. Using glass content to determine the
720 reactivity of fly ash for thermodynamic calculations. Cement and Concrete Research 115, 103849.
721 https://doi.org/10.1016/j.cemconcomp.2020.103849.

722 [63] B.L. Damineli, F.M. Kemeid, P.S. Aguiar, V.M. John, Measuring the eco-efficiency of cement
723 use, Cement and Concrete Composites 32(8) (2010) pp. 555 – 562.
724 https://doi.org/10.1016/j.cemconcomp.2010.07.009.

725 [64] H. Van Damme, Concrete material science: Past, present, and future innovations, Cement and
726 Concrete Research 112 (2018) pp. 5 – 24. https://doi.org/10.1016/j.cemconres.2018.05.002.

727 [65] G. Habert, S.A. Miller, V.M. John, J.L. Provis, A. Favier, A. Horvath, K.L. Scrivener,
728 Environmental impacts and decarbonization strategies in the cement and concrete industries,
729 Nature Reviews Earth & Environment 1 (2020) pp. 559 – 573. https://doi.org/10.1038/s43017-020-
730 0093-3.

731

List of Tables

List of Figures

Table 1 Selected experimental data

Reference	Mix ID	*w/b*	Binder (kg/m^3)	Fly ash (%)	GGBS (%)	SP (%)	Aggregate (CA/FA)	Curing condition	Curing days	Porosity (%)	Training
Ahmad and Azad (2013)	1	0.4	300	0	0	0	1.6	air	28	9.58	True
	2	0.4	350	0	0	0	1.6	air	28	11.08	True
	3	0.4	400	0	0	0	1.6	air	28	11.27	True
	16	0.6	300	0	0	0	1.8	air	28	11.07	False
	17	0.6	350	0	0	0	1.8	air	28	11.63	True
	18	0.6	400	0	0	0	1.8	air	28	12.64	True
Shafiq *et al*. (2007)	UK0	0.55	325	0	0	0	1.5	water	3	11.12	False
	UK30	0.49	325	30	0	0	1.5	water	3	10.80	True
	UK40	0.48	325	40	0	0	1.5	water	3	10.79	True
	MY0	0.56	325	0	0	0	1.5	water	180	9.77	False
	MY30	0.525	325	30	0	0	1.5	water	180	8.38	True
	MY40	0.515	325	40	0	0	1.5	water	180	8.27	False
Van den Heede *et al*. (2010)	F0-1	0.5	350	0	0	0	1.32	air	28	15.40	False
	F0-2	0.4	400	0	0	0.228	1.67	air	28	13.42	True
	F35	0.4	400	35	0	0.228	1.67	air	28	15.23	False
	F35	0.4	400	35	0	0.228	1.67	air	91	13.54	True
	F50-4	0.4	400	50	0	0.3	1.67	air	91	12.90	True
	F67	0.4	400	67	0	0.228	1.67	air	91	16.50	False
Al-Amoudi *et al*. (1996)	–	0.35	350	0	0	0	2	air	28	10.7	False
	–	0.35	350	20	0	0	2	air	28	11.3	True
	–	0.35	350	40	0	0	2	air	28	12.1	True
	–	0.55	350	0	0	0	2	air	365	11.8	True
	–	0.55	350	20	0	0	2	air	365	10.2	False
	–	0.55	350	40	0	0	2	air	365	11.3	True
Younsi *et al*. (2011)	RefI	0.60	301	0	0	0	1.27	air	28	16	True
	FA30	0.53	341	30	0	0.51	1.27	air	28	15.9	True
	RefI	0.60	301	0	0	0	1.27	water	28	13.8	True
	FA30	0.53	341	30	0	0.51	1.27	water	28	14.4	True
Cheng *et al*. (2008)	WB35	0.35	591	0	0	0.7	1.7	air	7	6.40	True
	WB35-20	0.35	591	0	20	0.7	1.7	air	7	6.39	True
	WB35-40	0.35	591	0	40	0.7	1.7	air	7	6.84	True
	WB70	0.7	296	0	0	0	1.2	air	56	7.66	True
	WB70-20	0.7	296	0	20	0	1.2	air	56	7.13	True
	WB70-40	0.7	295	0	40	0	1.2	air	56	5.72	True

Table 2 Statistical summary of continuous variables in the whole dataset

Attributes	Minimum	Maximum	Mean	Standard deviation
w/b ratio	0.35	0.7	0.48	0.10
Binder content (kg/m^3)	295	591	370	74
Fly ash content (%)	0	67	15	17.6
GGBS content (%)	0	40	4.4	10.6
Superplasticizer (%)	0	1.58	0.1	0.24
Aggregate (CA/FA)	1.2	2	1.7	0.3
Curing days	1	365	89	109
Porosity (%)	2.39	18.05	10.36	2.88

Table 3 Hyperparameters for regression trees ensemble

Hyperparameters	Random Forest	Gradient Boosting
Number of trees	300	[10, 500]
Maximum number of splits	179	[1, 20]
Minimum size of terminal nodes	[1, 20]	[1, 90]
Number of features to select for each split	[1, 8]	8
Learning rate	N/A	[0.001, 1]

Table 4 Summary of prediction performance statistics over 100 simulations

Statistics	Random Forest			Gradient Boosting		
	RMSE	MAPE	R^2	RMSE	MAPE	R^2
Minimum	0.86	5.50%	0.894	0.52	3.36%	0.872
Maximum	0.93	6.10%	0.909	1.02	7.12%	0.967
Mean	0.89	5.77%	0.901	0.68	4.66%	0.942

Table 5 Concrete compositions for sensitivity test

Attributes	Fly ash concrete	GGBS concrete
w/b ratio	0.4	0.4
Binder content (kg/m^3)	400	400
Fly ash content (%)	0, 10, 20, 30, 40	0
GGBS content (%)	0	0, 10, 20, 30, 40
Superplasticizer (%)	0	0
Aggregate (CA/FA)	2	2
Curing days	7, 28, 90, 180, 270	3, 7, 28, 56
Curing condition	air	air

Table 6 Random Forests *out-of-bag* prediction performance statistics over 100 simulations

	RMSE	MAPE	R^2
Minimum	1.19	7.66%	0.807
Maximum	1.26	8.34%	0.830
Mean	1.22	7.90%	0.822

Table 7 Chemical characteristics of cement and low-calcium fly ash [40]

wt. %	SiO_2	Al_2O_3	Fe_2O_3	CaO	SO_3	MgO	Na_2O	K_2O	L.O.I.
Cement	22.3	3.6	3.6	64.6	1.9	2.1	0.1	0.2	1.2
Fly ash	60.5	23.0	7.5	2.1	0.3	1.0	-	-	1.4

Figure 1

Click here to access/download;Figure;Fig 1.pptx

Click here to view linked References

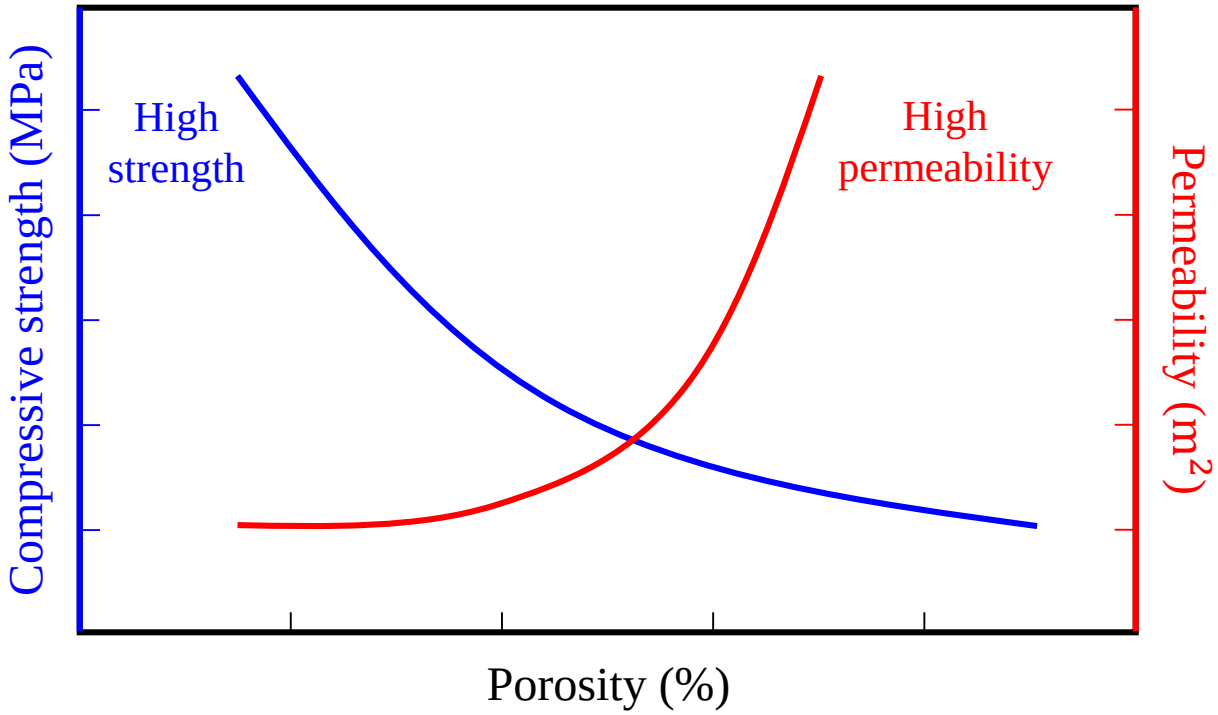


Figure 2a

Click here to access/download;Figure;Fig 2a.tif

Click here to view linked References

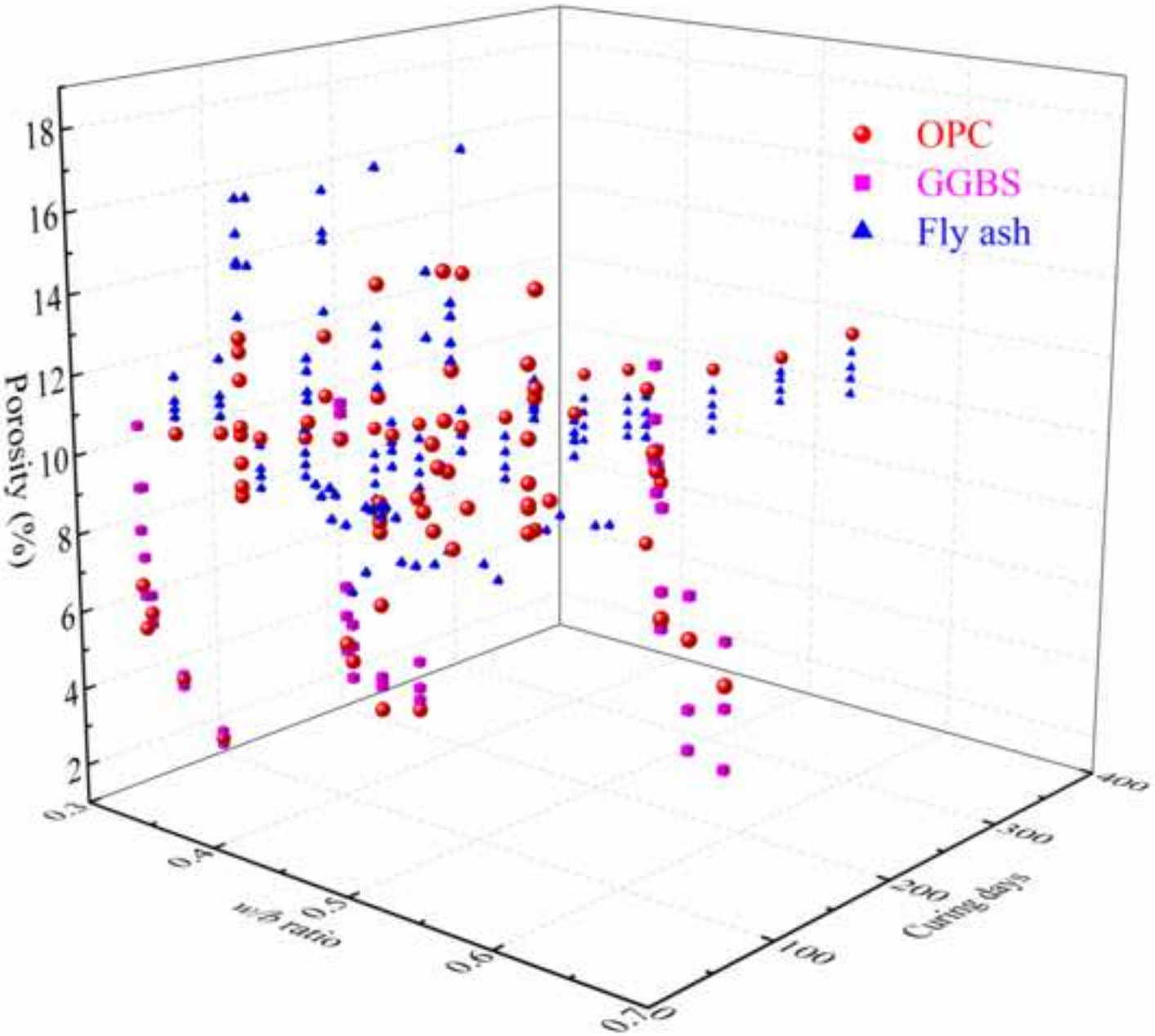


Figure 2b

Click here to access/download;Figure;Fig 2b.tif

Click here to view linked References

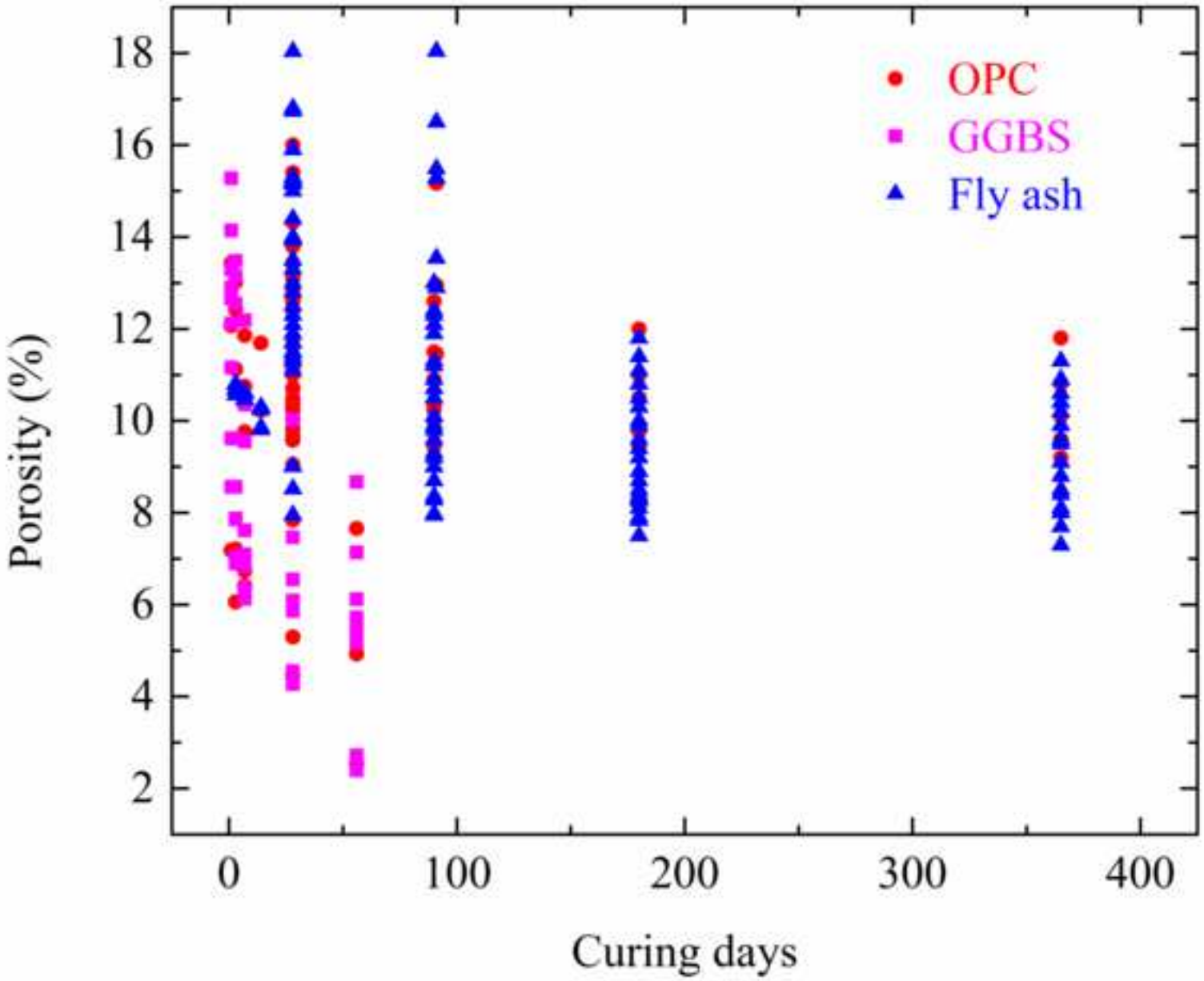


Figure 2c

Click here to access/download;Figure;Fig 2c.tif

Click here to view linked References

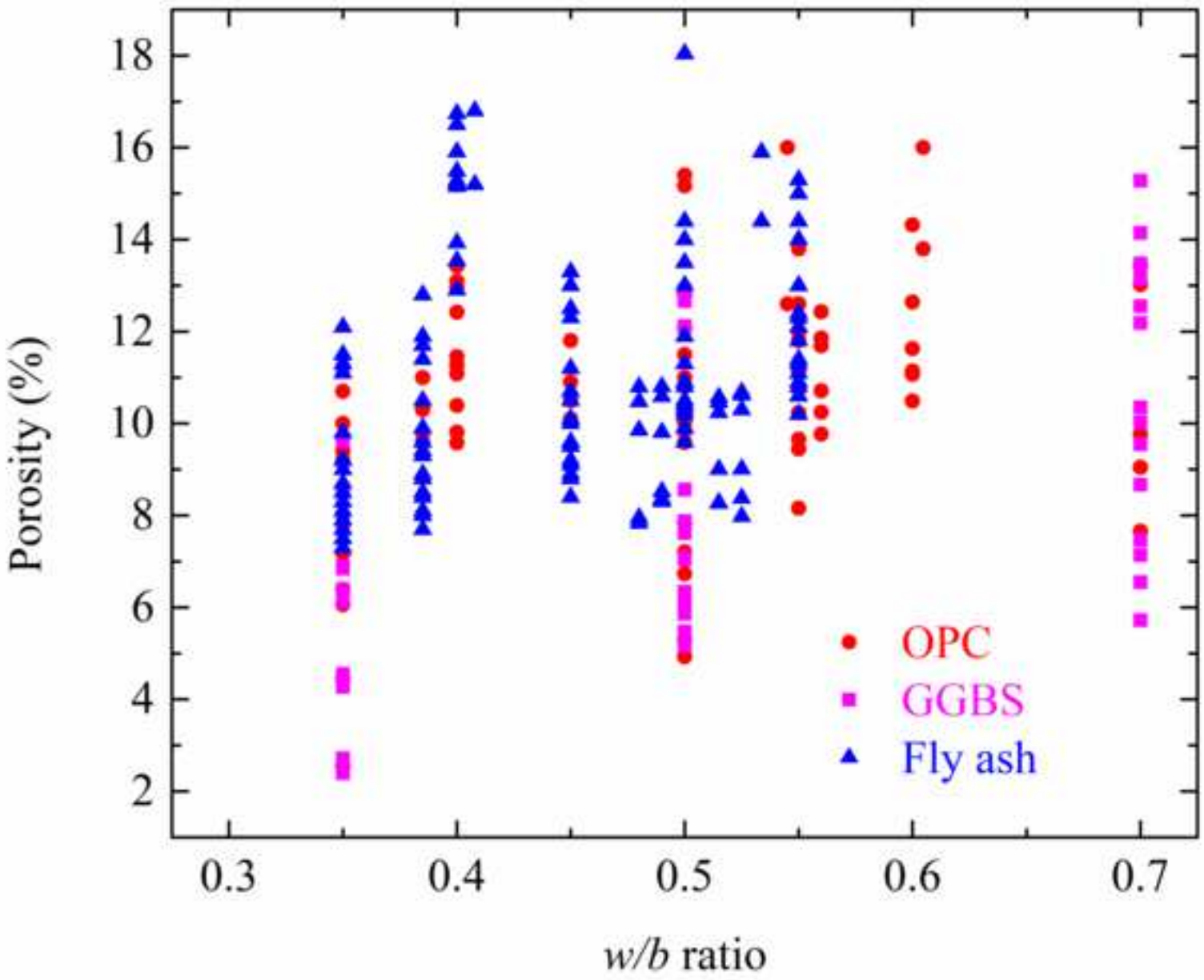
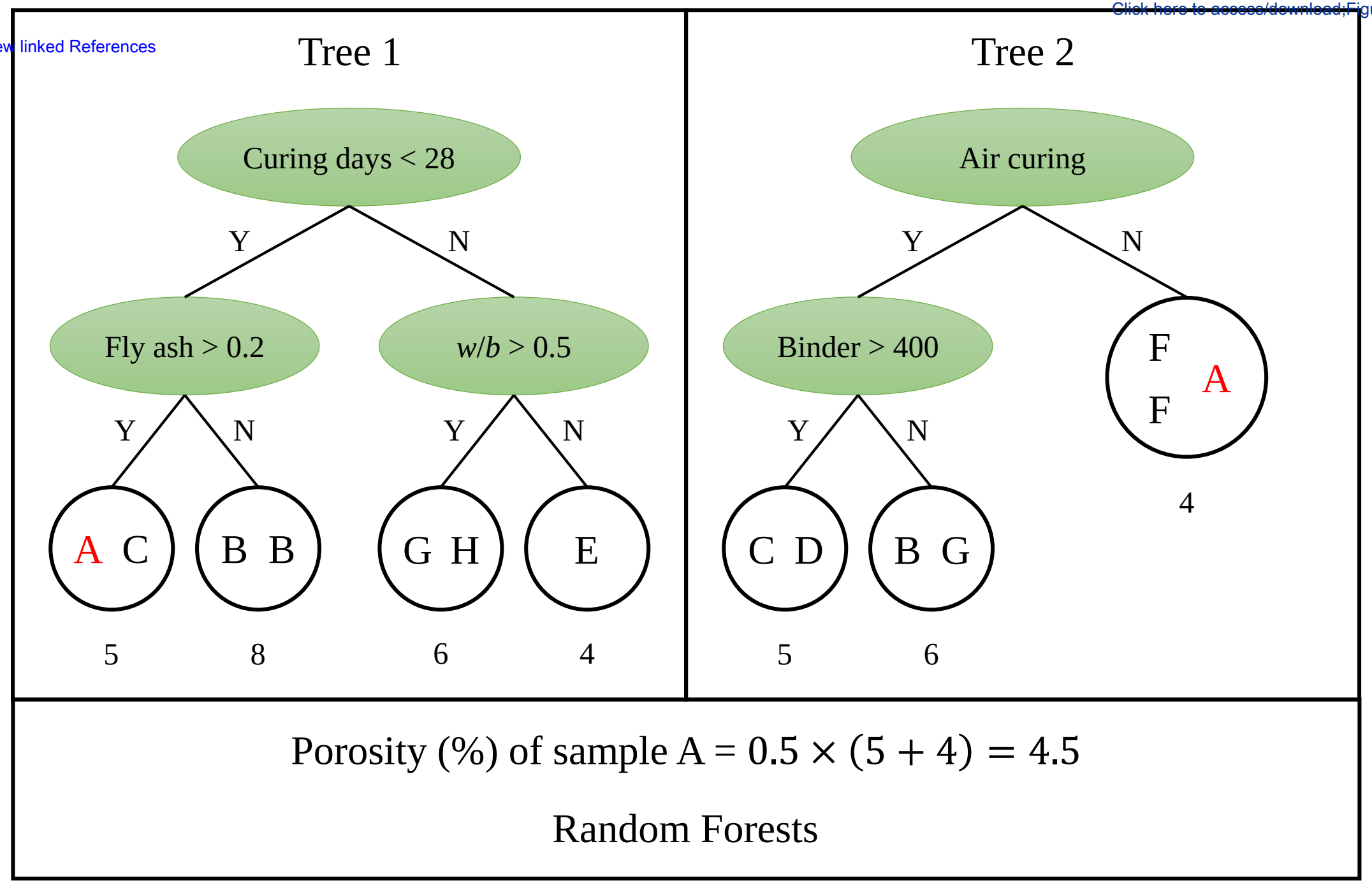


Figure 3



Algorithm: Least-squares Boosting for regression trees

Set $F_0(x) = 0$

For b = 1 **to** B **do**:

$r^b = y - F_{b-1}(x)$

Fit a regression tree $\hat{f}^b(x)$ to the training data (x, r^b)

$F_b(x) = F_{b-1}(x) + \lambda \hat{f}^b(x)$

End

Output: $\hat{f}(x) = F_B(x)$

Figure 5a

Click here to access/download;Figure;Fig 5a.tif

Click here to view linked References

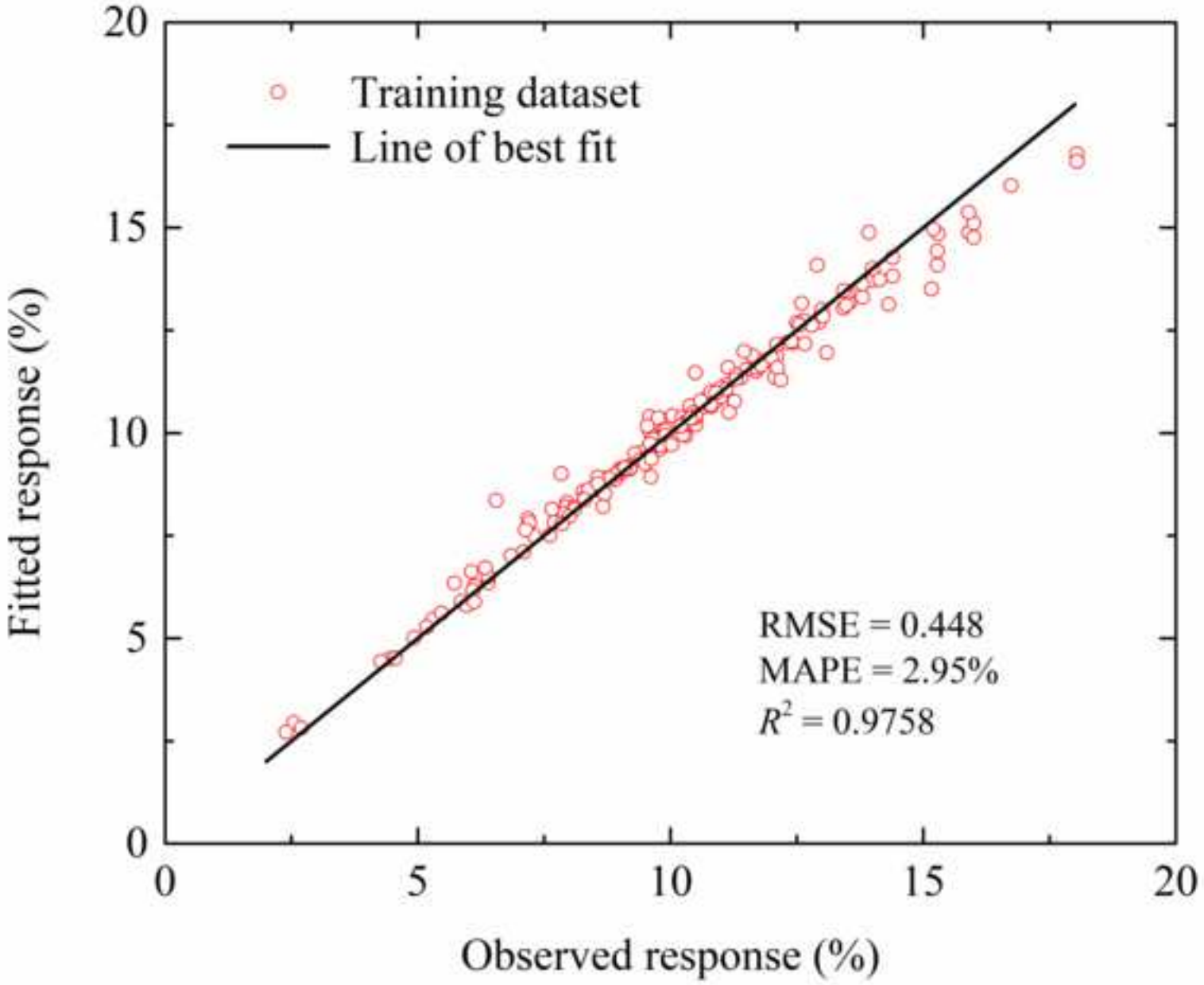


Figure 5b

Click here to access/download;Figure;Fig 5b.tif

Click here to view linked References

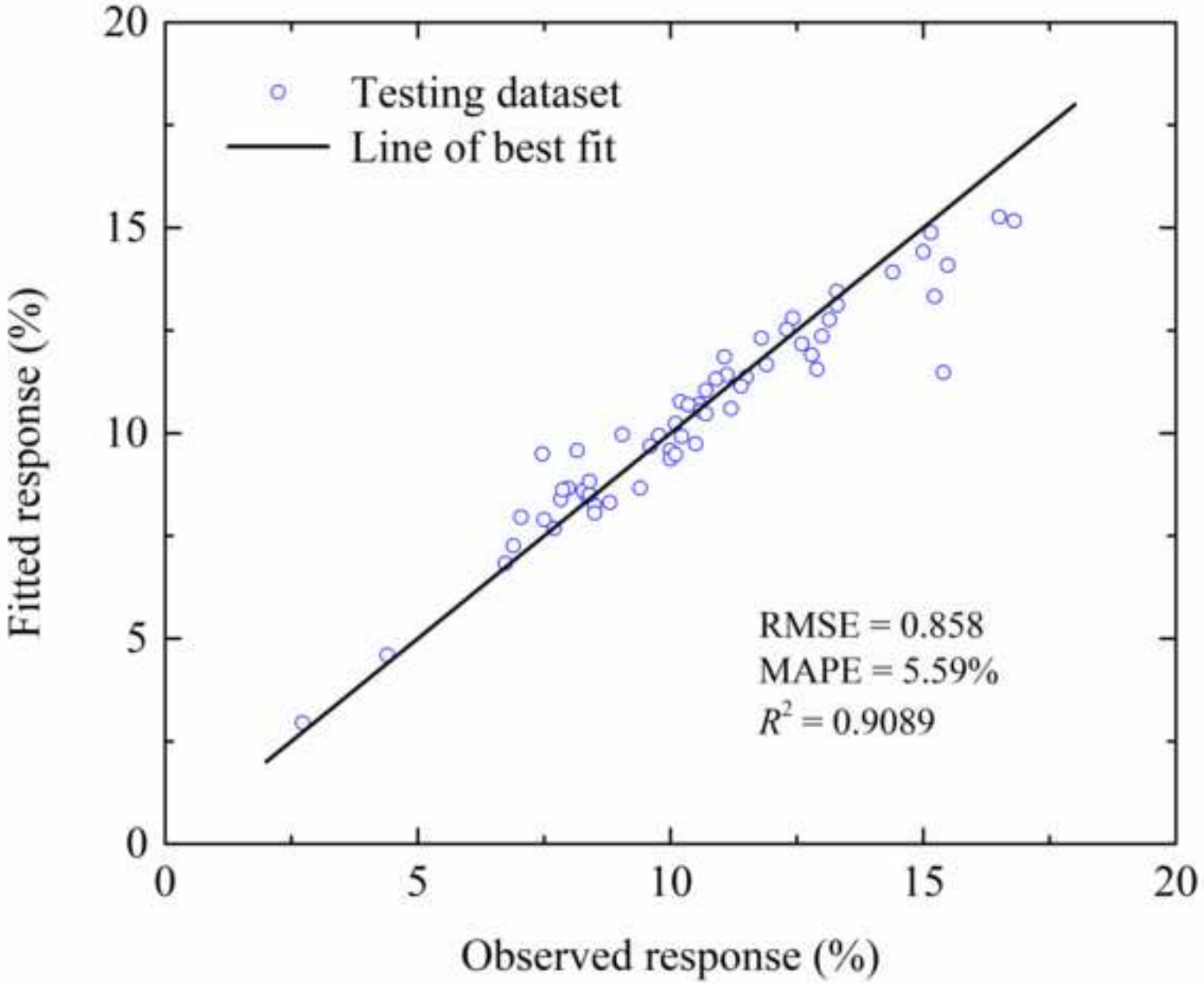


Figure 6

Click here to access/download;Figure;Fig 6.tif

Click here to view linked References

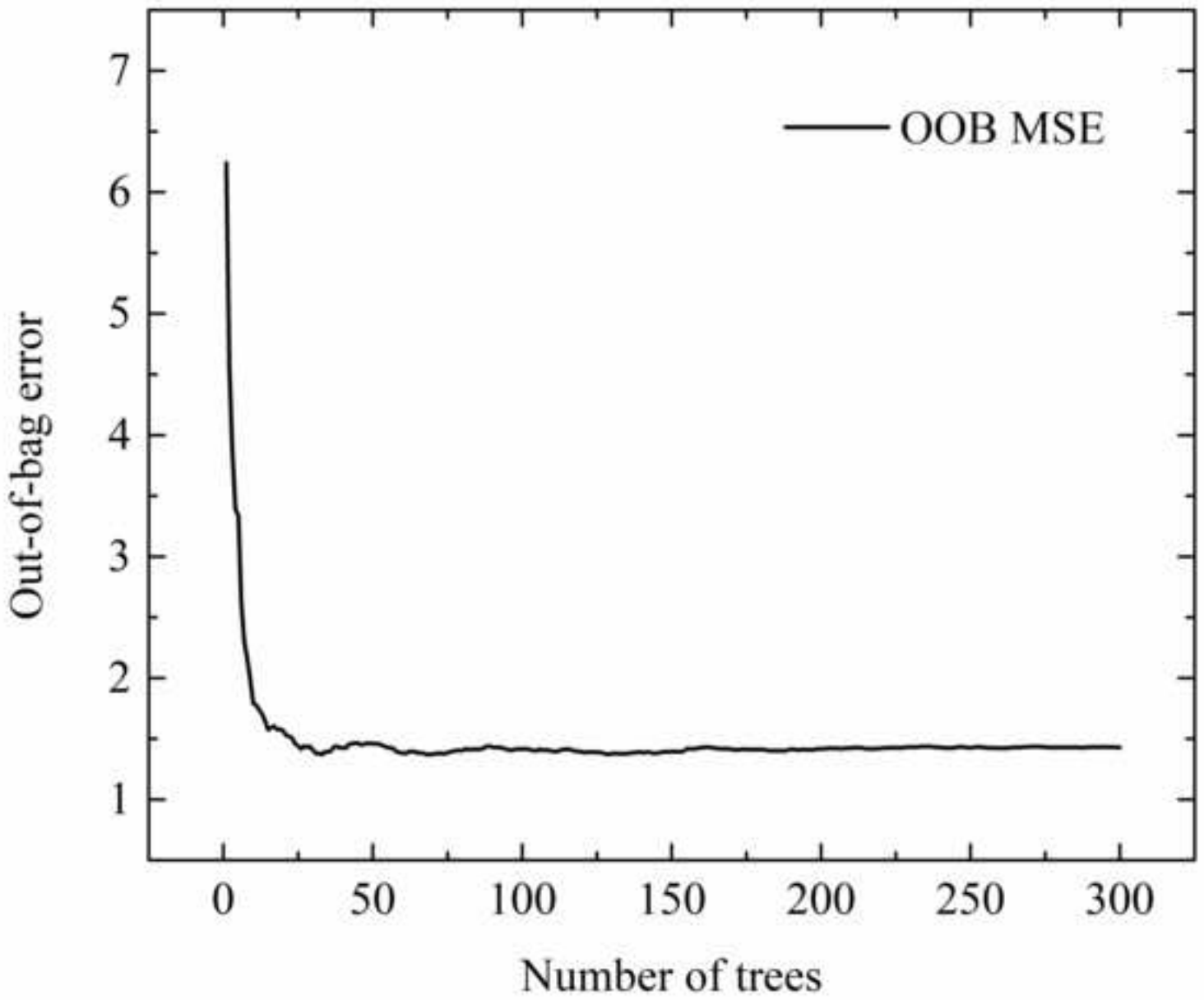


Figure 7a

Click here to access/download;Figure;Fig 7a.tif

Click here to view linked References

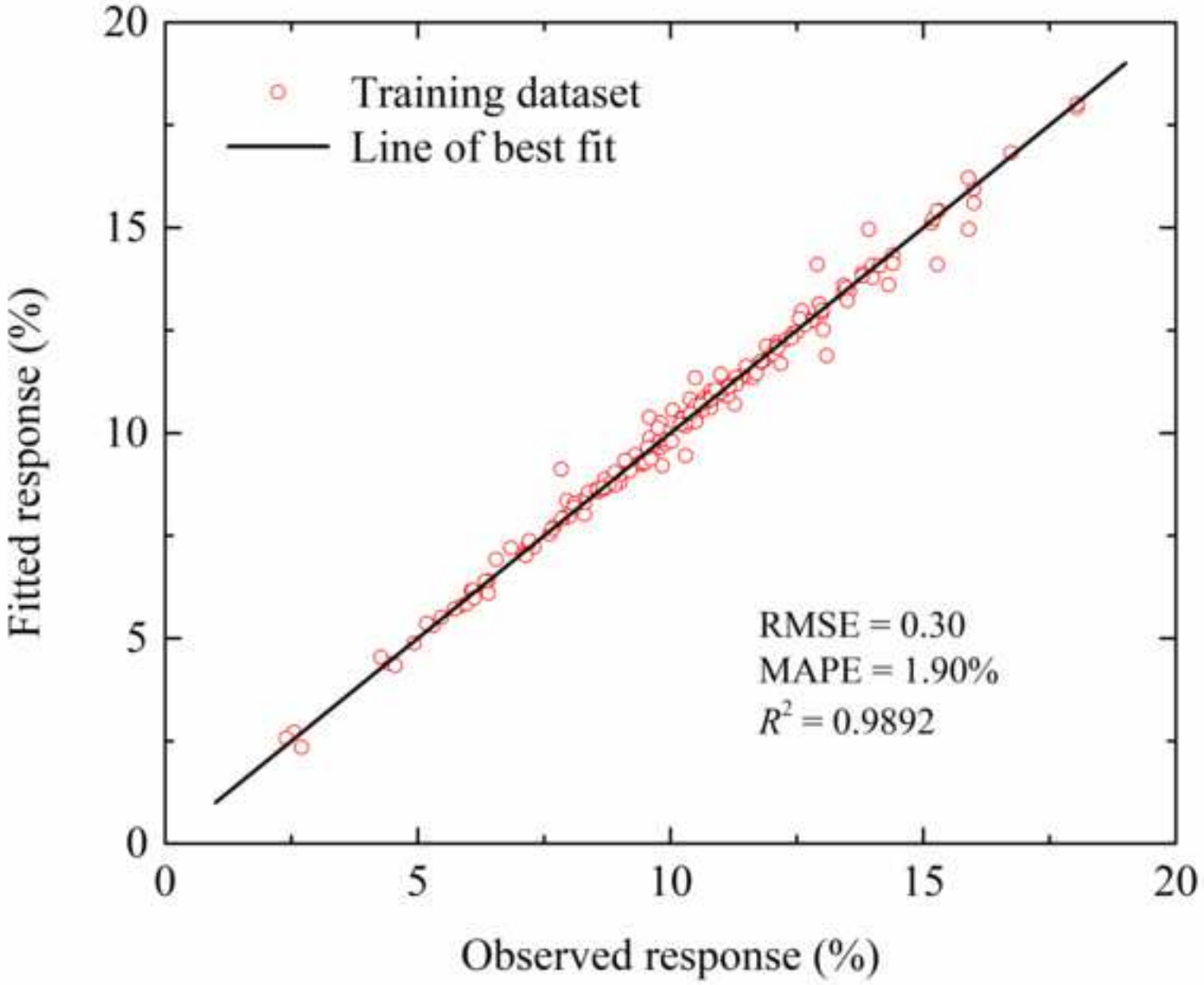
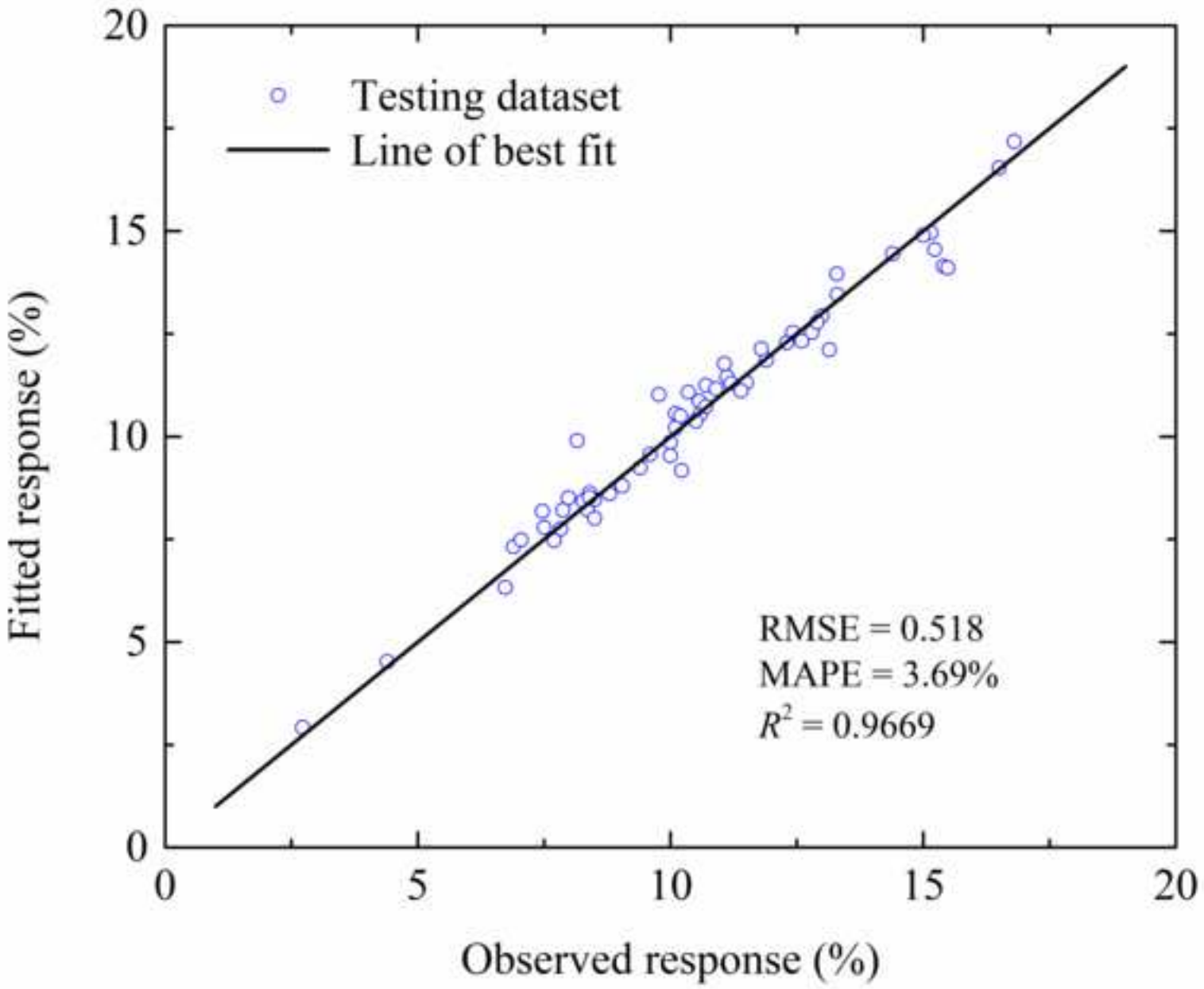


Figure 7b

Click here to access/download;Figure;Fig 7b.tif

Click here to view linked References



Algorithm: *Out-of-Bag* Predictor Importance Estimates by Permutation

Train random forests with B learners on the training data with total p predictor variables

For $b = 1$ **to** B **do**:

- Identify the OOB observations and the subset of J predictors ($J \le p$)
- Estimate prediction accuracy on the OOB sample $oobError^b$
- **For** $j = 1$ **to** J **do**:
  - Randomly permute the observations of x_j in the OOB sample
  - Re-estimate prediction accuracy on the permuted OOB sample $oobError_j^b$
  - Calculate the difference in accuracy $\Delta_j^b = oobError_j^b - oobError^b$
- **End**

End

Compute mean $\overline{\Delta}_j$ and standard deviation σ_{Δ_j} for each predictor x_j over B learners

Output: $VI(x_j) = \overline{\Delta}_j / \sigma_{\Delta_j}$ for $j = 1, \ldots p$.

Figure 9

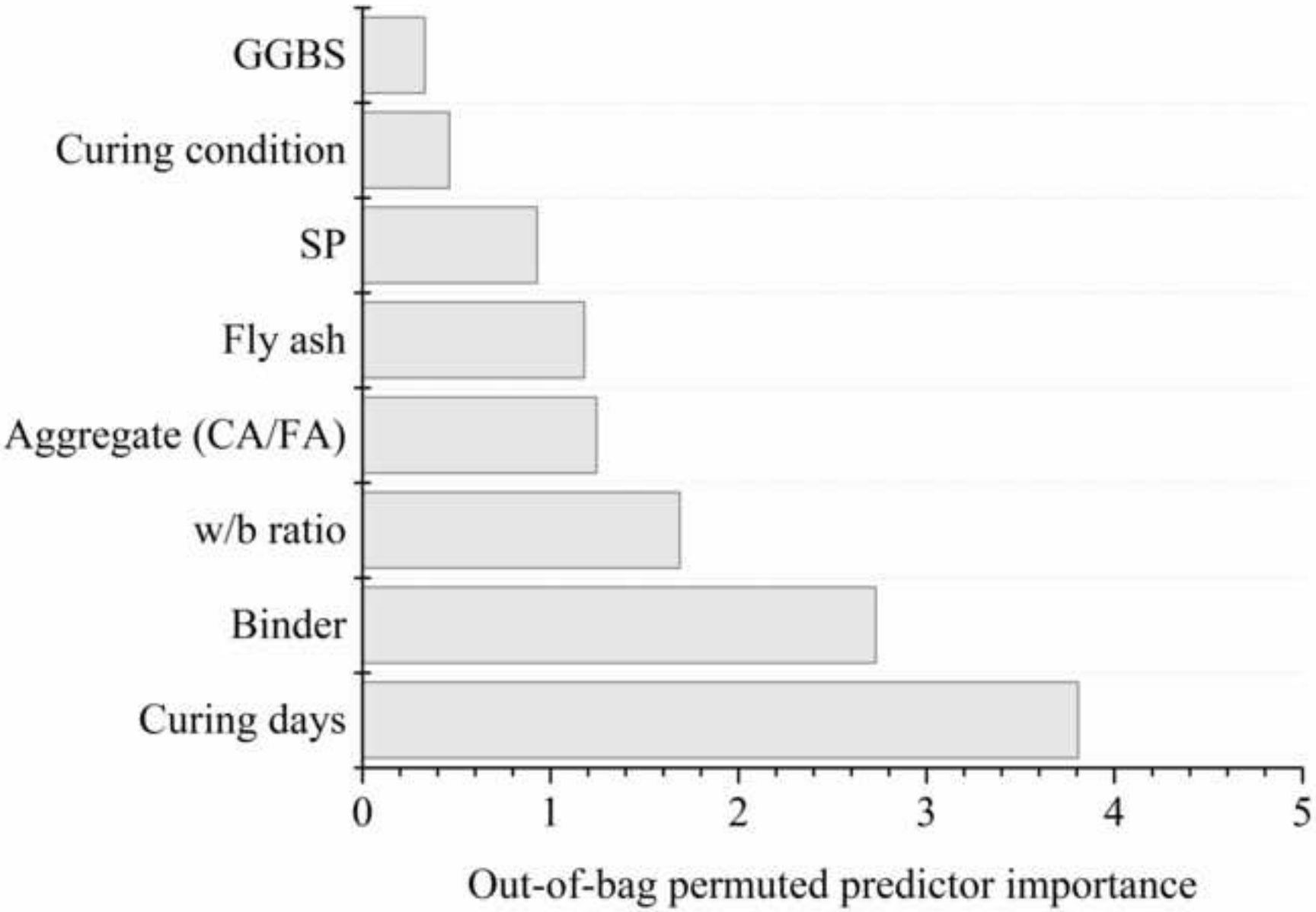


Figure 10

Click here to access/download;Figure;Fig 10.tif

Click here to view linked References

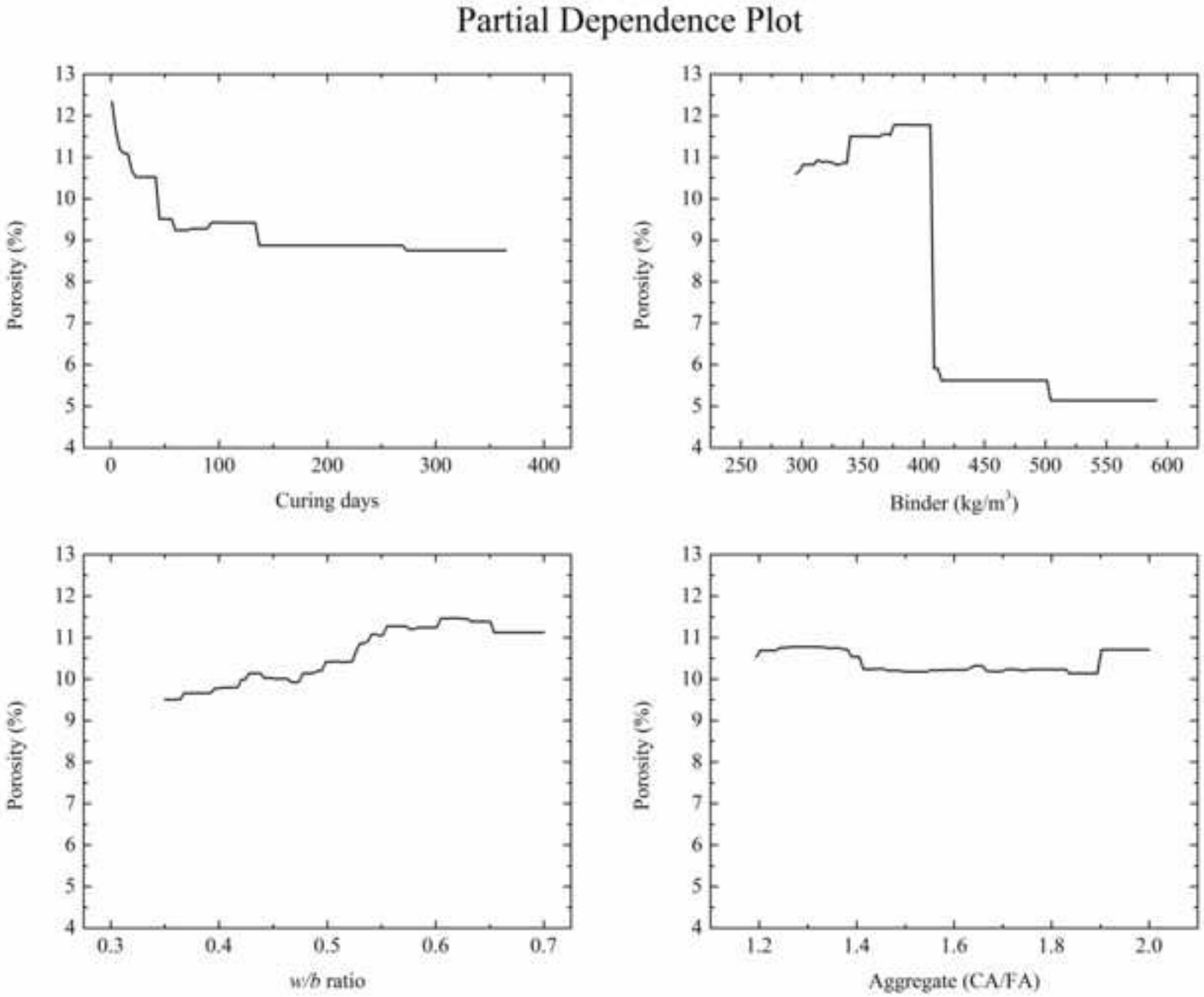


Figure 11

Click here to access/download;Figure;Fig 11.tif

Click here to view linked References

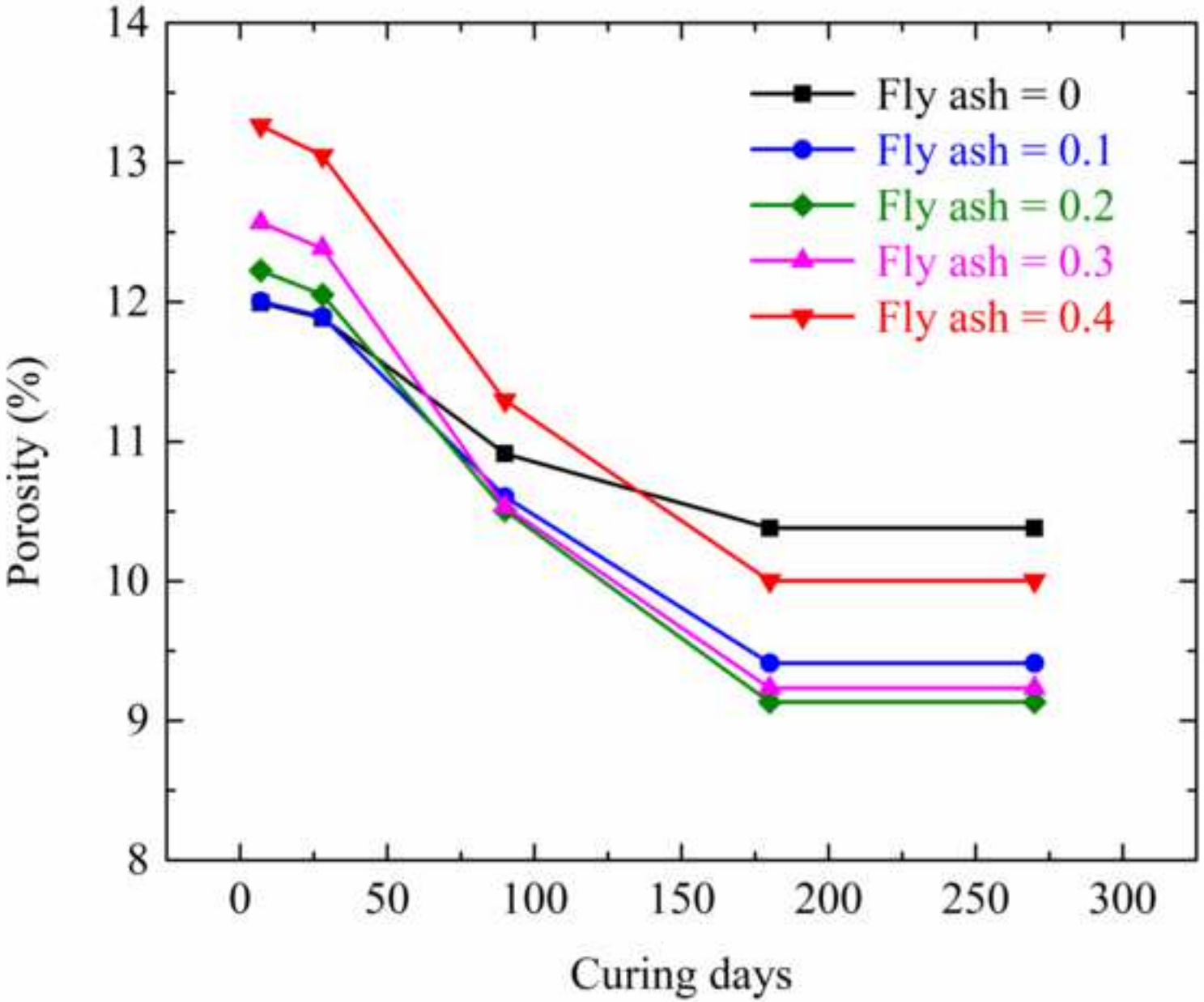


Figure 12

Click here to access/download;Figure;Fig 12.tiff

Click here to view linked References

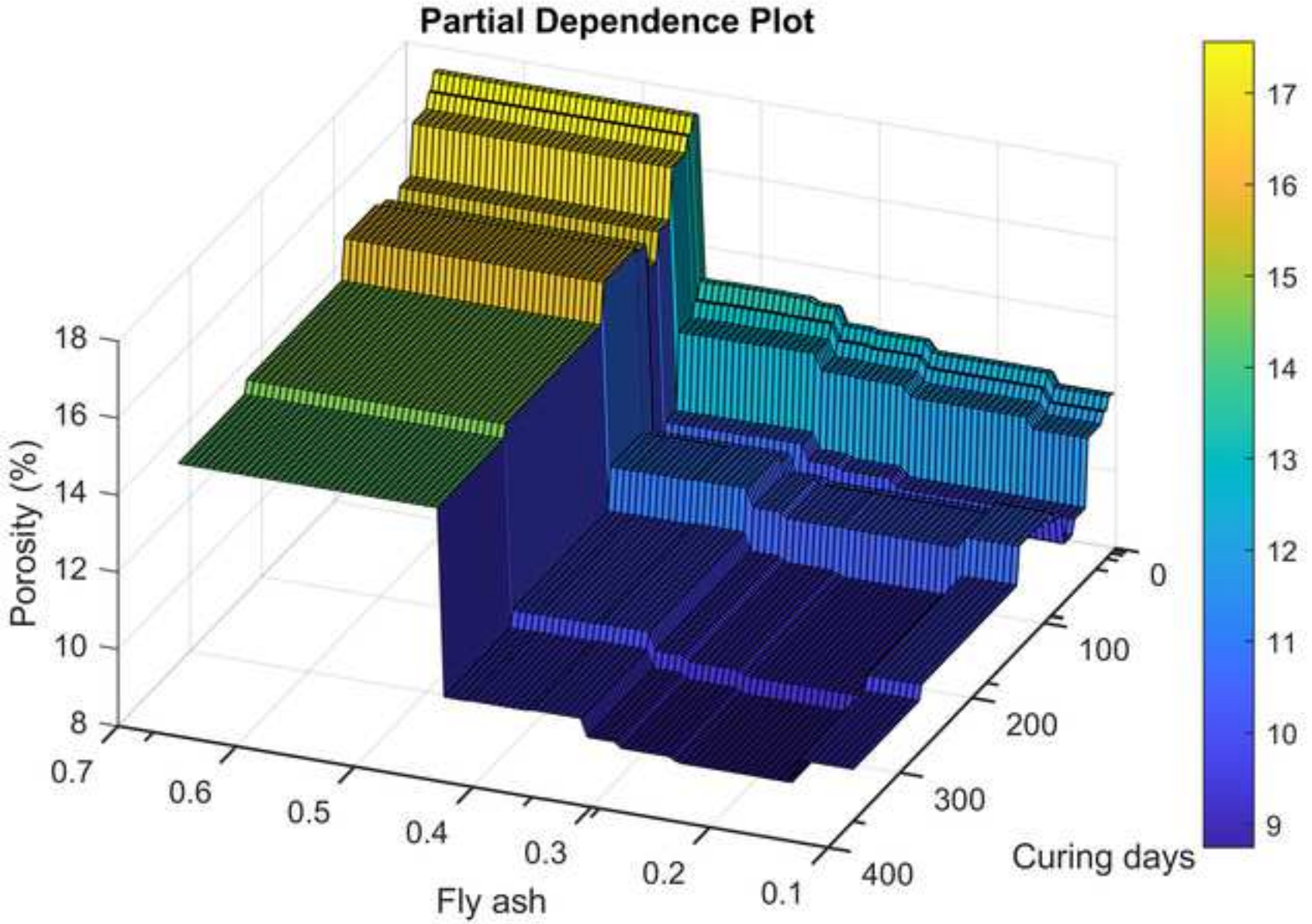


Figure 13

Click here to access/download;Figure;Fig 13.tif

Click here to view linked References

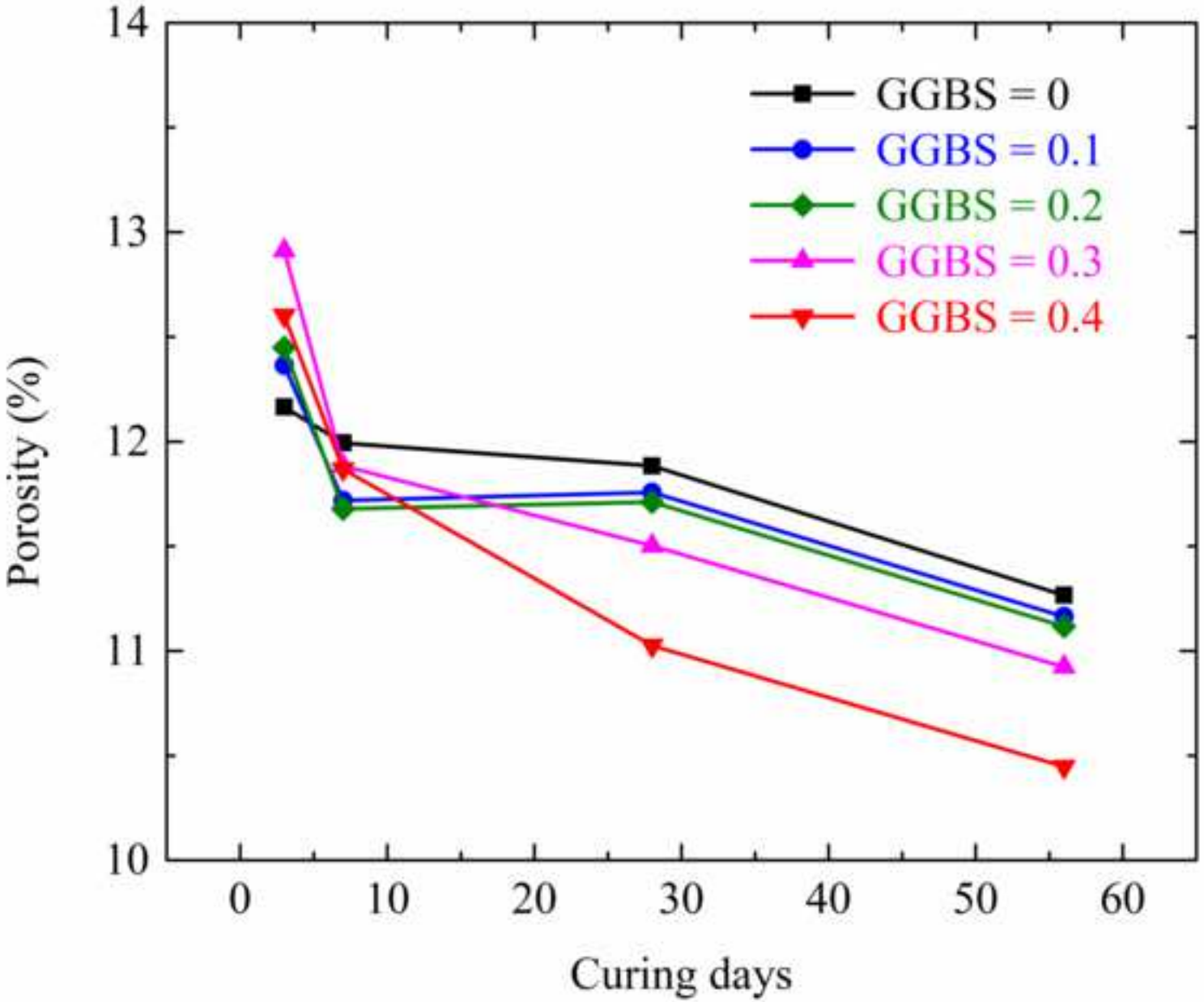


Figure 14

Click here to access/download;Figure;Fig 14.tiff

Click here to view linked References

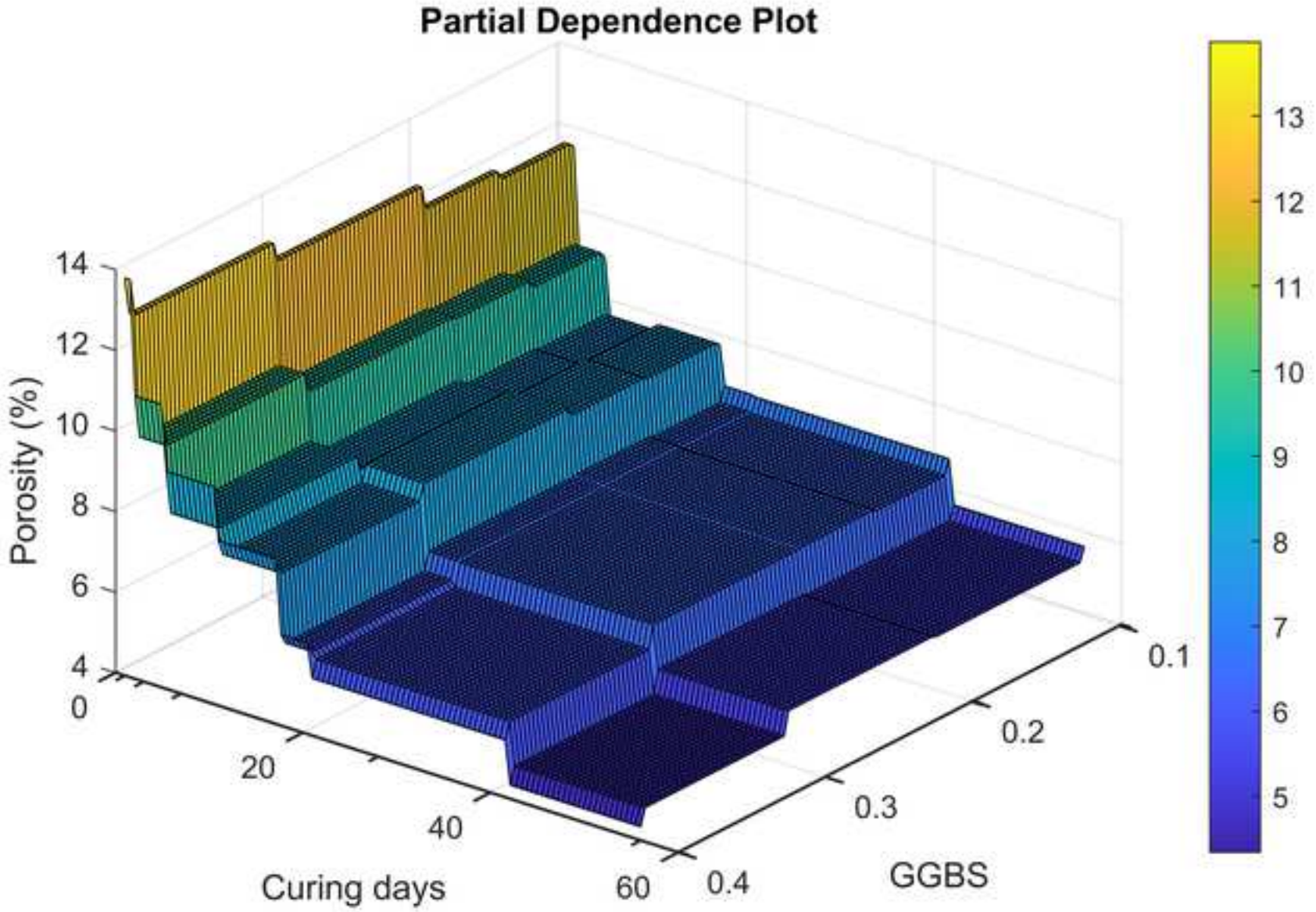


Figure 15a

Click here to access/download;Figure;Fig 15a.tif

Click here to view linked References

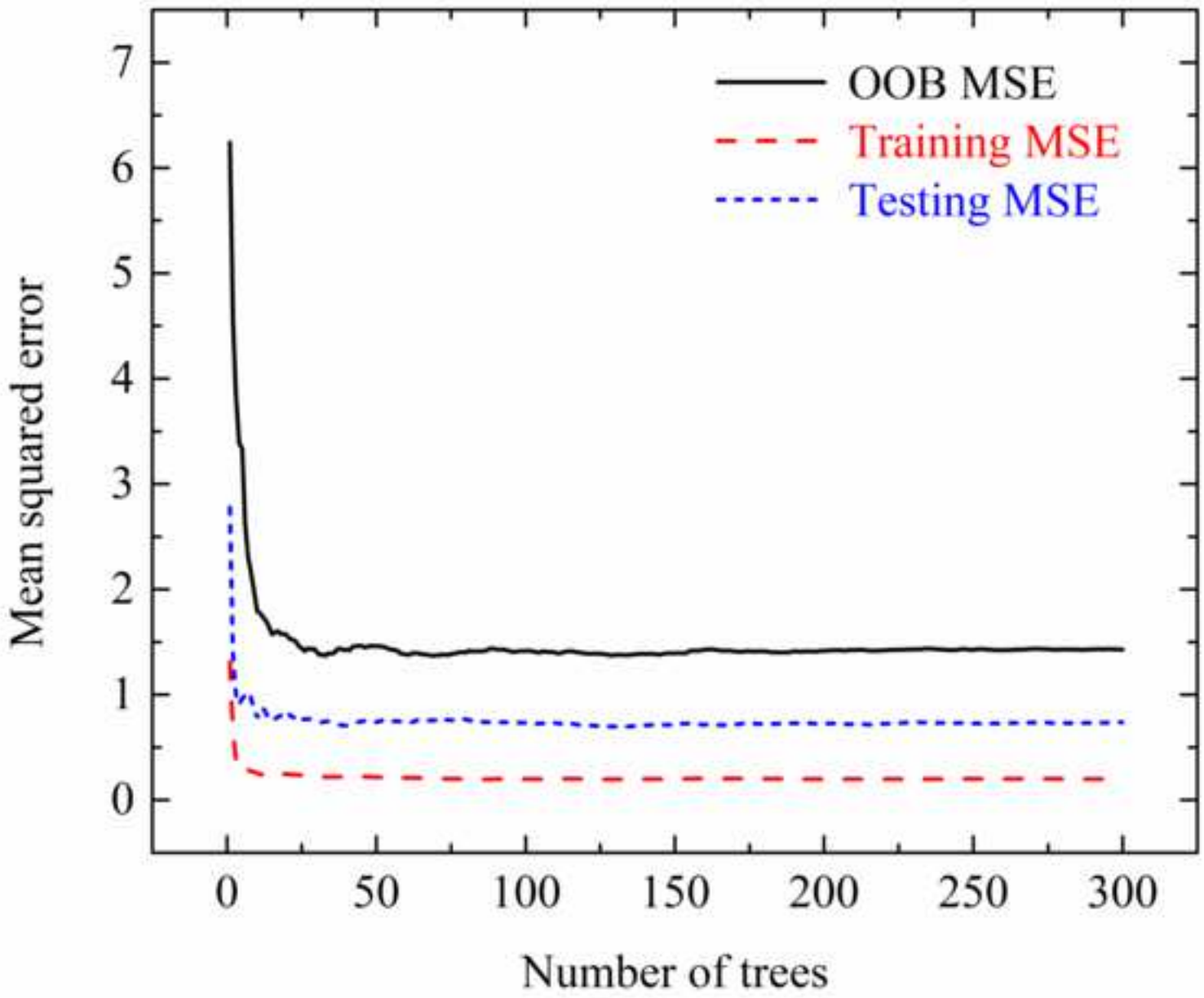


Figure 15b

Click here to access/download;Figure;Fig 15b.tif

Click here to view linked References

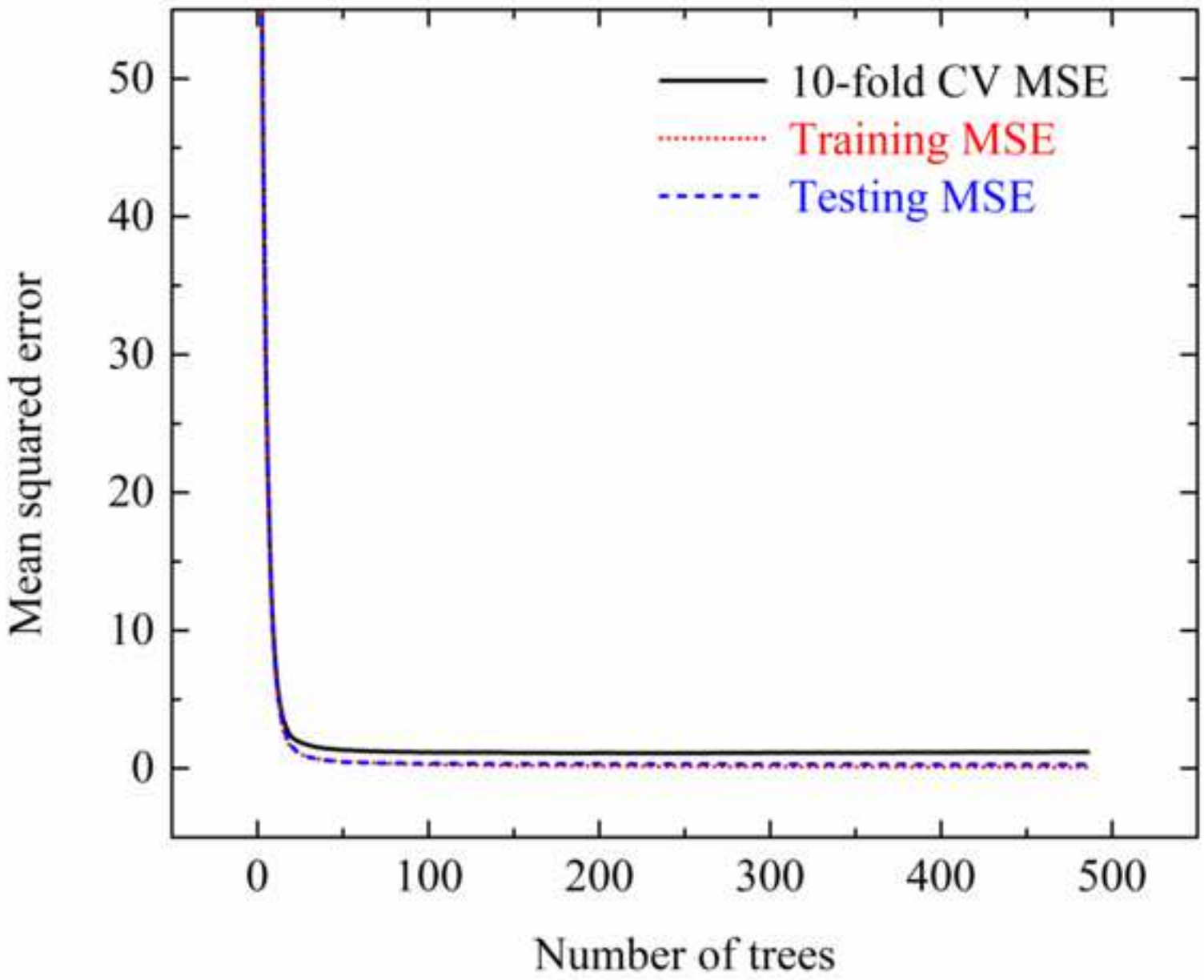


Figure 16

Click here to access/download;Figure;Fig 16.tif

Click here to view linked References

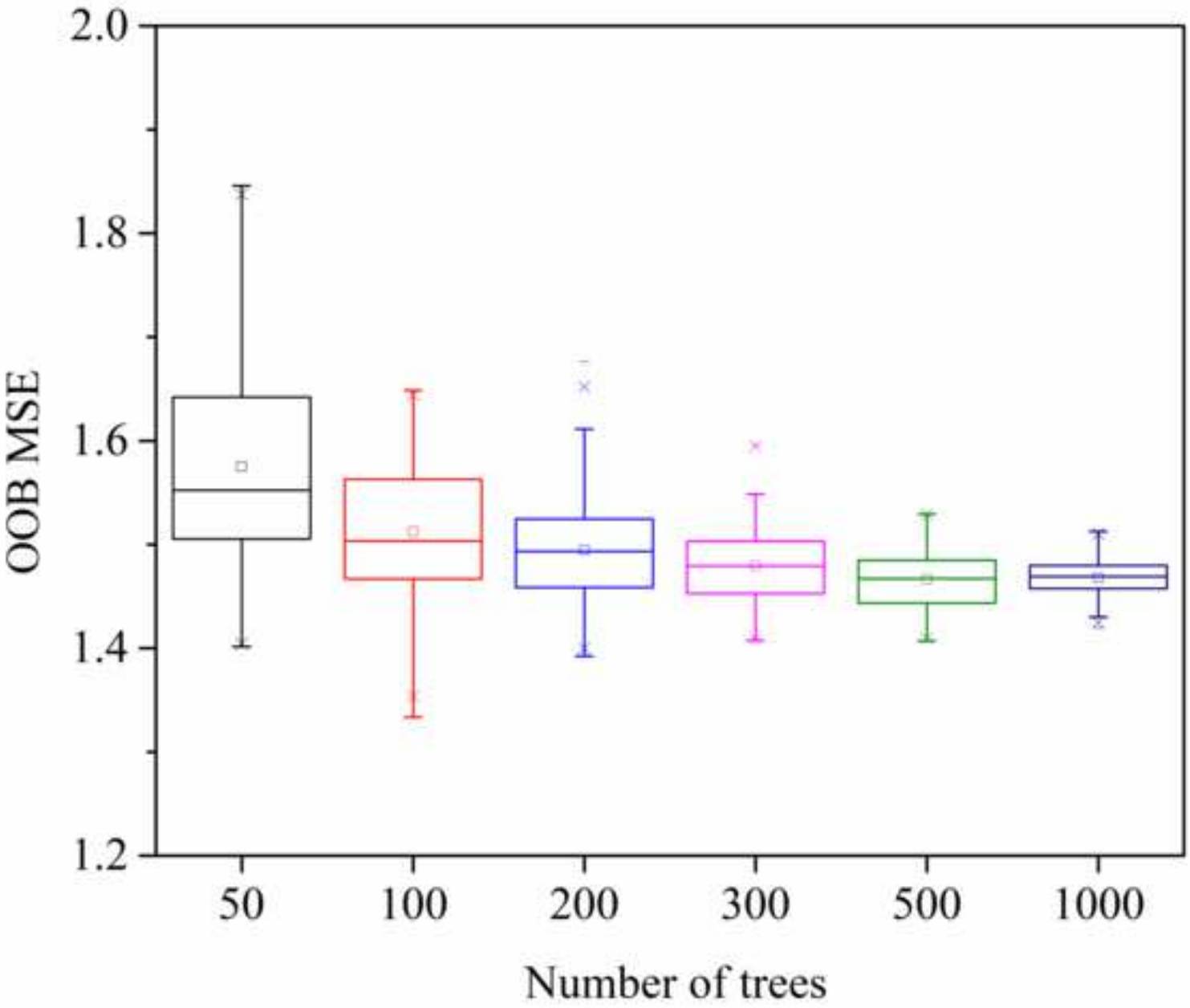
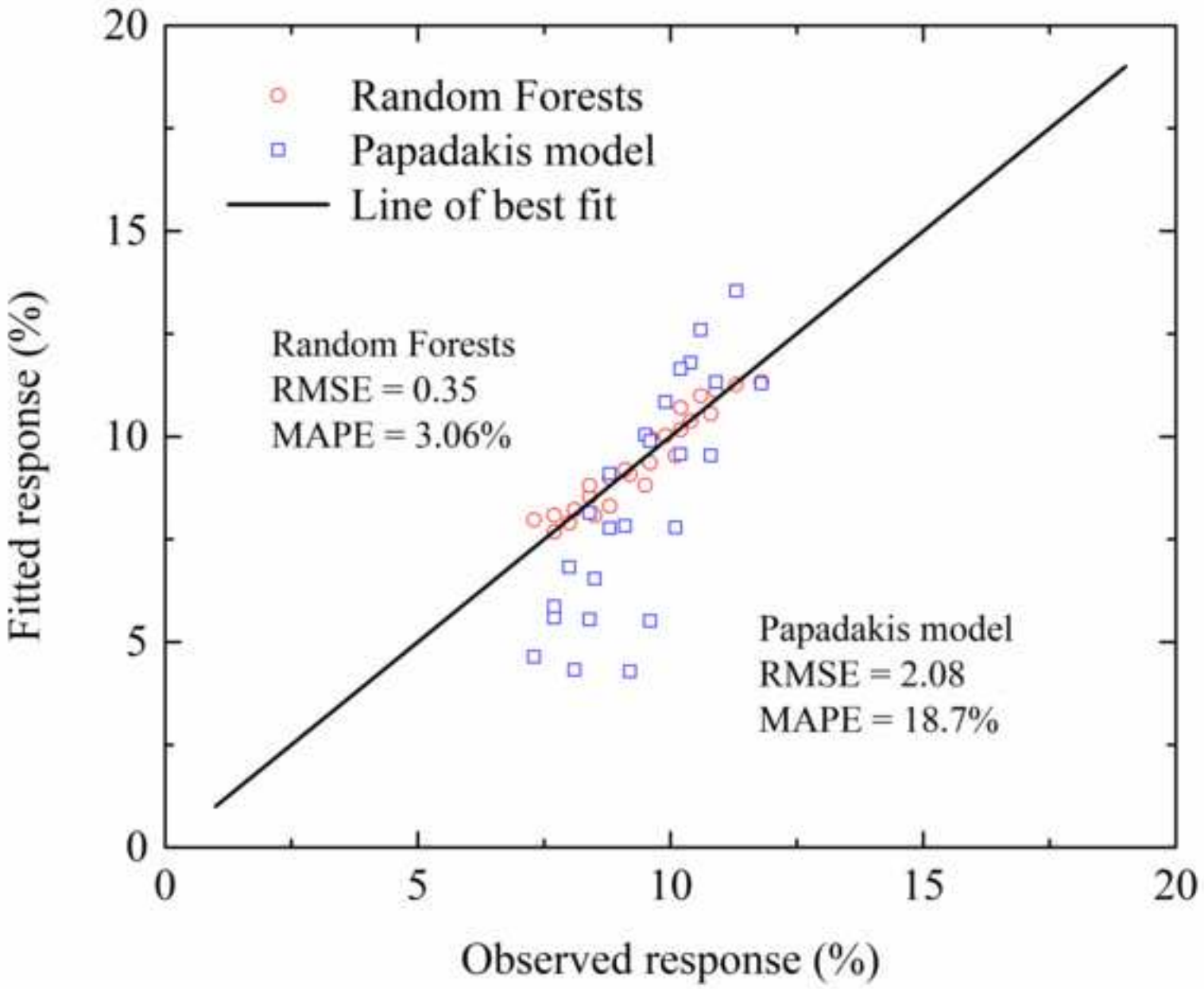


Figure 17

Click here to access/download;Figure;Fig 17.tif

Click here to view linked References



Declaration of interests

☒ 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.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: