





Article

Comparative Analysis of Machine Learning Models for Predicting Interfacial Bond Strength of Fiber-Reinforced Polymer-Concrete

Miljan Kovačević ^{1,*}, Marijana Hadzima-Nyarko ², Predrag Petronijević ³, Tatijana Vasiljević ¹ and Miroslav Radomirović ¹

¹ Faculty of Technical Sciences, University of Pristina, Knjaza Milosa 7, 38220 Kosovska Mitrovica, Serbia; tatijana.vasiljevic@pr.ac.rs (T.V.); miroslav.radomirovic@pr.ac.rs (M.R.)

² Faculty of Civil Engineering and Architecture Osijek, Josip Juraj Strossmayer University of Osijek, Vladimira Preloga 3, 31000 Osijek, Croatia; mhadzima@gfos.hr

³ Faculty of Civil Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11000 Belgrade, Serbia; pecap@grf.bg.ac.rs

* Correspondence: miljan.kovacevic@pr.ac.rs; Tel.: +381-606173801

Abstract: This study presents a detailed analysis of various machine learning models for predicting the interfacial bond strength of fiber-reinforced polymer (FRP) concrete, including multiple linear regression, Multigene Genetic Programming (MGGP), an ensemble of regression trees, Gaussian Process Regression (GPR), Support Vector Regression (SVR), and neural networks. The evaluation was based on their predictive accuracy. The optimal model identified was the GPR ARD Exponential model, which achieved a mean absolute error (MAE) of 1.8953 MPa and a correlation coefficient (R) of 0.9658. An analysis of this optimal model highlighted the most influential variables affecting the bond strength. Additionally, the research identified several models with lower expression complexity and reduced accuracy, which may still be applicable in practical scenarios.

Keywords: fiber-reinforced polymer; bond strength; machine learning models



Academic Editors: Jerzy Roslon,
Michał Podolski and Bartłomiej Sroka

Received: 20 November 2024

Revised: 13 January 2025

Accepted: 14 January 2025

Published: 17 January 2025

Citation: Kovačević, M.; Hadzima-Nyarko, M.; Petronijević, P.; Vasiljević, T.; Radomirović, M. Comparative Analysis of Machine Learning Models for Predicting Interfacial Bond Strength of Fiber-Reinforced Polymer-Concrete. *Computation* **2025**, *13*, 17. <https://doi.org/10.3390/computation13010017>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fiber-Reinforced Polymers (FRPs) are essential in modern construction due to their high strength-to-weight ratio, corrosion resistance, and ease of application, making them ideal for retrofitting concrete and masonry structures. Retrofitting unreinforced masonry (URM) buildings in earthquake-prone regions with FRPs is an effective method to improve seismic resistance, enhance load-bearing capacity, and increase ductility, thus addressing vulnerabilities caused by outdated construction techniques [1,2].

Traditional empirical models for predicting FRP-concrete bond strength have shown limitations in accurately capturing the complex interactions between these materials. This has led to a growing interest in applying machine learning (ML) techniques, which are capable of handling large datasets and uncovering intricate patterns that might be missed by conventional approaches. Recent studies have demonstrated the effectiveness of ML algorithms.

Wu and Jiang's research [3] focused on quantifying and modeling interfacial bond parameters through both experimental and analytical studies. They developed a comprehensive database comprising 628 shear tests of externally bonded FRP joints. Their analysis identified key factors affecting bond parameters, particularly focusing on bond strength and fracture energy. Additionally, they derived new models for the width factor as a function of width ratio and concrete strength, offering more accurate bond strength

and fracture energy predictions. This work contributed to a better understanding of how bond behavior is influenced by variables such as the width of the FRP and the strength of the concrete [3].

Zhou et al. [4] developed an artificial neural network (ANN) model to predict the interfacial bond strength between FRP and concrete. Utilizing a substantial database of 969 test results, they trained the ANN model using backpropagation neural networks (BPNN), incorporating weighted values, biases, and transfer functions. Their model demonstrated higher accuracy than 20 existing models, with significantly lower predictive errors. The study also provided an explicit formula derived from the ANN model, which can be used in practical design applications for predicting bond strength [4].

In their research, Li et al. [5] compiled a large dataset to assess the bond strength and bond stress-slip behavior of FRP-reinforced concrete members. Their study also incorporated an evaluation of environmental durability factors that could affect bond performance. The study is crucial for its contribution to understanding how reduction factors in bond strength are influenced by both the environment and material properties [5].

Su et al. [6] explored the potential of ML approaches, including multiple linear regression, support vector machine (SVM), and artificial neural networks (ANN), for predicting the interfacial bond strength (IBS) between FRP and concrete. Two datasets were used: Dataset 1 (122 IBS values), and Dataset 2 (136 IBS values). The variables in Dataset 1 include FRP properties such as elastic modulus, tensile strength, and bond length, while Dataset 2 uses fewer input features but focuses on stiffness and groove dimensions. Among the ML models, the SVM model demonstrated superior predictive ability with R^2 values of 0.79 for Dataset 1 and 0.85 for Dataset 2. This research highlights the capability of advanced ML models in predicting FRP-concrete bond strength [6].

The study by Haddad and Haddad [7] employed an artificial neural network (ANN) technique to predict the bond strength between FRP and concrete. The researchers compiled an extensive dataset comprising over 440 data points, allowing the ANN model to assess various parameters such as FRP properties, concrete compressive strength, and bond length. The model demonstrated impressive predictive accuracy, achieving high correlation coefficients (R^2 nearing 0.98) and significantly outperformed traditional empirical methods. This study underscores the effectiveness of ANN models in predicting complex interactions between FRP and concrete in structural applications [7].

Chen et al. [8] developed a prediction model using the ensemble learning method known as Gradient Boosted Regression Trees (GBRT) to estimate the interfacial bond strength between FRP and concrete. Their model was trained using a comprehensive database of 520 tested samples. The GBRT model demonstrated superior accuracy, with R^2 values of 0.9627 during training and 0.9269 during testing. The model outperformed both ANN and SVM models, making it a robust tool for predicting FRP-concrete bond strength in practical engineering applications [8].

Barkhordari et al. [9] introduced hybrid models that combine population-based algorithms (Bald Eagle Search, Manta Ray Foraging Optimization, and Runge-Kutta (RUN) optimizer) with ANNs to estimate FRP-concrete interfacial bond strength. The study utilized a large dataset of experimental samples and found that the RUN-ANN model achieved the best performance, with an R^2 value of 0.92. Moreover, the Shapley Additive Explanations (SHAP) method was used to interpret the model, identifying FRP bond length and width as key factors influencing bond strength predictions [9].

Alabdullh et al. [10] applied a hybrid ensemble ML approach to predict the bond strength of FRP laminates bonded to concrete. Using a database of 136 samples, the team trained and validated several standalone ML models, including ANN, Extreme Learning Machine, and GPR. The hybrid ensemble model (HENS) achieved superior predictive

accuracy, with an R^2 value of 0.9783 for training and 0.9287 for testing, demonstrating its effectiveness in overcoming overfitting issues common in traditional ML models [10].

In their study, Kim et al. [11] employed ensemble ML techniques to predict FRP-concrete interfacial bond strength. Using a dataset of 855 single-lap shear tests, they evaluated various ensemble methods, with the CatBoost algorithm achieving the best performance. The algorithm showed a root mean square error (RMSE) of 2.310, a coefficient of variation (COV) of 21.8%, and an R^2 value of 96.1%, suggesting its high accuracy and potential for practical applications [11].

Recent studies have explored various advanced methods to improve predictive accuracy and optimize engineering practices, with a common focus on hybrid models. Jamhiri et al. [12] proposed a hybrid approach combining multivariate regression with growth forecasting, demonstrating the effectiveness of robust regression [12]. Bui et al. [13] utilized hybrid ML algorithms, such as Random Forest (RF), in combination with bagging, parameter selection, and filtered classification techniques to enhance prediction capabilities [13]. Additionally, Parhi and Patro [14] focused on using hybrid ML models, emphasizing the role of optimization techniques in improving the performance of conventional models [14].

This research aims to advance existing methodologies by developing a robust ML model specifically designed to predict FRP-concrete bond strength, using a comprehensive dataset of experimental results. The study evaluates several ML models to identify the most significant input variables and further interprets their predictions using advanced methods for feature importance analysis. Through this approach, the study aims to contribute to the ongoing efforts in optimizing FRP applications in civil engineering, ensuring more reliable and efficient reinforcement solutions.

2. Materials and Methods

2.1. Multiple Linear Regression

Linear regression analysis is a statistical method used to determine the causal effect of one or more independent variables on a dependent variable. When the problem involves one dependent variable (Y) and multiple independent variables (X_1, X_2, \dots, X_p), multiple linear regression can be used. The model can be described by the Equation (1):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \epsilon \quad (1)$$

In this equation β_0 is the intercept, $\beta_1, \beta_2, \dots, \beta_p$ are the coefficients of the independent variables, ϵ represents the error term.

To estimate the parameters ($\beta_0, \beta_1, \beta_2, \dots, \beta_p$) of the linear model, the least squares method is used. The goal is to minimize the sum of the squares of the differences between the observed values and the values predicted by the model. This is done using the following Equation (2):

$$\hat{\beta} = (X^T X)^{-1} X^T Y \quad (2)$$

where $\hat{\beta}$ is the vector of estimated parameters, X is the matrix of input variables, and Y is the vector of observed output values.

2.2. Multi Gene Genetic Programming

Genetic Programming (GP) is an evolutionary algorithm-based methodology used for modeling and optimization. It generates models by simulating the process of natural selection, where a population of potential solutions evolves over time, with the best solutions “surviving” and “reproducing”. This methodology is particularly suited to problems where the functional form of the model is not known in advance.

In regression tasks, GP evolves models that predict continuous outputs based on input variables. The objective is to find a symbolic representation (a mathematical formula) that best approximates the relationship between input and output data. The flexibility of GP allows it to discover complex, non-linear relationships that may be difficult to capture using traditional regression techniques [15,16].

The process starts by generating an initial population of random solutions (Figure 1). Each individual in this population represents a possible model for the problem, and it is generated using one or more decision trees. In the initial iteration, the tree model is generated by randomly selecting mathematical functions, constants, and model variables. Within this framework, each tree can be thought of as representing a single “gene” in the model. The trees culminate in terminal nodes, which are assigned either as input variables from the model or as fixed constants. All other intermediate nodes in the tree serve as functional nodes.

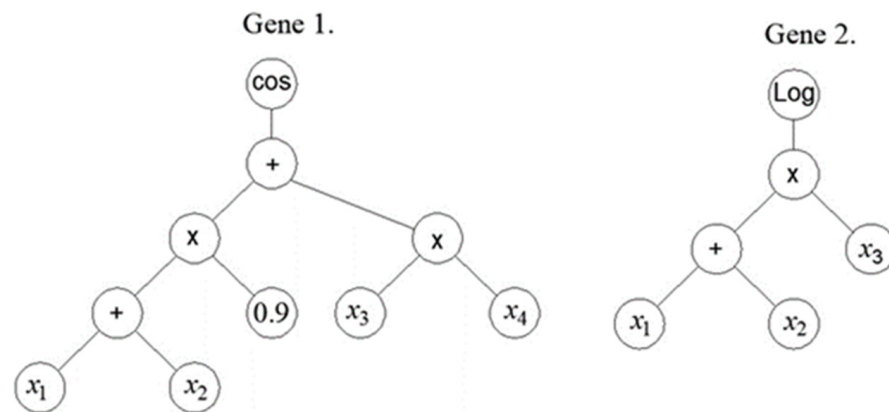


Figure 1. Example of an MGGP model with two genes [16].

Each model in the population is evaluated based on a fitness function, which quantifies how well the model performs in predicting the desired output. Commonly used metrics include the RMSE, MAE, and other error-based criteria.

Selection process of models is based on their fitness and models with better performance have a higher probability of being selected, but some randomness is introduced to maintain diversity within the population.

Selected models are paired to create new “offspring” models by exchanging parts of their structures (Figure 2). In constructing new models, the best-performing models from the previous generation are utilized to generate a new population through processes such as crossover, mutation, and direct copying. In practice, the selection of models for reproduction is probabilistic, based on their fitness score or the complexity of the model structure. Model complexity is often determined by the number of nodes and subtrees within a tree, reflecting the expressive capacity of the model.

During the crossover process, genetic material from two parent models can be exchanged entirely or partially (Figure 2). For example, suppose the model J_1 contains the genes $[G_{(1,1)}, G_{(2,1)}, G_{(3,1)}, G_{(4,1)}, G_{(5,1)}]$ and another model J_2 contains the genes $[G_{(1,2)}, G_{(2,2)}, G_{(3,2)}, G_{(4,2)}, G_{(5,2)}]$. If we select a random set of genes for crossover, the models may look like (3), (4):

$$J_1 : [G_{(1,1)}, < \mathbf{G}_{(2,1)}, \mathbf{G}_{(3,1)} > G_{(4,1)}, G_{(5,1)}] \tag{3}$$

$$J_2 : [G_{(1,2)}, G_{(2,2)}, < \mathbf{G}_{(3,2)} > G_{(4,2)}, G_{(5,2)}] \tag{4}$$

The bolded genes represent those selected for crossover, and their exchange results in the following offspring (5), (6):

$$O_1 : [G_{(1,1)}, G_{(3,2)}, G_{(4,1)}, G_{(5,1)}] \tag{5}$$

$$O_2 : [G_{(1,2)}, G_{(2,2)}, G_{(2,1)}, G_{(3,1)}, G_{(4,2)}, G_{(5,2)}] \tag{6}$$

This high-level crossover process enables the creation of new models that inherit features from both parent models.

Additionally, crossover can occur at the gene level, known as low-level crossover, where only a portion of a gene is exchanged.

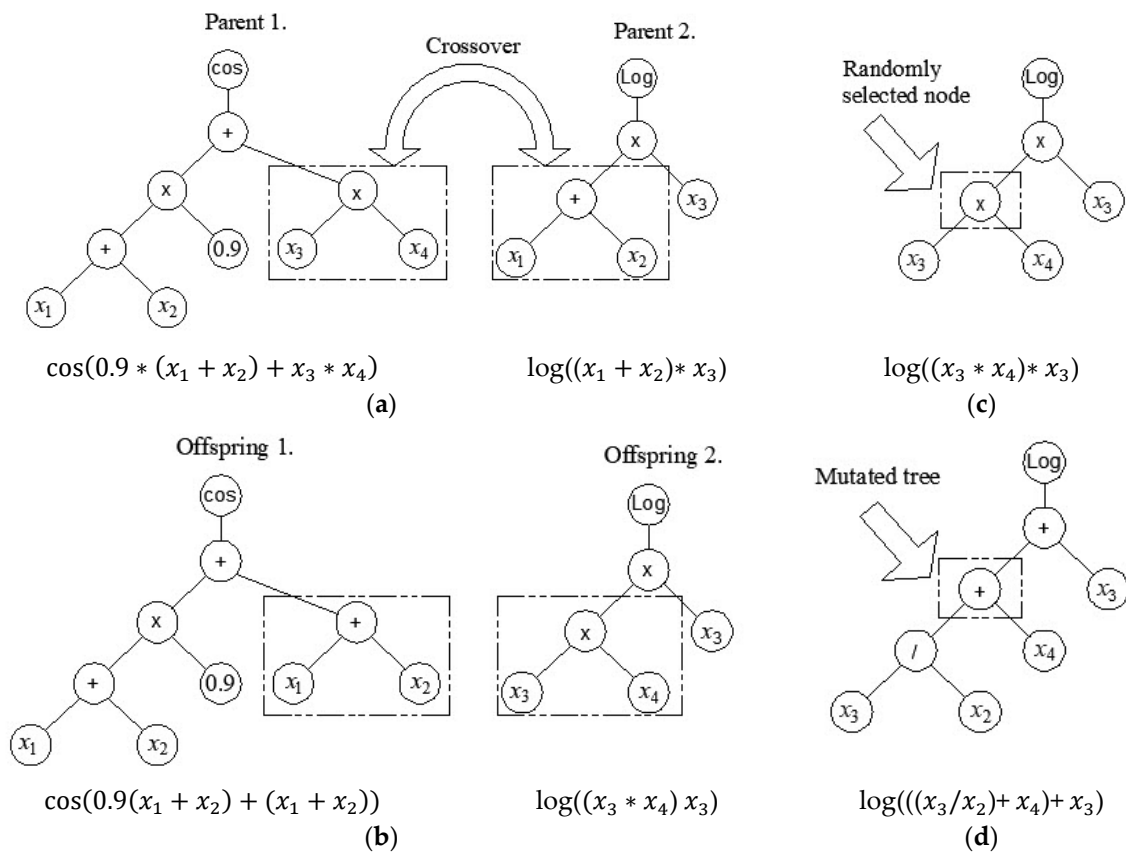


Figure 2. Gene combination and mutation in MGGP models: (a) Selection of gene segments for crossover, (b) Crossover of gene segments, (c) Selection of gene segments for mutation, (d) Mutation of a gene segment [16].

In such cases, only part of the tree is altered (Figure 2a,b). Mutation can also occur at this level, where a node within a gene is randomly selected, and a newly created subtree is inserted at that location (Figure 2c,d). These evolutionary procedures are repeated over multiple generations to refine the model.

The resulting model is pseudo-linear, as it represents a linear combination of non-linear components modeled as trees. Mathematically, this multigene regression model is represented as (7) [15,16]:

$$\hat{y} = b_0 + b_1t_1 + b_2t_2 + \dots + b_Gt_G \tag{7}$$

where b_0 is the bias term, b_i are the scaling parameters, and t_i are the outputs from the i -th tree (gene).

The overall gene response matrix is denoted by (8):

$$G = [1, t_1, t_2, \dots, t_G] \tag{8}$$

with dimensionality $N \times (G + 1)$, and the coefficient vector is (9):

$$b = [b_0 \ b_1 \ b_2 \ \dots \ b_G]^T \tag{9}$$

The vector b is estimated using the least-squares method, and its solution is computed as (10):

$$b = (G^T G)^{-1} G^T y \tag{10}$$

Thus, the final multigene regression model can be written as (11):

$$\hat{y} = Gb \tag{11}$$

The process of selection, crossover, and mutation is repeated over many generations. Over time, the population evolves, and the models typically improve in performance. The algorithm terminates when a stopping criterion is met. This criterion could be a predefined number of generations, reaching a certain fitness threshold, or convergence in the population.

2.3. Regression Trees Ensembles: Boosted Trees, Bagging and Random Forest

Regression trees are a subset of decision trees tailored to predict continuous outcomes. These models partition the input space into regions using binary recursive splitting, where each leaf represents a specific numerical prediction (Figure 3).

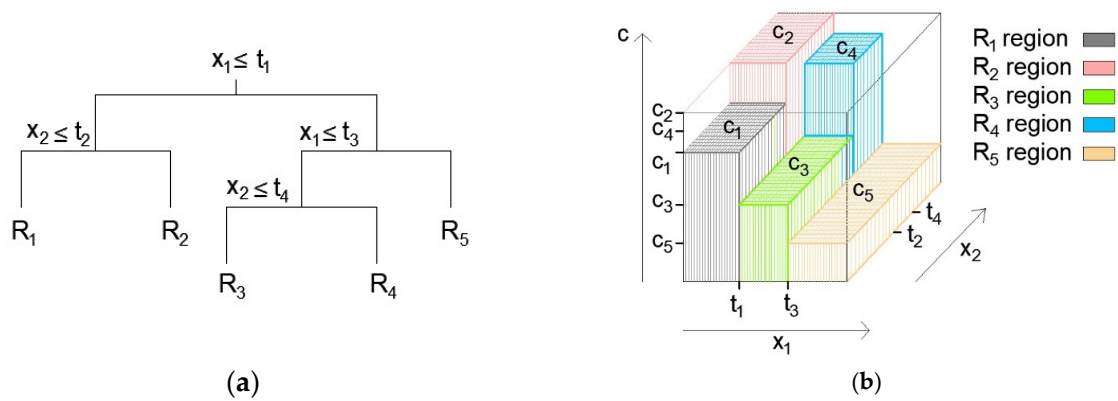


Figure 3. Region-based space partitioning (a) and 3D regression surface in a regression tree (b).

The objective at each node is to select the best split by minimizing the residual sum of squares between the predicted and actual values in the resulting regions [17–19].

For a given dataset containing N observations with p input variables, denoted as (x_i, y_i) , where x_i represents an input vector and y_i the corresponding output, the tree construction aims to minimize the following expression (12):

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right] \tag{12}$$

where $R_1(j, s)$ and $R_2(j, s)$ are the two regions after splitting on the j -th variable at point s , c_1 and c_2 are the mean predictions within these regions.

Once the input space is divided into M regions, the prediction for any new observation x is the mean of the outcomes y_i for the region R_m to which x belongs, given by (13):

$$f(x) = \sum_{m=1}^M c_m I(x \in R_m) \tag{13}$$

where c_m represents the average output in region R_m and $I(x \in R_m)$ is an indicator function that is 1 if $x \in R_m$ and 0 otherwise.

Regression trees, while interpretable and flexible, tend to suffer from high variance, meaning that even small changes in the training data can result in significant differences in the structure and predictions of the tree. This instability makes regression trees prone to overfitting, especially when the model becomes too complex by learning specific noise in the training data.

2.3.1. Boosting Methodology

Boosting is an ensemble learning technique in which models are trained sequentially. Each successive model aims to improve the performance of the ensemble by correcting errors from previous iterations (Figure 4). Specifically, Gradient Boosting focuses on minimizing a predefined loss function by iteratively adding new models to an ensemble [20–24]. Each newly added regression tree optimizes its predictions based on the residuals, the differences between the actual and predicted values of the preceding models.

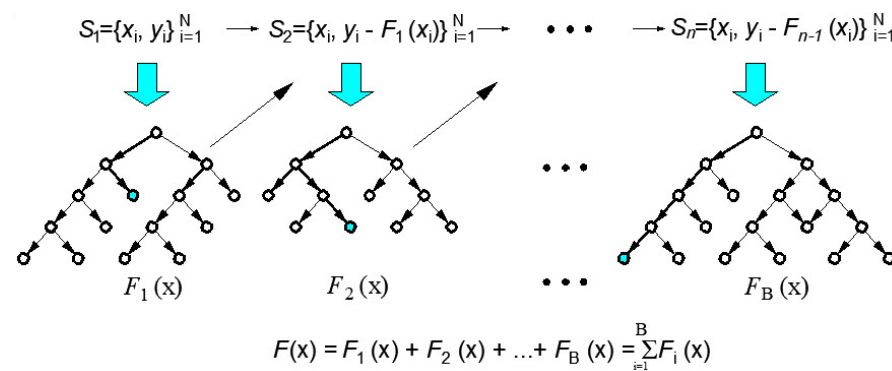


Figure 4. Gradient boosting ensemble model.

Mathematically, for a given set of training data $D = \{(x_i, y_i)\}_{i=1}^I$, where $x_i \in R^n$ represents the input features and $y_i \in R_y$ are the target variable, the objective of Gradient Boosting is to minimize a differentiable loss function $L(y, \hat{y})$ where \hat{y} is the predicted output. The general form of the model can be expressed as (14):

$$F(x) = \sum_{i=1}^B \alpha_m h_i(x) \tag{14}$$

where $F(x)$ represents the final model prediction, α_m is the weight of each weak learner, and $h_m(x)$ is the prediction from the m -th weak learner, typically a regression tree.

At each iteration m , the new tree $h_m(x)$ is trained to minimize the residuals from the previous iteration. The residuals $r_i(m)$ at the m -th step are calculated as (15):

$$r_i^{(m)} = - \left[\frac{\partial L(y_i, \hat{y}_i^{(m)})}{\partial \hat{y}_i^{(m)}} \right] \tag{15}$$

In the case of quadratic error (i.e., squared loss), the residuals simplify to (16):

$$r_i^{(m)} = y_i - F_{m-1}(x_i) \tag{16}$$

Here, $F_{m-1}(x_i)$ is the predicted value from the ensemble up to iteration $m - 1$. The new regression tree is then fitted to the residuals in order to minimize the overall loss.

2.3.2. Bagging and Random Forest (RF) Methodology

To address limitations of individual regression trees and improve the predictive performance, ensemble methods such as bagging and RF are commonly used (Figure 5). These methods combine multiple decision trees to reduce variance and create more robust models. Bagging (Bootstrap Aggregating) works by generating multiple bootstrap samples from the original dataset. Each sample is created by randomly selecting data points with replacement, meaning some data points may be repeated while others are left out. For each bootstrap sample, a separate regression tree is trained independently. Instead of relying on a single tree, bagging reduces variance by averaging the predictions from all trees in the ensemble, thus providing a more stable and accurate prediction [20–24].

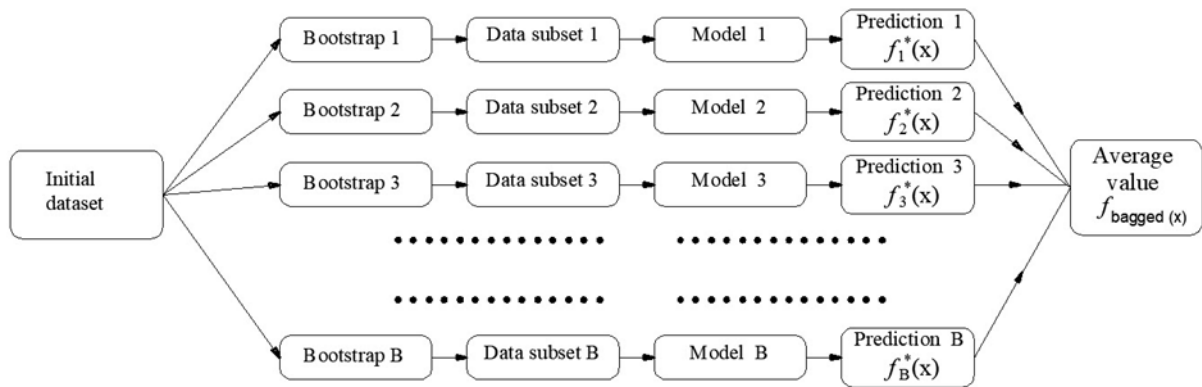


Figure 5. Bootstrap aggregation (Bagging) in regression tree ensembles.

Mathematically, if $f_b^*(x)$ represents the prediction from the b -th bootstrap sample, the final prediction from bagging is given by the average of all individual predictions (17):

$$f_{bagged}(x) = \frac{1}{B} \sum_{b=1}^B f_b^*(x) \tag{17}$$

where B is the total number of bootstrap samples, and $f_b^*(x)$ is the prediction from the b -th regression tree. By averaging the predictions, bagging significantly reduces the model’s variance, making it less sensitive to the idiosyncrasies of individual training datasets. This leads to improved generalization on unseen data while still maintaining the flexibility and interpretability of individual decision trees.

RF build upon bagging by introducing an additional layer of randomness. Instead of considering all variables for each split, RF select a random subset of the variables at each node, which decorrelates the trees and further reduces variance. This process ensures that no single variable dominates the tree-building process.

2.4. Support Vector Regression (SVR)

Support Vector Regression (SVR) is an adaptation of the Support Vector Machines (SVM) algorithm, designed for continuous output prediction tasks. SVR transforms the input space into a higher-dimensional feature space using kernel functions, allowing for

the linear separation of data points that may not be linearly separable in the original input space [25–27].

Given a dataset $(D = \{(x_i, y_i)\}_{i=1}^l)$, where $x_i \in R^n$ and $y_i \in R$ are the corresponding continuous output values, the goal of SVR is to find a function $f(x)$ that has at most an ϵ deviation from the actual output y_i for all training data while maintaining a balance between model complexity and prediction error. The regression function $f(x, w)$ in SVR is given by (18):

$$f(x, w) = \sum_{i=1}^N w_i \phi(x_i) + b \tag{18}$$

where $\phi(x_i)$ is a non-linear transformation of the input x_i into a higher-dimensional space, w is the weight vector, and b is the bias term. This model aims to minimize the following objective (19):

$$R(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \mathcal{L}_\epsilon(y_i - f(x_i, w)) \tag{19}$$

where $\frac{1}{2} \|w\|^2$ is the regularization term that controls the model complexity, C is a regularization parameter that determines the trade-off between the model’s complexity and its tolerance to deviations, and (\mathcal{L}_ϵ) is the loss function with an ϵ -insensitive zone, defined as (20):

$$\mathcal{L}_\epsilon(y_i - f(x_i, w)) = \begin{cases} 0 & \text{if } |y_i - f(x_i, w)| \leq \epsilon \\ |y_i - f(x_i, w)| - \epsilon & \text{otherwise.} \end{cases} \tag{20}$$

This loss function ensures that deviations within ϵ are ignored, focusing only on larger errors.

Minimizing R is equivalent to minimizing (21):

$$R_{w, \zeta, \zeta^*} = \frac{1}{2} \left[\|w\|^2 + C \left(\sum_{i=1}^l \zeta + \sum_{i=1}^l \zeta^* \right) \right], \tag{21}$$

where ζ and ζ^* are the slack variables, which are shown in Figure 6.

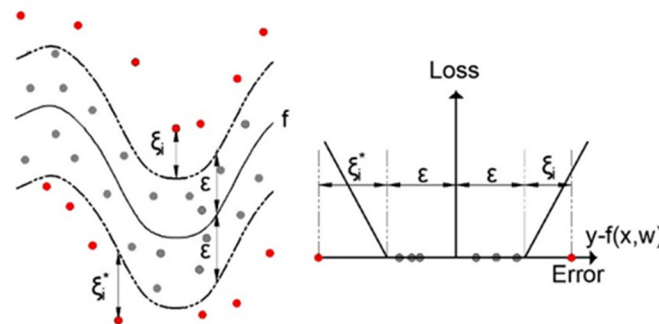


Figure 6. Nonlinear support vector regression with ϵ -insensitivity zone [16].

The optimization problem in SVR is typically solved using Lagrange multipliers, transforming the problem into its dual form. The dual form allows the incorporation of kernel functions, which can compute inner products in the high-dimensional feature space without explicitly mapping the input vectors to that space. This is crucial for handling non-linear relationships within the data.

The dual optimization problem can be expressed as (22):

$$R(\alpha, \alpha^*) = \sum_{i=1}^l y_i (\alpha_i^* - \alpha_i) - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) K(x_i, x_j) - \epsilon \sum_{i=1}^l (\alpha_i^* + \alpha_i) \tag{22}$$

subject to the constraints (23):

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, 0 \leq \alpha_i, \alpha_i^* \leq C \tag{23}$$

where α_i and α_i^* are Lagrange multipliers, $K(x_i, x_j)$ is the kernel function that computes the dot product in the feature space without explicitly performing the transformation. The kernel function $K(x_i, x_j)$ can take various forms which facilitate the mapping of input data into higher-dimensional spaces suitable for separation.

The prediction function for new data points x in SVR is then given by (24):

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*)K(x_i, x) + b \tag{24}$$

Commonly used kernel functions in SVR include:

- Linear kernel $K(x_i, x_j) = x_i^T x_j$
- Sigmoid kernel: $K(x_i, x_j) = \tanh(\gamma \langle x_i, x_j \rangle + r), \gamma > 0$
- Radial Basis Function (RBF) kernel $K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2)$.

The performance of the SVR model depends heavily on selecting appropriate hyper parameters, such as: regularization parameter C , ϵ , kernel-specific parameters, where for instance, γ in the RBF kernel, which controls the width of the Gaussian function etc. Tuning these hyper parameters is crucial for ensuring the model generalizes well to unseen data. This is typically achieved using grid search techniques to identify the optimal values for C , ϵ , and any kernel-specific parameters.

2.5. Gaussian Process Regression (GPR)

Gaussian Process Regression (GPR) is a Bayesian regression technique that offers a probabilistic framework for modeling the relationship between input features and continuous output variables. One of the significant advantages of GPR is its ability to quantify uncertainty in predictions [28].

In GPR, it is assumed that the underlying function mapping inputs to outputs is sampled from a Gaussian Process. A Gaussian Process is characterized as a collection of random variables where any finite number of these variables have a joint Gaussian distribution.

Consider the nonlinear regression problem (25):

$$y = f(x) + \epsilon, \quad \epsilon \sim N(0, \sigma^2). \tag{25}$$

Here, $f(\cdot)$ is an unknown function mapping the input x to the output y , and ϵ represents normally distributed noise with mean zero and variance σ^2 . GPR assumes that the unknown function $f(\cdot)$ follows a Gaussian Process, which is specified by a mean function $\mu(\cdot)$ and a covariance function (or kernel) $k(\cdot, \cdot)$.

Given a dataset with n observations, $y = \{y_1, \dots, y_n\}$, these observations are modeled as a sample from a multivariate Gaussian distribution (26):

$$(y_1, \dots, y_n)^T \sim N(\boldsymbol{\mu}, K), \tag{26}$$

where:

- $\boldsymbol{\mu} = (\mu(x_1), \dots, \mu(x_n))^T$ is the vector of mean values.
- K is the covariance matrix, where the element $K_{ij} = k(x_i, x_j) + \sigma^2 \delta_{ij}$, with δ_{ij} being the Kronecker delta function.

For a new test input x^* , GPR predicts its corresponding output y^* . The joint distribution of both the training data outputs y and the new output y^* is a multivariate Gaussian distribution (27):

$$(y_1, \dots, y_n, y^*) \sim N(\mu^*, \Sigma) \tag{27}$$

where:

- $\mu^* = (\mu(x_1), \dots, \mu(x_n), \mu(x^*))^T$ is the mean vector.
- Σ is the covariance matrix, which can be divided into blocks (28):

$$\Sigma = \begin{bmatrix} K & K^* \\ K^{*T} & K^{**} \end{bmatrix} \tag{28}$$

In this matrix:

- $K^* = (K(x^*, x_1), K(x^*, x_2), \dots, K(x^*, x_n))^T$ is the covariance between the test point and training points.
- $K^{**} = K(x^*, x^*)$ is the variance at the test point.

The conditional distribution of y^* given the training data $y = (y_1, \dots, y_n)^T$ is also Gaussian, with mean and variance values, respectively (29), (30):

$$\hat{y}^* = \mu(x^*) + K^{*T} K^{-1} (y - \mu), \tag{29}$$

$$\hat{\sigma}^{*2} = K^{**} + \sigma^2 - K^{*T} K^{-1} K^*. \tag{30}$$

The covariance function (kernel) plays a key role in defining the properties of the functions that the GP can model. A commonly used kernel is the Squared Exponential (SE) kernel with Automatic Relevance Determination (ARD) (31):

$$k(x_p, x_q) = v^2 \exp \left[-\frac{1}{2} \sum_{i=1}^n \left(\frac{x_p^i - x_q^i}{r_i} \right)^2 \right]. \tag{31}$$

In this equation r_i controls the length scale for each input dimension, indicating how much influence each dimension has on the prediction. A large r_i means the corresponding input dimension is less important.

The parameters $\{v, r_1, \dots, r_n\}$ and the noise variance σ^2 are called hyper parameters. These can be optimized by maximizing the log marginal likelihood of the observed data (32):

$$L(v, r_1, \dots, r_n, \sigma^2) = -\frac{1}{2} \log \det K - \frac{1}{2} y^T K^{-1} y - \frac{n}{2} \log 2\pi. \tag{32}$$

2.6. Artificial Neural Networks

A Multi-Layer Perceptron (MLP) is a type of feedforward neural network designed for supervised learning tasks (Figure 7).

Its architecture consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of neurons that receive signals, process them through weighted synaptic connections, and apply a non-linear activation function to generate an output [16,29].

The input layer receives the data, which is then propagated through the network. Each neuron in the hidden layers processes the inputs by computing a weighted sum of its inputs, adding a bias term, and applying a non-linear activation function, such as the hyperbolic tangent function. The output layer neurons typically use a linear activation function when the model is applied to regression tasks.

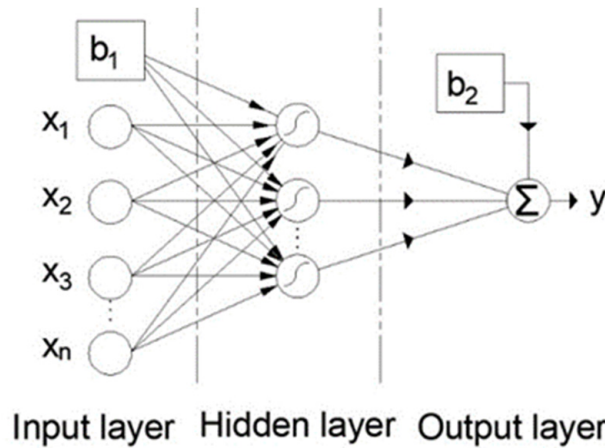


Figure 7. Multi-layer perceptron (MLP) model [16].

Mathematically, the activation of a neuron y_k a hidden layer can be expressed as (33):

$$y_k = \phi \left(\sum_{j=1}^m w_{kj}x_j + b_k \right) \tag{33}$$

where x_1, x_2, \dots, x_m are the input variables, $w_{k1}, w_{k2}, \dots, w_{km}$ represent the weights for neuron k , b_k is the bias term, and ϕ is the activation function.

The maximum number of neurons in hidden layer N_H was selected as the upper limit, and various network architectures were explored, beginning with a single neuron in the hidden layer. Subsequent architectures incrementally increased the number of neurons up to the maximum value determined by the following Equation (34) [16]:

$$N_H \leq \min \left(2N_I + 1, \frac{N_S}{N_I + 1} \right) \tag{34}$$

where:

- N_I —Number of inputs,
- N_S —Number of samples.

The choice of activation function plays a pivotal role in the learning capacity of the MLP. In this model, a hyperbolic tangent (tanh) activation function is employed in the hidden layer neurons. For the output layer, a linear activation function is utilized, which is suitable for regression problems.

The MLP model can be trained using the Levenberg-Marquardt (LM) algorithm, where weights are updated iteratively through gradient descent. The loss function, the mean squared error (MSE), is minimized during training to improve the accuracy of the model.

3. Mean-Based Shapley Value Analysis for Feature Importance in Machine Learning Models

The Shapley method, originating from game theory, is employed to assign “credit” or contribution to each “player” (in the context of machine learning, each feature or variable) within a “cooperative” game (the predictive model). In game theory, the contribution of each “player” to the final outcome of the game is analyzed. Similarly, in machine learning, the “game” is defined as the model’s prediction, while the “players” are represented by the input features [30–32].

Using the Shapley method, every conceivable combination (or subset, including the empty subset) of features, excluding a specific feature f , is examined. Changes in the

model's outcome are then observed when the feature f is added to each subset. For each combination S (a subset of features), the following difference is computed [30–32] (35):

$$\text{Contribution} = \text{Model}(S \cup \{f\}) - \text{Model}(S) \quad (35)$$

The core principle of the Shapley method involves averaging these marginal contributions across all subsets. The total sum of contributions is divided by 2^{n-1} , where n denotes the total number of features. This averaging process yields the Shapley value for feature f , which quantifies, on average, the impact of that feature on the model's outcome, accounting for all possible subsets of other features.

4. Dataset

The database consists of 855 test results collected from 38 research studies [33–70], focusing on the key parameters that influence bond strength [4]. The individual references on which the database was based are detailed in the Table A1. The single-lap shear test (Figure 8), recognized as one of the most effective methods for evaluating the bond strength at the FRP–concrete interface, forms the foundation of this study.

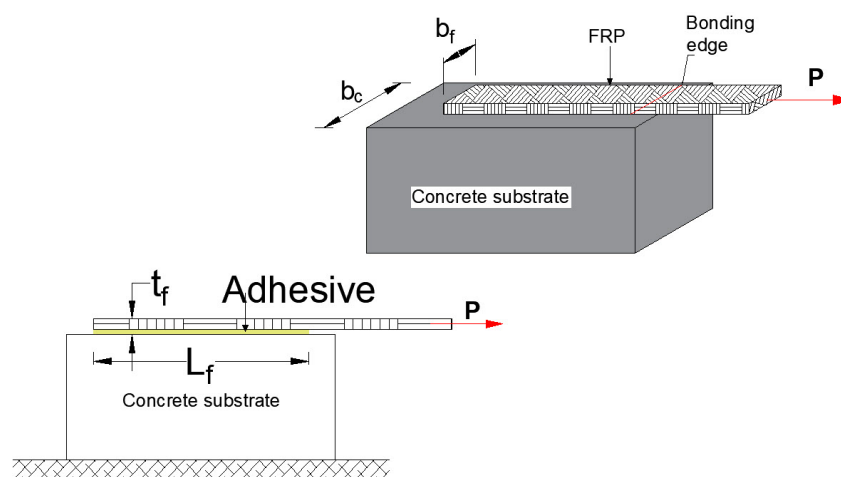


Figure 8. Single-lap shear bond test configuration.

The collected dataset includes critical material properties and geometric parameters. Material properties encompass the compressive strength of a concrete cylinder (f'_c), and the elastic modulus of FRP sheets. Geometric parameters include the thickness (t_f) and width (b_f) of the FRP sheet, the bond length (L_f) of the FRP sheet, and the width of the concrete substrate (b_c). The compressive strength of concrete cubes samples (f_c) was converted to cylinder strength (f'_c).

The filtered dataset was divided into a training set (70% of the data) for model development and a test set (30% of the data) for assessing model performance. Depending on the model, MSE (Mean Squared Error) or RMSE was used as the primary metric during training. Model accuracy on the test set was evaluated using four metrics: two absolute measures (RMSE and MAE) and two relative measures (MAPE and the correlation coefficient R). The division of the complete dataset for ML models should be conducted in a manner that ensures similar statistical characteristics across subsets (Tables 1 and 2).

This approach aims to create training, validation, and test sets with comparable distributions and statistical indicators, such as mean, standard deviation, and data range, thereby preserving data integrity and enhancing the model's ability to generalize effectively (Tables 1 and 2).

Table 1. Statistical summary and descriptive characteristics of variables in the training set.

	b_c (mm)	f'_c (MPa)	E_f (GPa)	t_f (mm)	b_f (mm)	L_f (mm)	P_u (kN)
min	80.00	8.00	22.50	0.08	10.00	20.00	2.40
max	500.00	74.67	425.10	4.00	150.00	400.00	56.50
average	144.31	39.38	203.66	0.50	57.62	175.42	17.80
mode	150.00	48.56	230.00	0.17	50.00	100.00	11.90
median	150.00	36.50	230.00	0.17	50.00	150.00	15.73
std	56.93	15.23	77.97	0.53	26.57	102.31	10.13

Table 2. Statistical summary and descriptive characteristics of variables in the testing set.

	b_c (mm)	f'_c (MPa)	E_f (GPa)	t_f (mm)	b_f (mm)	L_f (mm)	P_u (kN)
min	56.93	8.00	22.50	0.08	10.00	20.00	2.40
max	500.00	74.67	425.10	4.00	150.00	400.00	56.50
average	144.84	38.94	208.34	0.53	57.50	180.23	18.55
mode	150.00	65.73	230.00	0.17	50.00	100.00	12.75
median	150.00	36.27	230.00	0.17	50.00	162.50	16.47
std	68.25	15.03	75.95	0.58	25.21	105.20	9.85

To rigorously evaluate the representativeness of the training and testing datasets, three statistical methods were applied: the Kolmogorov-Smirnov (KS) test, Kullback-Leibler (KL) Divergence, and a permutation test.

The KS test, a non-parametric method comparing the empirical cumulative distribution functions of two samples (Massey, 1951), yielded results ($h = 0$, $p = 0.9153$) that support the null hypothesis, indicating no significant difference between the distributions. This finding suggests that the random splitting process successfully preserved the original data's statistical properties [70].

The Figure 9 displays the empirical cumulative distribution functions (CDFs) for both the training and testing sets. Visually, the two curves (blue for training and red for testing) are nearly indistinguishable, following each other closely across the entire range of observed data values. This close alignment indicates that there is no substantial difference in how the data are distributed between the two subsets.

The KL Divergence, a measure of relative entropy between two probability distributions (Kullback & Leibler, 1951), produced a value of 0.0014, reflecting minimal divergence and thus a near-identical distributional structure between the training and testing sets [71]. In the displayed Figure 10, the training and testing datasets probability density functions (PDFs) nearly overlap across the range of data values. Such close alignment between the two curves indicates that both datasets share remarkably similar distributional properties. The minimal differences you might observe—such as subtle shifts in peak height or slight variations in tail regions—appear minor and do not fundamentally alter the shape or location of the distributions. This strong similarity is consistent with a low Kullback-Leibler (KL) divergence value, which quantifies how one probability distribution deviates from another. A near-zero KL divergence confirms that there is almost no “information loss” when one distribution is used to approximate the other, reinforcing the conclusion that the training and testing datasets originate from the same underlying population distribution.

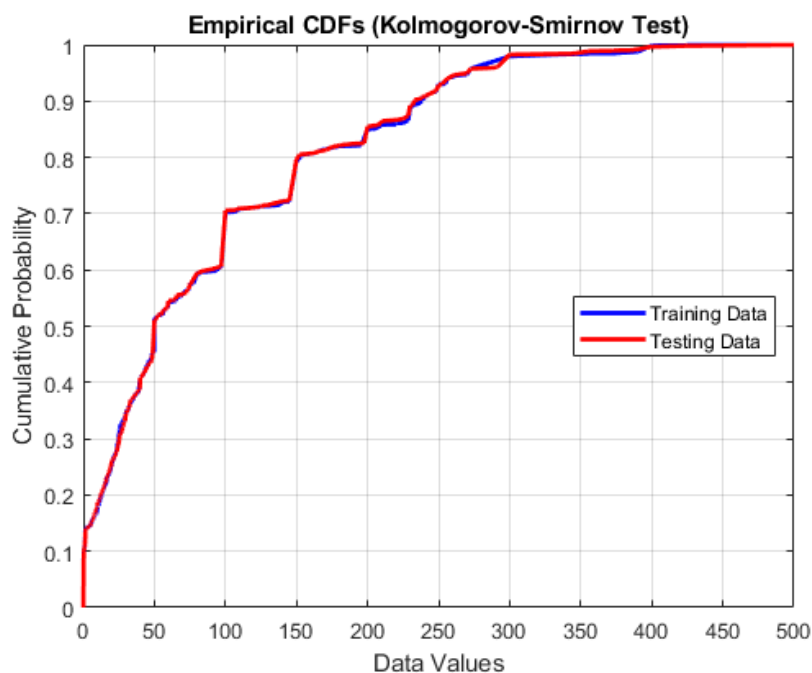


Figure 9. Empirical cumulative distribution functions (CDFs) of training and testing data (Kolmogorov-Smirnov test).

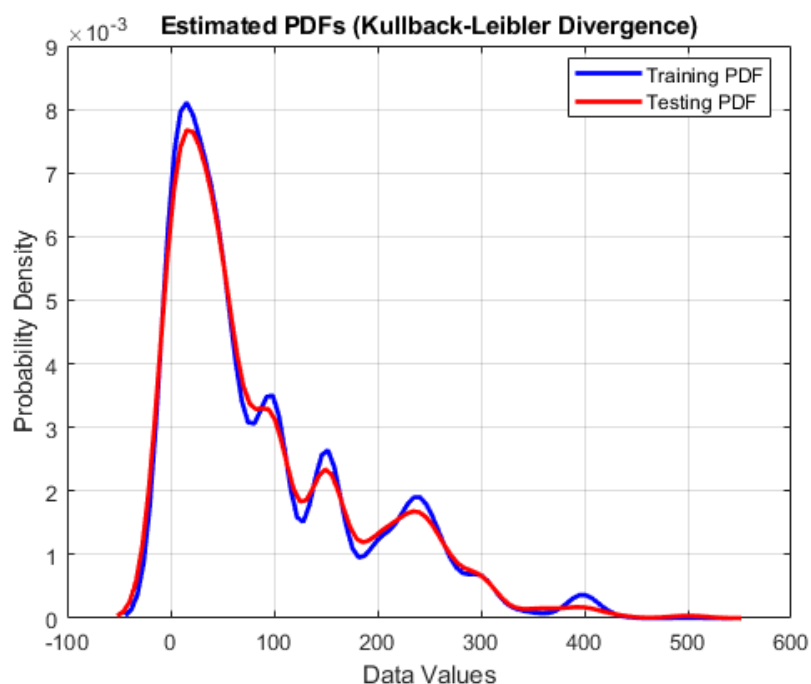


Figure 10. Estimated probability density functions (PDFs) for training and testing data (Kullback-Leibler divergence).

To further bolster these conclusions, a permutation test with 100,000 permutations returned a p -value of 0.7202, providing additional evidence that any observed differences are likely due to random variation rather than systematic bias (Good, 2000). Taken together, the outcomes of these tests strongly validate the representativeness of the testing dataset, ensuring that subsequent model training and evaluation processes can be conducted with confidence in the statistical fidelity of the data [72].

In Figure 11, the observed difference line is positioned in a region where there is still a substantial amount of probability mass, indicating that the observed difference is well

within the main distribution of permuted differences. This suggests that the observed difference is not situated in the extreme tail of the distribution; instead, it falls into a relatively common or “central” area where many of the permuted mean differences lie. As a result, Figure 11 supports the interpretation that the observed difference is not statistically unusual. This visual assessment is consistent with the test’s high p -value, which quantitatively indicates that observed differences of this magnitude occur frequently under the null hypothesis (i.e., when the two distributions are effectively the same).

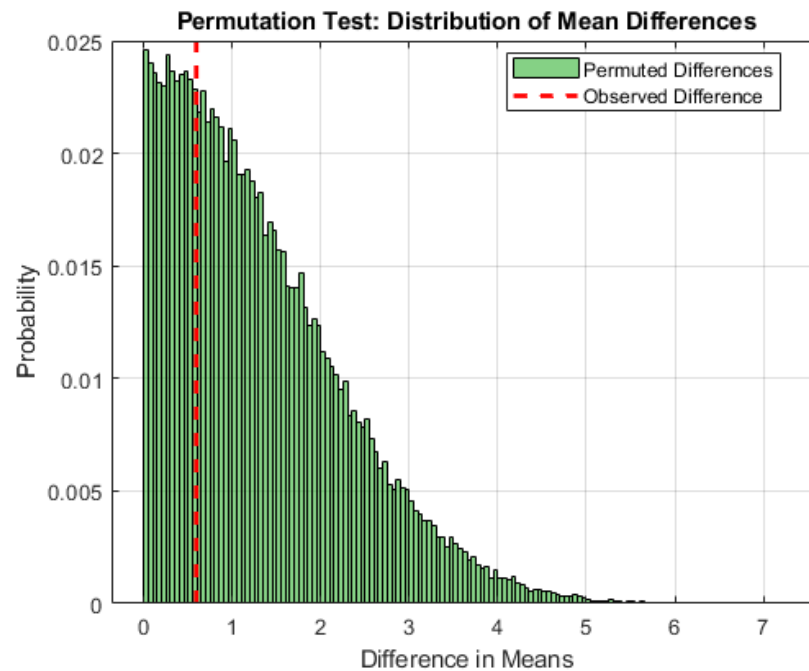


Figure 11. Distribution of mean differences from permutation resampling (permutation test).

5. Results and Discussion

A linear regression framework, enhanced not only with main effects but also with various interaction and polynomial terms, was employed to capture the complexities in bond strength prediction.

Interaction terms, such as $E_f \cdot t_f$, $b_c \cdot b_f$, $t_f b_f$ allow us to examine how pairs of variables jointly influence bond strength, capturing multi-variable dependencies essential when one variable’s effect is modified by another.

Polynomial terms (squares of the variables), like b_c^2 , f_c^2 , E_f^2 etc. addressed nonlinear relationships between predictors and bond strength, enabling the model to fit complex trends while retaining the interpretability of linear regression.

By broadening the model to include these interaction and polynomial terms, we achieved improved predictive accuracy, as validated by RMSE, MAE, MAPE, and R metrics.

Table 3 presents the estimated coefficients, standard errors, t-statistics, and p -values for each parameter in the developed linear regression model. The most significant predictors of bond strength are the interaction between FRP thickness and sheet width ($t_f b_f$), the interaction between FRP elastic modulus and thickness ($E_f t_f$), FRP elastic modulus (E_f), and concrete compressive strength (f_c'), along with their interactions.

Polynomial terms indicate that the relationship between certain variables and bond strength is not strictly linear, with diminishing effects observed at higher values.

Performance metrics including RMSE, MAE, MAPE, and the correlation coefficient (R) were calculated to assess the model’s (defined by Equation (40)) accuracy.

Table 3. Detailed regression results: estimates, standard errors, t-statistics, and p-values for model parameters.

Parameter	Estimate	Standard Error	tStat	p Value
(Intercept)	−15.9200	2.8688	−5.5492	4.3568×10^{-8}
b_c	−0.0393	0.0138	−2.8525	0.0045
f'_c	0.3682	0.0702	5.2473	2.1625×10^{-7}
E_f	0.0695	0.0098	7.0878	3.9461×10^{-12}
b_f	0.0654	0.0476	1.3739	0.1700
L_f	0.0574	0.0112	5.1353	3.8429×10^{-7}
$E_f t_f$	0.0443	0.0054	8.2612	9.6963×10^{-16}
$b_c b_f$	0.0014	0.0002	6.0212	3.0620×10^{-9}
$t_f b_f$	0.1313	0.0145	9.0761	1.7139×10^{-18}
$f'_c L_f$	−0.0005	0.0001	−3.9565	8.5421×10^{-5}
b_c^2	−0.0001	0.0001	−2.4285	0.0155
$f_c'^2$	−0.0025	0.0007	−3.3629	8.2189×10^{-4}
E_f^2	−0.0001	0.0000	−4.3966	1.3068×10^{-5}
b_f^2	−0.0009	0.0002	−3.8854	1.1386×10^{-4}
L_f^2	−0.00007	0.0000	−3.4915	5.1660×10^{-4}

The obtained regression equation is defined by the following expression (36):

$$P_u = -15.92 - 0.0393 b_c + 0.3682 f'_c + 0.0695 E_f + 0.0654 b_f + 0.0574 L_f + 0.0443 E_f t_f + 0.0014 b_c b_f + 0.1313 t_f b_f - 0.0005 f'_c L_f - 0.0001 b_c^2 - 0.0025 f_c'^2 - 0.0001 E_f^2 - 0.0009 b_f^2 - 0.00007 L_f^2 \tag{36}$$

In this study, a Multigene Genetic Programming (MGGP) approach was implemented to analyze the impact of key parameters, including the number of genes (1–6) and tree depths (1–6), on model complexity and performance. The method explored a wide range of configurations to find the optimal balance between complexity and predictive accuracy.

The MGGP process was repeated ten times to account for randomness in model initialization, with each iteration starting from a unique set of random parameters, resulting in a distinct model at the end of each run. All runs were then merged at the end to consolidate the results.

Table 4 summarizes the key parameters. These settings were chosen to enable extensive exploration of model structures while controlling computational complexity.

Table 4. The parameters setting for MGGP models.

Parameter	Setting
Function set	times, minus, plus, rdivide, square, exp, log, mult3, sqrt, cube, power
Population size	From 100 to 1000 with step 100
Number of generations	1000
Max number of genes	6
Max tree depth	6
Tournament size	2
Elitism	0.05% of population
Crossover probability	0.84
Mutation probability	0.14
Probability of Pareto tournament	0.70

During training, two primary objectives were prioritized: minimizing RMSE and reducing model complexity. These objectives formed a multi-objective function to balance prediction accuracy (weight: 0.7) with interpretability (weight: 0.3). RMSE quantified how

well predictions aligned with actual data, while complexity ensured models remained interpretable and practical.

The evolutionary process spanned 100 generations, with an initial population size of 100 to explore diverse solutions. The initial population was tested ranging from 100 to 1000 individuals, with increments of 100. This process was repeated across ten independent runs, generating multiple models. The best models from these runs were merged into a final population representing the most promising solutions.

Model selection emphasized strong predictive performance (high correlation coefficients) and low complexity. The selected models formed the Pareto front (Figure 12), a set of optimal solutions where no single objective (accuracy or complexity) can be improved without compromising the other. This ensured the final models were both effective and interpretable.

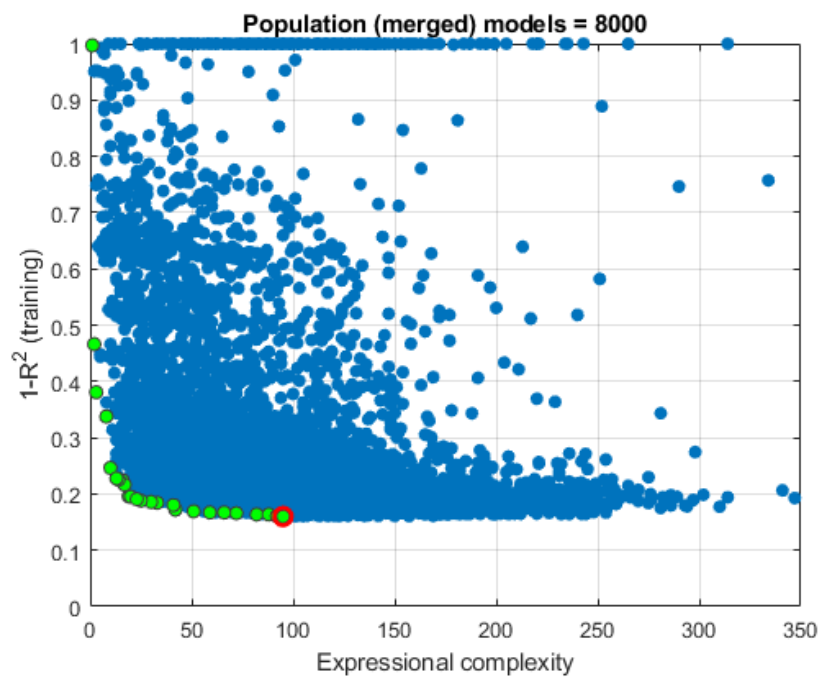


Figure 12. Graphic representation of the Pareto front for the initial population of 800 models (individuals): The models of the Pareto front are marked with green circles, and the optimal model is marked with a red circle.

Key parameters of the evolutionary process included a tournament size of 2, an elitism rate of 0.05%, a crossover probability of 0.84, and a mutation probability of 0.14. These settings balanced exploration and exploitation while maintaining population diversity and preserving high-quality solutions.

The Pareto front was defined using RMSE and R² values, with the optimal model highlighted by GPTIPS 2.0 software [73]. Further analysis of models along the Pareto front considered additional criteria, including MAE, MAPE, and R, to identify models with both high performance and lower complexity.

The analytical expression of the optimal Pareto front model (Model ID 7961) is defined by the following Equation (37):

$$\sigma = b + \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4 = 0.53 + \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4 + \sigma_5 + \sigma_6 \tag{37}$$

Analytical expressions for individual genes are given in Table 5.

Table 5. Analytical expressions for the genes of the optimal model of the Pareto front for the initial population of 800 models.

$\sigma_1 = - (0.408 \cdot b_c) \frac{f_c}{f_c + 2.0 \cdot t_f}$	$\sigma_2 = \frac{-6.54 \times 10^{-6} \cdot b_f^2}{t_f^2}$
$\sigma_3 = 0.00141 \cdot b_c \cdot b_f$	$\sigma_4 = -1.26 \times 10^{-4} \cdot b_c - 1.26 \times 10^{-4} (b_c + b_f + t_f)^2$
$\sigma_5 = - \frac{0.595 \cdot b_c \cdot b_f^2 \cdot t_f}{f_c + 2.0 \cdot L_f}$	$\sigma_6 = 0.246 \cdot (E_f \cdot b_f \cdot t_f)^{\frac{1}{2}}$

Each of the symbolic analytical expressions or genes listed in Table 5 is also structurally represented in the form of a tree in Figure 13.

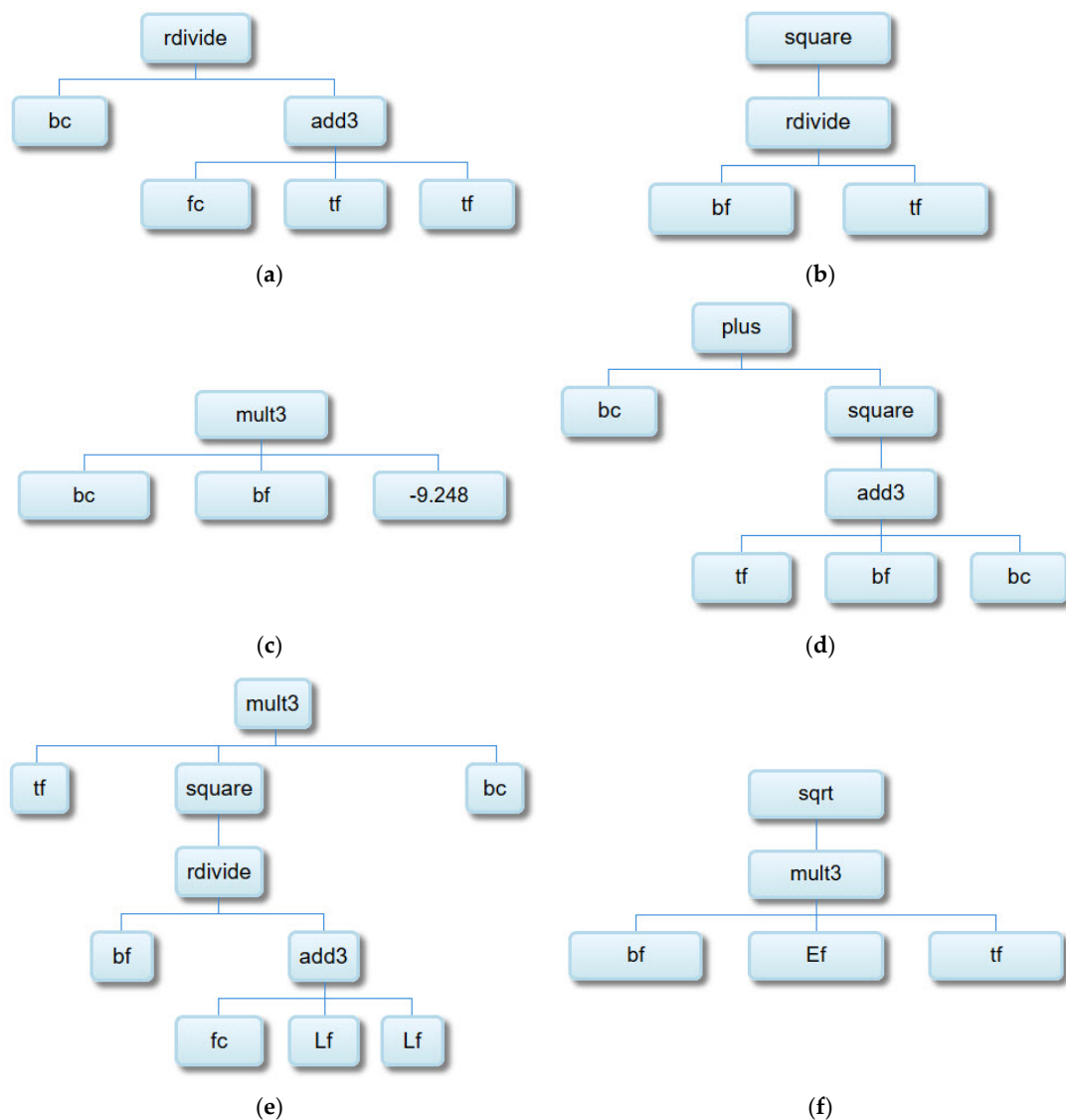


Figure 13. Tree structure representation of genes from the optimal MGGP model: (a) Gene 1, (b) Gene 2, (c) Gene 3, (d) Gene 4, (e) Gene 5 and (f) Gene 6.

From Table 6, it can be observed that the optimal model from the set of non-inferior solutions, with a complexity of 95, has very similar accuracy to the model labeled as Model ID 6959, which has a complexity of 42.

Table 6. Accuracy criterion values for 10 models on the Pareto front.

Model ID	Model Complexity	RMSE	MAE	MAPE/100	R
7961	95	4.4436	3.4294	0.2167	0.9147
7570	92	4.6349	3.6170	0.2417	0.9079
1766	82	4.8264	3.7418	0.2456	0.9004
3867	88	4.8137	3.7101	0.2404	0.9004
7161	72	4.5985	3.5375	0.2362	0.9095
7164	66	4.6936	3.6328	0.2407	0.9056
7167	59	4.6545	3.5922	0.2383	0.9056
6726	51	4.6894	3.5855	0.2383	0.9059
6959	42	4.6829	3.5373	0.2292	0.9061
7292	41	4.9642	3.7915	0.2514	0.8941

The relationship between the accuracy of the obtained model that consists of a population of 800 individuals and two parameters, namely the Number of genes and the Depth of tree, is illustrated in Figure 14.

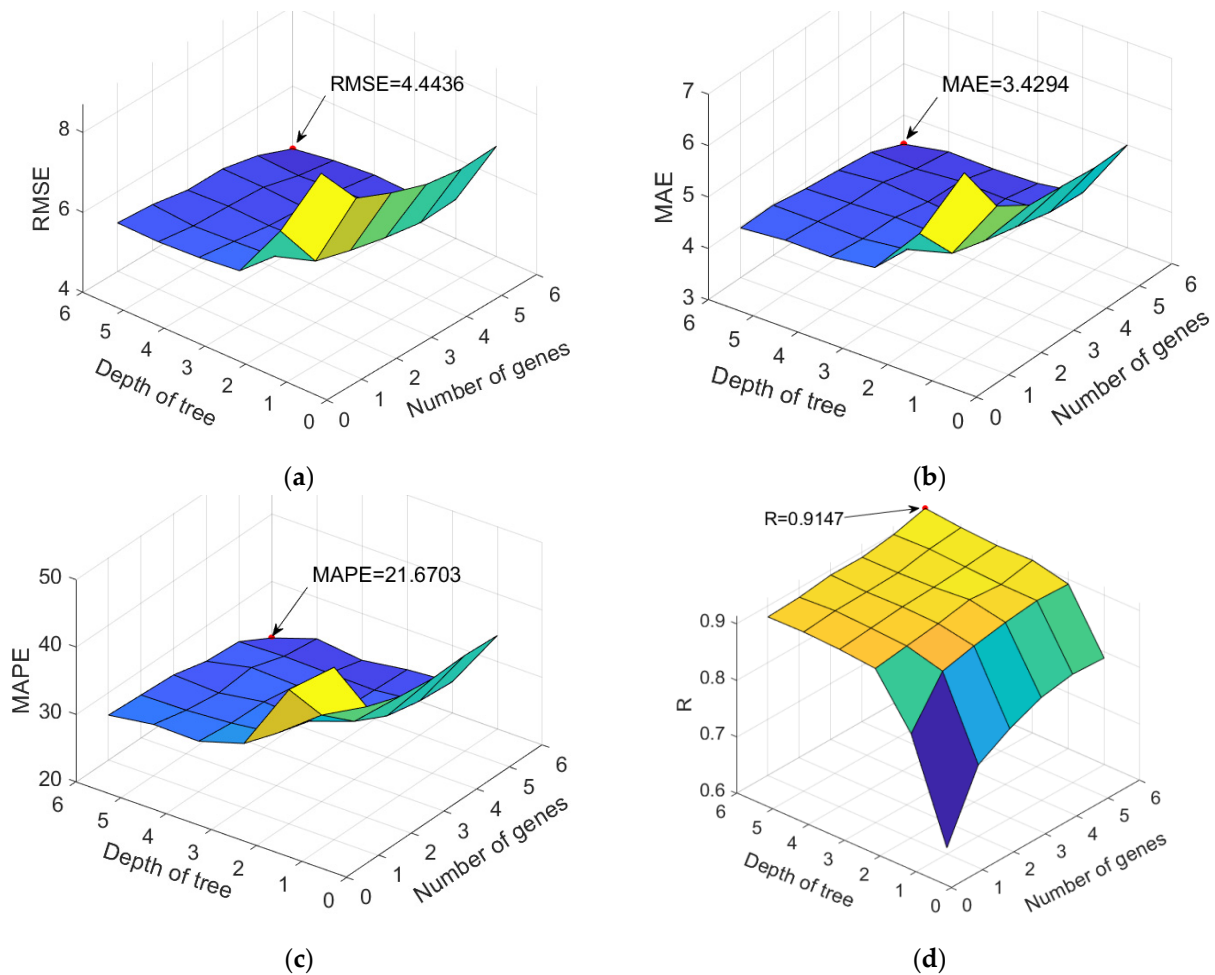


Figure 14. Dependence of accuracy criteria on hyper parameters for the MGGP model: (a) RMSE, (b) MAE, (c) MAPE and (d) R.

By employing the MGGP model through the GPTIPS software [15,73], models were developed that retained only the variables significant to the problem, as determined by evolutionary processes. During this process, irrelevant variables were typically eliminated

to enhance model simplicity and focus. However, in this analysis, all six input variables were found to significantly contribute to the model’s accuracy and were therefore retained in the final formulation.

Determining each gene’s contribution in the model is essential for increasing transparency and understanding how specific interactions among variables drive predictions. By quantifying each gene’s impact, one can identify which relationships are most influential, focus on refining the portions of the model that matter most, and efficiently allocate experimental or engineering resources.

The calculation of each term’s contribution in the given model is based on the decomposition of a complex regression model into its individual components $\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6$ and a bias term. The theoretical framework for calculating and analyzing the contribution of each term is as follows (38):

$$\text{Contribution of } \sigma_i = |\sigma_i| \tag{38}$$

The absolute value ensures that both positive and negative contributions are considered for their magnitude. The mean contribution of each σ_i across all test samples N is given by (39):

$$\text{Mean Contribution of } \sigma_i = \frac{1}{N} \sum_{j=1}^N |\sigma_{i[j]}| \tag{39}$$

The percentage contribution of each term is calculated relative to the total magnitude of all terms, including the bias (40):

$$\text{Percentage Contribution of } \sigma_i = \frac{\text{Mean Contribution of } \sigma_i}{\sum_{j=1}^6 \text{Mean Contribution of } \sigma_j + |\text{Bias}|} \times 100 \tag{40}$$

When analyzing the contribution of genes to the model, the most significant genes are $\sigma_6, \sigma_3,$ and $\sigma_4,$ whose contributions dominate the overall prediction (Figure 15). These contributions are directly linked to the variables present in the genes as well as their mutual interactions. The combined contribution of the three aforementioned genes accounts for nearly 90%.

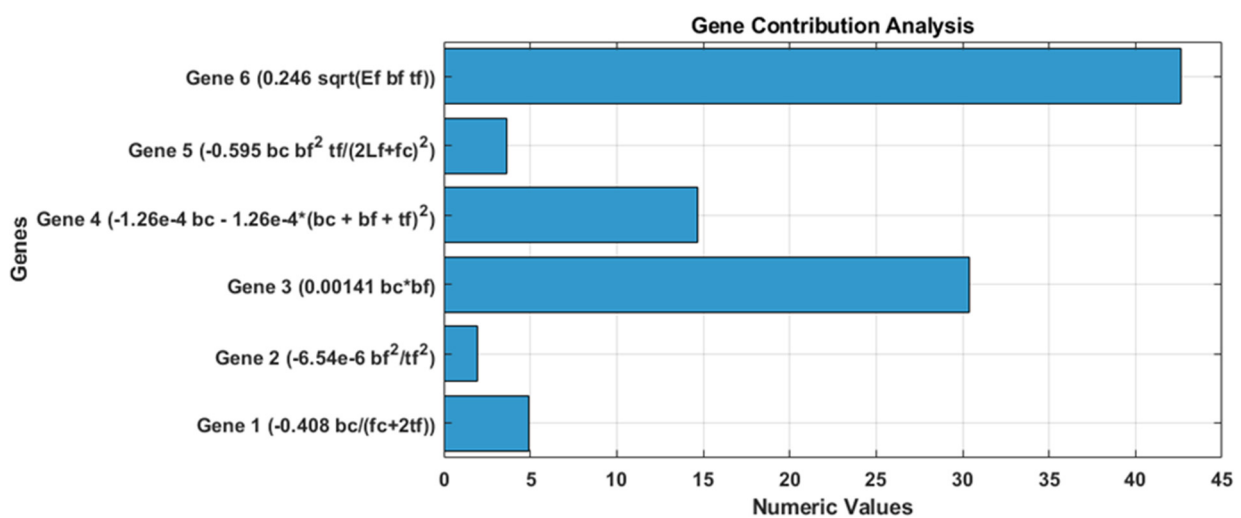


Figure 15. Contribution of individual gens.

Gene 6, with a percentage contribution of 42.64%, involves the interaction between $E_f, b_f,$ and t_f in square root form, suggesting that material properties (such as elasticity modulus and dimensions) play a key role in determining the model’s outcome. Gene 3,

with a percentage contribution of 30.40%, represents a simple interaction between two geometric dimensions, b_c and b_f , highlighting how the substrate width and material width jointly affect the predicted value. Gene 4, contributing 14.68%, includes a combination of the linear effect of b_c and the quadratic effect (nonlinear interactions between width and thickness) of geometric dimensions b_c, b_f, t_f .

In this study, the Shapley method was also implemented in the MGGP model by calculating the average prediction for each combination of active features on the test set, while inactive features were replaced with baseline values obtained as the mean of the training set. The difference between the average predictions when including versus excluding a specific feature represents the marginal contribution, and the Shapley value is derived by aggregating these contributions across all subsets (Figure 16).

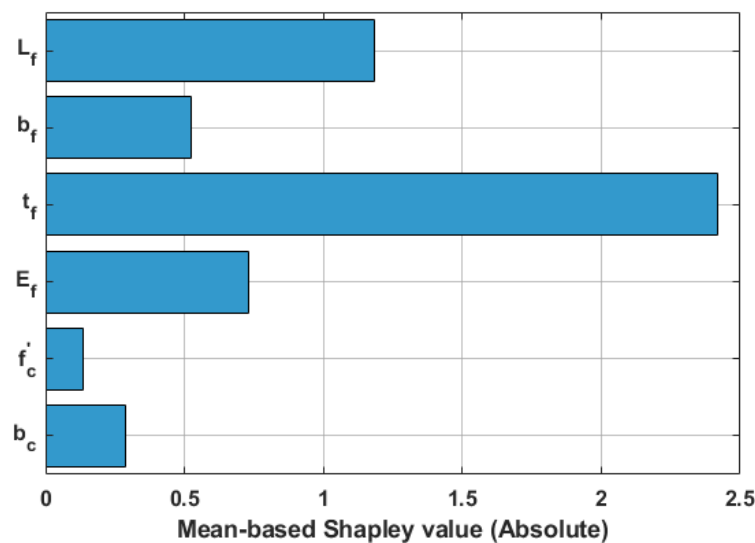


Figure 16. Significance of input variables for the MGGP model using Shapley values.

Based on the calculated average Shapley values, the most influential variables in the model are the thickness of the FRP strip (t_f), the bond length (L_f), and the elasticity modulus (E_f). The width of the FRP strip (b_f) and the substrate width b_c have a somewhat smaller influence. The least impactful variable is the compressive strength of concrete (f'_c). The relationship between variables and genes can be explained as follows: The variables t_f, L_f , and E_f dominate the predictions due to strong nonlinear and mechanical interactions, particularly evident in Gene 6. Meanwhile, b_f and b_c have a moderate influence through linear and nonlinear terms found in Gene 3 and Gene 4. The variable f'_c demonstrates the least importance in this model, suggesting that the geometric and mechanical characteristics of the FRP are more critical for predictions than the concrete's compressive strength.

In this research, a Gradient Boosted Trees (GBT) ensemble was employed, using a grid search to systematically optimize key parameters (Figure 17).

The grid search explored combinations of learning rates and tree depths to identify the best-performing configuration:

- Number of Trees (NumLearningCycles = 100): A fixed value of 100 trees was used to balance model complexity and prevent overfitting, enabling precise evaluation of learning rate and tree depth effects.
- Learning Rate (λ): Learning rates ranging from 0.001 to 1.0 were tested, with 0.1 providing the best trade-off between convergence speed and error minimization.
- Tree Depth (Max Number of Splits): Tree depths, represented by the maximum number of splits, were varied from 1 to 512. The optimal depth was calculated using $\log_2(n - 1)$ where n is the number of data points. Taking the logarithm of $n - 1$ for base 2 helps

determine the approximate number of splits required for full separation of the dataset. This ensured the trees remained appropriately deep relative to dataset size while preventing overfitting.

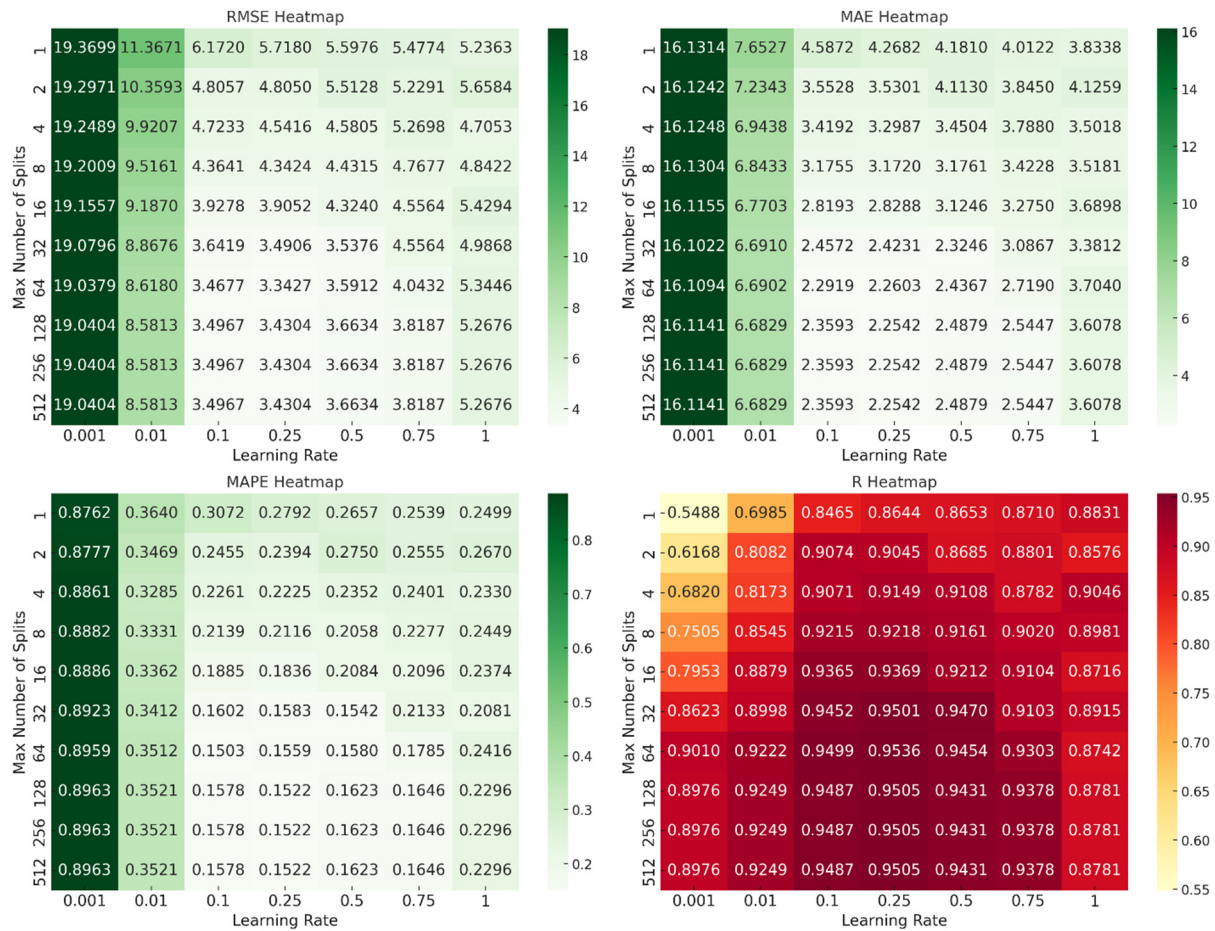


Figure 17. Dependence of the accuracy criteria on the hyper parameters of the boosted trees models.

The grid search identified 64 splits and a learning rate of 0.25 as the optimal combination, achieving the best RMSE of 3.3427 and the highest R-value of 0.9536. The best MAE of 2.2542 was achieved with 128 splits and a learning rate of 0.1, while the best MAPE of 0.1503 was observed with 64 splits and a learning rate of 0.1 (Figure 17).

Beyond 64 splits, further increases in tree depth yielded diminishing returns, as shown by the leveling off of RMSE and R-squared improvements. Therefore, the optimal number of splits was determined to be 64, where the model achieved the best overall performance across multiple metrics (Figure 17).

In this study, permutation importance was employed to determine the relative significance of input variables in the GBT model for predicting the output variable P_u . This method evaluates the impact of each variable by measuring the increase in prediction error when the variable’s values are randomly shuffled, disrupting its relationship with the target variable. Permutation importance is particularly effective in understanding variable contributions within complex, nonlinear models like gradient boosting.

However, permutation importance assumes feature independence, which may not always hold when predictors are correlated. To address this, a Variance Inflation Factor (VIF) analysis was conducted to evaluate multicollinearity among the input variables. VIF values quantify how much the variance of a regression coefficient increases due to multicollinearity. In this analysis, all VIF values were found to be between 1.08 and

1.78 (Figure 18), indicating very low multicollinearity. This confirms that the predictors are sufficiently independent to ensure robust and reliable results from the permutation importance method.

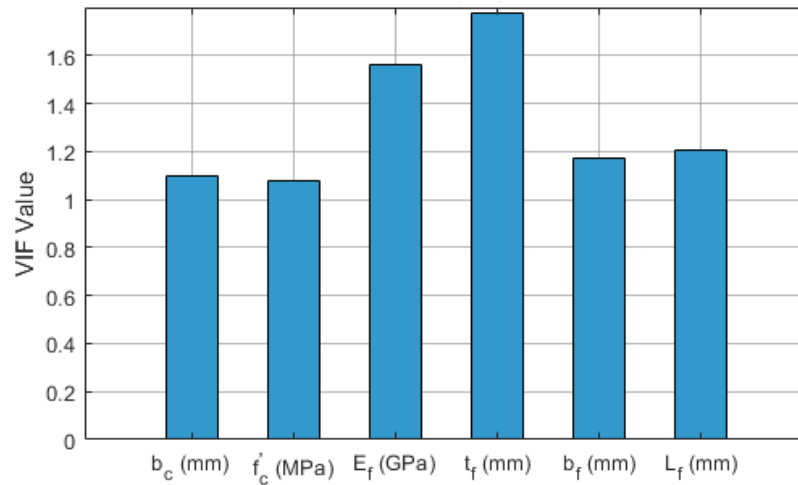


Figure 18. Variance inflation factor (VIF) values for input variables.

The results, illustrated in Figure 19, reveal that variables b_f (width of the FRP) and t_f (bond length) have the highest importance, significantly influencing the model’s accuracy. In contrast, variables such as b_c (concrete width) and f'_c (concrete compressive strength) have a lesser impact on the predictive power.

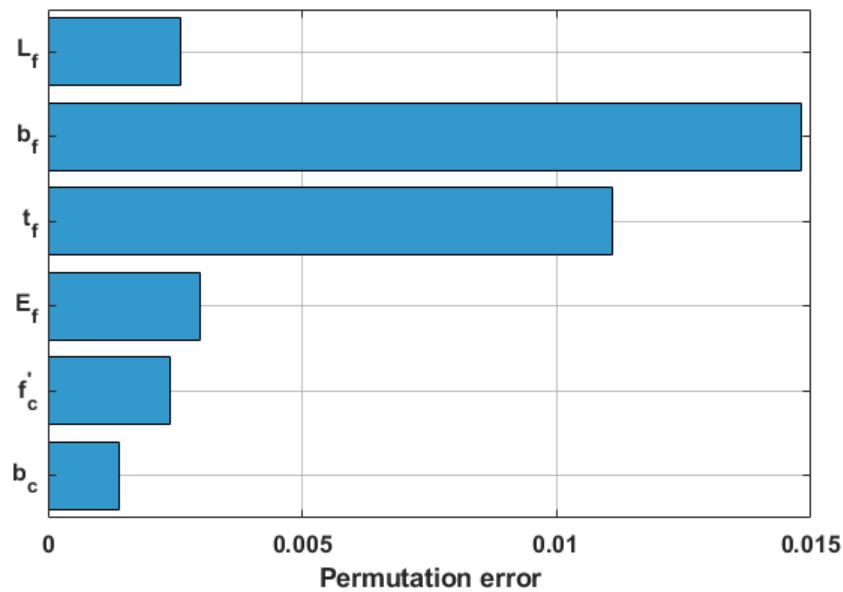


Figure 19. Significance of input variables for the boosted trees model using the permutation method.

Unlike permutation importance, Shapley analysis provides a robust framework that systematically evaluates every possible subset of features.

The absolute Shapley values confirm the dominance of L_f and t_f in model predictions. Geometric variables like b_c and b_f also play vital roles, while E_f and f'_c exhibit relatively lower contributions (Figure 20).

Bagging (Bootstrap Aggregation) was the primary method used to implement the ensemble model in this research. Depending on whether a subset of input variables or the entire set is used, a Random Forest (RF) or TreeBagger (TB) model was created, respectively.

This approach involves generating multiple decision trees (500 trees in this case) on different bootstrap samples of the data and averaging their predictions to enhance robustness and reduce variance.

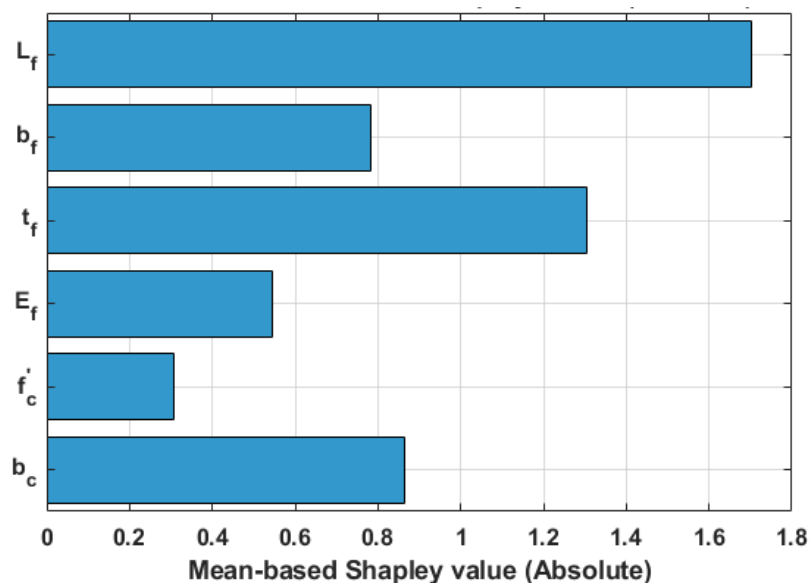


Figure 20. Significance of input variables for the boosted trees model using Shapley values.

The research specified 500 learning cycles, where 500 base decision trees were trained, and their predictions aggregated to form the final output. This large number of trees ensured model stability and allowed the capture of complex data patterns.

Two key hyperparameters were tuned during the training process:

Min leaf size: The minimum number of observations required to form a leaf in a decision tree. Smaller leaf sizes result in deeper trees, enabling the model to capture detailed patterns but increasing the risk of overfitting. In this study, the Min leaf size was varied from 1 to 10.

Number of variables: At each decision tree split, a subset of predictor variables was randomly selected for consideration. The number of variables sampled at each split was varied from 1 to 6.

Results across metrics (RMSE, MAE, MAPE, and R) showed that smaller Min Leaf Size values (1 and 2) consistently led to better performance (Figure 21). This indicates that deeper trees are necessary to capture the data's complexity. The best-performing models used 3 or 4 variables per split, striking a balance between simplicity and model complexity.

The best configurations identified are:

- Best RMSE: 3.7860, with Min leaf size = 1 and Number of variables = 3.
- Best MAE: 2.5821, with Min leaf size = 1 and Number of variables = 4.
- Best MAPE: 0.1778, with Min leaf size = 1 and Number of variables = 4.
- Best R-squared: 0.9415, with Min leaf size = 1 and Number of variables = 3.

The SVR model utilized radial basis function (RBF), linear, and sigmoid kernels, with grid search cross-validation applied to optimize its hyperparameters. Prior to training, all data were normalized to a range between 0 and 1, achieved by subtracting the minimum value and dividing by the range (max-min). Normalization was critical as the SVR optimization process is sensitive to input data scales.

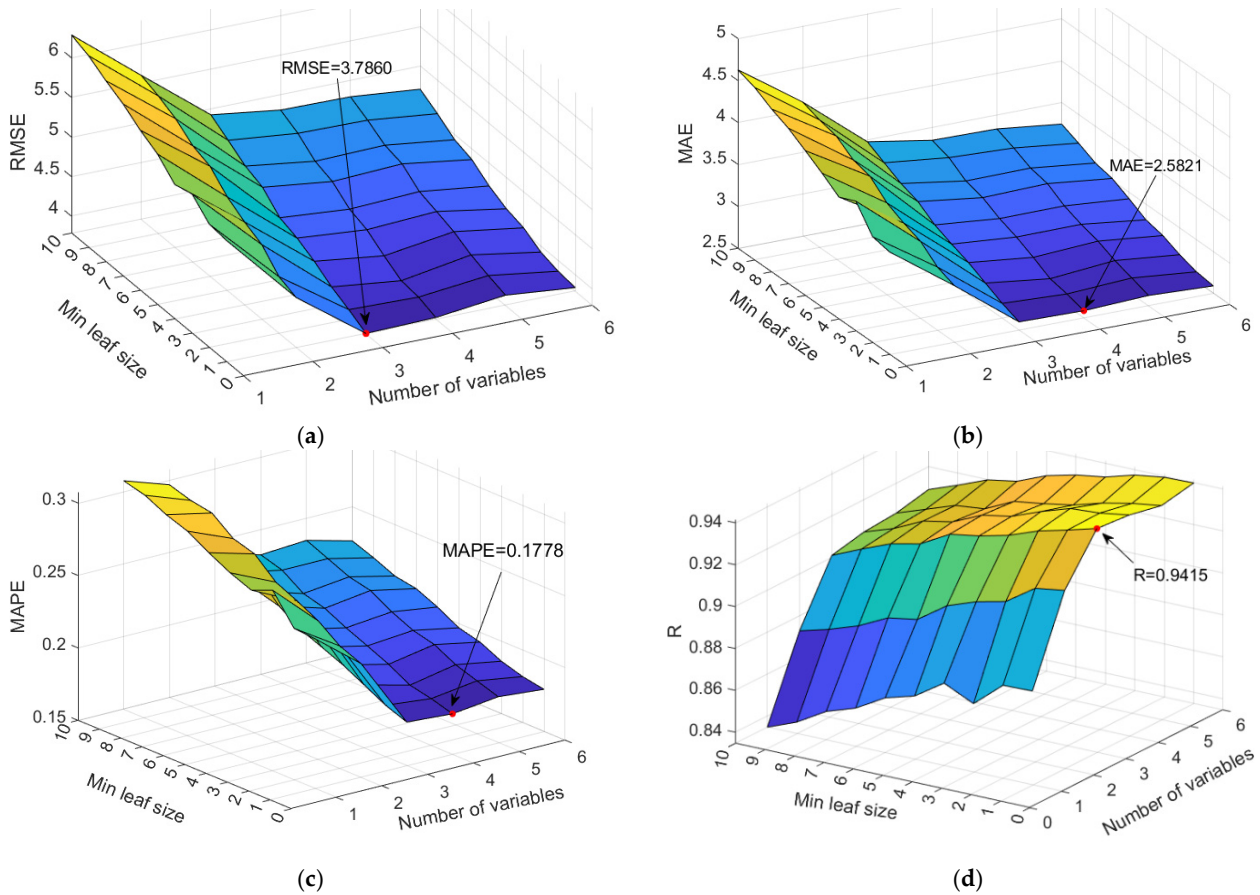


Figure 21. Dependence of the accuracy criteria on the hyper parameters of the Random Forest and TreeBagger models: (a) RMSE, (b) MAE, (c) MAPE and (d) R.

Grid search was used to determine the optimal values for three SVR hyperparameters:

- C (Cost/Regularization): Balances the trade-off between allowing slack variables (errors) and tightening the decision boundary. Higher values focus on correctly classifying training points but risk overfitting.
- Gamma (γ): Controls the influence of individual training points. Smaller gamma values imply a broader influence, while larger values indicate a more localized impact.
- Epsilon (ϵ): Defines the tolerance margin where no penalty is applied to prediction errors, controlling model sensitivity.

For the RBF kernel, coarse grid search varied $\log_2(C)$, $\log_2(\gamma)$, and $\log_2(\epsilon)$ from -5 to 5 in increments of 1. The mean squared error (MSE) was calculated for each combination on the test set to identify the best parameters. A finer grid search was subsequently applied around the best initial values with reduced step sizes, iteratively refining the model to minimize MSE further.

The same procedure was applied to the linear and sigmoid kernels. The optimal parameters for each kernel were:

- $C = 1.4513$; $\epsilon = 0.0043$; $\gamma = 20.7363$ for the RBF kernel;
- $C = 0.3208$ and $\epsilon = 0.0432$ for the linear kernel;
- $C = 23.6326$; $\epsilon = 0.0521$; $\gamma = 0.0118$ for sigmoid kernel.

The accuracy criterion values for the analyzed models are given in Table 7 (Optimal model values are bolded).

Table 7. Comparative analysis of results using linear, RBF, and sigmoid kernel in the SVR method.

Model	RMSE	MAE	MAPE/100	R
Lin. kernel	8.7154	6.6468	0.2105	0.7751
RBF kernel	4.9646	3.5352	0.1171	0.9332
Sig. kernel	8.7104	6.6094	0.2073	0.7718

In this research, various covariance functions were employed to design the Gaussian Process Regression (GPR) model, each offering unique strengths in capturing relationships between input variables and the target. The covariance functions explored included exponential, squared exponential, Matérn (with 3/2 and 5/2 degrees of smoothness), and rational quadratic kernels. These kernels shape how correlations in the data are modeled, influencing how predictions are impacted by nearby points.

To enhance performance, Automatic Relevance Determination (ARD) versions of these kernels were also tested. ARD assigns individual length scales to each input variable, enabling the model to automatically assess the relative importance of predictors. This is particularly beneficial for datasets with features of varying significance.

To ensure comparability among features and prevent any single feature from disproportionately affecting the model, the input data was standardized using Z-score normalization. This process rescaled each feature to have a mean of zero and a standard deviation of one, ensuring uniform influence across all variables.

This transformation is essential in GPR models to ensure that all variables contribute appropriately to the prediction process, especially when using covariance functions sensitive to the scale of the input data.

The GPR model employed a constant basis function, meaning the model assumes a fixed baseline for the predictions, allowing the kernel functions to capture the complexity in the relationships. The model was trained and evaluated using a holdout validation approach. The values of the hyperparameters are given in Tables 8 and 9.

Table 8. Parameters of GPR model covariance functions.

GP Model Covariance Function	Covariance Function Parameters		
Exponential	$k((x_i, x_j \Theta)) = \sigma_f^2 \exp\left[-\frac{1}{2} \frac{r}{\sigma_l^2}\right]$		
	$\sigma_l = 48.8828$	$\sigma_f = 35.9025$	
Squared Exponential	$k((x_i, x_j \Theta)) = \sigma_f^2 \exp\left[-\frac{1}{2} \frac{(x_i - x_j)^T (x_i - x_j)}{\sigma_l^2}\right]$		
	$\sigma_l = 1.1564$	$\sigma_f = 11.6670$	
Matern 3/2	$k((x_i, x_j \Theta)) = \sigma_f^2 \left(1 + \frac{\sqrt{3}r}{\sigma_l}\right) \exp\left[-\frac{\sqrt{3}r}{\sigma_l}\right]$		
	$\sigma_l = 1.9430$	$\sigma_f = 12.5620$	
Matern 5/2	$k((x_i, x_j \Theta)) = \sigma_f^2 \left(1 + \frac{\sqrt{5}r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2}\right) \exp\left[-\frac{\sqrt{5}r}{\sigma_l}\right]$		
	$\sigma_l = 1.6073$	$\sigma_f = 12.0472$	
Rational Quadratic	$k((x_i, x_j \Theta)) = \sigma_f^2 \left(1 + \frac{r^2}{2a\sigma_l^2}\right)^{-\alpha}; r = 0$		
	$\sigma_l = 1.9747$	$a = 0.0057$	$\sigma_f = 39.8560$

where $r = \sqrt{(x_i - x_j)^T (x_i - x_j)}$.

Table 9. Parameters of GPR ARD model covariance functions.

Covariance Function Parameters					
σ_1	σ_2	σ_3	σ_4	σ_5	σ_6
ARD Exponential:					
45.0052	813.3275	41.5953	15.0703	83.9007	199.8780
ARD Squared exponential:					
0.1181	6.1165	0.1452	0.3453	2.0239	1.5321
ARD Matern 3/2:					
0.4342	10.3369	0.8032	0.2128	1.7551	3.0059
ARD Matern 5/2:					
0.2738	8.1979	0.6294	0.1177	1.1174	2.2108
ARD Rational quadratic:					
1.2252	23.3200	1.2758	0.3892	2.1230	5.8575

where $r = \sqrt{\sum_{m=1}^d \frac{(x_{im} - x_{jm})^2}{\sigma_m^2}}$.

The parameters of the covariance functions used in the analysis were optimized through gradient-based methods, applied to the log marginal likelihood expression (Tables 8 and 9). The length scale values for each input variable offer insights into the importance of these variables concerning the model’s predictive performance (Table 9). Variables with smaller length scales tend to have a greater influence on the model, as the relationship between the input and output changes more rapidly with those features. In contrast, larger length scales indicate less significant features.

All models with ARD perform better than their non-ARD counterparts (Tables 10 and 11, where the optimal values are highlighted in bold font).

Table 10. Comparative analysis of results without using ARD.

Model	RMSE	MAE	MAPE/100	R
Exp.	3.2790	2.1928	0.1498	0.9558
Sq.Exp.	3.8110	2.6042	0.1794	0.9395
Mattern 3/2	3.5651	2.3879	0.1632	0.9475
Mattern 5/2	3.6651	2.4752	0.1697	0.9443
Rat.Quadratic	3.3152	2.2421	0.1593	0.9551

Table 11. Comparative analysis of results using ARD.

Model	RMSE	MAE	MAPE/100	R
ARD Exp.	2.9039	1.8953	0.1257	0.9650
ARD Sq.Exp.	3.4447	2.3562	0.1590	0.9511
ARD Mattern 3/2	2.8671	1.9319	0.1329	0.9658
ARD Mattern 5/2	3.0073	2.0360	0.1410	0.9623
ARD Rat.Quadratic	2.9167	1.9377	0.1323	0.9647

The ARD mechanism allows the models to focus on the most important features, leading to better predictive performance and lower error metrics.

The provided Figure 22 uses the inverse of these length scales to visually highlight variable importance, the smaller the length scale, the larger the corresponding bar in the graph, indicating greater influence on the output. To further validate the predictive capabilities of the optimal GPR ARD Exponential model, a Shapley analysis was conducted.

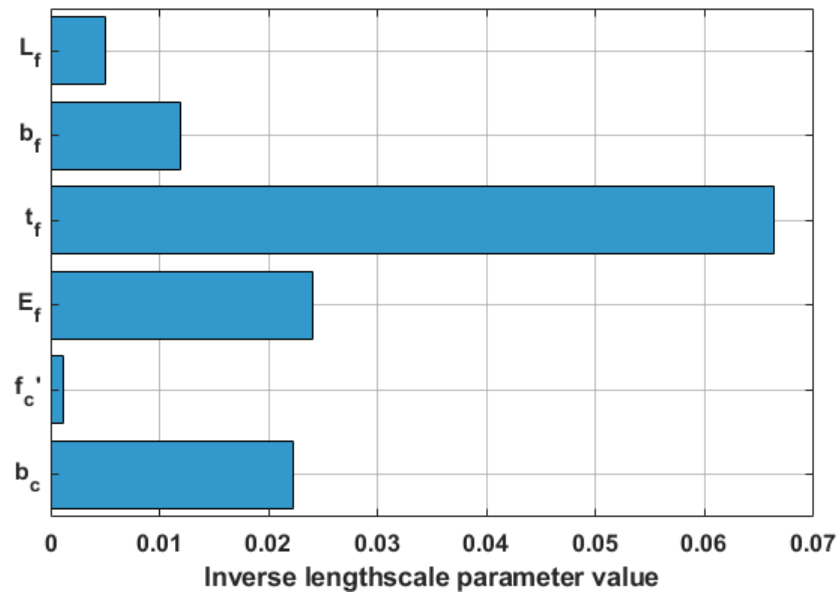


Figure 22. Variable importance in the optimal GPR ARD Exponential model using inverse lengthscale parameter values.

The mean-based Shapley values indicate that FRP thickness (t_f) exerts the strongest influence on the model’s average prediction, followed by the FRP strip width (b_f) and the elastic modulus (E_f). Additionally, both the bond length (L_f) and the concrete compressive strength (f'_c) exhibit minor effects on the average outcome, while concrete substrate width (b_c) has a small contribution (Figure 23).

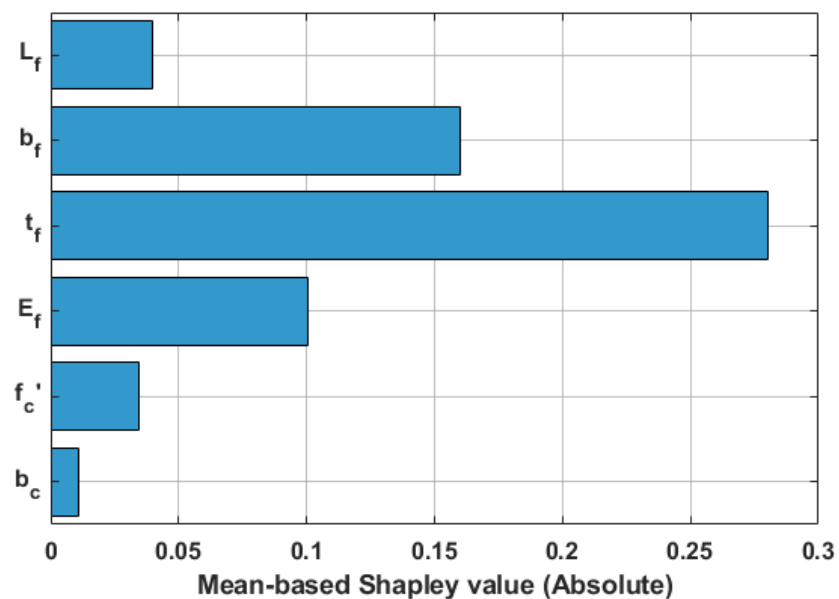


Figure 23. Variable importance in the GPR ARD Exponential model using Shapley values.

The neural network model architecture was established through a trial-and-error method. In this study, the input layer comprises 6 neurons, and the output layer consists of a single neuron.

The optimal number of neurons in the hidden layer was determined experimentally. Beginning with a single neuron, the number of neurons was incrementally increased, with each configuration assessed based on RMSE, MAE, R, and MAPE metrics. Models utilizing the LM algorithm were tested, with data divided into the same training and test sets for analysis.

All model variables underwent linear transformation to ensure all variables were scaled to the range $[-1, 1]$. Here, the minimum value was mapped to -1 and the maximum to 1 , with intermediate values scaled linearly.

Scaling was applied to equalize the influence of each variable, as a variable’s absolute size may not reflect its actual impact. The same standard settings in MATLAB 2020 a. were applied across all model architectures during training (Table 12).

Table 12. Parameter settings for model calibration in MATLAB.

Parameter	Value
Epoch limit	1000
MSE target (performance)	0
Gradient limit	1.00×10^{-7}
Mu value	Range from 0.005 to 1.00×10^{10}

As guidelines for estimating the upper limit of the neuron count in the hidden layer, the previously mentioned expressions were used, with a recommendation to adopt the lower value as follows:

$$N_H \leq 2 \times N_i + 1 = 2 \times 6 + 1 = 13$$

$$N_H \leq \frac{N_s}{N_i + 1} = \frac{855}{6 + 1} = 122.14.$$

The RMSE, MAE, MAPE, and R values for different network architectures, tested with up to 23 neurons, are illustrated in Figure 24.

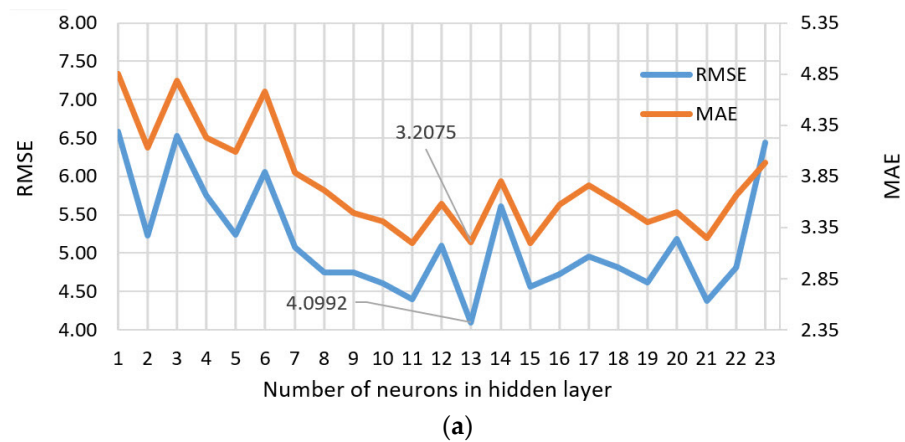


Figure 24. Cont.

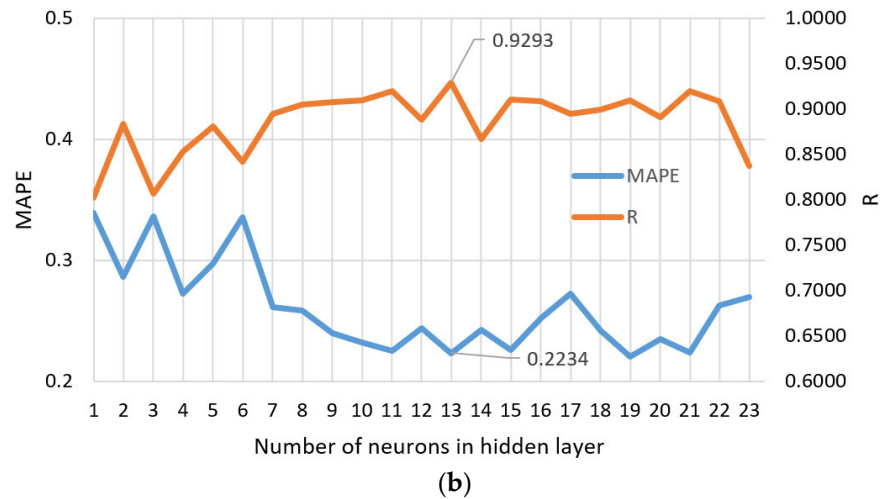


Figure 24. Relationship between the accuracy criteria and the number of hidden neurons: (a) RMSE and MAE, (b) MAPE and R.

These metrics indicate that the model with 13 neurons in the hidden layer achieved optimal performance, close to the upper boundary of the initially tested range. Consequently, the range was expanded to evaluate the impact of additional neurons. However, adding more neurons did not enhance accuracy; rather, it highlighted a risk of overfitting. Beyond this limit, the model began to fit too closely to the training data, reducing its ability to generalize effectively to unseen data.

Table 13 presents a comparative analysis of several ML models used for predicting bond strength, evaluated based on four key accuracy metrics: RMSE, MAE, MAPE, and the correlation coefficient (R).

Table 13. Comparative analysis of all optimal analyzed models based on the selected accuracy criteria.

Model	RMSE	MAE	MAPE/100	R
Linear with interactions	4.9278	3.8491	0.2748	0.8955
MGGP	4.4436	3.4294	0.2167	0.9147
Gradient Boosted Trees	3.3427	2.2603	0.1559	0.9536
Random Forest (64 splits)	3.7860	2.5821	0.1778	0.9415
TreeBagger	3.8302	2.5847	0.1790	0.9399
SVR RBF	4.9646	3.5352	0.1171	0.9332
GPR Exponential	3.2790	2.1928	0.1498	0.9558
GPR ARD Exponential	2.8671	1.9319	0.1329	0.9658
NN 6-13-1	4.0992	3.2075	0.2234	0.9293

Among the models evaluated, the GPR ARD Exponential model outperforms all others, with the lowest RMSE (2.8671 MPa), MAE (1.9319 MPa), and MAPE (0.1329%) while achieving the highest correlation coefficient (R = 0.9658). This indicates its superior accuracy and predictive power. The optimal values of the model accuracy criteria in Table 13 are displayed in bold font.

The GBT model also shows strong performance, with an RMSE of 3.3427 MPa and an R of 0.9536, making it a competitive alternative for situations requiring complex, non-linear modeling. Similarly, RF and TB models also perform well, with RMSE values of 3.7860 MPa and 3.8302 MPa, respectively, and R values above 0.939. These models are effective in reducing variance and offer robustness through ensemble learning.

The SVR RBF model, while slightly less accurate than the tree-based models, still achieves reasonable results with an R of 0.9332 and a relatively low MAE (3.5352 MPa), suggesting its applicability in scenarios where a simpler model is preferred.

On the other hand, the MGGP, despite their interpretability, show lower accuracy with higher RMSE values (4.4436 MPa).

This indicates that while these models may be useful for applications requiring simple, symbolic expressions, they may not be the best choice when high accuracy is critical.

Finally, the Neural Network (NN 6-13-1) model, with an RMSE of 4.0992 MPa and R of 0.9293, offers a balanced trade-off between model complexity and accuracy. However, it still falls short compared to the top-performing models like GPR and Gradient Boosted.

In summary, the GPR ARD Exponential model is the optimal choice based on all evaluated metrics, though simpler models such as GBT or RF may still be valuable depending on the application, especially in terms of balancing complexity and performance.

The analysis of feature importance using Shapley values, (mean-based method), highlights that FRP thickness (t_f) and strip width (b_f) are the most influential variables in predicting bond strength. Across models, t_f exhibits the highest Shapley values, underscoring its critical role in stress transfer and bond-slip behavior. Physically, a thicker FRP layer enhances load-carrying capacity and delays debonding failure by distributing stresses more effectively along the bond line. Similarly, b_f ranks as the second most important variable due to its role in uniform stress distribution, reducing stress concentrations, and stabilizing the bond mechanism. These findings are consistent with prior research by Lu et al. [74] and Teng et al. [75], as well as studies employing machine learning methods to predict the load-carrying capacity of FRP strips, such as those by Zhou et al. [4] and Zhang et al. [76].

The elastic modulus of FRP (E_f) and bond length (L_f) have a moderate influence on bond strength predictions. E_f governs the stiffness of the FRP system, limiting relative deformation and enhancing bond performance. Alongside t_f , E_f significantly impacts bond behavior. The stiffness of the strip ($K_f = E_f \times t_f$), identified in similar machine learning-based studies [4], has also been highlighted as a critical factor for predicting load capacity. On the other hand, L_f ensures adequate anchorage for stress transfer, but its contribution is context-dependent and diminishes beyond the effective bond length. This finding aligns with the work of Chen et al. [77], which noted the stabilizing effects of E_f and the diminishing returns of L_f beyond critical thresholds. Additionally, studies by Zhou et al. and Zhang et al. classify L_f as a less influential variable.

The concrete width b_c and compressive strength (f'_c) are the least influential variables. While b_c contributes to the structural stability of the bond system, it is overshadowed by the direct impact of FRP-specific properties. Similarly, f'_c ranks as the least important feature. A possible explanation for the limited impact of f'_c is that variables such as t_f , b_f , and E_f exhibit stronger nonlinear correlations with bond strength, reducing the model's dependence on f'_c . Moreover, the narrow range of f'_c variation (~25–50 MPa) may prevent the model from capturing significant effects. Another potential reason is that the FRP system may fail before the concrete becomes the “weak link”, minimizing f'_c influence. These results are in agreement with studies by Zhang et al., who also found that FRP properties outweigh f'_c in determining bond strength.

Figure 25 provides a direct comparison between the target and predicted values for the optimal GPR ARD Exponential model. This visual further demonstrates how well the model predicts the bond strength, with the predictions closely matching the target values across the dataset. The closeness of the predicted values to the target values confirms the high accuracy and reliability of the model in predicting bond strength.

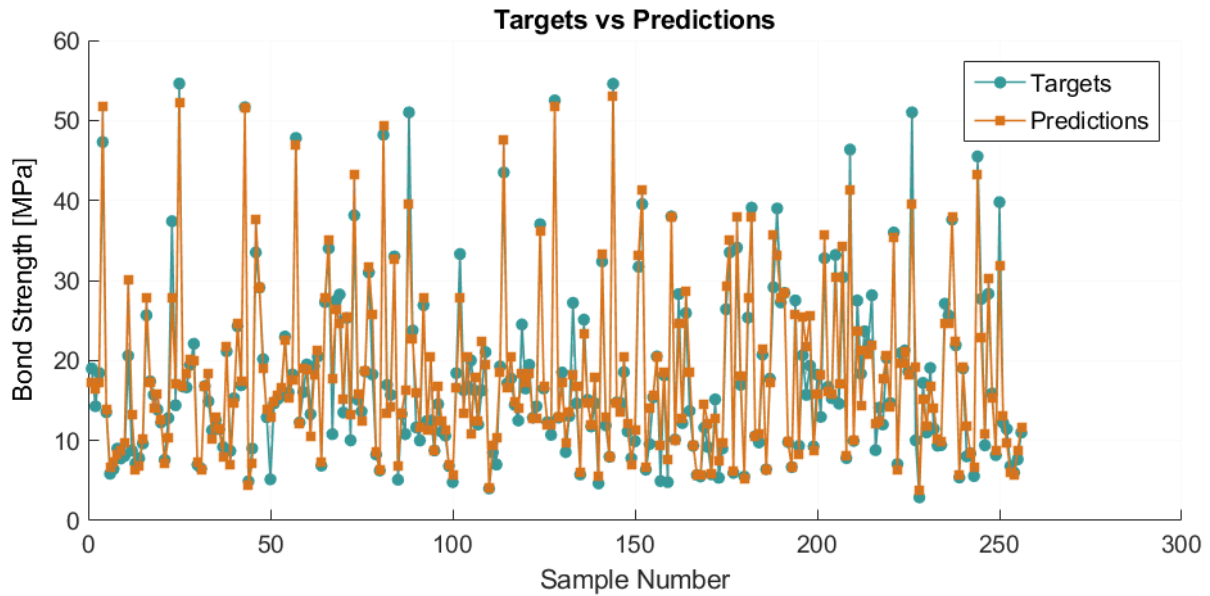


Figure 25. Diagram of the target values and predictions for the optimal GPR ARD Exponential model.

The regression plot (Figure 26) reflects a strong correlation between the actual and predicted results, indicating the model’s ability to effectively capture the relationship between the input features and bond strength.

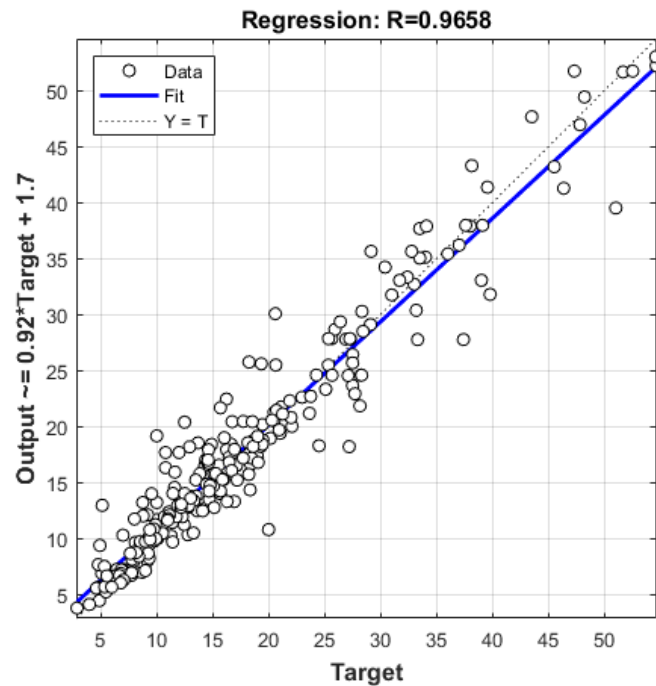


Figure 26. Regression plot of the target and predicted values for the GPR ARD Exponential model.

An analysis of the variability in bond strength was performed for different concrete compressive strength intervals, specifically for concrete strengths up to 25 MPa, from 25 to 50 MPa, and above 50 MPa (Figure 27). In all these categories of concrete quality, similar performance metrics were obtained, demonstrating comparable accuracy across the subsets. The results revealed that the model with the GPR ARD Exponential kernel maintained nearly identical accuracy in all concrete strength intervals, further confirming the robustness and generalizability of the model across varying concrete qualities (Figure 27).

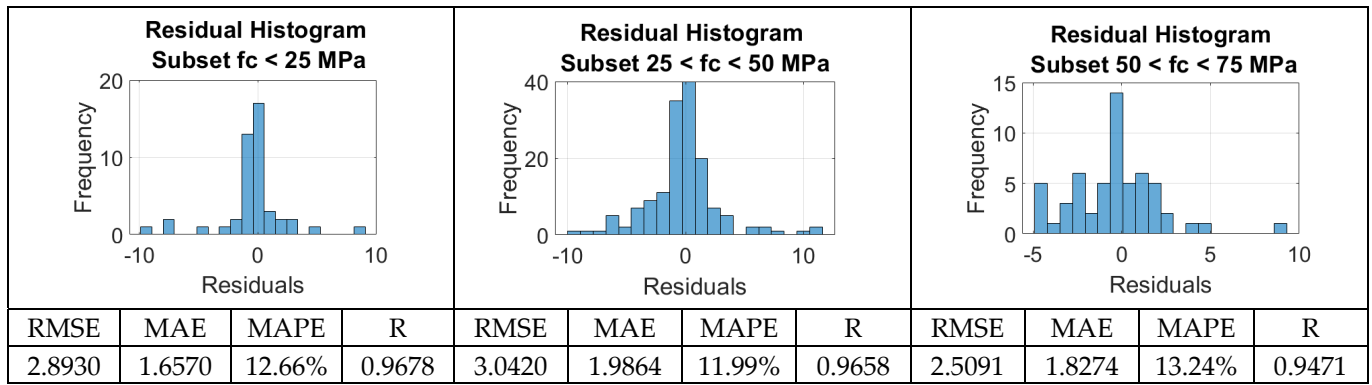


Figure 27. Residual histogram for the different concrete strength samples.

While the model is relatively complex, a MATLAB Graphical User Interface (GUI) was developed to facilitate practical implementation by experts.

The model’s code includes a built-in database, making it easy to adapt the model through the expansion of the database with additional data. The MATLAB code is provided in the Supplementary Materials (MATLAB code with the complete database of models used in the research.). The GUI functions by allowing the user to input the corresponding values for the six input variables, and the prediction is obtained by clicking the ‘Predict Bond Strength’ button (Figure 28).

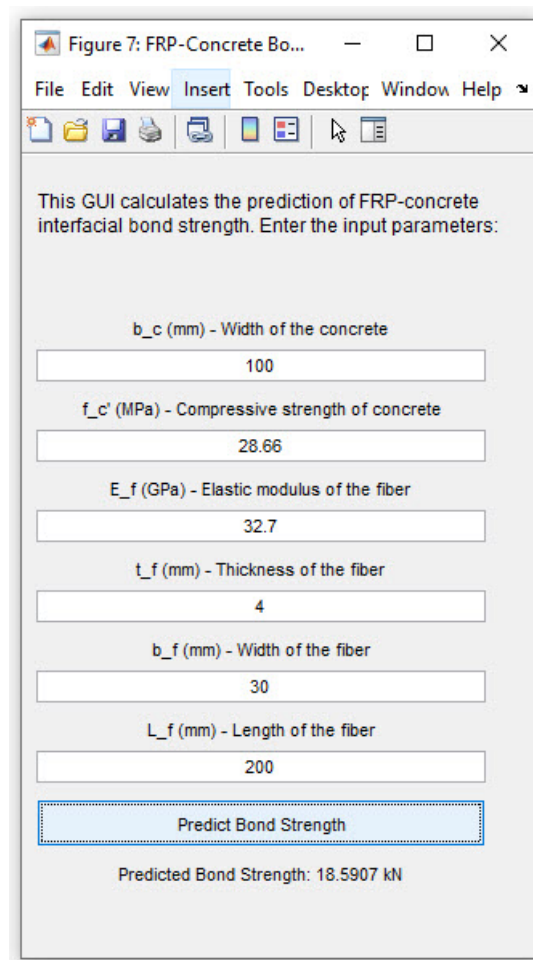


Figure 28. MATLAB graphical user interface (GUI).

6. Conclusions

The primary contribution of this study lies in the development and evaluation of robust machine learning models for predicting bond strength between FRP and concrete. By utilizing a comprehensive dataset from 38 independent studies, this research provides significant insights into the factors influencing bond strength.

Using an extensive dataset of 855 experimental results, the study assessed multiple advanced algorithms, including multiple linear regression with interactions, Multigene Genetic Programming (MGGP), ensemble methods (Gradient Boosted Trees, TreeBagger, Random Forest), Support Vector Regression (SVR), Gaussian Process Regression (GPR), and neural networks.

Among the models evaluated, GPR with Automatic Relevance Determination (ARD) proved to be the most accurate, achieving an RMSE of 2.8671 MPa, MAE of 1.8953 MPa and a correlation coefficient (R) of 0.9658.

The Gradient Boosted Trees ensemble model also showed high accuracy with an RMSE of 3.3427 MPa, MAE of 2.2603 MPa and R of 0.9536, especially when optimized through grid search for learning rate and tree depth, making it a strong candidate for complex, non-linear interactions between variables. While MGGP achieved slightly lower accuracy, it provided interpretable models that balance predictive performance with simplicity, which is advantageous for practical applications.

The analysis of mean-based Shapley values reveals that FRP thickness (t_f) has the most significant impact on the model's average prediction, followed by the width of the FRP strip (b_f) and the elastic modulus (E_f). In contrast, bond length (L_f) and concrete compressive strength (f'_c) show only minimal effects on the average prediction, with concrete substrate width (b_c) contributing the least.

In terms of practical applications, these models can be used to guide the design and optimization of FRP systems in civil engineering projects. The ability to predict bond strength with high accuracy can help engineers select appropriate materials and design parameters, leading to more durable and cost-effective FRP-reinforced concrete structures. Moreover, the models could be integrated into decision-support systems for real-time predictions during construction, enhancing the efficiency of FRP application.

However, several areas for future research remain. One potential improvement is the incorporation of additional experimental methods, such as double-shear tests, to further validate the findings across different testing conditions. Further research could also explore the impact of long-term environmental factors (e.g., temperature, humidity) on bond strength, which was not fully captured in the current dataset. Additionally, refining the models through more advanced techniques, such as deep learning approaches, could improve prediction accuracy and handle larger, more diverse datasets. Also, limitation of the method is the lack of data regarding the type of failure when the ultimate capacity of FRP-strengthened concrete is reached. Expanding the database to include information on failure modes would enhance the analysis and provide a more comprehensive understanding of bond behavior. Additionally, while the Shapley method was used to determine the importance of variables—representing a modern and robust approach—it can become computationally demanding when dealing with a larger number of variables.

In conclusion, while this study provides a solid foundation for understanding bond strength prediction in FRP-concrete systems, future work should aim to broaden the dataset, incorporate environmental variables, and explore new modeling techniques to improve prediction accuracy and applicability across real-world conditions. These improvements will help ensure that the models can be widely adopted in industry, further advancing the use of FRP for reinforced concrete structures.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/computation13010017/s1>, The Supplementary Materials accompanying the article include a detailed MATLAB model for implementing the optimal Gaussian Process Regression with Automatic Relevance Determination (GPR ARD) model. This package is complemented by a user-friendly graphical user interface (GUI) that allows users to interactively explore the model’s capabilities, adjust parameters, and visualize the results. Additionally, a GUI has been also developed for the MGGP model and for calculating the contributions of individual terms in the MGGP model. The MATLAB models and GUIs for the optimal GPR ARD and MGGP models are available online.

Author Contributions: Conceptualization, M.K., M.H.-N. and P.P.; methodology, M.K., M.H.-N. and P.P.; software, M.K., M.H.-N. and P.P.; validation, M.K., T.V. and M.R.; formal analysis, M.K., T.V. and M.R.; investigation, M.K.; resources, M.H.-N.; data curation, M.K., T.V. and M.R.; writing—original draft preparation, M.K. and M.H.-N.; writing—review and editing, M.K., M.H.-N. and P.P.; visualization, M.K., T.V. and M.R.; supervision, M.H.-N.; project administration, M.H.-N.; funding acquisition, M.H.-N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All experimental data used for developing the GPR ARD model is included in the MATLAB file for creating the model and the corresponding graphical user interface. In addition, a graphical user interface (GUI) has also been developed for the MGGP model and for calculating the contributions of individual terms in the MGGP model. Furthermore, the training and test datasets, as well as the entire database, are provided in the form of an attached Excel file.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Comprehensive compilation of published experimental data [4].

Reference	Num. of Tests	f_c (MPa)	E_f (GPa)	t_f (mm)	L_f (mm)	b_f (mm)	b_c (mm)	P_u (kN)
Adhikary and Mutsuyoshi [33]	7	24–36.5	230	0.11–0.33	100–150	100	150	16.75–28.25
Bilotta et al. [34]	29	21.46–26	170–241	0.166–1.4	100–400	50–100	150	17.24–33.56
Bilotta et al. [35]	13	19	109–221	1.2–1.7	300	60–100	160	29.86–54.79
Bimal and Hiroshi [36]	7	24–36.5	230	0.111–0.334	100–150	100	150	16.8–28.3
Carlo et al. [37]	14	58–63	230–390	0.165–0.495	65–130	50	100	12.1–29.8
Chajes et al. [38]	15	24–48.87	108.48	1.016	51–203	25.4	152.4–228.6	8.09–12.81
Czaderski and Olia [39]	8	32–33	165–175	1.23–1.68	300	100	150	43.5–56.1
Dai et al. [40]	19	33.1–35	74–230	0.11–0.59	210–330	100	400	15.6–51
Faella et al. [41]	3	32.78–37.55	140	1.4	200–250	50	150	31–39.78
Fen et al. [42]	11	8–36	240.72–356.75	0.111	50–120	50–100	150	7.13–17.34
Hoseini and Mostofinejad [43]	22	36.5–41.1	238	0.131	20–250	48	150	7.58–10.12
Kanakubo et al. [44]	12	23.8–57.6	252.2–425.1	0.083–0.334	300	50	100	7–25.6
Kamiharako et al. [45]	17	34.9–75.5	270	0.111–0.222	100–250	10–90	100	3.1–14.9
Ko et al. [46]	13	27.7–31.4	165–210	1–1.4	300	60–100	150	27.5–56.5
Liu [47]	57	16–51.6	272.66	0.167	50–300	50	100	10.97–23.87
Lu et al. [48]	3	47.64–64.08	230–390	0.22–0.501	200–250	40–100	100–500	14.1–38
Maeda et al. [49]	5	40.8–44.91	230	0.11–0.22	65–300	50	100	5.8–16.25
Nakaba et al. [50]	41	24.41–65.73	124.5–425	0.167–2	250–300	40–50	100	8.73–27.24
Pham and Al-Mahaidi [51]	23	44.57	209	0.176	60–220	70–100	140	18.8–42.8
Ren [52]	28	22.96–46.07	83.03–207	0.33–0.507	60–150	20–80	150	4.61–22.8
Savoia and Ferracuti [53]	14	52.6	165–291.02	0.13–1.2	200–400	50–80	150	14.4–41
Savoia et al. [54]	20	26	180–241	0.166–1.2	100–400	80–100	150	18.97–40
Sharma et al. [55]	24	23.76–28.66	32.7–300	1.2–4	100–300	30–50	100	12.5–46.35
Tan [56]	6	30.8	97–235	0.111–0.169	70–130	50	100	6.46–11.43
Täljsten [57]	5	41.2–68.33	162–170	1.2–1.25	100–300	50	200	17.3–35.1
Takeo et al. [58]	25	24.7–29.25	230–373	0.111–0.501	100–300	40	100	6.75–14.35
Toutanji et al. [59]	10	17.0–61.5	110	0.495–0.99	100	50	200	11.64–19.03
Ueda et al. [60]	15	23.79–48.85	230–372	0.11–0.55	65–300	10–100	100–500	2.4–38
Ueno et al. [61]	40	23–74.5	42.625–43.537	1.03–1.8	200–230	40	80	9.52–18.29
Wu and Jiang [3]	65	25.3–59.02	238.1–248.3	0.167	30–400	50	150	7.38–30.15
Wu et al. [62]	22	65.73	23.9–390	0.083–1	250–300	40–100	100	11.8–27.25
Woo and Lee [63]	51	24–40	152.2	1.4	50–300	10–50	200	4.55–27.8
Fu et al. [64]	24	24.1–70	230	0.17–0.84	50–300	30–70	100	7.8–31.13
Yao [65]	59	19.12–27.44	22.5–256	0.165–1.27	75–240	25–100	100–150	4.75–19.07
Yuan et al. [66]	1	23.79	256	0.165	190	25	150	5.74
Zhang et al. [67]	20	38.9–43.5	94–227	0.262–0.655	250	50–150	200–250	13.03–52.49
Zhao et al. [68]	5	16.4–29.36	240	0.083	100–150	100	150	11–12.75
Zhou [69]	102	48.56–74.67	71–237	0.111–0.341	20–200	15–150	150	3.75–28

References

1. Hadzima-Nyarko, M.; Čolak, S.; Bulajić, B.Đ.; Ademović, N. Assessment of Selected Models for FRP-Retrofitted URM Walls under In-Plane Loads. *Buildings* **2021**, *11*, 559. [[CrossRef](#)]
2. Hadzima-Nyarko, M.; Ademović, N.; Pavić, G.; Kalman Šipoš, T. Strengthening techniques for masonry structures of cultural heritage according to recent Croatian provisions. *Earthq. Struct.* **2018**, *15*, 473–485. [[CrossRef](#)]
3. Wu, Y.; Jiang, C. Quantification of Bond-Slip Relationship for Externally Bonded FRP-to-Concrete Joints. *J. Compos. Constr.* **2013**, *17*, 673–686. [[CrossRef](#)]
4. Zhou, Y.; Zheng, S.; Huang, Z.; Sui, L.; Chen, Y. Explicit Neural Network Model for Predicting FRP-Concrete Interfacial Bond Strength Based on a Large Database. *Compos. Struct.* **2020**, *240*, 111998. [[CrossRef](#)]
5. Li, J.; Gravina, R.J.; Smith, S.T.; Visintin, P. Bond Strength and Bond Stress-Slip Analysis of FRP Bar to Concrete Incorporating Environmental Durability. *Constr. Build. Mater.* **2020**, *261*, 119860. [[CrossRef](#)]
6. Su, M.; Zhong, Q.; Peng, H.; Li, S. Selected Machine Learning Approaches for Predicting the Interfacial Bond Strength between FRPs and Concrete. *Constr. Build. Mater.* **2021**, *270*, 121456. [[CrossRef](#)]
7. Haddad, R.; Haddad, M. Predicting FRP-concrete bond strength using artificial neural networks: A comparative analysis study. *Struct. Concr.* **2021**, *22*, 38–49. [[CrossRef](#)]
8. Chen, S.-Z.; Zhang, S.-Y.; Han, W.-S.; Wu, G. Ensemble Learning Based Approach for FRP-Concrete Bond Strength Prediction. *Constr. Build. Mater.* **2021**, *302*, 124230. [[CrossRef](#)]
9. Barkhordari, M.S.; Armaghani, D.J.; Sabri, M.M.S.; Ulrikh, D.V.; Ahmad, M. The Efficiency of Hybrid Intelligent Models in Predicting Fiber-Reinforced Polymer Concrete Interfacial-Bond Strength. *Materials* **2022**, *15*, 3019. [[CrossRef](#)]
10. Alabdullh, A.A.; Biswas, R.; Gudainiyani, J.; Khan, K.; Bujbarah, A.H.; Alabdulwahab, Q.A.; Amin, M.N.; Iqbal, M. Hybrid Ensemble Model for Predicting the Strength of FRP Laminates Bonded to the Concrete. *Polymers* **2022**, *14*, 3505. [[CrossRef](#)] [[PubMed](#)]
11. Kim, B.; Lee, D.-E.; Hu, G.; Natarajan, Y.; Preethaa, S.; Rathinakumar, A.P. Ensemble Machine Learning-Based Approach for Predicting of FRP-Concrete Interfacial Bonding. *Mathematics* **2022**, *10*, 231. [[CrossRef](#)]
12. Jamhiri, B.; Jalal, F.E.; Chen, Y. Hybridizing Multivariate Robust Regression Analyses with Growth Forecast in Evaluation of Shear Strength of Zeolite-Alkali Activated Sands. *Multiscale Multidiscip. Model. Exp. Des.* **2022**, *5*, 317–335. [[CrossRef](#)]
13. Bui, D.T.; Khosravi, K.; Tiefenbacher, J.; Nguyen, H.; Kazakis, N. Improving Prediction of Water Quality Indices Using Novel Hybrid Machine-Learning Algorithms. *Sci. Total Environ.* **2020**, *721*, 137612. [[CrossRef](#)]
14. Parhi, S.K.; Patro, S.K. Prediction of Compressive Strength of Geopolymer Concrete Using a Hybrid Ensemble of Grey Wolf Optimized Machine Learning Estimators. *J. Build. Eng.* **2023**, *71*, 106521. [[CrossRef](#)]
15. Searson, D.P.; Leahy, D.E.; Willis, M.J. GPTIPS: An Open-Source Genetic Programming Toolbox For Multigene Symbolic Regression. In Proceedings of the International Multiconference of Engineers and Computer Scientist (IMECS 2010), Hong Kong, China, 17–19 March 2010; Volume I.
16. Kovačević, M.; Lozančić, S.; Nyarko, E.K.; Hadzima-Nyarko, M. Application of Artificial Intelligence Methods for Predicting the Compressive Strength of Self-Compacting Concrete with Class F Fly Ash. *Materials* **2022**, *15*, 4191. [[CrossRef](#)] [[PubMed](#)]
17. Elith, J.; Leathwick, J.R.; Hastie, T. A working guide to boosted regression trees. *J. Anim. Ecol.* **2008**, *77*, 802–813. [[CrossRef](#)] [[PubMed](#)]
18. Hastie, T.; Tibsiran, R.; Friedman, J. *The Elements of Statistical Learning*; Springer: Berlin/Heidelberg, Germany, 2009.
19. Kovačević, M.; Ivanišević, N.; Petronijević, P.; Despotović, V. Construction cost estimation of reinforced and prestressed concrete bridges using machine learning. *Građevinar* **2021**, *73*, 1–13. [[CrossRef](#)]
20. Breiman, L.; Friedman, H.; Olsen, R.; Stone, C.J. *Classification and Regression Trees*; Chapman and Hall/CRC: Wadsworth, OH, USA, 1984.
21. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [[CrossRef](#)]
22. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
23. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [[CrossRef](#)]
24. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
25. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer: New York, NY, USA, 1995.
26. Kecman, V. *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*; MIT Press: Cambridge, MA, USA, 2001.
27. Smola, A.J.; Sholkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [[CrossRef](#)]
28. Rasmussen, C.E.; Williams, C.K. *Gaussian Processes for Machine Learning*; The MIT Press: Cambridge, MA, USA, 2006.
29. Hagan, M.T.; Menhaj, M.B. Training Feedforward Networks with Marquardt Algorithm. *IEEE Trans. Neural Netw.* **1994**, *5*, 989–993. [[CrossRef](#)] [[PubMed](#)]
30. Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4765–4774.

31. Shapley, L. A value for n-person games. In *Contributions to the Theory of Games II*; Kuhn, H., Tucker, A., Eds.; Princeton University Press: Princeton, NJ, USA, 1953; pp. 307–317.
32. Molnar, C. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 2nd ed.; Christoph Molnar: Munich, Germany, 2022.
33. Adhikary, B.B.; Mutsuyoshi, H. Study on the Bond between Concrete and Externally Bonded CFRP Sheet. In *Proceedings of the 5th International Symposium on Fiber Reinforced Concrete Structures (FRPRCS-5)*; Thomas Telford Publishing: London, UK, 2001; pp. 371–378.
34. Bilotta, A.; Di Ludovico, M.; Nigro, E. FRP-to-Concrete Interface Debonding: Experimental Calibration of a Capacity Model. *Compos. Part B Eng.* **2011**, *42*, 1539–1553. [[CrossRef](#)]
35. Bilotta, A.; Ceroni, F.; Di Ludovico, M.; Nigro, E.; Pecce, M.; Manfredi, G. Bond Efficiency of EBR and NSM FRP Systems for Strengthening Concrete Members. *J. Compos. Constr.* **2011**, *15*, 629–638. [[CrossRef](#)]
36. Bimal, B.A.; Hiroshi, M. Study on the bond between concrete and externally bonded CFRP sheet. In *Proceedings of the 5th International Symposium on FRP Reinforcement for Concrete Structures*, Cambridge, UK, 16–18 July 2001; University of Cambridge: Cambridge, UK, 2001; pp. 371–378.
37. Pellegrino, C.; Tinazzi, D.; Modena, C. Experimental Study on Bond Behavior Between Concrete and FRP Reinforcement. *J. Compos. Constr.* **2008**, *12*, 180–188. [[CrossRef](#)]
38. Chajes, M.J.; Finch, W.W., Jr.; Januszka, T.F.; Thomson, T.A., Jr. Bond and Force Transfer of Composite-Material Plates Bonded to Concrete. *Struct. J.* **1996**, *93*, 209–217. [[CrossRef](#)]
39. Czaderski, C.; Olia, S. EN-Core Round Robin Testing Program—Contribution of Empa. In *Proceedings of the 6th International Conference on FRP Composites in Civil Engineering (CICE 2012)*, Rome, Italy, 13–15 June 2012; pp. 1–8. Available online: <https://www.dora.lib4ri.ch/empa/islandora/object/empa:9237> (accessed on 12 February 2020).
40. Dai, J.-G.; Sato, Y.; Ueda, T. Improving the load transfer and effective bond length for FRP composites bonded to concrete. *Proc. Jpn. Concr. Inst.* **2002**, *24*, 1423–1428.
41. Faella, C.; Nigro, E.; Martinelli, E.; Sabatino, M.; Salerno, N.; Mantegazza, G. Aderenza tra calcestruzzo e Lamina di FRP utilizzate come placcaggio di elementi inflessi. Parte I: Risultati sperimentali. In *Proceedings of the XIV Congresso C.T.E.*, Mantova, Italy, 9 November 2002; pp. 7–8.
42. Fen, Z.; Gu, X.; Zhang, W.; Liu, L. Experimental study on bond behavior between carbon fiber reinforced polymer and concrete. *Struct. Eng.* **2008**, *24*, 154–163.
43. Hosseini, A.; Mostofinejad, D. Effective Bond Length of FRP-to-Concrete Adhesively-Bonded Joints: Experimental Evaluation of Existing Models. *Int. J. Adhes. Adhes.* **2014**, *48*, 150–158. [[CrossRef](#)]
44. Kanakubo, T.; Nakaba, K.; Yoshida, T.; Yoshizawa, H. A Proposal for the Local Bond Stress–Slip Relationship between Continuous Fiber Sheets and Concrete. *Concr. Res. Technol.* **2001**, *12*, 33–43. (In Japanese) [[CrossRef](#)] [[PubMed](#)]
45. Kamiharako, A.; Shimomura, T.; Maruyama, K. The Influence of the Substrate on the Bond Behavior of Continuous Fiber Sheet. *Proc. Jpn. Concr. Inst.* **2003**, *25*, 1735–1740.
46. Ko, H.; Matthys, S.; Palmieri, A.; Sato, Y. Development of a simplified bond stress–slip model for bonded FRP–concrete interfaces. *Constr. Build. Mater.* **2014**, *68*, 142–157. [[CrossRef](#)]
47. Liu, J. Effect of Concrete Strength on Interfacial Bond Behavior of CFRP–Concrete. Ph.D. Thesis, Shenzhen University, Shenzhen, China, 2012. (In Chinese)
48. Lu, X.Z.; Ye, L.P.; Teng, J.G.; Jiang, J.J. Meso-Scale Finite Element Model for FRP Sheets/Plates Bonded to Concrete. *Eng. Struct.* **2005**, *27*, 564–575. [[CrossRef](#)]
49. Maeda, T.; Asano, Y.; Sato, Y.; Ueda, T.; Kakuta, Y. A study on bond mechanism of carbon fiber sheet. In *Proceedings of the 3rd International Symposium on Non-Metallic (FRP) Reinforcement for Concrete Structures*, Sapporo, Japan, 14–16 October 1997; pp. 279–285.
50. Nakaba, K.; Kankubo, T.; Furuta, T.; Yoshizawa, H. Bond behavior between fiber-reinforced polymer laminates and concrete. *ACI Struct. J.* **2001**, *98*, 359–367.
51. Pham, H.B.; Al-Mahaidi, R. Modelling of CFRP–Concrete Shear-Lap Tests. *Constr. Build. Mater.* **2007**, *21*, 727–735. [[CrossRef](#)]
52. Ren, H. Study on Basic Mechanical Properties and Long-Term Mechanical Properties of Concrete Structures Strengthened with Fiber Reinforced Polymer. Ph.D. Thesis, Dalian University of Technology, Dalian, China, 2003. (In Chinese)
53. Savoia, M.; Ferracuti, B. Strengthening of RC Structure by FRP: Experimental Analyses and Numerical Modelling. Ph.D. Thesis, University of Bologna, Bologna, Italy, 2006.
54. Savoia, M.; Bilotta, A.; Ceroni, F.; Di Ludovico, M.; Fava, G.; Ferracuti, B.; Mazzotti, C.; Nigro, E.; Olivito, R.S.; Pecce, M.; et al. Experimental Round Robin Test on FRP–Concrete Bonding. In *Proceedings of the 9th International Symposium on Fiber Reinforced Polymer Reinforcement for Concrete Structures (FRPRCS-9)*, Sydney, Australia, 13–15 July 2009.
55. Sharma, S.K.; Mohamed Ali, M.S.; Goldar, D.; Sikdar, P.K. Plate–Concrete Interfacial Bond Strength of FRP and Metallic Plated Concrete Specimens. *Compos. Part B Eng.* **2006**, *37*, 54–63. [[CrossRef](#)]

56. Tan, Z. Experimental Study on the Performance of Concrete Beams Strengthened with GFRP. Master's Thesis, Tsinghua University, Beijing, China, 2002. (In Chinese)
57. Taljsten, B. Plate Bonding: Strengthening of Existing Concrete Structures with Epoxy Bonded Plates of Steel or Fibre Reinforced Plastics. Ph.D. Thesis, Luleå University of Technology, Luleå, Sweden, 1994.
58. Takeo, M.; Tanaka, T. Analytical solution for bond-slip behavior in FRP-concrete systems. *Compos. Struct.* **2020**, *240*, 111998.
59. Toutanji, H.; Saxena, P.; Zhao, L.; Ooi, T. Prediction of Interfacial Bond Failure of FRP–Concrete Surface. *J. Compos. Constr.* **2007**, *11*, 427–436. [[CrossRef](#)]
60. Ueda, T.; Sato, Y.; Asano, Y. Experimental Study on Bond Strength of Continuous Carbon Fiber Sheet. In *Proceedings of the 4th International Symposium on Fiber Reinforced Polymer Reinforcement for Reinforced Concrete Structures*; ACI: Farmington Hills, MI, USA, 1999; pp. 407–416.
61. Ueno, S.; Toutanji, H.; Vuddandam, R. Introduction of a Stress State Criterion to Predict Bond Strength between FRP and Concrete Substrate. *J. Compos. Constr.* **2015**, *19*, 04014028. [[CrossRef](#)]
62. Wu, Z.S.; Yuan, H.; Yoshizawa, H.; Kanakubo, T. Experimental/Analytical Study on Interfacial Fracture Energy and Fracture Propagation Along FRP–Concrete Interface. In *Fracture Mechanics for Concrete Materials: Testing and Applications*; SP-201; American Concrete Institute: Farmington Hills, MI, USA, 2001.
63. Woo, H.; Lee, H. Bond strength degradation in FRP-strengthened concrete under cyclic loads. *Compos. Struct.* **2004**, *64*, 7–16.
64. Fu, Q.; Xu, J.; Jian, G.G.; Yu, C. Bond Strength between CFRP Sheets and Concrete. In *FRP Composites in Civil Engineering, Proceedings of the International Conference on FRP Composites in Civil Engineering (CICE 2001), Hong Kong, China, 12–15 December 2001*; Teng, J.G., Ed.; Elsevier Science: Oxford, UK, 2001; pp. 357–364.
65. Yao, J. Debonding in FRP-Strengthened RC Structures. Ph.D. Thesis, The Hong Kong Polytechnic University, Hong Kong, China, 2004.
66. Yuan, H.; Wu, Z.S.; Yoshizawa, H. Theoretical solutions on interfacial stress transfer of externally bonded steel/composite laminates. *JSCE* **2001**, *18*, 27–39. [[CrossRef](#)]
67. Zhang, H.; Smith, S.T. Fibre-Reinforced Polymer (FRP)-to-Concrete Joints Anchored with FRP Anchors: Tests and Experimental Trends. *Can. J. Civ. Eng.* **2013**, *40*, 731–742. [[CrossRef](#)]
68. Zhao, H.D.; Zhang, Y.; Zhao, M. Study on Bond Behavior of Carbon Fiber Sheet and Concrete Base. In *Proceedings of the First Chinese Academic Conference on FRP-Concrete Structures*; Building Research Institute of Ministry of Metallurgical Industry: Beijing, China, 2000. (In Chinese)
69. Zhou, Y.W. Analytical and Experimental Study on the Strength and Ductility of FRP–Reinforced High Strength Concrete Beam. Ph.D. Thesis, Dalian University of Technology, Dalian, China, 2009. (In Chinese)
70. Massey, F.J. The Kolmogorov–Smirnov Test for Goodness of Fit. *J. Am. Stat. Assoc.* **1951**, *46*, 68–78. [[CrossRef](#)]
71. Kullback, S.; Leibler, R.A. On Information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [[CrossRef](#)]
72. Good, P.I. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*; Springer: Berlin/Heidelberg, Germany, 2000.
73. Searson, D.P. GPTIPS 2: An Open-Source Software Platform for Symbolic Data Mining. In *Handbook of Genetic Programming Applications*; Gandomi, A.H., Ed.; Springer: New York, NY, USA, 2015; Chapter 22.
74. Lu, X.Z.; Teng, J.G.; Ye, L.P.; Jiang, J.J. Bond–slip models for FRP sheets/plates bonded to concrete. *Eng. Struct.* **2005**, *27*, 920–937. [[CrossRef](#)]
75. Teng, J.G.; Chen, J.F.; Smith, S.T.; Lam, L. Behaviour and strength of FRP-strengthened RC structures: A state-of-the-art review. *Constr. Build. Mater.* **2003**, *17*, 491–503. [[CrossRef](#)]
76. Zhang, F.; Wang, C.; Liu, J.; Zou, X.; Sneed, L.H.; Bao, Y.; Wang, L. Prediction of FRP-concrete interfacial bond strength based on machine learning. *Eng. Struct.* **2023**, *274*, 115156. [[CrossRef](#)]
77. Chen, J.F.; Teng, J.G. Anchorage strength models for FRP and steel plates bonded to concrete. *J. Struct. Eng.* **2001**, *127*, 784–791. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.