Results in Engineering 24 (2024) 103135

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering

Machine learning techniques for predicting the peak response of reinforced concrete beam subjected to impact loading

Ali Husnain [a], Munir Iqbal [a,*], Hafiz Ahmed Waqas [a], Mohammed El-Meligy [b,c], Muhammad Faisal Javed [a,d,**], Rizwan Ullah [e]

[a] Department of Civil Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Swabi, Khyber Pakhtunkhwa, 23640, Pakistan
[b] Applied Science Research Center, Applied Science Private University, Amman, Jordan
[c] Jadara University Research Center, Jadara University, PO Box 733, Irbid, Jordan
[d] Western Caspian University, Baku, Azerbaijan
[e] Department of Civil Engineering, Comsats University Islamabad-Abbottabad Campus, Pakistan

ARTICLE INFO

Keywords:
Machine learning
Peak response
Gene expression programming
Shapley

ABSTRACT

To meet the growing need for resilient structures in seismic and high-impact zones, accurate prediction of the response of reinforced concrete (RC) beams under impact loads is essential. Traditional methods, such as experimental testing and high fidelity finite element models, are often time consuming and resource intensive. To address these challenges, this study investigates various ensemble and non-ensemble machine learning techniques—including support vector machine, gaussian process regression (GPR), k-nearest neighbor (KNN), gene expression programming, random forest, decision tree, boosted tree, adaptive boosting tree, gradient boosting algorithm, stochastic gradient descent, and artificial neural network—for predicting the peak response of RC beams under impact loads. A set of 145 experimental data points from 12 different sources is used to train and evaluate these machine learning models. Key parameters in the data include beam width and depth, span, reinforcement ratios, concrete strength, steel yield strength, deflection, and impact characteristics. Except for KNN, all models showed satisfactory generalization capabilities with R^2 values over 0.8. Statistical errors such as RMSE, a-10 index, MAE, and a-20 index are within acceptable limits. The GPR model is the most effective with R^2 value of 0.95. Moreover, Shapely analysis identified beam depth, impact velocity, and beam breadth as critical factors. Overall, this study demonstrates the efficacy of machine learning in accurately predicting the behavior of RC structures under impact loads, providing valuable tools for civil engineers in design and analysis.

Nomenclature used in the current study

Beams width	b (mm)	Area of top longitudinal steel bars	A_s' (mm^2)
Beams depth	h (mm)	Yield strength of steel bars	f_y' (Mpa)
Total length	L (mm)	Yield strength of the confinement steel bars	f_{vy} (Mpa)
Net span	L_n (mm)	Stirrup ratio	rho_v vs p_v (%)
Reinforcement ratio	rho vs a_s (%)	Impact velocity	V (m/s)
Concrete strength	f_c' (MPa)	Impact mass	M (Kg)
Area of bottom longitudinal steel bars	A_s (mm^2)	Impact force	F_p (KN)
Yield strength of the bottom steel bars		f_y (MPa)	

1. Introduction

Throughout their service life, reinforced concrete (RC) structures encounter various impact loadings, such as vehicle collisions, rockfalls, and blast events. These events differ greatly from static loads due to their rapid application of high energy over a short period, requiring a specialized approach to understand and mitigate their impact on RC structures [1]. In particular, the case of RC simply supported beams impacted at the midspan by a rigid body has captivated several researchers due to its practical significance and the complex dynamics involved [2]. For such scenarios, investigations have proposed a

* Corresponding author.
** Corresponding author.
E-mail address: Muniriqbal0345@gmail.com (M. Iqbal).

https://doi.org/10.1016/j.rineng.2024.103135
Received 29 June 2024; Received in revised form 18 September 2024; Accepted 12 October 2024
Available online 18 October 2024

dual-phase response, consisting of a localized phase and a global phase [3]. During the local phase, impact generates high peak force, inducing the formation of shear cracks near the impact zone. Experimental evidence from multiple impact tests shows that shear cracks appear at the impact zone before any notable bending occurs [4]. This damage localized by the impact force, in the local phase, facilitates the formation of a plastic hinge at the mid-span during the subsequent global response phase.

To prevent shear cracks and plastic hinge formation, it is imperative to analyze and predict the peak response of RC beam subjected to impact loading. However, the inherent complexity of dynamic behavior requires a multifaceted approach for accurate prediction. These may include empirical testing, advanced finite element modeling (FEM), and the development of analytical models designed to capture key degrees of freedom [5]. Moreover, for accurate prediction, there is a need to investigate the effect of geometrical properties, material characteristics, input energy levels and reinforcement detailing on the peak response of RC beam [6].

In response to these needs, researchers have conducted numerous investigations using various experimental setups [7]. For a comprehensive review of the different configurations employed in RC beam impact testing, refer to Refs. [8–10]. One commonly used method is drop-weight impact testing. However, this method has limitations due to variations in sample sizes and testing procedures. The addition of measuring tools such as load cells and high-speed cameras can also complicate the setup. In one study utilizing the drop-weight impact test, an equation was proposed to determine the static shear capacity [11]. Nevertheless, the equation failed to account for the combined effect of the drop weight's mass and impact speed, both of which, according to the impulse-momentum theorem, significantly influence RC beam behavior. Additionally, rigid impact testing often results in maximum damage concentrated in the impacted area, providing an upper bound for the beam's response [12]. These limitation underscores the need for a more comprehensive approach.

Fortunately, the advancement of computational resources has enabled the widespread use of FEM for simulating the dynamic response of RC beams under impact loads. FEM turns out to be a valuable alternative to traditional experimental testing. Ozbolt et al. [13] used FEM for demonstrating the impact response of RC beams with shear reinforcement, particularly those susceptible to shear failure. Likewise, Zhao et al. [14,15] used FEM for drop-weight tests and achieved good agreement between numerical results and experimental data. Hwang et al. [16] developed FEM validated energy-based empirical formula, which demonstrated improved accuracy compared to existing formulas. Adhikary et al. [17] conducted a detailed parametric study through FEM, demonstrating that the maximum mid-span displacement is highly influenced by impact velocity, shear-span to effective depth ratio, and the transverse and longitudinal reinforcement ratios. Fan et al. [18] developed a macro-element model that incorporated a shear spring to capture both normal and punching shear responses, effectively depicting the interaction between impacting objects and RC members. Pham and Hao [19,20] employed FEM to show a strong correlation between the development of plastic hinges and the displacement response of the beams. Despite its widespread use, FEM remains computationally demanding and relies on assumptions that require experimental validation.

To lessen computational burdens, researchers have explored simplified semi-empirical models with fewer degrees of freedom. For example, Yi et al. [21] proposed a two-mass model to capture both local and global responses during impact. Their model demonstrated reasonable accuracy in predicting shear failure when compared to experiments and FEM simulations. Zhao et al. [15] proposed a three-degree-of-freedom model. They split the maximum displacement into shear and flexural components, resulting in estimation errors ranging from 2.9 % to 44.4 % when compared to the FEM results. They also proposed an empirical equation for maximum displacement using kinetic energy and static ultimate bending capacity as inputs. Their equation aligned with both experimental and numerical results [22]. Similar energy-based empirical formulas were developed by Kishi et al. [23] and Zhan et al. [24,25]. Fujikake et al. [26,27] proposed a two-degree-of-freedom mass-spring-damper system for the dynamic response of beams under various loads. However, for agreement with experiments, their model required the assumption of high damping at collision points. Recognizing this limitation of lesser-degree-of-freedom system, Saatci and Vecchio [28] advocated for more sophisticated models.

Recognizing the limitations of simplified analytical models and the complexity of FEM, researchers have explored machine learning as a promising alternative [21,29–32]. Machine learning provides several advantages, such as high accuracy and robustness in complex predictions, and a lower application barrier due to its diminished requirement for in-depth physical understanding [33–36]. While training can be computationally intensive, machine learning models excel in prediction efficiency, often providing better estimate than empirical or semi-empirical formulas [37–39]. Up till now, machine learning models have primarily been utilized to predict the dynamic response of RC structural members under blast loading [40–43]. Almustafa and Nehdi [42] introduced a hybrid model that integrates gradient-boosted regression trees with the Henry gas solubility optimization algorithm to predict the maximum displacement of RC beams under blast loading. They further applied a random forest classifier to forecast the failure mode and crack patterns in these beams. Zhou et al. [44] advanced this research by developing a multi-hidden-layer neural network to evaluate the damage severity of RC columns subjected to blast loads. In addition, they utilized a multi-layer long-short-term memory neural network for failure mode prediction. Research on RC slabs has also explored the application of machine learning models to forecast their behavior under blast conditions [45]. For instance, Y. Shen et al. [46] compiled a dataset of 610 experimental points to build a highly accurate failure mode prediction model for flat slabs, comparing the performance of eight different machine learning models with three empirical models to determine the most effective approach [47]. Despite the extensive development of machine learning models for blast loading conditions, their application to impact scenarios remains relatively unexplored. With reference to impact loading, Thai et al. [48] employed a gradient boosting algorithm (GBA) for RC slab failure mode classification under impact, and Shen et al. [46] optimized an artificial neural network (ANN) model for similar purposes. Dadi et al. [49] investigated the applicability of seven machine learning techniques for predicting the maximum displacements of RC beams subjected to impact loads. Zhang et al. [50], also developed machine learning models for similar purpose. However, no prior research has developed and compared the performance of ensemble and non-ensemble machine learning algorithms for the prediction of peak response of RC beams under impact loads.

This study investigates the applicability of machine learning techniques for predicting the peak response of RC beams under impact loads. In this pursuit, both individual and ensemble machine learning techniques such as support vector machine (SVM), gaussian process regression (GPR), k-nearest neigbour (KNN), gene expression programming (GEP), random forest (RF), decision tree (DT), ensemble boosted tree (EBT), adaptive boosting (AdaBoost) tree, GBA, stochastic gradient descent (SGD) and ANN—scaled conjugate gradient algorithms (SCG)—are developed. The independent parameters use for model development include beam width and depth, total and clear span of the beam, longitudinal and transverse reinforcement ratios, concrete strength, steel yield strength, deflection, and impact velocity, mass and force. To assess the effect of these parameters on peak response of the beam, shapely additive explanations (SHAP) is used.

2. Simplified collision process and peak force

The collision process, as shown in Fig. 1, is simplified by assuming a

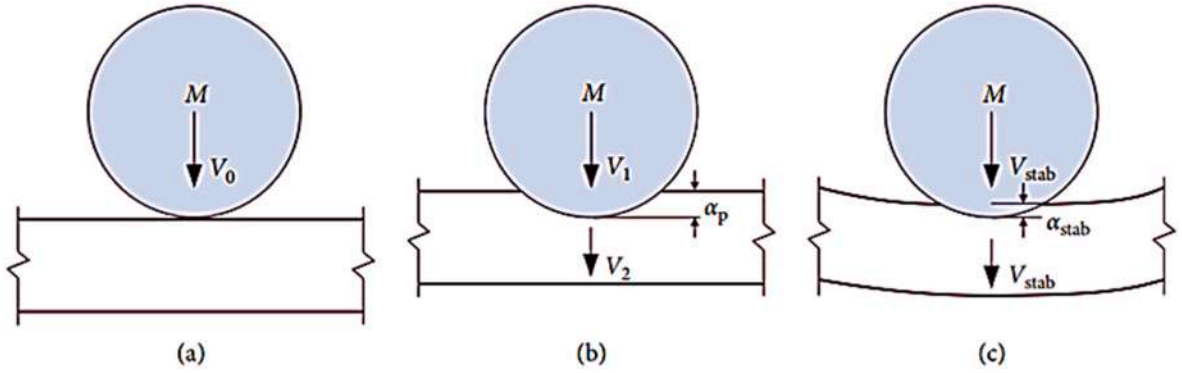

Fig. 1. Collision during the initial impact at (a) t = 0, (b) t = tp (c) t = t_s [52].

specific impact force profile based on previous experimental observations. This profile, as depicted in Fig. 2, consists of three distinct phases: a rapid increase to peak force (OA), a subsequent decrease to nearly zero (AB), a sustained global response phase with constant force (CD), and a final gradual decline to zero (DE). During the global response phase, the RC beam continues to deform elastically and plastically, absorbing most of the impact energy and reaching its maximum deflection [51,52].

The impact force between Points O and B in Fig. 2 is approximated as a triangular pulse [53]. This approximation assumes that the rise time and fall time of the pulse are equal, leading to the following relationship:

$$t_s = 2t_p \quad \text{Eq. 1}$$

where 't_s' represents the duration of the initial impact pulse and t_p denotes the time at the peak impact force. The peak impact force typically occurs within the first one or 2 ms of the impact, with most of the beam deformations concentrated at the point of impact due to inertial effects [53]. The deformations beneath the contact surface are complex and may result in minor cracks.

During the impact process, stress waves propagate through the beam, with shear waves having the most significant influence on the local response. Prior studies have shown that the effective participation mass is initially smaller than predicted by Biggs' method due to the propagation of shear waves. Additionally, as impact velocity increases, the time to reach peak impact force decreases, leading to a further reduction in effective mass [54].

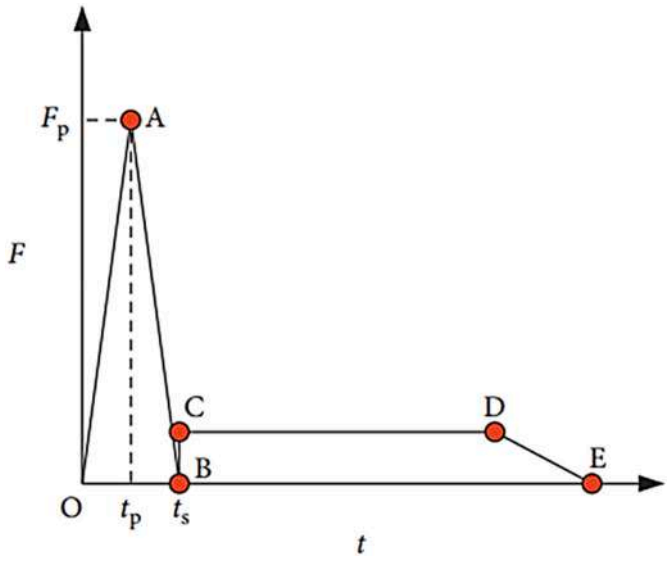

Fig. 2. Simplified time history of impact load [52].

3. Data acquisition and description

The key parameters affecting the peak response of RC beams under impact loading include concrete compressive strength, longitudinal tensile reinforcement, yield strength of the longitudinal reinforcement, vertical shear reinforcement, impactor mass, drop-weight velocity, and beam geometry [55–58] To investigate the influence of these parameters on peak response and establish a comprehensive understanding of RC beam behavior under impact loading, a comprehensive database of 145 RC beams subjected to mid-span drop-weight impact was assembled from existing literature [59–70]. These simply supported beams with rectangular cross-sections were impacted using either spherical or flat-nosed impactors at velocities ranging from 1 to 16 m/s, with most tests falling within the low-velocity impact region. Impact masses varied from 100 to 1800 kg, with most concentrated between 300 and 600 kg. The RC beam widths and heights spanned from 100 to 300 mm and 150–500 mm, respectively, while beam spans ranged between 1000 and 5000 mm. Longitudinal tensile reinforcement ratios varied from 0.25 % to 3.25 %, and shear reinforcement ratios ranged from 0 to 1.4 %.

Table 1 presents a comprehensive overview of the statistical metrics applied to the data. It entails a range of descriptive statistics—such as range, mean, variance, standard deviation, median, mode, skewness, and kurtosis. For instance, L_n shows the greatest range (4000 mm), while p_v exhibits the smallest range (1.4). Higher values of variance and standard deviation for L and F_p suggests greater variability of the two parameters. Skewness shows distribution's asymmetry, with positive values indicating a right skew and vice versa. Kurtosis is the "tailedness" of the distribution, with values greater than 3 suggesting heavier tails. The values for skewness and kurtosis for most parameters are low, implying distributions that are nearly normal.

The provided image in Fig. 3 is a heatmap representation of a Pearson correlation coefficient matrix, which visually displays the correlation between different pairs of variables used in a study. The variables are listed along both axes, indicating their relationships. The color intensity and hue represent the strength and direction of the correlations, with a scale ranging from −0.655 to 1, where values near −1 show a strong inverse correlation, values near 1 shows a strong positive correlation, and values near 0 indicate no correlation.

The dendrograms on the top and left axes visualizes the hierarchical clustering of variables based on their correlation coefficients. Variables with similar correlation patterns are grouped together, facilitating identification of potential relationships. The color transitions from blue to red, with blue representing strong negative correlations and red representing strong positive correlations. Shades of purple indicate poor correlations. This heatmap is valuable for deciphering the interrelationships between variables. It simplifies the process of understanding the patterns in the data. Fig. 4 display the marginal histograms of various input factors, depicting their relation with the beam's capacity under impact force.

Table 1
Statistical analysis of the obtained data.

Parameter	Units	Range	Mean	Variance	St. Dev.	Median	Mode	Skewness	Kurtosis
b	mm	200.0	168.7	1060.5	32.6	150.0	150.0	0.6	1.0
h	mm	350.0	294.1	7066.2	84.1	250.0	250.0	0.9	0.2
L	mm	4600.0	2593.0	636874.2	798.0	2400.0	2400.0	1.8	5.2
L_n	mm	4000.0	2128.0	469958.8	685.5	2000.0	2000.0	1.7	4.7
A_s	–	22.0	39.7	36.3	6.0	40.0	40.0	0.0	−0.2
f_c'	MPa	25.7	33.8	59.5	7.7	34.7	40.0	−0.3	−1.5
A_s	mm^2	1767.0	721.8	253042.4	503.0	603.0	265.0	1.1	0.4
f_y	MPa	175.0	427.0	4001.1	63.3	404.0	520.0	0.4	−1.4
A_s'	mm^2	1924.0	564.8	234762.6	484.5	402.0	265.0	1.8	2.6
f_y'	MPa	175.0	427.0	4001.1	63.3	404.0	520.0	0.4	−1.4
f_{vy}	MPa	550.0	304.4	7227.6	85.0	295.0	295.0	−1.4	7.8
p_v	%	1.4	0.4	0.2	0.4	0.2	0.8	1.5	1.3
V	m/s	14.2	5.1	4.0	2.0	5.0	5.0	1.2	4.0
M	Kg	1550.0	406.6	65899.2	256.7	300.0	300.0	3.6	13.8
F_p	KN	2258.0	817.4	457908.1	676.7	751.0	0.0	0.4	−1.0

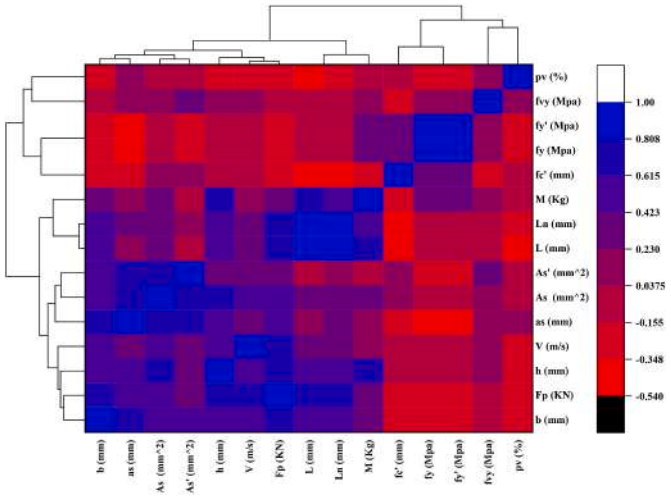

Fig. 3. Heat map showing the correlation between different parameters.

4. Proposed machine learning techniques

4.1. Non-ensemble machine learning techniques

4.1.1. Gene expression programming

GEP is a grey-box model that constructs non-linear expression trees (ETs) to model complex phenomena like peak dynamic response values. It operates by encoding genotypes and phenotypes separately, facilitating the evolution of genetic variations. Each gene in GEP contains variables, arithmetic operations, and constants, with crucial empirical information stored in chromosomes. These chromosomes are interpreted using Karva language, which simplifies the conversion between gene sequences and ETs, enabling efficient phenotype inference [71–73].

The structure of GEP genes mimics biological open reading frames, starting with a codon and possibly containing noncoding regions that enhance solution stability and genetic diversity. The output is generated through the interaction of sub-ETs, where their arrangement captures the non-linear relationships required for accurate predictions [74]. This gene-based evolutionary process allows GEP to explore a wide solution space, improving the precision and reliability of predictions [75].

Key hyperparameters used in this GEP model include the number of chromosomes, head size, and the number of genes. Constants within each gene were floating-point values with expression complexity capped at 10, and ephemeral random constants were set between −10 and 10. The mutation rate was fixed at 0.0015, and 'addition' was selected as the linking function. Table 2 summarizes the hyperparameters, optimized to ensure model accuracy. Figs. 5 and 6 illustrate the GEP model's flowchart and its predictive performance, respectively. The developed equation for peak response is given in Eq. (2). The symbols used in GEP equation is provided in Table 3.

$$Peak\ response = W + X + Y + Z \quad \text{Eq. 2}$$

$$W = E \times M + \frac{CG(1+E)}{(-0.33-(G+(D-12.02)(N-I))} \quad \text{Eq. 3}$$

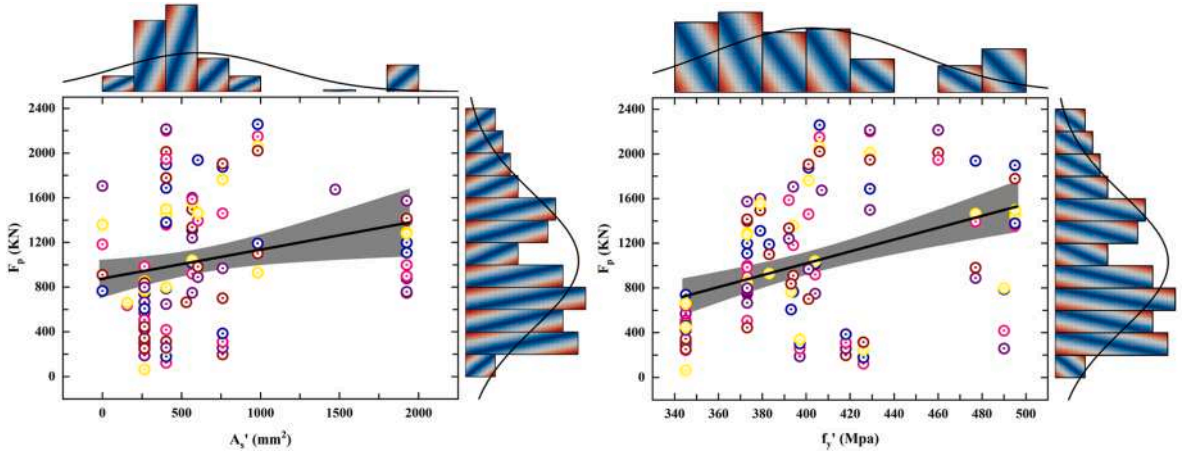

Fig. 4. Marginal histogram showing the distribution of data.

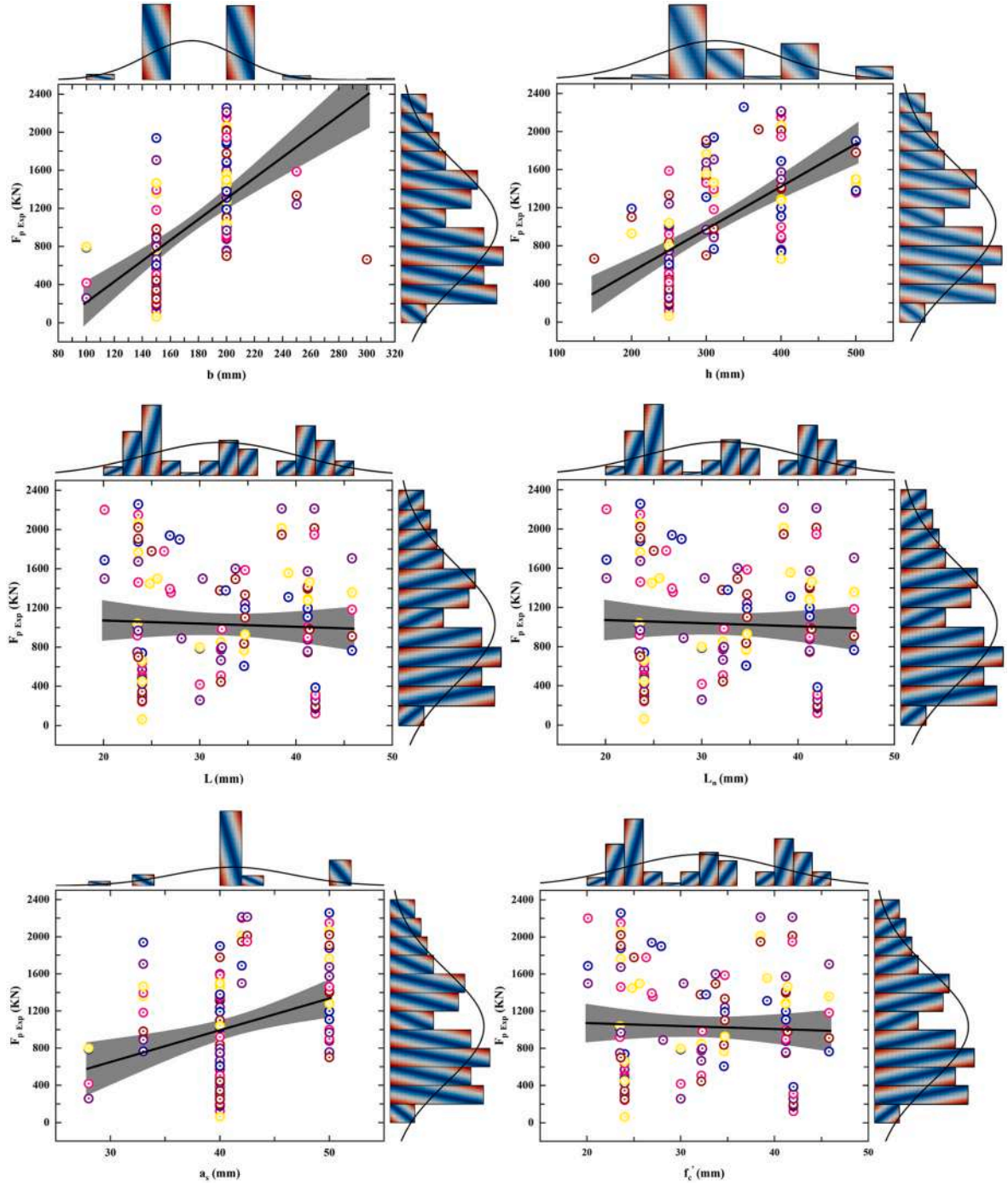

Fig. 4. (*continued*).

$$X = \left[K - J - \frac{E}{A} + 0.64(1.812L) + \frac{K}{F}\right] + B + A \quad \text{Eq. 4}$$

$$Y = \frac{L + \left(-1.23 + \frac{46.06 - K(C - B + 2I + G+)}{C}\right)}{C} + H;\ Z = \frac{(-2J/9.15 \times H)}{(G + J - 6.01E + 8.92)} \quad \text{Eq. 5}$$

4.1.2. K-nearest neighbor

KNN algorithm is a widely used classification method known for its simplicity and adaptability. KNN identifies the closest neighbors to the data, classifying it as a nonparametric technique. This method does not rely on assumptions about data distribution, rather, it categorizes new data based on proximity to existing data points, similar to clustering techniques. The algorithm classifies new objects based solely on their attributes and training samples, without relying on any underlying model [76,77].

KNN is considered a lazy learning method because it stores training instances and predicts labels for test instances based on majority voting among the K nearest neighbors. The selection of *K* significantly impacts prediction quality, with smaller K values leading to higher prediction variance [78]. Determining the optimal *K* value often involves parameter optimization techniques such as cross-validation. KNN efficiency decreases with large datasets; therefore, various acceleration algorithms, such as template trees, have been proposed to address this challenge [79]. These algorithms, using triangle inequality for distance

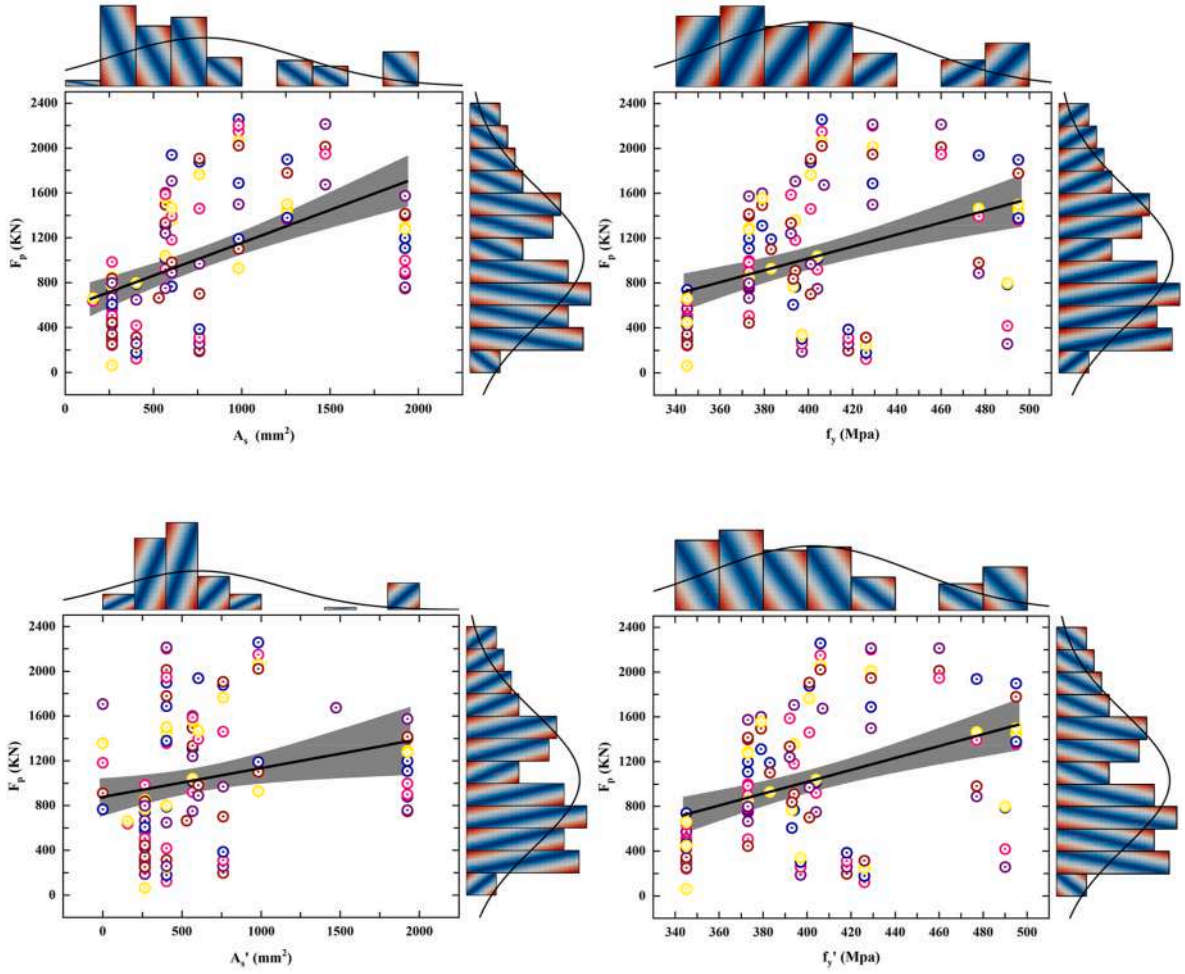

Fig. 4. (*continued*).

Table 2
Parameters employed for GEP model development.

Training and validation	85 % and 15 %
Chromosomes	100
Head size	12
Operators	+, −, × , ÷, power
Genes	4
Constant/gene	10

computation, enhance the matching process by identifying samples that are closer or distant from each other. The flowchart for the KNN method is presented in Fig. 7.

For regression problems, the K nearest neighbors are calculated using Eq. (6):

$$M = \frac{1}{K}\sum_{i=1}^{N} m_i \quad \text{Eq. 6}$$

In above equation, m_i shows the target value for the i_{th} training instance, and m is the output value for the new example. Different distance metrics (see Eq. (7) to Eq. (10)), like quadratic euclidean, euclidean, manhattan, and chebyshev, are used to calculate the distance between neighbors.

$$D(y,q) = \sqrt{\sum (y-q)^2} \quad \text{Eq. 7}$$

$$D(y,q) = (y-q)^2 \quad \text{Eq. 8}$$

$$D(y,q) = \mathrm{Abs}(y-q) \quad \text{Eq. 9}$$

$$D(y,q) = \mathrm{Max}(|y-q|) \quad \text{Eq. 10}$$

In the above equations, $D(y,q)$ is the distance metric, y is the new point and q is the training instance. The classification precision depends mainly on the metric used to calculate distances [80].

In this study, the distance metrics used to estimate the peak response of the RC beam is Euclidean, as given in Eq. (7), with the number of neighbors set to 2 (see Table 4). The curve fitting and scatter plot for the developed model is shown in Fig. 8.

4.1.3. Decision tree algorithm

DTs are known for their simplicity, interpretability, and effectiveness in handling large datasets, making them prominent categorization algorithms. They represent discrete predictive functions as a series of decision points leading to terminal nodes labeled with respective classes. Each attribute of an observation is presented by a node. The classification process involves traversing the tree to reach a leaf node, which corresponds to the predicted class [81,82]. The splitting of the decision nodes is shown in Fig. 9.

Classification in DTs occurs in two phases: tree building and pruning. Tree building is a top-down approach that recursively partitions the tree until all data items belong to the same class. Pruning improves prediction accuracy by minimizing overfitting. Various algorithms, such as ID3, C4.5, C5.0, and CART, are known for their high speed, learning ability, and simplicity. These algorithms leverage the DT's ability to handle mixed-type data and provide decision rules that can be directly

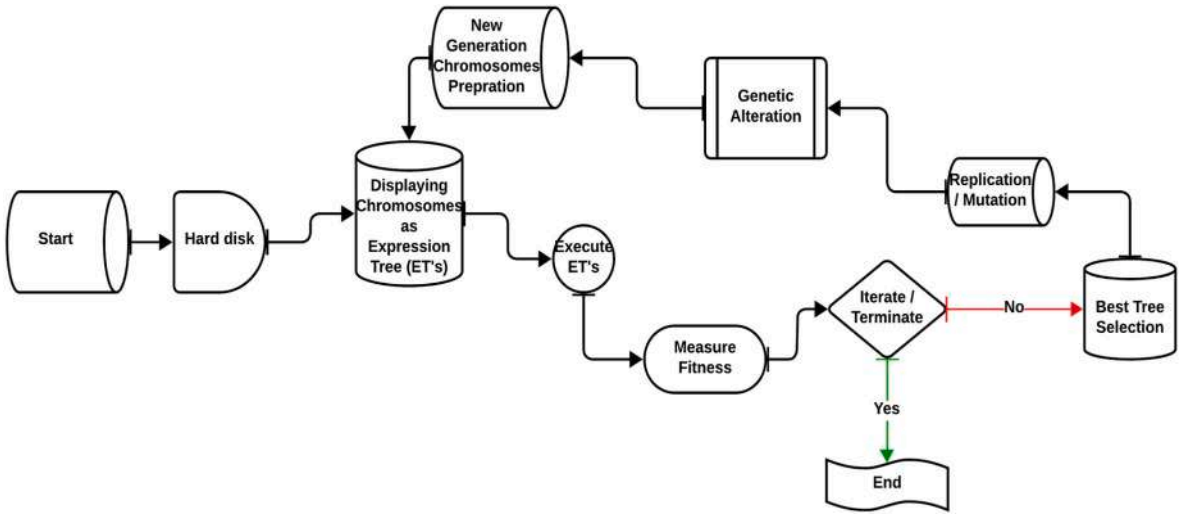

Fig. 5. Flowchart showing the core processes in GEP.

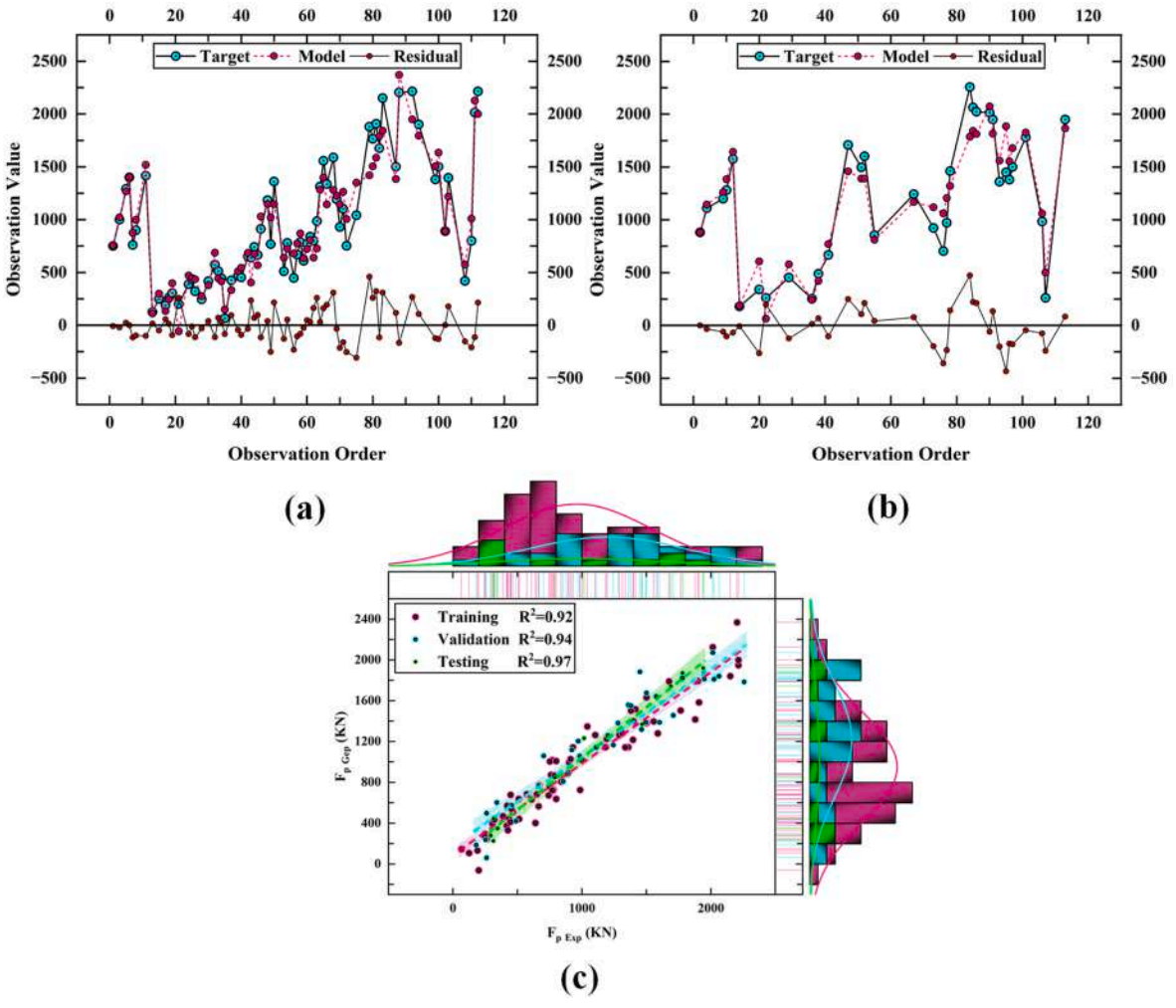

Fig. 6. GEP model (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the whole data.

converted into user-friendly "if-then" statements, enhancing interpretability [82].

Due to the increasing size of datasets, interest in fast DT learning algorithms is growing. Developing fast tree-growing algorithms is crucial, with a focus on searching restricted model spaces or using powerful search heuristics. Most modern algorithms use an impurity-based heuristic, such as information gain, to measure the purity of resulting subsets after splitting the data. The time complexity of standard DT learning algorithms is O ($m \cdot n^2$). Independent information gain (IG) is a key concept in optimizing the DT learning process which aims to

Table 3
Symbols used in GEP equation.

Symbol	Parameter	Symbol	Parameter
A	Width of the beam	H	Yield strength of steel
B	Depth of the beam	I	Area of top reinforcement
C	Total span of beam	J	Yield strength of stirrup
D	Clear span of beam	K	Area of stirrup
E	Steel ratio	L	Steel reinforcement ratio
F	Compressive strength	N	Velocity of the load
G	Area of bottom steel	M	Mass of the impact body

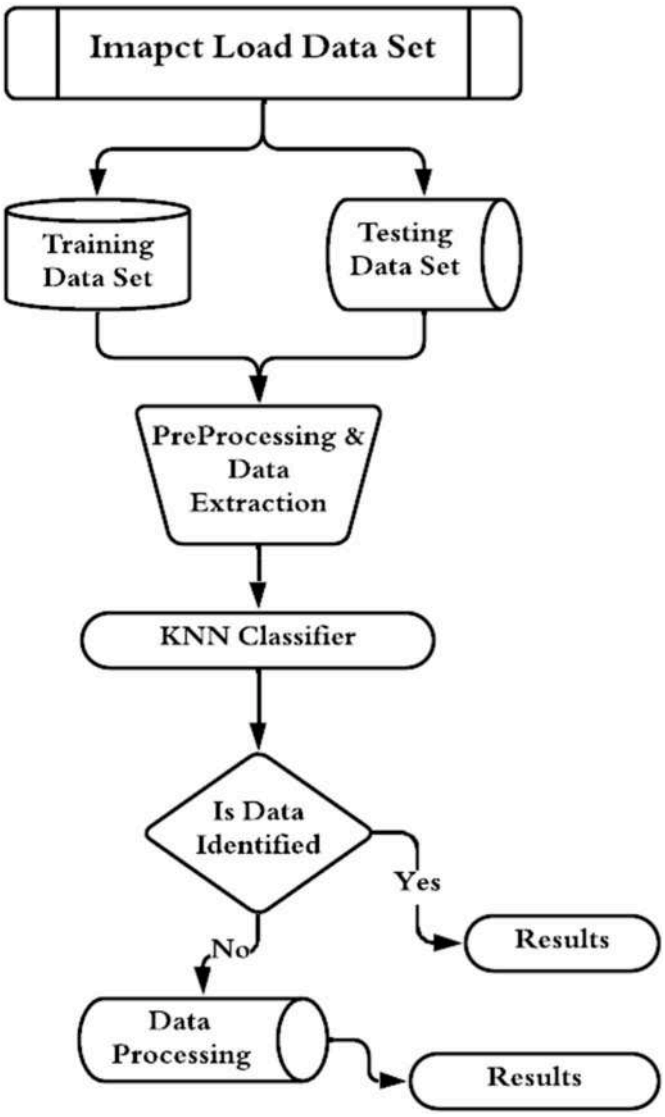

Fig. 7. Flowchart of KNN.

Table 4
Parameters involved in KNN model development.

Total neighbors	Training and validation	Metric	Weight
2	80 % and 20 %	Euclidean	Uniform

reduce computational costs. IG quantifies the reduction in entropy in the original dataset S.

$$IG(S,Y) = Entrophy(S) - \sum_{y} \frac{|S_y|}{|S|} Entrophy(S_y) \quad \text{Eq. 11}$$

In Eq. (11), S is the set of training instances, Y is an attribute and y is a value, S_y is a subset of S consisting of the instances with $Y = y$, and (S) is the entropy which is defined as:

$$Entrophy(S) = -\sum_{i=2}^{|C|} P_s(c_i)\log P_s(c_i) \quad \text{Eq. 12}$$

In Eq. (12) $P_s(c_i)$ is the estimated by the percentage of instance belonging to c_i in S and $|C|$ is the number of classes.

DTs are valued for their clear structure which makes them easy to understand and implement. They are versatile in handling both numerical and categorical data types, and their robustness against outliers makes them suitable for a wide range of datasets [82].

However, DTs face challenges when dealing with linear relationships. When input features and outcomes have a linear association, DTs use splits to approximate this relationship and produce a step function. This lack of smoothness means that even small changes in input features can lead to large variations in predicted outcomes, resulting in inefficiencies [83]. Parameters such as pruning, splitting, and binary trees are crucial in model development and are summarized in Table 5. The curve fitting and scatter plot for the developed model are shown in Fig. 10.

4.1.4. Stochastic gradient descent algorithm

SGD is frequently utilized to determine the weights and biases in neural networks. The objective function in SGD is expressed with v and u. The vector of partial derivatives of f with respect to u and v is represented as ∇Y. The exponential moving averages of the gradient ($x_{1,j}$, $y_{1,j}$) and the squared gradient ($x_{2,j}$, y_2) are updated using two hyperparameters, δ and α. These moving averages approximate the first and second moments of the gradients, with the hyperparameters governing the rates of exponential decay [84]. The steps involved in the SGD method are as follows.

Step 1. calculate objective function K (u_j, Y_j) and its gradient.

$$\nabla K_j = \left[\frac{\varnothing K(u_j, Y_j)}{\varnothing u_j}, \frac{\varnothing K(u_j, Y_j)}{\varnothing Y_j}\right] \quad \text{Eq. 13}$$

Step 2. Calculate the exponential moving averages of the gradient and squared gradient using:

$$x_{1j} = \alpha_1 x_{1j} + (1-\alpha_1)\nabla K_j$$

$$x_{2j} = \alpha_2 x_{2j-1} + (1-\alpha_2)\nabla K_j \odot \nabla K_j) \quad \text{Eq. 14}$$

Here, $\nabla K_j \odot \nabla K_j$ indicates the element wise multiplication.

Step 3. Correct the exponential moving averages:

$$\widehat{x}_{1j} = \frac{x_{1j}}{[1-(\alpha_2)^j}$$

$$\widehat{x}_{2j} = \frac{x_{2j}}{(1-(\alpha_2)^j} \quad \text{Eq. 15}$$

In the step 3: the superscript 'j' in $(\alpha_1)^2$ and $(\alpha_2)^2$ indicates α_1 and α_2.

Step 4. Determine the damage indices and unknown force:

$$\frac{u_j}{Y_j} = \frac{u_{j-1}}{Y_{j-1}} - \frac{\delta_j \bullet \widehat{x}_{1j}}{(\sqrt{\widehat{x}_{2j}} + \gamma)}$$

$$\delta_j = \delta \frac{\sqrt{1-(\alpha_2)^j}}{(1-\alpha_1)^j} \quad \text{Eq. 16}$$

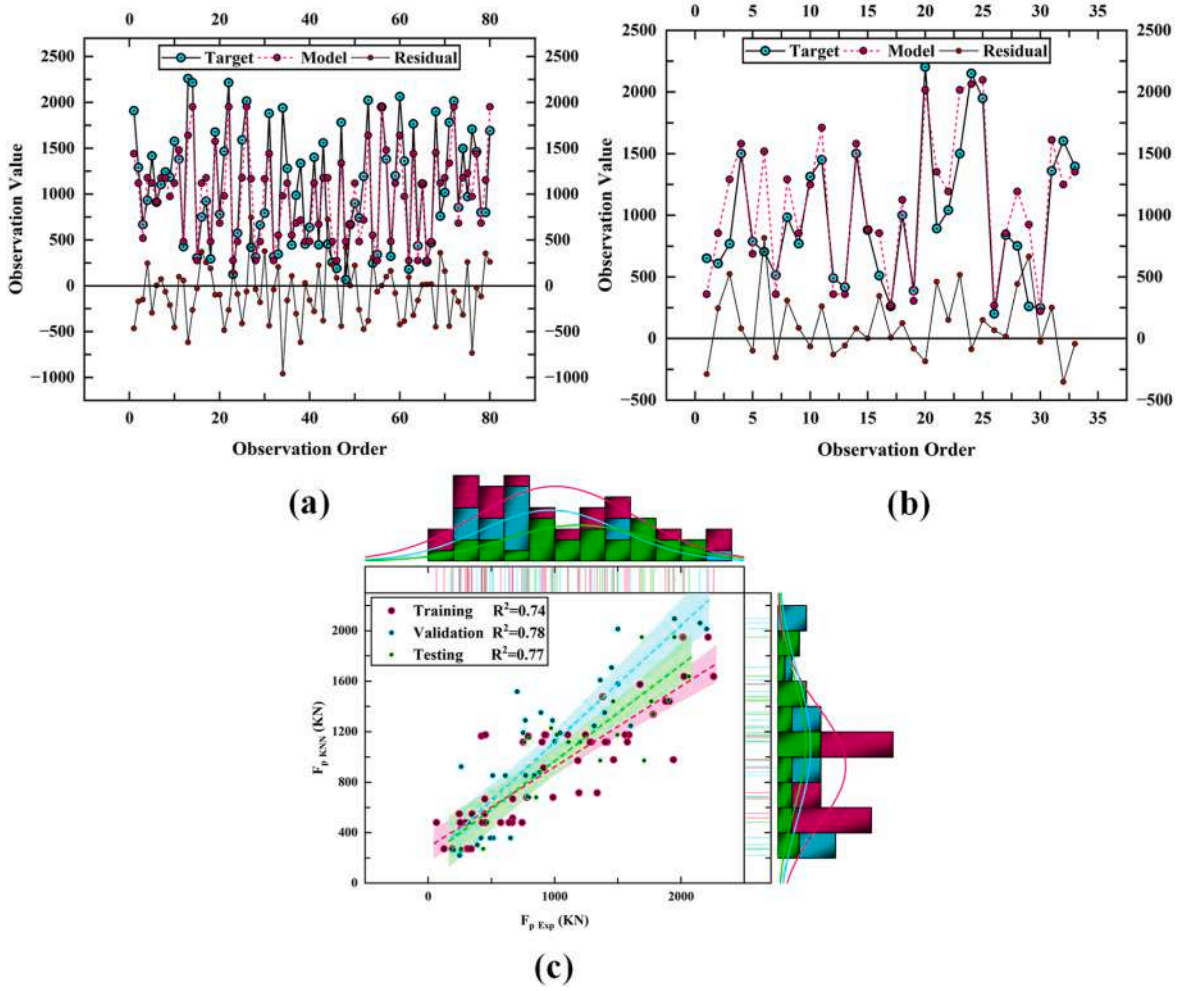

Fig. 8. KNN model (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the whole data.

Step 5. Repeat Steps 1 to 4 until the convergence criterion '*Tol*' is met.

$$\frac{\|u_j - u_{j-1}\|}{\|u_{j-1}\|} + \frac{\|Y_j - Y_{j-1}\|}{\|Y_{j-1}\|} < Tol \qquad \text{Eq. 17}$$

In this study, we set $\propto_1 = 0.9$, $\propto_2 = 0.99$, $\delta = 0.001$, and $\gamma = 1 \bullet 0^{-8}$.

The rate at which SGD converges depends on its ability to accurately estimate the true gradient, a task prone to noise interference. Gradual reduction in the learning rate facilitates a slow decrease in error, while rapid declines can impede convergence. Achieving an optimal convergence rate, under specific conditions involves a proportional decay of the learning rate with the iteration number, t. This ensures a corresponding reduction in residual error, denoted as $E[\rho \sim t^{-1}]$ [77,84,85]. In this study, the parameters utilized in model development are classification loss function (Hinge) and regression loss function (squared loss) (see Table 6). The visual depiction of the model's curve fitting and scatter plot are shown in Fig. 11.

4.1.5. Support vector machines

SVMs, initially suggested by Vapnik, have garnered attention in contemporary data-driven applications. Originally conceived as linear classifiers, SVM have since been extended to non-linear classification tasks and regression. SVMs map input features to a higher-dimensional feature space. A hyperplane is constructed to maximize the margin between two parallel hyperplanes, a concept known as the maximum margin classifier (see Fig. 12). The primary aim is to minimize empirical classification errors while maximizing the geometric margin to enhance generalization performance [86,87].

For a training dataset $\{y_m\}_m^N = 1$ with N samples, the SVM model is formulated as:

$$f(x) = W^T \phi(x) + b \qquad \text{Eq. 18}$$

In Eq. (18), ϕ: x→ϕ(x) ∈ $\mathbb{R}^H$ signifies a non-linear function. Initially, assuming linear separability, ϕ(x) is defined as ϕ(x) = x. The model parameters include w, a weight vector normal to the hyperplane, and b, the hyperplane bias [88].

The optimization problem, subject to soft-margin constraints, aims to define a hyperplane which maximize the margin between training data. This optimization is achieved using the Lagrange multipliers method, resulting in a cost function.

$$\mathrm{M}\left(\{b_g, b^*{}_g\}_{g=1}^N\right) = -\frac{1}{2}\sum_{j=1}^{N}(b_i - b_i^*)(b_i - b_i^*)K\left(b_j - b_j^*\right) - \in \sum_{i=1}^{N}(b_i - b_i^*) + \sum_{i=1}^{N}(b_i - b_i^*)y_i \qquad \text{Eq. 19}$$

In Eq. (19), $\{b_g, b^*{}_g\}_{g=1}^N$ and $K\left(b_j - b_j^*\right)$ denote the Lagrange multipliers and Kernel function, respectively.

$$K(b_i, b_j) = \langle \varnothing(b_j) | \varnothing(b_j) \rangle. \qquad \text{Eq. 20}$$

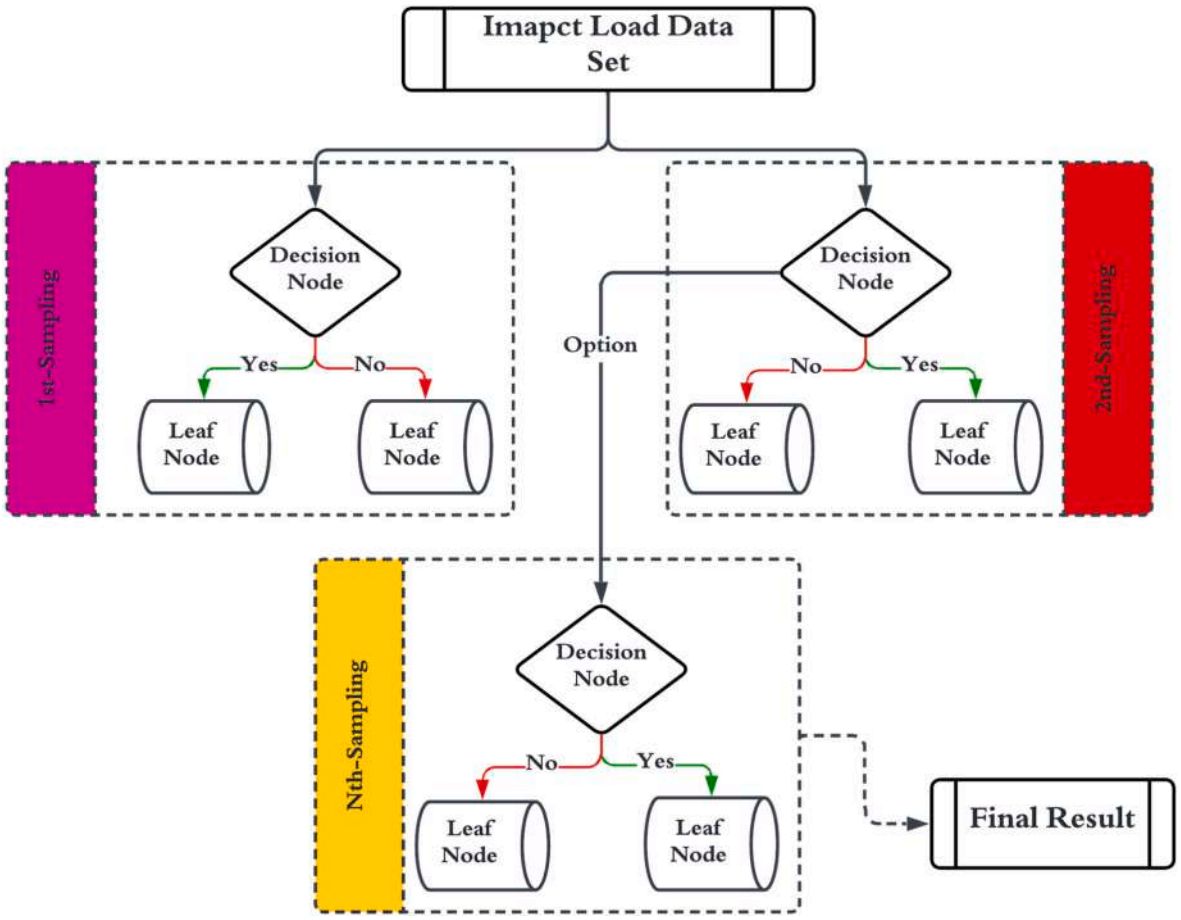

Fig. 9. Division of decision nodes.

Table 5
Parameters used in DT modeling.

Pruning	Min samples split	Training and validation	Max depth	Splitting criteria	Binary trees
Post-pruning	2	80 % and 20 %	10	Gini impurity <0.01	Yes

Simplifying the optimization of this cost function involves introducing kernel notation. SVMs directly define the kernel as a function of the input-feature vector.

$$K_{linear}(y,\acute{y}) = y,\acute{y} \quad \text{Eq. 21}$$

$$K_{polynomial} = (\gamma y\acute{y} + r)^{\rho} \quad \text{Eq. 22}$$

$$K_{RBF}(y,\acute{y}) = \exp\left(-\gamma \| y - y^{'} \|^2\right) \quad \text{Eq. 23}$$

$$K_{sigmoid}(y,\acute{y}) = \tanh(\gamma y\acute{y} + r). \quad \text{Eq. 24}$$

Once we estimate $\{\widehat{b}_g, \widehat{b}^{*}_{g}\}_{g=1}^{N}$, the margin can be inferred as:

$$\widehat{W} = \sum_{g=1}^{N} \left(\widehat{b}_g - \widehat{b}_g^{*}\right) \varnothing \left(y_g\right) \quad \text{Eq. 25}$$

Such as $f(x)$ can be directly estimated as:

$$\widehat{f}(x) = \sum_{g=1}^{N} \left(\widehat{b}_g - \widehat{b}_g^{*}\right) K(y_i, y) + \widehat{b} \quad \text{Eq. 26}$$

By maximizing the defined cost function, the computation of 'b' can be skipped by preprocessing and centralizing the data, thus enforcing the bias to be zero [88]. Table 7 summarizes the parameters employed for model development. Fig. 13 shows the curve fitting and scatter plot generated by the model.

4.1.6. Gaussian process regression algorithm

Recently, GPR, a kernel-based probabilistic model, has gained significant popularity in nonlinear system modeling. To elucidate GPR further, let's define the training dataset as $\{(x_i, y_i); i = 1, 2, \ldots, T\}$, where $x_i \in \mathbb{R}^d$ represents the d-dimensional predictors and $y_i \in \mathbb{R}$ denotes the target variable. In contrast to a basic linear regression model $y = x_T\beta + \varepsilon$, where ε follows a normal distribution (0, σ^2), GPR integrates latent variables and explicit basis functions (b) to interpret the target variable. These latent variables within the Gaussian process, denoted as (x_i), collectively follow a joint Gaussian distribution. The covariance structure of (x_i) governs the smoothness of the target, with the basis function aiding in projecting predictors onto the feature space [89,90]. The visual depiction of GPR process is shown in Fig. 14.

In essence, a Gaussian process is characterized by its covariance and mean functions. The covariance, denoted as $(x, x') = \text{Cov}\,[l(x), l(x')]$, and the mean, denoted as $m(x) = \mathbb{E}?[l(x)]$, play pivotal roles. Consequently, GPR is formulated as $y = (x)\ T\beta + l(x)$, where $l(x)$ follows a Gaussian process with mean 0 and covariance $k(x, x')$, and $b(x) \in \mathbb{R}^p$. A hyperparameter θ is employed to parameterize the covariance function $(x,$

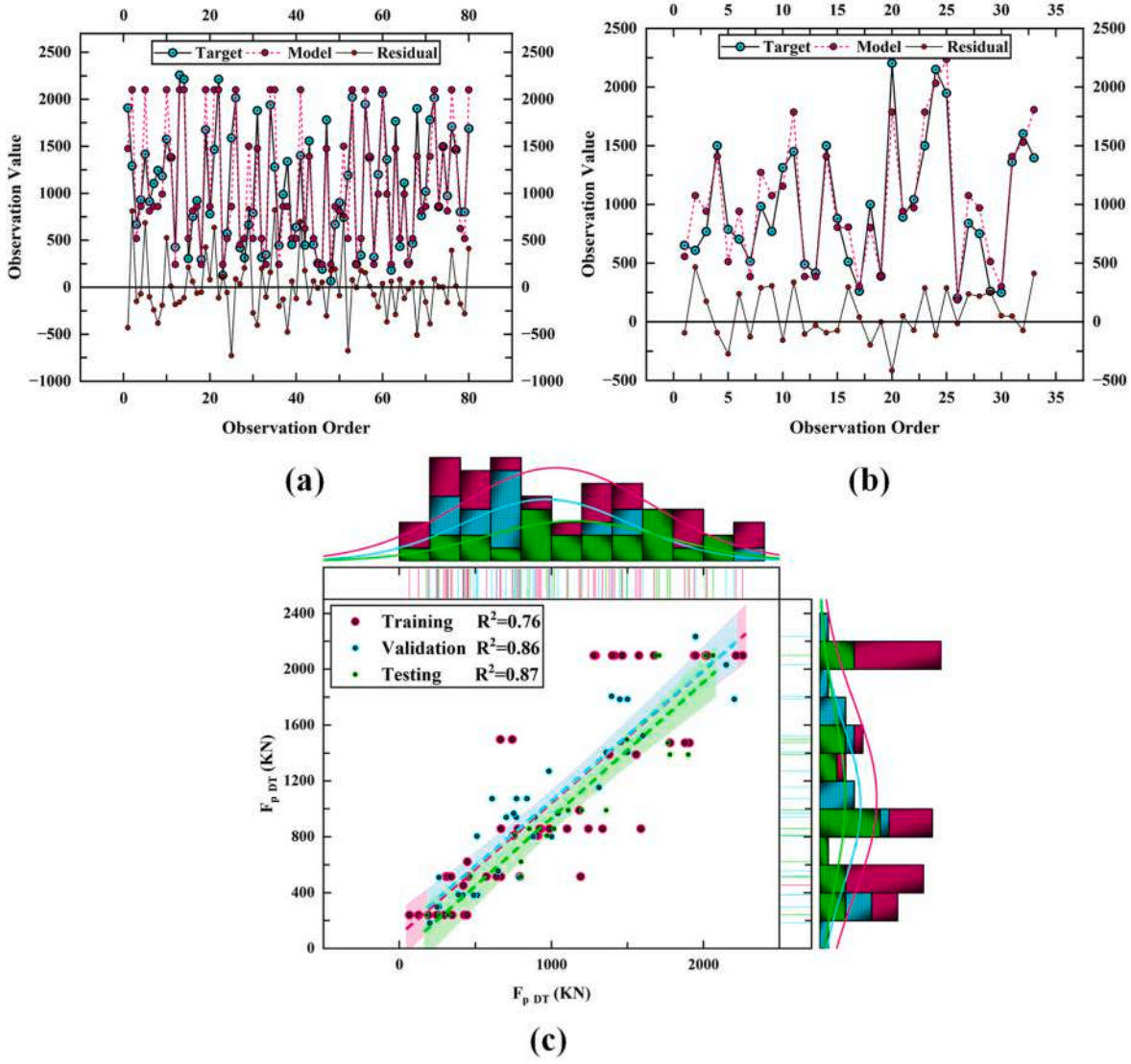

Fig. 10. DT model (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the whole data.

Table 6
SGD modeling parameters.

Model Type	Loss Function	Regularization	Strength (α)	Learning rate	Shuffling	Starting Value	Training and validation
Regression	Huber Loss	Elastic Net (L1/L2)	1.00E-06	0.01	Yes	0.01	75 % and 25 %

x') as $k(x, x'|\theta)$. During the training of a GPR model, an algorithm typically estimates parameters such as β, σ^2, and θ [91,92].

The kernel function in Gaussian regression characterizes the correlation among variables. Different kernel functions can lead to various fitting outcomes, with isotropic kernels being favored over nonisotropic ones. The Matern 5/2 kernel was chosen which is provided in Eq. (27).

$$\textit{Isotropic Matern}\frac{5}{2}: k(x_i, x_j|\theta) = \sigma_f^2\left(1+\frac{\sqrt{5}r}{\sigma_l}+\frac{\sqrt{5}r^2}{3\sigma_l^2}\right)e^{-\frac{\sqrt{5}r}{\sigma_l}} \quad \text{Eq. 27}$$

For isotropic kernels, σ_l denotes the length scale, σ_f shows the signal deviation, $r = \sqrt{(x_i - x_j)'(x_i - x_j)}$ and $\alpha > 0$ signifies the scale parameter. Ensuring the positivity of σ_l and σ_f can be accomplished through θ, where $\theta = (\theta_1, \theta_2) = (\log \sigma_l, \log\sigma_f)$. Moreover, among the available basis functions—including the empty, constant, linear, and pure quadratic basis function—constant basis functions is selected for the current study.

Gaussian regression is known for its strong generalization capabilities; however, it falls short in adaptability when data exhibits sudden variations in different areas. This limitation makes a single kernel function insufficient for accurate fitting [93,94]. Table 8 summarizes the parameters employed in model development. Fig. 15 displays the resulting curve fitting and scatter plot for the trained model.

4.1.7. Artificial neural networks

ANNs are essential tools for solving prediction problems by mimicking the brain's learning processes. Designed to reflect the brain's structure, ANNs use interconnected "neurons" to identify complex, nonlinear patterns, making them effective in a wide range of forecasting applications. An ANN generally includes an input layer, hidden layers, and an output layer. Neurons in these layers are linked by adjustable weights. The network processes inputs by multiplying them with these weights, summing them to produce a net output, and then applying an

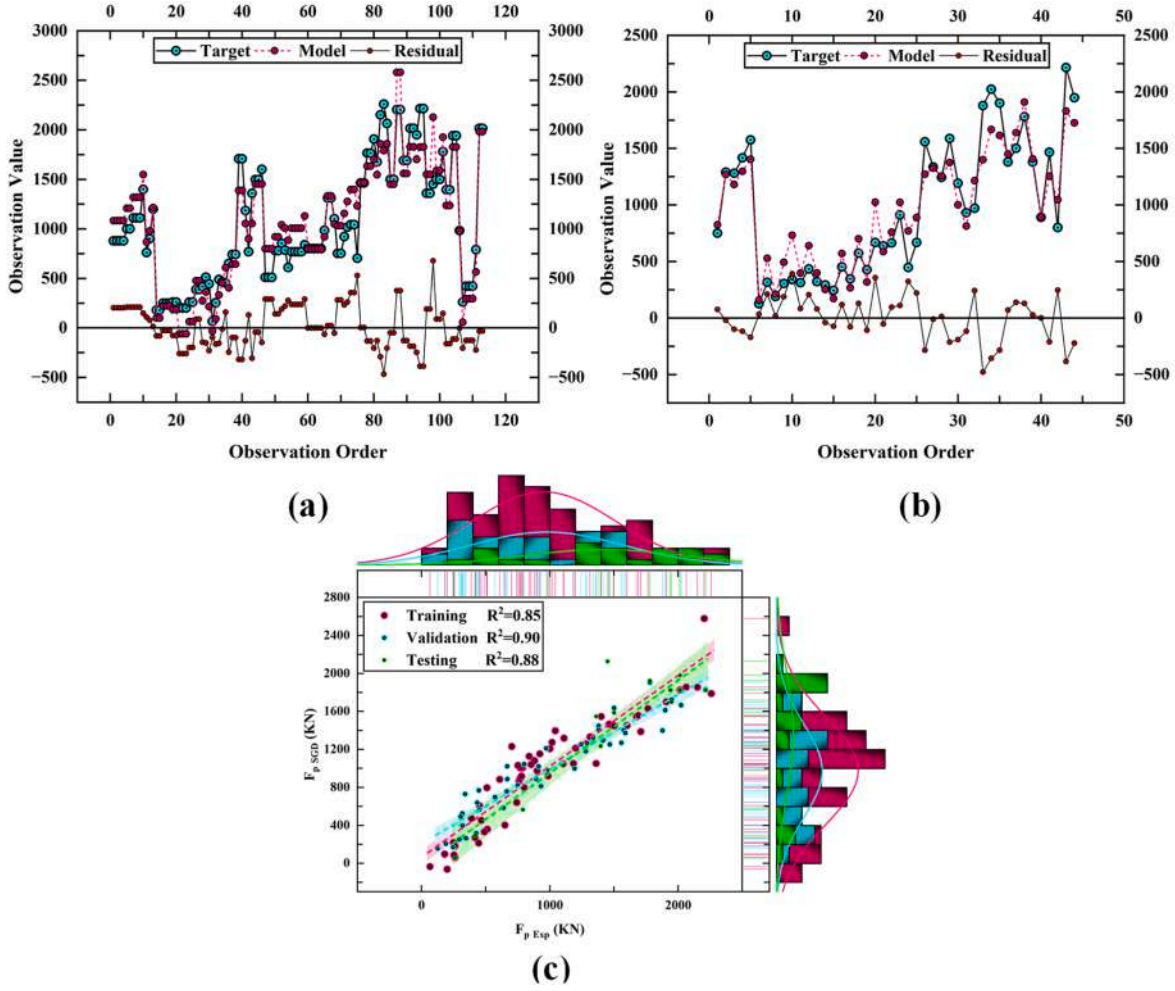

Fig. 11. SGD model (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the whole data.

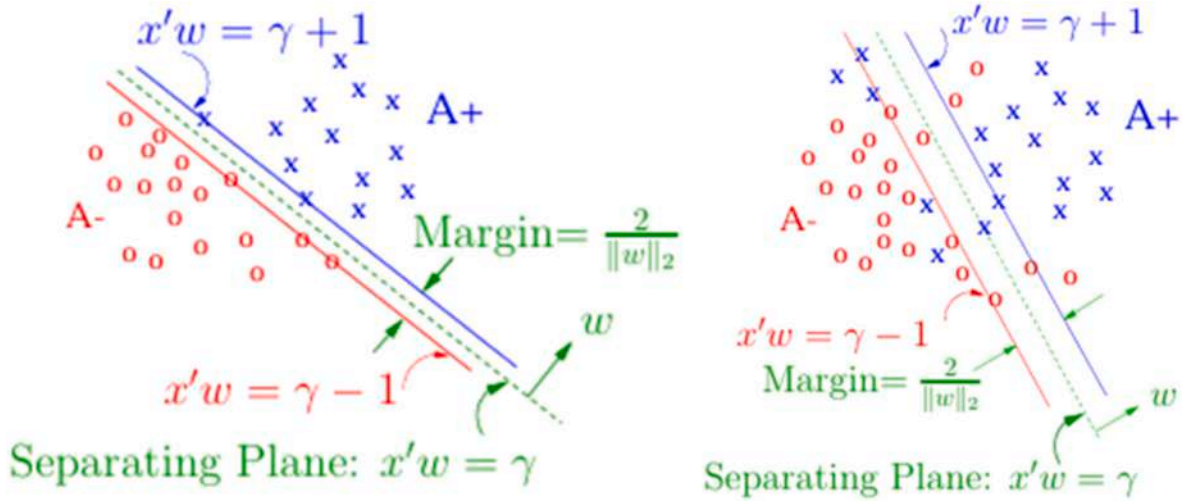

Fig. 12. Visual depiction of separating plane.

activation function to generate the final output [95,96]. Back-propagation neural networks (BPNNs), a common method, iteratively adjust weights and biases to reduce the difference between the network's predictions and the actual outcomes. Eq. (28) and Eq. (29) delineate the weighted sums of inputs and outputs, respectively.

$$[net]_j = \sum_{i=1}^{n} w_{ij} x_i + b \quad \text{Eq. 28}$$

$$[out]_j = f(net)_j = \frac{1}{1 + e^{-\propto (net)_j}} \quad \text{Eq. 29}$$

Table 7
Parameters involved in SVM model development.

Estimation rate	20,000 observations/second
Training and validation	60 % and 40 %
Preset	Gaussian
Kernel Type	Gaussian SVM
Box Constraint, Epsilon	Automatic
Data Standardization	Enabled
Kernel Scale	Automatic

In the above equations, $(net)_j$ stands for the weighted sum. W_{ij} represents the weight connecting the i_{th} neuron in the previous layer to the j_{th} neuron.

In Fig. 16, the neurons in the hidden layers is chosen to optimize the match between ANN computations and predicted outputs while minimizing errors, with the input layer neurons corresponding to the study parameters. However, increasing the number of neurons or layers can lead to overfitting. To combat this, early stopping techniques are used to enhance model generalization and reduce overfitting risks. The data is split into training, testing, and validation sets to evaluate the model's performance [97,98].

The training process begins with initialization of weights and biases which is followed by iterative adjustments based on initial factor mu, decrease factor mu-dec, and increase factor mu-inc. Training goes on for a specified number of iterations (epochs), with the network's performance assessed using mean squared error (MSE).

4.1.8. Scaled conjugate gradient algorithm

The SCG algorithm is effective technique for training ANNs. Like the levenberg marqaurdt (LM) method, SCG uses curvature information from the network to improve learning [22]. However, unlike LM, SCG

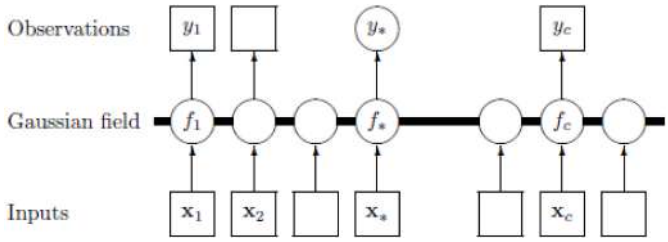

Fig. 14. Visual depiction of GPR process.

Table 8
Parameters involved in SVM model development.

Hyperparameter	Typical Values
Kernel	Isotropic matern (5/2)
Length scale	1
Number of restarts	5
Max iterations	1000
Warm start	TRUE
Training and validation	60 % and 40 %

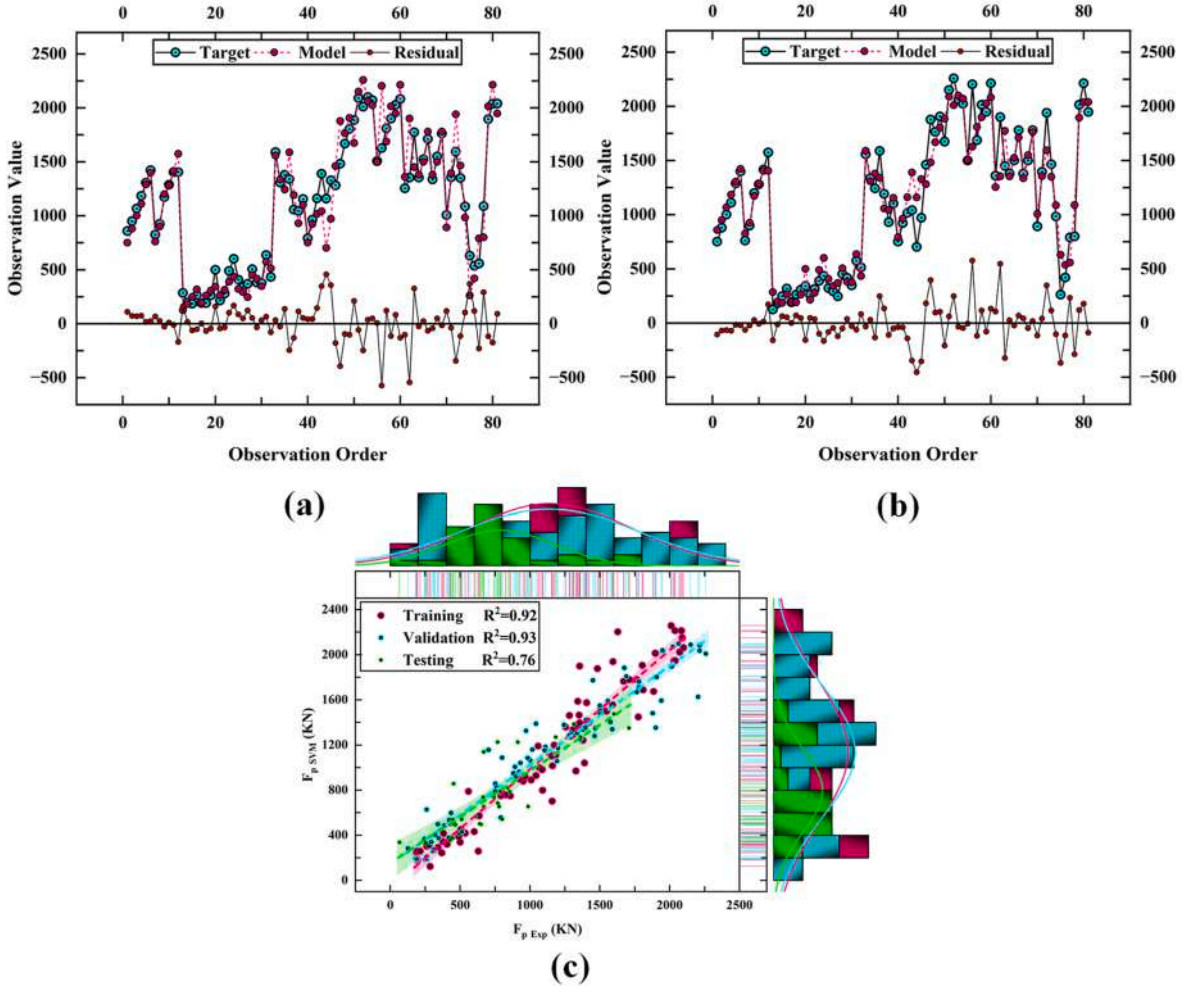

Fig. 13. SVM model (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the whole data.

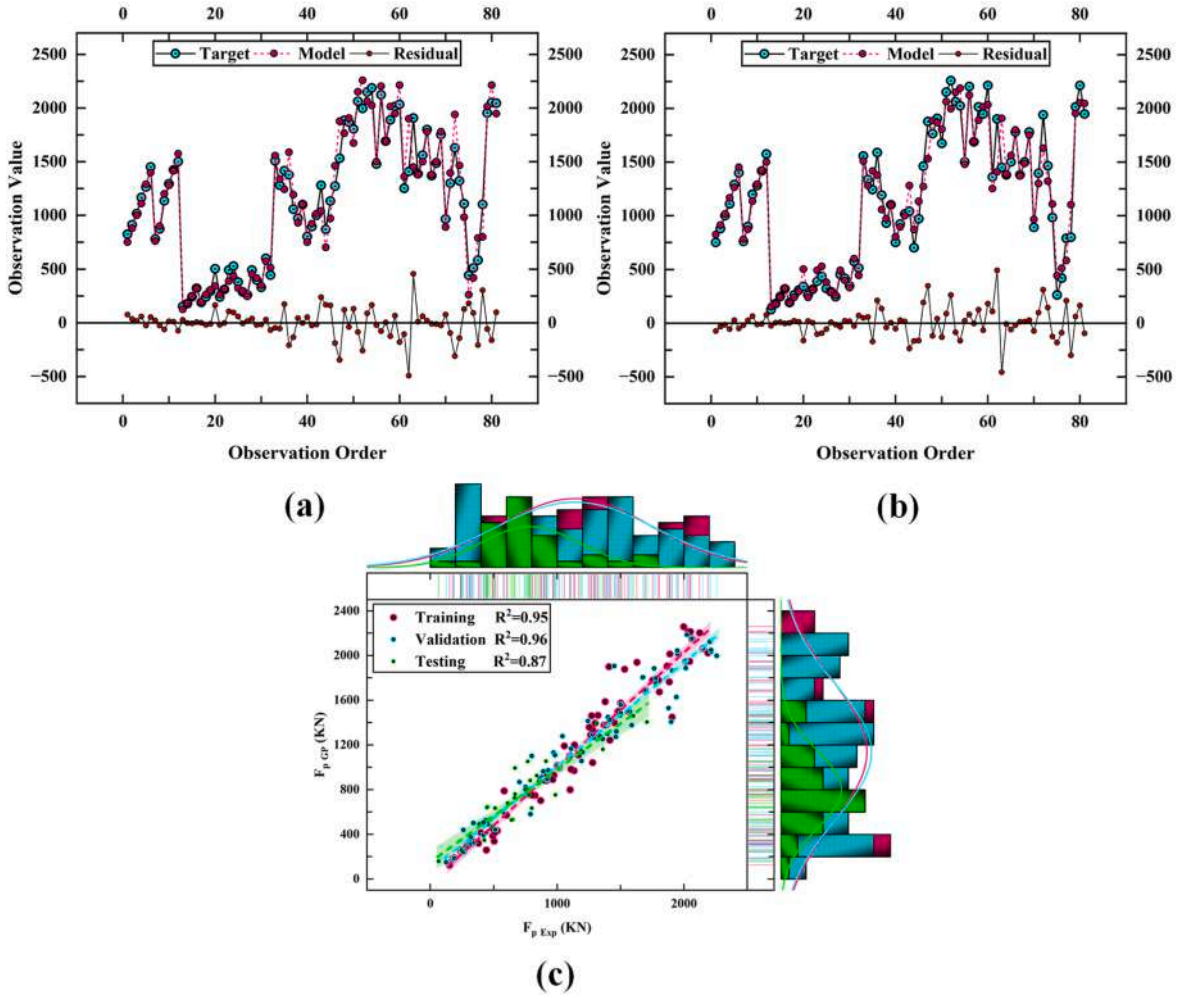

Fig. 15. GPR process (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the whole data.

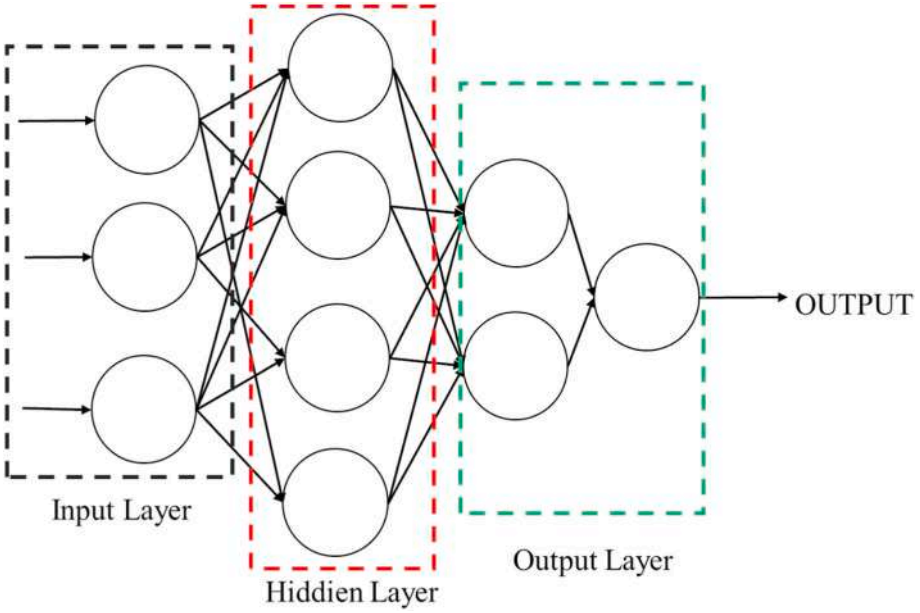

Fig. 16. Core process of ANN.

achieves this by efficiently calculating gradients, leading to lower memory requirements. It also improves efficiency by bypassing computationally expensive line searches in each iteration. Tools like MATLAB provide built-in functions (e.g., 'trainscg') for implementing SCG in ANN training. This method iteratively updates the network's weights and biases. SCG can be easily applied when the network's weights, net inputs, and transfer functions are differentiable. Moreover, SCG dynamically calculates its step size based on a quadratic approximation of the error function, reducing the need for user-defined parameters [95,99,100].

While BPNN adjusts weights in the direction of steepest descent (negative gradient), this method, though effective in minimizing the error function, doesn't guarantee the fastest convergence. In contrast, conjugate gradient methods explore directions that often lead to quicker convergence compared to steepest descent, while still maintaining the error reduction achieved in previous steps [101]. The term "conjugate direction" refers to this efficient search path.

An important advantage of line search-based algorithms, including conjugate gradient methods, is their ability to independently update user-defined parameters at each iteration [100]. This independence is crucial for the algorithm's effectiveness. Most conjugate gradient algorithms refine the step size in each iteration, moving along the conjugate direction and calculating the optimal step size to minimize the error function along that path. Besides line searches, alternative methods can also estimate the step size. Table 9 summarizes the model parameters while Fig. 17 displays the resulting curve fitting and scatter plot for the trained model.

4.2. Ensemble machine learning techniques

4.2.1. Random forest

A RF classifier consists of an ensemble of tree-structured classifiers. Each tree in this ensemble casts a vote to determine the most prevalent class for a given input x. This mechanism demonstrates how RF integrates multiple tree-structured classifiers to make a collective decision [102,103].

The development of RF was significantly influenced by prior research in geometric feature selection. Notable contributions include Amit and Geman's work in 1997, Ho's advancements in the random subspace method in 1998, and Dietterich's exploration of random split selection in 2000. Breiman's pioneering work on random forests, combined with empirical studies by other researchers, highlighted the effectiveness of RF compared to established methods like boosting and support vector machines [103,104].

RF utilize bagging (bootstrap aggregating) to diversify the trees by generating distinct training data subsets through random resampling. This process enhances stability and prediction accuracy while minimizing correlation between trees. Within RF, trees are grown without pruning, making them computationally efficient. During tree expansion, RF selects the optimal feature/split point from a randomly chosen subset of input features, reducing inter-tree correlation and decreasing generalization error [78,105]. Each tree in the RF model is built using a training sample set and a unique random variable for the k_{th} tree, ensuring independence and identical distribution across all trees. This process yields a collection of classifiers {h (x, Θ_k), k = 1, …, n}, with x representing the input vector. Through k iterations, multiple classification model systems are generated, and the final prediction is determined via a simple majority vote.

$$H(x) = arg\ \ Max_y \sum_{i=1}^{k} I(h_i(x) = Y) \quad \text{Eq. 30}$$

The combination of classification models is represented by H(x), a single DT model is indicated by h_i, the output variable is denoted by Y, and the indicator function is represented by dot. Each tree participates in the voting process to get the best categorization outcome for a particular input variable [106]. Fig. 18 shows an illustration of this procedure.

Within RF algorithm, the margin function quantify the votes for the correct class at a given point (X, Y). Eq. (31) provides the margin function formulation as:

$$mg(X, Y) = av_k I(h_k(X) = Y) - max_{j \neq Y} av_k I(h_k(X) = j) \quad \text{Eq. 31}$$

A larger margin value indicates higher confidence in the classification conclusion and greater accuracy in classification prediction [103]. For this study, 10 trees and 5 node splits are chosen (see Table 10). The scatter plot and model fitting curve of the developed model is shown in Fig. 19.

4.2.2. Boosting family

To improve the effectiveness of machine learning, three main ensemble learning methods are used: bagging, boosting, and stacking. Boosting, a technique in supervised learning introduced by Freund and Schapire, works by creating models sequentially, where each new model addresses the errors made by the previous ones. This approach involves the iterative addition of weak learners, with a focus on the data points that were previously misclassified. Like bagging, boosting can be applied to both regression and classification problems [81,107]. Various boosting algorithms have been developed, including GBA, AdaBoost, and ensemble boosted tree.

4.2.2.1. (a) boosted tree. Boosted trees (BT) utilize the least squares boosting ensemble (LSBoost). LSBoost consists of multiple weak learners along with a meta-learner that assigns weights to each learner and combines their predictions using voting methods to improve accuracy in regression tasks. The boosting process involves splitting the data into training and validation sets [107,108]. Initially, the algorithm trains individual weak learners one by one, usually using DTs, and then fits the residual errors to enhance performance. LSBoost uses the least squares criterion for assessing loss [81]. Fig. 20 shows the subsets formed as a result of dataset splitting.

BT are implemented using the Matlab regression learner tool. This tool employs the 'fitrensemble' function to train ensemble models, allowing users to tweak key parameters for model optimization. One important parameter is the minimum leaf size, which sets the minimum number of training samples required at each leaf node of the regression tree. Adjusting this parameter helps balance model complexity and predictive accuracy, with smaller leaf sizes often improving training accuracy but potentially reducing generalization. Another critical parameter for boosted trees is the learning rate, which influences the shrinkage rate during ensemble learning. Lower learning rates typically necessitate more iterations but can lead to better accuracy. Additionally, users can choose the number of predictors to sample during tree construction, providing flexibility in feature selection [81]. These parameters collectively enable users to fine-tune model performance, ensuring an optimal trade-off between accuracy and generalization. Table 11 summarizes the parameters employed for model development. Fig. 21 displays the model fitting curve and scatter plot obtained from the developed model.

4.2.2.2. (b) adaptive boosting. AdaBoost maintains a probability

Table 9
Parameters employed for model development.

Parameter	Typical Values
Initial Damping Factor (λ)	1.0
Sigma (σ)	10^{-6}
Tolerance	10^{-5}
Max Iterations	1000
Training and validation	90 % and 10 %
Epsilon (ϵ)	10^{-10}
Gradient Norm	10^{-5}
Learning Rate	Typically determined internally

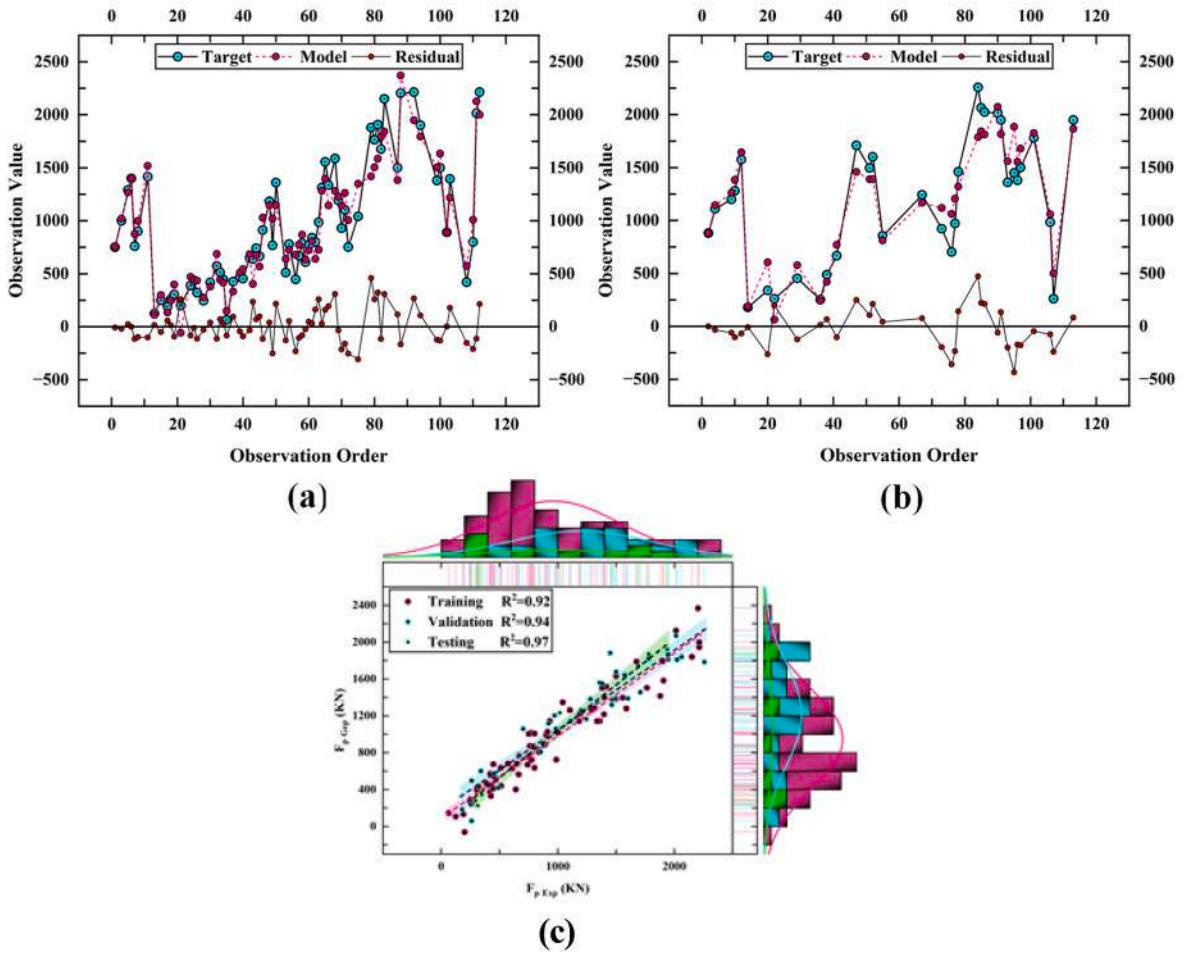

Fig. 17. SCG algorithm (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the data (d) Scatter plot for the validation data.

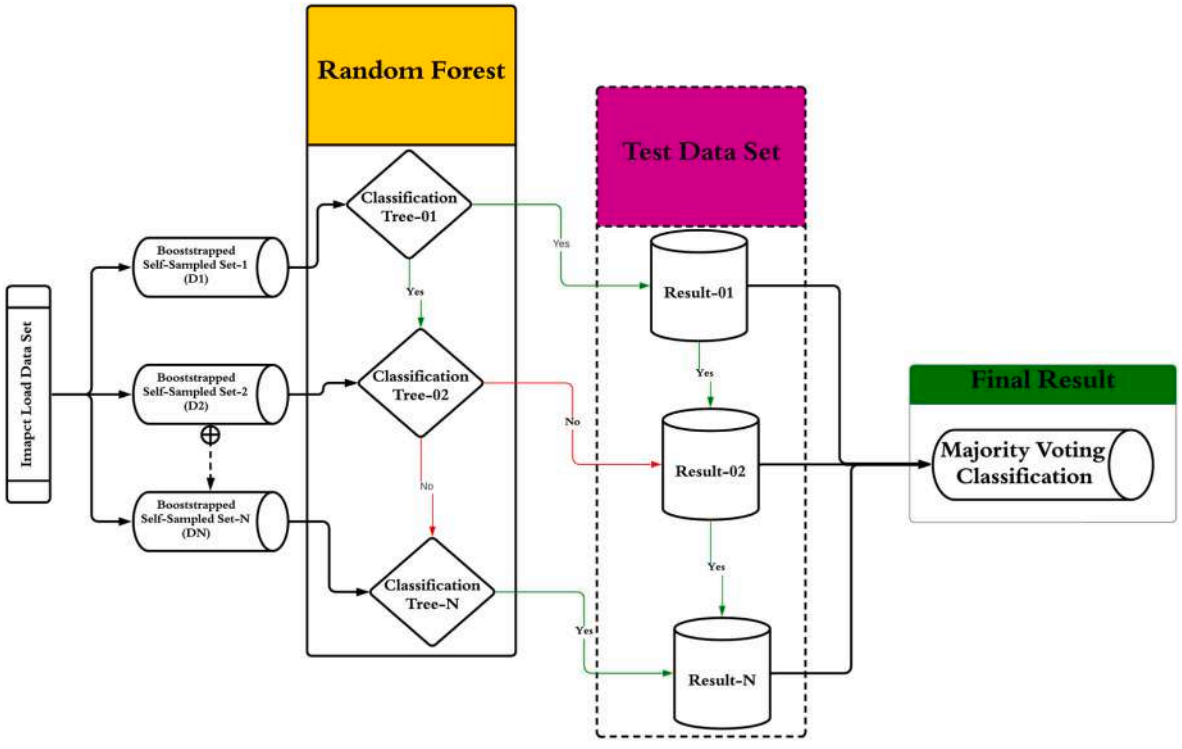

Fig. 18. Flowchart of RF

Table 10
Parameters associated with RF.

Developed trees	Train and validation	Total features	Replicability	Do not split when
10	90 % and 10 %	Unlimited	No	Instances = 5

distribution over the training samples, which it adjusts iteratively. A specific learning algorithm generates a member classifier and calculates its error rate on the training samples. AdaBoost then uses this error rate to update the probability distribution [109]. The flowchart of AdaBoost is shown in Fig. 22.

Considering the regression complexity, the training data Θ can be shown as:

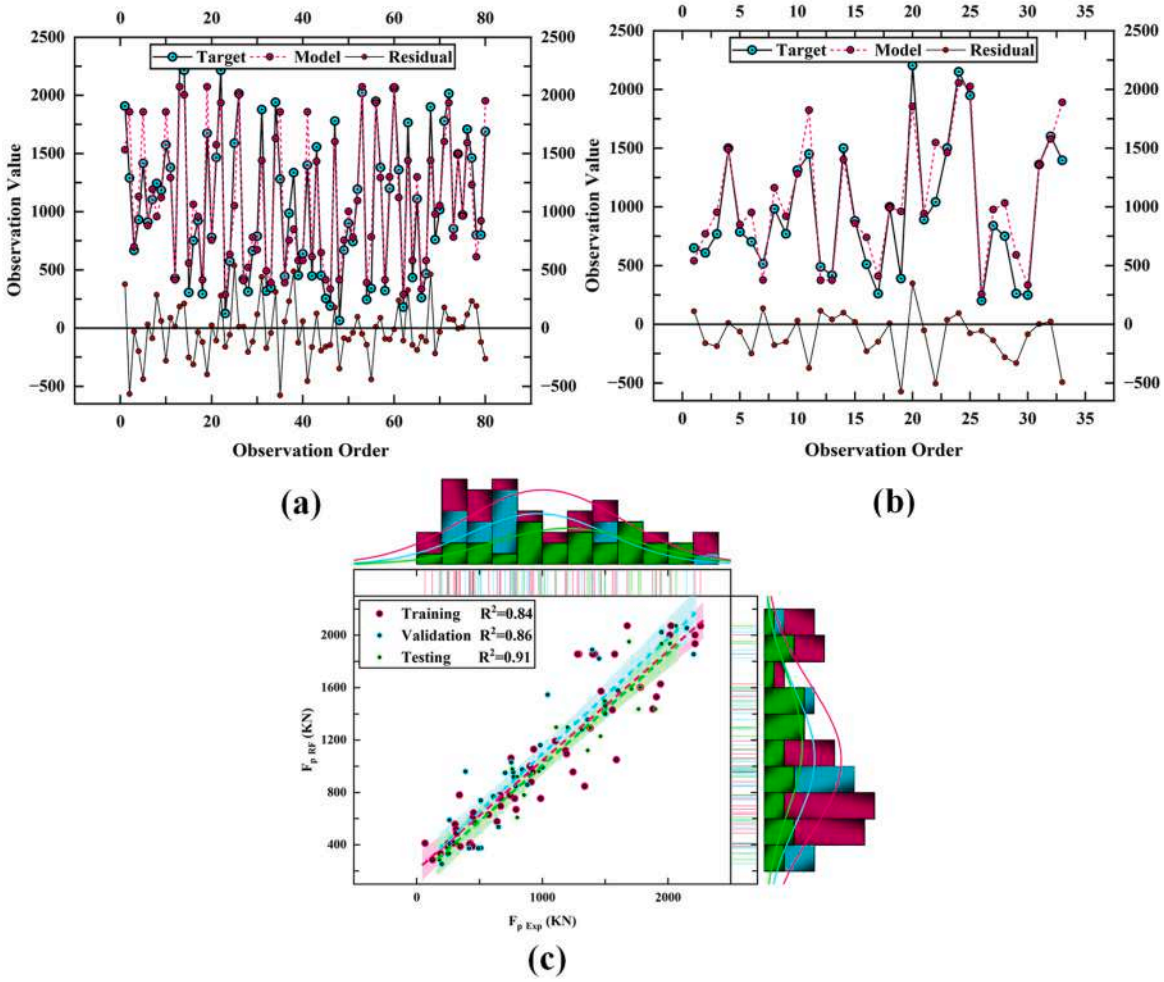

Fig. 19. RF model (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the whole data.

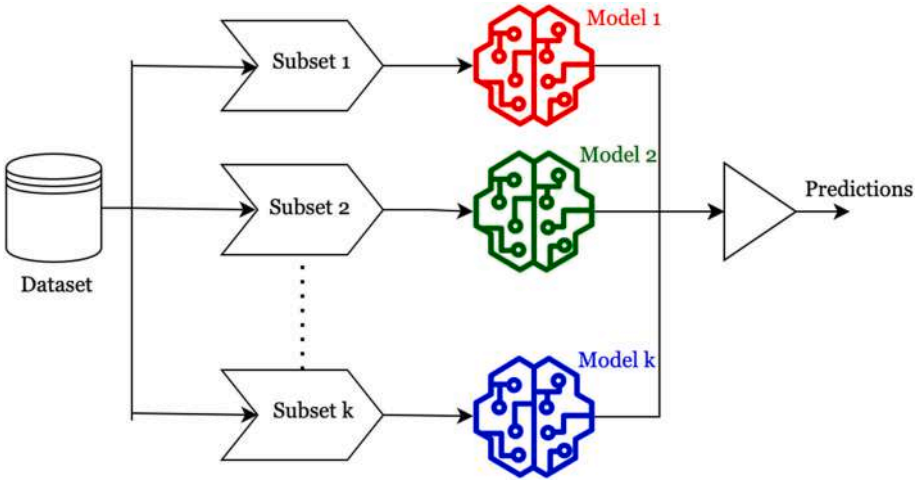

Fig. 20. Schematic presentation of data division.

Table 11
Parameters employed in model development.

Parameter	Default value
Learning Rate (η)	0.1
Training and validation	55 % and 45 %
Number of Estimators	100
Max Depth	3
Min Samples Split	2
Min Samples Leaf	1
Subsample	1.0 (or 100 %)

$$\Theta = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\} \quad \text{Eq. 32}$$

In Eq. (32), each data point is represented by (Xi, Yi), where i ranges from 1 to n, and 'n' indicates the total number of samples. Here, Yi is the output value, and Xi is the input vector. This data is then applied to a base or weak learner, represented as G(X), using specific learning methods. The relative estimation error (e_i) for each sample is then calculated as shown in Eq. (33) [86,110]:

$$e_i = L(Y_i, G(X_i)) \quad \text{Eq. 33}$$

For simplicity, among the linear, square, and exponential loss, linear loss function, as given in Eq. (34), is used:

$$e_i = \frac{Y_i - G(X_i)}{E} \quad \text{Eq. 34}$$

Eq. (34) uses $E = \max|Y_i - G(X_i)|$, which represents the maximum estimation error observed across all data points. Relying solely on the performance of an individual learner can result in suboptimal outcomes. Therefore, the objective of AdaBoost is to sequentially generate ($G_k(X)$) and combine them to create a strong learner H(X) using a predefined aggregation technique. The amalgamated technique involves [111]:

$$H(X) = \nu \sum_{k=1}^{N} \left(ln \frac{1}{\propto_k} \right) g(X) \quad \text{Eq. 35}$$

In Eq. (35), (X) is the median of $\alpha_k G_k$, where α_k represents the weight assigned to $G_k(X)$. Additionally, $\nu \in [0,1]$ is used to address overfitting issues. By employing a re-weighting method, the data representation is updated to generate $G(X)$. This method involves adjusting the importance of each sample based on the errors of the previous learner $G_{k-1}(X)$. Consequently, samples with higher estimation errors are given greater weight, increasing their influence in subsequent training phases. This iterative process uses the estimation error e_{ki}, as outlined in Eq. (36) [112].

$$e_k = \sum_{i=1}^{m} e_{ki} \quad \text{Eq. 36}$$

Poor learner weight is given in Eq. (37).

$$\propto_k = \frac{e_k}{1 - e_k} \quad \text{Eq. 37}$$

and the weight $w_{k+1,i}$ of individual point for coming phase is tuned as given in Eq. (38).

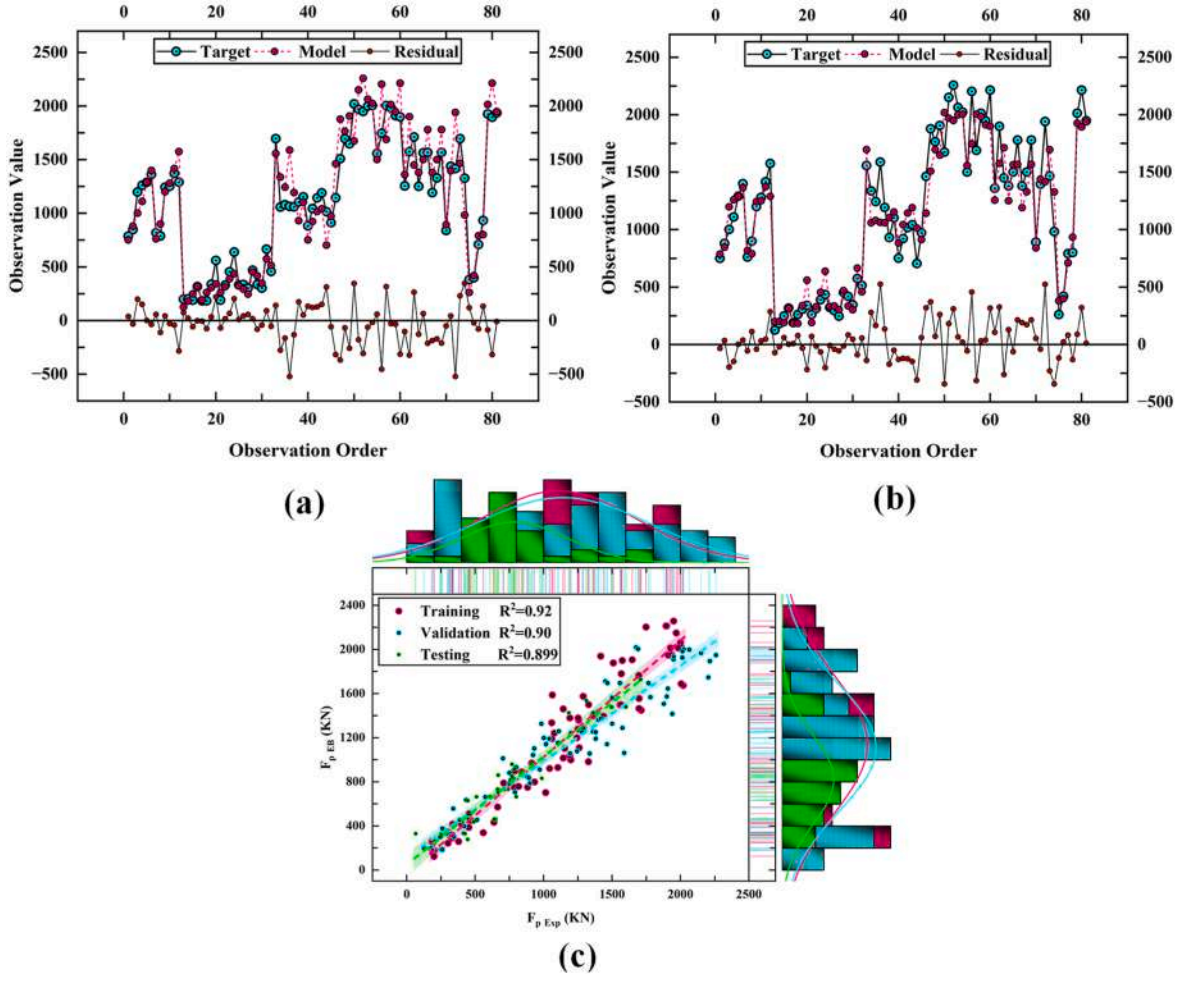

Fig. 21. Ensemble boosted tree (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the whole data.

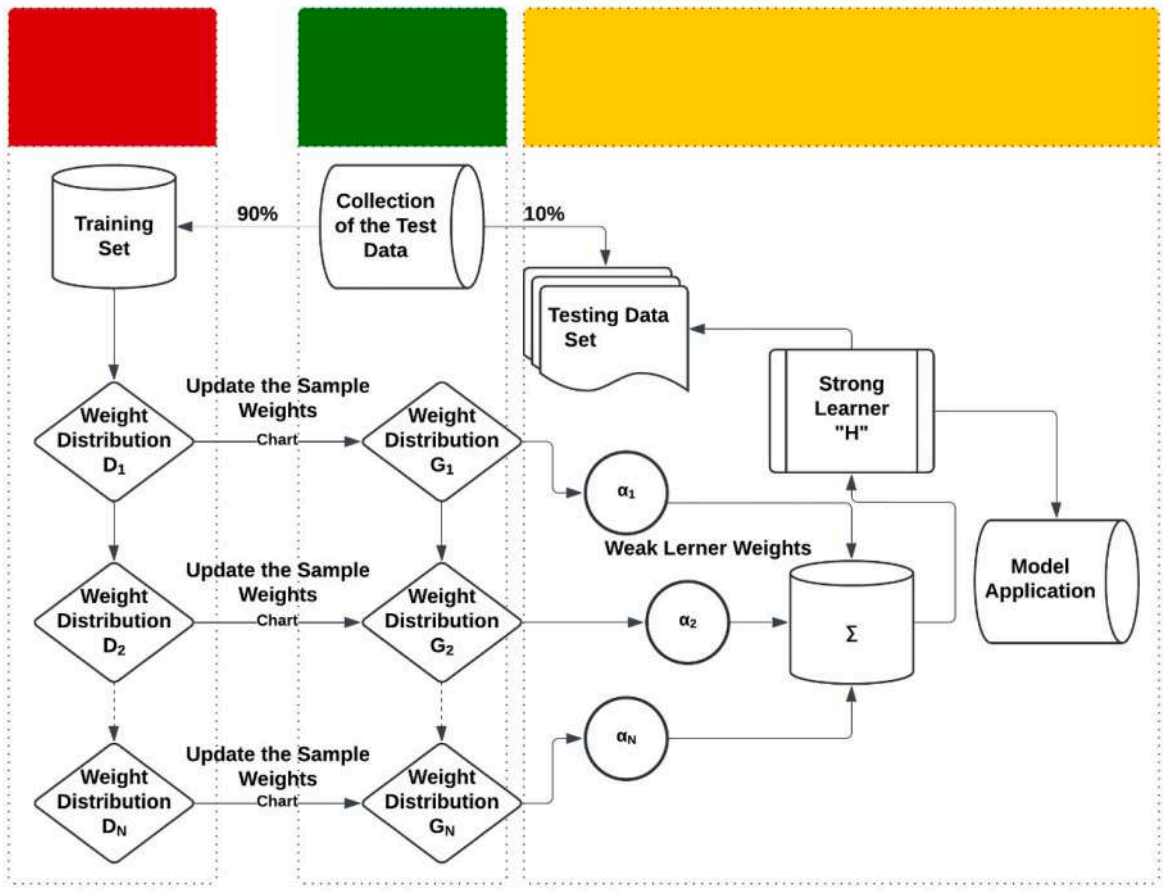

Fig. 22. Flowchart of adaptive boosting algorithm.

$$w_{k+1,i} = \frac{w_{k,i}\alpha_k^{1-e_{ki}}}{\sum_{i=1}^{m} w_{k,i}\alpha_k^{1-e_{ki}}} \quad \text{Eq. 38}$$

The processed samples are represented by w_{ki}, where samples with higher prediction errors are given increased importance to improve learning in subsequent iterations. Nearly any regression algorithm can serve as a weak learner in AdaBoost [110]. In this instance, we choose DT due to their proven effectiveness across various fields. Table 12 summarizes the model paramters while Fig. 23 shows the curve fitting and scatter plot generated by the model.

4.2.2.3. (c) gradient boosting algorithm. Gradient boosting is a versatile learning technique suitable for both regression and classification tasks. It creates an ensemble of DTs and uses a stage-wise approach, similar to other boosting methods, which allows it to optimize various differentiable loss functions [113,114]. The concept of gradient boosting was first identified by Leo Breiman, who viewed it as an optimization algorithm for an appropriate cost function [115]. Further advancements were made by others, who contributed to the concept of functional gradient boosting. This approach frames boosting algorithms as iterative processes of functional gradient descent, optimizing a cost function in function space by iteratively choosing functions (weak hypotheses) that move in the direction opposite to the gradient. This perspective has advanced boosting algorithms across various machine learning and statistical domains, extending beyond traditional regression and classification tasks [116]. The basic process of gradient boosting is shown in Fig. 24.

Table 12
Parameters involved in the AdaBoost process.

Base learner	Total predictors	Training and validation	Classification Algorithm	Regression loss
Tree	Fifty	80 % and 20 %	Samme.r	Exponential

In a dataset for the regularized learning objective D_D comprising n_n examples and m_m features, $D = \{(B_i, A_i)\}$, $(D = n \times R^m \times R)$, an additive functions is employed to predict the outcome.

$$\widehat{A_i} = \varnothing(B_i) = \sum_{a=1}^{a} f_a(B_i), f_a \in F \quad \text{Eq. 39}$$

In Eq. (39), $F = \{f(\mathrm{B}) = z_{p(B)}\}$ represents the regression trees space. The number of leaves in the tree is denoted by the variable K, and the structure of each tree that translates an example to the associated leaf index is represented by the variable "p". Every f_a is associated with a separate tree structure (p) and leaf weights (z). Every regression tree, in contrast to conventional DTs, has a continuous score on every leaf. The score on the i_{th} leaf is represented by z_i in this study. The decision rules in the study's trees (provided by p) are categorized into their leaves, and the final prediction is calculated by adding together the scores in the relevant leaves (supplied by w) [117,118]. In order to discover the functions employed in the model, minimize the subsequent regularized goal:

$$M(\varnothing) = \sum_i m(\widehat{A_i}, A_i) + \sum_i \Omega(fa) \quad \text{Eq. 40}$$

The difference between the model A_i and the goal $\widehat{A_i}$ is quantified by "m," which is a differentiable function of the convex loss in this case. The model's complexity is penalized by the second term. To avoid overfitting, the extra regularization term helps to smooth the final learning weights. The regularized objective will naturally choose a model that

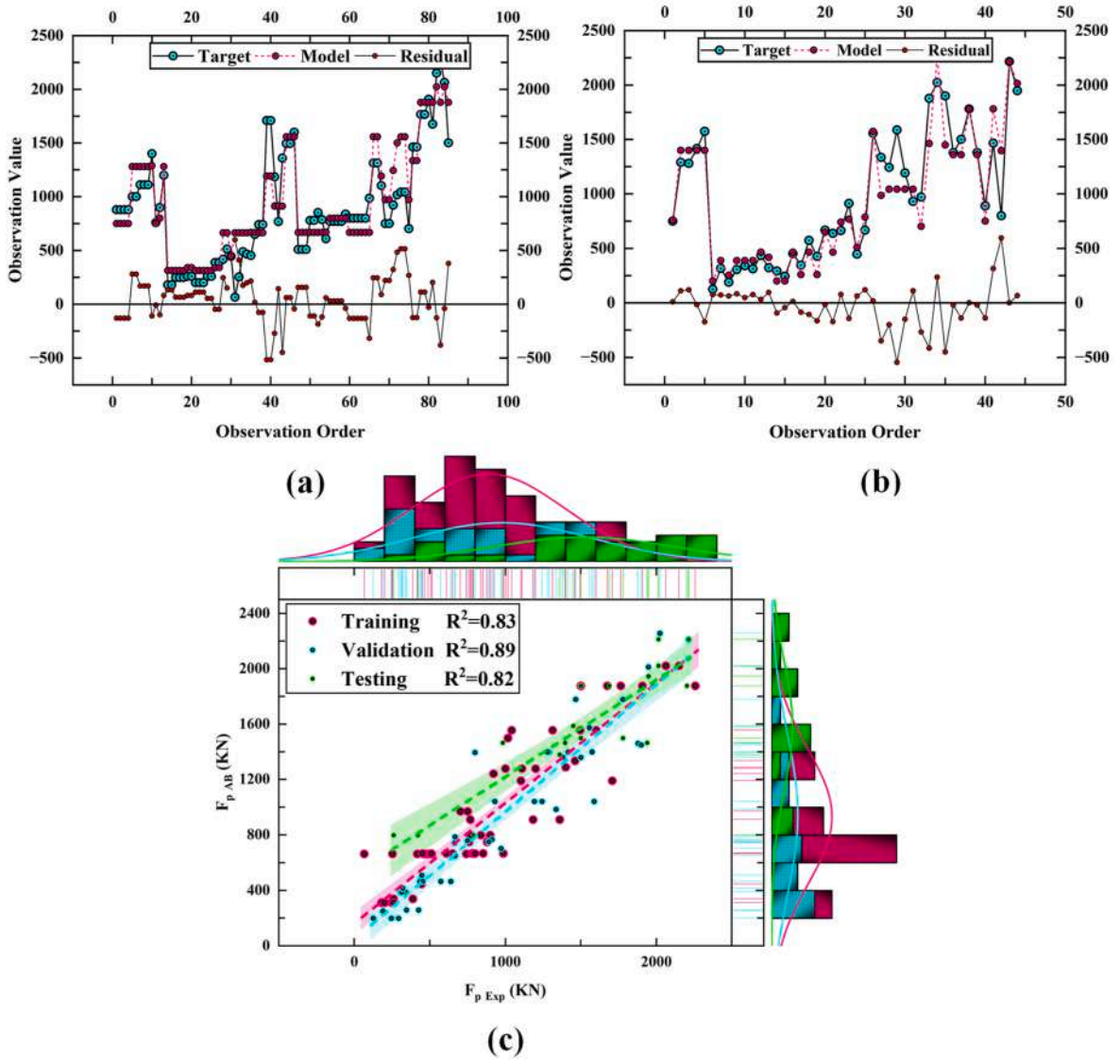

Fig. 23. Adaptive boosting algorithm (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the whole data.

uses straightforward and predictable functions [119].

The ensemble tree model described in Eq. (41) incorporates functions as parameters, making it unsuitable for optimization using conventional methods in Euclidean space. Instead, the model is trained additively [114]. Formally, $\widehat{A_i}$ represents the estimation of the i_{th} instance at the t_{th} iteration, and *ft* must be added to lower the following objective:

$$M^{(t)} = \sum_{i=1}^{n} m\left(A_i, \widehat{A}_i^{(t-1)} + f_{(t)}\right) + \Omega(f_t) \qquad \text{Eq. 41}$$

This implies that *ft*, which significantly enhances the model as per Eq. (42), is added in a slow manner. In a broader context, a second-order approximation is utilized for efficient optimization [5].

$$M^{(t)} = \sum_{i=1}^{n} \left[m\left(A_i, \widehat{A}_i^{(t-1)}\right) + J_i f_t(B_i) + \frac{1}{2} h_i f_t^2(B_i)\right] + \Omega(f_t) \qquad \text{Eq. 42}$$

In Eq. (42), $J_i = \{\partial_{\widehat{A}(t-1)}\} = j$, and $H_i = \left(\partial^2_{\widehat{A}(t-1)}\right)$ represent the first and second-order gradient statistics on the loss function, respectively. Table 13 summarizes the parameters employed for model development. Fig. 25 displays the model fitting curve and scatter plot obtained from the developed model.

5. Model performance evaluation

5.1. Predictive performance

Fig. 26 illustrates the predictive performance of various machine learning models, including GB, RF, GEP, AdaBoost, DT, and others. The y-axis represents the ratio of experimental to model prediction, while the x-axis displays the mix constituents. A wider spread along the x-axis indicates greater variation in the input parameters.

Overall, all models showed good performance, with majority of points lying near the of 1, demonstrating strong agreement between estimated and experimental values. Tight clustering around the reference line indicates good accuracy. However, some outliers exceeding 1.5 or falling below 0.5, suggest areas for improvement through model adjustments. In summary, while machine learning models generally demonstrated satisfactory predictive performance, further refinement is required to address outlier instances of overestimation and underestimation.

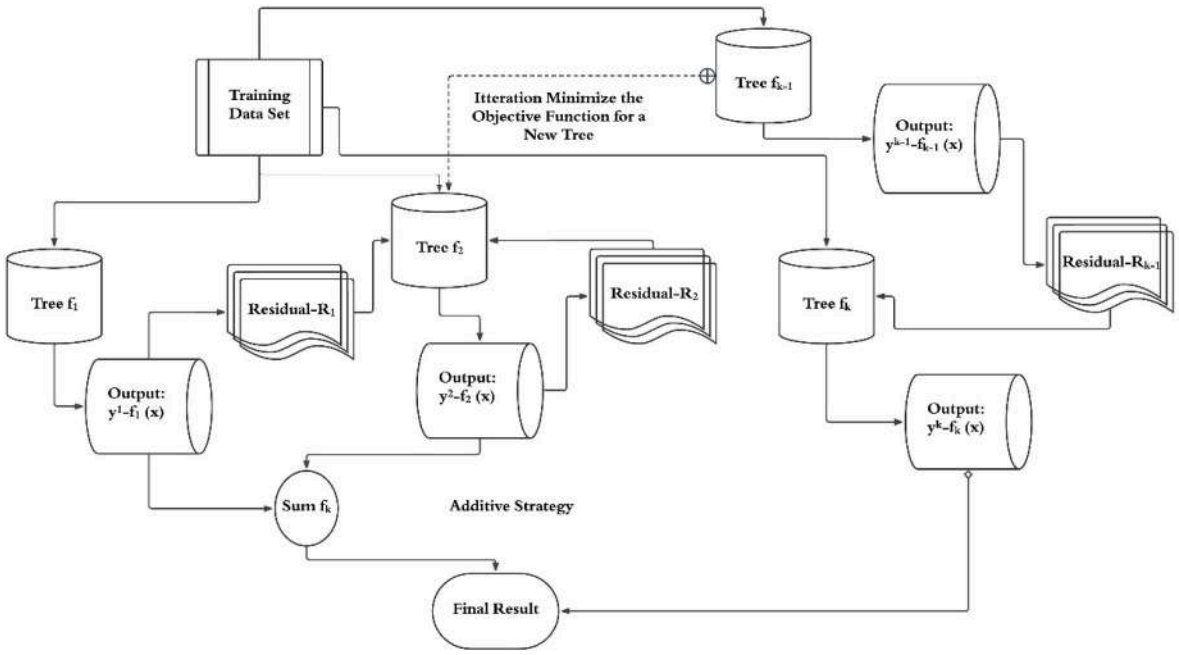

Fig. 24. Gradient boosting basic process.

Table 13
Modeling parameters involved in gradient boosting.

Developed Trees	Regularization	Training and validation	Replicable	Instances used in training	Total instances
100	Lambda: 1	90 % and 10 %	Yes	1	2

5.2. Statistical performance

To evaluate the accuracy of the models, various performance metrics were employed. Initially, the correlation coefficient (R) was examined to assess how well the predicted values matched the actual values. However, relying solely on the correlation coefficient is insufficient for determining model accuracy. Therefore, mean absolute error (MAE), a-10 index, a-20 index and root mean square error (RMSE) were also considered as evaluation metrics. These metrics are detailed in Eq. (43) through Eq. (47). A R value above 0.8 signifies a strong fit between the model and actual data. Lower MAE and RMSE values indicate higher predictive accuracy. Engineering indices like the a10-index and a20-index measure prediction errors within ±10 % and ±20 % ranges, respectively. Values of these indices closer to one denote better model performance.

$$R = \frac{\sum_{i=1}^{n}(T_i - \widehat{T}_i)(P_i - \widehat{P}_i)}{\sqrt{\sum_{i=1}^{n}(T_i - \widehat{T}_i)^2 \sum_{i=1}^{n}(P_i - \widehat{P}_i)^2}} \quad \text{Eq. 43}$$

$$RMSE = \sqrt{\sum_{i=1}^{n}(T_i - \widehat{P}_i)^2 \Big/ n} \quad \text{Eq. 44}$$

$$\text{MAE} = \sum_{i=1}^{n}(T_i - \widehat{P}_i)^2 \Big/ n \quad \text{Eq. 45}$$

$$\text{A-10} = \text{P}_{10}/\text{n} \quad \text{Eq. 46}$$

$$\text{A-20} = \text{P}_{20}/\text{n} \quad \text{Eq. 47}$$

In the above equations, the actual and predicted values are denoted by T_i and P_i, respectively. The number of data points is indicated by n. The “ symbol is used to represent mean value.

As shown in Fig. 27(a) and (b), spider plots were utilized to evaluate the performance of various machine learning models based on key metrics, including the a-10 index, R^2, RMSE, a-20 index, and MAE. With the exception of the KNN algorithm, all models exhibited robust generalization capabilities, consistently achieving R^2 values above 0.8 and correlation coefficients R exceeding 0.9 across both the training and validation dataset. The statistical errors, specifically RMSE (ranging from 80 to 250), a-10 (ranging from 30 to 70), MAE (ranging from 60 to 180), and a-20 (ranging from 20 to 60), remained within acceptable thresholds, underscoring the models’ reliable predictive performance. GPR emerged as the most effective, exhibiting the highest R^2 value of 0.95 and the lowest RMSE and MAE values (95 and 60, respectively). It also recorded the highest a-10 and a-20 values. Conversely, the KNN model performed the poorest, with the lowest R^2 value of 0.75 and relatively high MAE and RMSE values.

In the depicted Taylor diagram (see Fig. 28), various models are evaluated for their effectiveness, with each model represented as a point on the diagram. The angle from the x-axis signifies the correlation coefficient, while the distance from the origin represents the standard deviation of the model. Overall, the models showed satisfactory performance across the entire dataset which is indicated by their close proximity to the referenced model. Specifically, all models exhibit R ranging from 0.85 to 0.95, suggesting strong generalizability.

6. Shapely additive explanation

The increasing complexity of machine learning models, while enhancing their predictive accuracy, often reduces their interpretability, rendering them "black boxes" with outputs that are difficult to under-

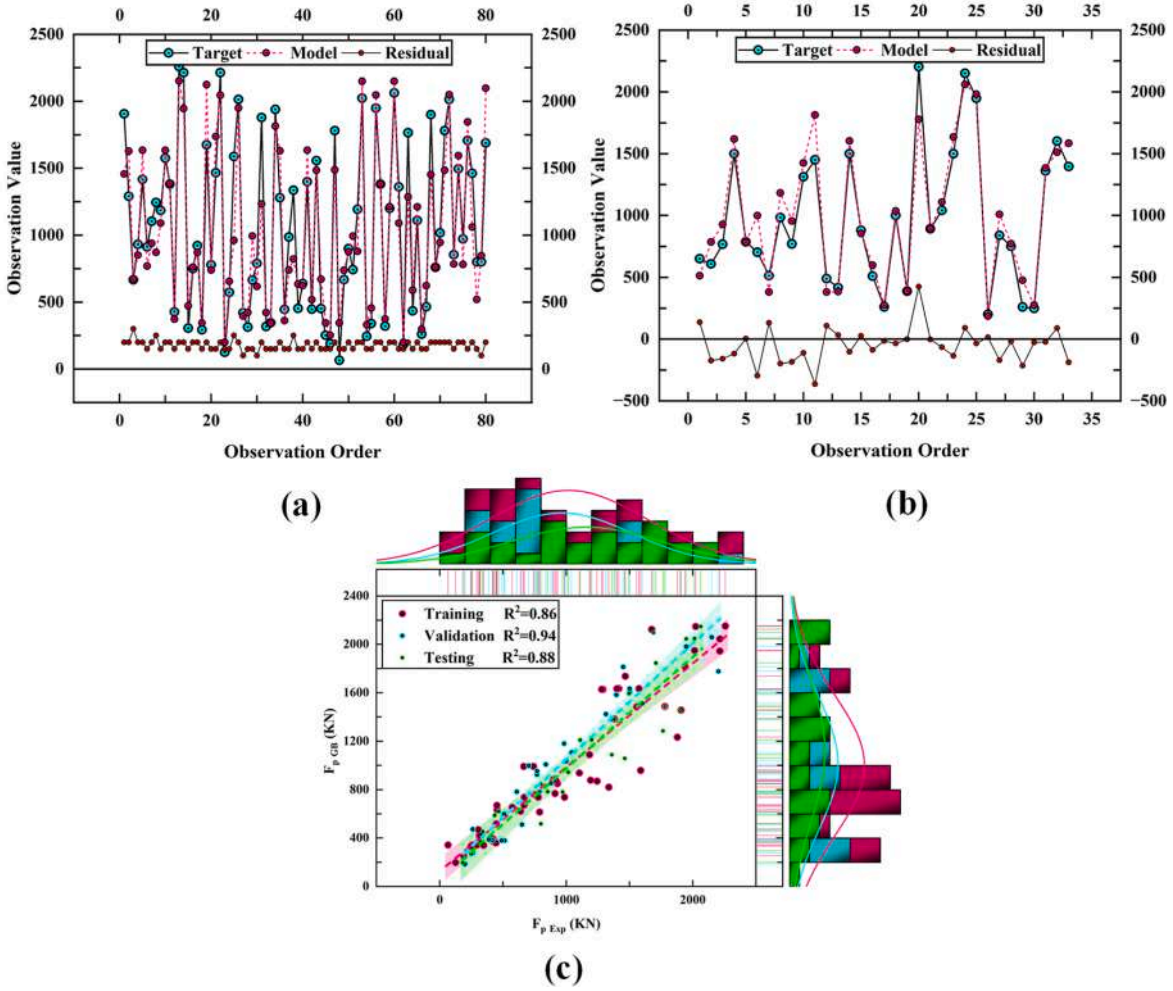

Fig. 25. Gradient boosting model (a) curve fitting for training data (b) curve fitting for validation data (c) Scatter plot for the whole data.

stand. Consequently, improving the interpretability and explainability of models is essential. One effective approach is the application of shapley additive explanations (SHAP), which uses XGboost models to make the intricate outputs as discernible pattern. SHAP values quantify the contribution of individual features to model predictions, facilitating a deeper understanding of significant features and their impact on the final prediction. Higher shapley values indicate greater relevance of the corresponding feature [120,121]. The influence of 'j' on f(x_i) for a given point is denoted as φ_j(f), and is calculated using Eq. (48), which sums the marginal contributions of each feature across all possible feature combinations.

$$\varphi_j(f) = \frac{\sum S \subseteq [x^1, \ldots\ldots, x^p]}{\{x^j\}} \left[\frac{|S|!(p-|S|-1)!}{p!}\right] \left[f(S \sqcup \{x^j\}) - f(S)\right] \quad \text{Eq. 48}$$

Eq. (48) illustrates that S is a subset of the features and p is the total number of features in the model. By analyzing how projected errors vary with changes in features, SHAP assigns relevance weights to features. Additionally, SHAP uses a linear feature attribution technique to clarify the performance of trained machine learning models and generate an interpretable model based on its findings [77,122].

$$f(x) = h(x_s) = \varnothing_0 + \sum_{i=1}^{p} \varnothing_i {x_s}^i \quad \text{Eq. 49}$$

Eq. (49)—with p denoting the number of input features and ϕ_0 representing the constant term—maps the input variables x and x_s using the notation $x = m(x_s)$. According to Lundberg and Lee [123], Eq. (49) enhances the predicted value $h(x_s)$ by including terms ϕ_0, ϕ_1, and ϕ_3, while excluding ϕ_4 in $h(x_s)$ (as illustrated in Fig. 29).

A unified solution for Eq. (49) offers three main benefits: handling missing data, ensuring local accuracy, and maintaining reliability. Reliability is crucial to prevent attributes from inadvertently shifting to another feature. When features are missing, they are deemed insignificant; hence, $x_s^i = 0$, $\phi_i = 0$ is assigned. The summation of features within an output function creates a model that aligns the output f with x_s as the simplified input, achieving local accuracy [124].

6.1. Shapley plots

Fig. 30 illustrates the significance of the various parameters influencing the response force. The parameters with the most substantial contributions are beam depth, impact velocity and beam breadth with mean SHAP values of +298.79, +182.94, and +92.02, respectively. Additionally, the stirrup ratio and the yield strength of the steel demonstrated higher mean SHAP values. All of the remaining parameters were found to have marginal impact on the response force.

The violin SHAP plot in Fig. 31 highlights the significant influence of beam depth, impact velocity, and beam breadth on the peak response force, corroborating findings from other empirical studies. In particular, the broad distribution of SHAP values for beam depth, reaching up to +400, suggests a strong positive impact on the response force. This result aligns with the conclusions of Zhan et al. [55] and May et al. [56], who similarly noted a positive correlation between increased beam

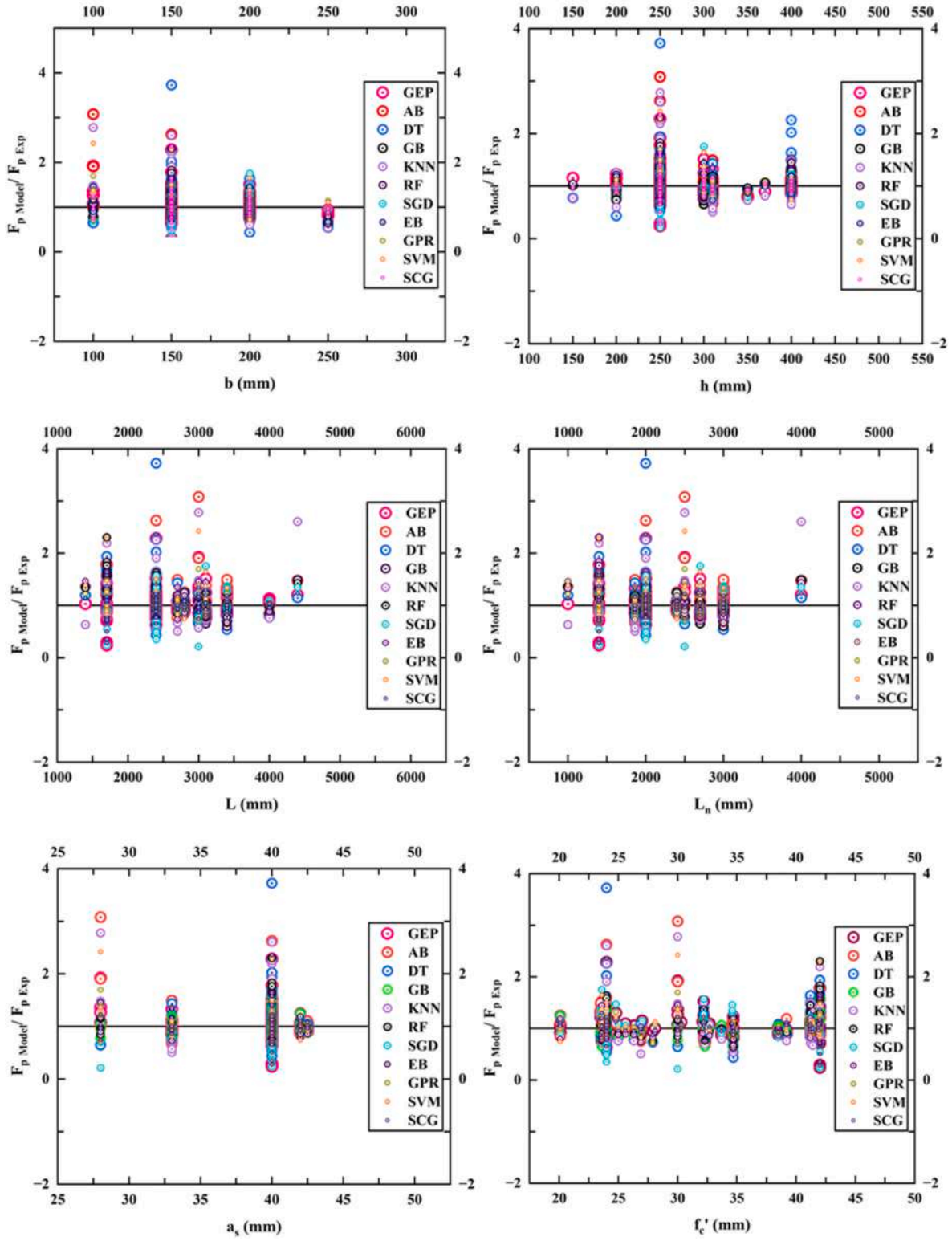

Fig. 26. Predictive performance of the developed models.

depth and peak impact force. These authors attribute the increased force to the beam's enhanced stiffness, as larger depths allow the beam to absorb more energy upon impact. The SHAP analysis in the present study confirms that beam depth plays a critical role in governing the beam's load-bearing capacity during impact.

Similarly, the substantial SHAP values for impact velocity (clustered around +200) emphasize its critical role, consistent with the findings from various other studies. For instance, the kinetic energy relationship $KE = \frac{1}{2}mv^2$ explored in Adhikary et al. [125] and Bhatti et al. [61] indicates that higher impact velocities significantly raise the kinetic energy imparted to the RC beam, thereby increasing the peak impact force. These studies further support the SHAP analysis, which highlights

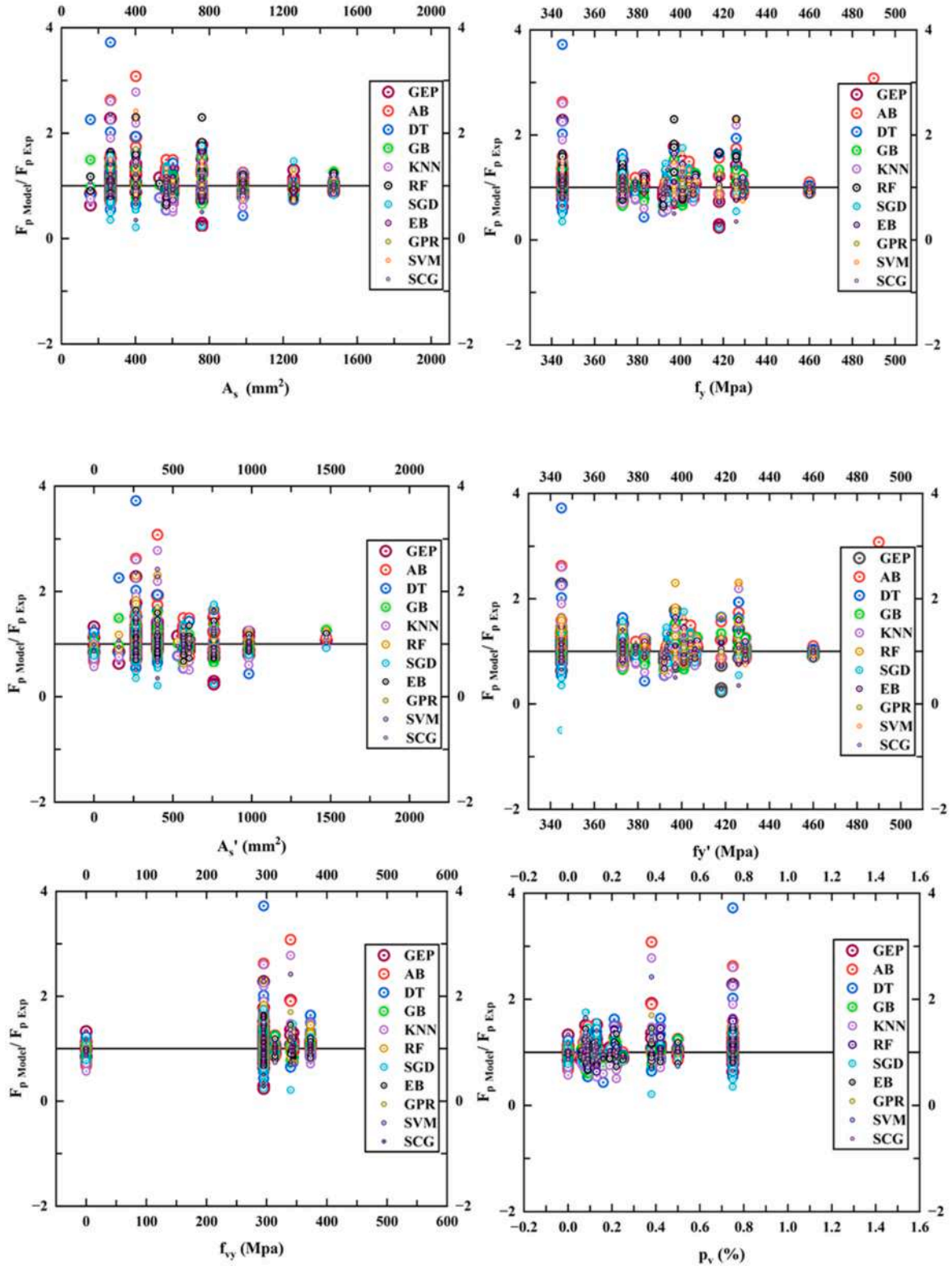

Fig. 26. (*continued*).

impact velocity as one of the dominant parameters influencing response force.

Beam breadth also emerges as a key factor in the SHAP analysis, with positive contributions clustering around +100. This finding corresponds to earlier research by Hughes and Mahmoud [57] and Goldston et al. [58], where wider beams were shown to draw higher impact forces due to their ability to distribute load over a larger surface area, thus increasing the force required for failure. In line with these experimental results, the SHAP values in the current study confirm that beam breadth significantly affects the peak response force.

In addition to these primary parameters, secondary factors such as stirrup ratio, yield strength of steel, and concrete compressive strength

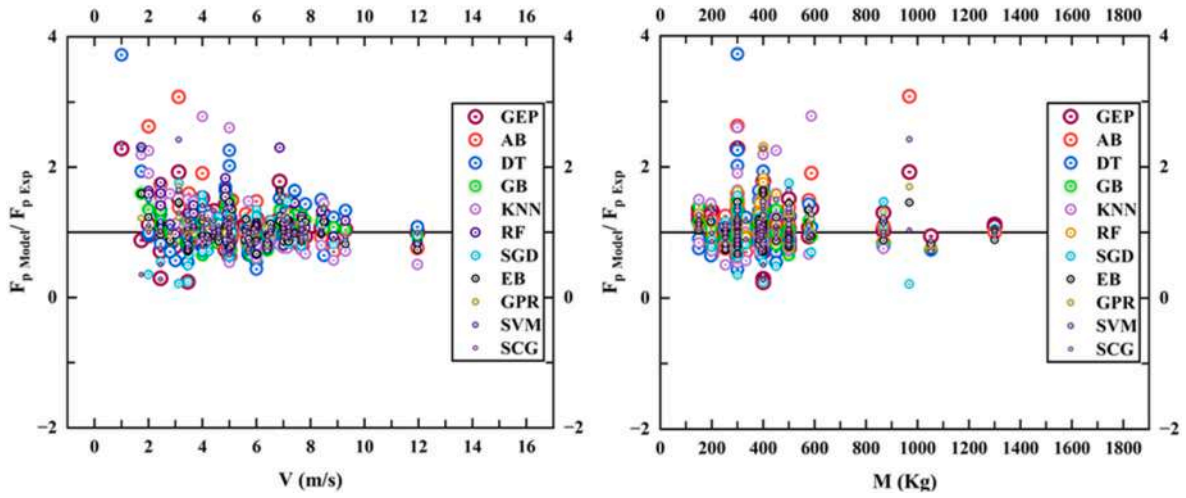

Fig. 26. (*continued*).

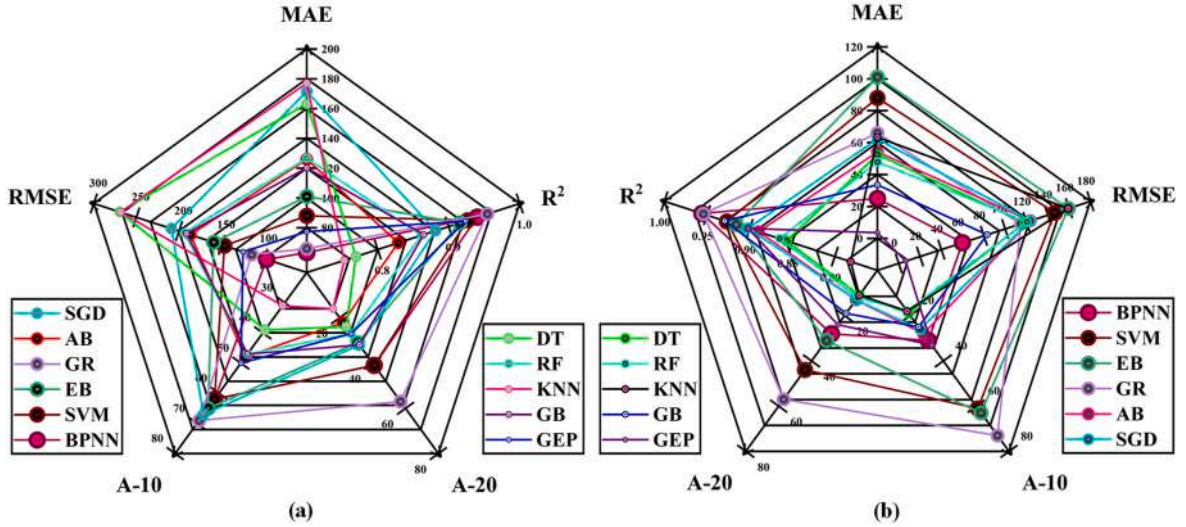

Fig. 27. Spider plot for the developed model (a) Training dataset (b) Validation dataset.

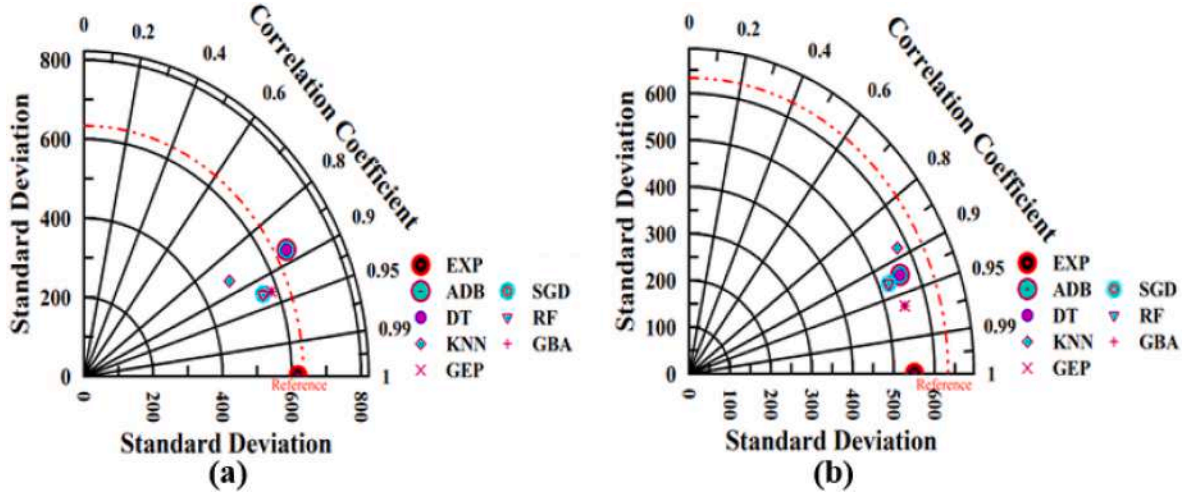

Fig. 28. Taylor diagram (a) Training data (b) Validation data.

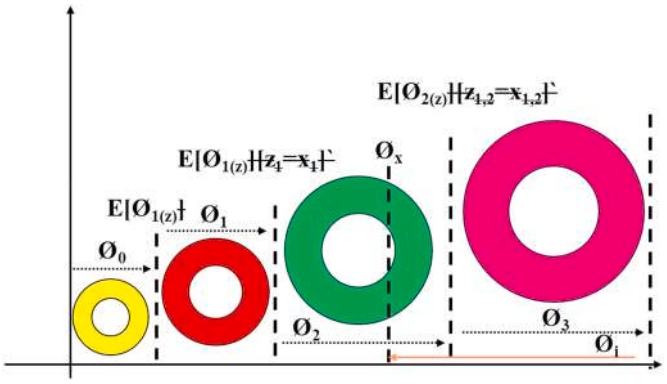

Fig. 29. SHAP feature inclusion and exclusion.

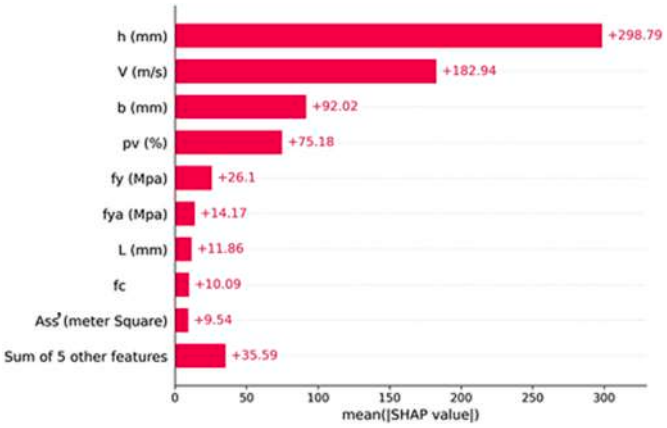

Fig. 30. SHAP feature importance plot.

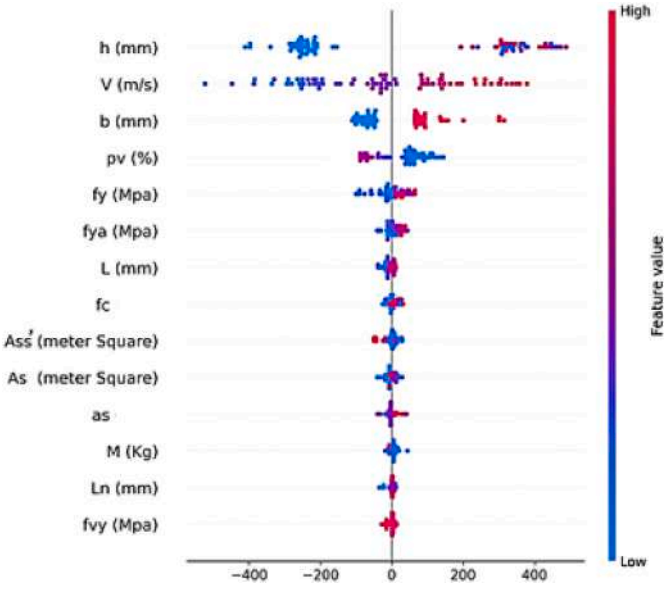

Fig. 31. SHAP violon shaped plot.

also demonstrate noteworthy SHAP values in the present study, though their contributions are less pronounced. However, previous studies, including those by Zhan et al. [55] and Adhikary et al. [125], found a positive correlation between concrete compressive strength and peak impact force. They argued that stronger concrete enhances beam stiffness, leading to a higher force absorption capacity during impact—an observation that resonates with the moderate SHAP values for compressive strength in our model.

Additionally, the stirrup ratio and yield strength of steel, while contributing less significantly in this study, have been identified as important factors in another research. For instance, Goldston et al. [58] and Bhatti et al. [61] emphasized the role of vertical shear reinforcement (stirrup ratio) in confining the core concrete and preventing buckling of longitudinal reinforcements, which increases the peak impact force. The yield strength of steel was similarly found to enhance beam stiffness and ultimate load capacity in studies by Adhikary et al. [125] and Fujikake et al. [62], further supporting its influence on peak response as observed in our SHAP analysis.

In summary, the SHAP analysis in our study corroborates many of the findings from previous experimental and numerical studies, reinforcing the importance of parameters like beam depth, impact velocity, and beam breadth, while also highlighting the secondary yet relevant roles of other material and geometric properties. These comparisons emphasize the robustness of the current model in capturing the key factors influencing peak response in RC beams under impact loading, thereby reinforcing the scientific value of the research.

Fig. 32 provides a SHAP waterfall plot derived from XGBoost model. This plot dissects the prediction, demonstrating how each feature influences the model's output, which in this instance is a peak response value of 1527.37. The base value, denoted as E[f(x)], is 1019.23, representing the average prediction of the model. Each subsequent bar indicates the additive effect of a feature on this base value, cumulatively leading to the final prediction. The most significant positive contributions arise from beam depth and impact velocity. Beam depth (300 mm) contributes +233.72 to the peak response, the highest among all features. Impact velocity (7 m/s) follows closely, adding +188.45 to the prediction. The stirrup ratio, with a value of 0.07 %, also adds a +61.32 to the response. In contrast, the yield strength of steel (fy), at 379 MPa, contributes −54.71, indicating a negative impact on the response. Beam breadth, measuring 200 mm, contributes an additional +53.72, while the cross-sectional area of steel and concrete compressive strength contribute +15.95 and −13.33, respectively. The remaining features have marginal impact on the overall prediction. It should be kept in mind that current values are specific to one set of points in the data and the values may change for the other available sets.

7. Conclusion

This study explores the applicability of various machine learning techniques for predicting the peak response of reinforced concrete (RC) beams under impact loads. The investigation encompasses both individual and ensemble methods, including support vector machine, Gaussian process regression (GPR), K-nearest neighbor (KNN), gene expression programming, random forest, decision tree (DT), ensemble

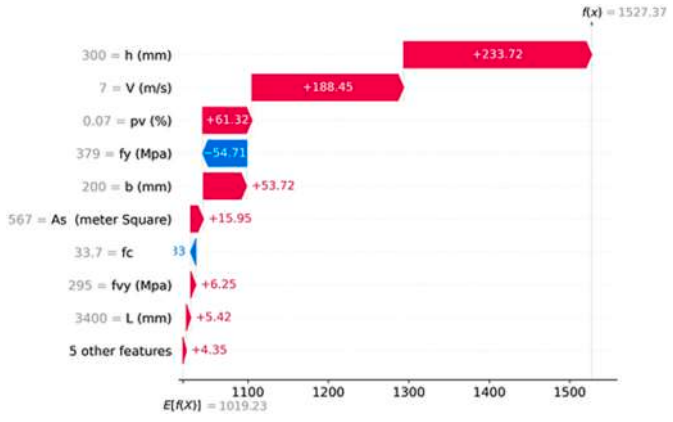

Fig. 32. SHAP water fall plot.

boosted tree, adaptive boosting tree, gradient boosting algorithm, stochastic gradient descent, and scaled conjugate gradient algorithms. Key independent parameters for model development included beam width and depth, total and clear span, longitudinal and transverse reinforcement ratios, concrete strength, steel yield strength, deflection, and impact velocity, mass, and force.

- All models, except KNN, demonstrated satisfactory generalization capabilities, achieving R^2 and R values above acceptable thresholds (0.8 for R^2 and 0.9 for R) across the entire datasets. Statistical errors such as RMSE (ranging from 80 to 250), a-10 (ranging from 30 to 70), MAE (ranging from 60 to 180), and a-20 (ranging from 20 to 60) were within permissible limits. Moreover, the ratio of actual to predicted response values averaged close to 1 for all models, though occasional deviations—with estimated-to-actual ratios falling below 0.5 or exceeding 1.5—suggested instances of overfitting and underfitting.
- Among the models, GPR emerged as the most effective, exhibiting the highest R^2 value of 0.95 and the lowest RMSE and MAE values (95 and 60, respectively). It also recorded the highest a-10 and a-20 values. Conversely, the KNN model performed the poorest, with the lowest R^2 value of 0.75 and relatively high MAE and RMSE values.
- Beam depth, impact velocity, and beam breadth are the dominant factors influencing the response force, with mean absolute SHAP values of +298.8, +182.94, and +92.02, respectively. These high values indicate the substantial impact of these parameters on the model's predictions. While other parameters such as stirrup ratio, compressive strength of concrete, and yield strength of steel play secondary roles, they still contribute to the response force.
- The SHAP analysis further reveals a direct proportionality of peak response with beam depth, width, impact velocity, and yield strength of steel. The remaining parameters influence on peak response is less clear.
- While the results demonstrate satisfactory performance of machine learning models in predicting peak response under impact loading, further research could explore integrating hybrid models that combine machine learning with physics-based approaches to improve interpretability and accuracy. Additionally, extending the database to include more diverse case studies involving varying material properties and loading conditions would enhance the models' generalization capabilities.

In conclusion, this study provides valuable insights into the applicability of machine learning techniques for predicting the peak response of RC beams under impact loads. The findings highlight the effectiveness of GPR and underscore the importance of beam depth, impact velocity, and beam breadth in determining the response force. These results can inform future research and design practices in structural engineering.

CRediT authorship contribution statement

Ali Husnain: Validation, Software, Methodology, Formal analysis, Conceptualization. **Munir Iqbal:** Writing – original draft, Validation, Software, Methodology, Formal analysis, Conceptualization. **Hafiz Ahmed Waqas:** Writing – review & editing, Software, Methodology, Formal analysis, Conceptualization. **Mohammed El-Meligy:** Conceptualization, Funding acquisition, Methodology, Software, Writing – review & editing. **Muhammad Faisal Javed:** Writing – review & editing, Software, Methodology, Formal analysis, Conceptualization. **Rizwan Ullah:** Conceptualization, Data curation, Methodology, Software, Validation.

Data availability

Data will be made available on request.

Funding

This research received no external funding.

Declaration of competing interest

The authors affirm that there are no identifiable conflicting financial interests or personal relationships that might have been perceived to affect the integrity of the work presented in this paper.

Data availability

Link to the data is attached in the data availability statement within manuscript.

References

[1] J.Z. C, P.C. Li, J.Y. Jia, H. Wu, L.L. Ma, Impact-resistant design of RC slabs in nuclear power plant buildings, Nucl. Eng. Technol. 54 (2022) 3745–3765, https://doi.org/10.1016/j.net.2022.05.027.
[2] W.F. Zhao Huida, Jihong Ye, Qian, Jiang, Dynamic responses and damage behavior of hollow RC piers against rockfall impact, Thin-Walled Struct. 187 (2023) 110771, https://doi.org/10.1016/j.tws.2023.110771.
[3] H.C. Li Wensu, Zhijie Huang, Hong Hao, Tuan T. Ngo, Thong M. Pham, Kge Jack Yeoh, Dynamic response of monolithic and precast concrete joint with wet connections under impact loads, Eng. Struct. 250 (2022) 113434–NA, https://doi.org/10.1016/j.engstruct.2021.113434.
[4] M.D. Adem Yimer Tanish, Dynamic response of concrete beams reinforced with GFRP and steel bars under impact loading, Eng. Fail. Anal. 161 (2024) 108329, https://doi.org/10.1016/j.engfailanal.2024.108329.
[5] T.E. Yılmaz R. Tuğrul, Norimitsu Kıshı, Özgür Anıl, Investigation of impact behavior of shear deficient RC beams using nonlinear FEA, Mech. Base. Des. Struct. Mach. 52 (2022) 848–866, https://doi.org/10.1080/15397734.2022.2124173.
[6] K.X. Sun Jie, Yongding Tian, Dynamic behavior investigation of reinforced concrete bridge under multi-hazard effect of rockfall impact and material corrosion, Structures 58 (2023) 105557, https://doi.org/10.1016/j.istruc.2023.105557.
[7] X.W. Wang Qinyuan, Wenlong Chen, Experimental study on the impact resistance of steel fiber reinforced all-lightweight concrete beams under single and hybrid mixing conditions, Buildings 13 (2023) 1251, https://doi.org/10.3390/buildings13051251.
[8] J.L. Sánchez-Haro Ignacio, Capellán, Guillermo, Simplified model to consider influence of gravity on impacts on structures: experimental and numerical validation, Int. J. Impact Eng. 173 (2023) 104474, https://doi.org/10.1016/j.ijimpeng.2022.104474.
[9] H.T. Hao, T. Tung, Huawei Li, Thong M. Pham, Wensu Chen, On the accuracy, reliability and controllability of impact tests of RC beams, Int. J. Impact Eng. 157 (2021) 103979–NA, https://doi.org/10.1016/j.ijimpeng.2021.103979.
[10] P. Foraboschi, Falling mass bearing capacity of reinforced concrete beams, Eng. Fail. Anal. 138 (2022) 106396, https://doi.org/10.1016/j.engfailanal.2022.106396.
[11] D.Y. Zhao Wei-Jian, Sashi K. Kunnath, Simplified approach for assessing shear resistance of reinforced concrete beams under impact loads, ACI Struct. J. 113 (2016) 747–756, https://doi.org/10.14359/51688617.
[12] S. Abrate, Modeling of impacts on composite structures, Compos. Struct. 51 (2001) 129–138, https://doi.org/10.1016/s0263-8223(00)00138-0.
[13] J. Ožbolt, A. Sharma, Numerical simulation of reinforced concrete beams with different shear reinforcements under dynamic impact loads, Int. J. Impact Eng. 38 (2011) 940–950.
[14] J.Y. Liu Yiping, Yunlei Zhao, Yuan Li, Dynamic behavior analysis of I-shaped RC beams under combined blast and impact loads, Appl. Sci. 13 (2023) 1943, https://doi.org/10.3390/app13031943.
[15] W.F. Zhao Huida, Jihong Ye, Jiang Qian, Performance of precast concrete bridge piers with grouted sleeve connections against vehicle impact, Structures 44 (2022) 1874–1885, https://doi.org/10.1016/j.istruc.2022.09.004.
[16] J.F. Dong, Q.Y. Wang, Z.W. Guan, H.K. Chai, High-temperature behaviour of basalt fibre reinforced concrete made with recycled aggregates from earthquake waste, J. Build. Eng. 48 (2022). https://doi.org/10.1016/j.jobe.2021.103895.
[17] S.D.L. Adhikary Bing, Fujikake, Kazunori, State-of-the-art review on low-velocity impact response of reinforced concrete beams, Mag. Concr. Res. 68 (2016) 701–723, https://doi.org/10.1680/jmacr.15.00084.
[18] Z.Y. Wang Jiachuan, Youzhu Lin, Feng Fan, Menghan Sun, Experimental and analytical study on the double steel plates-UHPC sandwich slabs under low-velocity impact, Thin-Walled Struct. 184 (2023) 110548, https://doi.org/10.1016/j.tws.2023.110548.
[19] T.M.C. Pham Wensu, Hong Hao, Review on impact response of reinforced concrete beams: contemporary understanding and unsolved problems, Adv. Struct. Eng. 24 (2021) 2282–2303, https://doi.org/10.1177/1369433221997716.

[20] P.D. T, M. Tran Thong, Hong Hao, Experimental and analytical investigations of prefabricated segmental concrete beams post-tensioned with unbonded steel/FRP tendons subjected to impact loading, Int. J. Impact Eng. 186 (2024) 104868, https://doi.org/10.1016/j.ijimpeng.2023.104868.
[21] Y. Yu, Y. Zheng, X.-Y. Zhao, Mesoscale modeling of recycled aggregate concrete under uniaxial compression and tension using discrete element method, Construct. Build. Mater. 268 (2021) 121116, https://doi.org/10.1016/j.conbuildmat.2020.121116.
[22] K.Y. Al-Bukhaiti Liu, Zhao Shichun, Abas, Hussein, Dynamic simulation of CFRP-shear strengthening on existing square RC members under unequal lateral impact loading, Struct. Concr. 24 (2022) 1572–1596, https://doi.org/10.1002/suco.202100814.
[23] A.Q.K. Bhatti Norimitsu, Mikami, Hiroshi; Ando, Tomohiro, Elasto-plastic impact response analysis of shear-failure-type RC beams with shear rebars, Mater. Des. 30 (2009) 502–510, https://doi.org/10.1016/j.matdes.2008.05.068.
[24] S.Z. Lin Bing, Sumei Zhang, Yijie Zhang, Xiamin Hu, Dynamic responses of concrete-filled steel tubes impacted horizontally by a rigid vehicle: experimental study and numerical modelling, Thin-Walled Struct. 199 (2024) 111826, https://doi.org/10.1016/j.tws.2024.111826.
[25] C.G. Zhang Gholamreza, Mousavi, asma alsadat, state-of-the-art review on responses of RC structures subjected to lateral impact loads, Arch. Comput. Methods Eng. 28 (2020) 2477–2507, https://doi.org/10.1007/s11831-020-09467-5.
[26] S.D.L. Adhikary Bing, Kazunori Fujikake, Dynamic behavior of reinforced concrete beams under varying rates of concentrated loading, Int. J. Impact Eng. 47 (2012) 24–38, https://doi.org/10.1016/j.ijimpeng.2012.02.001.
[27] S.D.L. Adhikary Bing, Kazunori Fujikake, Low velocity impact response of reinforced concrete beams: experimental and numerical investigation, Int. J. Prot. Struct. 6 (2015) 81–111, https://doi.org/10.1260/2041-4196.6.1.81.
[28] S.V. Saatci Frank J, Effects of shear mechanisms on impact behavior of reinforced concrete beams, ACI Struct. J. 106 (2009) 78–86, https://doi.org/10.14359/56286.
[29] O. Alshboul, G. Almasabha, K.F. Al-Shboul, A. Shehadeh, A comparative study of shear strength prediction models for SFRC deep beams without stirrups using Machine learning algorithms, Structures 55 (2023) 97–111, https://doi.org/10.1016/j.istruc.2023.06.026.
[30] R. Niyirora, W. Ji, E. Masengesho, J. Munyaneza, F. Niyonyungu, R. Nyirandayisabye, Intelligent damage diagnosis in bridges using vibration-based monitoring approaches and machine learning: a systematic review, Results Eng 16 (2022) 100761, https://doi.org/10.1016/j.rineng.2022.100761.
[31] Q.W. Li Zitong, Ling Li, Hong Hao, Wensu Chen, Yanda Shao, Machine learning prediction of structural dynamic responses using graph neural networks, Comput. Struct. 289 (2023) 107188, https://doi.org/10.1016/j.compstruc.2023.107188.
[32] D.D. Lai Cristoforo, Xiao, Yan, Interpretable machine-learning models for maximum displacements of RC beams under impact loading predictions, Eng. Struct. 281 (2023) 115723, https://doi.org/10.1016/j.engstruct.2023.115723.
[33] Y. Yu, J. Xu, J. Su, L. Xu, Y. Luo, Investigating specimen size and shape effects on compressive mechanical behaviors of recycled aggregate concrete using discrete element mesoscale modeling, Construct. Build. Mater. 438 (2024) 137196, https://doi.org/10.1016/j.conbuildmat.2024.137196.
[34] Y. Yu, G.-H. Fang, R. Kurda, A.R. Sabuj, X.-Y. Zhao, An agile, intelligent and scalable framework for mix design optimization of green concrete incorporating recycled aggregates from precast rejects, Case Stud. Constr. Mater. 20 (2024) e03156, https://doi.org/10.1016/j.cscm.2024.e03156.
[35] J.H. Li YiFei, ChunYuan Liu, DanDan Xiao, Numerical study of the effect of fatigue on impact performance of RC beams, Sci. China Technol. Sci. 66 (2022) 346–362, https://doi.org/10.1007/s11431-022-2189-9.
[36] A.M. Hernández-Díaz, J. Pérez-Aracil, E. Lorente-Ramos, C.M. Marina, C. Peláez-Rodríguez, S. Salcedo-Sanz, Machine learning as alternative strategy for the numerical prediction of the shear response in reinforced and prestressed concrete beams, Results Eng 22 (2024) 102139, https://doi.org/10.1016/j.rineng.2024.102139.
[37] K.F. Al-Shboul, Unraveling the complex interplay between soil characteristics and radon surface exhalation rates through machine learning models and multivariate analysis, Environ. Pollut. 336 (2023) 122440, https://doi.org/10.1016/j.envpol.2023.122440.
[38] A. Shehadeh, O. Alshboul, K.F. Al-Shboul, O. Tatari, An expert system for highway construction: multi-objective optimization using enhanced particle swarm for optimal equipment management, Expert Syst. Appl. 249 (2024) 123621, https://doi.org/10.1016/j.eswa.2024.123621.
[39] A.I.M. Almadi, R.E. Al Mamlook, I. Ullah, O. Alshboul, N. Bandara, A. Shehadeh, Vehicle collisions analysis on highways based on multi-user driving simulator and multinomial logistic regression model on US highways in Michigan, Int. J. Crashworthiness 28 (2023) 770–785, https://doi.org/10.1080/13588265.2022.2130608.
[40] M.K. Almustafa, M.L. Nehdi, Machine learning model for predicting structural response of RC columns subjected to blast loading, Int. J. Impact Eng. 162 (2022) 104145.
[41] M.K. Almustafa, M.L. Nehdi, Machine learning prediction of structural response of steel fiber-reinforced concrete beams subjected to far-field blast loading, Cem. Concr. Compos. 126 (2022) 104378.
[42] M.K. Almustafa, M.L. Nehdi, Novel hybrid machine learning approach for predicting structural response of RC beams under blast loading, Structures 39 (2022) 1092–1106, https://doi.org/10.1016/j.istruc.2022.04.007.
[43] a Yus, C.H.S. Bruns, D. Turnbull, S. Dolnicar, A. Chapple, T. Adorno, M. Horkheimer, K. Negus, M. Pickering, J. Leckenby, H. Li, MILLS, Anthony F. Transferencia de calor, Irwin, A. (2004): 161-169. Fenkçi IV, Maternal Fizyoloji. "Çiçek MN, Ed." Kadın Hastalıkları ve Doğum Bilgisi, Öncü Basımevi, R.C. Team, G. Payerle, T. Adorno, M. Horkheimer, D., S. Dolnicar, A. Chapple, T. Adorno, M. Horkheimer, K. Negus, M. Pickering, J. Leckenby, H. Li, Scholar 下午9, Mass Commun. Soc. 10 (1974) 349–169, http://www.sciencedirect.com/science/article/pii/S0160738315000444, 1995.
[44] X.-Q. Zhou, B.-G. Huang, X.-Y. Wang, Y. Xia, Deep learning-based rapid damage assessment of RC columns under blast loading, Eng. Struct. 271 (2022) 114949, https://doi.org/10.1016/j.engstruct.2022.114949.
[45] M.K. Almustafa, M.L. Nehdi, Machine learning model for predicting structural response of RC slabs exposed to blast loading, Eng. Struct. 221 (2020) 111109, https://doi.org/10.1016/j.engstruct.2020.111109.
[46] Y. Shen, L. Wu, S. Liang, Explainable machine learning-based model for failure mode identification of RC flat slabs without transverse reinforcement, Eng. Fail. Anal. 141 (2022) 106647, https://doi.org/10.1016/j.engfailanal.2022.106647.
[47] W.J. Baniya, W. Zaki, C.-M.R. Farrag, D.-A. Rashed, Behavior of composite pre-flat slabs in resisting punching shear forces, Alex. Eng. J. 59 (2020) 333–347, https://doi.org/10.1016/j.aej.2019.12.045.
[48] D.-K. Thai, T. Minh Tu, T.Q. Bui, T.T. Bui, Gradient tree boosting machine learning on predicting the failure modes of the RC panels under impact loads, Eng. Comput. 37 (2021) 1–12, https://doi.org/10.1007/s00366-019-00842-w.
[49] D. Lai, C. Demartino, Y. Xiao, Interpretable machine-learning models for maximum displacements of RC beams under impact loading predictions, Eng. Struct. 281 (2023) 115723, https://doi.org/10.1016/j.engstruct.2023.115723.
[50] J. Zhang, Y. Jing, S. Chen, D. Yang, Data-driven prediction approach for RC beam performance under low velocity impact loading, Eng. Fail. Anal. 143 (2023) 106884, https://doi.org/10.1016/j.engfailanal.2022.106884.
[51] G. Almasabha, A. Shehadeh, O. Alshboul, O. Al Hattamleh, Structural performance of buried reinforced concrete pipelines under deep embankment soil, Constr. Innov. 24 (2024) 1280–1296, https://doi.org/10.1108/CI-10-2021-0196.
[52] W. Zhao, J. Qian, P. Jia, Peak response prediction for RC beams under impact loading, Shock Vib. 2019 (2019) 6813693, https://doi.org/10.1155/2019/6813693.
[53] P. Isaac, A. Darby, T. Ibell, M. Evernden, Experimental investigation into the force propagation velocity due to hard impacts on reinforced concrete members, Int. J. Impact Eng. 100 (2017) 131–138, https://doi.org/10.1016/j.ijimpeng.2016.09.005.
[54] T.M. Pham, H. Hao, Plastic hinges and inertia forces in RC beams under impact loads, Int. J. Impact Eng. 103 (2017) 1–11, https://doi.org/10.1016/j.ijimpeng.2016.12.016.
[55] C. Zhang, G. Gholipour, A.A. Mousavi, Nonlinear dynamic behavior of simply-supported RC beams subjected to combined impact-blast loading, Eng. Struct. 181 (2019) 124–142, https://doi.org/10.1016/j.engstruct.2018.12.014.
[56] M.M. I, C. Yi, J.O.D. R, F.Y. T, P.J. T, Reinforced concrete beams under drop-weight impact loads, Comput. Concr. 3 (2006) 79–90, https://doi.org/10.12989/CAC.2006.3.2_3.079.
[57] B.P. Hughes, A.T. Mahmood, Impact behaviour of prestressed concrete beams in flexure, Mag. Concr. Res. 36 (1984) 157–164, https://doi.org/10.1680/macr.1984.36.128.157.
[58] M. Goldston, A. Remennikov, M.N. Sheikh, Experimental investigation of the behaviour of concrete beams reinforced with GFRP bars under static and impact loading, Eng. Struct. 113 (2016) 220–232, https://doi.org/10.1016/j.engstruct.2016.01.044.
[59] Z. De-Bo, Y. Wei-Jian, K.K. S, Shear mechanisms in reinforced concrete beams under impact loading, J. Struct. Eng. 143 (2017) 4017089, https://doi.org/10.1061/(ASCE)ST.1943-541X.0001818.
[60] N.K. and H. Mikami, Empirical Formulas for Designing Reinforced Concrete Beams under Impact Loading, ACI Struct. J. 109 (n.d.). https://doi.org/10.14359/51683870.
[61] A.Q. Bhatti, N. Kishi, H. Mikami, T. Ando, Elasto-plastic impact response analysis of shear-failure-type RC beams with shear rebars, Mater. Des. 30 (2009) 502–510, https://doi.org/10.1016/j.matdes.2008.05.068.
[62] S. Das Adhikary, B. Li, K. Fujikake, Low velocity impact response of reinforced concrete beams: experimental and numerical investigation, Int. J. Prot. Struct. 6 (2015) 81–111, https://doi.org/10.1260/2041-4196.6.1.81.
[63] L. Jin, R. Zhang, G. Dou, X. 健栋, X. Du, Experimental and numerical study of reinforced concrete beams with steel fibers subjected to impact loading, Int. J. Damage Mech. 27 (2018) 1058–1083, https://doi.org/10.1177/1056789517721616.
[64] N. Kishi, H. Mikami, K.G. Matsuoka, T. Ando, Impact behavior of shear-failure-type RC beams without shear rebar, Int. J. Impact Eng. 27 (2002) 955–968, https://doi.org/10.1016/S0734-743X(01)00149-X.
[65] J.F. Dong, Y. Xu, Z.W. Guan, Q.Y. Wang, Freeze-thaw behaviour of basalt fibre reinforced recycled aggregate concrete filled CFRP tube specimens, Eng. Struct. 273 (2022). https://doi.org/10.1016/j.engstruct.2022.115088.
[66] D.-B. Zhao, D. Zhao, Anti-impact behavior and design method for RC beams, Zhendong Yu Chongji/Journal Vib. Shock. 34 (2015) 139–145, https://doi.org/10.13465/j.cnki.jvs.2015.11.025.
[67] F. Kazunori, L. Bing, S. Sam, Impact response of reinforced concrete beam and its analytical evaluation, J. Struct. Eng. 135 (2009) 938–950, https://doi.org/10.1061/(ASCE)ST.1943-541X.0000039.
[68] S. Tachibana, H. Masuya, S. Nakamura, Performance based design of reinforced concrete beams under impact, Nat. Hazards Earth Syst. Sci. 10 (2010) 1069–1078, https://doi.org/10.5194/nhess-10-1069-2010.

[69] G. Dou, X. Du, L. Li, D.D.-B. Zhao, D.D.-B. Zhao, Y.J. Yu, C.-H. Kim, J.-Y. Cho, N. Kishi, H. Mikami, K.G. Matsuoka, T. Ando, L. Jin, R. Zhang, G. Dou, X. 健栋, X. Du, S. Das Adhikary, B. Li, K. Fujikake, A.Q. Bhatti, N. Kishi, H. Mikami, T. Ando, N. K, H. Mikami, Z. De-Bo, Y. Wei-Jian, K.K. S, Experimental study on the behavior of high strength reinforced concrete beams under impact load, Int. J. Impact Eng. 27 (2015) 139–145, https://doi.org/10.13465/j.cnki.jvs.2015.11.025.

[70] G. Dou, X. Du, L. Li, Experimental study on the behavior of high strength reinforced concrete beams under impact load, Tianjin Daxue Xuebao (Ziran Kexue Yu Gongcheng Jishu Ban)/Journal Tianjin Univ. Sci. Technol. 47 (2014) 1072–1080, https://doi.org/10.11784/tdxbz201403083.

[71] A. Gholampour, A.H. Gandomi, T. Ozbakkaloglu, New formulations for mechanical properties of recycled aggregate concrete using gene expression programming, Construct. Build. Mater. 130 (2017) 122–145, https://doi.org/10.1016/j.conbuildmat.2016.10.114.

[72] A. Ahmad, K. Chaiyasarn, F. Farooq, W. Ahmad, S. Suparp, F. Aslam, Compressive strength prediction via gene expression programming (Gep) and artificial neural network (ann) for concrete containing rca, Buildings (2021), https://doi.org/10.3390/buildings11080324.

[73] Y. Liu, J. Dong, S. Yuan, K. Li, X. Hu, Q. Wang, Variable fatigue loading effects on corrugated steel box girders with recycled concrete, J. Constr. Steel Res. 215 (2024) 108526. https://doi.org/10.1016/j.jcsr.2024.108526.

[74] S.K. Babanajad, A.H. Gandomi, A.H. Alavi, New prediction models for concrete ultimate strength under true-triaxial stress states: an evolutionary approach, Adv. Eng. Software 110 (2017) 55–68, https://doi.org/10.1016/j.advengsoft.2017.03.011.

[75] A. Iqbal, P. Sambyal, C.M. Koo, 2D MXenes for electromagnetic shielding: a review, Adv. Funct. Mater. 30 (2020) 1–25, https://doi.org/10.1002/adfm.202000883.

[76] V. Calofir, R.-I. Munteanu, M.-S. Simoiu, K.-C. Lemnaru, Innovative approach to estimate structural damage using linear regression and K-nearest neighbors machine learning algorithms, Results Eng 22 (2024) 102250, https://doi.org/10.1016/j.rineng.2024.102250.

[77] J. Xin, W. Xu, B. Cao, T. Wang, S. Zhang, A deep-learning-based MAC for integrating channel access, rate adaptation and channel switch. https://arxiv.org/abs/2406.02291v1, 2024. (Accessed 4 August 2024).

[78] J. Zhang, Y. Huang, F. Aslani, G. Ma, B. Nener, A hybrid intelligent system for designing optimal proportions of recycled aggregate concrete, J. Clean. Prod. 273 (2020) 122922–NA, https://doi.org/10.1016/j.jclepro.2020.122922.

[79] Q. Ge, X. Huang, S. Fang, S. Guo, Y. Liu, W. Lin, M. Xiong, Conditional generative adversarial networks for individualized treatment effect estimation and treatment selection, Front. Genet. 11 (2020) 585804, https://doi.org/10.3389/fgene.2020.585804.

[80] S. Uddin, I. Haque, H. Lu, M.A. Moni, E. Gide, Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction, Sci. Rep. 12 (2022) 6256, https://doi.org/10.1038/s41598-022-10358-x.

[81] I. Syarif, E. Zaluska, A. Prugel-Bennett, G. Wills, Application of bagging, boosting and stacking to intrusion detection, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 7376 LNAI (2012) 593–602, https://doi.org/10.1007/978-3-642-31537-4_46.

[82] S.M. Lundberg, G. Erion, H. Chen, A. DeGrave, J.M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.I. Lee, From local explanations to global understanding with explainable AI for trees, Nat. Mach. Intell. 2 (2020) 56–67, https://doi.org/10.1038/s42256-019-0138-9.

[83] X. Bao, H. Yuan, J. Shen, C. Liu, X. Chen, H. Cui, Numerical analysis of seismic response of a circular tunnel-rectangular underpass system in liquefiable soil, Comput. Geotech. 174 (2024). https://doi.org/10.1016/j.compgeo.2024.106642.

[84] L. Bottou, Stochastic gradient descent tricks, in: G. Montavon, G.B. Orr, K.-R. Müller (Eds.), Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 421–436, https://doi.org/10.1007/978-3-642-35289-8_25.

[85] L. Shi, S.T.K. Lin, Y. Lu, L. Ye, Y.X. Zhang, Artificial neural network based mechanical and electrical property prediction of engineered cementitious composites, Construct. Build. Mater. 174 (2018) 667–674, https://doi.org/10.1016/j.conbuildmat.2018.04.127.

[86] J.S. Chou, C.F. Tsai, A.D. Pham, Y.H. Lu, Machine learning in concrete strength simulations: multi-nation data analytics, Construct. Build. Mater. 73 (2014) 771–780, https://doi.org/10.1016/J.CONBUILDMAT.2014.09.054.

[87] T. Yasuda, S. Ookawara, S. Yoshikawa, H. Matsumoto, Machine learning and data-driven characterization framework for porous materials: permeability prediction and channeling defect detection, Chem. Eng. J. 420 (2021) 130069, https://doi.org/10.1016/j.cej.2021.130069.

[88] H. Guo, J. Zhao, J. Yin, L. Yao, Structural testing of polyimide nanocomposite films with SAXS and SVM-PUK, Polym. Test. 70 (2018) 30–38, https://doi.org/10.1016/j.polymertesting.2018.06.025.

[89] A. Ghosh, F. Mertens, G. Bernardi, M.G. Santos, N.S. Kern, C.L. Carilli, T. L. Grobler, L.V.E. Koopmans, D.C. Jacobs, A. Liu, Foreground modelling via Gaussian process regression: an application to HERA data, Mon. Not. R. Astron. Soc. 495 (2020) 2813–2826.

[90] B. Huang, F. Kang, X. Li, S. Zhu, Underwater dam crack image generation based on unsupervised image-to-image translation, Autom. Constr. 163 (2024). https://doi.org/10.1016/j.autcon.2024.105430.

[91] H. Wei, S. Zhao, Q. Rong, H. Bao, Predicting the effective thermal conductivities of composite materials and porous media by machine learning methods, Int. J. Heat Mass Transf. 127 (2018) 908–916, https://doi.org/10.1016/j.ijheatmasstransfer.2018.08.082.

[92] Y. Zhang, X. Xu, Machine learning lattice constants for cubic perovskite A2XY6 compounds, J. Solid State Chem. 291 (2020) 121558, https://doi.org/10.1016/j.jssc.2020.121558.

[93] D.A. Van Dao Hojjat, Ly, Bang Hai, Lu Minh Le, Vuong Minh Le, Tien-Thinh Le, Binh Thai Pham, A sensitivity and robustness analysis of GPR and ANN for high-performance concrete compressive strength prediction using a Monte Carlo simulation, Sustainability 12 (2020) 830–NA, https://doi.org/10.3390/su12030830.

[94] Y. Zhang, X. Xu, Predicting springback radii and angles in air bending of high-strength sheet steel through Gaussian process regressions, Int. J. Interact. Des. Manuf. 16 (2022) 863–870, https://doi.org/10.1007/s12008-022-00945-7.

[95] N. Mohamad, F. Zaini, A. Johari, I. Yassin, A. Zabidi, in: Comparison between Levenberg-Marquardt and Scaled Conjugate Gradient Training Algorithms for Breast Cancer Diagnosis Using MLP, 2010, pp. 1–7, https://doi.org/10.1109/cspa.2010.5545325.

[96] D. Van Dao, H.B. Ly, S.H. Trinh, T.T. Le, B.T. Pham, Artificial intelligence approaches for prediction of compressive strength of geopolymer concrete, Materials 12 (2019), https://doi.org/10.3390/ma12060983.

[97] G. Bruce, M. Lightfoot, The use of artificial neural networks to model distortion caused by welding, Int. J. Model. Simul. 27 (2007) 32–37, https://doi.org/10.1080/02286203.2007.11442396.

[98] J. Bilski, J. Smoląg, J.M. Zurada, Parallel approach to the Levenberg-marquardt learning algorithm for feedforward neural networks, in: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, J.M. Zurada (Eds.), Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci., Springer International Publishing, Cham, 2015, pp. 3–14, https://doi.org/10.1007/978-3-319-19324-3_1.

[99] J. Sabi'u, K.O. Aremu, A. Althobaiti, A. Shah, Scaled three-term conjugate gradient methods for solving monotone equations with application, Symmetry (Basel). 14 (2022) 1–20, https://doi.org/10.3390/sym14050936.

[100] M.F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Network. 6 (1993) 525–533, https://doi.org/10.1016/S0893-6080(05)80056-5.

[101] S. Kostić, D. Vasović, Prediction model for compressive strength of basic concrete mixture using artificial neural networks, Neural Comput. Appl. 26 (2015) 1005–1024, https://doi.org/10.1007/s00521-014-1763-1.

[102] J. Wang, J. Ji, Z. Jiang, L. Sun, Traffic Flow Prediction Based on Spatiotemporal Potential Energy Fields, IEEE Trans. Knowl. Data Eng. 35 (2023) 9073–9087. https://doi.org/10.1109/TKDE.2022.3221183.

[103] M. Schonlau, R.Y. Zou, The random forest algorithm for statistical learning, STATA J. 20 (2020) 3–29, https://doi.org/10.1177/1536867X20909688.

[104] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and regression trees, Classif. Regres. Trees (2017) 1–358, https://doi.org/10.1201/9781315139470.

[105] R.L. Marchese Robinson, A. Palczewska, J. Palczewski, N. Kidley, Comparison of the predictive performance and interpretability of random forest and linear models on benchmark data sets, J. Chem. Inf. Model. 57 (2017) 1773–1792.

[106] H. Dabiri, V. Farhangi, M.J. Moradi, M. Zadehmohamad, M. Karakouzian, Applications of decision tree and random forest as tree-based machine learning techniques for analyzing the ultimate strain of spliced and non-spliced reinforcement bars, Appl. Sci. 12 (2022) 1–13, https://doi.org/10.3390/app12104851.

[107] A.V. Konstantinov, L.V. Utkin, Interpretable machine learning with an ensemble of gradient boosting machines, Knowl. Base Syst. 222 (2021) 106993, https://doi.org/10.1016/j.knosys.2021.106993.

[108] M.S. Alajmi, A.M. Almeshal, Least squares boosting ensemble and quantum-behaved particle swarm optimization for predicting the surface roughness in face milling process of aluminum material, Appl. Sci. 11 (2021), https://doi.org/10.3390/app11052126.

[109] K.W. Walker, Z. Jiang, Application of adaptive boosting (AdaBoost) in demand-driven acquisition (DDA) prediction: a machine-learning approach, J. Acad. Librarian 45 (2019) 203–212, https://doi.org/10.1016/j.acalib.2019.02.013.

[110] D.C. Feng, Z.T. Liu, X.D. Wang, Y. Chen, J.Q. Chang, D.F. Wei, Z.M. Jiang, Machine learning-based compressive strength prediction for concrete: an adaptive boosting approach, Construct. Build. Mater. 230 (2020) 117000, https://doi.org/10.1016/j.conbuildmat.2019.117000.

[111] Y. Ding, H. Zhu, R. Chen, R. Li, An efficient AdaBoost algorithm with the multiple thresholds classification, Appl. Sci. 12 (2022), https://doi.org/10.3390/app12125872.

[112] P. Wu, H. Zhao, Some analysis and research of the AdaBoost algorithm, Commun. Comput. Inf. Sci. 134 (2011) 3–5, https://doi.org/10.1007/978-3-642-18129-0_1.

[113] A. Natekin, A. Knoll, Gradient boosting machines, a tutorial, Front. Neurorobot. 7 (2013) 21, https://doi.org/10.3389/fnbot.2013.00021.

[114] M. Yao, Y. Zhu, J. Li, H. Wei, P. He, Research on predicting line loss rate in low voltage distribution network based on gradient boosting decision tree, Energies 12 (12) (2019) 2522, https://doi.org/10.3390/EN12132522 (2019) 2522.

[115] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140, https://doi.org/10.1007/bf00058655.

[116] H.I. Erdal, O. Karakurt, E. Namli, High performance concrete compressive strength forecasting using ensemble models based on discrete wavelet transform, Eng. Appl. Artif. Intell. 26 (2013) 1246–1254, https://doi.org/10.1016/J.ENGAPPAI.2012.10.014.

[117] S.H. Hwang, S. Mangalathu, J. Shin, J.S. Jeon, Machine learning-based approaches for seismic demand and collapse of ductile reinforced concrete

building frames, J. Build. Eng. 34 (2021) 101905, https://doi.org/10.1016/j.jobe.2020.101905.
[118] S. Mahjoubi, W. Meng, Y. Bao, Auto-tune learning framework for prediction of flowability, mechanical properties, and porosity of ultra-high-performance concrete (UHPC), Appl. Soft Comput. 115 (2022) 108182, https://doi.org/10.1016/j.asoc.2021.108182.
[119] J. Cao, J. Gao, H. Nikafshan Rad, A.S. Mohammed, M. Hasanipanah, J. Zhou, A novel systematic and evolved approach based on XGBoost-firefly algorithm to predict Young's modulus and unconfined compressive strength of rock, Eng. Comput. 38 (2022) 3829–3845, https://doi.org/10.1007/s00366-020-01241-2.
[120] M. Vega García, J.L. Aznarte, Shapley additive explanations for NO2 forecasting, Ecol. Inform. 56 (2020) 101039, https://doi.org/10.1016/j.ecoinf.2019.101039.
[121] Y. Nohara, K. Matsumoto, H. Soejima, N. Nakashima, Explanation of machine learning models using shapley additive explanation and application for real data in hospital, Comput. Methods Programs Biomed. 214 (2022) 106584, https://doi.org/10.1016/j.cmpb.2021.106584.
[122] S.F.A. Shah, B. Chen, M. Zahid, M.R. Ahmad, Compressive strength prediction of one-part alkali activated material enabled by interpretable machine learning, Construct. Build. Mater. 360 (2022) 129534, https://doi.org/10.1016/j.conbuildmat.2022.129534.
[123] Z. Wu, C. Ma, L. Zhang, H. Gui, J. Liu, Z. Liu, Predicting and compensating for small-sample thermal information data in precision machine tools: A spatial-temporal interactive integration network and digital twin system approach, Appl. Soft Comput. 161 (2024) 111760, https://doi.org/10.1016/J.ASOC.2024.111760.
[124] R. Alyousef, M.F. Rehman, M. Khan, M. Fawad, A.U. Khan, A.M. Hassan, N. A. Ghamry, Machine learning-driven predictive models for compressive strength of steel fiber reinforced concrete subjected to high temperatures, Case Stud. Constr. Mater. 19 (2023) e02418, https://doi.org/10.1016/j.cscm.2023.e02418.
[125] B.L. Satadru Das Adhikary and Kazunori Fujikake, Effects of High Loading Rate on Reinforced Concrete Beams (with Appendix), ACI Struct. J. 111 (n.d.). https://doi.org/10.14359/51686579.