

Full length article

Novel hybrid WOA-GBM model for patch loading resistance prediction of longitudinally stiffened steel plate girders

Viet-Linh Tran ^{*}, Duy-Duan Nguyen

Department of Civil Engineering, Vinh University, Vinh 461010, Viet Nam

ARTICLE INFO

Keywords:

Adaptive boosting
Extreme gradient boosting
Gradient boosting machine
Patch loading resistance
Steel plate girder
Whale optimization algorithm

ABSTRACT

In steel plate girders (SPGs), a patch loading usually causes a local failure in the vicinity of the loading area of the girder web. However, estimating the patch loading resistance (PLR) of SPGs is challenging due to the complexity of the problem. This paper aims to develop a novel hybrid WOA-GBM model based on a whale optimization algorithm (WOA) and a gradient boosting machine (GBM) for predicting the PLR of longitudinally SPGs. Firstly, 137 tests of longitudinally stiffened SPGs subjected to patch loading are carefully collected and divided into training and test sets. Then, the most critical parameters of the GBM model are determined using 10-fold cross-validation integrated with the WOA. The results obtained from the WOA-GBM model are compared with those from adaptive boosting (AdaBoost) and extreme gradient boosting (XGBoost) models. The results show that the WOA-GBM model outperforms other models. Additionally, SHapley Additive exPlanation (SHAP) method is used to explain the prediction of the proposed WOA-GBM model globally and locally. Finally, an efficient graphical user interface (GUI) tool and a web application (WA) are developed to apply the proposed WOA-GBM model for practical use.

1. Introduction

Patch loading is a concentrated transverse force acting perpendicular to the web of the beams or girders. This loading type needs to be considered in designing the SPGs because it usually causes a local failure in the vicinity of the loading area of the girder web [1]. Therefore, the SPGs are often designed with transverse or longitudinal stiffeners to increase the PLR. In the location where the patch loading is fixed, the transverse stiffeners on the web can improve the resistance of SPGs. However, in the case of moving loads, the exact position of the patch loading is not known in advance. Still, it is not possible to provide stiffeners at all critical locations. Hence, the longitudinal stiffeners are necessary to enhance the loading capacity and increase the resistance to shear and bending of the girders subjected to patch loading.

In the last decades, researchers have carried out several tests to investigate the longitudinal stiffeners' behaviors on the PLR of the SPGs. The experimental results [1–7] showed that the PLR of SPGs significantly increased by using the longitudinal stiffeners. Based on the experimental results, the PLR formulas have been empirically derived for practical design by design codes [8,9] and some researchers [3,10–12]. In this regard, the PLR of the unstiffened girder is multiple with a correction factor for the girder with a longitudinal stiffener using the traditional regression methods. However, most of these empirical formulas are developed based on a limited number of experimental results. Therefore, they lead to insufficient prediction capability for PLR of longitudinally stiffened SPGs.

Over the last decade, machine learning (ML) has been considered a powerful method for predicting and assessing structural performances [14]. The ML method can be used for several objectives, e.g., regression [15–24], classification [25–27], optimization [28,29], etc. Regarding the SPGs under patch loading, Fonseca et al. [30,31] presented a parametric study in which they used a neural network system to forecast the PLR of the unstiffened SPGs. In addition, Fonseca et al. [32] used a neuro-fuzzy system to predict and classify the behavior of the unstiffened SPGs subjected to patch loading. Cevik [12] and Cevik et al. [33] developed new formulas using genetic programming and stepwise regression methods to predict the PLR of longitudinally stiffened SPGs. Recently, Kurtoglu [34] developed a support vector machines (SVM) model for predicting the PLR of longitudinally stiffened SPGs. Truong et al. [35] developed an extreme gradient boosting algorithm (XGBoost) model for PLR of longitudinally stiffened SPGs. The results obtained from these studies showed that the ML models were better than existing design codes and empirical formulas.

Although the studies mentioned above used ML models for PLR of the unstiffened and longitudinally stiffened SPGs, no study has explored the impact of features or input variables on the results predicted by ML models. From the point of application perspective, it is critical to have models that elucidate the input–output relationships to inform decision strategies and properly deploy a model [36]. The feature importance approach can be used for explaining ML models' outputs.

^{*} Corresponding author.

E-mail addresses: vietlinh.dhv@gmail.com (V.-L. Tran), duan468@gmail.com (D.-D. Nguyen).

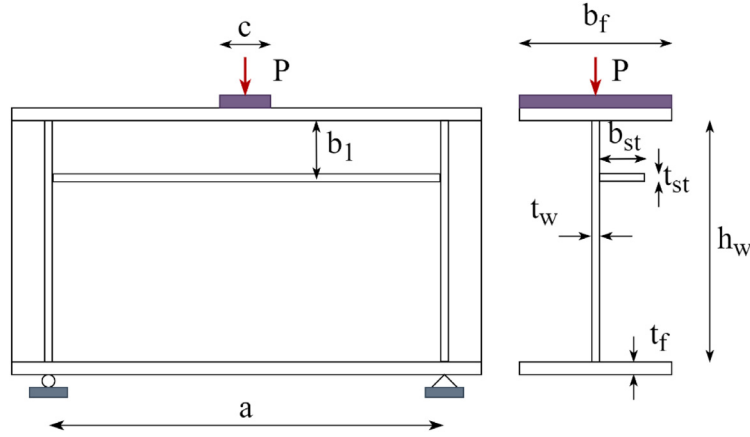


Fig. 1. Typical longitudinally stiffened SPGs under patch loading.

However, it does not describe each prediction score locally and how a feature relates to the final prediction. In other words, it cannot show the positive or negative impact on the output. Additionally, several studies showed that hyperparameters could significantly affect the final performance of ML models [37–42]. A major disadvantage of using the default parameters of ML models is that they introduce a bias and variance in the model that causes overfitting or underfitting. It eventually leads to poor generalizability and inaccurate predictions on new data samples.

This paper develops a novel hybrid WOA-GBM model using the whale optimization algorithm (WOA) and the gradient boosting machine (GBM) for predicting the PLR of longitudinally stiffened SPGs. The results obtained from the WOA-GBM model are compared with those of the adaptive boosting (AdaBoost) and extreme gradient boosting (XGBoost) models. Accuracy criteria, including the determination coefficient (R^2), the root-mean-square error (RMSE), and the mean absolute error (MAE), are used for the assessment of the models. Moreover, the importance and contribution of the factors that influence the PLR of longitudinally stiffened SPGs are investigated using the SHapley Additive exPlanation (SHAP) method. Finally, a graphical user interface (GUI) and a web application (WA) are developed based on the best model to provide a convenient tool for the preliminary estimation of the PLR of longitudinally stiffened SPGs.

2. Data collection

In this study, the authors compile a database of 137 experimental results of simply supported SPGs from many sources [1–3,5,7,32,43–48] to develop the ML models. In this database, several tests have been conducted recently by Markovic and Kovacevic [1], Rogač et al. [44], and Kovacevic and Markovic [7]. It is noted that samples with the same input values but different outputs are considered to keep one. The configuration of longitudinally stiffened SPGs subjected to patch loading is shown in Fig. 1.

In this figure, a is the web panel length, h_w is the web height, t_w is the web thickness, b_f is the flange width, t_f is the flange thickness, c is the applied load length, b_l is the distance between loaded flange and longitudinal stiffener, b_{st} is the width of the stiffener, t_{st} is the thickness of stiffener, and P is patch loading. In addition, the web yield strength (f_{yw}) and the flange yield strength (f_{yf}) are composed in this database.

All of the specimens in the database include only one longitudinal stiffener in the SPGs. The statistical properties of the final database are provided in Table 1. Fig. 2 shows the density distribution of the input and output parameters of the database. In addition, the Pearson correlation coefficient between any two variables was calculated, as shown in Fig. 3.

3. Overview of machine learning methods and optimization algorithm

3.1. Adaptive boosting (AdaBoost)

AdaBoost was firstly introduced by Freund and Schapire [49]. It is an ensemble ML model, where a base learner is trained using the training sets to update the weights according to the performance of the previous iterations. The variants of the AdaBoost method are determined by how these weights are updated and which instances are affected. A brief description of the AdaBoost algorithm is as follows [50]:

AdaBoost algorithm

```

1: function AdaBoost ( $D = \{(x_i, y_i)\}_{i=1}^N$  – training set,
    $nEstimators$  – number of learners)
2:   initialize:  $T[1, \dots, nEstimators]$ ,  $w_i = 1/N$ 
3:   for  $t$  in  $1, \dots, nEstimators$  do
4:      $T_t = Fit\_Estimator(D, w)$ 
5:      $\varepsilon_t = \sum_{x_i \in D | T_t(x_i) \neq y_i} w_i / \sum_{x_i} w_i$ 
6:      $\alpha_t = \log \frac{1-\varepsilon_t}{\varepsilon_t}$ 
7:     for  $x_i$  in  $D$  do
8:       if  $T_t(x_i) = y_i$  then
9:          $w_i = w_i \cdot \frac{\varepsilon_t}{1-\varepsilon_t}$ 
10:      end if
11:    end for
12:     $Normalize(w)$ 
13:  end for
14:  output:  $C(x) = \arg \max_l \sum_{t \in [1, nEstimators]} |T_t(x) = l| \alpha_t$ 
15: end function

```

where $D = x_i, y_{i=1}^N$ is a training set, x_i are the input variables and y_i the target variable or outputs, to predict future values y by providing the inputs x . N stands for the number of instances of the dataset. T is the set of base learners of the ensemble. t marks each of the iterations. T_t means the estimator of iteration t . ε denotes the error made by previous estimators. w are the weights given to training instances by boosting methods.

3.2. Gradient boosting machine (GBM)

GBM is another prevalent ensemble method proposed by Friedman [51] that combines multiple decision trees to create a more robust model. In contrast to AdaBoost, the weights of the training sets are not adjusted, but each predictor is trained using the residual errors of its predecessor as labels. In regression, the GBM starts by initializing the model by a first guess. Then, at each step, a new decision tree is fitted

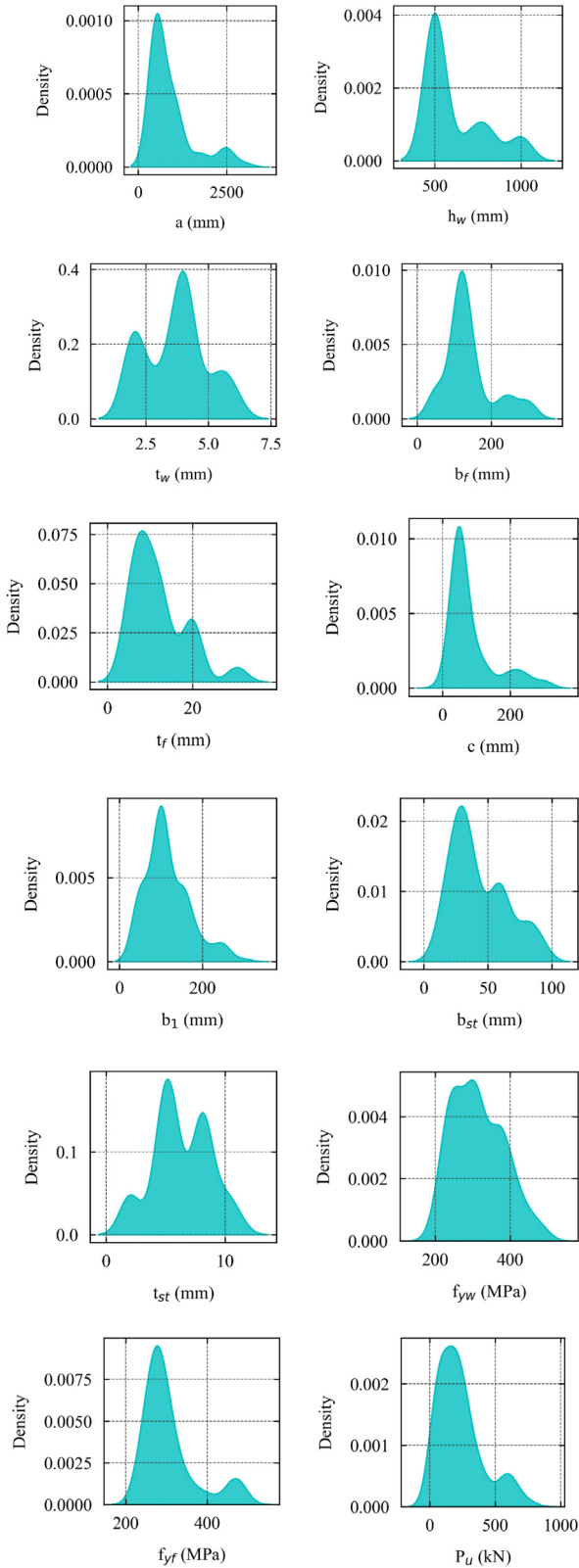


Fig. 2. Density distribution of the input and output parameters.

to the current residual and added to the previous model to update the residual. This process is so-called stage-wise, meaning that at each new step, the decision trees added to the model in previous steps are not modified. The basic advantage of GBM is that it prevents overfitting and

uses lesser computational resources through objective function [52]. A simplified illustration of the GBM algorithm is provided by the following pseudo-code [53]:

GBM algorithm

```

1: function GBM ( $D = \{(x_i, y_i)\}_{i=1}^N$  – training set,  $nEstimators$  –
   number of learners,  $\mathcal{L}(y, F(x))$  – loss function,  $v$  – learning rate)
2:   initialize:  $F_0(x) = \operatorname{argmin}_\gamma \sum_{i=1}^N \mathcal{L}(y_i, \gamma)$ 
3:   for  $t$  in  $[1, nEstimators]$  do
4:     // Compute pseudo-residual  $r_{i,m}$  for each sample  $x_i$ 
5:     for  $i$  in  $[1, N]$  do
6:        $r_i = -\left[\frac{\partial \mathcal{L}(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F(x)_{t-1}}$ 
7:     end for
8:      $T_t = \operatorname{Fit\_Regression\_Tree}(D, r)$ 
9:     for  $i$  in  $[1, J]$  do
10:       $R_j = \operatorname{Name\_Leaves}(T_t)$ 
11:      // Compute output  $\gamma_j$  of each leaf  $j$ 
12:       $\gamma_j = \operatorname{argmin}_\gamma \sum_{x_i \in R_j} \mathcal{L}(y_i, F_{t-1}(x_i) + \gamma)$ 
13:    end for
14:     $F_t(x) = F_{t-1}(x) + v \sum_{j=1}^J \gamma_j I(x \in R_j)$ 
15:  end for
16:  output:  $F(x)$ 
17: end function

```

3.3. Extreme gradient boosting (XGBoost)

XGBoost [54] is one of the most reputed methods of the past few years. XGBoost is designed as a highly scalable and accurate tree boosting system. The XGBoost model is an improvement of the GBM method. It uses many additive functions to predict the result as

$$\bar{y}_i = y_i^0 + \eta \sum_{k=1}^M f_k(X_i) \quad (1)$$

where \bar{y}_i is the predicted result for the i th sample of which the vector of the features is X_i ; M is the number of estimators and each estimator f_k (with k from 1 to M) corresponds to an independent tree structure; y_i^0 is the initial guess that is the mean of the measured values in the training set; η is the learning rate (shrinkage parameter) that helps to smoothly improve the model while adding a new tree and avoid over-fitting.

The training process is realized in an additive manner. In Eq. (1), at k th step, a k th estimator is added to the model and the k th predicted result \bar{y}_i^k is calculated from the predicted value at the previous step $\bar{y}_i^{(k-1)}$ and the estimation f_k of the additional k th estimator as

$$\bar{y}_i^k = \bar{y}_i^{(k-1)} + \eta f_k \quad (2)$$

where f_k is defined by the leaves weights that are found by minimizing the objective function of the k th tree that is defined by

$$obj = \gamma T + \sum_{j=1}^T \left[G_j \omega_j + \frac{1}{2} (H_j + \lambda) \omega_j^2 \right] \quad (3)$$

where T is the number of k th tree leaves and ω_j with j from 1 to T are the leaves weights; λ and γ are the regularization parameters that control the simplicity of the tree structure to avoid overfitting. The parameters G_j and H_j are the sums of overall samples associated with the j th leaf of the first and second gradients of the loss function, respectively.

The k th tree is constructed by splitting the leaves starting from a single leaf. Such procedure is realized by maximizing the gain parameter that is defined by

$$gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (4)$$

where G_L and H_L are associated with the left leaf and G_R and H_R are associated with the right leaf after the splitting.

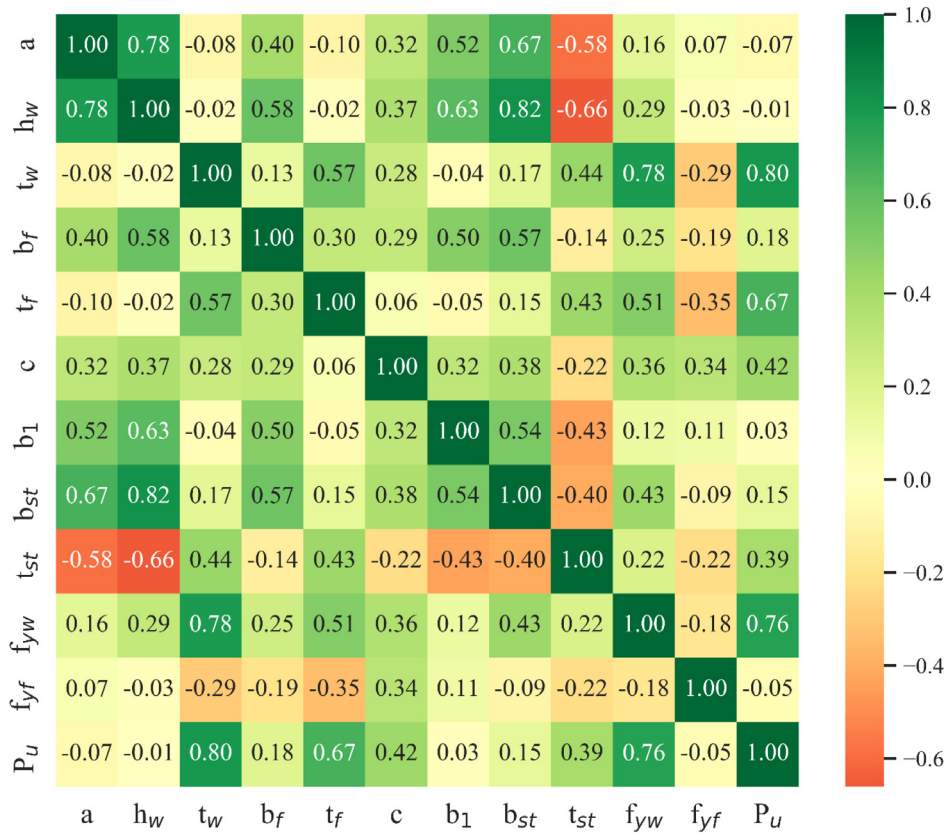


Fig. 3. Pearson correlation coefficient between any two variables.

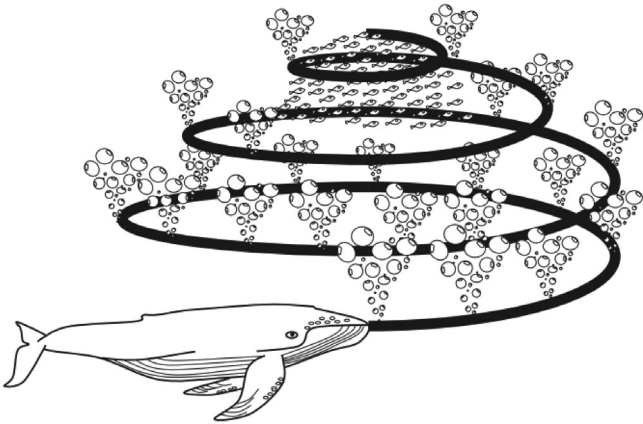


Fig. 4. Bubble-net feeding behavior of humpback whales [13].

3.4. Whale optimization algorithm (WOA)

WOA is a novel nature-inspired meta-heuristic optimization algorithm, which was proposed by Mirjalili and Lewis [13]. In the WOA, the optimal solution is found out using the hunting strategy of the humpback whales. The hunting process of the humpback whales involves two stages: surrounding the target, subsequently producing a net of bubbles to confine the target, as shown in Fig. 4. The mathematical model of the bubble net system in humpback whales can be explained as the following.

Step 1: Bubble net attacking method (exploitation phase).

1. **Encircling prey.** Initially, the WOA presupposes the current leading candidate as the current best solution. Next, other

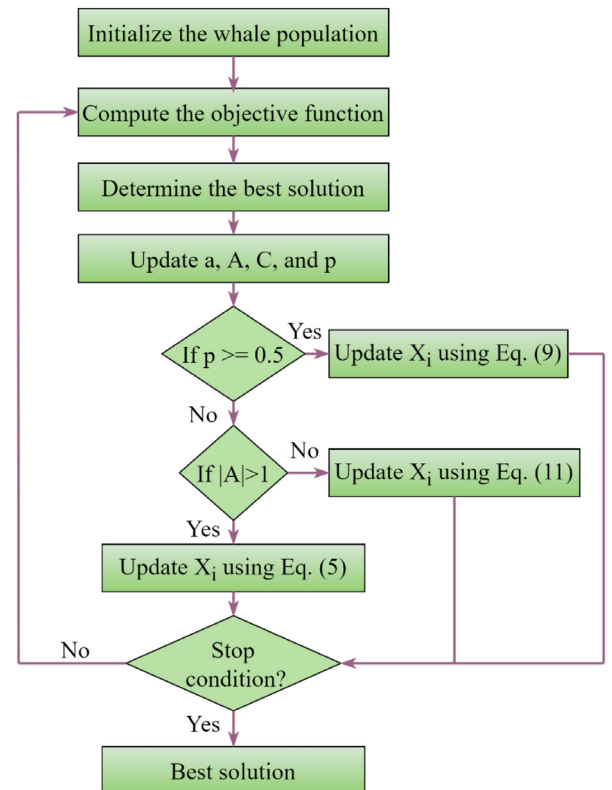


Fig. 5. Flowchart of the WOA.



Fig. 6. 10-fold cross-validation.

Table 1
Statistical properties of experimental data.

	a (mm)	h_w (mm)	t_w (mm)	b_f (mm)	t_f (mm)	c (mm)	b_1 (mm)	b_{st} (mm)	t_{st} (mm)	f_{yw} (MPa)	f_{yf} (MPa)	P_u (kN)
Min.	500.00	500.00	2.00	50.00	5.00	0.00	50.00	12.00	2.00	204.00	239.00	34.00
Mean	903.75	614.52	3.69	139.59	11.97	83.10	116.39	42.91	6.34	314.26	306.09	230.85
Max.	3000.00	1000.00	6.00	300.50	30.60	300.00	300.00	90.00	11.00	483.00	485.00	777.90
StD	627.53	172.87	1.2	63.81	6.31	68	53.49	21.4	2.33	67.79	64.81	172.6
CoV	0.6944	0.2813	0.3	0.457	0.53	0.818	0.46	0.5	0.37	0.216	0.212	0.748

searching agents are utilized to alter their positions to arrive at the current best agent's location. The mathematical model of this strategy is manifested in Eqs. (5) and (6).

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad (5)$$

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \quad (6)$$

In the equations, $\vec{X}^*(t)$ is the whale's previous best location at step t . $\vec{X}(t+1)$ denotes the current position of a whale, \vec{D} is the distance from the whale point to the target, and the $||$ indicates the absolute value. \vec{A} and \vec{C} serve as coefficient vectors, and their computations are given below:

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \quad (7)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (8)$$

where r is a random vector produced with steady diffusion in the interval of $[0, 1]$ and a declines from two to zero by order of iterations.

2. **Spiral updating position.** The distance between the whale located at (X, Y) and the prey located at (X^*, Y^*) is represented, then a helix-shaped movement is implemented by using the spiral equation as shown in Eq. (9).

$$\vec{X}(t+1) = e^{bl \cdot \cos(2\pi l)} \cdot \vec{D}^* + \vec{X}^*(t) \quad (9)$$

where $\vec{D}^* = |\vec{X}^*(t) - \vec{X}(t)|$, b denotes the constant used to determine a logarithmic spiral shape and l denotes a random number in the range $[-1, 1]$.

When updating whale's positions, WOA assumes a likelihood of 50% to choose among shrinking encircling mechanism and the

spiral model as follows:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^* - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ e^{bl \cdot \cos(2\pi l)} \cdot \vec{D}^* + \vec{X}^*(t) & \text{if } p \geq 0.5 \end{cases} \quad (10)$$

where p represents a random quantity within $(0, 1)$.

Step 2: Search for prey (exploration).

In this stage, the WOA utilizes a random search to explore the target. The A vector is supported with a random value not equal to 1. The agents' position is rearranged based on randomly choosing the search agent instead of looking for the best agent. The exploration strategy is beneficial for WOA to address the issue of local optimization. The exploration position update is expressed in Eqs. (11) and (12).

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad (11)$$

$$\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}| \quad (12)$$

where \vec{X}_{rand} represents a random location based on the current population. The flowchart of the WOA is shown in Fig. 5.

4. Evaluation metrics

To evaluate the ML models, three performance metrics including the correlation coefficient (R^2), the root mean square error ($RMSE$), and the mean absolute error (MAE) are used. They are defined as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^N (t_i - o_i)^2}{\sum_{i=1}^N (t_i - \bar{t})^2} \quad (13)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (t_i - o_i)^2} \quad (14)$$

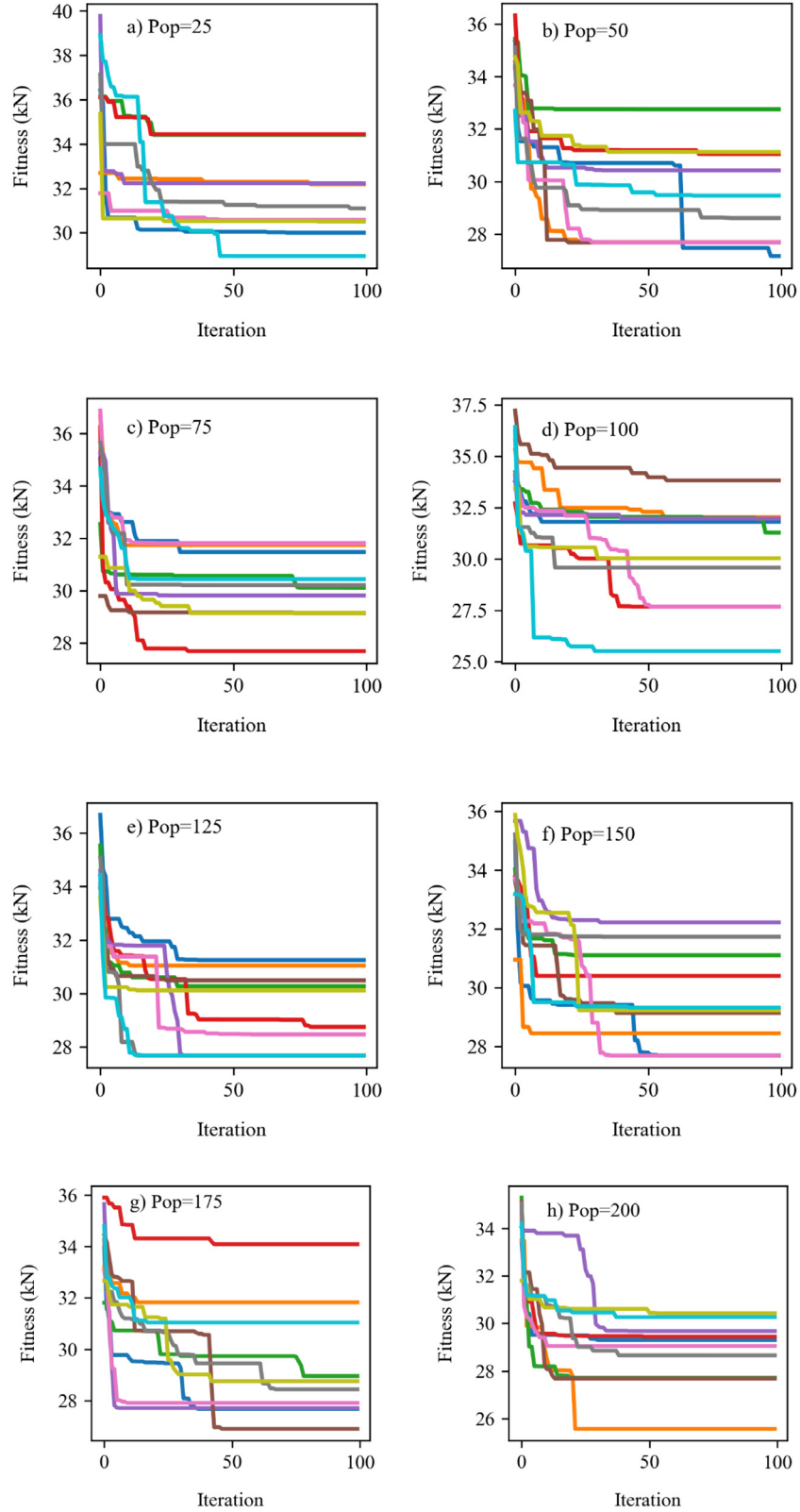


Fig. 7. Convergence curve of WOA-GBM model with different population sizes.

$$MAE = \frac{\sum_{i=1}^N |t_i - o_i|}{N} \quad (15)$$

where N is the number of samples, $[t_1, \dots, t_N]^T$ and $[o_1, \dots, o_N]^T$ are the actual and the predicted vector values, respectively. \bar{t} is the average of actual values.

5. Development of the WOA-GBM model

In this study, the web panel length (a), the web height (h_w), the web thickness (t_w), the flange width (b_f), the flange thickness (t_f), the applied load length (c), the distance between loaded flange and

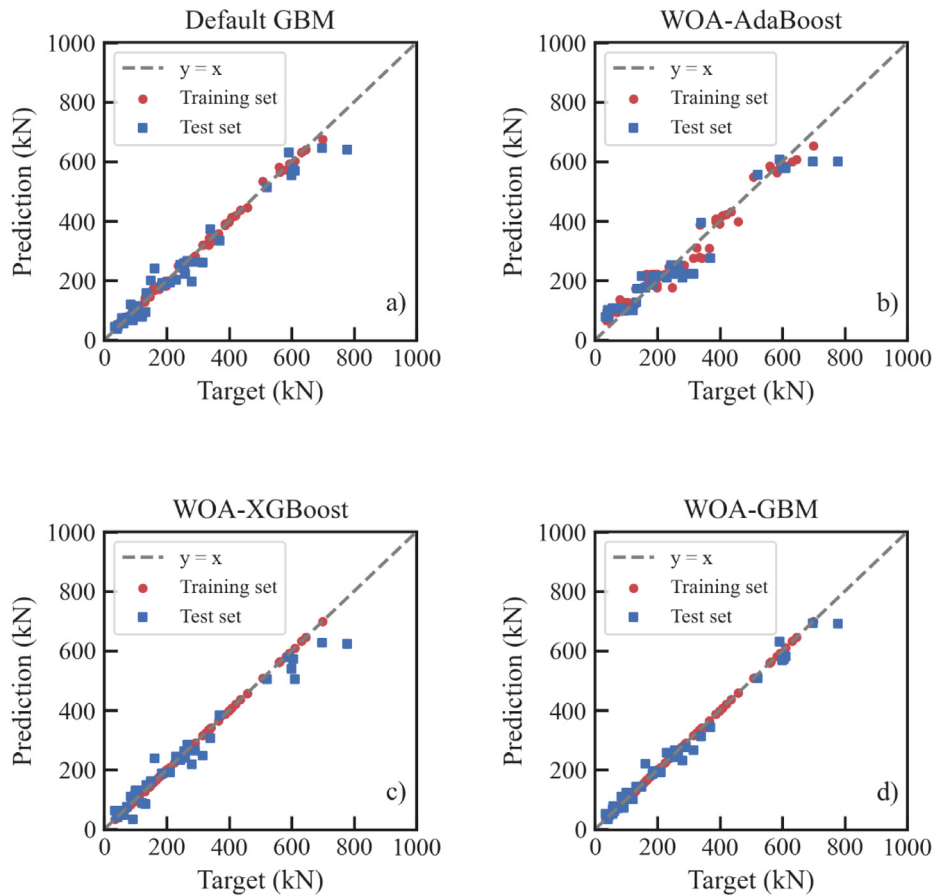


Fig. 8. Comparison of different ML models.

longitudinal stiffener (b_1), the width of the stiffener (b_{st}), the thickness of stiffener (t_{st}), the web yield strength (f_{yw}), and the flange yield strength (f_{yf}) are chosen to establish the ML models for estimating the PLR of longitudinally stiffened SPGs (P_u).

Overfitting is one of the biggest concerns in any ML method because it is usually the downside of an unnecessarily over-complex model. Commonly, the dataset was split into training and testing sets. However, it is not often possible to hold out enough data points to accurately estimate the predictive performance of the models without affecting the estimation quality. Cross-validation (CV) is the most traditional method to overcome a scarcity of data. The dataset is randomly divided into k subsamples, called folds, of roughly equal size in this method. Therefore, the first model is estimated using $k - 1$ folds as a training dataset, and the remaining fold is used to calculate the prediction accuracy metric. This procedure is reiterated k times, and at each time, a different fold is used as a test set. The accuracy is then expressed as an average accuracy acquired by the k models in k validation rounds. Although there is no formal rule for choosing the value of k , in practice, $k = 5$ or $k = 10$ is widely used by many researchers. In this study, 10-fold is used to fine-tune the optimal parameters for the ML models, as shown in Fig. 6.

To consider the effect of training and test partitions, five cases in which the database is divided by 0.9-0.1, 0.85-0.15, 0.8-0.2, 0.75-0.25, and 0.7-0.3 as training and test data, are investigated. The training set is used to find the optimal parameters using a 10-fold CV integrated with the WOA, while the performance of the models is then evaluated using the test set. The WOA is applied to train all models by a hyper-parameter subset manually specified and it must be guided by some performance metric. For all the hyper-parameter combinations, the training process was repeated 10 times, and the average $RMSE$ is used as the objective function for the optimization process. The crucial

Table 2

Crucial parameters and their ranges of the GBM model.

Model	Parameter and range				
GBM	learning_rate (0.01–1.0)	n_estimators (5–100)	Subsample (0.1–1.0)	max_depth (1–10)	Alpha (0.1–0.9)

Table 3

Performance of the WOA-GBM model with different training–test data ratios.

Ratio	Pop size	Training data			Test data		
		R^2	$RMSE$	MAE	R^2	$RMSE$	MAE
0.9-0.1	125	0.988	0.086	0.062	0.982	28.624	19.340
0.85-0.15	200	0.985	0.094	0.086	0.969	31.985	25.812
0.8-0.2	150	0.987	0.090	0.072	0.978	29.915	21.825
0.75-0.25	75	0.982	0.140	0.096	0.956	33.517	28.280
0.7-0.3	100	1.000	0.049	0.039	0.984	25.467	18.583

Unit of $RMSE$ and MAE : kN.

parameters and their ranges of the GBM model used to optimize are listed in Table 2.

After many tests, it is found that after 100 iterations, the fitness value seems to be stable. Increasing the number of iterations, the calculation time will increase and the result is not improved. Therefore, the number of iterations is set as equal to 100 in this study. Moreover, several population sizes of 25, 50, 75, 100, 125, 150, 175, and 200 are selected for the optimization process to choose the best one. Each population size is run 10 times, and the best result is chosen. Table 3 shows the best model performance of five training–test ratios.

The results show that five cases perform well in training and test data. However, the 0.7-0.3 case results provide the best prediction with the highest value of R^2 and lowest values of $RMSE$ and MAE for

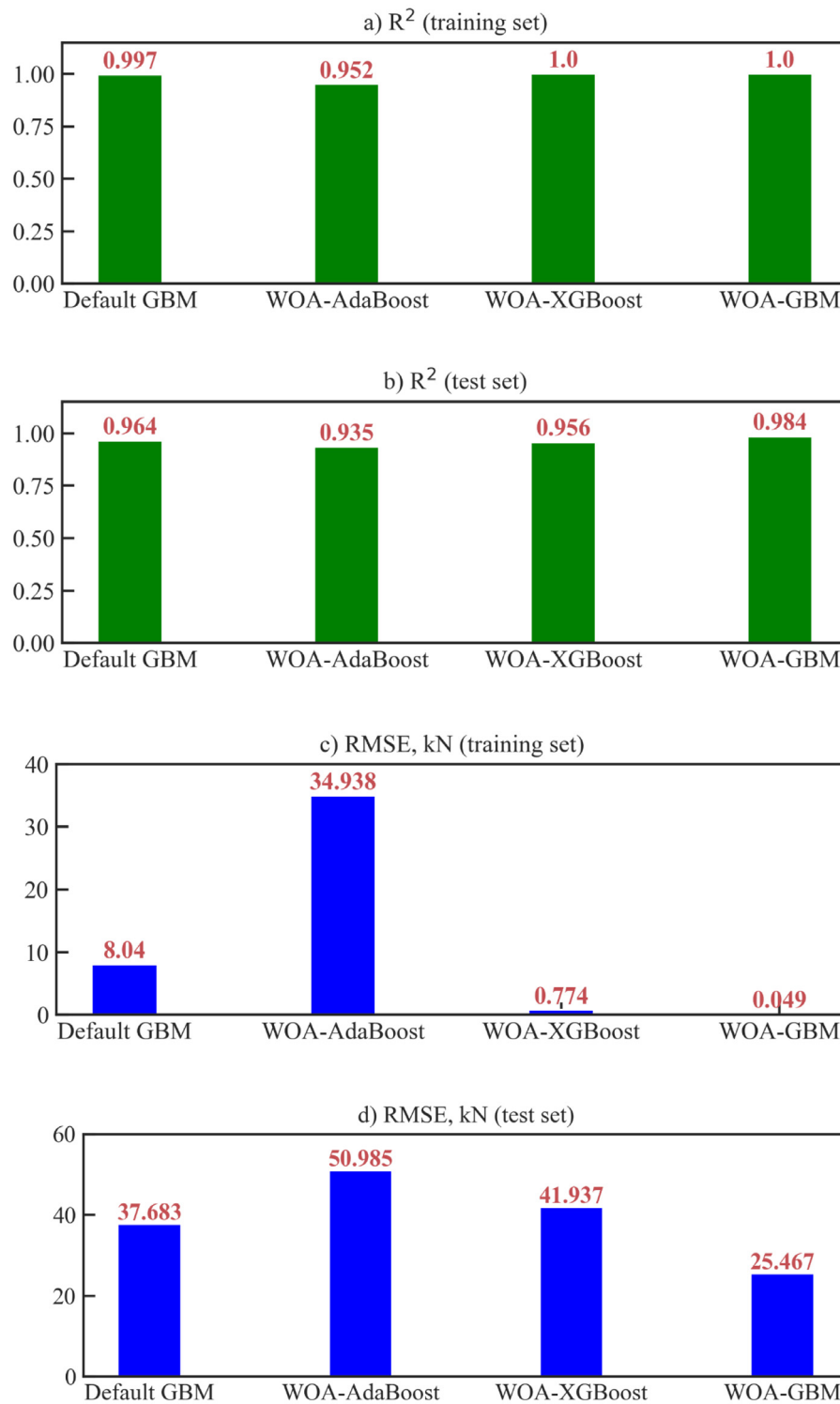


Fig. 9. Performance of different ML models.

both training and test data. The R^2 , $RMSE$, and MAE values for the training data are 1.0, 0.049 kN, and 0.039 kN, respectively. For the test data, the R^2 , $RMSE$, and MAE values are 0.984, 25.467 kN, and 18.583 kN, respectively. Therefore, the database is randomly divided into a training set (70%) and a test set (30%). Fig. 7 shows the convergence curves of the WOA-GBM models for each population size. Based on these results, a population size of 100 is chosen for the WOA-GBM model. The optimal parameters of the WOA-GBM model are presented in Table 4.

Table 4
Optimal parameters of the WOA-GBM model.

Model	Optimal parameters				
WOA-GBM	learning_rate	n_estimators	Subsample	max_depth	Alpha
	0.79753	92	0.83729	3	0.32957

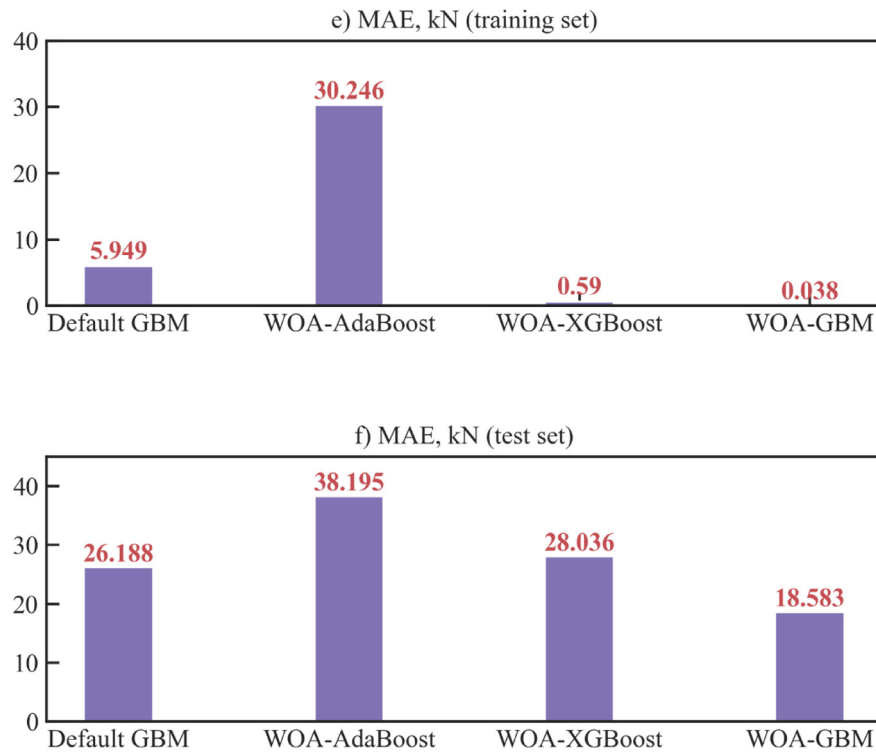


Fig. 9. (continued).

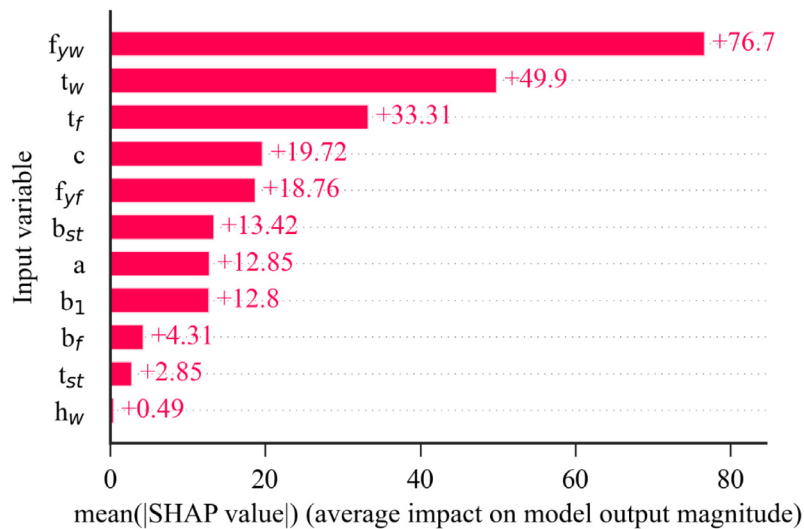


Fig. 10. Global importance factors based on the SHAP.

6. Results and discussions

6.1. The performance of ML models

To evaluate the performance of the WOA-GBM model, the results obtained from the default GBM, WOA-AdaBoost, and WOA-XGBoost models are also used for comparison. Figs. 8–9 show the performance of these ML models. It can be seen that four models show good agreement with the experimental results as R^2 values were larger than 0.90 for both training and test sets. In the training phase, the prediction performance in terms of R^2 can be ranked as WOA-GBM (1.0) = WOA-XGBoost (1.0) > GBM (0.997) > WOA-AdaBoost (0.952). Similarly, the ranking for $RMSE$ is WOA-GBM (0.049 kN) > WOA-XGBoost (0.744 kN) > GBM (8.04 kN) > WOA-AdaBoost (34.938 kN) and for MAE

is WOA-GBM (0.038 kN) > WOA-XGBoost (0.59 kN) > GBM (5.949 kN) > WOA-AdaBoost (30.246 kN). In the test phase, the prediction performance in terms of R^2 is WOA-GBM (0.984) > GBM (0.964) > WOA-XGBoost (0.956) > WOA-AdaBoost (0.935). The ranking for $RMSE$ is WOA-GBM (25.467 kN) > GBM (37.683 kN) > WOA-XGBoost (41.937 kN) > WOA-AdaBoost (50.985 kN) and for MAE is WOA-GBM (18.583 kN) > GBM (26.188 kN) > WOA-XGBoost (28.036 kN) > WOA-AdaBoost (38.195 kN). Obviously, the WOA-GBM model shows the best performance with the highest value of R^2 , lowest value of $RMSE$ and MAE .

Moreover, Table 5 shows the statistical results of the predictions to targets ratio based on the four ML models for the entire data. It is noted that among four ML models, the WOA-GBM model has superior performance compared to the other models in terms of the lowest standard deviation ($std = 0.090$) and covariance ($cov = 0.089$), and

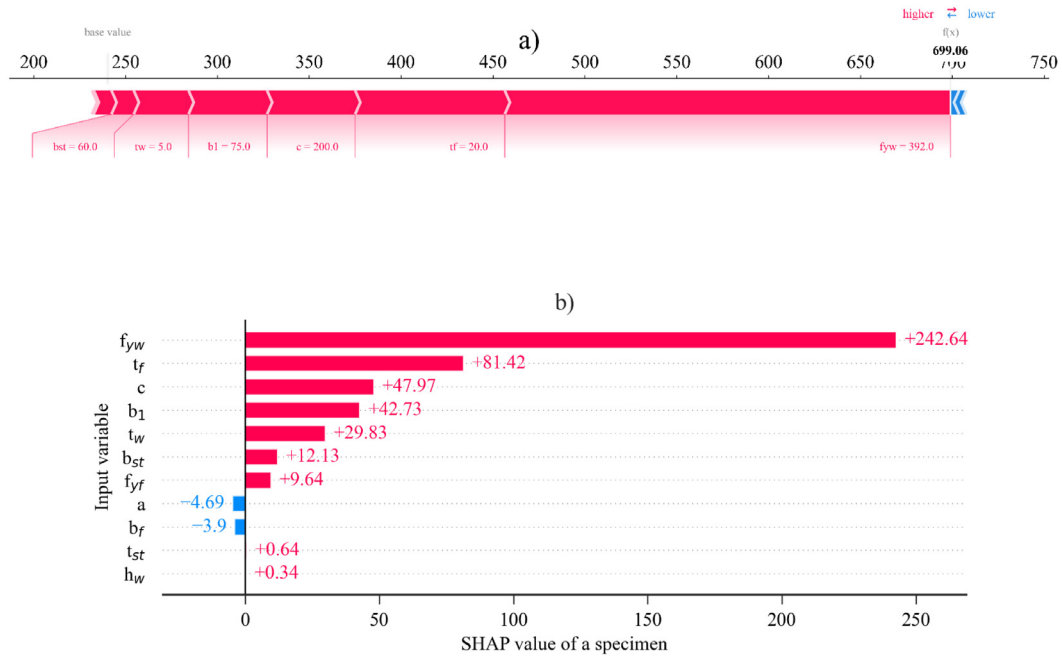


Fig. 11. Explanation of the patch loading prediction of a specimen. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

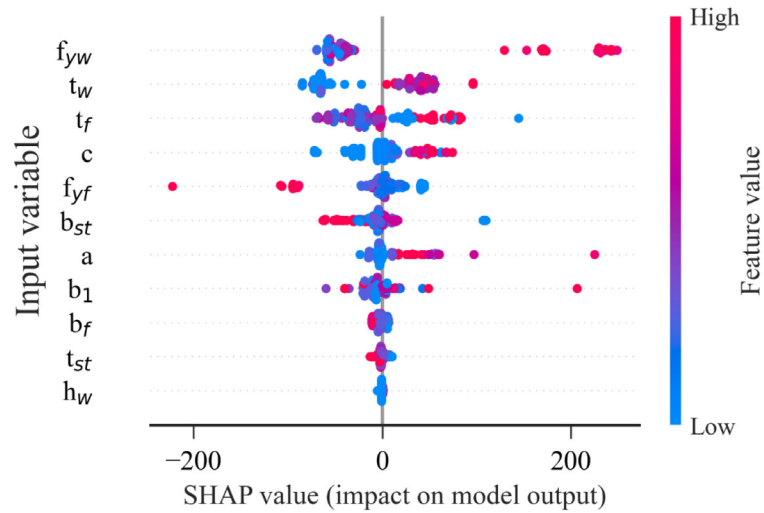


Fig. 12. Summary plots for the PLR. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5

Statistical results of the predictions to targets ratio based on the six ML models.

Model	WOA-AdaBoost	GBM	WOA-GBM	WOA-XGBoost
Mean	1.205	1.010	1.006	1.008
Std	0.407	0.111	0.090	0.092
Cov	0.338	0.110	0.089	0.091

the mean value is close to 1. It means that the proposed WOA-GBM model provides the most stable, safe, and accurate predictions. Thus, only the WOA-GBM model is investigated in subsequent parts of this study.

6.2. Model explanation-based SHapley Additive exPlanation (SHAP)

In this study, the Shapley Additive Explanations (SHAP) method proposed by Lundeborg and Lee [55] is utilized to assess the importance

and contributions of each input variable for predicting the PLR of longitudinal stiffened steel girders and for interpreting the WOA-GBM model globally and locally.

The idea behind Shapley values comes from the game theory [36]. The SHAP method [55] assigns each input feature an important value for a particular prediction. It is an additive feature attribution method that defines the output of a model as the sum of the real values attributed to each input feature. Additive feature attribution methods have an explanation model defined as a linear function of binary features as in the following equation:

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i \quad (16)$$

where g is the explanation model, $z' \in \{0, 1\}^M$ is the coalition vector, M is the number of input features, ϕ_0 represents a constant value when all inputs are missing, and $\phi_i \in \mathbb{R}$ is the feature attribution values.

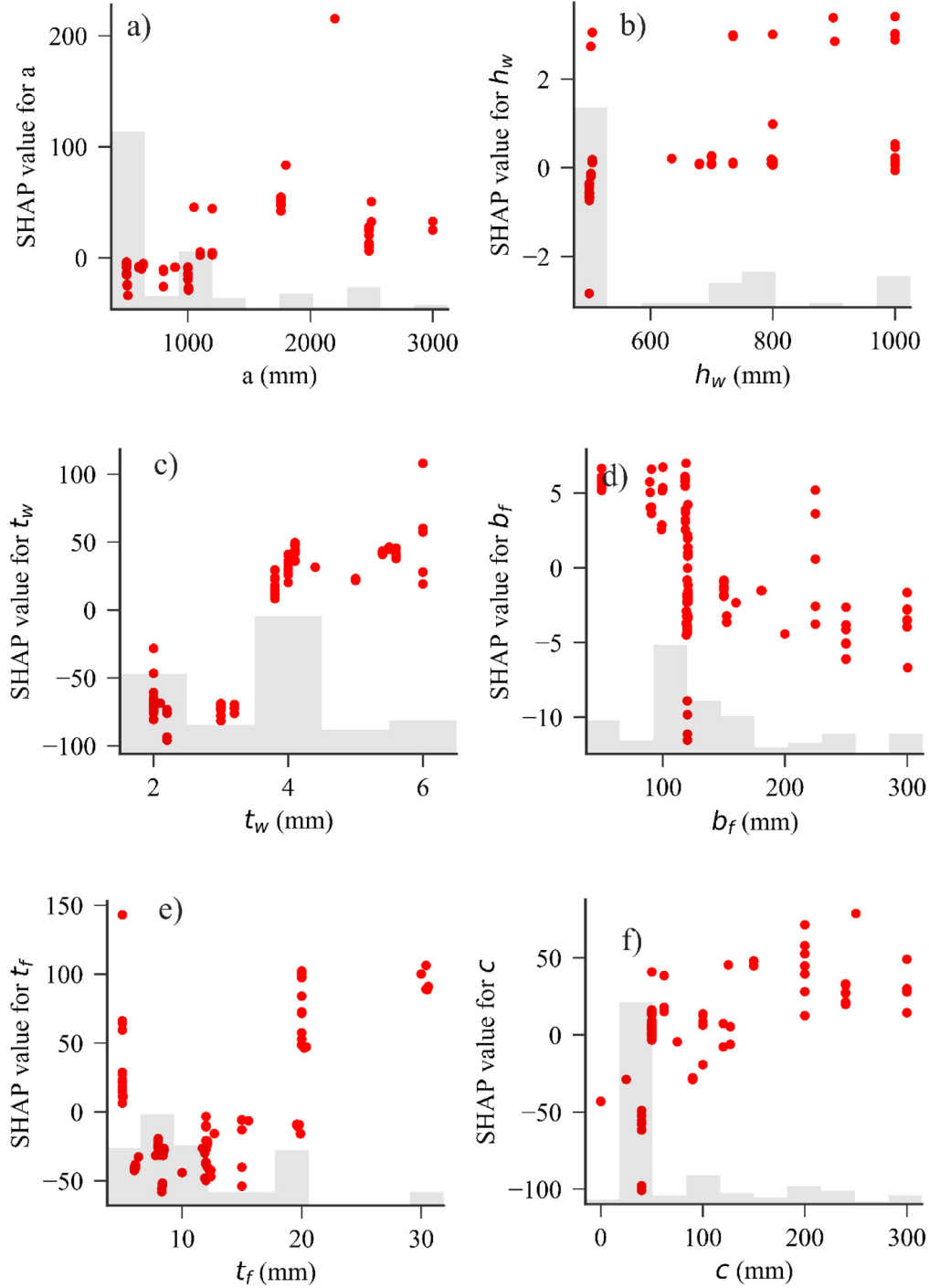


Fig. 13. Shalely value of each input variable.

The Shapley value of the feature i is computed as

$$\phi_i = \sum_{S \subseteq F \setminus i} \frac{|S|! (|F| - |S| - 1)!}{|F|!} [f_{S \cup i}(x_{S \cup i}) - f_S(x_S)] \quad (17)$$

where F is the set of all features, x_S represents the values of the input features in the set S .

In this section, the Shapley values are directly used to explain the PLR of the WOA-GBM model.

Fig. 10 shows the feature importance of the input variables on the PLR prediction using the SHAP method. These important factors are the averages of the absolute Shapley values per feature across the data. It can be seen that, in general, the web yield strength (f_{yw}) and the web

thickness (t_w) have a significant influence on the PLR of longitudinally stiffened SPGs. The flange thickness (t_f), the applied load length (c), and the flange yield strength (f_{yf}) also affect the PLR of longitudinally stiffened SPGs, but the importance is much less than the f_{yw} and t_w and slightly higher than the distance between loaded flange and longitudinal stiffener (b_1), the web panel length (a), and the width of the stiffener (b_{st}). While the remaining variables (b_f , t_{st} , h_w) have relatively low impacts on the results.

Fig. 11 shows a typical prediction plot obtained with the WOA-GBM model for a specimen with $a = 1000$ mm, $h_w = 700$ mm, $t_w = 5$ mm, $b_f = 225$ mm, $t_f = 20$ mm, $c = 200$ mm, $b_1 = 75$ mm, $b_{st} = 60$ mm, $t_{st} = 5$ mm, $f_{yw} = 392$ MPa, and $f_{yf} = 355$ MPa. In this figure,

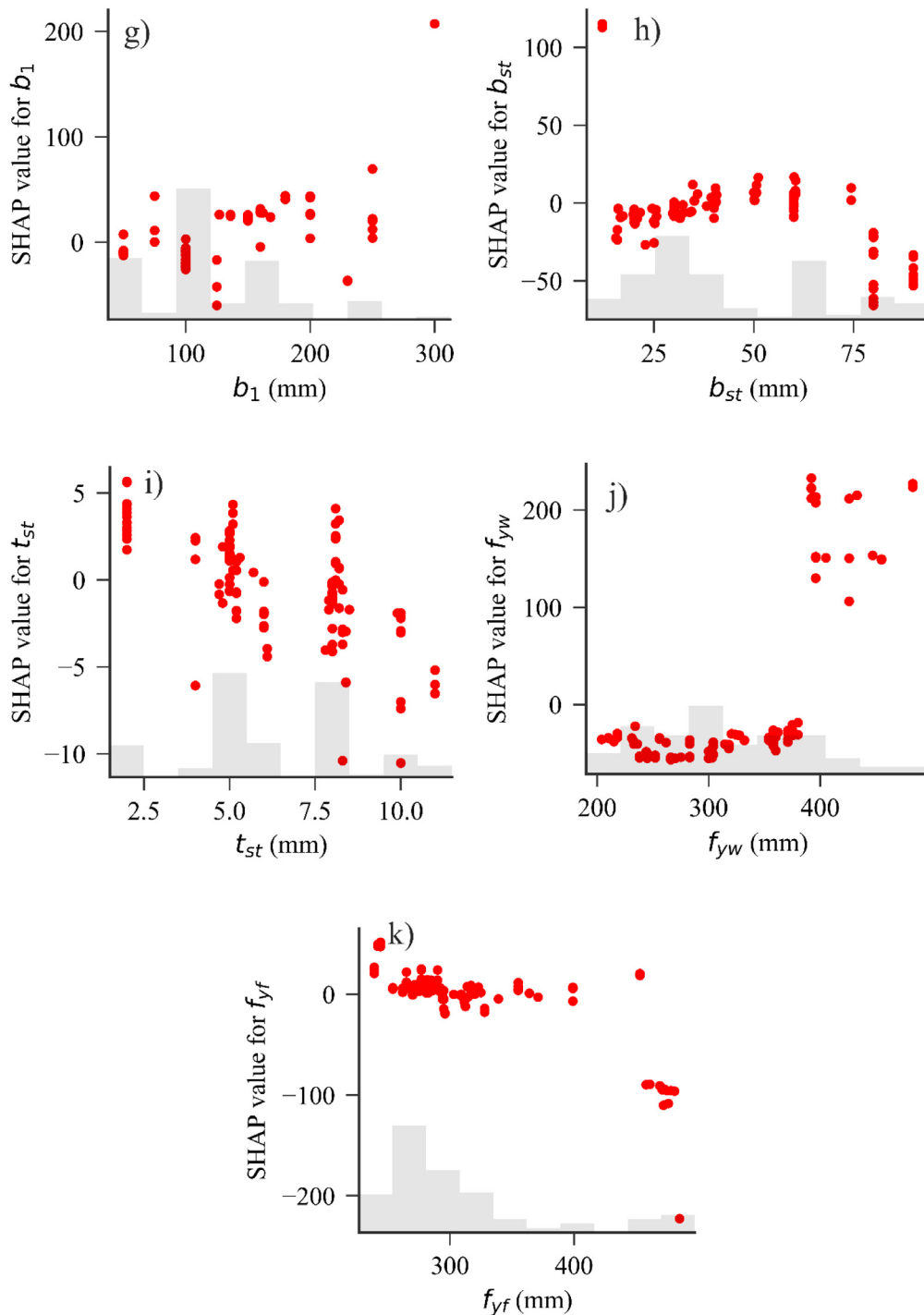


Fig. 13. (continued).

the base value (230.85 kN) is the average of the experimental results. As can be seen that the PLR of the specimen is higher than the base value. Input variables pushing the prediction higher are shown in red, those pushing the prediction lower are in blue. The width of the bar denotes the extent of the corresponding increases and decreases. The most important variables that contribute to this prediction are f_{yw} , t_f , c , b_1 , t_w , and b_{st} .

Fig. 12 shows the SHAP summary plot that illustrates the effect of each parameter (positively or negatively) on prediction for all experimental data. In the figure, the x-axis represents the specific Shapley value and the y-axis the input variables, ordered by importance. Each

point represents a Shapley value of an input variable for each specimen, and the y-axis (from top to bottom) indicates the input variables in the order of importance. Red and blue colors represent the high and low variable values, respectively. The red point on the right-hand side of the plot indicates a positive correlation with the PLR, whilst the red point on the left-hand side of the plot indicates a negative correlation. Generally, it can be seen that with the increase in the web yield strength (f_{yw}), the web thickness (t_w), the flange thickness (t_f), the applied load length (c), and the web panel length (a), the PLR will increase. By contrast, the PLR tends to decrease with increasing values for features like the flange yield strength (f_{yf}) and the width of the stiffener (b_{st}).

GUI Tool for Predicting Patch Loading Resistance

Developed by Dr. Viet-Linh Tran
Department of Civil Engineering, Vinh University, Vietnam
Email: vietlinh.dhv@gmail.com

Input Parameters

The web panel length (a)	1000.0	mm
The web height (hw)	700.0	mm
The web thickness (tw)	5.0	mm
The flange width (bf)	225.0	mm
The flange thickness (tf)	20.0	mm
The applied load length (c)	200.0	mm
The distance between loaded flange and longitudinal stiffener (b1)	75.0	mm
The width of the stiffener (bst)	60.0	mm
The thickness of stiffener (tst)	5.0	mm
The web yield strength (fyw)	392.0	MPa
The flange yield strength (fyf)	355.0	MPa

Patch Loading Resistance

Patch Loading Resistance (Pu)	699.0562	kN
-------------------------------	----------	----

Predict **Cancel**

Fig. 14. The practical GUI tool for PLR prediction.

To obtain further insight, Fig. 13 shows the Shapley values of each input variable.

6.3. Graphical user interface (GUI) tool and web application (WA)

The previous section shows that the proposed WOA-GBM model can achieve superior accuracy in predicting the PLR of longitudinally stiffened SPGs. Therefore, developing a robust and efficient tool has become imperative for practical use. In this study, such a Graphical User Interface (GUI) tool is developed based on the proposed WOA-GBM model. The GUI tool has been implemented using the Python programming language and allows users to perform tasks interactively. The interface of this GUI tool is shown in Fig. 14. Using this GUI tool, users can enter the numeric values for the web panel length, the web height, the web thickness, the flange width, the flange thickness, the applied load length, the web yield strength, and the flange yield strength. Lastly, the PLR of longitudinally stiffened SPGs is displayed directly by clicking the Predict button. Besides, a WA is developed to use the developed WOA-GBM model effectively. The WA is available at <https://plr-webapp-tvlinh.herokuapp.com/>. For these reasons, the GUI tool and WA are expected to be convenient methods for PLR prediction of longitudinally stiffened SPGs with less effort.

7. Conclusions

This paper investigates the feasibility of hybridizing the gradient boosting machine (GBM) method with the whale optimization algorithm (WOA) for predicting the PLR of longitudinally stiffened SPGs. A total of 137 experimental datasets are carefully collected and used to develop the ML models. Additionally, rank input variables and explain the contributing factors to the PLR estimation are presented using the SHapley Additive exPlanations (SHAP) method. Finally, an efficient GUI tool and a WA are developed to apply for practical use in predicting the PLR of longitudinally stiffened SPGs. The following conclusions can be drawn from this study:

- (1) The WOA can improve the performance of the GBM model.
- (2) The results demonstrate the superior accuracy of the WOA-GBM model, which has the highest R^2 (0.993), lowest $RMSE$ (14.101 kN), and MAE (5.723 kN), outperforms the WOA-AdaBoost and WOA-XGBoost models. The corresponding mean predicted/actual value is 1.006, with a standard deviation of 0.090 and covariance of 0.089.
- (3) Using the SHAP method, the web yield strength and web thickness significantly influence the PLR of longitudinally stiffened SPGs more than others.

(4) The developed GUI tool and a WA are convenient and flexible to estimate the PLR of longitudinally stiffened SPGs quickly.

CRediT authorship contribution statement

Viet-Linh Tran: Software, Methodology, Data curation, Conceptualization, Writing – review & editing, Visualization, Validation, Supervision. **Duy-Duan Nguyen:** Writing – original draft, Visualization, Investigation.

Declaration of competing interest

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

References

- [1] N. Markovic, S. Kovacevic, Influence of patch load length on plate girders. Part I: Experimental research, *J. Constr. Steel Res.* 157 (2019) 207–228, <http://dx.doi.org/10.1016/j.jcsr.2019.02.035>.
- [2] L. Rockey, K. Bergfelt, A. Larsson, Behaviour of Longitudinally Reinforced Plate Girders when Subjected to Inplane Patch Loading, Vol. 7819, Chalmers Univ. Technol. Div. Steel Timber Struct. Publ. S, Goteborg, Sweden, 1978.
- [3] A. Bergfelt, Patch Loading on a Slender Web: Influence of Horizontal and Vertical Web Stiffeners on the Load Carrying Capacity, Vol. 791, Chalmers Univ. Technol. Div. Steel Timber Struct. Publ. S, Goteborg, Sweden, 1979.
- [4] H. Shimizu, S. Yoshida, S. Okuhara, An experimental study on patch loaded web plates, in: ECCS Colloq. Stab. Plate Shell Struct, Ghent Univ., 1987, pp. 85–94.
- [5] M. Janus, K. Kutmanova, I. Skaloud, Experimental Investigation Into the Ultimate Load Behaviour of Longitudinally Stiffened Steel Webs under Partial Edge Loading, Vol. 2, Acta Tech. CSAV, 1988, pp. 158–195.
- [6] N. Marković, N. Hajdin, A contribution to the analysis of the behaviour of plate girders subjected to patch loading, *J. Constr. Steel Res.* 21 (1992) 163–173, [http://dx.doi.org/10.1016/0143-974X\(92\)90025-A](http://dx.doi.org/10.1016/0143-974X(92)90025-A).
- [7] S. Kovacevic, N. Markovic, Experimental study on the influence of patch load length on steel plate girders, *Thin-Walled Struct.* 151 (2020) 106733, <http://dx.doi.org/10.1016/j.tws.2020.106733>.
- [8] Bs 5400, Concrete and Composite Bridges - Part 3: Code of Practice for Design of Steel Bridges, Steel, BSI, 2000.
- [9] 2006 Eurocode 3, EN 1993-1-5, Design of Steel Structures. Part 1-5: Plated Structural Elements, CEN. (n.d.).
- [10] I. Kutmanová, M. Škaloud, Ultimate limit state of slender steel webs subject to (i) constant and (ii) repeated partial edge loading, *J. Constr. Steel Res.* 21 (1992) 147–162, [http://dx.doi.org/10.1016/0143-974X\(92\)90024-9](http://dx.doi.org/10.1016/0143-974X(92)90024-9).
- [11] C. Graciano, B. Edlund, Failure mechanism of slender girder webs with a longitudinal stiffener under patch loading, *J. Constr. Steel Res.* 59 (2003) 27–45, [http://dx.doi.org/10.1016/S0143-974X\(02\)00022-6](http://dx.doi.org/10.1016/S0143-974X(02)00022-6).
- [12] A. Cevik, A new formulation for longitudinally stiffened webs subjected to patch loading, *J. Constr. Steel Res.* 63 (2007) 1328–1340, <http://dx.doi.org/10.1016/j.jcsr.2006.12.004>.
- [13] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* 95 (2016) 51–67, <http://dx.doi.org/10.1016/j.advengsoft.2016.01.008>.
- [14] H. Salehi, R. Burgueño, Emerging artificial intelligence methods in structural engineering, *Eng. Struct.* 171 (2018) 170–189, <http://dx.doi.org/10.1016/j.engstruct.2018.05.084>.
- [15] V.L. Tran, D.K. Thai, S.E. Kim, A new empirical formula for prediction of the axial compression capacity of CCFST columns, *Steel Compos. Struct.* 33 (2019) 181–194, <http://dx.doi.org/10.12989/scs.2019.33.2.181>.
- [16] V.L. Tran, D.K. Thai, D.D. Nguyen, Practical artificial neural network tool for predicting the axial compression capacity of circular concrete-filled steel tube columns with ultra-high-strength concrete, *Thin-Walled Struct.* 151 (2020) <http://dx.doi.org/10.1016/j.tws.2020.106720>.
- [17] V.L. Tran, S.E. Kim, A practical ANN model for predicting the PSS of two-way reinforced concrete slabs, *Eng. Comput.* (2020) <http://dx.doi.org/10.1007/s00366-020-00944-w>.
- [18] D.D. Nguyen, V.L. Tran, D.H. Ha, V.Q. Nguyen, T.H. Lee, A machine learning-based formulation for predicting shear capacity of squat flanged RC walls, *Structures* 29 (2021) 1734–1747, <http://dx.doi.org/10.1016/j.jistruc.2020.12.054>.
- [19] V.L. Tran, D.K. Thai, S.E. Kim, Application of ANN in predicting ACC of SCFST column, *Compos. Struct.* 228 (2019) 111332, <http://dx.doi.org/10.1016/j.compstruct.2019.111332>.
- [20] V.L. Tran, S.E. Kim, Efficiency of three advanced data-driven models for predicting axial compression capacity of CFST columns, *Thin-Walled Struct.* 152 (2020) <http://dx.doi.org/10.1016/j.tws.2020.106744>.
- [21] V.L. Tran, Y. Jang, S.E. Kim, Improving the axial compression capacity prediction of elliptical CFST columns using a hybrid ANN-IP model, *Steel Compos. Struct.* 39 (2021) 319–335, <http://dx.doi.org/10.12989/scs.2021.39.3.319>.
- [22] S. Oh, H.K. Jin, S.J. Joe, H. Ki, Prediction of structural deformation of a deck plate using a GAN-based deep learning method, *Ocean Eng.* 239 (2021) <http://dx.doi.org/10.1016/j.oceaneng.2021.109835>.
- [23] S. Gracia, J. Olivito, J. Resano, B. Martin-del Brio, M. de Alfonso, E. Álvarez, Improving accuracy on wave height estimation through machine learning techniques, *Ocean Eng.* 236 (2021) <http://dx.doi.org/10.1016/j.oceaneng.2021.108699>.
- [24] S. Gu, J. Wang, G. Hu, P. Lin, C. Zhang, L. Tang, F. Xu, Prediction of wind-induced vibrations of twin circular cylinders based on machine learning, *Ocean Eng.* 239 (2021) <http://dx.doi.org/10.1016/j.oceaneng.2021.109868>.
- [25] Q.H. Doan, T. Le, D.K. Thai, Optimization strategies of neural networks for impact damage classification of RC panels in a small dataset, *Appl. Soft Comput.* 102 (2021) 107100, <http://dx.doi.org/10.1016/j.asoc.2021.107100>.
- [26] X. Gao, C. Lin, Prediction model of the failure mode of beam-column joints using machine learning methods, *Eng. Fail. Anal.* 120 (2021) 105072, <http://dx.doi.org/10.1016/j.engfailanal.2020.105072>.
- [27] H. Huang, H.V. Burton, Classification of in-plane failure modes for reinforced concrete frames with infills using machine learning, *J. Build. Eng.* 25 (2019) 100767, <http://dx.doi.org/10.1016/j.jobe.2019.100767>.
- [28] K. Tian, Z. Li, L. Huang, K. Du, L. Jiang, B. Wang, Enhanced variable-fidelity surrogate-based optimization framework by Gaussian process regression and fuzzy clustering, *Comput. Methods Appl. Mech. Engrg.* 366 (2020) <http://dx.doi.org/10.1016/j.cma.2020.113045>.
- [29] K. Tian, Z. Li, J. Zhang, L. Huang, B. Wang, Transfer learning based variable-fidelity surrogate model for shell buckling prediction, *Compos. Struct.* 273 (2021) <http://dx.doi.org/10.1016/j.compstruct.2021.114285>.
- [30] E.T. Fonseca, P.C.G.d.S. Vellasco, S.A.L. De Andrade, M.M.B.R. Vellasco, Neural network evaluation of steel beam patch load capacity, *Adv. Eng. Softw.* 34 (2003) 763–772, [http://dx.doi.org/10.1016/S0965-9978\(03\)00104-2](http://dx.doi.org/10.1016/S0965-9978(03)00104-2).
- [31] E.T. Fonseca, P.C.G. Pedro, S.A.L. De Andrade, M.M.B.R. Vellasco, A patch load parametric analysis using neural networks, *J. Constr. Steel Res.* 59 (2003) 251–267, [http://dx.doi.org/10.1016/S0143-974X\(02\)00024-X](http://dx.doi.org/10.1016/S0143-974X(02)00024-X).
- [32] E.T. Fonseca, P.C.G.d.S. Vellasco, M.M.B.R. Vellasco, S.A.L. de Andrade, A neuro-fuzzy evaluation of steel beams patch load behaviour, *Adv. Eng. Softw.* 39 (2008) 558–572, <http://dx.doi.org/10.1016/j.advengsoft.2007.07.005>.
- [33] A. Cevik, M. Tolga Göğüş, I.H. Güzelbey, H. Filiz, A new formulation for longitudinally stiffened webs subjected to patch loading using stepwise regression method, *Adv. Eng. Softw.* 41 (2010) 611–618, <http://dx.doi.org/10.1016/j.advengsoft.2009.12.001>.
- [34] A.E. Kurtoglu, Patch load resistance of longitudinally stiffened webs: Modeling via support vector machines, *Steel Compos. Struct.* 29 (2018) 309–318, <http://dx.doi.org/10.12989/scs.2018.29.3.309>.
- [35] V. Truong, G. Papazafeiropoulos, Q. Vu, V. Pham, Predicting the patch load resistance of stiffened plate girders using machine learning algorithms, 2021, p. 240.
- [36] C. Molnar, Interpretable machine learning, 2021, pp. 305–342, <http://dx.doi.org/10.1201/9780367816377-16>.
- [37] J. Duan, P.G. Asteris, H. Nguyen, X.N. Bui, H. Moayedi, A novel artificial intelligence technique to predict compressive strength of recycled aggregate concrete using ICA-XGBoost model, *Eng. Comput.* 37 (2021) 3329–3346, <http://dx.doi.org/10.1007/s00366-020-01003-0>.
- [38] Y. Feng, D. Wang, Y. Yin, Z. Li, Z. Hu, An xgboost-based casualty prediction method for terrorist attacks, *Complex Intell. Syst.* 6 (2020) 721–740, <http://dx.doi.org/10.1007/s40747-020-00173-0>.
- [39] W. Zhang, C. Wu, H. Zhong, Y. Li, L. Wang, Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization, *Geosci. Front.* 12 (2021) 469–477, <http://dx.doi.org/10.1016/j.gsf.2020.03.007>.
- [40] N.-V. Luat, S. Whan Han, K. Lee, Genetic algorithm hybridized with extreme gradient boosting to predict axial compressive capacity of CCFST columns, *Compos. Struct.* 278 (2021) 114733, <http://dx.doi.org/10.1016/j.compstruct.2021.114733>.
- [41] J. Nayak, B. Naik, P.B. Dash, A. Souri, V. Shanmuganathan, Hyper-parameter tuned light gradient boosting machine using memetic firefly algorithm for hand gesture recognition, *Appl. Soft Comput.* 107 (2021) <http://dx.doi.org/10.1016/j.asoc.2021.107478>.
- [42] S. Sheikhi, An effective fake news detection method using WOA-xgbTree algorithm and content-based features, *Appl. Soft Comput.* 109 (2021) <http://dx.doi.org/10.1016/j.asoc.2021.107559>.
- [43] H. Dogaki, M. Murata, M. Kishigami, N. Tanabe, T. Yonezawa, Ultimate Strength of Plate Girders with Longitudinal Stiffeners under Patch Loading, Vol. 33, Technol. Reports, Kansai Univ., 1990, pp. 121–132.
- [44] M. Rogač, S. Aleksić, D. Lučić, Influence of patch load length on resistance of I-girders. Part-I: Experimental research, *J. Constr. Steel Res.* 175 (2020) 1–21, <http://dx.doi.org/10.1016/j.jcsr.2020.106369>.
- [45] L.J.-P.S. Walbridge, Patch loading tests of bridge girders with longitudinal web stiffeners [in French], Rapp. d'essais École Polytech. Fédérale Lausanne, ICOM 447, 2001, 2001 [in French].

- [46] L.J.-P.A. Carretero, Introduction des forces concentrées dans les poutres élancées [in French], *Constr. Met.* (1998) 5–18.
- [47] H. Dubas, P. Tschamper, Stabilité des ailes soumises à une charge concentrée et à une flexion globale, *Constr. Met.* (2) (1990) 25–39.
- [48] A. Bergfelt, Girder Web Stiffening for Patch Loading, Vol. 831, Chalmers Univ. Technol. Div. Steel Timber Struct. Publ. S, Göteborg, Sweden, 1983.
- [49] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *J. Comput. System Sci.* 55 (1997) 119–139, <http://dx.doi.org/10.1006/jcss.1997.1504>.
- [50] S. González, S. García, J. Del Ser, L. Rokach, F. Herrera, A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities, *Inf. Fusion* 64 (2020) 205–237, <http://dx.doi.org/10.1016/j.inffus.2020.07.007>.
- [51] J.H. Friedman, Greedy function approximation: A gradient boosting machine, *Ann. Statist.* 29 (2001) 1189–1232.
- [52] P. Bühlmann, T. Hothorn, Boosting algorithms: Regularization, prediction and model fitting, *Stat. Sci.* 22 (2007) 477–505, <http://dx.doi.org/10.1214/07-STS242>.
- [53] S. González, S. García, J. Del Ser, L. Rokach, F. Herrera, A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities, *Inf. Fusion* 64 (2020) 205–237, <http://dx.doi.org/10.1016/j.inffus.2020.07.007>.
- [54] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794, <http://dx.doi.org/10.1145/2939672.2939785>.
- [55] S.M. Lundberg, S.I. Lee, A unified approach to interpreting model predictions, *Adv. Neural Inf. Process. Syst.* 2017 (2017) 4766–4775.