

# FORECASTING MECHANICAL PROPERTIES OF STEEL STRUCTURES THROUGH DYNAMIC METAHEURISTIC OPTIMIZATION FOR ADAPTIVE MACHINE LEARNING

Ngoc-Mai NGUYEN<sup>1</sup>✉, Jui-Sheng CHOU<sup>2</sup>

<sup>1</sup>Department of Civil Engineering and Environmental Informatics, Minghsin University of Science and Technology, Hsinchu, Taiwan

<sup>2</sup>Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

## Article History:

- received 30 October 2023
- accepted 4 March 2024

**Abstract.** Machine learning (ML) presents a promising method for predicting mechanical properties in structural engineering, particularly within complex nonlinear structures under extreme conditions. Despite its potential, research has shown a disproportionate focus on concrete structures, leaving steel structures less explored. Furthermore, the prevalent combination of metaheuristic optimization (MO) and ML in existing studies is often subjective, pointing to a significant gap in identifying and leveraging more effective hybrid models. To bridge these gaps, this study introduces a novel system named the Multiple Metaheuristic Optimizers – Multiple Machine Learners (MMOMML) system, designed for predicting mechanical strength in steel structures. The MMOMML system amalgamates 17 MO algorithms with 15 ML techniques, generating 255 hybrid models, including numerous novel configurations not previously examined. With a user-friendly interface, MMOMML enables structural engineers to tackle inference challenges efficiently, regardless of their coding proficiency. This capability is convincingly demonstrated through two practical applications: steel beams' shear strength and steel cellular beams' elastic buckling. By offering a versatile and robust tool, the MMOMML system meets construction engineers' and researchers' practical and research needs, marking a significant advancement in the field.

**Keywords:** mechanical strength, structural properties, steel structures, machine learning, hybrid models, metaheuristic optimization algorithms, prediction/estimation problems, application interface, multiple metaheuristic optimizers, multiple machine learners.

✉Corresponding author. E-mail: [nnmai@must.edu.tw](mailto:nnmai@must.edu.tw)

## 1. Introduction

Structural engineering involves the analysis and design of load-bearing structures. However, current structural analysis and design approaches require a time-consuming calibration process to handle complex structural systems that exhibit highly nonlinear behavior under extreme actions. This process is excessively complicated and impractical for use in most projects. Machine learning (ML), the most successful branch of artificial intelligence (AI), is well suited to make structural engineering more predictable because of its ability to handle complex nonlinear structural systems under extreme actions (Salehi & Burgueño, 2018; Sun et al., 2021; Zhang et al., 2023). Thai (2022) reviewed 474 Scopus-indexed studies from 1989 to 2021 on ML applications in structural engineering, concluding that scholarly interest in these applications has been rising, especially over the last five years.

Thai (2022) distinguished 474 studies on ML applications into five distinct areas of study, including (1) Pre-

diction of structural members, (2) Prediction of structural fire resistance, (3) Structural health monitoring (SHM) and damage detection, (4) Structural analysis and design; and (5) Prediction of concrete mechanical properties. The area focused on structural members includes studies designed to predict member strength (i.e., shear strength, flexural strength, axial strength, torsional strength, buckling strength, and bond strength) and member deformation (i.e., deflection, drift, and rotation). The area focused on mechanical properties includes studies designed to predict compressive strength, tensile strength, bending strength, and Young's modulus and to optimize concrete mix designs. The number of published studies on ML applications in structural engineering by topic is presented in Table 1.

As shown in Table 1, 373 of the 474 studies (78.7%) addressed problems related to concrete and reinforced concrete (RC) structural components. Although structural steel is the preferred material for high-rise building

construction in many developed countries (Deng et al., 2020; Liu et al., 2018), only 84 studies (17.7%) addressed problems related to steel structures. This may be attributable to the complex, non-linear nature of the inference problems involved in calculating mechanical properties in steel structures and to the time-consuming calibrations required for related structural analyses and design methods, which are overly complicated for practical implementation and generate low-accuracy results (Asif Bin Kabir et al., 2021; Truong et al., 2022). The absence of related research on steel structures highlights the critical need in the construction industry for ML applications that assist engineers in addressing prediction/estimation problems in steel structures.

**Table 1.** Numbers of studies on ML applications in structural engineering, by topic

Area of study			No. of publications
Prediction of structural members	Shear resistance	Concrete and RC structures	71
		Steel structures	3
		Other types of structures	0
	Axial resistance	Concrete and RC structures	33
		Steel structures	2
		Other types of structures	1
	Bond strength	Concrete and RC structures	25
		Steel structures	0
		Other types of structures	1
	Buckling strength	Concrete and RC structures	0
		Steel structures	23
		Other types of structures	0
	Flexural resistances	Concrete and RC structures	7
		Steel structures	2
		Other types of structures	1
	Torsion resistances	Concrete and RC structures	6
		Steel structures	0
		Other types of structures	0
	Deflection	Concrete and RC structures	8
		Steel structures	0
		Other types of structures	6
	Drift and rotation capacity	Concrete and RC structures	1
		Steel structures	1
		Other types of structures	3

End of Table 1

Area of study		No. of publications	
Prediction of structural fire resistance	Concrete and RC structures	13	
	Steel structures	6	
	Other types of structures	1	
SHM and damage detection	Beams/columns	Concrete and RC structures	11
		Steel structures	1
		Other types of structures	0
	Plate/panel members	Concrete and RC structures	21
		Steel structures	3
		Other types of structures	3
	Other systems	Concrete and RC structures	16
		Steel structures	22
		Other types of structures	0
Structural analysis and design	Design	Concrete and RC structures	7
		Steel structures	8
		Other types of structures	0
	Analysis of joints and frames	Concrete and RC structures	32
		Steel structures	13
		Other types of structures	1
Prediction of concrete mechanical properties		122	
<b>TOTAL</b>	<b>Studies on concrete and RC structures</b>	<b>373</b>	
	<b>Studies on steel structures</b>	<b>84</b>	
	<b>Studies on other types of structures</b>	<b>17</b>	

Hybrid machine learning models combining ML techniques with metaheuristic optimization (MO) algorithms have demonstrated advantages in terms of increased prediction accuracy and reduced model dependence on user expertise (Cao et al., 2021, 2022). However, the MO algorithms and ML techniques used in these hybrid models have typically been selected subjectively by researchers. For example, Tran-Ngoc et al. (2019) proposed a cuckoo search (CS)-tuned artificial neural network (ANN) model for detecting damage in bridges and beam-like structures. Although the performance of the proposed model was superior to that of the single ML model used as the comparison, the decision to use CS and ANN rather than other techniques was not explained, leaving the possibility that other hybrid AI models may solve related problems even more effectively and efficiently.

Furthermore, the programs used in hybrid models have rarely been shared publicly, often requiring civil engineers, who typically have limited programming knowledge, to use unreliable code published on the internet. These drawbacks mean that the application of ML in steel structures remains mainly limited to research settings, leaving a gap between research and practical application. Civil engineers and construction managers require a user-friendly ML tool that may be used without coding knowledge to handle forecasting and prediction problems effectively and efficiently.

Therefore, this project was designed to create a novel application interface that integrates different advanced AI techniques, including ML and MO algorithms, to extract usable information hidden in the data and to infer new facts. This application interface, the “multiple metaheuristic optimizers – multiple machine learners” (MMOMML), is a user-friendly, assistive tool that allows users without cod-

ing knowledge to determine the AI model best suited to handle a particular dataset. Moreover, the proposed interface provides a reliable benchmark for verifying the reliability of new and existing inference models. The ability of MMOMML to resolve numerous prediction and forecasting problems in steel structures fulfills the expectations of both users and researchers.

The breakdown percentages of the various ML methods used in structural engineering from 1990 to 2021, shown in Figure 1, support the suitability of using these methods to resolve structural engineering problems. Thus, these methods were adopted as the core models in the proposed MMOMML system. Based on our survey of the most prominently used MO metaphors, presented in Figure 2 (Chou & Nguyen, 2020), integration of these algorithms was proposed to optimize ML technique performance and increase the automation level. Therefore, the proposed MMOMML system was designed to encompass

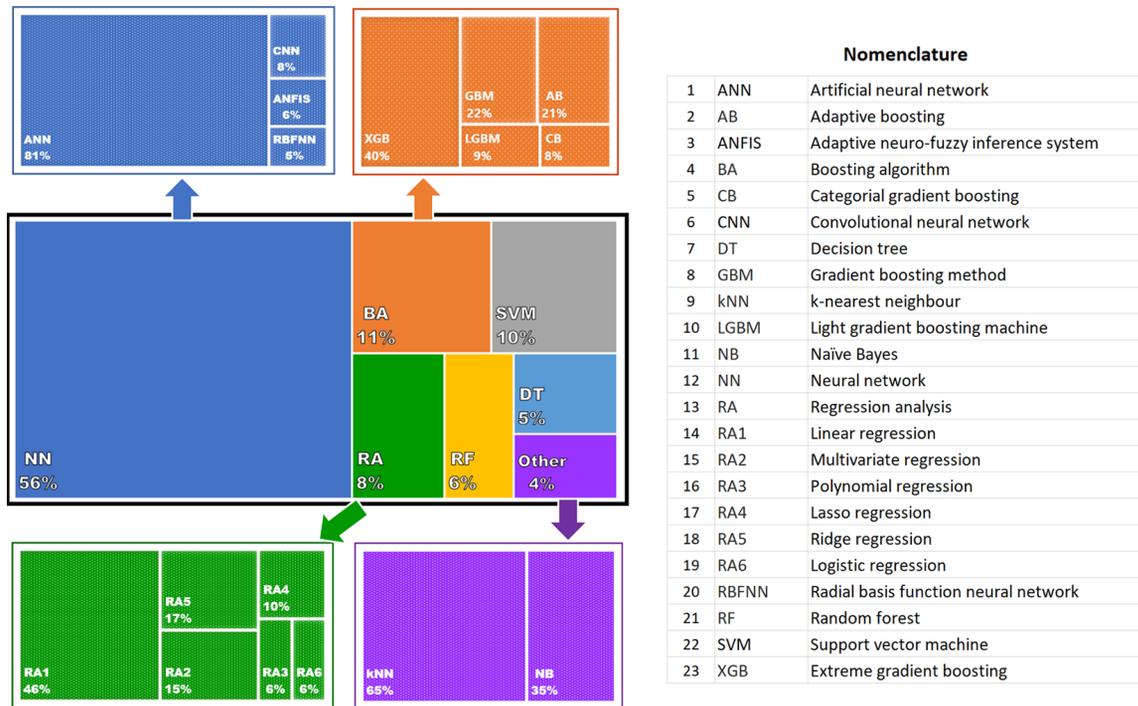


Figure 1. ML methods used in the structural engineering field

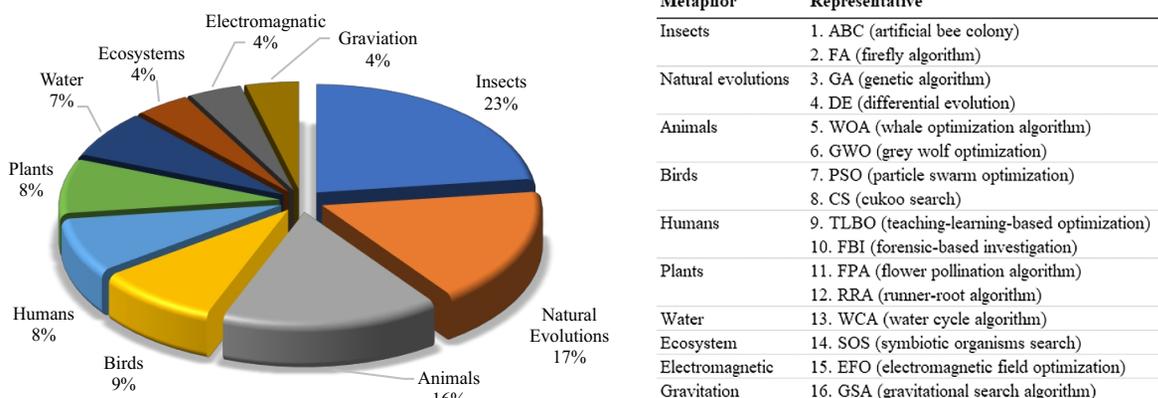


Figure 2. Metaphor-based optimization methods from literature and their associated algorithms

255 powerful and efficient hybrid AI models by integrating 17 MO algorithms and 15 ML techniques. The system was developed to automatically train models to produce the performance criteria values for each model and then determine the most appropriate value for the targeted structure problem.

MMOMML may further enrich the options for improving model performance by suggesting using novel hybrid models that have not previously been used. For example, the system may be used to create a hybrid of FBI and either XGB or LGBM, neither of which have been investigated in the structural engineering field. Furthermore, the MMOMML interface will be updated with emerging MO algorithms and AI techniques and will not be limited only to the proposed MO and AI techniques. Thus, promising hybrid models may be generated progressively to improve prediction accuracy in solving structural engineering problems.

The proposed MMOMML system is expected to resolve numerous prediction and forecasting problems in steel structures and fulfill the expectations of both users and researchers. This powerful interface will be user-friendly and help structural engineers efficiently resolve steel structure inference problems using on-hand data without requiring prior coding knowledge. The remainder of this paper is organized as follows: Section 2 provides a review of the existing literature about machine learning applications in the context of steel structures; Section 3 outlines the research methodology employed in this study, encompassing the model evaluation metrics utilized; Section 4 introduces MMOMML framework and implementation; Section 5 presents the experimental datasets that were obtained in this study, discusses the predictions, and compares model performances; and the final section presents concluding remarks and outlines the research contributions made.

## 2. Literature review

As previously noted, only 84 articles indexed in Scopus and published between 1990 and 2021 were designed to investigate the feasibility of applying ML to steel-structure-related engineering problems (Thai, 2022). These articles cover four areas of study: Prediction of structural members (31 articles; 36.9%), Steel structure analysis and design (21 articles; 25.0%), Prediction of fire resistance (6 articles; 7.1%), and SHM and damage detection (26 articles; 31.0%). As shown in Figure 3, fewer than four studies have addressed many categories within these four areas, providing insufficient samples to analyze the merits of applying ML to related problems.

The techniques used in the 84 ML studies on steel-structure-related engineering problems are outlined in Table 2. Backpropagation method-based ANN was the most frequently used technique to predict mechanical behavior in steel structures. In terms of other techniques considered, Zakir Sarothi et al. (2022) used ANN, SVM, DT, RF, kNN, linear regression, lasso regression, ridge regression, AB, XGB, and CatBoost to predict bearing capacity in double shear bolted connections, finding that RF delivered the best performance. Also, Ghiasi et al. (2016) conducted a comparative study on the performances of ANN, ANFIS, LMNN, LS-SVM, MARS, ELM, RF, and GP in classifying damage detected in steel structures, finding LS-SVM to be the best model with the lowest prediction error.

In light of the limited nature of current studies, the potential of other ML techniques to predict mechanical behavior in steel structures should be investigated to assess their potential advantages over currently dominant ANN techniques. Several scholars have proposed applying SVM, LS-SVM, and RF to problems in SHM and damage detection (Ghiasi et al., 2016; Zhou et al., 2012) and structural fire resistance prediction (Hasni et al., 2017). The authors of these studies confirmed the proposed models as superior to ANN in reducing prediction error and computational time. In addition to the above, ANFIS and RBFNN (variants of ANN), DT (Fu, 2020), and CNN (a deep learning technique) have been used in the literature to predict the mechanical behavior of steel structures.

However, most published studies have used conventional ML techniques that require the manual setting of hyperparameter values. Thus, a trial-and-error or grid search method is typically needed to fine-tune these values.

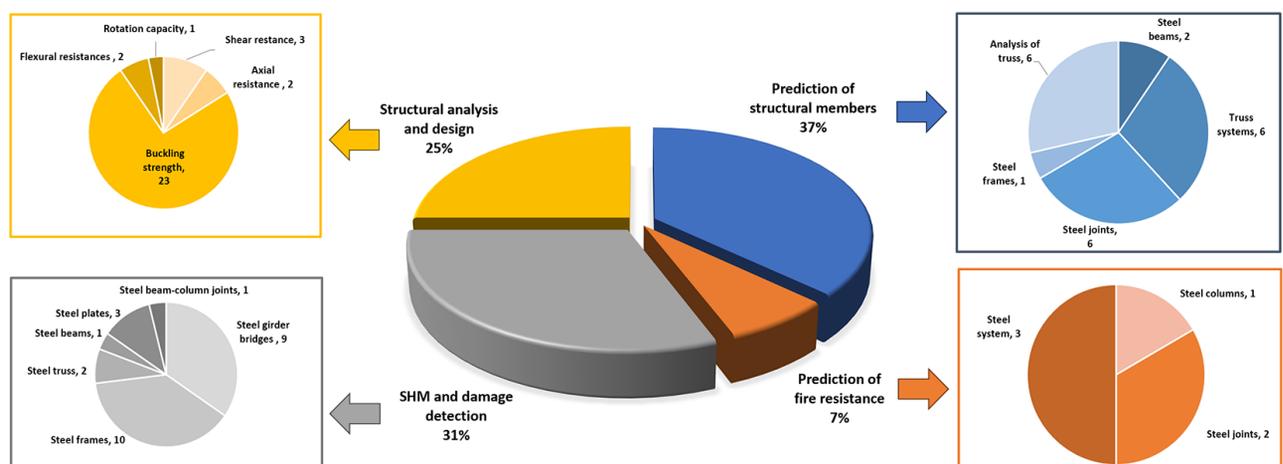


Figure 3. ML applications in steel structural engineering

**Table 2.** Applications of ML on steel structures

Topic		Type	ML algorithms
Prediction of structural members	Shear resistance	Cellular steel beams	ANN
		Bolted connection joints	kNN, ANN, SVM, DT, RF, RA1, RA4, RA5, AdaBoost, XGBoost, CatBoost
	Axial resistance	Steel columns with single-angle sections	ANN
		Steel tube columns strengthened by FRP	ANN
	Local buckling strength	Steel plates	ANN
		Steel panels with hat-stiffeners	ANN
		Thin cylindrical shells	ANN
		Web crippling of CFS sheeting	ANN
		Web crippling of CFS C-section	ANN
		CFS beams with C-sections	ANN
	Distortional buckling strength	Castellated steel beams	ANN
		CFS beams with C-sections	ANN
		Steel beams with I-shaped section	ANN
		Castellated steel beams	ANN
	Global buckling strength	Elliptical tubes	ANN
		Slender columns	ANN
		Beam-columns with various end conditions	ANN
Lateral-torsional buckling	Steel columns with tapered I-section	ANN	
	Cellular I-beams	ANN	
Flexural resistances	Steel beam with CHS	ANN	
Rotation capacity	Steel I-beams	ANN	
Prediction of structural fire resistance	Fire resistance	Steel columns	ANN
	Joints	Semi-rigid steel joints	ANN
		T-joints with hollow steel sections	ANN
	Systems	Steel frames	ANN, DT, kNN
Steel trusses		ANN	
SHM and damage detection	Bridges	Steel girder bridges	ANN, ANFIS, SVM
	Frame/truss systems	Steel frames	ANN, RNFNN, CNN, SVM, RF
		Steel truss	RBFNN, ANN, ANFIS, MARS, LS-SVM, RF, RA2, GP
	Beams/columns	Steel beams	ANN
	Plate/panel members	Corroded steel plates and bolts	CNN
		Steel gusset plates	CNN
Joints	Steel beam-column joints	CNN	
Structural analysis and design	Design	Steel beams	ANN, ANFIS
		Truss systems	ANN, RBFNN
	Analysis of steel joints	Moment-rotation curve	ANN
		Hysteresis behavior	ANN
		Moment and rotation	ANN, ANFIS
Analysis of frames	Reliability analysis of steel frames	ANN	

However, these methods generally do not achieve optimal model performance because of the extensive range of potential continuous hyperparameter values. Several scholars who investigated using a metaheuristic algorithm instead of trial-and-error/grid search methods reported promising results (Tran-Ngoc et al., 2019). Tran-Ngoc et al. (2019) used the cuckoo search algorithm to optimize the configuration of an ANN model, obtaining better results than those generated by the standard ANN model.

In addition, a literature review by Thai (2022) highlighted several concerns related to published studies in which ML was used to predict problems in steel structures. These concerns include: (1) low diversity among the ML techniques used, as most studies used and analyzed the performance of ANN techniques only; (2) few studies proposed advanced hybrid ML models designed to improve prediction accuracy related to steel structure behavior; and (3) failure to fully realize the capabilities of ML because

study authors selected an ML or advanced variant sub-jectively and then compared it to conventional ML techniques only. To address these concerns, a novel MMOMML system designed to help engineers determine the most potent advanced hybrid model is proposed in this study to facilitate efficient and accurate predictions of problems in steel structures.

### 3. Research methods

As mentioned previously, the objective of this study was to build a user-friendly AI interface that integrates 17 robust MO algorithms and 15 powerful ML techniques and generates 255 hybridization models that are more effective and efficient than current inference models. 17 MO algorithms included in the proposed MMOMML are ABC (Karaboga & Basturk, 2007), FA (Yang, 2010), GA (Goldberg, 1989; Holland, 1975), DE (Storn & Price, 1997), WOA (Mirjalili & Lewis, 2016), PSO (Kennedy, 2011), CS (Yang & Deb, 2009), TLBO (Sahu et al., 2015), FBI (Chou & Nguyen, 2020), FPA (Yang, 2012), RRA (Merrikh-Bayat, 2015), WCA (Eskandar et al., 2012), GWO (Mirjalili et al., 2014), SOS (Cheng & Prayogo, 2014), EFO (Abedinpourshotorban et al., 2016), GSA (Rashedi et al., 2009), and equilibrium optimizer (EO) (Faramarzi et al., 2020).

The 15 ML models included in the proposed MMOMML are linear regression (Galton, 1886; Pearson, 1904), multivariate regression (Kendall, 1957), logistic regression (Conolly, 1958), multivariate adaptive regression splines (MARS) (Friedman, 1991), classification and regression tree (CART) (Gordon et al., 1984), least square support vector regression (LSSVR) (Suykens & Vandewalle, 1999), SVR (Cortes & Vapnik, 1995; Vapnik et al., 1996), ANN (McCulloch & Pitts, 1943; Rumelhart et al., 1986), ANFIS (Jang, 1993), RBFNN (Moody & Darken, 1989), random forest (RF) (Breiman, 2001), LogitBoost (Friedman et al., 2000), XGB (Chen & Guestrin, 2016), LightGBM (LGBM) (Ke et al., 2017), and AdaBoost (AB) (Freund & Schapire, 1997; Hastie et al., 2009).

#### 3.1. Single machine learning models

This section focuses on ML algorithms commonly applied in structural engineering, categorized into six groups based on their prevalent use in the literature concerning steel structures.

##### 3.1.1. Regression analysis (RA)

Regression analysis (RA) is a predictive modeling approach rooted in statistics, initially designed to explore the intricate relationship between independent (predictor) and dependent (target) variables. Over time, RA has been integrated into Machine Learning (ML), operating within the framework of supervised learning algorithms. Its primary goal is to predict output values based on corresponding input variable values. In the context of ML, regression models differ based on three main factors: (i) the number of incorporated variables, (ii) the type of variables includ-

ed, and (iii) the geometric configuration of the regression line. Commonly used RA models in structural engineering encompass Linear regression, Multivariate regression, Logistic regression, and Multivariate Adaptive Regression Splines (MARS) (Cheng & Cao, 2016).

##### 3.1.2. Artificial neural networks (ANNs)

ANN is a versatile machine learning algorithm widely used in structural engineering. It mimics biological neurons and consists of layers, including an input layer for data reception and an output layer for result prediction. The Back-propagation Neural Network (BPNN) is a common variant that adjusts weights during training to minimize errors. Radial Basis Function Neural Network (RBFNN) uses RBF as an activation function, offering fast training with a single hidden layer. Adaptive Neuro-Fuzzy Inference System (ANFIS) combines fuzzy logic and NN to minimize errors between input data and predictions, with an architecture featuring fuzzy, product, normalized, de-fuzzy, and overall output layers.

##### 3.1.3. Support vector machine (SVM)

The SVM algorithm has gained significant prominence owing to its efficacy and simplicity, rendering it among the most influential and widely employed methodologies. The fundamental concept underpinning the SVM algorithm entails an initial differentiation of data feature groups, succeeded by pursuing an optimal separating hyperplane endowed with a maximal margin. To address nonlinearly separable data, kernel functions such as the Radial Basis Function (RBF) and the sigmoid function are harnessed, facilitating the transformation of the original data into a novel space wherein linear separation becomes feasible. LSSVR is an SVM algorithm and extension of support vector regression (SVR) that incorporates many advanced features, including expedited computation and heightened generalizability (Suykens et al., 2002).

##### 3.1.4. Decision tree (DT)

Alternatively referred to as the Classification and Regression Tree (CART), DT constitutes a tree-structured paradigm employed to elucidate and portray the intricacies of decision-oriented procedures. The prevalence of DT has surged notably, attributed to its intrinsic simplicity and adeptness in accommodating diverse data types encompassing numerical and categorical attributes. The compositional makeup of this model comprises four constituent elements: a foundational root node, an array of bifurcating branches, intermediary decision nodes, and culminating leaf nodes, commonly known as terminal nodes.

##### 3.1.5. Random forest (RF)

RF employs an ensemble learning strategy utilizing Decision Trees (DTs) as weak learners. Multiple DTs are cultivated within a forest framework through bagging with parallel training. The stochastic feature selection process, earning the name "Random Forest", involves creating a

multitude of individual DTs. The outcome aggregation consists of a consensus mechanism, with classification using a majority vote and regression employing an averaging procedure. This approach enhances predictive models in various applications, offering robust and accurate results.

### 3.1.6. Boosting algorithm (BA)

Boosting Algorithms (BA) enhance predictive models in structural engineering by combining multiple models for improved effectiveness. Key strategies include AdaBoost, employing Decision Trees (DT) and Random Forest (RF) algorithms; Gradient Boosting Machine (GBM), which enhances AdaBoost with DT weak learners and minimizes the loss function using gradient descent; XGBoost, optimizing GBM with techniques like parallel processing for efficiency; LightGBM, prioritizing computational efficiency with a unique leaf-wise tree expansion strategy; and LogitBoost (LB), a binary classification-focused variant using logistic regression as its base classifier. Collectively, these techniques contribute to swift and precise model predictions in structural engineering applications.

## 3.2. Metaheuristic optimization algorithms

Optimization problems aim to find optimal solutions within a solution set that maximizes (or minimizes) an objective function while adhering to constraints. Metaheuristics, versatile heuristic algorithms, address large-scale problems effectively and have gained popularity due to their efficiency. They excel in solving engineering problems by working efficiently with simple concepts, not requiring gradient information, overcoming local optima issues, and being applicable across various disciplines (Gandomi & Alavi, 2012; Mirjalili & Lewis, 2016; Wang, 2018).

### 3.2.1. Single solution-based metaheuristics (S-MOs)

S-MOs navigate problem spaces iteratively, generating candidate solutions from the current solution. In the generation phase, a set of candidates ( $C(s)$ ) is created through local transformations. In the replacement phase, a new solution ( $s'$ ) from  $C(s)$  is chosen to replace the current solu-

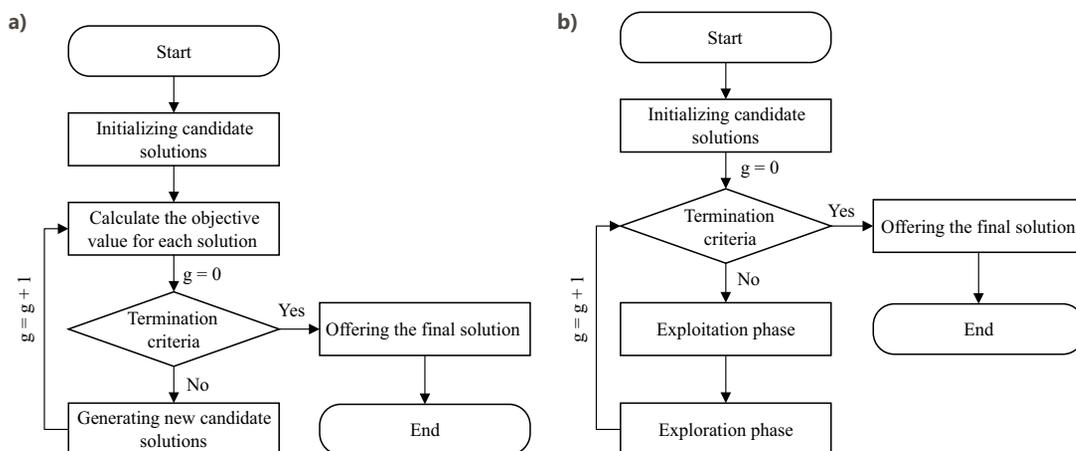
tion. This process repeats until the stopping criterion is met. Noteworthy examples of S-MOs, also termed local search algorithms, include local search, simulated annealing (SA), and tabu search (TS). While these algorithms once held prominence, recent advancements have diminished usage in contemporary research.

### 3.2.2. Population-based metaheuristics

Population-based metaheuristic optimizers (P-MOs) are crucial in iteratively enhancing solution populations. These optimizers commence with an initial set (see Figure 4), systematically generating new populations and replacing existing ones until a predefined stopping criterion is met. Notable examples of P-MOs include evolutionary algorithms, ant colony optimization, scatter search, differential evolution, particle swarm optimization, bee colony, and artificial immune systems. The efficacy of P-MOs lies in their ability to balance exploration and exploitation strategies. While early P-MOs such as genetic algorithm (GA) and particle swarm optimization (PSO) primarily focus on exploitation, which may result in confinement to local solutions, contemporary P-MOs like artificial bee colony (ABC), forensic-based investigation (FBI), and symbiotic organisms search (SOS) strategically incorporate both exploration and exploitation. This nuanced approach enhances the probability of discovering globally optimal solutions.

## 3.3. Hybrid machine learning and MO algorithms

Although the advantages of using ML models to address prediction problems are well established, selecting suitable ML models and properly configuring them is daunting, even for users with rich experience using them. Most related algorithms incorporate hyperparameters (HPs), the values of which directly influence the bias and, thus, the predictive performance of the induced models. Despite the increase in the number and popularity of ML tools, users still need help to determine the best HP settings. This is a complex task and may lead practitioners to choose an inferior algorithm over a superior algorithm, as users typi-



**Figure 4.** General flowchart of MO algorithms: a – general flowchart of a single-phase MO; b – general flowchart of a multiple-phase MO

cally adjust HP values using a subjective process of trial and error.

Ideally, the HP values should be defined separately for each problem (Hamdia et al., 2021) using an optimization process that works to find the (near) best settings. Several tuning techniques have been used for this purpose, of which grid search (GS) and random search (RS) are the simplest and most common (Yang & Shami, 2020). The former is better suited to low-dimensional problems with few HPs to set. Due to the large hyperspace, GS cannot explore more promising regions for more complex scenarios. Conversely, RS can explore all possible solutions in hyperspace but cannot perform informed searches, which may lead to high computational costs.

Sequential model-based optimization (SMBO) (Hutter et al., 2011) is a more recently developed technique that has gained attention due to its probabilistic nature. SMBO replaces the target function (ML algorithm) with a surrogate model with faster computational speeds. However, this technique has many HPs and only eliminates the shortcoming of requiring the function targeted for optimization to be evaluated iteratively. Meta-heuristics have also been used to fine-tune the most suitable HP values with an optimization loop, as shown in Figure 5. This method considers the number of pre-defined HPs as the dimension of the optimization problem, with the HPs used as variables for the accuracy objective function. Despite their high computation cost, MOs are widely accepted as a reasonable alternative to GS and RS due to their obviation of tedious trial-and-error efforts and ensuring elegant performance from ML models.

### 4. The MMOMML system

#### 4.1. Training the hybrid models

Data instances must be collected and processed to feed the training process. The data process first normalizes all attributes into a standard range of [0, 1], ensuring the constructed model avoids the negative impact of attributes with large values by making all attributes equally crucial in the initial stage. The normalized dataset is then divided into learning and validation data, which are used for machine learners' learning and validation processes. To compare model generalization, test data are partitioned to verify the difference between actual values and the predicted values generated by the optimal trained model.

The optimization process starts with initializing tuning parameter values, which differ for each ML. For example, the most miniature square support vector machine uses *sigma* and *gamma*. At the same time, the hyperparameters of the radial basis function neural network consist of a hidden neuron number and the width of the Gaussian function. It should be noted that the initial boundaries of the tuning parameters may be extended as the optimal values are approached. Information about the optimized hyperparameters and their respective ranges for all machine learning models implemented in the developed MMOMML system is provided in Appendix (Table A1).

Each set of hyperparameter values combined with the learning data forms a prediction model that may then be validated using the validation data. Hence, each iteration of the optimization process generates many potential pre-

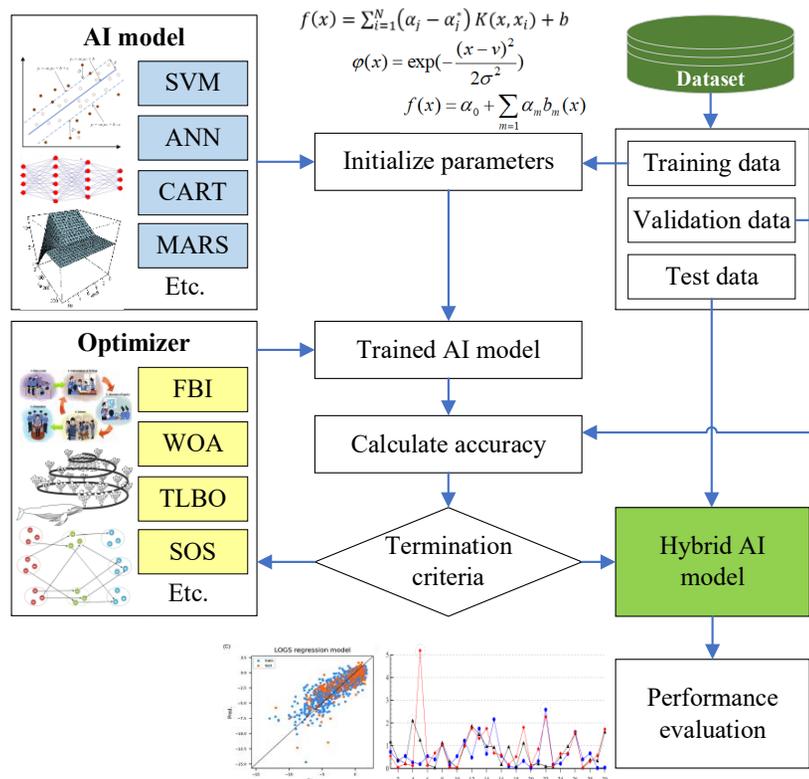


Figure 5. General procedure for integrating a metaheuristic optimizer using machine learning

prediction models with differing levels of accuracy. An objective function is established to identify the optimal set of tuning parameters by comparing the generalization of the corresponding prediction models, as shown in Eqn (1).  $E_{train}$  and  $E_{validate}$  may be formed from the mean square error (MSE), mean absolute percentage error (MAPE), or the coefficient of determination ( $R^2$ ) for the regression problem. MSE was used in this study as the estimation error because this value equalizes the errors in prediction values by severely punishing more significant prediction errors. The objective function consists of training and validation errors that allow a trade-off between model generalization and complexity.

$$f = E_{train} + E_{validate} \tag{1}$$

where  $E_{train}$  and  $E_{validate}$  indicate the average training error and validating error, respectively.

The metaheuristic algorithm uses values of the objective functions to recognize proper solutions (sets of hyper-parameter values). Based on the operation procedure of each MO used in the optimization, solutions are evolved to obtain better values at the next iteration. Generally, a better solution yielding a smaller objective function value will be retained for further calculation, while worse solutions are abandoned. This search loop continues until the termination criterion is met. Specifically, the search loop concludes when either the maximum iteration limit is reached, or there is no improvement in the accuracy objective value for a set number of consecutive iterations. The optimal set of tuning parameter values is found once the optimization process stops and is saved to produce the outcome of new instances in the future. These optimal values may be used as test data to compare model performance. The operational procedure used to develop the hybrid models in the interface is presented in Figure 6.

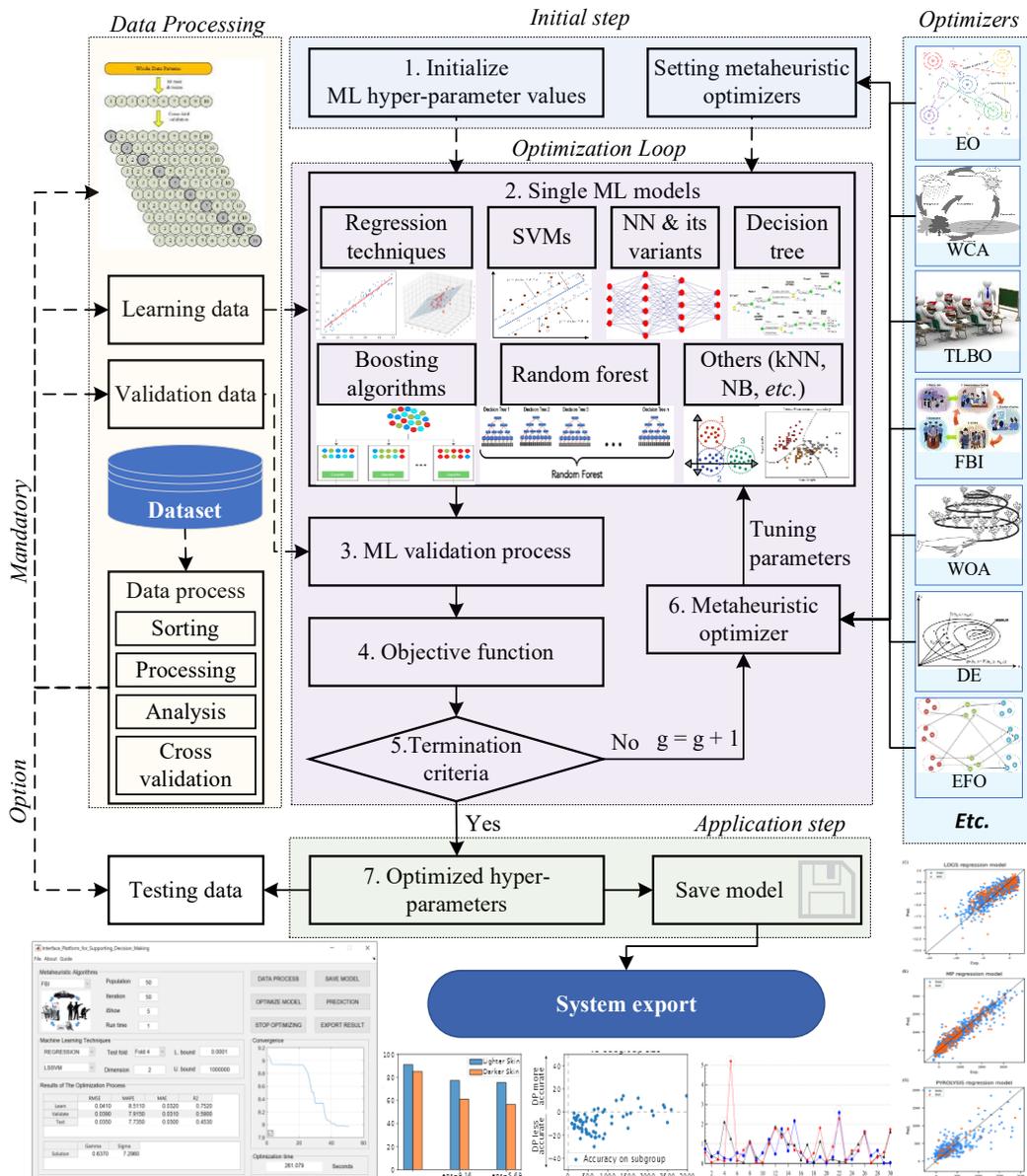


Figure 6. Operational procedure used to develop hybrid models in the interface

### 4.2. The MMOMML interface

The structure of the proposed MMOMML system, presented in Figure 7, includes four main phases: Data input, Modelling, Model saving, and Deployment. The MMOMML interface was designed to resolve numerous prediction and forecasting problems related to steel structures and to meet the needs/expectations of both industry users and researchers. This user-friendly, powerful interface can assist structural engineers in efficiently solving their steel structure inference problems using on-hand data and without prior coding knowledge. The steps for using the interface to train a hybrid ML model are shown in Figure 8.

- **Step 1.** When using the interface for the first time, the user selects DATA PROCESS to handle and save the data in a suitable format that accelerates the training and optimization processes. After clicking the DATA PROCESS button, a new “Data Process” window will display the data processing options listed in the bullet points below. Selecting the “Load Data” button allows the user to browse the datasheet, which should be saved in .xls, .xlsx, or .csv format.
  - Shuffle → randomly changes the sequence of data points.
  - Export CSV → allows users to record a new sequence of data points while the shuffle option is active.
  - Standardize → normalizes the original data values to a standard range of [0, 1].
  - *k* cross-fold → divides the entire dataset into *k* exclusive folds (default *k* = 10).

- **Step 2.** The user selects an MO algorithm from the list for optimization. The user may set desired values for the general parameters, including several optimization iterations and population size, and display the optimization results for the specified steps. Suppose the selected MO has additional tuning parameters. In that case, these will appear with the default values suggested by the original authors of the MO but may also be adjusted by the user if desired. For example, additional tuning parameters associated with GA include mutation rate, crossover rate, and generation gap.
- **Step 3.** The user designates the type of inference problem by selecting classification or regression.
- **Step 4.** The MACHINE LEARNERS button is activated, allowing the user to select the desired machine learner from a list. Other setting options are continuously activated to allow the user to:
  - Perform test folds for *k*-fold cross-validation runs or randomly set portions of train/test data
  - Set the lower bounds and upper bounds of the hyperparameters
- **Step 5.** The user clicks on the OPTIMIZE MODEL button to start the optimization process, which returns the outcome. Upon completion of model performance optimization, the results are displayed in two tables and one figure, as follows:
  - In the first table, the evaluation index for the regression problem for training, validation, and testing phases are RMSE, MAPE, MAE, and R2, respectively. Meanwhile, CAR, PRE, REC, F1\_score, NPV, TPR, and NPR are chosen to assess the robustness of the model for a classification problem.

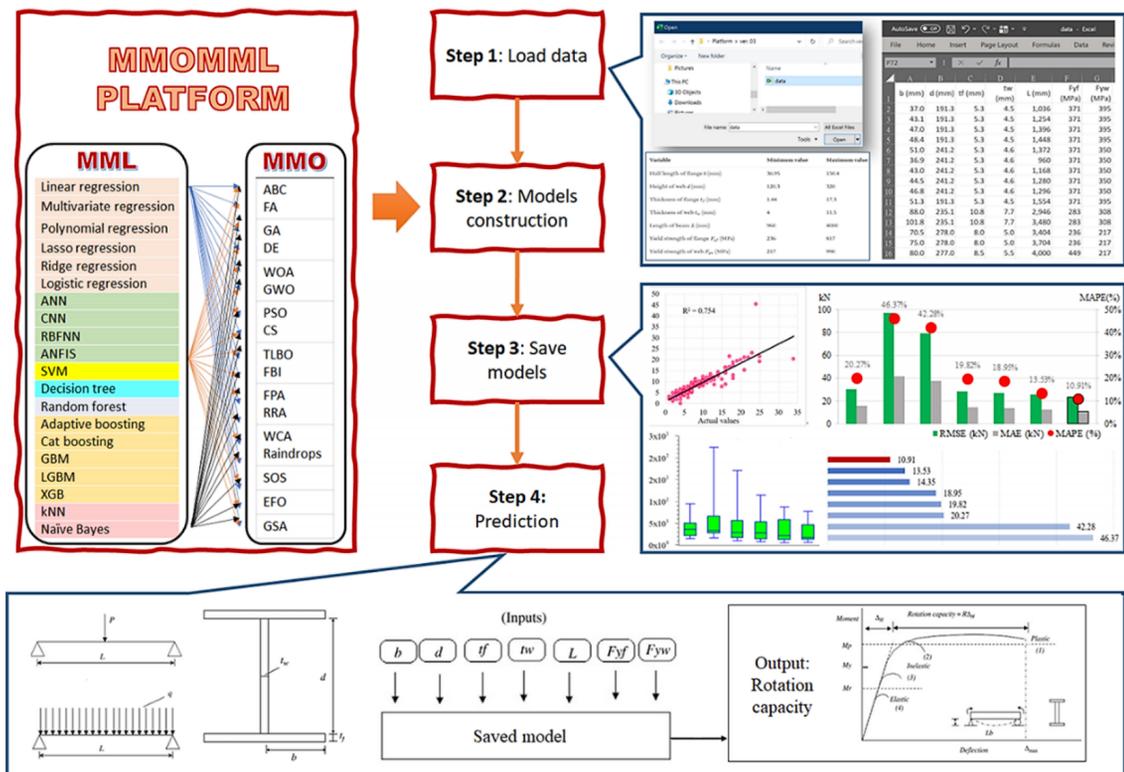


Figure 7. Structure of the MMOMML system

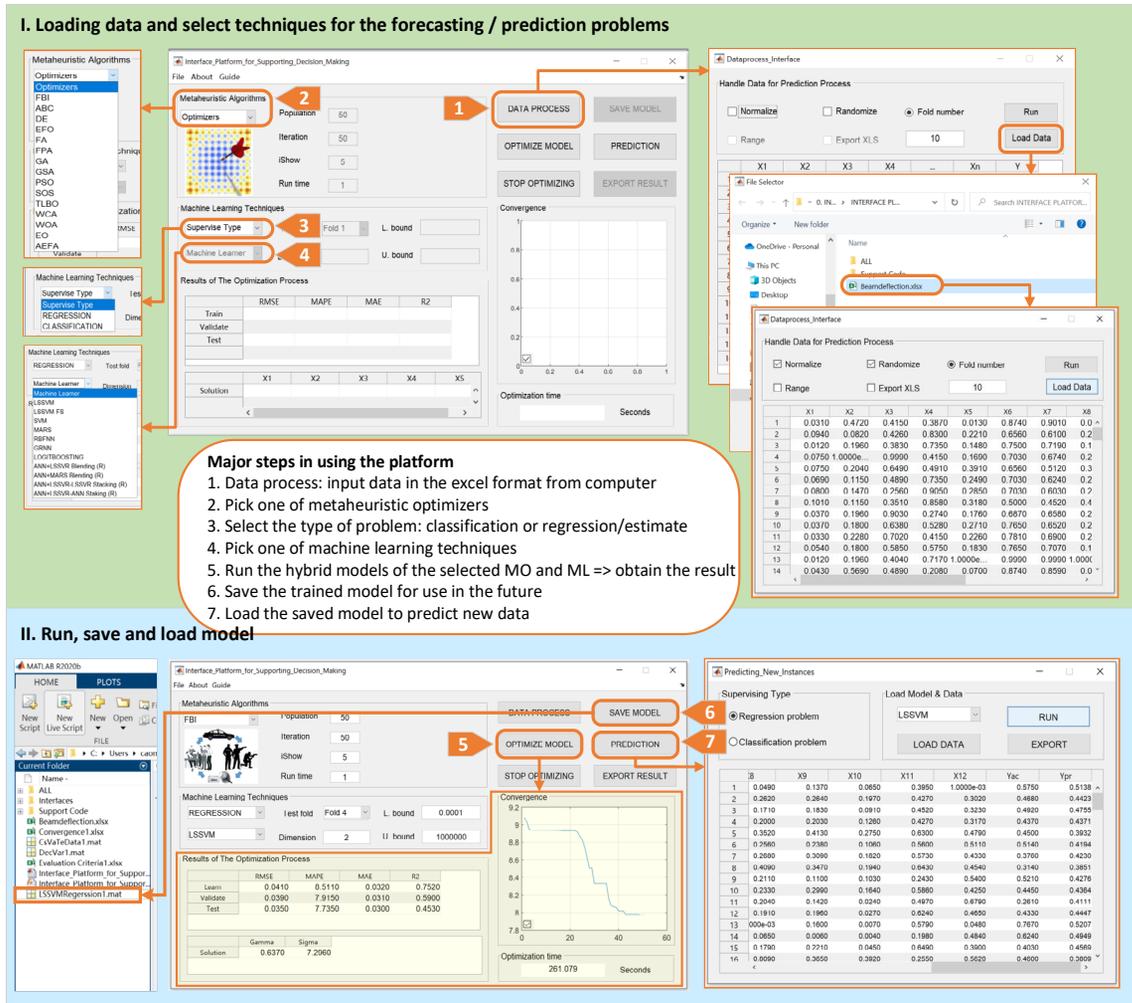


Figure 8. Main interfaces of the proposed hybrid ML interface

- The second table presents the optimal value set of the hyperparameters provided by the selected MO.
- The figure exhibits the convergence curve of the MO optimization process and the computational time.
- **Step 6.** After completion of the optimization process, the user may save the optimized model by clicking on the SAVE MODEL button. The outcome of the optimization process may be exported to .xlsx files for further analysis by clicking on the EXPORT RESULT button.
- **Step 7.** The user may load a previously saved optimized model to infer outcomes from new data patterns by clicking on the PREDICTION button to open a new window, which facilitates the following tasks:
  - Select the inference type (regression or classification)
  - Browse to select a saved model.
  - Browse to load new data in Excel format.
  - Click on the RUN button to execute the prediction task and click on the EXPORT button to save the outcome to a .xlsx file.

## 5. Case data and experimental results

Two steel structure datasets were utilized in this research to validate the efficiency and reliability of the proposed

MMOMML system, which included data on shear strength in steel beams with flat webs and elastic buckling in steel cellular beams.

### 5.1. Shear strength in steel beams

#### 5.1.1. Data processing

The shear strength of steel beams is a crucial parameter influencing the susceptibility of webs to various failure modes, making this variable a pivotal factor in structural design and safety. Experimental studies on the ultimate shear resistance of these girders have revealed that they exhibit typical diagonal shear buckling of the web and develop plastic hinges in the flanges at failure loads. Bending theory may be employed to analyze how internal forces carried by the web and flanges in a plate girder respond to minor shear loads (see Figure 9).

The ultimate shear resistance of steel plate girders has been investigated extensively using experimental and theoretical methods. Despite the extensive research conducted in this area over the past three decades, current methods remain unable to accurately predict the ultimate shear resistance of plate girders (Elamary et al., 2023), underscoring the need for more reliable and precise procedures to enhance structural analysis and design.

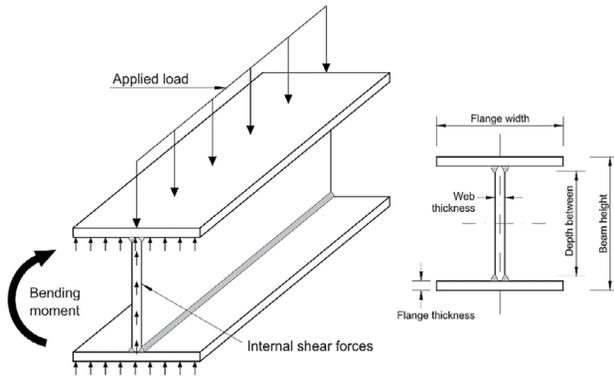


Figure 9. Plate girder responses to minor shear loads

In this study, experimental data on the shear strength of steel beams with flat webs were collected from various articles in the literature. These data were then classified, filtered, and organized based on beam characteristics and other effective parameters. The systematically processed data generated a curated dataset of 90 experimental test results derived from 19 published papers (Adoriso, 1982; Basler et al., 1960; Carskaddan, 1968; Cooper et al., 1964; Evans et al., 1977, 1979; Evans, 1984, 1986; Evans & Tang, 1983; Kamtekar et al., 1972, 1974; Konishi, 1965; Lee & Yoo, 1998; Roberts & Shahabian, 2000; Rockey & Skaloud, 1972; Rockey et al., 1981; Sakai et al., 1966; Skaloud, 1971; Tang & Evans, 1984), which was used to validate the proposed MML system. The complete dataset is provided in Appendix (Table A2). A critical overview of the key parameters relevant to the experimental shear load at failure ( $V_{exp}$ ) of steel plate girders is provided in Table 3. This comprehensive dataset facilitates an in-depth understanding of the behavior and performance of steel plate girders under various loading conditions.

### 5.1.2. Numerical results

Accurate predictive modeling requires that model performance be comprehensively evaluated. The ten-fold cross-validation technique used in this study is a widely cited, robust approach to assessing the predictive capabilities of machine learning models. Ten-fold cross-validation involves partitioning a dataset into ten distinct subsets or

folds. In each iteration, nine folds are utilized for model training, while the remaining fold is used for validation. This process is iteratively repeated ten times, ensuring that each fold serves as a validation set precisely once. Ten-fold cross-validation is a practical approach to mitigating random-selection bias in model evaluations. The outcomes of these iterations are aggregated to provide a holistic understanding of the generalizability of a model to other diverse data subsets.

In this study, a suite of four evaluation metrics, including Mean Square Error ( $MSE$ ), Mean Absolute Percentage Error ( $MAPE$ ), and Coefficient of Determination ( $R^2$ ), was used to comprehensively assess model performance in terms of predictive accuracy, error distribution, and explanatory power:

1. Mean Squared Error ( $MSE$ ):  $MSE$  calculates the average of the squared differences between predicted and actual values. This metric provides a higher-level view of error distribution, facilitating a broader-scale model accuracy evaluation.
2. Mean Absolute Percentage Error ( $MAPE$ ):  $MAPE$  computes the average percentage difference between predicted and actual values. This metric is valuable in scenarios where understanding relative error's impact is crucial, allowing for an intuitive interpretation of predictive accuracy.
3. Coefficient of Determination ( $R^2$ ):  $R^2$  assesses the proportion of variance in the dependent variable that is explainable by the independent variables in the model. This metric provides insights into the model's explanatory power and its ability to capture the underlying relationships within the data.

The mathematical formulas for those metrics are as follows:

$$MSE = \left(\frac{1}{n}\right) \times \sum_{i=1}^n (|p_i - y_i|); \quad (2)$$

$$MAPE = \left(\frac{1}{n}\right) \times \sum_{i=1}^n \left(\frac{|p_i - y_i|}{y_i}\right) \times 100; \quad (3)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - p_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}. \quad (4)$$

Table 3. Input and output values of shear strength in steel plate girders

Description	Sym.	Role	Min	Max	Mean	Std.
Clear distance between web stiffeners	$b$ (mm)	X1	172.0	3,810.0	889.3	690.5
Web depth	$d$ (mm)	X2	121.0	1,270.0	673.8	347.3
Flange width	$b_f$ (mm)	X3	25.4	459.0	186.9	103.0
Web thickness	$t_w$ (mm)	X4	1.0	10.0	2.8	2.2
Flange thickness	$t_f$ (mm)	X5	3.1	51.2	13.6	8.9
Web yield stress	$\sigma_{yw}$ (N/mm <sup>2</sup> )	X6	169.0	760.0	260.5	110.1
Flange yield stress	$\sigma_{yf}$ (N/mm <sup>2</sup> )	X7	206.0	790.0	315.3	129.7
Lever arm	$l_a$ (mm)	X8	172.0	3,810.0	1,385.8	946.6
Experimental shear load at failure	$V_{exp}$ (kN <sup>2</sup> )	Y	13.0	5,004.0	344.4	766.7

Note: The lever arm of a steel beam is the perpendicular distance from the line of action of a force applied to the beam to the point about which the beam is expected to rotate.

This study comprehensively understood the predictive models' performance by combining the robust ten-fold cross-validation with the multidimensional, four-metric evaluation. In line with the objectives of this study, integrating these methodologies facilitates informed decision-making on model selection and refinement as well as real-world applicability.

Across all of the hybrid models in the MMOMML system, many generated outcomes indicate suboptimal performance, as indicated by unsatisfactory evaluation metric ( $MSE$ ,  $MAPE$ , and  $R^2$ ) values. The results demonstrate the complexities involved in accurately predicting the outcomes within the given dataset, reflecting the inherent challenges in modeling the behavior of steel structures. Considering  $MAPE$  as the primary assessment criterion,  $MAPE \leq 20\%$  values were presumed to indicate good prediction accuracy, and values  $> 20\%$  were assumed to indicate normal to bad prediction accuracy (Kumar & Kaur, 2016). The performance of models achieving  $MAPE$  value less than 20% (good prediction accuracy) is shown in Table 4, offering insights into models with comparatively favorable performance.

As shown in Table 4, XGB, LSSVR, and RBFNN were quite productive and suitable for the given dataset. FBI\_LSSVR emerged as the most promising candidate by outperforming its counterparts across all evaluation metrics, underscoring its potential for enhanced predictive accuracy. Specifically, FBI\_LSSVR exhibits the lowest  $MAPE$  value (14.13%) and the highest  $R^2$  value ( $R^2 = 0.970$ , approximately 1). SOS\_XGB gave the second-best performance with a  $MAPE = 14.23\%$ . Some of the other models exhibited a trade-off between accuracy and correlation. For example, GWO\_XGB achieved a low  $MAPE$  (15.6%) and a relatively low  $R^2$  (0.875).

Two important insights may be taken from Table 4. First, ML models must be carefully selected to match the specific characteristics of each dataset. Suppose XGB, LSSVR, or RBFNN is chosen appropriately and hybridized with an MO algorithm for parameter optimization. In that case, the resulting hybrid model can be expected to yield predictions with relatively high accuracy. Conversely, when selecting ML models beyond those in Table 4 (e.g., MARS), predictive performance can be expected to be suboptimal (with a  $MAPE$  exceeding 20%) even after hybridization with an MO algorithm. Second, MO algorithms must be carefully selected for parameter optimization based on the ML context. When used with ML algorithms compatible with the provided dataset (XGB, LSSVR, or RBFNN), the appropriate MO algorithms for parameter optimization achieve  $MAPE$  values from 14.13% to 18.83%. However, using a metaheuristic algorithm not listed in Table 4 (e.g., GA\_LSSVR) can be expected to result in suboptimal model performance, with  $MAPE$  values  $> 20\%$ .

Notably, FBI is the only candidate algorithm capable of seamless hybridization with XGB, LSSVR, and RBFNN models, optimizing parameter values to enhance predictive accuracy. By contrast, the other MO algorithms in the proposed system exhibit limitations in either their hybridization potential with these models or their ability to identify optimal parameter values, resulting in compromised accuracy and the  $MAPE$  values of the hybrid models surpassing 20%. These findings underscore the superior efficacy of the FBI algorithm as a versatile multi-objective optimization technique adept at finely adjusting parameters across a diverse array of ML models.

For enhanced visual analysis, the contrast between actual and predicted outputs for the two models is plotted graphically in Figure 10, with FBI\_LSSVR shown to be more accurate overall than GA\_LSSVR. For FBI\_LSSVR, the

**Table 4.** Performance of models with  $MAPE \leq 20\%$  in the MMOMML system

Model	Learning			Test		
	$MAPE$ %	$MSE$ (kN <sup>2</sup> )	$R^2$	$MAPE$ %	$MSE$ (kN <sup>2</sup> )	$R^2$
ABC-LSSVR	16.25	45.67	0.972	18.43	49.28	0.946
EO-LSSVR	11.83	37.45	0.960	16.15	45.31	0.960
<b>FBI-LSSVR</b>	<b>10.96</b>	<b>38.76</b>	<b>0.948</b>	<b>14.13</b>	<b>47.91</b>	<b>0.970</b>
SOS-LSSVR	12.00	42.40	0.959	17.33	59.08	0.945
TLBO-LSSVR	13.19	34.18	0.973	17.72	53.33	0.936
WCA-LSSVR	12.93	74.71	0.860	17.84	64.10	0.943
FA-RBFNN	11.17	37.46	0.962	17.03	45.65	0.969
FBI-RBFNN	9.00	38.46	0.964	15.22	50.63	0.962
FPA-RBFNN	10.23	41.65	0.956	14.71	56.09	0.941
SOS-RBFNN	8.19	44.02	0.970	17.22	56.23	0.951
FBI-XGB	7.94	25.82	0.981	14.47	55.35	0.945
SOS-XGB	8.41	56.66	0.963	14.23	55.15	0.937
EO-XGB	9.87	48.83	0.956	15.77	62.99	0.896
RRA-XGB	10.33	35.89	0.974	18.83	62.33	0.892
GWO-XGB	8.36	77.18	0.904	15.60	76.83	0.875

visualization reveals a remarkable alignment and even coincidence between the predicted and actual data points. Conversely, for GA\_LSSVR, the disparity between the expected and actual results is more prominent, indicating the model's relative inaccuracy and the broader differences between predicted and exact outcomes.

## 5.2. Elastic buckling of steel cellular beams

### 5.2.1. Description of the data

The elastic buckling behavior of perforated steel beams, specifically cellular beams with repeating web openings, has been studied for over a century (Degtyarev & Tsavdaridis, 2022; Sweedan, 2011). The advantages of these beams over solid-web steel beams include reduced weight, higher strength-to-weight ratio, integration of utilities, and improved aesthetics. The critical geometrical parameters of perforated steel beams are provided in Figure 11 (Rajana et al., 2020). The presence of multiple large openings in cellular beams considerably reduces beam shear strength and introduces various possible failure modes that in-

crease the complexity of their flexural behavior and design.

Prior investigations have examined the specific aspects of cellular beams and beams with different web-opening shapes (Grilo et al., 2018; Panedpojaman et al., 2014; Tsavdaridis & D'Mello, 2011). Rajana et al. (2020) performed an extensive numerical parametric study investigating elastic and inelastic buckling in cellular beams subjected to strong axis bending. The effects of the initial geometric imperfection, material nonlinearity, manufacture-introduced residual stresses, web opening diameter, web-post width, web height, flange width, web and flange thickness, end web-post width, and span of the beams, as well as their combinations, were examined thoroughly.

Rajana et al. (2020) investigated the impact of different parameters on elastic buckling loads and generated an extensive database of finite element simulation results comprising 3645 samples. This comprehensive dataset was used in this study to validate the effectiveness and reliability of the proposed MMOMML system in predicting elastic buckling behavior in steel cellular beams. The statistical characteristics of this dataset are shown in Table 5.

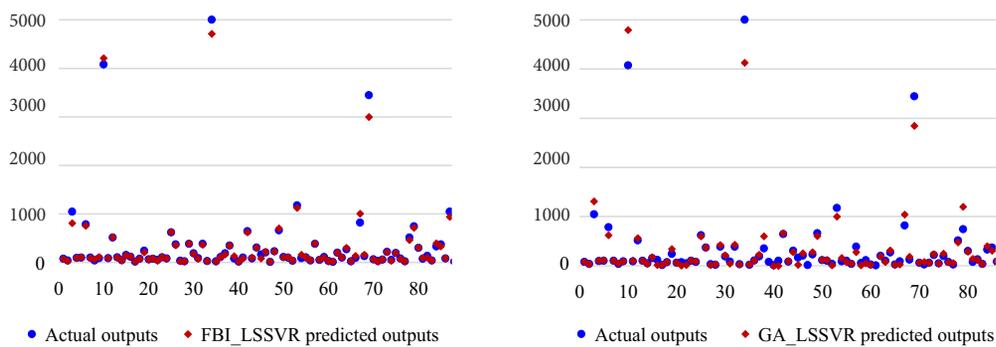


Figure 10. Actual-versus-predictive values of FBI\_LSSVR and GA\_LSSVR

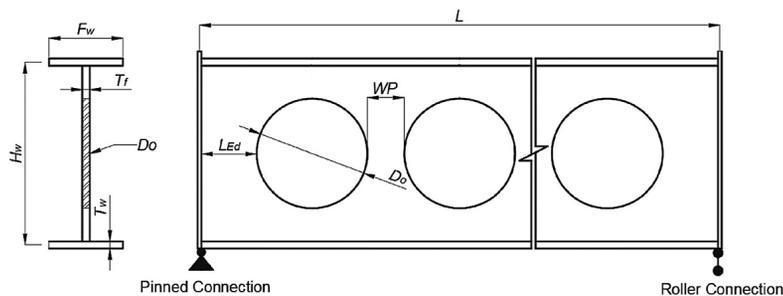


Figure 11. Important dimensional parameters of cellular beams (Rajana et al., 2020)

Table 5. Input and output values of elastic buckling loads in steel cellular beams

Parameter	Sym.	Role	Min	Max	Average	Std.
Length of beams	$L$ (mm)	X1	4000	8000	6,000.0	1414.4
Section height	$H_w$ (mm)	X2	420	700	560.0	114.3
Perforation diameter	$D_o$ (mm)	X3	247	560	383.6	93.0
Width of web post	$WP$ (mm)	X4	24.7	274.4	112.5	68.5
Width of flange	$F_w$ (mm)	X5	162	270	216.0	44.1
End web-post distance (end perforation to support)	$L_{Ed}$ (mm)	X6	12	718	265.4	157.5
Flange thickness	$T_f$ (mm)	X7	15	25	20.0	4.1
Web thickness	$T_w$ (mm)	X8	9	15	12.0	2.4
Elastic buckle load	$Y_{cr}$ (kN/m) <sup>2</sup>	Y	26.4	1,361.7	225.7	182.5

### 5.2.2. Numerical results

Due to the substantial size (3645 samples) of the dataset in this case study, the following approach was used to select a subset of the hybrid ML models in the MMO-MML system for testing:

**Step 1:** The dataset was used first to train all ML models in the MMOMML system. The parameters of these ML models were optimized using the FBI algorithm due to its demonstrated superiority over the other optimization algorithms across a wide range of scenarios (Chou & Nguyen, 2020) and due to its validation in Subsection 5.1.2 as a powerful and effective MO algorithm for fine-tuning parameters across a wide range of ML models. The two most suitable ML models for the given dataset were identified based on the results.

**Step 2:** The two most appropriate ML models were combined with each optimization algorithm available in the system to identify the most effective hybrid ML model for the specific characteristics of the case study. This process aimed to enhance model performance and achieve the best possible results.

In this case study, 10-fold cross-validation was also deployed to enhance model generalizability and mitigate overfitting concerns. As previously mentioned, in the first step, the FBI algorithm was deployed to optimize the parameters of the machine-learning models in the MMOMML system. The results in Table 6 showcase the predictive powers of the five most accurate hybrid models. These results encompass the mean values obtained through the 10-fold cross-validation procedure and the standard deviation values for the four performance metrics, which separately address each model's learning and testing phases.

The results in Table 6 identify FBI\_XGB as the most effective hybrid model. FBI\_XGB exhibited the best performance across all four metrics, earning a *MAPE* of 1.54%, *MSE* of 3.77 (kN/m)<sup>2</sup>, and *R*<sup>2</sup> of 0.997. FBI\_LSSVR was the second-best hybrid model in terms of accuracy (*MAPE* = 2.68%, and *R*<sup>2</sup> = 0.996). Alternatively, FBI\_MARS earned higher average *MSE* and *MAPE* values, indicating lower prediction accuracy. The standard deviations provide in-

sights into the variability in these metrics across different evaluations, with lower values generally indicating more consistent model performance. Overall, the results highlight the efficacy of FBI\_XGB and FBI\_LSSVR in achieving reliably accurate predictions, as well as the other models' lower reliability and higher variability in achieving the same.

Next, XGB and LSSVR, the two most productive models, were integrated with each optimization algorithm into the MMOMML system to identify the optimal hybrid ML model for the unique demands of the current case study. The comprehensive evaluation metrics on the hybrid MO\_XGB and MO\_LSSVR models are presented in Tables 7 and 8, respectively.

The results of the *MSE*, *MAPE*, and *R*<sup>2</sup> metrics shown in Table 7 shed light on the predictive efficacy of the top-ten MO\_XGB models across both learning and test phases. Notably, FBI\_XGB stands out as a strong performer, with consistently low average *MSE* values of 3.67 and 3.77 for the learning and test phases, respectively, and minimal *MAPE* values of 1.44% and 1.54%, respectively. The relatively narrow standard deviations accompanying these values underscore this model's reliability and ability to generate accurate predictions consistently.

Further analysis of Table 7 reveals intriguing patterns among the models. ABC\_XGB, EO\_XGB, SOS\_XGB, TLBO\_XGB, WCA\_XGB, and GWO\_XGB exhibited similar average *MAPE* values of less than 2%, indicating their robust predictive capabilities. However, WCA\_XGB exhibited slightly higher standard deviations, implying varying degrees of performance across different evaluation instances. In addition, both FBI\_XGB and GWO\_XGB stand out for their notable average *R*<sup>2</sup> values of 0.997 and above in both learning and test phases, implying their strong explanatory powers in representing observed variances in the dataset.

The results of the *MSE*, *MAPE*, and *R*<sup>2</sup> metrics shown in Table 8 shed light on the predictive efficacy of the top-ten MO\_LSSVR models across both the learning and test phases. Notably, EO\_LSSVR showcased consistent predictive strength, with average *MAPE* values of 2.58% and 2.68%,

**Table 6.** Performances of the top-five hybrid models

Model		Learning			Test		
		<i>MSE</i> (kN/m) <sup>2</sup>	<i>MAPE</i> %	<i>R</i> <sup>2</sup>	<i>MSE</i> (kN/m) <sup>2</sup>	<i>MAPE</i> %	<i>R</i> <sup>2</sup>
FBI-LSSVR	Avg.	4.88	2.58	0.995	5.08	2.68	0.996
	Std.	0.57	0.25	0.004	0.76	0.20	0.003
FBI-LGBM	Avg.	5.97	2.80	0.995	5.74	2.75	0.996
	Std.	0.54	0.44	0.003	0.76	0.44	0.001
FBI-MARS	Avg.	16.58	9.67	0.981	16.76	9.84	0.979
	Std.	1.58	1.01	0.004	1.43	0.93	0.007
FBI-RBFNN	Avg.	11.10	6.64	0.987	11.53	6.93	0.987
	Std.	0.88	0.40	0.007	0.93	0.84	0.009
<b>FBI-XGB</b>	<b>Avg.</b>	<b>3.67</b>	<b>1.44</b>	<b>0.998</b>	<b>3.77</b>	<b>1.54</b>	<b>0.997</b>
	Std.	0.58	0.19	0.001	0.45	0.13	0.002

respectively, in the learning and test phases. Although the standard deviations indicate some variability, the model demonstrated reliable performance across both phases.

A similar trend was exhibited by FBI-LSSVR, with closely aligned average metrics and a balanced standard deviation underlining its stability.

**Table 7.** Performances of the top-ten MO\_XGB models

Model		Learning			Test		
		MSE (kN/m) <sup>2</sup>	MAPE %	R <sup>2</sup>	MSE (kN/m) <sup>2</sup>	MAPE %	R <sup>2</sup>
<b>FBI-XGB</b>	<b>Avg.</b>	<b>1.44</b>	<b>3.67</b>	<b>0.998</b>	<b>1.54</b>	<b>3.77</b>	<b>0.997</b>
	Std.	0.19	0.58	0.001	0.13	0.45	0.002
ABC-XGB	Avg.	1.88	4.50	0.996	1.91	4.57	0.996
	Std.	0.13	0.28	0.002	0.18	0.67	0.003
EO-XGB	Avg.	1.84	4.46	0.996	1.88	4.47	0.996
	Std.	0.24	0.52	0.003	0.26	0.70	0.002
TLBO-XGB	Avg.	2.06	4.95	0.994	1.96	4.82	0.995
	Std.	0.26	0.52	0.005	0.24	0.58	0.003
SOS-XGB	Avg.	1.76	4.26	0.996	1.74	4.27	0.996
	Std.	0.23	0.46	0.005	0.27	0.44	0.003
FPA-XGB	Avg.	2.16	5.11	0.995	2.20	5.12	0.995
	Std.	0.41	1.00	0.003	0.34	0.82	0.004
WCA-XGB	Avg.	1.95	4.68	0.995	1.95	4.66	0.995
	Std.	0.15	0.26	0.003	0.27	0.85	0.004
RRA-XGB	Avg.	2.54	5.65	0.995	2.62	5.84	0.995
	Std.	0.29	0.79	0.002	0.34	1.03	0.003
GWO-XGB	Avg.	1.65	4.10	0.997	1.70	4.12	0.997
	Std.	0.07	0.43	0.001	0.22	0.64	0.002
WOA-XGB	Avg.	2.11	4.81	0.997	2.10	4.84	0.996
	Std.	0.33	0.61	0.001	0.28	0.54	0.00

**Table 8.** Performances of the top-ten MO\_LSSVR models

Model		Learning			Test		
		MSE (kN/m) <sup>2</sup>	MAPE %	R <sup>2</sup>	MSE (kN/m) <sup>2</sup>	MAPE %	R <sup>2</sup>
ABC-LSSVR	Avg.	2.61	4.97	0.995	2.77	5.17	0.995
	Std.	0.25	0.57	0.003	0.30	0.71	0.003
FBI-LSSVR	Avg.	2.58	4.88	0.995	2.68	5.08	0.996
	Std.	0.25	0.57	0.004	0.20	0.76	0.003
DE-LSSVR	Avg.	2.72	5.14	0.995	2.85	5.20	0.994
	Std.	0.11	0.28	0.002	0.29	0.66	0.007
GWO-LSSVR	Avg.	2.67	5.33	0.995	2.74	5.29	0.994
	Std.	0.17	0.61	0.003	0.28	0.80	0.005
FA-LSSVR	Avg.	2.86	5.44	0.993	2.74	5.00	0.996
	Std.	0.14	0.57	0.005	0.22	0.35	0.002
<b>EO-LSSVR</b>	<b>Avg.</b>	<b>2.72</b>	<b>5.21</b>	<b>0.995</b>	<b>2.64</b>	<b>5.02</b>	<b>0.996</b>
	Std.	0.22	0.52	0.002	0.20	0.62	0.003
SOS-LSSVR	Avg.	2.72	5.11	0.996	2.78	5.30	0.995
	Std.	0.15	0.23	0.001	0.26	0.70	0.003
TLBO-LSSVR	Avg.	2.65	5.12	0.995	2.82	5.46	0.993
	Std.	0.13	0.44	0.003	0.32	1.05	0.008
WOA-LSSVR	Avg.	2.69	5.15	0.995	2.82	5.41	0.995
	Std.	0.21	0.56	0.003	0.28	0.83	0.003
WCA-LSSVR	Avg.	2.67	4.94	0.996	2.80	5.22	0.993
	Std.	0.30	0.55	0.005	0.23	0.70	0.009

The comprehensive assessment of the models in Table 8 reveals consistently high  $R^2$  values from 0.993 to 0.996, indicating diverse performance levels across distinct evaluation instances. This variability in  $R^2$  values underscores the adeptness of these models in capturing data variance. Concurrently, the low  $MAPE$  values observed (2.64% to 2.85%) underscore the high prediction accuracy of these models, substantiating their reliability and suitability for practical applications requiring both robust explanatory and precise forecasting capabilities.

Interpreting the interplay between average values and standard deviations is crucial to comprehensively understanding model performance consistency. While average values provide a measure of central tendency, standard deviations elucidate the dispersion of results, highlighting the stability of a model across different iterations. The data in Tables 7 and 8 permits the predictive aptitudes of the various learning models to be compared and contrasted, allowing the most reliable contenders to be identified and highlighting those models that may exhibit varied outcomes under differing conditions.

## 6. Conclusions

In this research, an innovative application interface called the multiple metaheuristic optimizers – multiple machine learners (MMOMML) system was developed and used to address the predictive and estimation challenges encountered in steel structures. Despite the current prevalence of using machine learning (ML) to predict mechanical strength in structural engineering, much of the related work in the literature has focused on concrete and reinforced concrete structural components. Furthermore, many of these studies have relied on subjectively selected metaheuristic optimization (MO) algorithms and ML models to construct their hybrid ML models, leaving open the possibility that other, unexamined hybrid models may perform even better.

To bridge these gaps, we harnessed two specific datasets related to distinct steel structure properties, in conjunction with advanced algorithms and models, to develop the novel MMOMML system. Advanced artificial intelligence (AI) techniques encompassing 17 MO algorithms and 15 ML techniques were integrated into this system to generate a comprehensive suite of 255 hybrid AI models that expose unexplored, potentially lucrative avenues for novel hybrid model development. Moreover, by automating the training of models using historical real-world data, the MMOMML system bridges the gap in the literature regarding predicting structural behavior in steel structural components. The user-friendly system interface empowers structural engineers to navigate inference challenges related to steel structures, leveraging existing data without requiring coding knowledge. Furthermore, the potential of this system to address a wide range of practical challenges affecting steel structures recommends MMOMML as a dependable tool for structural engineers that meets the expectations of both practitioners and researchers.

Two datasets addressed the shear strength of steel beams and elastic buckling in steel cellular beams were used to validate the proposed MMOMML system. The former comprised 90 experimental results curated from 19 published papers, and the latter comprised a comprehensive database of 3645 finite element simulation results. The system performance evaluation used the ten-fold cross-validation technique augmented by four evaluation metrics, including  $MSE$ ,  $MAPE$ , and  $R^2$ . The ten-fold cross-validation technique and metrics provide comprehensive insights into the proposed system's predictive accuracy, error distribution, and explanatory power. To support transparency and reproducibility, all information about the 10-fold cross-validation dataset and the optimal values of all hyperparameters in hybrid models within the MMOMML system is provided at <https://drive.google.com/drive/folders/1jt2966noFuHkterKaUIFTkhFENNyXsGi?usp=sharing>.

The rigorous analysis in this study highlights two key observations. The first is the pivotal role of selecting the most appropriate ML model for a given dataset. Appropriate models combined with MO algorithms yield substantial predictive accuracy. The second is that judicious MO algorithm selection significantly impacts ML parameter optimization, influencing performance accuracy. This research contributes to enhancing the accuracy and reliability of predictive modeling in steel structures by providing the most effective hybrid ML models for each specific dataset, highlighting the potential of the MMOMML system as a valuable tool for engineers to address structural challenges in these structures.

The MMOMML system is expected to enhance the accuracy and reliability of predictive modeling in steel structures. Its potential for advancement is clear, with plans to expand MMOMML system capabilities by incorporating broader spectrums of advanced ML models and novel MO algorithms. Further diversification of ML models and MO algorithms will provide the MMOMML system with an even more comprehensive array of data related to steel structures, enabling structural engineers to glean invaluable insights into their projects. The ability of the MMOMML system to adapt and evolve ensures its relevance across a myriad of construction industry applications, providing a robust foundation for continued exploration and innovation. The next step is to make the MMOMML system publicly available via a dedicated online platform or website to facilitate broader access and verification. This will empower readers and researchers to utilize the system, train it with their datasets, and validate its performance.

In conclusion, the MMOMML system, a pioneering amalgamation of data-driven insights and advanced AI techniques, marks a significant stride toward transforming how engineers approach challenges in steel structures. The ability of this system to generate accurate predictions, optimize parameters, and foster interdisciplinary collaboration between AI and engineering offers the opportunity to redefine the landscape of structural analysis and design. The MMOMML system is well situated to facilitate cata-

lyzing advancements in both the theoretical and practical realms of steel-structure engineering by providing a dynamic, adaptable, and efficient predictive tool for related engineers and researchers. While this study focused on validating the system's performance within the context of steel structures, it is anticipated that the system's applicability can extend to address estimation and prediction challenges in diverse domains, including but not limited to concrete structures and construction engineering management. This demonstrates the versatility of the MMOMML system in addressing a broader spectrum of predictive modeling challenges across various construction-related domains.

## Acknowledgements

The authors would like to thank the National Science and Technology Council, Taiwan, for financially supporting this research under contract number 111-2222-E-159-002-.

## Competing interests

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

## References

- Abedinpourshotorban, H., Mariyam Shamsuddin, S., Beheshti, Z., & Jawawi, D. N. A. (2016). Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm. *Swarm and Evolutionary Computation*, 26, 8–22. <https://doi.org/10.1016/j.swevo.2015.07.002>
- Adoriso, D. (1982). *Model studies on plate girders subjected to shear loading*. University of Wales Cardiff, UK.
- Asif Bin Kabir, M., Sajid Hasan, A., & Muntasir Billah, A. H. M. (2021). Failure mode identification of column base plate connection using data-driven machine learning techniques. *Engineering Structures*, 240, Article 112389. <https://doi.org/10.1016/j.engstruct.2021.112389>
- Basler, K., Mueller, J., Thurlimann, B., & Yen, B. (1960). *Web buckling tests on welded plate girders, Part 2: Tests on plate girders subjected to bending* (WRC Bulletin, 64), Reprint No. 165 (60-5).
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32. <https://doi.org/10.1023/A:1010933404324>
- Cao, M.-T., Nguyen, N.-M., Chang, K.-T., Tran, X.-L., & Hoang, N.-D. (2021). Automatic recognition of concrete spall using image processing and metaheuristic optimized LogitBoost classification tree. *Advances in Engineering Software*, 159, Article 103031. <https://doi.org/10.1016/j.advengsoft.2021.103031>
- Cao, M.-T., Hoang, N.-D., Nhu, V. H., & Bui, D. T. (2022). An advanced meta-learner based on artificial electric field algorithm optimized stacking ensemble techniques for enhancing prediction accuracy of soil shear strength. *Engineering with Computers*, 38, 2185–2220. <https://doi.org/10.1007/s00366-020-01116-6>
- Carskaddan, P. S. (1968). Shear buckling of unstiffened hybrid beams. *Journal of the Structural Division*, 94, 1965–1990. <https://doi.org/10.1061/JSDAEG.0002039>
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794), San Francisco, California, USA. <https://doi.org/10.1145/2939672.2939785>
- Cheng, M.-Y., & Cao, M.-T. (2016). Estimating strength of rubberized concrete using evolutionary multivariate adaptive regression splines. *Journal of Civil Engineering and Management*, 22(5), 711–720. <https://doi.org/10.3846/13923730.2014.897989>
- Cheng, M.-Y., & Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers & Structures*, 139, 98–112. <https://doi.org/10.1016/j.compstruc.2014.03.007>
- Chou, J.-S., & Nguyen, N.-M. (2020). FBI inspired meta-optimization. *Applied Soft Computing*, 93, Article 106339. <https://doi.org/10.1016/j.asoc.2020.106339>
- Conolly, B. W. (1958). A difference equation technique applied to the simple queue. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20, 165–167. <https://doi.org/10.1111/j.2517-6161.1958.tb00285.x>
- Cooper, P. B., Lew, H. S., & Yen, B. T. (1964). Welded constructional alloy steel plate girders. *Journal of the Structural Division*, 90(1), 1–36. <https://doi.org/10.1061/JSDAEG.0001023>
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, 273–297. <https://doi.org/10.1007/BF00994018>
- Degtyarev, V. V., & Tsavdaridis, K. D. (2022). Buckling and ultimate load prediction models for perforated steel beams using machine learning algorithms. *Journal of Building Engineering*, 51, Article 104316. <https://doi.org/10.1016/j.jobbe.2022.104316>
- Deng, E.-F., Zong, L., Ding, Y., Zhang, Z., Zhang, J.-F., Shi, F.-W., Cai, L.-M., & Gao, S.-C. (2020). Seismic performance of mid-to-high rise modular steel construction – A critical review. *Thin-Walled Structures*, 155, Article 106924. <https://doi.org/10.1016/j.tws.2020.106924>
- Elamary, A. S., Mohamed, M. A., Sharaky, I. A., Mohamed, A. K., Alharthi, Y. M., & Ali, M. A. M. (2023). Utilizing artificial intelligence approaches to determine the shear strength of steel beams with flat webs. *Metals*, 13(2), Article 232. <https://doi.org/10.3390/met13020232>
- Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110–111, 151–166. <https://doi.org/10.1016/j.compstruc.2012.07.010>
- Evans, H. (1984). *A report on the full scale tests on a girder with a stiffened web subjected to combined shear and bending loads*. University of Wales, College of Cardiff, UK.
- Evans, H. (1986). An appraisal, by full-scale testing, of new design procedures for steel girders subjected to shear and bending. *Proceedings of the Institution of Civil Engineers*, 81, 175–189. <https://doi.org/10.1680/iicep.1986.599>
- Evans, H. R., & Tang, K. (1983). *An investigation of the ultimate load behaviour of longitudinally stiffened plate girder webs loaded predominantly in shear*. University College Cardiff, UK.
- Evans, H., Rockey, K., & Porter, D. (1977). Tests on longitudinally reinforced plate girders subjected to shear. In *Proceedings of the Conference of Structural Stability* (pp. 295–304), Liege, Belgium.
- Evans, E., Rockey, K., & Tang, K. (1979). *An investigation into the rigidity of longitudinal web stiffeners for plate girders*. University of Wales, College of Cardiff, UK.
- Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, 191, Article 105190. <https://doi.org/10.1016/j.knosys.2019.105190>

- Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139. <https://doi.org/10.1006/jcss.1997.1504>
- Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1), 1–67. <https://doi.org/10.1214/aos/1176347963>
- Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2), 337–407. <https://doi.org/10.1214/aos/1016218223>
- Fu, F. (2020). Fire induced progressive collapse potential assessment of steel framed buildings using machine learning. *Journal of Constructional Steel Research*, 166, Article 105918. <https://doi.org/10.1016/j.jcsr.2019.105918>
- Galton, F. (1886). Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15, 246–263. <https://doi.org/10.2307/2841583>
- Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12), 4831–4845. <https://doi.org/10.1016/j.cnsns.2012.05.010>
- Ghiasi, R., Torkzadeh, P., & Noori, M. (2016). A machine-learning approach for structural damage detection using least square support vector machine based on a new combinational kernel function. *Structural Health Monitoring*, 15(3), 302–316. <https://doi.org/10.1177/1475921716639587>
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional.
- Gordon, A., Breiman, L., Friedman, J., Olshen, R., & Stone, C. J. (1984). Classification and regression trees. *Biometrics*, 40(3), Article 874. <https://doi.org/10.2307/2530946>
- Grilo, L. F., Fakury, R. H., Castro e Silva, A. L. R. d., & Verissimo, G. d. S. (2018). Design procedure for the web-post buckling of steel cellular beams. *Journal of Constructional Steel Research*, 148, 525–541. <https://doi.org/10.1016/j.jcsr.2018.06.020>
- Hamdia, K. M., Zhuang, X., & Rabczuk, T. (2021). An efficient optimization approach for designing machine learning models based on genetic algorithm. *Neural Computing and Applications*, 33, 1923–1933. <https://doi.org/10.1007/s00521-020-05035-x>
- Hasni, H., Alavi, A. H., Jiao, P., & Lajnef, N. (2017). Detection of fatigue cracking in steel bridge girders: A support vector machine approach. *Archives of Civil and Mechanical Engineering*, 17(3), 609–622. <https://doi.org/10.1016/j.acme.2016.11.005>
- Hastie, T., Rosset, S., Zhu, J., & Zou, H. (2009). Multi-class ada-boost. *Statistics and its Interface*, 2(3), 349–360. <https://doi.org/10.4310/SII.2009.v2.n3.a8>
- Holland, J. (1975). *Adaptation in neural and artificial system*. University of Michigan Press.
- Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In C. A. C. Coello (Eds.), *Lecture notes in computer science: Vol. 6683. Learning and intelligent optimization. LION 2011* (pp. 507–523). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-25566-3\\_40](https://doi.org/10.1007/978-3-642-25566-3_40)
- Jang, J.-S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665–685. <https://doi.org/10.1109/21.256541>
- Kamtekar, A., Dwight, J., & Threlfall, B. (1972). *Tests on hybrid plate girders* (Report No. 2). Cambridge University, Department of Engineering.
- Kamtekar, A., Dwight, J., & Threlfall, B. (1974). *Tests on hybrid plate girders* (Report No. 3). Cambridge University, Department of Engineering.
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39, 459–471. <https://doi.org/10.1007/s10898-007-9149-x>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA.
- Kendall, M. G. (1957). *A course in multivariate analysis*. Charles Griffin & Co.
- Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning* (pp. 760–766). Springer. [https://doi.org/10.1007/978-0-387-30164-8\\_630](https://doi.org/10.1007/978-0-387-30164-8_630)
- Konishi, I. (1965). *Theories and experiments on the load carrying capacity of plate girders* (Report of Research Committee of Bridges, Steel Frames and Welding in Kansai District in Japan). Japan.
- Kumar, S., & Kaur, T. (2016). Development of ANN based model for solar potential assessment using various meteorological parameters. *Energy Procedia*, 90, 587–592. <https://doi.org/10.1016/j.egypro.2016.11.227>
- Lee, S. C., & Yoo, C. H. (1998). Strength of plate girder web panels under pure shear. *Journal of Structural Engineering*, 124(2), 184–194. [https://doi.org/10.1061/\(ASCE\)0733-9445\(1998\)124:2\(184\)](https://doi.org/10.1061/(ASCE)0733-9445(1998)124:2(184))
- Liu, X., Zhou, X., Zhang, A., Tian, C., Zhang, X., & Tan, Y. (2018). Design and compilation of specifications for a modular-pre-fabricated high-rise steel frame structure with diagonal braces. Part I: Integral structural design. *The Structural Design of Tall and Special Buildings*, 27, Article e1415. <https://doi.org/10.1002/tal.1415>
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5, 115–133. <https://doi.org/10.1007/BF02478259>
- Merrikh-Bayat, F. (2015). The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. *Applied Soft Computing*, 33, 292–303. <https://doi.org/10.1016/j.asoc.2015.04.048>
- Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Moody, J., & Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2), 281–294. <https://doi.org/10.1162/neco.1989.1.2.281>
- Panedpojaman, P., Thepchatri, T., & Limkatanyu, S. (2014). Novel design equations for shear strength of local web-post buckling in cellular beams. *Thin-Walled Structures*, 76, 92–104. <https://doi.org/10.1016/j.tws.2013.11.007>
- Pearson, K. (1904). On the theory of contingency and its relation to association and normal correlation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 203, 53–71.
- Rajana, K., Tsavdaridis, K. D., & Koltsakis, E. (2020). Elastic and inelastic buckling of steel cellular beams under strong-axis bending. *Thin-Walled Structures*, 156, Article 106955. <https://doi.org/10.1016/j.tws.2020.106955>
- Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, 179(13), 2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- Roberts, T. M., & Shahabian, F. (2000). Design procedures for combined shear and patch loading of plate girders. *Proceedings of*

- the Institution of Civil Engineers – Structures and Buildings*, 140, 219–225. <https://doi.org/10.1680/stbu.2000.140.3.219>
- Rockey, K., & Skaloud, M. (1972). The ultimate load behaviour of plate girders loaded in shear. *The Structural Engineer*, 50, 29–47. <https://doi.org/10.5169/seals-12050>
- Rockey, K., Valtinat, G., & Tang, K. (1981). The design of transverse stiffeners on webs loaded in shear – an ultimate load approach. *Proceedings of the Institution of Civil Engineers*, 71, 1069–1099. <https://doi.org/10.1680/iicep.1981.1757>
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533–536. <https://doi.org/10.1038/323533a0>
- Sahu, B. K., Pati, S., Mohanty, P. K., & Panda, S. (2015). Teaching–learning based optimization algorithm based fuzzy-PID controller for automatic generation control of multi-area power system. *Applied Soft Computing*, 27, 240–249. <https://doi.org/10.1016/j.asoc.2014.11.027>
- Sakai, F., Doi, K., Nishino, F., & Okumura, T. (1966). *Failure tests of plate girders using large sized models* (Structural Engineering Laboratory Report). Department of Civil Engineering, University of Tokyo, Japan.
- Salehi, H., & Burgueño, R. (2018). Emerging artificial intelligence methods in structural engineering. *Engineering Structures*, 171, 170–189. <https://doi.org/10.1016/j.engstruct.2018.05.084>
- Skaloud, M. (1971). Ultimate load and failure mechanism of thin webs in shear. In *IABSE Colloquium* (Vol. 11, pp. 115–127). <https://doi.org/10.5169/seals-12056>
- Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341–359. <https://doi.org/10.1023/A:1008202821328>
- Sun, H., Burton, H. V., & Huang, H. (2021). Machine learning applications for building structural design and performance assessment: State-of-the-art review. *Journal of Building Engineering*, 33, Article 101816. <https://doi.org/10.1016/j.jobe.2020.101816>
- Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9, 293–300. <https://doi.org/10.1023/A:1018628609742>
- Suykens, J. A. K., Gestel, T. V., Brabanter, J. D., Moor, B. D., & Vandewalle, J. (2002). *Least squares support vector machines*. World Scientific Publishing Company. <https://doi.org/10.1142/5089>
- Sweedan, A. M. I. (2011). Elastic lateral stability of I-shaped cellular steel beams. *Journal of Constructional Steel Research*, 67(2), 151–163. <https://doi.org/10.1016/j.jcsr.2010.08.009>
- Tang, K. H., & Evans, H. R. (1984). Transverse stiffeners for plate girder webs – an experimental study. *Journal of Constructional Steel Research*, 4(4), 253–280. [https://doi.org/10.1016/0143-974X\(84\)90002-6](https://doi.org/10.1016/0143-974X(84)90002-6)
- Thai, H.-T. (2022). Machine learning for structural engineering: A state-of-the-art review. *Structures*, 38, 448–491. <https://doi.org/10.1016/j.istruc.2022.02.003>
- Tran-Ngoc, H., Khatir, S., De Roeck, G., Bui-Tien, T., & Wahab, M. A. (2019). An efficient artificial neural network for damage detection in bridges and beam-like structures by improving training parameters using cuckoo search algorithm. *Engineering Structures*, 199, Article 109637. <https://doi.org/10.1016/j.engstruct.2019.109637>
- Truong, V.-H., Pham, H.-A., Huynh Van, T., & Tangaramvong, S. (2022). Evaluation of machine learning models for load-carrying capacity assessment of semi-rigid steel structures. *Engineering Structures*, 273, Article 115001. <https://doi.org/10.1016/j.engstruct.2022.115001>
- Tsavidaridis, K. D., & D'Mello, C. (2011). Web buckling study of the behaviour and strength of perforated steel beams with different novel web opening shapes. *Journal of Constructional Steel Research*, 67(10), 1605–1620. <https://doi.org/10.1016/j.jcsr.2011.04.004>
- Vapnik, V., Golowich, S., & Smola, A. (1996). Support vector method for function approximation, regression estimation and signal processing. In *Proceedings of the 9th International Conference on Neural Information Processing Systems (NIPS'96)* (pp. 281–287).
- Wang, G.-G. (2018). Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing*, 10, 151–164. <https://doi.org/10.1007/s12293-016-0212-3>
- Yang, X.-S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2), 78–84. <https://doi.org/10.1504/IJBIC.2010.032124>
- Yang, X.-S. (2012). Flower pollination algorithm for global optimization. In J. Durand-Lose, & N. Jonoska (Eds.), *Unconventional computation and natural computation. UCNC 2012. Lecture notes in computer science* (Vol. 7445, pp. 240–249). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-32894-7\\_27](https://doi.org/10.1007/978-3-642-32894-7_27)
- Yang, X.-S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)* (pp. 210–214), India. IEEE. <https://doi.org/10.1109/NABIC.2009.5393690>
- Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, 295–316. <https://doi.org/10.1016/j.neucom.2020.07.061>
- Zakir Sarothi, S., Sakil Ahmed, K., Imtiaz Khan, N., Ahmed, A., & Nehdi, M. L. (2022). Predicting bearing capacity of double shear bolted connections using machine learning. *Engineering Structures*, 251, Article 113497. <https://doi.org/10.1016/j.engstruct.2021.113497>
- Zhang, J., Cao, Y., Xia, L., Zhang, D., Xu, W., & Liu, Y. (2023). Intelligent prediction of the frost resistance of high-performance concrete: a machine learning method. *Journal of Civil Engineering and Management*, 29(6), 516–529. <https://doi.org/10.3846/jcem.2023.19226>
- Zhou, Q., Ning, Y., Zhou, Q., Luo, L., & Lei, J. (2012). Structural damage detection method based on random forests and data fusion. *Structural Health Monitoring*, 12(1), 48–58. <https://doi.org/10.1177/1475921712464572>

## APPENDIX

**Table A1.** Information about the optimized hyperparameters and their respective ranges for all machine learning models implemented in the developed MMOMML system

No.	Models	Hyperparameters	Type	Range
1	LSSVR	Gamma	Float	10e-6 ~ 10e6
		Sigma	Float	10e-6 ~ 10e6
2	SVR	Regularization parameter	Float	10e-3 ~ 10
		Gamma	Float	10e-3 ~ 10
		Epsilon (For regression problem)	Float	10e-4 ~ 1
3	ANN	Number of hidden layers	Integer	1 ~ 5
		Number of neurons in one layer	Integer	5 ~ 30
		Learning rate	Float	10e-3 ~ 1
4	RBFNN	Number of hidden neurons	Integer	5 ~ 100
		Spread of kernel	Float	10e-3 ~ 1
5	MARS	Maximum number of basis functions	Integer	10 ~ 50
		Penalty coefficient	Float	0.01 ~ 10
		Maximum interaction	Integer	1 ~ 10
6	LogitBoost	Learning cycle number	Integer	10 ~ 100
		Learning rate	Float	10e-3 ~ 1
		Minimum leaf size	Integer	1 ~ 100
		Maximum number of splits	Integer	1 ~ 100
7	ADABOOST	Maximum depth	Integer	1 ~ 20
		Number of estimators	Integer	5 ~ 100
		Learning rate	Float	10e-3 ~ 1
8	LGBM	Number of leaves	Integer	1 ~ 40
		Learning rate	Float	10e-3 ~ 1
		Feature fraction	Float	0.6 ~ 1
		Maximum depth	Integer	2 ~ 10
		Minimum sample number in a leaf	Integer	5 ~ 10
9	XGB	Maximum depth	Integer	1 ~ 10
		Number of estimators	Integer	5 ~ 100
		Learning rate	Float	10e-3 ~ 1
		Minimum sample number in a leaf	Integer	1 ~ 10
10	RF	Minimum sample number to split	Integer	2 ~ 10
		Number of estimators	Integer	5 ~ 100
11	Decision Tree	Minimum sample number to split	Integer	2 ~ 10
		Complexity parameter	Float	0 ~ 10
		Minimum sample number in a leaf	Integer	1 ~ 10
12	Linear Regression	Learning rate	Float	10e-3 ~ 1
13	Logistic Regression	Learning rate	Float	10e-3 ~ 1
14	ANFIS	Gaussian function center	Float	-5 ~ 5
		Number of Gaussian functions	Integer	1 ~ 10
		Width of Gaussian functions	Float	0.1 ~ 2
		Learning rate	Float	10e-3 ~ 1
15	Multivariate Regression	Regularization parameter	Float	10e-3 ~ 10
		Learning rate	Float	10e-3 ~ 1

**Table A2.** The collected dataset of shear strength for steel beams with flat webs

Specimen	$b$ mm	$d$ mm	$b_f$ mm	$t_w$ mm	$t_f$ mm	$\sigma_{yw}$ N/mm <sup>2</sup>	$\sigma_{yf}$ N/mm <sup>2</sup>	$l_a$ mm	$V_{exp}$ kN <sup>2</sup>	Ref.
A1	172	133	25.4	1.42	6.35	258	287	172	29.0	Sung and Yoo (1998)
A4	254	121	34.9	1.42	6.35	258	287	254	26.0	Sung and Yoo (1998)
C4	254	356	41.3	1.47	6.35	258	287	508	41.0	Sung and Yoo (1998)
G6-T1	1,905	1,270	308.0	4.90	19.80	253	261	1,905	516.0	Basler et al. (1960)
G6-T2	953	1,270	308.0	4.90	19.80	253	261	1,905	662.0	Basler et al. (1960)
G6-T3	635.0	1,270.0	308.0	4.90	19.8	253.0	261.0	1,905.0	787.0	Basler et al. (1960)
G7-T1	1,270.0	1,270.0	310.0	4.98	19.5	253.0	259.0	1,905.0	623.0	Basler et al. (1960)
G7-T2	1,270.0	1,270.0	310.0	4.98	19.5	253.0	259.0	1,905.0	645.0	Basler et al. (1960)
G8-T1	3,810.0	1,270.0	305.0	5.08	19.1	263.0	284.0	3,810.0	375.0	Basler et al. (1960)
G8-T2	1,905.0	1,270.0	305.0	5.08	19.1	263.0	284.0	1,905.0	445.0	Basler et al. (1960)
G8-T3	1,905.0	1,270.0	305.0	5.08	19.1	263.0	284.0	3,810.0	516.0	Basler et al. (1960)
G9-T1	3,810.0	1,270.0	305.0	3.33	19.1	307.0	288.0	3,810.0	213.0	Basler et al. (1960)
G9-T2	1,905.0	1,270.0	305.0	3.33	19.1	307.0	288.0	1,905.0	334.0	Basler et al. (1960)
G9-T3	1,905.0	1,270.0	305.0	3.33	19.1	307.0	288.0	3,810.0	352.0	Basler et al. (1960)
H1T2	1,905.0	1,270.0	459.0	9.98	24.8	745.0	703.0	1,905.0	3,450.0	Cooper et al. (1964)
H2T1	1,270.0	1,270.0	459.0	9.91	51.2	760.0	750.0	3,810.0	4,079.0	Cooper et al. (1964)
H2T2	635.0	1,270.0	459.0	9.91	51.2	760.0	750.0	3,810.0	5,004.0	Cooper et al. (1964)
B	1,200.0	1,200.0	240.0	4.50	12.0	490.0	491.0	1,200.0	745.0	Konishi (1965)
G2	1,150.0	440.0	200.0	8.00	30.0	431.0	412.0	1,150.0	824.0	Sakai et al. (1966)
G5	1,500.0	560.0	250.0	8.00	30.0	431.0	412.0	1,500.0	1,049.0	Sakai et al. (1966)
G6	687.0	560.0	250.0	8.00	30.0	431.0	412.0	687.0	1,176.0	Sakai et al. (1966)
G7	1,500.0	560.0	250.0	8.00	30.0	431.0	412.0	1,500.0	1,049.0	Sakai et al. (1966)
S3	577.0	477.0	101.0	3.20	10.5	317.0	272.0	577.0	198.0	Sakai et al. (1966)
US2/5	788.0	359.0	96.6	3.17	12.0	230.0	422.0	1,220.0	135.0	Kamtekar et al. (1972)
US3/5	788.0	359.0	96.1	2.70	12.0	257.0	422.0	1,620.0	90.0	Kamtekar et al. (1972)
TG14	305.0	305.0	76.2	0.97	3.1	219.0	305.0	610.0	25.4	Rockey and Skaloud (1972)
TG15	305.0	305.0	76.2	0.97	5.0	219.0	286.0	610.0	29.4	Rockey and Skaloud (1972)
TG16	305.0	305.0	76.2	0.97	6.5	219.0	337.0	610.0	31.8	Rockey and Skaloud (1972)
TG17	305.0	305.0	76.2	0.97	9.3	219.0	308.0	610.0	39.0	Rockey and Skaloud (1972)
TG18	305.0	305.0	76.2	0.97	13.0	219.0	304.0	610.0	50.5	Rockey and Skaloud (1972)
TG19	305.0	305.0	76.2	0.97	15.5	219.0	268.0	610.0	54.5	Rockey and Skaloud (1972)
TG22	305.0	305.0	76.2	2.03	6.5	229.0	337.0	610.0	78.5	Rockey and Skaloud (1972)
TG23	305.0	305.0	76.2	2.03	9.2	229.0	308.0	610.0	81.0	Rockey and Skaloud (1972)
TG24	305.0	305.0	76.2	2.03	13.0	229.0	307.0	610.0	96.0	Rockey and Skaloud (1972)
TG25	305.0	305.0	76.2	2.03	15.5	229.0	268.0	610.0	104.0	Rockey and Skaloud (1972)
STG1	551.0	279.0	127.0	2.00	7.9	255.0	275.0	551.0	60.0	Rockey and Skaloud (1972)
STG2	502.0	253.0	127.0	1.60	6.4	272.0	275.0	502.0	40.0	Rockey and Skaloud (1972)
STG4	498.0	251.0	102.0	1.25	6.4	246.0	275.0	498.0	35.0	Rockey and Skaloud (1972)
RTG1	305.0	305.0	76.0	1.27	4.5	244.0	275.0	610.0	40.0	Rockey and Skaloud (1972)
RTG2	305.0	305.0	76.0	1.27	4.7	244.0	275.0	610.0	41.0	Rockey and Skaloud (1972)
RTG4	254.0	254.0	76.0	0.95	4.7	259.0	275.0	508.0	24.0	Rockey and Skaloud (1972)
TS1/4	700.0	813.0	212.0	4.06	12.0	265.0	429.0	1,397.0	387.0	Kamtekar et al. (1974)
MSO	947.0	608.0	102.0	2.01	10.1	261.0	269.0	947.0	93.5	Evans et al. (1977)
SD1	594.0	594.0	250.0	2.00	12.0	276.0	212.0	2,670.0	129.0	Evans et al. (1979)
SD3	594.0	594.0	250.0	2.00	12.0	276.0	212.0	1,070.0	156.0	Evans et al. (1979)
TGV1-1	1,200.0	600.0	200.0	2.07	10.0	211.0	247.0	1,200.0	83.0	Rockey et al. (1981)
TGV1-2	600.0	600.0	200.0	2.07	10.0	211.0	247.0	1,200.0	111.0	Rockey et al. (1981)
TGV2-2	600.0	600.0	200.0	2.08	10.0	211.0	247.0	1,200.0	115.0	Rockey et al. (1981)
TGV3-2	600.0	600.0	200.0	2.01	10.0	211.0	247.0	1,200.0	113.0	Rockey et al. (1981)
TGV4	597.0	598.0	201.0	1.97	10.1	224.0	255.0	1,193.0	102.0	Rockey et al. (1981)

End of Table A2

Specimen	$b$ mm	$d$ mm	$b_f$ mm	$t_w$ mm	$t_f$ mm	$\sigma_{yw}$ N/mm <sup>2</sup>	$\sigma_{yf}$ N/mm <sup>2</sup>	$l_a$ mm	$V_{exp}$ kN <sup>2</sup>	Ref.
TGV5	595.0	598.0	201.0	1.98	10.0	232.0	252.0	1,189.0	105.0	Rockey et al. (1981)
TGV7-2	596.0	599.0	201.0	1.98	10.1	221.0	250.0	1,191.0	106.0	Rockey et al. (1981)
TGV10-1	595.0	599.0	200.0	1.91	10.0	219.0	284.0	1,189.0	102.0	Rockey et al. (1981)
TGV10-2	595.0	599.0	200.0	1.91	10.0	219.0	284.0	1,191.0	106.0	Rockey et al. (1981)
TGV11-2	597.0	599.0	200.0	1.91	10.0	220.0	211.0	1,194.0	102.0	Rockey et al. (1981)
S3/1	300.0	300.0	35.0	1.03	3.2	169.0	295.0	300.0	19.0	Adorasio (1982)
S4/1	345.0	351.0	39.5	1.07	3.2	169.0	295.0	345.0	21.0	Adorasio (1982)
S5/1	400.0	399.0	39.0	1.09	3.2	169.0	295.0	400.0	23.0	Adorasio (1982)
S2/1.5	375.0	249.0	39.5	1.05	3.2	169.0	295.0	375.0	15.5	Adorasio (1982)
S3/1.5	450.0	301.0	39.0	1.03	3.2	169.0	295.0	450.0	16.0	Adorasio (1982)
S4/1.5	522.0	352.0	39.1	1.10	3.3	169.0	295.0	522.0	13.0	Adorasio (1982)
LS1-PA	942.0	608.0	100.0	2.10	10.0	183.0	269.0	942.0	75.5	Evans and Tang (1983)
LS3-PA	947.0	608.0	100.0	2.46	10.1	201.0	283.0	947.0	103.0	Evans and Tang (1983)
MCS1-PB3	732.0	1,000.0	300.0	4.40	15.1	170.0	227.0	1,464.0	388.0	Evans (1984)
PA1	600.0	800.0	249.0	1.00	12.0	216.0	206.0	3,000.0	81.0	Tang and Evans (1984)
PA2	600.0	800.0	249.0	1.00	12.0	216.0	206.0	2,400.0	83.5	Tang and Evans (1984)
PA3	600.0	800.0	249.0	1.00	12.0	216.0	206.0	1,800.0	85.0	Tang and Evans (1984)
PB1	500.0	800.0	249.0	1.00	12.0	216.0	206.0	3,000.0	90.0	Tang and Evans (1984)
PB2	500.0	800.0	249.0	1.00	12.0	216.0	206.0	2,500.0	91.0	Tang and Evans (1984)
PC1	1,000.0	800.0	250.0	1.00	10.0	216.0	262.0	2,750.0	53.5	Tang and Evans (1984)
PC2	1,000.0	800.0	250.0	1.00	10.0	216.0	262.0	1,750.0	53.5	Tang and Evans (1984)
PD1	750.0	800.0	250.0	1.00	10.0	216.0	262.0	2,750.0	65.0	Tang and Evans (1984)
PD2	750.0	800.0	250.0	1.00	10.0	216.0	262.0	2,000.0	65.0	Tang and Evans (1984)
PD3	750.0	800.0	250.0	1.00	10.0	216.0	262.0	1,250.0	75.0	Tang and Evans (1984)
PC3	750.0	800.0	250.0	1.00	10.0	216.0	262.0	750.0	78.7	Tang and Evans (1984)
PB3	732.0	1,000.0	300.0	4.40	15.1	169.7	226.6	1,464.0	388.0	Evans (1986)
C-AC2	2,490.0	457.0	102.0	3.10	9.7	215.0	755.0	2,490.0	120.0	Carskaddan (1968)
C-AC4	2,515.0	457.0	127.0	4.30	16.3	236.0	783.0	2,515.0	245.0	Carskaddan (1968)
C-AC5	2,515.0	457.0	127.0	4.30	19.1	236.0	790.0	2,515.0	232.0	Carskaddan (1968)
TG3	1,000.0	1,000.0	200.0	2.50	16.4	200.0	281.0	1,000.0	190.0	Skaloud (1971)
TG3-1	1,000.0	1,000.0	200.0	2.50	16.4	200.0	281.0	1,000.0	190.0	Skaloud (1971)
TG4	1,000.0	1,000.0	200.0	2.50	20.2	200.0	281.0	1,000.0	219.0	Skaloud (1971)
TG4-1	1,000.0	1,000.0	200.0	2.50	20.1	200.0	281.0	1,000.0	207.0	Skaloud (1971)
TG5	1,000.0	1,000.0	200.0	2.50	29.7	200.0	281.0	1,000.0	308.0	Skaloud (1971)
TG5-1	1,000.0	1,000.0	200.0	2.50	29.7	200.0	281.0	1,000.0	300.0	Skaloud (1971)
PG1-1	600.0	600.0	200.0	4.10	12.5	343.0	257.0	600.0	373.0	Roberts and Shahabian (2000)
PG2-1	900.0	900.0	300.0	3.10	10.2	285.0	254.0	900.0	271.0	Roberts and Shahabian (2000)
PG3-1	900.0	600.0	200.0	3.20	10.1	282.0	264.0	900.0	202.0	Roberts and Shahabian (2000)
PG4-1	1,000.0	500.0	200.0	1.90	9.9	250.0	293.0	1,000.0	87.0	Roberts and Shahabian (2000)
CP1	747.0	500.0	100.0	2.04	8.0	246.0	256.0	747.0	168.0	Basler et al. (1960)