# Machine Learning for Design, Optimization and Assessment of Steel-Concrete Composite Structures: A Review

Xianlin Wang [a, b], Bozhou Zhuang [c, *], Danny Smyl [c], Haijun Zhou [a], M. Z. Naser [d,e]

[a] National Key Laboratory of Green and Long-Life Road Engineering in Extreme Environment, Shenzhen University, Shenzhen, 518060, China

[b] Department of Bridge Engineering, Tongji University, 1239 Siping Rd., Shanghai 200092, China

[c] School of Civil and Environmental Engineering, Georgia Institute of Technology, 756 W Peachtree St NW, Atlanta, GA 30308, USA

[d] School of Civil & Environmental Engineering and Earth Sciences (SCEEES), Clemson University, Lowry Hall, Clemson, SC 29631, USA

[e] Artificial Intelligence Research Institute for Science and Engineering (AIRISE), Clemson University, Riggs Hall, Clemson, SC 29631, USA

[*] Corresponding author. E-mail: bzhuang31@gatech.edu

**ABSTRACT**: Steel-concrete composite structures (SCCSs) combine the high compressive strength of concrete and tensile strength of steel to achieve optimal structural performance. However, the design of SCCSs is more complex than traditional reinforced concrete (RC) or steel structures due to the steel-concrete composite effects. In recent years, machine learning (ML) has been increasingly applied to SCCSs. However, there have been no related reviews on this topic and this literature gap serves as the motivation for this review. This paper presents the first extensive literature review for ML applications in the design, optimization and assessment of SCCSs. A total of 194 references are collected with most of them are directly related to the ML applications in SCCSs. We discussed ML workflows and models applied for SCCSs, and summarized applications of ML across different SCCS components, including mechanical connectors, steel-concrete interfacial bonding, steel-concrete composite beams, slabs, columns, and walls. The challenges and future research directions are also highlighted. This review provides a valuable reference for researchers and engineers working on the research and development of ML in SCCSs.

**Keywords:** Machine learning; Steel-concrete composite structures; Composite effects; Structural design, optimization and assessment; Mechanical connectors

## Abbreviations

| | | | |
|---|---|---|---|
| ABC | Accelerated bridge construction | KRR | Kernel ridge regression |
| ACC | Axial compression capacity | LACE | Local interpretable model-agnostic explanations |
| AdaBoost | Adaptive boosting | LASSO | Least absolute shrinkage and selection operator |
| AE | Acoustic emission | LCA | Life cycle assessment |
| AGWO | Augmented grey wolf optimizer | LDA | Linear discriminant analysis |
| AI | Artificial intelligence | LDB | Lateral-distortional buckling |
| ANFIS | Adaptive neuro-fuzzy inference system | LightGBM | Light gradient boosting machine |
| ANN | Artificial neural network | LIME | Local interpretable model-agnostic explanations |
| ANOVA | Analysis of variance | LM | Levenberg Marquardt |
| ARIMA | Autoregressive integrated moving average | LR | Logistic regression/linear regression |
| ATDF | Auto-tuning deep forest | LSSVM | Least squares support vector machine |
| BBO | Biogeography-based optimization | LSTM | Long short-term memory |
| BCMO | Balancing composite motion optimization | LWC | Lightweight concrete |
| BET | Bagged ensemble trees | M5P | M5 model trees |
| BMA | Bayesian model averaging | MAE | Mean absolute error |
| BNB | Bernoulli Naive Bayes | MAPE | Mean absolute percentage error |
| BR | Bayesian ridge | MCMC | Markov chain Monte Carlo |
| CART | Classification and regression tree | MCS | Monte Carlo simulation |
| CatBoost | Category boosting | MGGP | Multigene genetic programming |
| CFSTs | Concrete-filled steel tubes | MLR | Multiple linear regression |
| CG | Concrete grout | MMD | Maximum mean discrepancy |
| CNN | Convolutional neural network | MPMR | Minimax probability machine regression |
| CRC | Crumb rubber concrete | MSE | Mean square error |
| CS | Cuckoo search | MVFT | Modified Verbund-Fertigteil-Träger |
| CycleGANs | Cycle-consistent generative adversarial networks | NGBoost | Natural gradient boosting |
| DA | Dual annealing | NLTK | Natural language toolkit |
| DANN | Domain-adversarial neural networks | NMR | Nonlinear multi-regression |
| DBSCAN | Density-based spatial clustering of applications with noise | NSC | Normal strength concrete |
| DE | Differential evolution | NUS | Non-uniform shrinkage |
| DF | Deep forest | OSS | One-step secant |
| DNN | Deep neural network | PBL | Perforbond strip |
| DRN | Deep residual network | PCA | Principal component analysis |
| DT | Decision tree | PCE | Polynomial chaos expansions |
| EBT | Ensemble boosted tree | PDP | Partial dependence plots |
| ECC | Engineered cementitious composites | POS | Part-of-speech |
| EDA | Exploratory data analysis | PRF | Pseudo-random forest |
| EGWO | Enhanced grey wolf optimizer | PSO | Particle swarm optimization |
| ELM | Extreme learning machine | RAC | Recycled aggregate concrete |
| EN | Elastic net | RBF | Radial basis function |
| ET/ExtraTrees | Extremely randomized trees | RBFNN | Radial basis function neural network |
| FCM | Fuzzy C-means | RCGA | Real coded genetic algorithm |
| FEA | Finite element analysis | ReLU | Rectified linear unit |
| FEM | Finite element method | RF | Random forest |
| FFA | Firefly algorithm | RMSE | Root mean square error |
| FL | Fuzzy logic | RMSLE | Root mean squared logarithmic error |
| GA | Genetic algorithm | RR | Ridge regression |
| GANs | Generative adversarial networks | RSM | Response surface method |
| GBDT | Gradient boosting decision tree | RT | Regression tree |
| GBM | Gradient boosting machine | SA | Simulated annealing |
| GD | Gradient descent | SCA | Sine-cosine algorithm |
| GEP | Gene expression programming | SCCS | Steel-concrete composite structures |
| GMDH | Group method of data handling | SFRC | Steel fiber reinforced concrete |
| GP | Gaussian process | SHAP | SHapley additive explanations |
| GPR | Gaussian process regression | SHG | Second-harmonic generation |
| GRNN | General regression neural network | SHM | Structural health monitoring |
| GUI | Graphical user interface | SIFT | Scale-invariant feature transform |
| GWO | Grey wolf optimization | SMA | Slime mould algorithm |
| HGBDT | Histogram-based gradient boosting decision tree | SMBO | Sequential model-based optimization |
| HHO | Harris hawks optimization | SMOTE | Synthetic minority oversampling technique |
| HOG | Histogram of oriented gradients | SSA | Salp swarm algorithm |
| HSC | High strength concrete | SVM | Support vector machine |

| | | | |
|---|---|---|---|
| IAGA | Improved adaptive genetic algorithm | TAMO | Threshold accepting with a mutation operator |
| ICA | Competitive imperialism algorithm | TGANs | Tabular GANs |
| ICE | Individual conditional expectation | *t*-SNE | *t*-distributed stochastic neighbor embedding |
| IEPSO | Improved evolutionary particle swarm optimization | UHPC | Ultra-high performance concrete |
| IPSO | Improved particle swarm optimization | VAE | Variational autoencoders |
| IQR | Interquartile range | WAE | Wasserstein autoencoders |
| IWO | Invasive weed optimization | WOA | Whale optimization algorithm |
| KGPR | Kernel-based Gaussian process regression | XGBoost | eXtreme gradient boosting |
| *k*-NNs | *k*-nearest neighbors | YOLO | You only look once |

## 1. Introduction

Steel and concrete are two ubiquitous construction materials in structural engineering. As illustrated in **Fig. 1**, a structural component made of steel has high tensile strength under a tensile force *T* but is susceptible to buckling under a compression force *C*. Conversely, concrete has a high compressive strength but is susceptible to cracking under tension. To fully take advantage of both materials, steel-concrete composite structures (SCCSs) have been proposed. As shown in **Fig. 1**, steel can be placed in the tensile zone and concrete is positioned in the compressive region under a bending moment, *M*. Mechanical connectors are welded to connect the steel and concrete and provide shear and uplift resistance at the interface. Several types of mechanical connectors have been developed, with common examples being the headed stud connectors [1] and the perforbond strip (PBL) connectors [2]. The bonding between steel and concrete also contributes to the composite effects of SCCSs [3].

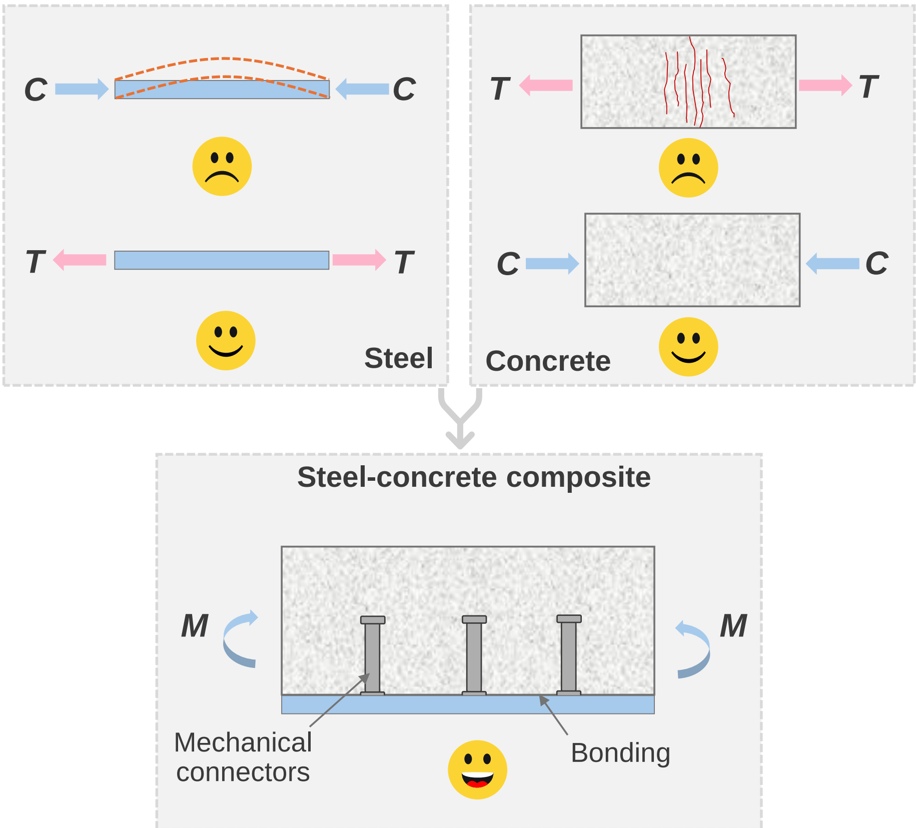


**Fig. 1.**Steel-concrete composite effects.

SCCSs have been widely applied in infrastructure, including buildings, bridges, tunnels, and nuclear facilities. Various SCCS configurations, such as steel-concrete composite slabs, beams, columns, and walls, have been proposed, designed, and constructed. The primary challenge in designing SCCSs is the composite interaction between steel and concrete components. For example, in the hogging moment regions of a continuous steel-concrete composite beam, the concrete slab is in tension while the steel beam is in compression. Composite connections provided by shear connectors need be released to mitigate cracking of the concrete slab [4][5][6]. In a steel-reinforced concrete column, the load-bearing capacities are provided by (1) the steel and concrete, (2) shear connectors, and (3) steel-concrete interface bonding. Calculating the strain distributions and load transfer of the steel-reinforced concrete column is still being investigated [7]. Therefore, the design of SCCSs is more challenging compared to traditional reinforced concrete (RC) or steel structures.

In recent years, machine learning (ML) and artificial intelligence (AI) have been increasingly applied in structural engineering. ML models are capable of learning patterns from the training dataset and apply the learned pattern to make predictions on the unseen samples. The application of ML in structural engineering has demonstrated advantages in structural design and construction automation [8][9], smart damage detection [10][11], structural health monitoring (SHM) [12][13], among others. There have been several literature reviews on ML applications on structural engineering [14][15][16], concrete properties [17][18], bridge design and inspection [19], and smart buildings [20][21]. However, there is an absence of a comprehensive literature review focusing on the ML applications on SCCSs. This gap serves as the primary motivation for this review. The objective of this review is to (1) outline typical ML workflow and models for SCCSs, (2) summarize recent applications of ML in the design, optimization, and assessment of SCCSs, and (3) discuss the challenges and future directions.

Therefore, this paper focuses on studies that apply ML techniques for SCCSs. A total of 194 references are included, and 169 references are directly related to the ML applications of SCCSs. The review methodology and bibliometric analysis are presented in Section 2. A typical ML workflow for SCCSs is proposed and discussed in Section 3. In Section 4, the applications of ML on mechanical connectors, steel-concrete interfacial bonding, steel-concrete composite beams, slabs, columns, and walls, are discussed in detail. The challenges and future directions are discussed in Section 5, and conclusions are listed in Section 6. To the best of the authors' knowledge, this is the first literature review that provides a comprehensive summary of ML applications in SCCSs.

## 2. Review methodology and bibliometric analysis

This study reviews 194 references, and 169 references are directly related to the ML applications of SCCSs. Literature is sourced from notable repositories such as Google Scholar (https://scholar.google.com/), the Web of Science database (https://www.webofscience.com/), and the ScienceDirect database (https://www.sciencedirect.com/). As shown in **Fig. 2**a, the earliest research on ML applications in SCCSs dates to the year 2009. The number of cumulative publications shows a significant increase from the year 2018. **Fig. 2**b shows the top ten journals in which the collected literature was published, with *Structures* (23.7%), *Engineering Structures* (21.5%), *Construction and Building Materials* (12.9%), *Journal of Building Engineering* (9.7%), and *Buildings* (7.5%) being the top five in the list.

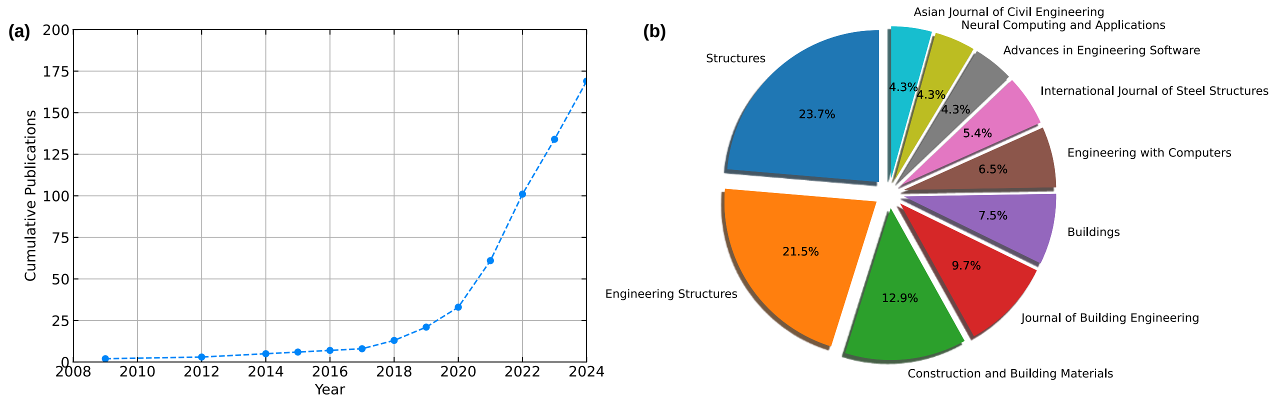


**Fig. 2.** Literature analysis: (a) cumulative publications over time; (b) top 10 journals by publication source.

A bibliometric analysis was conducted by using VOSviewer to examine keyword co-occurrence and connections. A total of 43 keywords were extracted from titles and abstracts. These keywords were grouped into five clusters based on a modularity optimization technique that identifies groups of items with dense connections, as shown in **Fig. 2**a. Each cluster is color-coded, and the font size reflects keyword occurrences. The most frequently occurring keywords in each cluster are “bond strength”, “beam”, “shear connector”, “column”, and “ML model”. In **Fig. 2** (b), a color map represents the publication years of keywords. Noteworthy keywords such as “beam”, “stud”, “composite slab”, “SHAP (SHapley Additive ExPlanations)”, and “fatigue life” are found to be more prevalent after the year 2023.

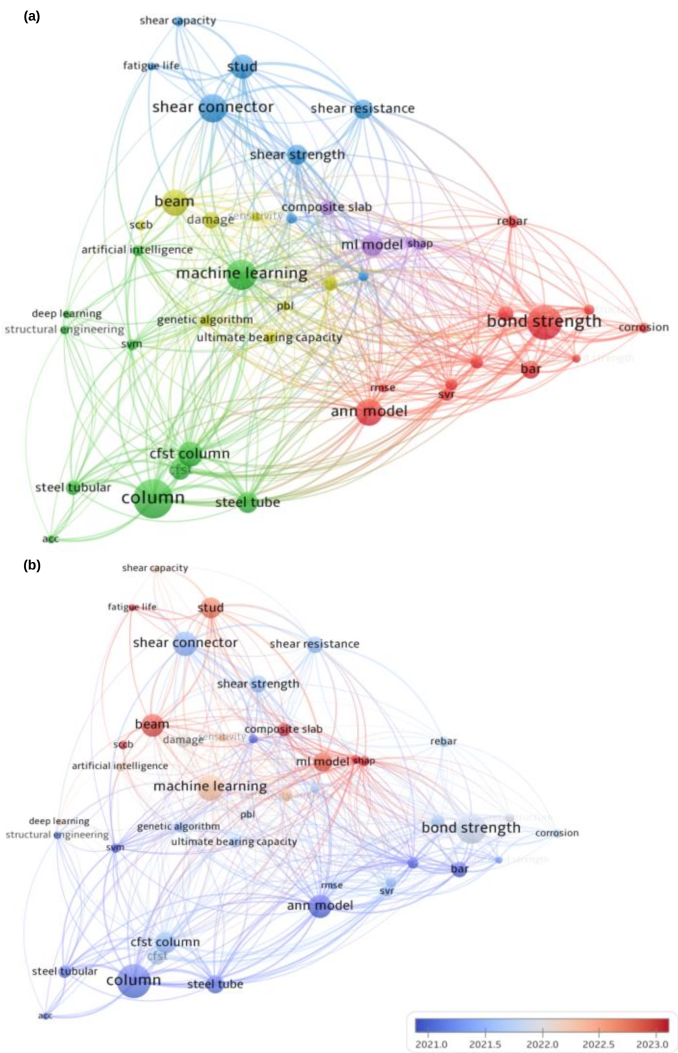


**Fig. 3.** Co-occurrence analysis of keywords: (a) keywords and cluster; and (b) keywords and their publication year.

## 3. Machine Learning Workflow

### *3.1 ML-based framework for design, optimization and assessment of SCCSs*

ML has strong capabilities in learning complex data related to the design, optimization and assessment of SCCSs. **Fig. 4** shows a ML-based framework for the SCCSs proposed by the authors. The complete workflow includes five key modules as follows.

**Module 1: Domain knowledge acquisition.** Before applying ML, one should determine the objectives of ML and has basic domain knowledge in SCCSs. This knowledge can be acquired by reading literature, conducting experiments and/or simulations. Given the complexity of SCCSs, experimental studies are the primary method to evaluate the structural performance of SCCSs. The design dimensions, structural configurations, composite effects and failure mechanisms of SCCSs should be understood first to determine ML objectives and construct the database.

**Module 2: Database construction.** The database can be collected from experiments, simulations, field measurements, or literature. Various features for SCCSs, such as geometric dimensions, material properties, and condition parameters, should be considered. Normalization/standardization for numerical features and one-hot encoding for categorical features are commonly used. Missing data and features should also be considered in this step. The database is then split into training, validation and testing sets.

**Module 3: ML model training and tuning.** Standalone and ensemble ML models can be developed for SCCSs using the constructed database. The ML models, training strategies, and loss functions should be selected accordingly based on the material combinations and structural details of SCCSs. The hyperparameters of ML models can be tuned by using grid search, sequential model-based optimization, *k*-folder cross validation, among others. The optimal model should be selected based on the lowest loss on the validation dataset.

**Module 4: Performance evaluation and interpretive analysis.** The accuracy and generalization of the ML models need to be tested on the testing set and validated with existing design criterion. Furthermore, interpretive analysis may be performed using techniques such as permutation feature importance and SHapley Additive ExPlanations (SHAP). The interpretive analysis helps to understand the feature importance and thus offers ML insights for SCCS design.

**Module 5: Cloud deployment and application.** The goal of developing ML techniques is to automate the design, optimization and assessment of SCCSs. ML models do not directly provide explicit formulas that engineers can readily use. Therefore, the trained ML models can be deployed with a graphical user interface (GUI) and thus engineers and designers can apply ML to SCCSs without the need for coding.

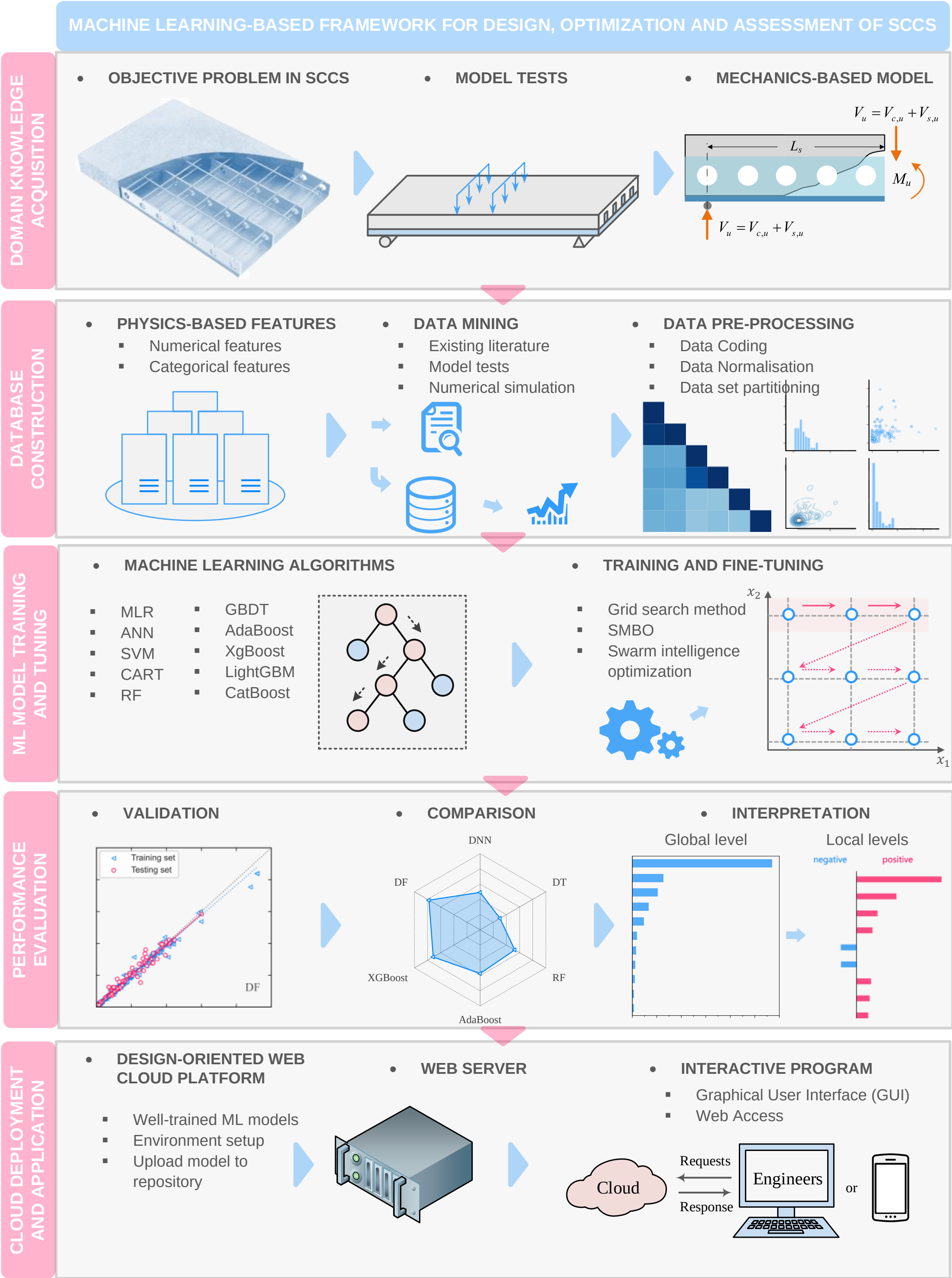


**Fig. 4** A schematic illustration of a ML-based framework for design, optimization and assessment of SCCSs (adapted from [22][23][24]). Note: MRL: multiple linear regression; ANN: artificial neural network; SVM: support vector machine; CART: classification and regression tree; RF: random forest; GBDT: gradient boosting decision tree; AdaBoost: adaptive boosting; XGBoost: eXtreme gradient boosting; LightGBM: light gradient boosting machine; CatBoost: category boosting; SMBO: sequential model-based optimization; GUI: graphical user interface.

### *3.2 Determination of objectives*

It is essential to first determine the tasks and objectives of applying ML for SCCSs. The common ML objectives for SCCSs are summarized in **Table 1**. Regression is widely used to predict continuous value(s) related to structural behavior of SCCSs, such as the shear resistance of headed stud connectors, bending capacity of composite beams, and the long-term performance of composite slabs. Classification involves assigning categories or labels to input data based on features. In the SCCS, classification can be used to identify and assess the condition of structures. Clustering helps group similar structural behaviors or failure patterns without predefining categories. In SHM for SCCSs, clustering can be used to identify patterns in vibration data to group structures with similar performance metrics or statuses [25]. Anomaly detection plays a critical role in identifying early signs of structural failure or degradation by monitoring parameters such as strain, temperature, or vibration over time. Computer vision techniques are applied for analyzing visual data such as images of structures. Convolutional neural networks (CNNs) can automatically analyze images from drones or inspections [26] to identify and quantify damage in SCCSs [31]. Time series forecasting is used to predict the future behavior of structures based on historical performance data. This is valuable in applications such as predicting the remaining service life of a structure, forecasting the progression of deflections and cracks in structural members, and anticipating long-term maintenance needs. Algorithms such as long short-term memory (LSTM) and autoregressive integrated moving average (ARIMA) can capture the evolving behavior of structures with time as the variable under cyclic loads or environmental conditions.

**Table 1**

Typical ML objectives in the field of SCCS.

| Objective | Descriptions | Algorithms | SCCS applications | Representative reference(s) |
|---|---|---|---|---|
| **Regression** | Predicts a continuous output based on input variables | SVM, ANN, AdaBoost, XGBoost, LightGBM | Predicting load-bearing capacity of composite beams | [27][28][29] [22] |
| **Classification** | Assigns predefined categories to the input data | LR, SVM, DT, RF, ANN | Failure mode classification in SCCSs based on test data | [2] |
| **Clustering** | Groups similar data points into clusters without predefined labels | *k*-means, DBSCAN, hierarchical clustering, Gaussian mixture models | Grouping similar structural damage patterns of SCCSs | [25] |
| **Anomaly Detection** | Identifies rare or unusual data points that differ significantly from the normal data | Isolation forest, one-class SVM, Autoencoders, *k*-NN | Detecting cracks or faults in steel-concrete composite bridges | [30] |
| **Computer Vision** | Analyzes and interprets visual data from images or videos | CNN, HOG, SIFT, YOLO | Automated crack detection in SCCSs | [31] |
| **Time Series Forecasting** | Predicts future values based on previously observed data | ARIMA, LSTM, prophet, exponential smoothing | Predicting future loads or deformations in SCCSs, reproducing the load-displacement curve of SCCSs | [32] |

Note: SCCS: steel-concrete composite structure; SVM: support vector machine; ANN: artificial neural network; AdaBoost: adaptive boosting; XGBoost: eXtreme gradient boosting; LightGBM: light gradient boosting machine; LR: logistic regression; DT: decision tree; RF: random forest; DBSCAN: density-based spatial clustering of applications with noise; *k*-NN: *k*-nearest neighbors; CNN: convolutional neural networks; HOG: histogram of oriented gradients; SIFT: scale-invariant feature transform; YOLO: you only look once; ARIMA: autoregressive integrated moving average; LSTM: long short-term memory.

### *3.3 Data Collection*

Data collection is a critical step of the ML approach. Data can be obtained from a variety of sources for SCCSs. First, experimental data can be collected from real-world tests or SHM systems by using sensors and gauges. Second, data can be extracted from published research papers, reports, and thesis. Third, data can be collected by conducting finite element analysis (FEA), or other numerical modeling.

### *3.4 Data Pre-processing*

Data pre-processing is crucial to ensure the quality and consistency of the data before applying ML algorithms. Proper data pre-processing improves model performance by ensuring that the dataset is clean, complete, and accurate. Key pre-processing steps in SCCS applications include addressing missing data, removing outliers, normalizing features, and transforming feature dimensions.

### 3.4.1 Missing data

Missing data can arise from sensor malfunctions, incomplete experiments, or data transmission errors. For SCCS applications, handling missing data typically involves three techniques: imputation, deletion, and interpolation. In imputation, missing values can be filled with estimated data from the mean, median, or a surrogate model to predict missing values. Deletion involves removing specific data give that it is unlikely to impact the dataset. Lastly, interpolation estimates missing data based on the trends in adjacent data points.

### 3.4.2 Data cleaning

Data cleaning involves identifying and removing errors, inconsistencies, or noise in the dataset. For example, outliers may occur due to sensor errors, extreme loading, or experimental anomalies. Identifying and removing these outliers ensures the ML model is not biased by extreme values. Common techniques to identify outliers include Z-scores and the interquartile range (IQR) method. Besides, structural data can be noisy due to environmental or instrumentation factors. Smoothing or filtering techniques can be applied to address noise and improve data quality.

## 3.5 Feature Engineering

### 3.5.1 Exploratory data analysis

Exploratory data analysis (EDA) is to understand the dataset before applying ML models. EDA offers insights into central tendencies, dispersion, and data distribution by analyzing the mean, median, standard deviation, minimum, and maximum values of features [22][27]. There are two ways to conduct EDA visualizations:

(1) Univariate visualization of each field in the raw dataset with summary statistics. For instance, the data distribution can be visualized by density histograms, bar plots (i.e., showing frequency or proportion), and box/violin plots to represent the five-number summary (i.e., minimum, first quartile, median, third quartile, and maximum), as shown in **Fig. 5**.

(2) Bivariate or multivariate visualizations and summary statistics that allow assessment of the relationship between variables and the target. For instance, heat maps (i.e., color-coded data representation) and multivariate charts are used to explore factor-response relationships as shown in **Fig. 6**.

### 3.5.2 Feature selection

Feature selection reduces the dimensionality of the data, improve model performance, and prevent overfitting by focusing on the most informative features that influence structural behavior. There are two primary types of features for SCCS, individual and combined features. Individual features directly affect the target output. Examples include the material properties (e.g., strength and elastic modulus), geometrical properties (e.g., dimensions, cross-sectional area, and moment of inertia), and structural details (e.g., connection types, reinforcement patterns, and joint types). Combined features integrate individual features to better reflect underlying physical principles. Examples include the tensile capacity of stud shank ($f_{su}A_s$) in the shear resistance of headed studs [27], and the frequency response function calculated in the damage identification of steel-concrete composite beams [33].

Accordingly, there are three feature selection techniques. First, features can be selected based on their correlation with the target variable (e.g., resistance or deflection), as well as inter-correlations among features. Highly correlated features are more likely to have a similar influence on the prediction. For example, concrete compressive strength and steel yield strength might be highly correlated when analyzing the axial compression capacity of concrete-filled steel tubes (CFST). Second, feature importance can be analyzed by specific ML models, such as decision trees, random forests, and gradient boosting. These ML models assign feature importance scores which helps in identifying influential features. Third, features inspired by physical principles or domain knowledge may better represent the characteristics of SCCSs as they have been well-understood from empirical studies or mechanics-based models.

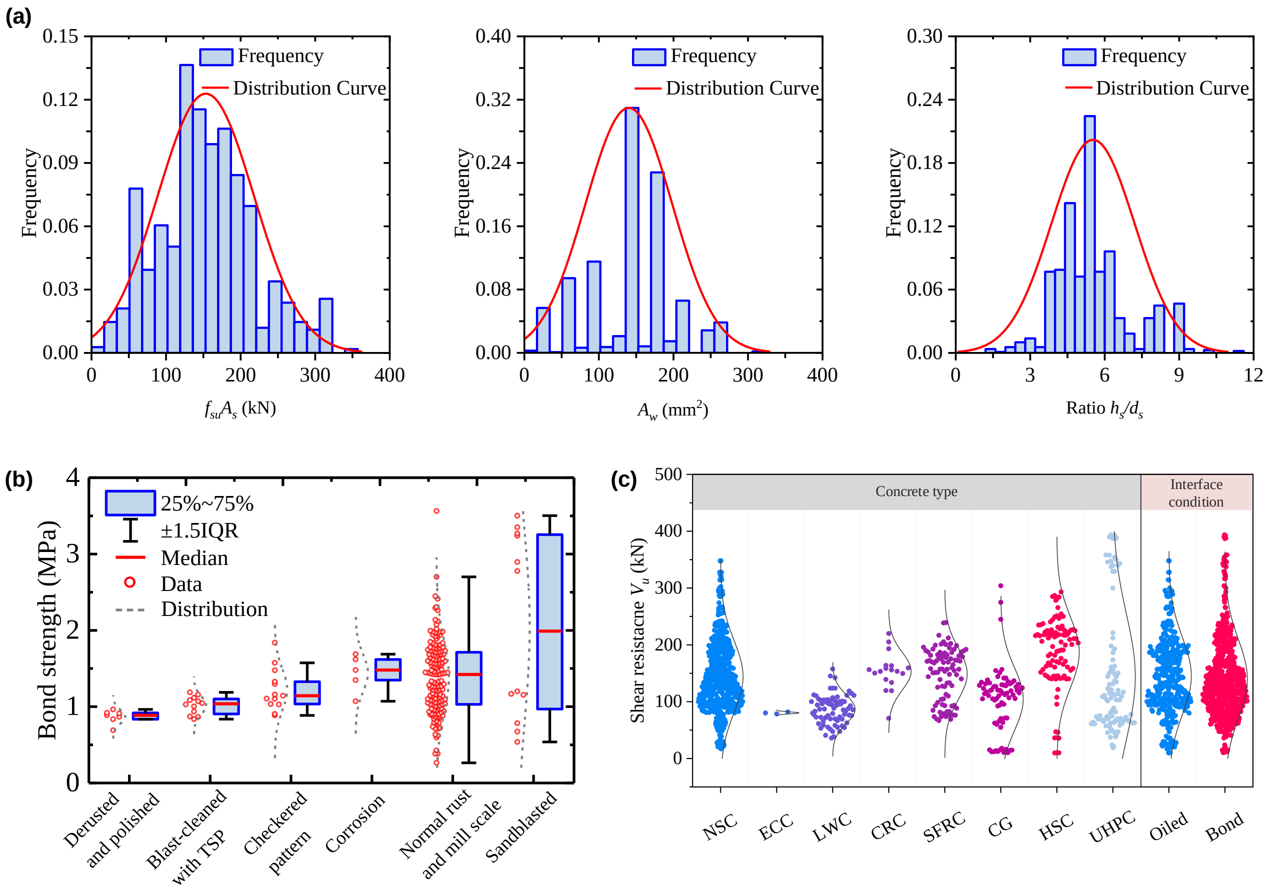


**Fig. 5** Univariate visualizations for exploratory data analysis: (a) histogram (adapted from [27]); (b) box plots (adapted from [22]) ; (c) violin plots (adapted from [27]). Note: IQR: interquartile range; NSC: normal strength concrete; ECC: engineered cementitious composites; LWC: lightweight concrete; CRC: crumb rubber concrete; SFRC: steel fiber reinforced concrete; CG: concrete grout; HSC: high strength concrete; UHPC: ultra-high performance concrete.

### *3.5.3 Feature transformation and dimension reduction*

#### *3.5.3.1 Normalization and standardization*

In structural engineering, datasets may include numerical variables with different units and scales. Therefore, normalization and standardization ensure that the features are on a comparable scale to balance the influence of each feature on the model. Normalization scales the data to a specific range, typically [0, 1] or [-1, 1], by adjusting the values to be proportional to their original range. The choice of range depends on the target loss function used for optimization. The min-max scaling method is commonly utilized by applying

$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where $x_{norm}$ and $x_i$ are the normalized and original values, respectively; $x_{min}$ and $x_{max}$ are the minimum and maximum values, respectively.

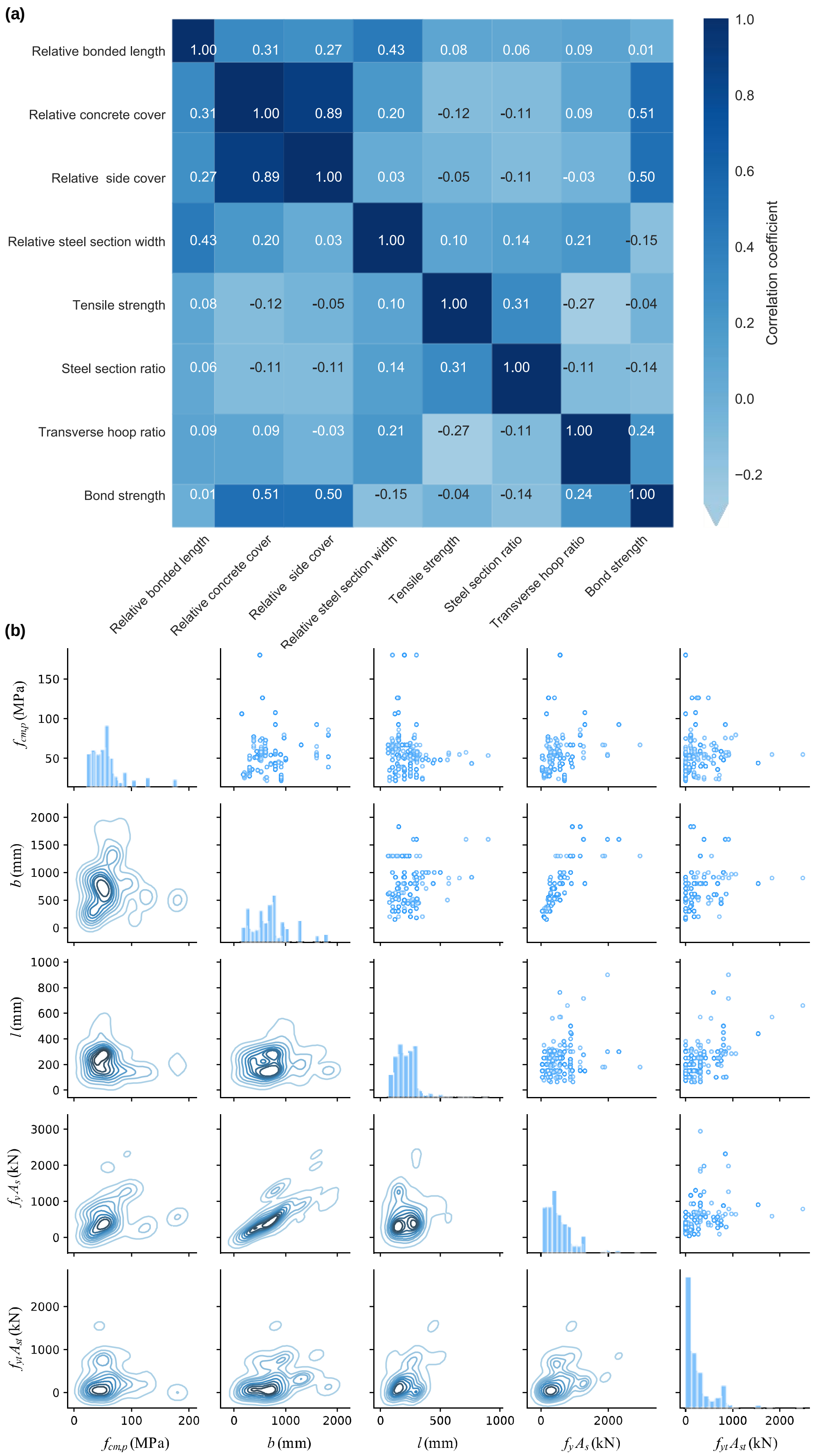


**Fig. 6** Bivariate visualizations for exploratory data analysis: (a) heat map (adapted from [22]); (b) multivariate chart (adapted from [23]).

Standardization transforms data to have a mean of zero and a standard deviation of one. This method is particularly useful when the feature in the database follows a Gaussian distribution. The standardization can be expressed by

$$x_{std} = \frac{x_i - \mu}{\sigma} \tag{2}$$

where $x_{std}$ is the standardized feature value; $\mu$ and $\sigma$ are the mean and standard deviation of the feature in the training set, respectively.

#### *3.5.3.2 Categorical features*

Categorical features refer to variables that represent discrete categories rather than continuous numerical values. There are two types of categorical features, nominal and ordinal features. Nominal features are without inherent order and ranking. For instance, different material types, such as normal strength concrete (NSC), steel fiber reinforced concrete (SFRC) [34,35], and ultra-high performance concrete (UHPC) [36][37], have no ordinal relationship between them [27]. Ordinal features are categorical features with a clear order but no numerically consistent difference between the categories. For instance, damage levels (e.g., minor, moderate, and severe damage) are ordered but not numerically spaced, as discussed in [10]. Categorical features must be converted into their numerical form for ML. Common converting techniques include one-hot encoding, label encoding, target encoding, and binary encoding. One-hot encoding converts categorical features into binary columns, where each unique category is represented by a separate column, and "1" indicates the presence of that category. For example, if the categorical feature is "concrete type" with categories NSC, SFRC, and UHPC, one-hot encoding will create three binary columns, each indicating the presence (1) or absence (0) of one of the materials [27]. Label encoding assigns an integer to each category. For example, for the "damage level" feature with categories of "minor", "moderate" and "severe", label encoding would assign 0, 1, and 2, respectively, to reflect the increasing severity of damage. Target encoding (or mean encoding) replaces categorical features with the mean of the target output for each category. Last, binary encoding is a hybrid method that converts categories into binary code and then splits each binary digit into separate feature columns.

#### *3.5.3.3 Dimensionality reduction*

Dimensionality reduction reduces the number of features in a dataset while preserving key information, which is crucial when handling high-dimensional data. It improves model performance by simplifying the data, mitigating overfitting (especially with limited data), addressing ill-posed problems, and enhancing computational efficiency by focusing on critical features. Key techniques for dimension reduction include principal component analysis (PCA), linear discriminant analysis (LDA), and *t*-distributed stochastic neighbor embedding (*t*-SNE). PCA is a widely used method that transforms features into uncorrelated components to reduce dimensionality while retaining variance. LDA is a supervised method that finds feature combinations that best separate classes in classification problems. *t*-SNE is a non-linear technique for visualizing high-dimensional data in 2D or 3D to reveal patterns like clusters or outliers.

### *3.6 Model selection and implementation*

The ML models applied for SCCSs can be divided into standalone, hybrid, and ensemble models. The selection of the model depends on the problem's complexity, available data, and performance metrics. The comparisons between the standalone, hybrid, and ensemble models are outlined in **Table 2**.

**Table 2**

Comparison of standalone, hybrid, and ensemble models for SCCS applications.

| Model type | Advantages | Disadvantages | Application scenarios | Representative algorithms |
|---|---|---|---|---|
| **Standalone Model** | Simple, easy to interpret, efficient for small datasets | May underperform on complex, non-linear problems | Simple tasks like linear regression or basic classification tasks | ANN, SVM, CART, GEP |
| **Hybrid Model** | Combines strengths of different approaches, improves accuracy, allows global search and avoid local minima | More complex to implement, high computational cost, scalability issues | Complex tasks where multiple variables (e.g., material properties, geometry, load conditions) interact in complex ways, like structural optimization tasks | PSO-ANN, GA-ANN, PSO-ANFIS, GA-ANFIS |
| **Ensemble Model** | High accuracy, reduces overfitting, handles complex problems well | Computationally expensive, difficult to interpret | High-stakes predictions like shear resistance, damage detection, and material performance | AdaBoost, RF, LightGBM, XGBoost |

Note: ANN: artificial neural network; SVM: support vector machine; CART: classification and regression tree; GEP: gene expression programming; PSO: particle swarm optimization; GA: genetic algorithm; ANFIS: adaptive neuro-fuzzy inference system; GA: genetic algorithm; AdaBoost: adaptive boosting; RF: random forest; LightGBM: light gradient boosting machine; XGBoost: eXtreme gradient boosting.

### *3.6.1 Standalone model*

#### *3.6.1.1 Artificial neural network*

An artificial neural network (ANN) is a ML model that mimics the way that the human brain processes information [38], as illustrated in **Fig. 7**a. It is composed of layers of interconnected "neurons" (also called nodes) that process data by adjusting the weights of these connections based on back-propagation algorithms. In the forward direction, each neuron can be derived by computing a weighted sum of neurons in the previous layer followed by the addition of a bias term, which can be expressed by

$$z_j^{(l)} = \sum_{i=1}^{n} w_{ji}^{(l)} x_i^{(l-1)} + b_j^{(l)} \tag{3}$$

where $z_j^{(l)}$ is the pre-activation value of neuron *j* in layer *l*; $w_{ji}^{(l)}$ is the weight connecting neuron *i* in the previous layer (*l*-1) to neuron *j* in layer *l*; $x_i^{(l-1)}$ is the output from neuron *i* in the previous layer (*l*-1) after activation; $b_j^{(l)}$ is the bias term for neuron *j* in layer *l*. Subsequently, an activation function is applied to introduce non-linearity to the model. Common activation functions include Sigmoid, ReLU (i.e., rectified linear unit), hyperbolic tangent (i.e., Tanh), Softmax, among others. During training, ANN uses backpropagation to update the weights and biases to minimize the loss function.

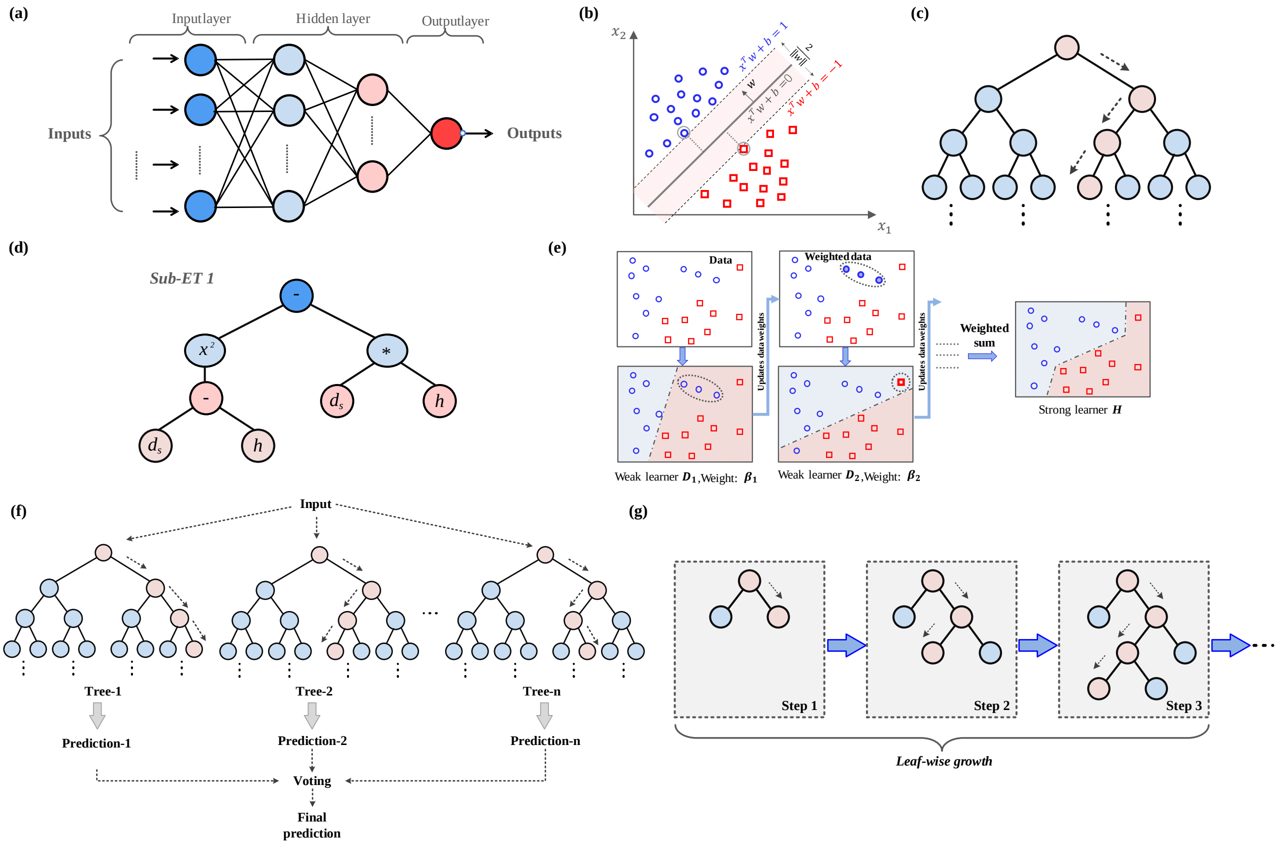


**Fig. 7** Schematic diagram of various ML models:(a) ANN; (b) SVM; (c) CART; (d) GEP; (e) AdaBoost; (f) RF; (g) LightGBM (adapted from [22]). Note: ANN: artificial neural network; SVM: support vector machine; CART: classification and regression tree; GEP: gene expression programming; AdaBoost: adaptive boosting; RF: random forest; LightGBM: light gradient boosting machine.

### *3.6.1.2 Support vector machine*

Support vector machine (SVM) is another supervised ML algorithm used for classification and regression tasks [39]. SVM fits the optimal hyperplane $\mathbf{w}^{\mathrm{T}}\boldsymbol{x}+b=0$ within a tolerance margin to minimize the error for points outside this margin, as shown in **Fig. 7**b. The goal of SVM is to minimize the model's complexity (i.e., the norm of the weight vector $\mathbf{w}$) while keeping deviations from actual values within a threshold $\varepsilon$, except for a few outliers. This leads to the following optimization problem expressed as

$$\text{Minimize} \quad \left\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)\right\} \tag{4}$$

$$\text{Subject to} \quad \begin{cases} y_i - \mathbf{w}^{\mathrm{T}}\phi(x_i) - b \leqslant \varepsilon + \xi_i \\ \mathbf{w}^{\mathrm{T}}\phi(x_i) + b - y_i \leqslant \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geqslant 0, i = 1, \ldots, n \end{cases} \tag{5}$$

where $\mathbf{w}$ is the weight vector term and $b$ is the bias term; $\xi_i, \xi_i^*$ are slack variables that allow for points above and below the $\varepsilon$-margin, respectively; $C$ is the regularization parameter that controls the trade-off between minimizing the model complexity and deviations from the $\varepsilon$-margin.

### *3.6.1.3 Classification and regression tree*

As shown in **Fig. 7**c, classification and regression tree (CART) creates a tree-like model of decisions based on the input features. It splits the data at each node and eventually assigns a label (for classification) or predicts a continuous value (for regression) as the output [40]. The objective of CART is to recursively partition the data into subsets that are as homogeneous as possible. For classification, the algorithm selects the best feature and threshold based on Gini impurity or entropy, while for regression, it uses the mean squared error (MSE) as the loss function.

##### *3.6.1.4 Gene expression programming*

Gene expression programming (GEP) is an algorithm that represents solutions as computer programs or mathematical expressions. It mimics the process of natural selection, mutation, and reproduction in biological evolution. In GEP, solutions are depicted as trees where internal nodes represent operators (e.g., +, -, *, sin, cos), and leaf nodes represent operands (e.g., constants or input variables), as shown in **Fig. 7**d. It helps discover mathematical models that best describe relationships between inputs and outputs.

### *3.6.2 Hybrid model*

Standalone models can provide accurate and reliable predictions in certain tasks, however, they have inherent limitations. For example, improper weights and biases initialization may lead the ANN to a local minimum rather than the global minimum [41]. Metaheuristic optimization algorithms, such as particle swarm optimization (PSO), genetic algorithm (GA), and grey wolf optimization (GWO), can help overcome these limitations. PSO is inspired by the social behavior of swarms (e.g., birds swarms) and efficiently explores solution spaces by updating positions based on individual and collective experiences. This makes it suitable for optimizing ANN parameters [42]. GA mimics the natural selection processes (e.g., crossover, mutation, and selection) to yield better solutions over generations based on Darwin's theory of evolution and natural genetics [43]. GWO is introduced by Mirjalili et al. [44] and it mimics grey wolf hunting hierarchies. It is widely used for solving optimization problems, including parameter tuning for ML models and structural design optimization.

Metaheuristic algorithms have been seen to optimize standalone models. For instance, the biogeography-based optimization (BBO) [45], PSO and its variants [46][45][47][48][49], competitive imperialism algorithm (ICA) [48], balancing composite motion optimization (BCMO) [50][51][52], one-step secant (OSS) algorithm [53], grey wolf optimization (GWO) and its enhanced versions [54][47][55][49], GA [45][56][57], have been applied to improve the predictive performance of ANN, adaptive neuro-fuzzy inference system (ANFIS) and SVM on the axial compression capacity (ACC) of CFST.

### *3.6.3 Ensemble model*

#### *3.6.3.1 Adaptive boosting*

Adaptive boosting (AdaBoost) is a popular ensemble technique developed by Freund and Schapire in 1997 [58]. Its core concept is to combine multiple weak learners (i.e., estimators that perform slightly better than random guessing) into a strong learner, as shown in **Fig. 7**e. AdaBoost focuses on difficult-to-estimate instances in subsequent iterations by adjusting the weights of the misclassified samples. Decision trees are typically used as a weak estimator. At $t$-th boosting iteration ($t$=1, 2,..., $T$), the weight of sample $i$ is denoted by $w_t(i)$ and can be initially set to be equal for all samples. The average loss of week estimator $f_t(x)$ is computed as

$$\overline{L_t} = \sum_{i=1}^{n} w_t(i) L_t(i) \tag{6}$$

where $L_t(i) = \left| f_t(x_i) - y_i \right| / \max_{i=1\ldots n}\left( \left| f_t(x_i) - y_i \right| \right)$. Subsequently, the coefficient $b_t$ for the weak estimator $f_t(x)$ is described by

$$\beta_t = \frac{\overline{L_t}}{1 - \overline{L_t}} \tag{7}$$

The weights of the incorrectly predicted samples will increase, making them more important for the next weak learner. Specifically, the weights of training examples are updated to emphasize the incorrectly predicted instances by following

$$w_{t+1}(i) = \frac{w_t(i) \beta_t^{(1-L_t(i))}}{Z_t} \tag{8}$$

where $Z_t$ is a normalization constant. The final strong estimator $f(x)$ is a weighted combination of the weak learners expressed by

$$f(x) = \sum_{t=1}^{T} \log\left(\frac{1}{\beta_t}\right) \text{median}\left\{ \beta_t f_t(x) \right\} \tag{9}$$

*3.6.3.2 Random forest*

Random forest (RF), developed by Leo Breiman in 2001 [59], is a simple yet effective ensemble learning algorithm. As shown in **Fig. 7**f, RF builds a "forest" of decision trees, each trained on a different data subset. For regression, the final prediction is the average of individual tree predictions, while for classification, the final prediction is determined by majority vote. Taking regression as an example, the final prediction can be expressed as

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x'), \tag{10}$$

where $B$ is the number of trees, and $f_b(x')$ is the individual prediction of input $x'$.

*3.6.3.3 Light gradient boosting machine*

Light gradient boosting machine (LightGBM) is a highly efficient gradient boosting framework optimized for large datasets [60]. It uses decision trees as base learners and optimizes both training speed and memory usage while maintaining high accuracy. LightGBM operates by sequentially adding weak learners (i.e., decision trees) to correct previous errors. Unlike traditional methods that grow trees level-wise, LightGBM grows leaf-wise by splitting the leaf with the maximum gain, as shown in **Fig. 7**e. The gain can be calculated by

$$\text{Gain} = \frac{1}{2}\left( \frac{\left(\sum_{i \in L} y_i\right)^2}{|L| + \lambda} + \frac{\left(\sum_{i \in R} y_i\right)^2}{|R| + \lambda} - \frac{\left(\sum_{i \in S} y_i\right)^2}{|S| + \lambda} \right) \tag{11}$$

where $L$ and $R$ are the left and right child nodes, respectively; $S$ is the parent node; $y_i$ represents the target variable. LightGBM also applies regularization techniques to prevent overfitting, which can be expressed as:

$$\text{Objective} = \text{Loss} + \lambda \sum_{j=1}^{K} w_j^2 + \alpha \sum_{j=1}^{K} \left| w_j \right| \tag{12}$$

where $K$ is the number of leaves; $w_j$ represents the leaf weights; and $\lambda$ and $\alpha$ are the regularization parameters.

*3.6.3.4 eXtreme Gradient Boosting*

eXtreme Gradient Boosting (XGBoost) is a powerful ML algorithm based on decision tree ensembles [61]. It improves traditional gradient boosting by incorporating a regularization term to reduce overfitting and enhance model robustness, which can be expressed as

$$\mathcal{L}(\phi)=\sum_i L\left(y_i,\hat{y}_i\right)+\sum_k \Omega\left(f_k\right) \quad (13)$$

where $\mathcal{L}(\phi)$ denotes the loss function; $L\left(y_i,\hat{y}_i\right)$ is the realistic loss between the real values $y_i$, and the predicted values $\hat{y}_i$; $\Omega\left(f_k\right)$ is the regularization function to control the model complexity.

#### *3.6.4 Model implementation*

Python has been one of the most popular programming languages for ML implementation. **Table 3** summarizes Python libraries commonly used for ML and data science. Scikit-learn is versatile for classification, regression, and clustering with built-in tools and models. TensorFlow and PyTorch are powerful libraries for deep learning. TensorFlow excels in production deployment and PyTorch is known for its dynamic computational graph. Keras provides an API for building neural networks. XGBoost, LightGBM, and CatBoost specialize in gradient boosting and handling structured data and categorical features. NLTK and spaCy are robust tools for natural language processing, and OpenCV is widely used for computer vision. These libraries together create a comprehensive python-based ML ecosystem.

### *3.7 Model Training, Validation and Testing*

#### *3.7.1 Objective function*

The objective function guides the training by measuring the difference between predicted output and ground truth. The objective function normally consists of a loss function which quantifies prediction errors, and a regularization term that controls the model complexity and mitigates overfitting. Therefore, the objective function for training ML can be written as

$$\text{Objective Function} = \text{Loss Function} + \text{Regularization Term(s)} \quad (14)$$

The loss function depends on the ML objective. Common examples are mean absolute error (MAE) and MSE for regression (see **Table 4**), and cross-entropy loss for classification, which is expressed as

$$\text{Cross-Entropy}=-\sum_{i=1}^{n} y_i \log\left(\hat{y}_i\right) \quad (15)$$

where $y_i$ is the true label (i.e., 0 or 1), and $\hat{y}_i$ is the predicted probability for the positive class.

**Table 3**

Commonly used open-source Python libraries for ML.

| Library Name | Characteristics | Objectives | Link |
| --- | --- | --- | --- |
| **scikit-learn** | Simple to use, rich in tools, widely used in research and industry | Classification, regression, clustering, dimensionality reduction, model selection | scikit-learn |
| **TensorFlow** | Powerful deep learning framework, supports distributed computing and production deployment | Deep learning, neural networks, reinforcement learning | TensorFlow |
| **Keras** | High-level neural networks API, supports rapid prototyping, easy to use | Deep learning, convolutional neural networks, recurrent neural networks | Keras |
| **PyTorch** | Dynamic computation graph, suitable for research and development, strong community support | Deep learning, neural networks, reinforcement learning | PyTorch |
| **XGBoost** | Efficient gradient boosting framework, excels in handling structured data | Classification, regression, ranking | XGBoost |
| **LightGBM** | Fast, distributed gradient boosting framework based on decision trees | Classification, regression, ranking | LightGBM |
| **CatBoost** | Excels at handling categorical features, | Classification, regression | CatBoost |

| | | | |
|---|---|---|---|
| | supports GPU acceleration | | |
| **NLTK** | Natural language processing toolkit, includes rich corpora and models | POS tagging, tokenization, text classification, sentiment analysis | NLTK |
| **spaCy** | Efficient natural language processing library, suitable for large projects | POS tagging, named entity recognition, dependency parsing | spaCy |
| **OpenCV** | Computer vision library, supports image and video processing | Image processing, feature detection, face recognition | OpenCV |

Note: POS: part-of-speech; XGBoost: eXtreme gradient boosting; LightGBM: light gradient boosting machine; CatBoost: category boosting; NLTK: natural language toolkit.

Regularization terms, such as L1 (Lasso) and L2 (Ridge) regularizations, are used to penalize complex models. L1 regularization penalizes the L1-norm of the model parameters by following

$$L1 = \alpha \sum_{j=1}^{K} \left| w_j \right| \tag{16}$$

L2 regularization penalizes the squared values of the model's weights:

$$L2 = \lambda \sum_{j=1}^{K} w_j^2 \tag{17}$$

It should be mentioned that the selection of the regularization term depends on the specific problem and dataset. L2 regularization is less robust to outliers due to the squared penalty on large parameters.

### *3.7.2 Hyperparameter tuning*

Hyperparameters are determined empirically before training and can have a significant impact on model performance. Therefore, selecting optimal hyperparameters is crucial for achieving accurate predictions. Several hyperparameter tuning techniques will be discussed below.

#### *3.7.2.1 Cross-validation*

*k*-fold cross-validation is widely used for hyperparameter tuning, particularly when the dataset is too small to be divided into a separate validation set. The overall training set is split into *k* equal-sized folds. One fold is used as the validation set, while the remaining (*k*-1) folds are used for training, as shown in **Fig. 8**. This process is repeated *k* times, and the averaged performance metrics (e.g., accuracy or MSE) are calculated to evaluate the model's performance with different hyperparameters.

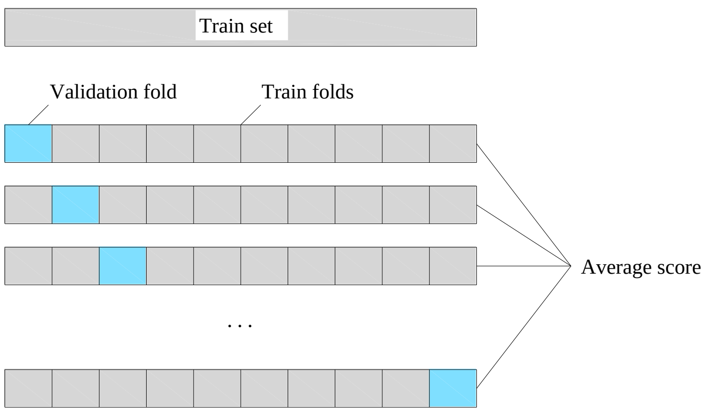


**Fig. 8** *k*-fold cross-validation.

#### *3.7.2.2 Grid search method*

Grid search evaluates all possible hyperparameter combinations using cross-validation and is more computationally expensive, as shown in **Fig. 9**. Studies by Lee et al. [62] and Feng et al. [63] have used grid search to tune categorical gradient boosting and AdaBoost, and they found the best combination of hyperparameters after exhaustive search.

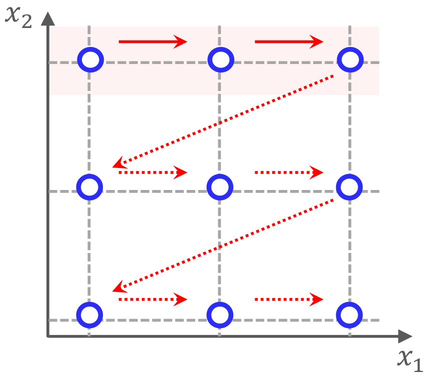

**Fig. 9** Grid search tuning strategy.

### *3.7.2.3 Sequential model-based optimization*

Sequential model-based optimization (SMBO), also known as Bayesian optimization, offers a more efficient strategy for hyperparameter tuning compared with the grid search. SMBO uses probabilistic models to identify promising hyperparameters based on previous evaluations to find the global minimum [64]. A SMBO framework is illustrated in **Fig. 10** to optimize hyperparameters of LightGBM. SMBO begins by building a probabilistic surrogate model (i.e., a Gaussian process or RF) to estimate the objective function, which is costly to evaluate through model training. An acquisition function (e.g., expected improvement) selects the next point in the hyperparameter space for evaluation. The steps above will refine the model iteratively, and the optimal hyperparameters can thus be accurately identified.

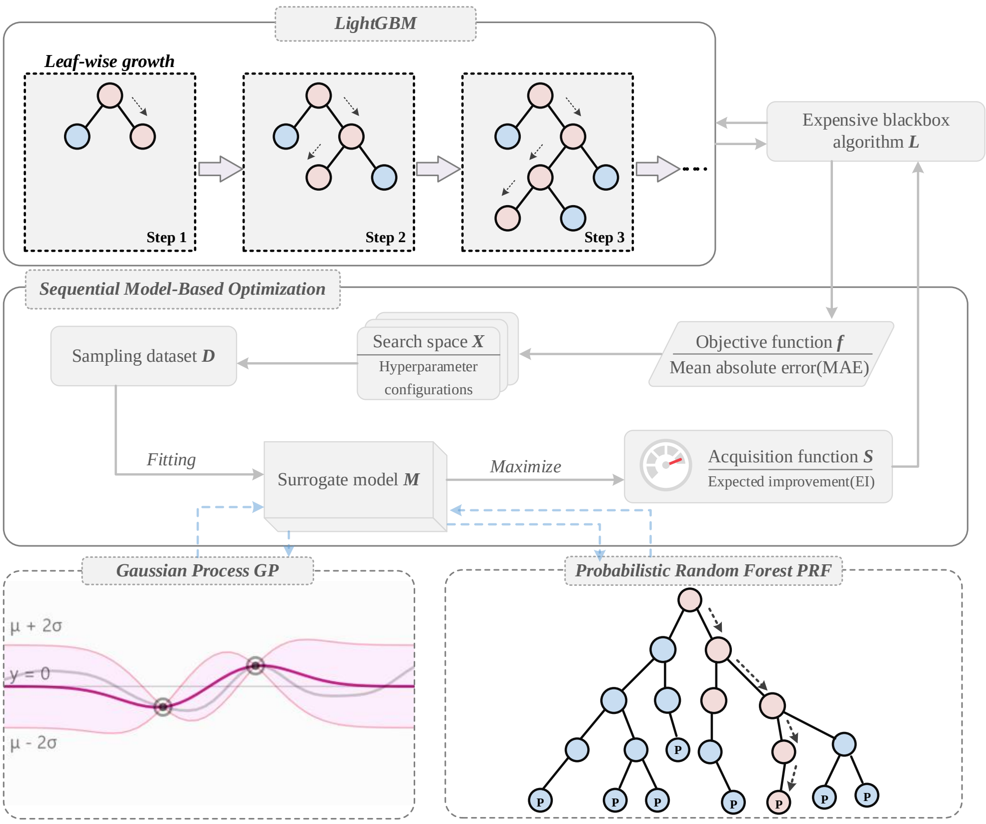

**Fig. 10** Overview of the SMBO framework for hyperparameter tuning (adapted from [27]).

### *3.7.3 Evaluation matrices*

To evaluate the performance and accuracy of ML models, various evaluation matrices can be used for classification and regression tasks. Common statistical metrics used in SCCS applications are summarized in **Table 4**. For classification, the metrics include recall, precision, accuracy, *F1*-score, and confusion matrix. Recall, also known as sensitivity, measures the proportion of true positives correctly identified, while precision measures the proportion of true positive predictions among all positive predictions. Accuracy measures the proportion of correctly classified instances among the total instances, and the *F1*-score is the harmonic mean of precision and recall. A confusion matrix is a table that summarizes the classification performance and provides a detailed breakdown of classification results. For regression, metrics include mean absolute error (MAE), mean absolute percentage error (MAPE), correlation coefficient ($R$), coefficient of determination ($R^2$), mean square error (MSE), root mean square error (RMSE), and root mean squared logarithmic error (RMSLE). MAE captures the absolute difference between predicted and actual SCCS performance, and MAPE accounts for the relative error in relation to the actual values. MSE and RMSE emphasize outliers by calculating the squared errors and their square roots. $R$ measures the linear relationship between predictions and actual values, and $R^2$ measures the proportion of variance explained by the model. RMSLE is ideal for cases where the target variable has a wide range or when underestimations need to be penalized more than overestimations.

**Table 4**

Statistical metrics for evaluating the performance of ML models for different tasks.

| **Classification model** | **Example references** | **Regression model** | **Example references** |
| --- | --- | --- | --- |
| $Recall=\frac{TP}{TP+FN}$ | [2][65] | $MAE=\frac{1}{N}\sum_{i=1}^{N}\left\|x_i-x_i\right\|$ | [53][66] |
| $Precision=\frac{TP}{TP+FP}$ | [2][65] | $MAPE=\frac{100\%}{N}\sum_{i=1}^{N}\left\|1-\frac{x_i}{x_i}\right\|$ | [27] |
| $Accuracy=\frac{TP+TN}{TP+TN+FP+FN}$ | [2][65] | $R=\frac{N\sum_{i=1}^{K}x_i x_i}{\sqrt{N\sum_{i=1}^{N}(x_i)^2-(\sum_{i=1}^{N}x_i)^2}\sqrt{N\sum_{i=1}^{N}(x_i)^2-(\sum_{i=1}^{N}x_i)^2}}$ | [67][68] |
| $F_1-\ score=\frac{2*Recall\ *Precision}{Recal+Precision}$ | [2][65] | $R^2=1-\frac{\sum_{i=1}^{N}(x_i-\hat{x}_i)^2}{\sum_{i=1}^{N}(x_i-\bar{x})^2}$ | [69][70] |
| $Confusion\ Matrix=\begin{pmatrix} TP & FP \\ FN & TN \end{pmatrix}$ | [2] | $MSE=\frac{1}{N}\sum_{i=1}^{N}(x_i-x_i)^2$ | [33][71] |
| | | $RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-x_i)^2}$ | [2][72] |
| | | $RMSLE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(x_i+1)-\log(x_i+1))^2}$ | [23] [27] |

Note: *TP*: true positive; *TN:* true negative; *FP:* false positive; *FN:* false negative; $x_i$: measured value; $x_i$ : predicted value; $N$: number of samples; $\bar{x}=\frac{1}{N}\sum_{i=1}^{N}x_i$ .

# 4. Applications of ML in Steel-Concrete Composite Structures

## *4.1 Mechanical connectors*

### *4.1.1 Headed stud connectors*

Headed studs are one of the most commonly used mechanical connectors for SCCSs. As shown in **Fig. 11**a, headed studs are welded to the steel structure and embedded in concrete to transfer shear and uplift force in SCCSs. ML applications for stud connectors focus on predicting shear resistance [27][73–80], stiffness [28], and ultimate slip [81], as well as optimizing design [82][83], identifying damage modes [25], and evaluating long term performance such as fatigue life [84][85] under various configurations. A summary is provided in **Table 5**.

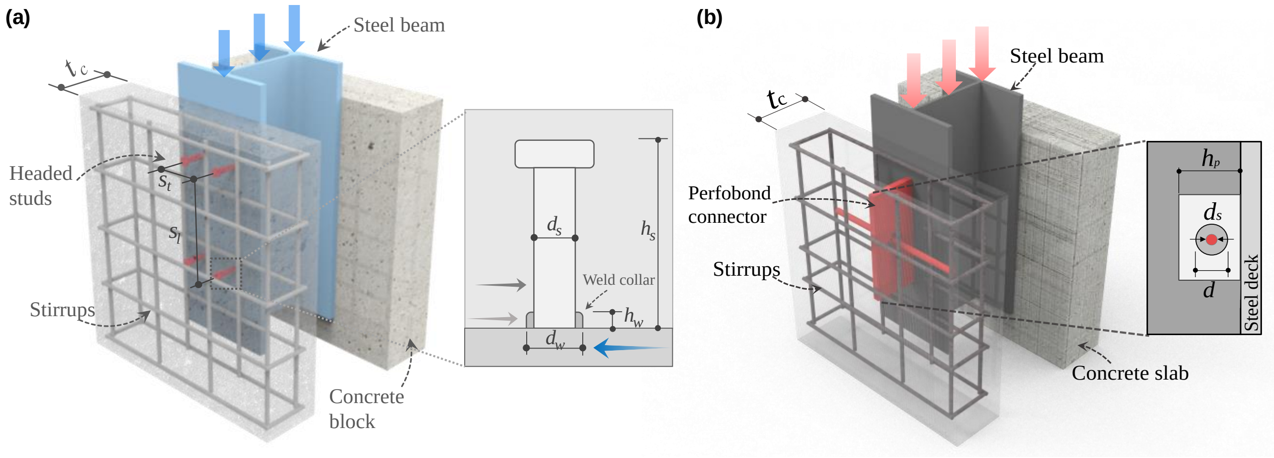


**Fig. 11** Schematic diagram of the push-out test of mechanical connectors: (a) headed studs; (b) PBL connectors (adapted from [27][2]).

**Table 5**
Applications of ML in predicting mechanical responses of headed stud connectors.

| Task | Application | Applied ML algorithm(s) | Reference(s) |
|---|---|---|---|
| Regression | Shear resistance | ANN | [73][80] |
| | | ELM, MPMR | [74] |
| | | LR, DT, BET, SVM, GPR, ANN | [75] |
| | | ANFIS, ANN, ELM | [77] |
| | | EN, BR, DNN, LightGBM, GP-LightGBM, PRF-LightGBM | [27] |
| | | SVM, ANN, DT, RF, GBDT | [78] |
| | | IEPSO-ANN, PSO-ANN, LM-ANN, ELM | [79] |
| | | ANN, GA-ANN, ELM, RF, SVM | [83] |
| | | NGBoost, XGBoost, LightGBM, CatBoost, SVM | [76] |
| | Shear stiffness | ATDF, DNN, RF, HGBDT | [28] |
| | Ultimate slip | LightGBM, RF, CatBoost, ExtraTrees, XGBoost, Voting | [81] |
| | Fatigue life | LR, DT, SVM, GPR, BET, ANN | [84][85] |
| Clustering | Damage identification | Fuzzy *C*-means | [25] |

Note: ANN: artificial neural network; ELM: extreme learning machine; MPMR: minimax probability machine regression; LR: linear regression; DT: decision tree; BET: bagged ensemble trees; SVM: Support vector machine; GPR: Gaussian process regression; ANFIS: adaptive neuro-fuzzy inference system; EN: elastic net; BR: Bayesian ridge; DNN: deep neural network; LightGBM: light gradient boosting machine; GP: Gaussian process, PRF: pseudo-random forest; RF: random forest; GBDT: gradient boosting decision tree; IEPSO: improved evolutionary particle swarm optimization; PSO: particle swarm optimization; LM: Levenberg-Marquardt algorithm, GA: genetic algorithm; NGBoost: natural gradient boosting; XGBoost: eXtreme gradient boosting; CatBoost: category boosting; ATDF: auto-tuning deep forest; HGBDT: histogram-based gradient boosting decision tree; ExtraTrees: extremely randomized trees.

Push-out tests are the primary experimental method to assess the shear performance of headed studs, as shown in **Fig. 11**a. Standalone models have been developed to predict the shear capacity of stud connectors. Notable examples include the ANN model developed by Abambres and He [73], Zhang et al. [78], Chen et al. [80] based on push-out test data. Additionally, Avci-Karatas [74] employed extreme learning machine (ELM) and minimax probability machine regression (MPMR), while Setvati and Hicks [75] trained a support vector machine (SVM) on a dataset of 242 samples. Furthermore, an adaptive neuro-fuzzy inference system (ANFIS) has been developed by Yosri to predict the shear strength of stud connectors [77]. These models outperformed traditional empirical equations. The SVM achieved an $R^2$ of 0.95 [75], and MPMR and EML had $R^2$ of 0.99 and 0.95, respectively [74].

Other studies focused on ensemble models to predict the shear resistance of stud connectors. Wang et al. [27] introduced an auto-tuned ensemble learning approach based on an extensive database of 1092 push-out tests of stud connectors. The ensemble model was tuned by the SMBO method and it outperformed standalone ML models and national standards such as AASHTO and EC4 in predicting the shear resistance. The similar ensemble strategy was adopted and applied by Zhang et al. [78]. They concluded that the gradient boosting decision tree (GBDT) model exhibited the highest accuracy compared to AASHTO with 80% lower RSME and MAPE. Additionally, Zhu et al. [79] and Sun et al. [83] proposed hybrid models that combine ANN with

GA and PSO to mitigate overfitting and improve prediction accuracy. While most studies focus on deterministic prediction of shear resistance, Degtyarev and Hicks [76] have made probabilistic predictions with confidence levels and uncertainties using the natural gradient boosting (NGBoost) model. Researchers have developed web applications for the trained ML model to aid the actual design of headed stud connectors [27][81][86].

ML has been less developed for other areas of headed studs besides its shear resistance. Wang et al. [28] developed an auto-tuning deep forest (ATDF) to predict the shear stiffness of headed stud connectors and compared the prediction results with existing equations. Yao et al. [25] applied the unsupervised fuzzy *C*-means clustering to analyze acoustic emission (AE) signals from the damage of the steel-concrete interface with stud shear connectors. Moreover, Roshanfar et al. [85] introduced six standalone ML models to predict the fatigue life of shear connectors in composite bridges and compared with *S-N* curves in AASHTO LRFD bridge design specifications.

### *4.1.2 Perfobond strip connectors (PBLs)*

PBLs are being increasingly used in SCCSs due to their high shear capacity and stiffness and improved fatigue performance. Several studies have applied ML methods to predict the shear resistance of PBLs using data from push-out tests, as shown in **Fig. 11**b. Wei et al. [87], Allahyari et al. [88] and Chen et al. [89] employed ANNs to predict the shear resistance of PBLs under different design parameters, such as the concrete and steel strength, steel plate thickness, opening diameter of steel plate, as well as perforating reinforcement diameter. Their results showed that ML had better prediction accuracy in predicting shear resistance of PBLs compared to traditional empirical equations. Allahyari et al. [88] developed a user-friendly equation for the strength prediction of PBLs based on ANNs. They found that both the ANN model and the proposed equation achieved a higher accuracy than existing empirical equations. Wang et al. [71] applied ANN to estimate the shear resistance of a novel PBL connector which is deeply encased in reinforced concrete. A strong correlation between predicted and actual value was achived with a $R^2$ of 0.97. Building on these studies, recent studies have transitioned from ordinary ANN to hybrid ML approaches. Khalaf et al. [90] and Chen et al. [91] integrated optimization techniques such as GA and improved adaptive genetic algorithms (IAGA) for predicting shear resistance of PBLs. Furthermore, Liu et al. [2] applied the ensemble learning algorithm, CatBoost, to PBLs and found it outperformed traditional methods with a 67.2% reduction in MAE. They also investigated the failure mode classification by integrating the synthetic minority oversampling technique (SMOTE) into the ML framework. This approach effectively addressed the imbalanced data distribution in the original dataset.

### *4.1.3 Other types of connectors*

Other types of mechanical connectors include steel bolts, anchors, channels, angles, plate connectors, and composite dowels. For bolt connectors, Li et al. [92] applied an ANN to predict the shear strength of the high-strength friction-grip bolts in composite beams. Design parameters such as concrete strength and bolt diameter are considered as inputs to the model. Similarly, Hosseinpour et al. [93] developed an ANN-based model to predict the shear strength of bolts based on parametric studies of finite element (FE) models. Furthermore, Saleem [94] explored the ANNs in assessing the capacity of anchor bolts using non-destructive testing considering the effects of ultrasonic pulse velocity. Olalusi and Spyridis [95] applied Gaussian process regression (GPR) and SVM to predict the concrete breakout capacity of single anchors under shear. The ML models showed more accurate predictions compared with traditional methods provided by Eurocode 2 and ACI 318. In the case of channel connectors, Shariati et al. [96] developed a hybrid ANN-PSO model to predict the load-slip behavior of channel shear connectors embedded in normal and high-strength concrete. The hybrid model showed improved prediction accuracy compared to the conventional ANN model. For angle connectors, Sadeghipour Chahnasir et al. [97] applied an SVM optimized with a Firefly algorithm (FFA) to evaluate the shear capacity. Shariati et al. [98] compared the performance of various ML models in predicting the shear resistance of angle connectors, including ANN, ANFIS, and ELM. They concluded that ELM performed slightly better than ANN and ANFIS with a reduced computational time. For steel plate connectors, Vijayakumar and Pannirselvam [99] integrated an ANN with GA for multi-objective optimization through push-out tests. Their optimization results indicated that plate connectors with a length of 40 mm, a height of 125 mm, and a thickness of 12 mm are the optimal dimensions to achieve the maximum ultimate load and minimal relative slip. For composite dowels, Xiong et al. [100] studied the PSO-ANN, ANFIS and ELM for predicting the pull-out resistance of puzzle-shaped and clothoidal-shaped dowels encased in UHPC. The embedment depth proved to be the most influential parameter and ELM achieved the most accurate prediction.

### *4.2 Steel-concrete interfacial bonding*

ML applications addressing steel-concrete interfacial bonding in SCCSs can be categorized into two groups, (1) the bond between steel bar and concrete [101][102], and (2) the bond between structural steel and concrete, as shown in **Fig. 12**.

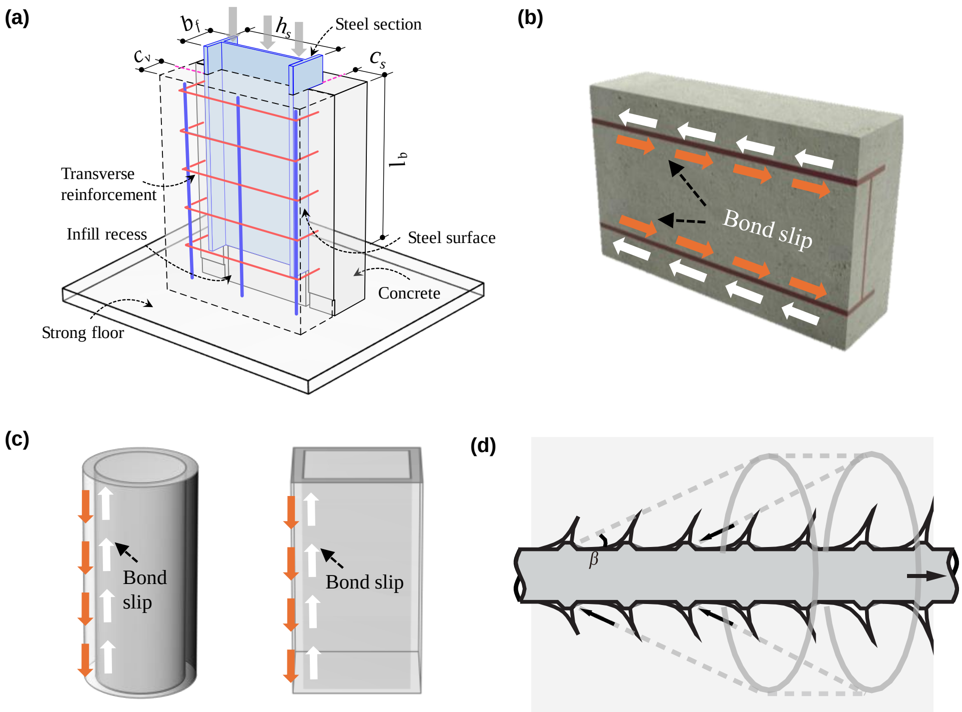


**Fig. 12** Schematic diagram of steel-concrete interfacial bonding: (a) push-out test for interfacial bonding (adapted from [22]); (b) steel section-concrete bond; (c) steel tube-concrete bond; (d) steel rebar-concrete bond.

Applications of ML in predicting the bond strength and behavior in SCCSs are summarized in **Table 6**. The first application of ML for steel section-concrete interfacial bond was conducted by Wang et al. [29], who proposed a hybrid approach combining

ANN with GA or PSO to predict the bond strength. An explicit formula was derived from the PSO-ANN model and a graphical tool was created for practical design practice. Building on this, Wang et al. [22] expanded the database to include 302 push-out tests (see **Fig. 12**a) and evaluated the explainable ensemble learning models in predicting the bond strength between steel sections with different surface treatments and various concrete types. Similar strategy was later validated by Zhang et al. [103] and Gupta et al. [104]. Recently, Yu et al. [105] investigated probabilistic ML models incorporating Bayesian updating process and the Markov chain Monte Carlo (MCMC) method to estimate characteristic bond stresses (i.e., initial, peak, and residual bond stresses). Their method enables the probabilistic calibration of deterministic models by integrating confidence levels within a performance-centric framework. For steel tube-concrete interfacial bond, bond strength prediction was conducted using ANN on 157 circular and 105 squared specimens by Allouzi et al. [106], and on 143 square and 254 circular specimens by Almasaeid et al. [70]. The bond-slip behavior was modeled using an ANN and validated with FE models that incorporated cohesion damage to simulate the behavior of CFST under axial loads [106].

The detection of steel-concrete interfacial debonding using ML techniques is attracting more attention recently. Steel-concrete interfacial debonding is an invisible damage but it significantly weakens strength and durability of SCCSs. Cao et al. [107] combined wavelet video diagrams and the deep learning model MobileNetv2 to convert acoustic signals into time-frequency diagrams for precise detection of steel tube-concrete debonding. Li et al. [108] introduced multi-damage indicators in the time domain and statistical and conventional features in the frequency domain to represent the interfacial characteristics. They employed five ML models, *k*-NN, SVM, LR (logistic regression), AdaBoost, and Bernoulli Naive Bayes (BNB) to perform percussion-based debonding detection through 2D damage imaging.

For steel-concrete interface at a smaller scale, the literature has focused on local bond strength of steel bars in concrete under various conditions (e.g., high temperature and fire conditions), as summarized in **Table 6**. Dahou et al. [109] and Makni et al. [110] developed ANN models on databases of 112 and 117 pull-out tests respectively local to predict the bond strength under normal conditions. Mahjoubi et al. [111] further presented a logic-guided neural network that combines data-driven methods with scientific knowledge to predict bond strength, interface slip, and the bond-slip relationship. Their model used a logic loss function and handled unstructured and incomplete data to supplement experimental data with logic-based data. In addition, ML techniques have been developed for bond strength of spliced steel bars [112]-[114] and their development length [115]. Another focus of recent ML application for local bond strength prediction is in the case of corroded rebars. For instance, Hoang et al. [116] adopted the least squares support vector machine (LSSVM) to predict the bond strength of corroded bars. Concha et al. [117], Seghier et al. [118], Owusu-Danquah et al. [119], Huang et al. [120], and Cavaleri et al. [121] developed different neural networks for this purpose. Additionally, Zhang et al. [122] developed a meta-learning approach, while Fu et al. [123], Wang et al. [124], and Wakjira et al. [125] employed ensemble learning techniques to enhance prediction accuracy. Recently, several studies have focused on using ML to predict the bond strength under elevated temperature. Mei et al. [126] applied NGBoost to establish a probabilistic model for the bond strength between steel rebars and concrete under high temperature. Similarly, Reshi et al. [69] compared the performance of five ML models in predicting the bond strength under elevated temperature with the best model to be RF. Moreover, Nematzadeh et al. [127] employed both GEP and ANN to predict the post-fire bond strength and bond-slip behavior of steel rebar embedded in steel fiber reinforced rubberized concrete. Besides, there is increasing interest in ML-based bond strength prediction for novel rebar and concrete materials. For instance, the bond strength between rebars and high-strength lightweight concrete [128], UHPC [66][68], and recycled aggregate concrete (RAC) [129] has been investigated using ML. Sun [130] applied SVM, RF, and XGBoost to model bond strength between ribbed stainless-steel rebar and concrete, and Li et al. [131] developed ensemble learning on 901 pull-out tests to study the reversed bond-slip behavior.

**Table 6**
Applications of ML in predicting the bond strength and behavior in SCCS.

| ML Task | Interfacial type | Objective | Representative ML algorithm | Reference(s) |
|---|---|---|---|---|
| Regression | Steel section-concrete | Bond strength prediction | ANN, GA-ANN, PSO-ANN | [29] |
| | | | MLR, ANN, SVM, CART, AdaBoost, LightGBM | [22] |
| | | | RF, AdaBoost, GBDT, XGBoost | [103] |
| | | | DT, AdaBoost, RF, XGBoost | [104] |
| | | Characteristic bond stress prediction | Bayesian updating, MCMC | [105] |
| | Steel tube-concrete | Bond-slip prediction | ANN, ANOVA | [106] |
| | | Bond strength prediction | ANN | [70] |
| | Steel rebar-concrete | Bond strength prediction | ANN | [109] [110] |
| | Steel rebar-LWC | | FL | [128] |
| | Steel rebar-UHPC | | MLR, SVM, PSO-ANN, IEPSO-ANN | [66] |
| | | | ANN, SVM, ANFIS | [68] |
| | Steel rebar-RAC | | RR, LASSO, ElasticNet, DT, RF, ET, GBDT, ANN | [129] |
| | Stainless-steel rebar-concrete | | SVM, RF, XGBoost | [130] |
| | Steel rebar-concrete | Bond-slip prediction | Logic-guided neural network | [111] |
| | | Spliced strength prediction | ANN, FL | [112] |
| | | | ANN | [113] |
| | | | SVM, NMR, ANN | [114] |
| | | Development length prediction | PCE, RSM, ANN | [115] |
| | | Corrosive bond strength prediction | LSSVM | [116] |
| | | | ANN | [117][119][120] |
| | | | ANN, RBF, GEP | [118] |
| | | | CNN | [121] |
| | | | Meta-learning | [122] |
| | | | BMA, DT, RF, GBDT, AdaBoost | [123] |
| | | | SVM, RF, AdaBoost, GBDT, DNN | [124] |
| | | | CART, KRR, *k*-NN, AdaBoost, GBDT, XGBoost | [125] |
| | | Elevated temperature bond strength prediction | NGBoost, SVM, DT, ANN, AdaBoost, RF, XGBoost | [126] |
| | | | RF, XGBoost, AdaBoost, DT, LR (linear regression) | [69] |
| | | Post fire bond strength prediction | GEP, ANN | [127] |
| | | Reversed bond-slip behavior | RF, AdaBoost, XGBoost | [131] |
| Classification | Steel tube-concrete | Debonding damage identification | CNN MobileNetv2 | [107] |
| | Steel plate-concrete | | *k*-NN, SVM, LR (logistic regression), AdaBoost, BNB | [108] |

Note: ANN: artificial neural network; GA: genetic algorithm; PSO: particle swarm optimization; MLR: multiple linear regression; SVM: support vector machine; CART: classification and regression tree; AdaBoost, LightGBM: light gradient boosting machine; RF: random forest; GBDT: gradient boosting decision tree; XGBoost: eXtreme gradient boosting; DT: decision tree; MCMC: Markov chain Monte Carlo; ANOVA: analysis of variance; FL: fuzzy logic; IEPSO: improved evolutionary particle swarm optimization; ANFIS: adaptive neuro-fuzzy inference system; RAC: recycled aggregate concrete; RR: ridge regression; LASSO: least absolute shrinkage and selection operator; ET: extremely randomized trees; NMR: nonlinear multi-regression; PCE: polynomial chaos expansions; RSM: response surface method; LSSVM: least squares support vector machine; RBF: radial basis function; GEP: gene expression programming; CNN: convolutional neural network; BMA: Bayesian model averaging; DNN: deep neural network; KRR: kernel ridge regression; *k*-NN: k-nearest neighbor; NGBoost: natural gradient boosting; LR: logistic regression; BNB: Bernoulli Naive Bayes.

### *4.3 Steel-concrete composite beam*

ML applications in optimizing and analyzing steel-concrete composite beams have been focused on three key areas: design optimization [132][133][65], prediction of mechanical behaviors [134]-[141], and damage detection [31][33][142]-[146]. Additionally, ML has been used to predict the temperature field in steel-concrete composite beams [147].

#### 4.3.1 Design optimization

Martínez-Muñoz et al. [65][132][133] conducted multiple analyses to optimize the design of a 60-100-60 m three-span steel-concrete composite bridge with a single box-girder using ML techniques. In [132], they developed a hybrid optimization method combining *k*-means clustering with swarm intelligence metaheuristics to find the optimal bridge design at the optimal cost and $CO_2$ emissions. The proposed hybrid sine-cosine algorithm (SCA) reduced construction costs by 1.1% compared to simulated annealing (SA) algorithm, but cost and emissions optimization showed inconsistencies due to steel grade variations affecting costs but not emissions. Similarly, Martínez-Muñoz et al. [133] minimized embodied energy and cost of the same bridge using *k*-means clustering, SCA and cuckoo search (CS) as the discretization technique, cutting computation time by 25.79% compared with the trajectory-based algorithm, threshold accepting with a mutation operator (TAMO). They also found that double composite action design on supports eliminates the need for continuous longitudinal stiffeners. In a subsequent study [65], their team incorporated deep neural networks (DNNs) to accelerate structural constraint computations in bridge design. By integrating *k*-means clustering with metaheuristic algorithms, they achieved an improvement in computation speed up to 50 times faster than conventional methods. This increased efficiency enabled a more comprehensive life cycle assessment (LCA) to balance environmental, social, and economic impacts.

#### 4.3.2 Mechanical behavior

ML has been applied to predict the mechanical behaviors such as the ultimate strength, shear capacity, deflection, and lateral-distortional buckling (LDB) of steel-concrete composite beams. Cellular steel-concrete composite beams are with web openings used in the composite floor system to allow a longer span and integration of ancillary facilities. The ultimate moment of LDB [134], deflection [135], and global shear capacity [136] of cellular steel-concrete composite beams have been predicted through ML. Specifically, ANNs, SVMs, XGBoost, and RFs were applied to predict the ultimate moment of LDB in the hogging moment region [134]. XGBoost performed the best with higher accuracy in terms of correlation coefficient and MSE with a training dataset generated by 458 FE model [134]. Mastan et al. [135] applied ANNs with the Levenberg Marquardt (LM) for the backpropagation of ANN to predict the deflection of composite beams with various web openings based on FE data. It was shown that web openings impact the bearing capacity and deformation of the composite beams, and ML models provided accurate deflection prediction. Ferreira et al. [136] focused on the global shear capacity of cellular composite beams with precast hollow-core using five ML models, where the CatBoost achieved the best performance. Their study found that the ratio of opening spacing to opening diameter is the most important feature for the global shear capacity of the cellular composite beam. For non-cellular steel-concrete composite beams, Hosseinpour et al. [137] applied ANNs and multiple regression models to predict ultimate LDB strength with a prediction error of 6-8% based on 425 FE models. Kumar et al [138] derived a closed-form expression from an ANN to predict the mid-span deflection of a simply supported composite beam considering long-term effects like concrete creep. Furthermore, Kumar et al. [139] developed GA-ANN and GWO-ANN metamodels to predict the deflection of steel-concrete composite beams. The dataset for training and testing was generated using Monte Carlo simulation-finite element method (MCS-FEM). Their results showed that the ANN-GA outperformed the ANN-GWO in terms of accuracy. Thirumalaiselvi et al. [140] and Xiong et al. [141] investigated two types of novel composite beam structure, laced steel-concrete composite (LSCC) beams [140] and modified Verbund-Fertigteil-Träger (MVFT) beams [141] using ML techniques. Multigene genetic programming (MGGP) showed good performance in load predictions, and MPMR excelled in deflection predictions for the laced steel concrete-composite (LSCC) beams [140]. For MVFT girders, neural networks and LSSVM produced results close to FE simulation to predict the bending strength.

### *4.3.3 Damage Detection*

**Table 7** summarizes the application of ML to detect damage in steel-concrete composite beams. Most studies trained ML models on datasets generated from FE simulations [31][33][142]-[146]. The damage was simulated by reducing the stiffness of components like steel beams, concrete slabs, and steel-concrete interfaces, and ML was applied to identify and localize the damage. For example, Bilotta et al. [31] applied a CNN to localize damage in stud connectors with simulated damage by reducing stiffness at the steel-concrete interface. In [33], a general regression neural network (GRNN) was used to identify damage based on modal strain energy change in composite beams. The damage was simulated by decreasing the stiffness of steel, concrete and their interface by 30% to 70%. Tan et al. [142] applied an ANN to vibration data to detect damage from steel beam and concrete slab on a steel-concrete composite beam. They reduced the stiffness of steel I-beam and concrete slab by 10% to 40%. Others establish datasets by conducting physical experiments. Zhang et al. [145] and Li et al. [146] conducted experiments to create physical damage on steel-concrete composite beams. Zhang et al. [145] trained a residual network-50 (ResNet-50) on data collected from fiber grating sensors to classify six types of damage, which was created by cutting notches on steel and concrete, and missing stud connectors. Li et al. [146] carried out reversed four-point bending tests of two 2.5 m-long steel-concrete composite beams to introduce actual damage under hogging moment regions, as shown in **Fig. 13**. Eight AE sensors were attached to the concrete surface to pick up the sound from damage. Concrete cracks and steel-concrete interface deboning were successfully located and quantified through the GA and hybrid hierarchical-*k*-means clustering analysis.

**Table 7**
Applications of ML in damage detection of steel-concrete composite beam.

| Ref. | Objectives of ML | ML model(s) | Source of ML dataset | Damage source | Damage simulation |
|---|---|---|---|---|---|
| [31] | Localize and identify damage in stud connectors | CNN | Simulation | Stud connectors | Reduce stiffness of the steel-concrete interface |
| [33] | Identify damage based on modal strain energy change | GRNN | Simulation | Steel beam, concrete slab, and steel-concrete interface | Reduce stiffness in simulation |
| [142] | Detect damage using vibration characteristics | ANN | Simulation | Steel beam and concrete slab | Reduce stiffness in simulation |
| [143] | Analyze sensitivity of steel-concrete composite beam bridges to damage | ET | Simulation | Bridge deck, concrete slab, steel beams, stud connectors, diaphragms, bearings, and piers | Introduce irregularities in the deck; reduce stiffness for concrete slab, stud connectors; diaphragms, piers, and bearings; create notches in steel |
| [144] | Localize the AE source | ANN | Simulation | Steel beam web | Pencil lead break test to generate crack-like AE signals |

| | | | | | |
|---|---|---|---|---|---|
| [145] | Classify and localize damage | ResNet-50 | Experiment | Steel beam, concrete slab, and shear connectors | Create notches to simulate damage from steel and concrete; miss stud connectors to simulate damage from mechanical connectors |
| [146] | Quantify and characterize damage with AE measurements | GA and hierarchical-*k*-means clustering | Experiment | Concrete cracks and steel-concrete interface debonding | Reversed four-point bending test of steel-concrete composite beams |

Note: GRNN: general regression neural network; CNN: convolutional neural network; ET: extremely randomized trees; ANN: artificial neural network; AE: acoustic emission; GA: genetic algorithm.

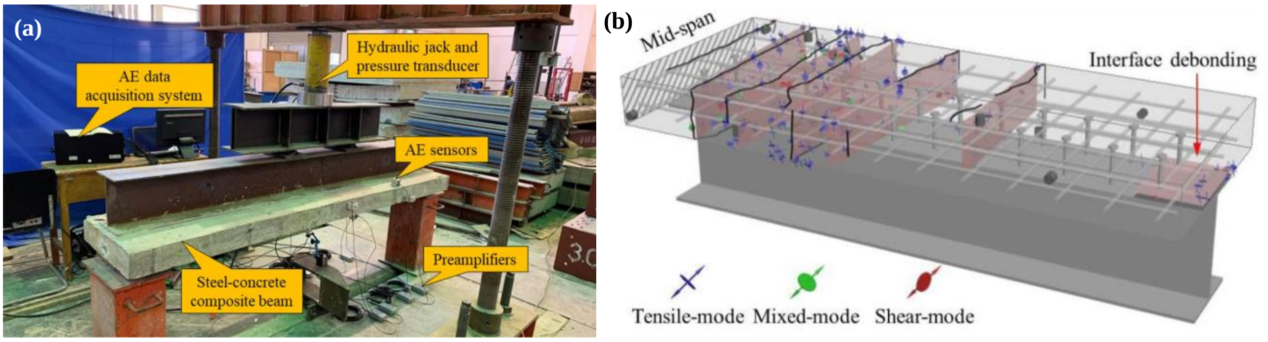


**Fig. 13** Damage detection and localization using hybrid hierarchical-*k*-means clustering: (a) reversed four-point bending test; (b) damage localization after loading (adapted from [146]). Note: AE: acoustic emission.

### *4.4 Steel-concrete composite slab*

Steel-concrete composite slabs are widely used in floor and bridge decks. However, predicting the performance of composite slabs under various conditions remains a challenge due to the complex interaction between steel and concrete, particularly under extreme conditions like fire accidents. To address these challenges, ML techniques have been employed to predict the performance of composite slabs.

Morasaei et al. [148], Panev et al. [149], and Shariati et al. [150] focused on the mechanical performance of composite slabs under fire accidents and elevated temperatures. Specifically, Morasaei et al. [148] applied ELM combined with optimization techniques such as PSO and GWO to predict the shear and tensile response of composite slabs under high temperatures. The ELM-GWO model was found to be more reliable in predicting slip and load than the ELM-PSO model. Panev et al. [149] used SVM to predict the fire insulation performance of shallow composite floor systems. They found that the SVM achieved a high accuracy of 96% in insulation predictions, yet the model struggled when dealing with data outside its training range. Shariati et al. [150] also focused on predicting high-temperature behavior of channel shear connectors in composite slabs at 550°C, 700°C, and 850°C. ELM, GP, and ANN were applied and the ELM outperformed the other models, particularly in predicting the load behavior of connectors. On the other hand, other researchers [151][152] are interested in predicting the deformation performance of composite slabs. In [151], ANNs were used to predict mid-span displacement in profiled composite slabs. The ANNs achieved a high accuracy with prediction errors below 10%. Zhang et al. [152] conducted full-scale experiments on hollow concrete composite slabs with recycled aggregates to establish the dataset, as shown in **Fig. 14**. The established dataset was further divided by a decision tree to train sub-ANNs. The existence of reinforcement on the bottom plate, thickness of the concrete layer, and hollow size were considered as the dividing criteria. The effectiveness of the sub-ANN framework was proven with accuracies on the testing set above 90% for the prediction on displacement, slip and strain.

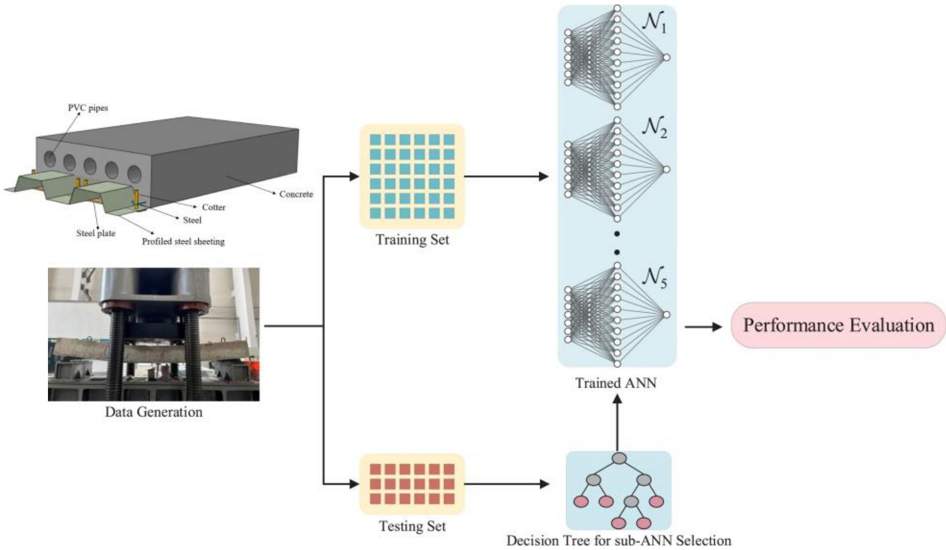


**Fig. 14** A sub-ANN framework to predict the mechanical performance of the steel sheet-hollow concrete composite slab (adapted from [152]). Note: ANN: artificial neural network.

Wang et al. [23] and Ruan et al. [32] investigated the ML applications in predicting the performance of precast slab joints for accelerated bridge construction (ABC). Deep forest (DF) was employed to predict the flexural capacity of precast slab joints in [23]. DF replaces the traditional neurons of DNN with RF to create a deep cascade structure. The output of each layer will be concatenated with the original input as the new inputs for the next layer. 391 samples from experiments were collected in [23] covering longitudinal rebars with diameter from 9.5 to 32.0 mm, yield strength from 335 to 575 MPa, and lap length from 64.0 to 900.0 mm, and joint concrete with strength from 22 to190 MPa. The DF model outperformed conventional models like RFs and DNN in predicting the flexural capacity of precast deck joints. One key finding was that the capacity of transverse rebars plays a crucial role in improving the overall flexural capacity when longitudinal rebars are constrained by design. Ruan et al. [32] utilized a physics-guided Long Short-Term Memory (LSTM) model to predict the non-linear bending moment-displacement curves of deck joints. The LSTM model was trained on a dataset created from bending tests and parametric analysis of FE models covering 3000 joint configurations. The physics-guided LSTM incorpated a penalty term to its loss function to ensure accurate prediction of the initial linear moment-displacement relationship. The model effectively predicted key performance metrics like stiffness, deflection, failure mode, and ultimate capacity, with a MAPE of less than 30%.

Additionally, Zhou et al. [153] utilized AE for damage detection in composite slabs. A deep residual network (DRN) was applied to classify and localize acoustic emissions on a large-scale composite slab extracted from a historical bridge. Their results showed that the DRN had better accuracy with AE sensors mounted on steel surfaces compared to concrete surfaces. However, the localization precision decreased when large cracks (i.e., 4-6 mm width) occur. Wang et al. [154] predicted the non-uniform shrinkage (NUS) of steel-concrete composite slabs using ML. 782 samples were collected from the literature and eight ML models were evaluated. The gradient boosting decision trees (GBDT) achieved the highest prediction accuracy. The measurement depth and concrete age were identified as the most influential variables in determining long-term shrinkage behavior.

### *4.5 Steel-concrete composite column*

CFSTs are one of the most efficient steel-concrete composite columns. Composite columns made of CFSTs have been widely used in high-rise buildings, bridges, and offshore structures due to their high strength, stiffness, ductility, and energy dispersion capacity. In a CFST column, the axial load is shared by both steel and concrete. Thus, the design and calculation of CFST columns are more challenging compared with RC columns. This challenge is being addressed by ML techniques, as summarized in **Table 8**.

**Table 8**
Applications of ML in predicting the axial compression capacity and behavior of CFSTs.

| Year | Reference | Data Quantity | Cross-section | Algorithm(s) | Input features | | | | | | | | | | | | | | | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | *L* | *B* | *H* | *D* | *t* | *B/t* | *D/t* | $f_y$ | $f_u$ | $f_c$ | $E_s$ | $E_c$ | $\lambda$ | *e* | shape | |
| 2014 | Ahmadi et al. [155] | 272 | Circular | ANN | √ | | | √ | √ | | | √ | | √ | | | | | | |
| 2015 | Jegadesh et al. [156] | 633 | Circular | ANN | √ | | | √ | √ | | | √ | | √ | | | | | | |
| 2017 | Du et al. [164] | 305 | Rectangular | ANN | | √ | | √ | √ | √ | √ | √ | | √ | | | | | | |
| 2019 | Tran et al. [67] | 300 | Square | ANN | √ | | √ | | √ | | | √ | | √ | | | | | | Master curves to derive a new empirical formula |
| 2019 | Ren et al. [46] | 540 | Square | PSO-SVM, DT, GPR, ANN | √ | | √ | | √ | | | √ | | √ | √ | √ | | | | |
| 2020 | Tran et al. [157] | 768 (FE) | Circular | ANN | √ | | | √ | √ | | | √ | √ | √ | | | | | | UHPC; GUI tool |
| 2020 | Nguyen et al. [53] | 422 | Rectangular | OSS-ANN, SVM, FL, EBT | √ | √ | √ | | √ | | | √ | | √ | | | | | | |
| 2020 | Zarringol et al. [167] | 3091 | Rectangular, Circular | ANN | √ | √ | √ | √ | √ | | | √ | | √ | | | | √ | √ | Strength reduction factors by MCS |
| 2020 | Duong et al. [50] | 150 | Rectangular | ANN, BCMO-ANN, DE-ANN, DA-ANN, SHG-ANN | √ | √ | √ | | √ | | | √ | | √ | | | √ | | | |
| 2020 | Nguyen et al. [173] | 99 | Rectangular | ANN, IWO-ANN | √ | √ | √ | | √ | | | √ | | √ | | | | | | |
| 2020 | Le [160] | 94 | Elliptical | RCGA-ANFIS, GD-ANFIS | √ | √ | | √ | √ | | | √ | | √ | | | | | | GUI tool |
| 2021 | Le et al. [161] | 314 | Square | KGPR, ANN, SVM, EBT, DT, FL, ANFIS-FCM | √ | | √ | | √ | | | √ | | √ | | | | | | GUI tool |
| 2021 | Ho et al. [158] | 1730 | Circular | LR, FL, RT, EBT, SVM, ANN, GPR | √ | | | √ | √ | | | √ | | √ | | | | √ | | Excel tool |
| 2021 | Ly et al. [45] | 222 | Elliptical, Circular | BBO-ANFIS, PSO-ANFIS, GA-ANFIS | √ | √ | | √ | √ | | | √ | | √ | | | | | | Monte Carlo approach to propagate the variability |
| 2021 | Naser et al. [56] | 3103 | Circular, Square, Rectangular | GA, GEP | √ | √ | √ | √ | √ | | | √ | √ | √ | √ | √ | | √ | √ | Compact and one-stepped predictive expressions |
| 2021 | Ngo et al. [54] | 802 | Circular | GWO-SVM, LR, ANN, SVM | √ | | | √ | √ | | √ | √ | | √ | | | | | | Normal, high, and ultimate strength concretes, AI-based tool |
| 2021 | Vu et al. [159] | 1017 | Circular | GBDT, RF, SVM, DT, DNN | √ | | | √ | √ | | | √ | | √ | | | | | | |
| 2021 | Seghier et al. [162] | 300 | Square | GEP | √ | | √ | | √ | | | √ | | √ | | | | | | Closed-form equations |
| 2021 | Lyu et al. [174] | 478 | Circular | SCA-SVM, ANN, RF, MLR | √ | | | √ | √ | | | √ | | √ | | | | | | |
| 2021 | Lee et al. [168] | 3103 | Rectangular, Circular | CatBoost, CART, AdaBoost, GBDT, RF, XGBoost, | √ | √ | √ | √ | √ | | | √ | | √ | | | | √ | | |

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | LightGBM, ANN, SVM | | | | | | | | | | | | | | | | | |
| 2022 | Bardhan et al. [47] | 559 | Circular | AGWO-ANN, EGWO-ANN, GWO-ANN, PSO-ANN, SSA-ANN, SMA-ANN, HHO-ANN, SVM, ELM, GMDH | √ | | | | √ | √ | | √ | √ | | √ | √ | √ | | | | |
| 2022 | Ngo et al. [55] | 136 | Circular | GWO-SVM, ANN, SVM, LR, RF, M5P | √ | | | | √ | √ | | √ | √ | | √ | | | √ | | | |
| 2022 | Cakiroglu et al. [165] | 719 | Rectangular | LR, RF, AdaBoost, GBM, LightGBM, XGBoost, CatBoost | √ | | √ | √ | | √ | | | √ | | √ | | | | | | SHAP method |
| 2022 | Le [163] | 314 | Square | GPR | √ | | | √ | | | | | √ | | √ | | | √ | | | GUI tool |
| 2022 | Mai et al. [57] | 300 | Square | FFA-RBFNN, DE-RBFNN, GA-RBFNN, ANN | √ | | | √ | | √ | | | √ | | √ | | | | | | |
| 2022 | Le et al. [51] | 880 | Rectangular, Square | ANN, BCMO-ANN | √ | | √ | √ | | √ | | | √ | | √ | √ | | | | | Explicit equation, Excel-based GUI tool |
| 2022 | Sarir et al. [48] | 149 | Square | PSO-ANN, ICA-ANN | √ | | | √ | | √ | | | √ | | √ | | | | | | |
| 2022 | Avci-Karatas [175] | 150 | Circular | ANN | √ | | | | √ | √ | | √ | √ | | √ | √ | √ | | | | New engineering index a20-index to further verify model reliability |
| 2023 | Memarzadeh et al. [176] | 993 | Circular, Square | GEP, ANN | | | | | | | √ | √ | √ | | √ | | | √ | | | Symmetrical cross-section |
| 2023 | Chen et al. [171] | 302 | Rectangular, Circular | ANN, SVM, *k*-NN, LightGBM, XGBoost, GBDT, RF | √ | | √ | | √ | √ | | | √ | | √ | | | | | | RAC, ultimate bearing capacity, peak strain and stress–strain model, SHAP method |
| 2023 | Duong et al. [166] | 1093 | Circular, Elliptical, Square, Rectangular | SVM | √ | | | | | | | | √ | | √ | | | | | √ | GUI tool |
| 2023 | Degtyarev et al. [169] | 3208 | Rectangular, Circular | AdaBoost, GBM, XGBoost, LightGBM, CatBoost | √ | | √ | √ | √ | √ | | | √ | | √ | | | √ | √ | √ | Reliability analysis to calibrate resistance reduction factors, web-based design tool, SHAP method |
| 2023 | Carvalho et al. [170] | 216 (FE) | Circular | ANN, RF | | | | | √ | | | √ | √ | | | | | √ | | | Stainless steel tubular columns, GUI tool |
| 2023 | Le et al. [52] | 1245 | Circular | ANN, BCMO-ANN, LR, FL, DT, EBT, GPR, SVM | √ | | | | √ | √ | | | √ | | √ | | | | | | Excel tool |
| 2024 | Deng et al. [177] | 220 | Circular | GMDH, GEP, RF | √ | | | | √ | √ | | | √ | | √ | √ | | | | | |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2024 | Gupta et al. [49] | 192 | Square | IPSO-ANN, PSO-ANN, GWO-ANN, IPSO-ANFIS, PSO-ANFIS, GWO-ANFIS | √ | | | √ | | √ | | | √ | | √ | |
| 2024 | Yu et al. [172] | 690 | Circular | ANN, WOA-ANN | √ | | | | √ | √ | | √ | √ | | √ | GUI tool |

Note: ANN: artificial neural network; AI: artificial intelligence; SVM: support vector machine; PSO: particle swarm optimization; DT: decision tree; GPR: Gaussian process regression; OSS: one-step secant; FL: fuzzy logic; EBT: ensemble boosted tree; BCMO: balancing composite motion optimization; DE: differential evolution; DA: dual annealing; SHG: second-harmonic generation; IWO: invasive weed optimization; RCGA: real coded genetic algorithm; ANFIS: adaptive neuro-fuzzy inference system; GD: gradient descent; KGPR: kernel-based Gaussian process regression; FCM: fuzzy *C*-means; LR: linear regression; RT: regression tree; BBO: biogeography-based optimization; PSO: particle swarm optimization; GA: genetic algorithm; GEP: gene expression programming; GWO: grey wolf optimization; GBDT: gradient boosting decision tree; RF: random forest; DNN: deep neural network; SCA: sine-cosine algorithm; MLR: multiple linear regression; CatBoost: category boosting; CART: classification and regression tree; AdaBoost: adaptive boosting; XGBoost: eXtreme gradient boosting; LightGBM: light gradient boosting machine; AGWO: augmented grey wolf optimizer; EGWO: enhanced grey wolf optimizer; SSA: salp swarm algorithm; SMA: Slime mould algorithm; HHO: Harris hawks optimization; ELM: extreme learning machine; GMDH: group method of data handling; M5P: M5 model trees; GBM: gradient boosting machine; FFA: firefly algorithm; RBFNN: radial basis function neural network; ICA: competitive imperialism algorithm; *k*-NN: *k*-nearest neighbors; IPSO: improved particle swarm optimization; WOA: whale optimization algorithm; FE: finite element; GUI: graphical user interface: MCS: Monte Carlo simulation, UHPC: ultra-high performance concrete; SHAP: SHapley additive explanations; RAC: recycled aggregate concrete.

It can be observed that research has been conducted for designing different shapes of CFST columns, including circular [155][156–159], elliptical [45][160], square [46][57][67][161][162][163], and rectangular [50][53][164][165], as shown in **Fig. 15**. Unified models applicable across different cross-sections are investigated [56][166]. ML applications for circular CFSTs are the most widely studied due to their common structural applications [155][156][157][158][159]. The elliptical cross-sections are less studied than the other cross-sections [45][160]. The first ML application for circular CFST columns was conducted by Ahmadi et al. [155]. They trained ANN to predict the ACC of circular CFST columns based on 272 samples with an average error of 5.8%. Similar studies were performed by Jegadesh et al. [156] and Tran et al. [157] on circular CFSTs using ANN, while the later trained ML models based on numerical parametric results.

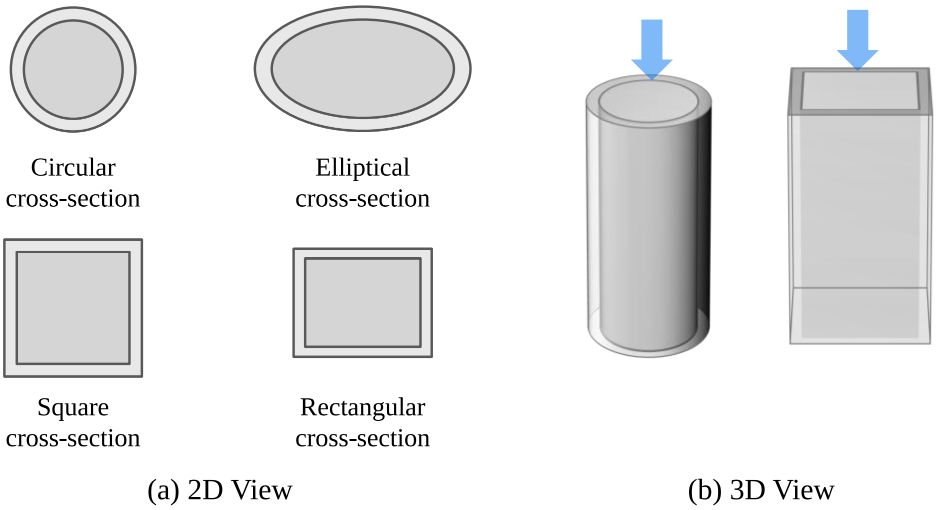


**Fig. 15** Different cross-sections of the concrete-filled steel tubes: (a) 2D view; (b) 3D view.

Moreover, ML techniques including standalone, hybrid, ensemble models were developed to predict the ACC of CFSTs. ANN, SVM and GEP are the most widely used standalone models. Du et al. [164] developed ANNs with five and ten inputs, such as length and width of cross section, tube thickness, yield strength of steel, and the concrete strength. Their model showed excellent prediction and generalization capacity compared with equations from EC4, ACI, GJB4142 and AISC360-10. Tran et al. [67] used ANN to generate a number of master curves to establish a new equation to predict the ACC of the square CFST columns. Zarringol et al. [167] discussed the strength reduction factors in equations developed from ANNs to ensure a safer design of CFST columns. For hybrid models, different metaheuristic optimization methods, such as BBO [45], PSO and its variants [45][46][47][48][49], competitive ICA [48], BCMO [50][51][52], OSS algorithm [53], GWO and its variants augmented grey wolf optimizer (AGWO), enhanced grey wolf optimizer (EGWO) [47][49][54][55], and GA [45][56][57], have been employed for improving the predictive performance of ANN, ANFIS, SVM or other models for ACC predictions of CFST. For example, Ly et al. [45] compared three hybrid ML algorithms with metaheuristic optimization methods of BBO, PSO, and GA in optimizing the weight parameters of ANFIS models. They concluded that PSO-ANFIS was the most efficient and robust model with a 20-index of 0.881 and $R^2$ of 0.942. Bardhan et al. [47] and Gupta et al. [49] compared the efficiency of GWO, PSO and their variants in optimizing the weights and biases of standalone models. For ensemble models, Lee et al. [168] trained several ML models, including CatBoost, CART, AdaBoost, GBDT, RF, XGBoost, and LightGBM, using a large database of 3,103 test samples. CatBoost yielded results closely matching experimental data. Cakiroglu et al. [165] employed LightGBM and CatBoost to predict the ACC with an accuracy of 97.9% and 98.3% respectively, which are more accurate than existing design codes. Degtyarev et al. [169] conducted a comparative study of five ensemble models for predicting the axial resistance of CFST columns using a comprehensive database of over 3200 test samples. A reliability analysis was performed to calibrate the resistance reduction factors.

Novel construction materials such as high-strength steel, stainless-steel [170], UHPC [54][157], and RAC [171] have been applied in composite columns. Tran et al. [157] and Ngo et al. [54] adopted ANN and SVM to estimate the ACC of UHPC-filled steel tube. Chen et al. [171] is interested not only on the ACC but also the peak strain and stress-strain model of RAC-filled steel tube using ML with SHAP method for explainability. Recent efforts have been conducted on the development of explicit equation and practical tools with a GUI. Tran et al. [67] proposed master curves to derive a new empirical equation based on ANN models. Seghier et al. [162] and Le et al. [51] adopted GEP to derive closed-form explicit equations, which outperformed the excited codes and equations. Furthermore, Le et al. [51][52][160][161][163], Ho et al. [158], Duong et al. [166], Carvalho et al. [170], and Yu et al. [172] developed AI tools with a GUI using Matlab or Python to improve design automation of CFST columns.

### *4.6 Steel-concrete composite wall*

Steel-concrete composite walls are designed to resist lateral loads in high-rise buildings. As shown in **Fig. 16**, composite walls have different configurations and are strengthened by steel either in the columns or in both the columns and the web. ML techniques have been found to predict the structural behaviors of composite walls, such as shear strength [178], flexural capacity [179], and deformation under impact load [180].

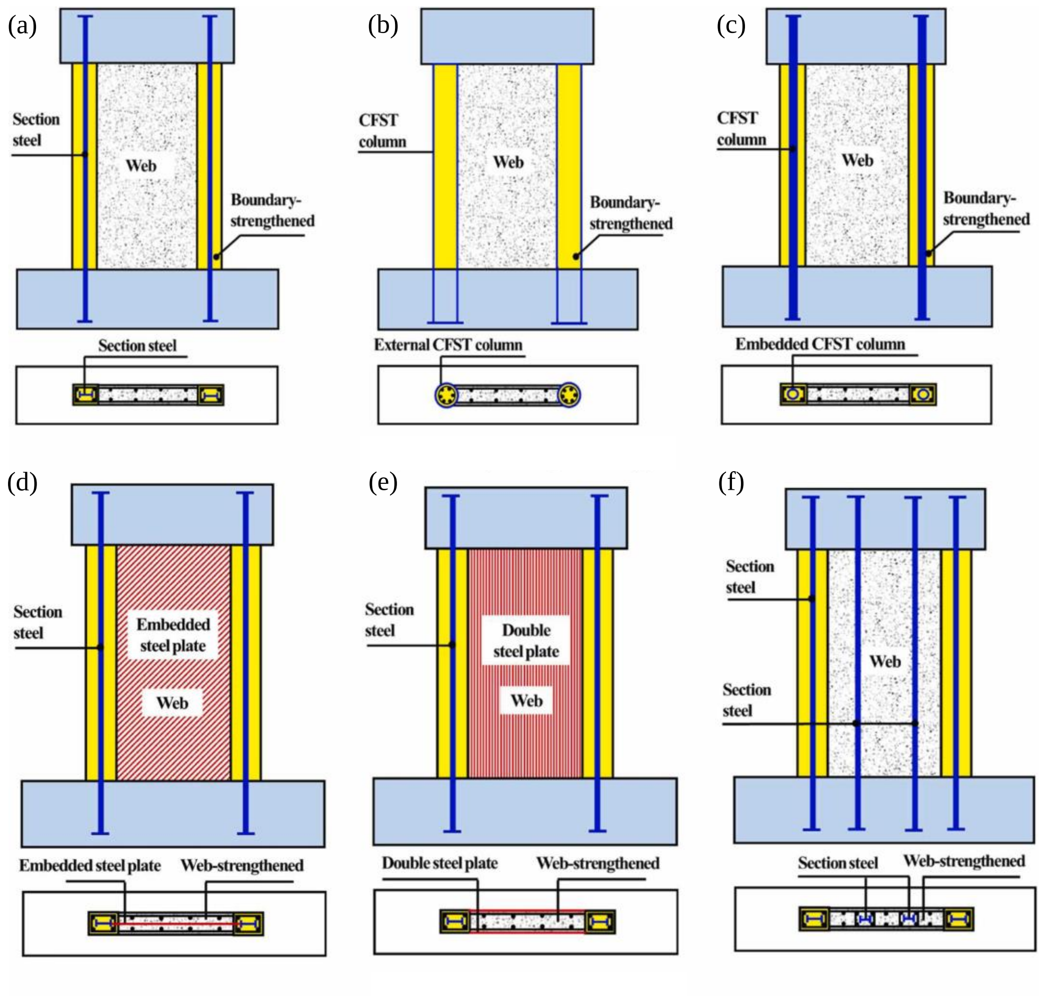


**Fig. 16** Typical steel-concrete composite walls: (a) with section steel reinforced columns; (b) with external CFST columns; (c) with embedded CFST columns; (d) with section steel reinforced columns and embedded steel plate web; (e) with section steel reinforced columns and double steel plate strengthened web; (f) with section steel reinforced columns and web (adapted from [178]).

In [178], Huang et al. trained 12 different ML models to predict the shear strength of steel-reinforced concrete composite shear walls (SRCCSW) on a dataset of 149 experiments. The XGBoost demonstrated the best accuracy in predicting the shear strength, and the height and shear-span ratio of the composite wall were the most influential factors on the shear strength. Mirrashid et al. [179] trained an ANN to predict the flexural capacity of composite shear walls using data from 47 tests. The ANN outperformed empirical models with a higher predictive accuracy. However, a limitation of the study was the small size of the experimental dataset. Zhao et al. [180] applied SVR, ANN, and GPR to predict the maximum deformation of steel-plate composite walls under impact loads. The training data was generated by augmenting the experimental data from 16 composite walls and running a single-degree-of-freedom (SDOF) model to obtain the ground truth of the deformation. GPR was found to be the most effective model, and GPR was further used to optimize the design of a composite wall used in a nuclear power plant to improve the cost-efficiency and impact resistance.

## 5. Challenges and Future Research

### *5.1 Trustworthy ML/AI for SCCS*

Trustworthy ML/AI [181] refers to the development of ML models or AI systems that are ideally reliable, transparent, and unbiased. Trustworthy ML/AI is crucial for the design automation and optimization of SCCSs using ML. The objective is to create robust ML models that can be generalized to new, unseen conditions without introducing large errors. A challenge in achieving trustworthy ML/AI for SCCSs arises from the diversity of datasets. From previous discussions, the datasets have been derived from simulations, experimental studies, or literature, with each set containing unique parameters, configurations, and assumptions. As a result, ML models that are trained and tested on these datasets will have biases, which limit model generalization and applicability. In other words, while ML models may perform well when optimizing SCCS designs under specific conditions, their guidance in actual design implementation remains uncertain. To overcome this limitation, it is essential to establish a comprehensive large database for SCCS design. This database should include a wide range of data covering various components, materials, load types, structural configurations, and performance of SCCS. By integrating data from diverse sources, including experimental results, simulations, and case studies, the large database would provide a more unbiased and robust foundation for training trustworthy ML models. This would allow ML-driven tools to optimize SCCS designs under a broader range of conditions for real-world applications.

### *5.2 Data augmentation*

Experimental data can be costly and time-consuming to collect for SCCSs, thus data augmentation via generative models offers a powerful approach to expand the dataset. By generating additional synthetic data that mimics the characteristics of the real dataset, the model can learn more diverse and complex patterns in the data to improve generalization. Traditionally, multi-fidelity approach and SMOTE have been proved to be effective for structural engineering application. For instance, Chen et al. [182] presented a multi-fidelity approach that used low-fidelity data to enhance the performance of ML models. Liu et. al. [2], Naser and Kodur [183], and Chen et al. [184] employed the SMOTE to augment the available test data [185–188]. Deep learning-based generative models, such as variational autoencoders (VAE), Wasserstein autoencoders (WAE), generative adversarial networks (GANs), are being increasingly applied to generate synthetic and realistic data. For instance, GANs can generate high-quality data through the adversarial training between a generator and a discriminator in an unsupervised manner. The synthetic samples implicitly follow the probability distribution of the real data and are difficult to be distinguished from their real counterparts. As shown in **Fig. 17**, Wang et. al. [8] applied GANs to establish a comprehensive synthetic tubular database containing 5000 samples. The augmented database was used to train ensemble learning models for evaluating the bond strength of reinforcements in 3D-printed concrete. As the next steps, transformers and diffusion models will be promising approaches to augment datasets related to SCCSs.

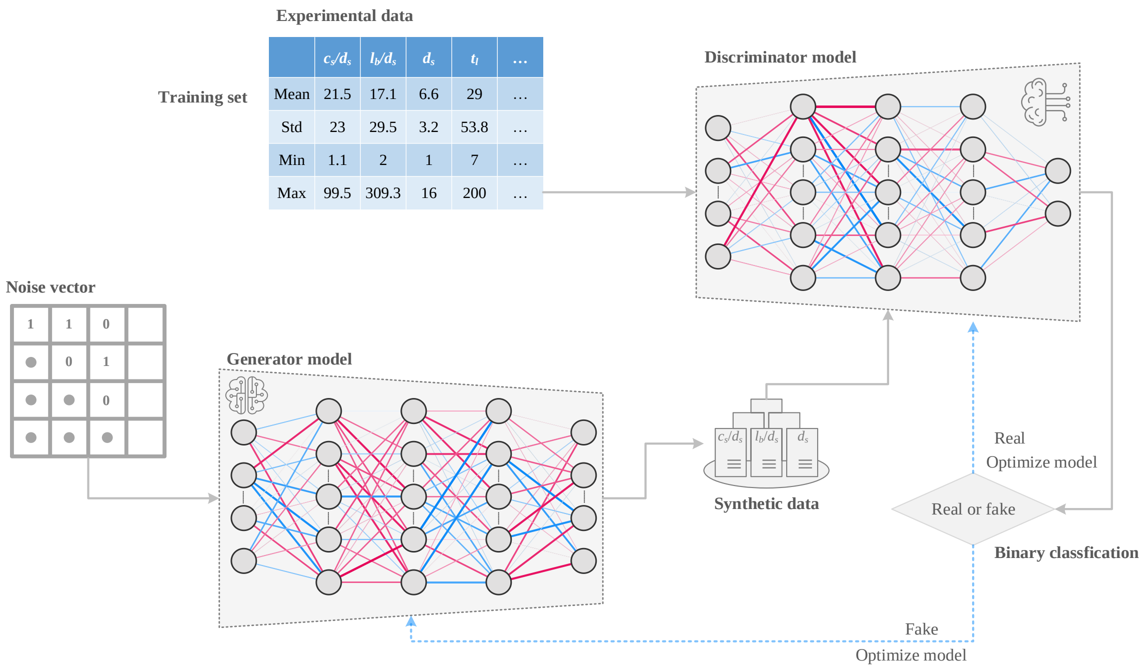


**Fig. 17** Schematic of deep generative adversarial network for data augmentation (adapted from [8]).

## *5.3 Model interpretability*

Model interpretability is crucial as it allows civil engineers to understand, validate, and trust ML models when applying ML to the design of SCCSs. Currently, model-agnostic interpretation methods have been adopted for the ML applications of design, optimization and assessment of SCCSs in terms of model flexibility, explanation flexibility, and representation flexibility. From the global explanation perspective, there is an increasing interest in the employment of permutation feature importance, SHAP method [189], as well as partial dependence plot (PDP) [190]. Permutation feature importance is a model inspection technique that measures the contribution of each feature to a fitted model's statistical performance, and it is particularly useful for non-linear or opaque estimators. It involves randomly shuffling the values of a single feature and observing the resulting degradation of the model's score. For instance, the stud diameter and concrete elastic modulus were quantitatively identified as the dominating features for shear stiffness of headed studs by Wang et. al. [28] through feature importance analysis, as shown in **Fig. 18**a. SHAP is based on the game theory and has been extensively used to explain ML models for SCCSs. For example, it has been used to understand ML models in predicting shear resistance of headed studs (see **Fig. 18**b) and PBLs connectors [2][27][76][78], bonding strength of steel-concrete interface [22][125][130], ACC of CFST [169][171], and NUS of steel-concrete composite slabs [154]. Explaining the interaction of various features on the predicted output through experimental results alone is challenging, but the SHAP feature interaction plot can effectively reveal the dependence between different features. For instance, Wang et al. [27] demonstrated that increasing the strength of concrete beyond a certain point has minimal impact on the resistance of headed studs. This limitation is due to the tensile capacity of the studs, as illustrated in **Fig. 18**c. Moreover, PDP and individual conditional expectation (ICE) plots can illustrate the marginal effect of a specific feature on the predicted target by setting other features as constants. For instance, Setvati and Hicks [191] employed ICE and PDP plots to determine the relationships between seven potential features and the stud resistance. Wang et. al. [8] used this approach to interpret varying tendencies of individual features on the bond strength prediction, as shown in **Fig. 18**d.

From a local explanation perspective, understanding how ML models make specific predictions for SCCSs is crucial for quantifying the impact of key features. SHAP force plot and local interpretable model-agnostic explanations (LIME) plot are two choices to address the local explanation. For instance, the prediction can be broken down into the contributions of each feature to the strength of steel-concrete interfacial bonding, as illustrated in **Fig. 19**a [22]. Similarly, the positive and negative attributes of each feature for flexural capacity assessment of SCCS joints were illustrated by LIME plot, as shown in **Fig. 19**b.

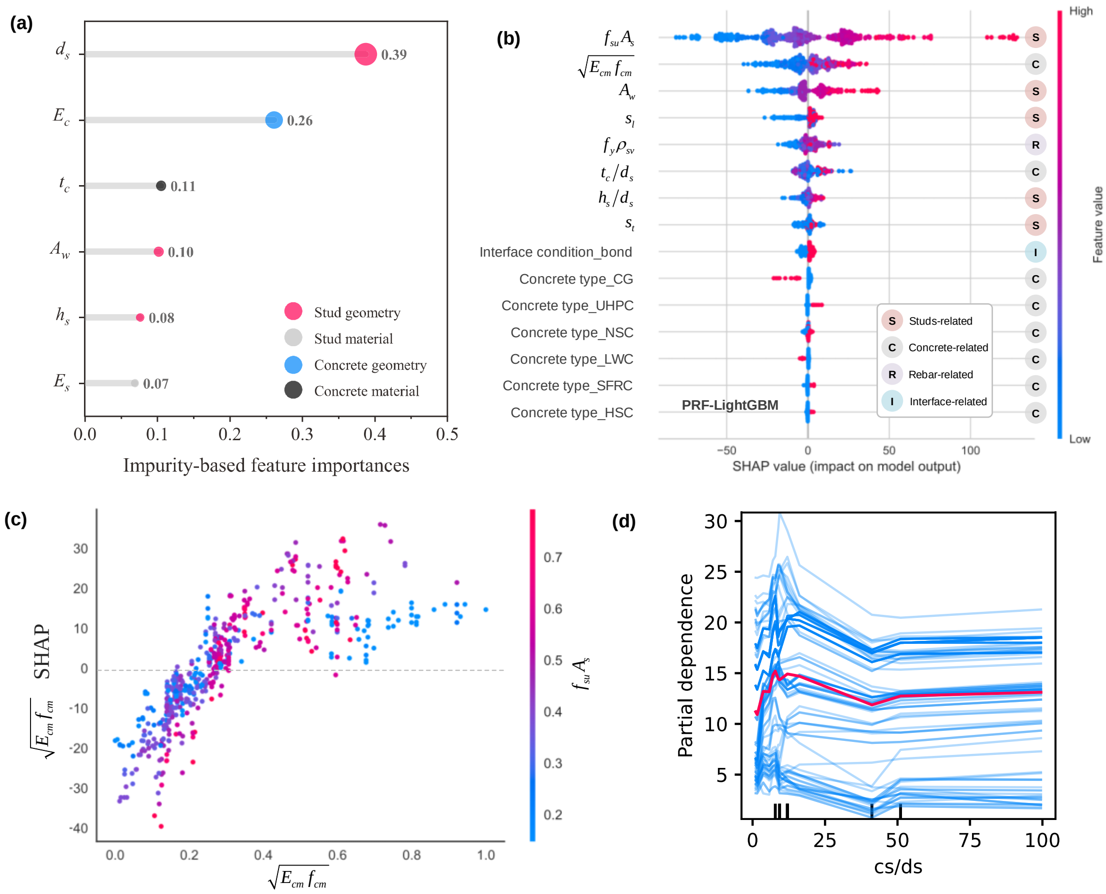


**Fig. 18** Typical model interpretability approaches for global explanation of predictions related to SCCSs: (a) permutation feature importance (adapted from [28]); (b) SHAP summary plot (adapted from [27]); (c) SHAP feature interaction plot (adapted from [27]); (d) ICE and PDP plots (adapted from [8]). Note: CG: concrete grout; UHPC: ultra-high performance concrete; NSC: normal strength concrete; LWC: lightweight concrete; SFRC: steel fiber reinforced concrete; HSC: high strength concrete; SHAP: SHapley additive explanations; ICE: individual conditional expectation; PDP: partial dependence plots.

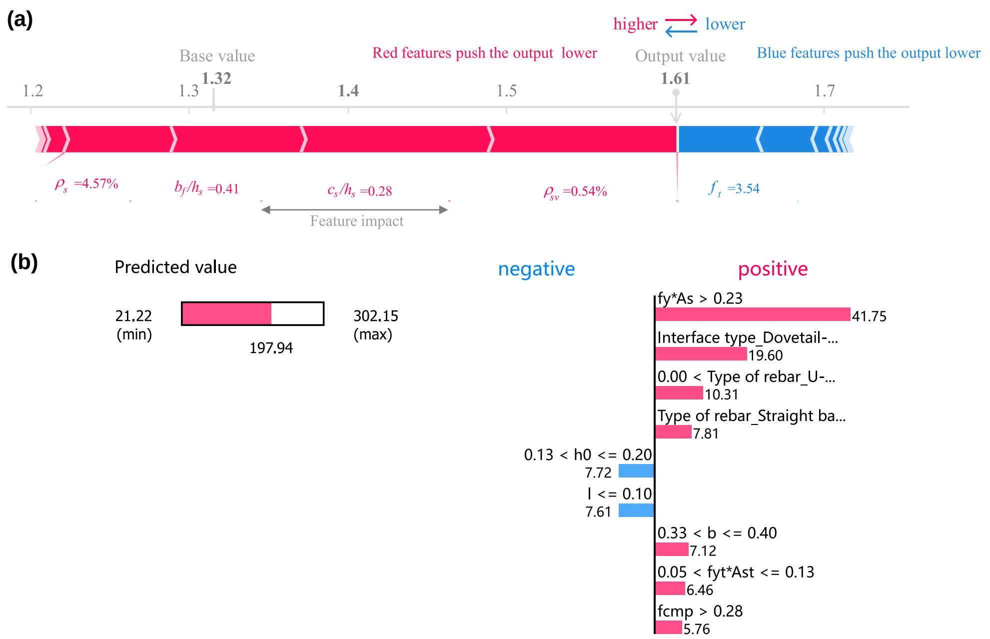


**Fig. 19** Typical model interpretability approaches for local explanation of predictions related to SCCSs: (a) SHAP force plot (adapted from [22]); (b) LIME plot (adapted from [23]). Note: SHapley additive explanations; LIME: local interpretable model-agnostic explanations.

### 5.4 Physics-informed ML

Physics-informed ML combines the strengths of data-driven models and physics-based laws. This approach is particularly useful in fields like SCCSs, where data can be sparse or expensive to collect. By incorporating well-understood physical principles, physics-informed ML may improve model accuracy and reliability. For SCCS applications, research in this area is in its infancy. For instance, Wang et. al. [27] integrated physics-informed knowledge, i.e., the underlying mechanical mechanism, into the ML framwork via feature extraction and combination for shear resistance prediction of headed studs. Ruan et al. [32] encoded the governing equations as a penalty term in the loss function for physics-guided learning for the failure process of precast deck joints in SCCS. In the future, there are several research endeavors for physics-informed ML on SCCS application. First, physical-inspired feature engineering should be studied to integrate physics-informed knowledge via feature extraction and combination. Second, physical laws can be considered as constraints and priors during the learning process, including material properties (e.g., stress-strain relationships, elasticity, deflection, rotation, curvature, among others), conservation of energy, and momentum. Third, partial differential equations can be considered in the loss function to enforce compliance with known physics during training. Last, hybrid ML models that combine physics-based models (e.g., FE model) should be investigated for SCCSs.

### 5.5 Digital-to-real with domain adaptation

Data in structural engineering are typically sourced from simulation models, laboratory testing, or actual field measures. Computational data for SCCSs can be more easily obtained compared with real-world data from physical experiments and field. Therefore, it is more efficient to establish large computational datasets for training ML models on SCCSs. However, digital simulations and real-world environments are two different domains with significant differences due to factors like boundary conditions, material properties, and contact non-linearity. This disparity can compromise the performance of ML models when transitioning from computational data (i.e., the source domain) to real-world data (i.e., the target domain). As a result, ML algorithms trained solely on computational data may not generalize well to real-world data.

To bridge this gap, domain adaptation techniques such as transfer learning [30], domain-adversarial neural networks (DANN) [192], and maximum mean discrepancy (MMD) [193] can be employed to achieve the digital-to-real application for SCCSs. Transfer learning allows ML models to be pre-trained on the computational datasets and then fine-tuned with limited real data. DANN leverages adversarial training to learn domain-invariant features by introducing a domain classifier that discriminates between source and target domains. Therefore, the feature extractor can learn indistinguishable features between the source and target domains. MMD-based methods minimize the statistical distance between source and target feature distributions and align the data in a shared feature space. Moreover, models like cycle-consistent generative adversarial networks (CycleGANs) [194] can be utilized to translate data from the computational domain to the real-world domain with structural uncertainties. In summary, digital-to-real with domain adaptations will enable ML models to be trained in the digital worlds, but still maintain high accuracy when applied to real SCCSs with physical data.

## 6. Conclusions

This paper presents a comprehensive review on the application of ML in the design, optimization, and assessment of SCCSs. Typical ML workflows and models used for SCCSs are discussed. Recent applications of ML on mechanical connectors, steel-concrete interfacial bonding, and various steel-concrete composite elements (i.e., beams, slabs, columns, and walls) are summarized. Key conclusions are as follows:

- The application of ML in SCCSs involves five steps: domain knowledge acquisition, database construction, ML model training and tuning, performance evaluation and interpretive analysis, and cloud deployment and application.
- ML has proven to be successful in the design, optimization, and assessment of SCCSs. Standalone models, hybrid models, and ensemble models have been developed for SCCSs applications. The selection of the models depends on the nature of the problem, the type of the available data, and the desired performance metrics.
- The application of ML for mechanical connectors primarily focuses on the prediction of the shear resistance, shear stiffness and relative slips. ANN, ANFIS, SVM are the most commonly used models. ML methods can make accurate predictions on the strength of steel-concrete interfacial bonding from small (e.g., $cm^2$) to large (e.g., $m^2$) scales under complex conditions such as elevated temperature.
- ML techniques are applicable to aid the design optimization, mechanical behavior prediction, and damage detection of steel-concrete composite beams, slabs, columns, and walls. Additionally, ML applications on composite columns, particularly CFSTs, have attracted growing interest. Research has focused on using ML to predict the ACC of CFST columns with various cross-sections, including circular, elliptical, square, and rectangular shapes.

In summary, this review provides critical insights into the ML applications for design, optimization, and assessment of SCCSs. Future research could focus on (1) trustworthy ML/AI for SCCSs by addressing dataset diversity, (2) data augmentation techniques for SCCSs, (3) improved model interpretability with model-agnostic methods like SHAP, PDP and LIME, (4) physics-informed ML approaches, and (5) domain adaptation techniques to achieve digital-to-real applications.

## References


[1] Wang X, Liu Y, Li Y, Lu Y, Li X. Bond behavior and shear transfer of steel section-concrete interface with studs: Testing and modeling. Constr Build Mater 2020;264. https://doi.org/10.1016/j.conbuildmat.2020.120251.

[2] Liu H, Wang X, Liu Y. Shear performance prediction of perfobond connector using interpretable ensemble learning on unbalanced database. Eng Struct 2024;Under revi.

[3] Wang X, Liu Y, Yang F, Lu Y, Li X. Effect of concrete cover on the bond-slip behavior between steel section and concrete in SRC structures. Constr Build Mater 2019;229:116855. https://doi.org/10.1016/j.conbuildmat.2019.116855.

[4] Zhuang B, Liu Y, Wang D. Shear mechanism of Rubber-Sleeved Stud (RSS) connectors in the steel-concrete interface of cable-pylon composite anchorage. Eng Struct 2020;223:111183. https://doi.org/10.1016/j.engstruct.2020.111183.

[5] Zhuang B, Liu Y. Study on the composite mechanism of large Rubber-Sleeved Stud connector. Constr Build Mater 2019;211:869–84. https://doi.org/10.1016/j.conbuildmat.2019.03.303.

[6] Zhuang B, Liu Y, Yang F. Experimental and numerical study on deformation performance of Rubber-Sleeved Stud connector under cyclic load. Constr Build Mater 2018;192:179–93. https://doi.org/10.1016/j.conbuildmat.2018.10.099.

[7] Wang X, Liu Y, Yang F, Lu Y, Li X. Effect of concrete cover on the bond-slip behavior between steel section and concrete in SRC structures. Constr Build Mater 2019;229. https://doi.org/10.1016/j.conbuildmat.2019.116855.

[8] Wang X, Banthia N, Yoo DY. Reinforcement bond performance in 3D concrete printing: Explainable ensemble learning augmented by deep generative adversarial networks. Autom Constr 2024;158:105164. https://doi.org/10.1016/j.autcon.2023.105164.

[9] Ma X, Wang X, Chen S. Trustworthy machine learning-enhanced 3D concrete printing: Predicting bond strength and designing reinforcement embedment length. Autom Constr 2024;168:105754. https://doi.org/10.1016/j.autcon.2024.105754.

[10] Zhuang B, Arcaro A, Gencturk B, Ghanem R. Machine learning-aided damage identification of mock-up spent nuclear fuel assemblies in a sealed dry storage canister. Eng Appl Artif Intell 2024;128:107484. https://doi.org/10.1016/j.engappai.2023.107484.

[11] Arcaro A, Zhuang B, Gencturk B, Ghanem R. Damage detection and localization in sealed spent nuclear fuel dry storage canisters using multi-task machine learning classifiers. Reliab Eng Syst Saf 2024;252:110446. https://doi.org/10.1016/j.ress.2024.110446.

[12] Li Z, Lan Y, Feng K, Lin W. Investigation of time-varying frequencies of two-axle vehicles and bridges during interaction using drive-by methods and improved multisynchrosqueezing transform. Mech Syst Signal Process 2024;220:111677. https://doi.org/https://doi.org/10.1016/j.ymssp.2024.111677.

[13] Li Z, Lan Y, Lin W. Footbridge damage detection using smartphone-recorded responses of micromobility and convolutional neural networks. Autom Constr 2024;166:105587. https://doi.org/https://doi.org/10.1016/j.autcon.2024.105587.

[14] Salehi H, Burgueño R. Emerging artificial intelligence methods in structural engineering. Eng Struct

2018;171:170–89. https://doi.org/10.1016/j.engstruct.2018.05.084.

[15] Thai HT. Machine learning for structural engineering: A state-of-the-art review. Structures 2022;38:448–91. https://doi.org/10.1016/j.istruc.2022.02.003.

[16] Tapeh ATG, Naser MZ. Artificial Intelligence, Machine Learning, and Deep Learning in Structural Engineering: A Scientometrics Review of Trends and Best Practices. vol. 30. Springer Netherlands; 2023. https://doi.org/10.1007/s11831-022-09793-w.

[17] Nunez I, Marani A, Flah M, Nehdi ML. Estimating compressive strength of modern concrete mixtures using computational intelligence: A systematic review. Constr Build Mater 2021;310:125279. https://doi.org/10.1016/j.conbuildmat.2021.125279.

[18] Li Z, Radlinska A. Artificial intelligencein concrete materials a scientometric view. Leveraging Artif Intell Eng Manag Saf Infrastruct 2022:161–83.

[19] Fan W, Chen Y, Li J, Sun Y, Feng J, Hassanin H, et al. Machine learning applied to the design and inspection of reinforced concrete bridges: Resilient methods and emerging applications. Structures 2021;33:3954–63. https://doi.org/10.1016/j.istruc.2021.06.110.

[20] Sun H, Burton H V., Huang H. Machine learning applications for building structural design and performance assessment: State-of-the-art review. J Build Eng 2021;33:101816. https://doi.org/10.1016/j.jobe.2020.101816.

[21] Pinto G, Wang Z, Roy A, Hong T, Capozzoli A. Transfer learning for smart buildings: A critical review of algorithms, applications, and future perspectives. Adv Appl Energy 2022;5:100084. https://doi.org/10.1016/j.adapen.2022.100084.

[22] Wang X, Chen A, Liu Y. Explainable ensemble learning model for predicting steel section-concrete bond strength. Constr Build Mater 2022;356:129239. https://doi.org/10.1016/j.conbuildmat.2022.129239.

[23] Wang X, Liu Y, Chen A, Ruan X. Flexural capacity assessment of precast deck joints based on deep forest. Structures 2022;41:270–86. https://doi.org/10.1016/j.istruc.2022.05.009.

[24] Xiang D, Liu Y, Shi Y, Xu X. Vertical shear capacity of steel-concrete composite deck slabs with steel ribs. Eng Struct 2022;262:114396. https://doi.org/10.1016/j.engstruct.2022.114396.

[25] Yao Y, Si H, Liu K, Ni T, Gao L. Experimental study and acoustic emission monitoring on damage mechanism of stud shear connectors. Structures 2024;63:106344. https://doi.org/10.1016/j.istruc.2024.106344.

[26] Chow JK, Liu K fu, Tan PS, Su Z, Wu J, Li Z, et al. Automated defect inspection of concrete structures. Autom Constr 2021;132:103959. https://doi.org/10.1016/j.autcon.2021.103959.

[27] Wang X, Liu Y, Chen A, Ruan X. Auto-tuning ensemble models for estimating shear resistance of headed studs in concrete. J Build Eng 2022;52:104470. https://doi.org/10.1016/j.jobe.2022.104470.

[28] Wang X, Liu H, Liu Y. Auto-tuning deep forest for shear stiffness prediction of headed stud connectors. Structures 2022;43:1463–77. https://doi.org/10.1016/j.istruc.2022.07.054.

[29] Wang X, Liu Y, Xin H. Bond strength prediction of concrete-encased steel structures using hybrid machine learning method. Structures 2021;32:2279–92. https://doi.org/10.1016/j.istruc.2021.04.018.

[30] Lin Y zhou, Nie Z hua, Ma H wei. Dynamics-based cross-domain structural damage detection through deep transfer learning. Comput Civ Infrastruct Eng 2022;37:24–54. https://doi.org/10.1111/mice.12692.

[31] Bilotta A, Morassi A, Turco E. Damage identification for steel-concrete composite beams through convolutional neural networks. JVC/Journal Vib Control 2024;30:876–89. https://doi.org/10.1177/10775463231152926.

[32] Ruan X, Zhang J, Wang X. Predicting failure process of precast deck joints using physics-guided LSTM model. Structures 2024;59. https://doi.org/10.1016/j.istruc.2023.105732.

[33] Sadeghi F, Yu Y, Zhu X, Li J. Damage identification of steel-concrete composite beams based on modal strain energy changes through general regression neural network. Eng Struct 2021;244. https://doi.org/10.1016/j.engstruct.2021.112824.

[34] Wang X, Liu Y, Yoo DY. Combined corrosion and inclination effects on pullout behavior of various steel fibers under wet-dry cycle deterioration. Cem Concr Compos 2023;142:105229. https://doi.org/10.1016/j.cemconcomp.2023.105229.

[35] Wang X, Liu Y, Liu S. Effect of chloride-induced corrosion on bond performance of various steel fibers in cracked SFRC. Cem Concr Compos 2023:105113. https://doi.org/https://.org/10.1016/j.cemconcomp.2023.105113.

[36] Guo W, Chen B, Yang Y, Xia Y, Xiao Q, Liu S, et al. Effect of Curing Regimes and Fiber Contents on Flexural Behaviors of Milling Steel Fiber-Reinforced Ultrahigh-Performance Concrete: Experimental and Data-Driven Studies. J Mater Civ Eng 2024;36:4024152. https://doi.org/10.1061/JMCEE7.MTENG-17429.

[37] Yang Y, Chen B, Xia Y, Liu S, Xiao Q, Guo W, et al. Study and data-driven modeling of flexural behaviors of ultra-high-performance concrete reinforced by milling steel fiber. Mech Adv Mater Struct 2023;0:1–14. https://doi.org/10.1080/15376494.2023.2282106.

[38] Hagan MT, Demuth HB, Beale MH, De Jesus O. Neural Network Design. 2nd Editio. USA: Martin T. Hagan; 2014.

[39] Vapnik V. The nature of statistical learning theory. Springer science \& business media; 1999.

[40] Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees. CRC press; 1984.

[41] Ben Chaabene W, Flah M, Nehdi ML. Machine learning prediction of mechanical properties of concrete: Critical review. Constr Build Mater 2020;260:119889. https://doi.org/10.1016/j.conbuildmat.2020.119889.

[42] Kennedy J, Eberhart R. Particle swarm optimization. Proc. ICNN’95 - Int. Conf. Neural Networks, vol. 4, 1995, p. 1942–8 vol.4. https://doi.org/10.1109/ICNN.1995.488968.

[43] Holland JH. Genetic algorithms and adaptation. Adapt Control Ill-Defined Syst 1984:317–33.

[44] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw 2014;69:46–61.

[45] Ly HB, Pham BT, Le LM, Le TT, Le VM, Asteris PG. Estimation of axial load-carrying capacity of concrete-filled steel tubes using surrogate models. Neural Comput Appl 2021;33:3437–58. https://doi.org/10.1007/s00521-020-05214-w.

[46] Ren Q, Li M, Zhang M, Shen Y, Si W. Prediction of ultimate axial capacity of square concrete-filled steel tubular short columns using a hybrid intelligent algorithm. Appl Sci 2019;9. https://doi.org/10.3390/app9142802.

[47] Bardhan A, Biswas R, Kardani N, Iqbal M, Samui P, Singh MP, et al. A novel integrated approach of augmented grey wolf optimizer and ANN for estimating axial load carrying-capacity of concrete-filled steel tube columns. Constr Build Mater 2022;337:127454. https://doi.org/10.1016/j.conbuildmat.2022.127454.

[48] Sarir P, Armaghani DJ, Jiang H, Sabri MMS, He B, Ulrikh DV. Prediction of Bearing Capacity of the Square Concrete-Filled Steel Tube Columns: An Application of Metaheuristic-Based Neural Network Models. Materials (Basel) 2022;15. https://doi.org/10.3390/ma15093309.

[49] Gupta M, Prakash S, Ghani S. Enhancing predictive accuracy: a comprehensive study of optimized machine learning models for ultimate load-carrying capacity prediction in SCFST columns. Asian J Civ Eng 2024;25:3081–98. https://doi.org/10.1007/s42107-023-00964-z.

[50] Thanh Duong H, Chi Phan H, Le TT, Duc Bui N. Optimization design of rectangular concrete-filled steel tube short columns with Balancing Composite Motion Optimization and data-driven model. Structures 2020;28:757–65. https://doi.org/10.1016/j.istruc.2020.09.013.

[51] Le TT, Asteris PG, Lemonis ME. Prediction of axial load capacity of rectangular concrete-filled steel tube columns using machine learning techniques. Springer London; 2021. https://doi.org/10.1007/s00366-021-01461-0.

[52] Le TT, Phan HC, Duong HT, Le MV. Optimal design of circular concrete-filled steel tubular columns based on a combination of artificial neural network, balancing composite motion algorithm and a large experimental database. Expert Syst Appl 2023;223:119940. https://doi.org/10.1016/j.eswa.2023.119940.

[53] Nguyen QH, Ly HB, Tran VQ, Nguyen TA, Phan VH, Le TT, et al. A novel hybrid model based on a feedforward neural network and one step secant algorithm for prediction of load-bearing capacity of rectangular concrete-filled steel tube columns. Molecules 2020;25. https://doi.org/10.3390/molecules25153486.

[54] Ngo NT, Le HA, Pham TPT. Integration of support vector regression and grey wolf optimization for estimating the ultimate bearing capacity in concrete-filled steel tube columns. Neural Comput Appl 2021;33:8525–42. https://doi.org/10.1007/s00521-020-05605-z.

[55] Ngo NT, Pham TPT, Le HA, Nguyen QT, Nguyen TTN. Axial strength prediction of steel tube confined concrete columns using a hybrid machine learning model. Structures 2022;36:765–80. https://doi.org/10.1016/j.istruc.2021.12.054.

[56] Naser MZ, Thai S, Thai HT. Evaluating structural response of concrete-filled steel tubular columns through machine learning. J Build Eng 2021;34:101888. https://doi.org/10.1016/j.jobe.2020.101888.

[57] Mai SH, Ben Seghier MEA, Nguyen PL, Jafari-Asl J, Thai DK. A hybrid model for predicting the axial compression capacity of square concrete-filled steel tubular columns. Eng Comput 2020. https://doi.org/10.1007/s00366-020-01104-w.

[58] Freund Y, Schapire RE. Experiments with a New Boosting Algorithm. Proc 13th Int Conf Mach Learn 1996:148–156. https://doi.org/10.1.1.133.1040.

[59] Breiman L. Random Forests. Mach Learn 2001;45:5–32. https://doi.org/10.1109/ICCECE51280.2021.9342376.

[60] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, et al. LightGBM: A highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst 2017;2017-Decem:3147–55.

[61] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. Proc ACM SIGKDD Int Conf Knowl Discov Data Min 2016;13-17-Augu:785–94. https://doi.org/10.1145/2939672.2939785.

[62] Lee S, Vo TP, Thai HT, Lee J, Patel V. Strength prediction of concrete-filled steel tubular columns using

Categorical Gradient Boosting algorithm. Eng Struct 2021;238:112109. https://doi.org/10.1016/j.engstruct.2021.112109.

[63] Feng D-C, Cetiner B, Azadi Kakavand MR, Taciroglu E. Data-Driven Approach to Predict the Plastic Hinge Length of Reinforced Concrete Columns and Its Application. J Struct Eng 2021;147:04020332. https://doi.org/10.1061/(asce)st.1943-541x.0002852.

[64] Hutter F, Hoos HH, Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 2011;6683 LNCS:507–23. https://doi.org/10.1007/978-3-642-25566-3_40.

[65] Martínez-Muñoz D, García J, Martí J V., Yepes V. Deep learning classifier for life cycle optimization of steel–concrete composite bridges. Structures 2023;57. https://doi.org/10.1016/j.istruc.2023.105347.

[66] Ibrahim Bibi Farouk A, Zhu J, Ding J, Haruna SI. Prediction and uncertainty quantification of ultimate bond strength between UHPC and reinforcing steel bar using a hybrid machine learning approach. Constr Build Mater 2022;345:128360. https://doi.org/10.1016/j.conbuildmat.2022.128360.

[67] Tran VL, Thai DK, Kim SE. Application of ANN in predicting ACC of SCFST column. Compos Struct 2019;228:111332. https://doi.org/10.1016/j.compstruct.2019.111332.

[68] You X, Yan G, Al-Masoudy MM, Kadimallah MA, Alkhalifah T, Alturise F, et al. Application of novel hybrid machine learning approach for estimation of ultimate bond strength between ultra-high performance concrete and reinforced bar. Adv Eng Softw 2023;180:103442. https://doi.org/10.1016/j.advengsoft.2023.103442.

[69] Reshi IA, Shah AH, Jan A, Tariq Z, Sholla S, Rashid S, et al. Machine learning enhanced modeling of steel-concrete bond strength under elevated temperature exposure. Struct Concr 2024:1–14. https://doi.org/10.1002/suco.202400334.

[70] Almasaeid HH, Salman DG, Abendeh RM, Allouzi RA, Rabayah HS. Interfacial bond capacity prediction of concrete-filled steel tubes utilizing artificial neural network. Cogent Eng 2024;11. https://doi.org/10.1080/23311916.2023.2297501.

[71] Wang X, Liu Y, Lu Y, Li X. Shear transfer mechanism of perforated web connection for concrete encased steel structures. Eng Struct 2022;252:113418. https://doi.org/10.1016/j.engstruct.2021.113418.

[72] Degtyarev V V., Hicks SJ, Hajjar JF. Design models for predicting shear resistance of studs in solid concrete slabs based on symbolic regression with genetic programming. Steel Compos Struct 2022;43:293–309. https://doi.org/10.12989/scs.2022.43.3.293.

[73] Abambres M, He J. Shear Capacity of Headed Studs in Steel-Concrete Structures: Analytical Prediction via Soft Computing. SSRN Electron J 2019. https://doi.org/10.2139/ssrn.3368670.

[74] Avci-Karatas C. Application of Machine Learning in Prediction of Shear Capacity of Headed Steel Studs in Steel–Concrete Composite Structures. Int J Steel Struct 2022;22:539–56. https://doi.org/10.1007/s13296-022-00589-z.

[75] Razavi Setvati M, Hicks SJ. Machine learning models for predicting resistance of headed studs embedded in concrete. Eng Struct 2022;254:113803. https://doi.org/10.1016/j.engstruct.2021.113803.

[76] Degtyarev V V., Hicks SJ. Machine learning-based probabilistic predictions of shear resistance of welded studs

in deck slab ribs transverse to beams. Steel Compos Struct 2023;49:109–23. https://doi.org/10.12989/scs.2023.49.1.109.

[77] Yosri AM, Farouk AIB, Haruna SI, Deifalla A farouk, Shaaban WM. Sensitivity and robustness analysis of adaptive neuro-fuzzy inference system (ANFIS) for shear strength prediction of stud connectors in concrete. Case Stud Constr Mater 2023;18:e02096. https://doi.org/10.1016/j.cscm.2023.e02096.

[78] Zhang F, Wang C, Zou X, Wei Y, Chen D, Wang Q, et al. Prediction of the Shear Resistance of Headed Studs Embedded in Precast Steel–Concrete Structures Based on an Interpretable Machine Learning Method. Buildings 2023;13. https://doi.org/10.3390/buildings13020496.

[79] Zhu J, Ibrahim Bibi Farouk A. Development of hybrid models for shear resistance prediction of grouped stud connectors in concrete using improved metaheuristic optimization techniques. Structures 2023;50:286–302. https://doi.org/10.1016/j.istruc.2023.02.040.

[80] Guang C. Shear capacity evaluation of studs in steel-high strength concrete composite structures. Appl Eng Sci 2024;17:100150. https://doi.org/10.1016/j.apples.2023.100150.

[81] Roh GT, Vu N, Jeon CH, Shim CS. Augmented Data-Driven Machine Learning for Digital Twin of Stud Shear Connections. Buildings 2024;14. https://doi.org/10.3390/buildings14020328.

[82] Degtyarev V V., Hicks SJ. Reliability-based design shear resistance of headed studs in solid slabs predicted by machine learning models. Archit Struct Constr 2023;3:447–73. https://doi.org/10.1007/s44150-022-00078-1.

[83] Sun G, Kang J, Shi J. Application of Machine Learning Models and Gsa Method for Designing Stud Connectors. J Civ Eng Manag 2024;30:373–90. https://doi.org/10.3846/jcem.2024.21348.

[84] Li Q, Luo A, Hong C, Wang G, Yin X, Xu S. Fatigue behavior of short-headed studs embedded in Ultra-high Toughness Cementitious Composites (UHTCC). Eng Struct 2024;300:117194. https://doi.org/10.1016/j.engstruct.2023.117194.

[85] Roshanfar M, Ghiami Azad AR, Forouzanfar M. Predicting fatigue life of shear connectors in steel-concrete composite bridges using artificial intelligence techniques. Fatigue Fract Eng Mater Struct 2024;47:818–32. https://doi.org/10.1111/ffe.14207.

[86] Degtyarev V V., Hicks SJ. Reliability-based design shear resistance of headed studs in solid slabs predicted by machine learning models. Archit Struct Constr 2023;3:447–73. https://doi.org/10.1007/s44150-022-00078-1.

[87] Wei X, Li Y. New Assessment method on shear resistance of perfobond shear connectors in steel-concrete composite structure. IABSE Conf Guangzhou 2016 Bridg Struct Sustain - Seek Intell Solut - Rep 2016:654–9. https://doi.org/10.2749/222137816819258979.

[88] Allahyari H, M. Nikbin I, Rahimi R. S, Heidarpour A. A new approach to determine strength of Perfobond rib shear connector in steel-concrete composite structures by employing neural network. Eng Struct 2018;157:235–49. https://doi.org/10.1016/j.engstruct.2017.12.007.

[89] Chen Y, Huang Y, Liu H, Liu Y, Zhang T. Ultimate bearing capacity prediction method and sensitivity analysis of PBL. Eng Appl Artif Intell 2023;123:106510. https://doi.org/10.1016/j.engappai.2023.106510.

[90] Khalaf JA, Majeed AA, Aldlemy MS, Ali ZH, Al Zand AW, Adarsh S, et al. Hybridized Deep Learning Model for Perfobond Rib Shear Strength Connector Prediction. Complexity 2021;2021.

https://doi.org/10.1155/2021/6611885.
[91] Chen Y, Zhang J, Liu Y, Zhao S, Zhou S, Chen J. Research on the prediction method of ultimate bearing capacity of PBL Based on IAGA-BPNN Algorithm. IEEE Access 2020;8:179141–55. https://doi.org/10.1109/ACCESS.2020.3026091.
[92] Li H, Yin X, Sha L, Yang D, Hu T. Data-Driven Prediction Model for High-Strength Bolts in Composite Beams. Buildings 2023;13. https://doi.org/10.3390/buildings13112769.
[93] Hosseinpour M, Daei M, Zeynalian M, Ataei A. Neural networks-based formulation for predicting ultimate strength of bolted shear connectors in composite cold-formed steel beams. Eng Appl Artif Intell 2023;118:105614. https://doi.org/10.1016/j.engappai.2022.105614.
[94] Saleem M. Assessing the load carrying capacity of concrete anchor bolts using non-destructive tests and artificial multilayer neural network. J Build Eng 2020;30:101260. https://doi.org/10.1016/j.jobe.2020.101260.
[95] Olalusi OB, Spyridis P. Machine learning-based models for the concrete breakout capacity prediction of single anchors in shear. Adv Eng Softw 2020;147:102832. https://doi.org/10.1016/j.advengsoft.2020.102832.
[96] Shariati M, Mafipour MS, Mehrabi P, Bahadori A, Zandi Y, Salih MNA, et al. Application of a Hybrid Artificial Neural Model in Behavior Prediction of Channel Shear Connectors Embedded in Normal and High-Strength Concrete. Appl Sci 2019;9:5534.
[97] Sadeghipour Chahnasir E, Zandi Y, Shariati M, Dehghani E, Toghroli A, Tonnizam Mohamad E, et al. Application of support vector machine with firefly algorithm for investigation of the factors affecting the shear strength of angle shear connectors. Smart Struct Syst 2018;22:413–24. https://doi.org/10.12989/sss.2018.22.4.413.
[98] Shariati M, Mafipour MS, Mehrabi P, Shariati A, Toghroli A, Trung NT, et al. A novel approach to predict shear strength of tilted angle connectors using artificial intelligence techniques. Eng Comput 2021;37:2089–109. https://doi.org/10.1007/s00366-019-00930-x.
[99] Vijayakumar R, Pannirselvam N. Multi-objective optimisation of mild steel embossed plate shear connector using artificial neural network-integrated genetic algorithm. Case Stud Constr Mater 2022;17:e01560. https://doi.org/10.1016/j.cscm.2022.e01560.
[100] Xiong Z, Liang Z, Liu X, Feldmann M. Steel-UHPC composite dowels' pull-out performance studies using machine learning algorithms. Steel Compos Struct 2023;48:531–45. https://doi.org/10.12989/scs.2023.48.5.531.
[101] Wang X, Li W, Liu Y, Yoo D. Bond performance of reinforcing bars in SFRC : Experiments and meso-mechanical model. Compos Struct 2023;318:117092. https://doi.org/10.1016/j.compstruct.2023.117092.
[102] Wang X, Liu Y, Yoo DY. Bond deterioration of corroded reinforcements in SFRC: Experiments and 3D laser scanning. Cem Concr Compos 2023;137:104946. https://doi.org/10.1016/j.cemconcomp.2023.104946.
[103] Zhang S, Xu J, Lai T, Yu Y, Xiong W. Bond stress estimation of profiled steel-concrete in steel reinforced concrete composite structures using ensemble machine learning approaches. Eng Struct 2023;294:116725. https://doi.org/10.1016/j.engstruct.2023.116725.
[104] Gupta M, Prakash S, Ghani S, Kumar N, Saharan S. Enhancing bond performance in SRC structures: a computational approach using ensemble learning techniques and sequential analysis. Asian J Civ Eng

2024;25:3329–47. https://doi.org/10.1007/s42107-023-00982-x.
[105] Yu Y, Xie T, Xu J, Lai Z, Elchalakani M. Probabilistic models for characteristic bond stresses of steel-concrete in steel reinforced concrete structures. Eng Struct 2024;300:117167. https://doi.org/10.1016/j.engstruct.2023.117167.
[106] Allouzi RA, Almasaeid HH, Salman DG, Abendeh RM, Rabayah HS. Prediction of Bond-Slip Behavior of Circular/Squared Concrete-Filled Steel Tubes. Buildings 2022;12. https://doi.org/10.3390/buildings12040456.
[107] Cao H, Li J, Chen X. Investigation of interfacial debonding identification for concrete filled steel tube columns based on Investigation of interfacial debonding identification for concrete filled steel tube columns based on acoustic signals. Measurement 2024:115511. https://doi.org/10.1016/j.measurement.2024.115511.
[108] Li Y, Yue Q, Li H, Gan S, Zhu J, Chen H. Multi-damage index-based interfacial debonding prediction for steel-concrete composite structures with percussion method. J Build Eng 2024;94:109964. https://doi.org/10.1016/j.jobe.2024.109964.
[109] Dahou Z, Mehdi Sbartaï Z, Castel A, Ghomari F. Artificial neural network model for steel-concrete bond prediction. Eng Struct 2009;31:1724–33. https://doi.org/10.1016/j.engstruct.2009.02.010.
[110] Makni M, Daoud A, Karray MA, Lorrain M. Artificial neural network for the prediction of the steel-concrete bond behaviour. Eur J Environ Civ Eng 2014;18:862–81. https://doi.org/10.1080/19648189.2014.909745.
[111] Mahjoubi S, Meng W, Bao Y. Logic-guided neural network for predicting steel-concrete interfacial behaviors. Expert Syst Appl 2022;198:116820. https://doi.org/10.1016/j.eswa.2022.116820.
[112] Golafshani EM, Rahai A, Sebt MH, Akbarpour H. Prediction of bond strength of spliced steel bars in concrete using artificial neural network and fuzzy logic. Constr Build Mater 2012;36:411–8. https://doi.org/10.1016/j.conbuildmat.2012.04.046.
[113] Hwang HJ, Baek JW, Kim JY, Kim CS. Prediction of bond performance of tension lap splices using artificial neural networks. Eng Struct 2019;198:109535. https://doi.org/10.1016/j.engstruct.2019.109535.
[114] Ahmad MS, Adnan SM, Zaidi S, Bhargava P. A novel support vector regression (SVR) model for the prediction of splice strength of the unconfined beam specimens. Constr Build Mater 2020;248:118475. https://doi.org/10.1016/j.conbuildmat.2020.118475.
[115] Yaseen ZM, Keshtegar B, Hwang HJ, Nehdi ML. Predicting reinforcing bar development length using polynomial chaos expansions. Eng Struct 2019;195:524–35. https://doi.org/10.1016/j.engstruct.2019.06.012.
[116] Hoang ND, Tran XL, Nguyen H. Predicting ultimate bond strength of corroded reinforcement and surrounding concrete using a metaheuristic optimized least squares support vector regression model. Neural Comput Appl 2020;32:7289–309. https://doi.org/10.1007/s00521-019-04258-x.
[117] Concha NC, Oreta AWC. Investigation of the effects of corrosion on bond strength of steel in concrete using neural network. Comput Concr 2021;28:77–91. https://doi.org/10.12989/cac.2021.28.1.077.
[118] Ben Seghier MEA, Ouaer H, Ghriga MA, Menad NA, Thai DK. Hybrid soft computational approaches for modeling the maximum ultimate bond strength between the corroded steel reinforcement and surrounding concrete. Neural Comput Appl 2021;33:6905–20. https://doi.org/10.1007/s00521-020-05466-6.
[119] Owusu-Danquah JS, Bseiso A, Allena S, Duffy SF. Artificial neural network algorithms to predict the bond

strength of reinforced concrete: Coupled effect of corrosion, concrete cover, and compressive strength. Constr Build Mater 2022;350:128896. https://doi.org/10.1016/j.conbuildmat.2022.128896.

[120] Huang T, Liu T, Xu N, Yue K, Li Y, Liu X, et al. A data-driven approach for predicting interface bond strength between corroded reinforcement and concrete. Structures 2023;57:105122. https://doi.org/10.1016/j.istruc.2023.105122.

[121] Cavaleri L, Barkhordari MS, Repapis CC, Armaghani DJ, Ulrikh DV, Asteris PG. Convolution-based ensemble learning algorithms to estimate the bond strength of the corroded reinforced concrete. Constr Build Mater 2022;359:129504. https://doi.org/10.1016/j.conbuildmat.2022.129504.

[122] Zhang C, Chun Q, Sun A, Lin Y, Wang HY. Improved Meta-learning Neural Network for the Prediction of the Historical Reinforced Concrete Bond–Slip Model Using Few Test Specimens. Int J Concr Struct Mater 2022;16. https://doi.org/10.1186/s40069-022-00530-y.

[123] Fu B, Chen SZ, Liu XR, Feng DC. A probabilistic bond strength model for corroded reinforced concrete based on weighted averaging of non-fine-tuned machine learning models. Constr Build Mater 2022;318. https://doi.org/10.1016/j.conbuildmat.2021.125767.

[124] Wang KL, Li J, Li L, Deng E, Zhang Y, Li Z. Machine learning to estimate the bond strength of the corroded steel bar-concrete. Struct Concr 2024;25:696–715. https://doi.org/10.1002/suco.202300401.

[125] Wakjira TG, Abushanab A, Alam MS, Alnahhal W, Plevris V. Explainable machine learning-aided efficient prediction model and software tool for bond strength of concrete with corroded reinforcement. Structures 2024;59:105693. https://doi.org/10.1016/j.istruc.2023.105693.

[126] Mei Y, Sun Y, Li F, Xu X, Zhang A, Shen J. Probabilistic prediction model of steel to concrete bond failure under high temperature by machine learning. Eng Fail Anal 2022;142:106786. https://doi.org/10.1016/j.engfailanal.2022.106786.

[127] Nematzadeh M, Shahmansouri AA, Zabihi R. Innovative models for predicting post-fire bond behavior of steel rebar embedded in steel fiber reinforced rubberized concrete using soft computing methods. Structures 2021;31:1141–62. https://doi.org/10.1016/j.istruc.2021.02.015.

[128] Tanyildizi H. Fuzzy logic model for the prediction of bond strength of high-strength lightweight concrete. Adv Eng Softw 2009;40:161–9. https://doi.org/10.1016/j.advengsoft.2007.05.013.

[129] Li L, Guo Y, Zhang Y, Xu K, Wang X. Bond strength between recycled aggregate concrete and rebar: Interpretable machine learning modeling approach for performance estimation and engineering design. Mater Today Commun 2024;39. https://doi.org/10.1016/j.mtcomm.2024.108945.

[130] Sun Y. Forecasting ultimate bond strength between ribbed stainless steel bar and concrete using explainable machine learning algorithms. Multidiscip Model Mater Struct 2024;20:401–16. https://doi.org/10.1108/MMMS-09-2023-0298.

[131] Li X, Qin Z, Zheng D, Zhang X, Li H. Reversed bond-slip model of deformed bar embedded in concrete based on ensemble learning algorithm. J Build Eng 2023;68:106081. https://doi.org/10.1016/j.jobe.2023.106081.

[132] Martínez-Muñoz D, García J, Martí J V., Yepes V. Discrete swarm intelligence optimization algorithms applied to steel–concrete composite bridges. Eng Struct 2022;266. https://doi.org/10.1016/j.engstruct.2022.114607.

[133] Martínez-Muñoz D, García J, Martí J V., Yepes V. Hybrid Swarm Intelligence Optimization Methods for Low-Embodied Energy Steel-Concrete Composite Bridges. Mathematics 2023;11. https://doi.org/10.3390/math11010140.

[134] de Oliveira VM, de Carvalho AS, Rossi A, Hosseinpour M, Sharifi Y, Martins CH. Data-driven design approach for the lateral-distortional buckling in steel-concrete composite cellular beams using machine learning models. Structures 2024;61. https://doi.org/10.1016/j.istruc.2024.106018.

[135] Mastan S, Anandh S, Sindhu Nachiar S. Numerical method and validation using ANN of composite steel–concrete beam for optimized geometry and emplacement of web opening. Asian J Civ Eng 2024;25:1539–59. https://doi.org/10.1007/s42107-023-00860-6.

[136] Ferreira FPV, Jeong S-H, Mansouri E, Shamass R, Tsavdaridis KD, Martins CH, et al. Five Machine Learning Models Predicting the Global Shear Capacity of Composite Cellular Beams with Hollow-Core Units. Buildings 2024;14:2256. https://doi.org/10.3390/buildings14072256.

[137] Hosseinpour M, Rossi A, Sander Clemente de Souza A, Sharifi Y. New predictive equations for LDB strength assessment of steel–concrete composite beams. Eng Struct 2022;258. https://doi.org/10.1016/j.engstruct.2022.114121.

[138] Kumar S, Patel KA, Chaudhary S, Nagpal AK. Rapid Prediction of Long-term Deflections in Steel-Concrete Composite Bridges Through a Neural Network Model. Int J Steel Struct 2021;21:590–603. https://doi.org/10.1007/s13296-021-00458-1.

[139] Kumar P, Kumar A, Kumar S, Ranjan R, Kumar P. Bending behaviour of steel–concrete composite beam with partial shear interface using MCS and ANN. Acta Mech 2024;235:4451–71. https://doi.org/10.1007/s00707-024-03949-4.

[140] Thirumalaiselvi A, Verma M, Anandavalli N, Rajasankar J. Response prediction of laced steel-concrete composite beams using machine learning algorithms. Struct Eng Mech 2018;66:399–409. https://doi.org/10.12989/sem.2018.66.3.399.

[141] Xiong Z, Li J, Zhu H, Liu X, Liang Z. Ultimate Bending Strength Evaluation of MVFT Composite Girder by using Finite Element Method and Machine Learning Regressors. Lat Am J Solids Struct 2022;19. https://doi.org/10.1590/1679-78257006.

[142] Tan ZX, Thambiratnam DP, Chan THT, Gordan M, Abdul Razak H. Damage detection in steel-concrete composite bridge using vibration characteristics and artificial neural network. Struct Infrastruct Eng 2020;16:1247–61. https://doi.org/10.1080/15732479.2019.1696378.

[143] Guo Z, Bu J, Zhang J, Cao W, Huang X. Theoretical and Numerical Investigation of Damage Sensitivity of Steel–Concrete Composite Beam Bridges. Buildings 2023;13. https://doi.org/10.3390/buildings13051109.

[144] Cheng L, Xin H, Groves RM, Veljkovic M. Acoustic emission source location using Lamb wave propagation simulation and artificial neural network for I-shaped steel girder. Constr Build Mater 2021;273. https://doi.org/10.1016/j.conbuildmat.2020.121706.

[145] Zhang C, Shi J, Huang C. Identification of Damage in Steel-Concrete Composite Beams Based on Wavelet Analysis and Deep Learning. SDHM Struct Durab Heal Monit 2024;18:465–83.

https://doi.org/10.32604/sdhm.2024.048705.

[146] Li D, Nie JH, Wang H, Yan JB, Hu CX, Shen P. Damage location, quantification and characterization of steel-concrete composite beams using acoustic emission. Eng Struct 2023;283. https://doi.org/10.1016/j.engstruct.2023.115866.

[147] Chu Y, Zhang Y, Li S, Ma Y, Yang S. A machine learning approach for identifying vertical temperature gradient in steel-concrete composite beam under solar radiation. Adv Eng Softw 2024;196. https://doi.org/10.1016/j.advengsoft.2024.103695.

[148] Morasaei A, Ghabussi A, Aghlmand S, Yazdani M, Baharom S, Assilzadeh H. Simulation of steel–concrete composite floor system behavior at elevated temperatures via multi-hybrid metaheuristic framework. Eng Comput 2022;38:2567–82. https://doi.org/10.1007/s00366-020-01228-z.

[149] Panev Y, Kotsovinos P, Deeny S, Flint G. The Use of Machine Learning for the Prediction of fire Resistance of Composite Shallow Floor Systems. Fire Technol 2021;57:3079–100. https://doi.org/10.1007/s10694-021-01108-y.

[150] Shariati M, Mafipour MS, Mehrabi P, Zandi Y, Dehghani D, Bahadori A, et al. Application of Extreme Learning Machine (ELM) and Genetic Programming (GP) to design steel-concrete composite floor systems at elevated temperatures. Steel Compos Struct 2019;33:319–32. https://doi.org/10.12989/scs.2019.33.3.319.

[151] Karthiga S, Umamaheswari N. Prediction of displacement of composite slab with profiled steel deck using artificial neural network. Asian J Civ Eng 2024;25:4179–96. https://doi.org/10.1007/s42107-024-01040-w.

[152] Zhang R, Jiao J, He M, Tao Z, He P. Design, implementation and performance prediction of profiled steel sheet-mixed aggregate recycled concrete hollow composite slab. J Build Eng 2023;79. https://doi.org/10.1016/j.jobe.2023.107839.

[153] Zhou Y, Liang M, Yue X. Deep residual learning for acoustic emission source localization in A steel-concrete composite slab. Constr Build Mater 2024;411. https://doi.org/10.1016/j.conbuildmat.2023.134220.

[154] Wang S, Liu J, Wang Q, Dai R, Chen K. Prediction of non-uniform shrinkage of steel-concrete composite slabs based on explainable ensemble machine learning model. J Build Eng 2024;88. https://doi.org/10.1016/j.jobe.2024.109002.

[155] Ahmadi M, Naderpour H, Kheyroddin A. Utilization of artificial neural networks to prediction of the capacity of CCFT short columns subject to short term axial load. Arch Civ Mech Eng 2014;14:510–7. https://doi.org/10.1016/j.acme.2014.01.006.

[156] Jegadesh S, Jayalekshmi S. Application of Artificial Neural Network for Calculation of Axial Capacity of Circular Concrete Filled Steel Tubular Columns. Int J Earth Sci Eng 2015;8:35–42.

[157] Tran VL, Thai DK, Nguyen DD. Practical artificial neural network tool for predicting the axial compression capacity of circular concrete-filled steel tube columns with ultra-high-strength concrete. Thin-Walled Struct 2020;151:106720. https://doi.org/10.1016/j.tws.2020.106720.

[158] Ho NX, Le TT. Effects of variability in experimental database on machine-learning-based prediction of ultimate load of circular concrete-filled steel tubes. Meas J Int Meas Confed 2021;176:109198. https://doi.org/10.1016/j.measurement.2021.109198.

[159] Vu QV, Truong VH, Thai HT. Machine learning-based prediction of CFST columns using gradient tree boosting algorithm. Compos Struct 2021;259:113505. https://doi.org/10.1016/j.compstruct.2020.113505.

[160] Le TT. Practical Hybrid Machine Learning Approach for Estimation of Ultimate Load of Elliptical Concrete-Filled Steel Tubular Columns under Axial Loading. Adv Civ Eng 2020;2020. https://doi.org/10.1155/2020/8832522.

[161] Le TT, Le MV. Development of user-friendly kernel-based Gaussian process regression model for prediction of load-bearing capacity of square concrete-filled steel tubular members. Mater Struct Constr 2021;54:1–24. https://doi.org/10.1617/s11527-021-01646-5.

[162] Ben Seghier MEA, Gao XZ, Jafari-Asl J, Thai DK, Ohadi S, Trung NT. Modeling the nonlinear behavior of ACC for SCFST columns using experimental-data and a novel evolutionary-algorithm. Structures 2021;30:692–709. https://doi.org/10.1016/j.istruc.2021.01.036.

[163] Le TT. Practical machine learning-based prediction model for axial capacity of square CFST columns. Mech Adv Mater Struct 2022;29:1782–97. https://doi.org/10.1080/15376494.2020.1839608.

[164] Du Y, Chen Z, Zhang C, Cao X. Research on axial bearing capacity of rectangular concrete-filled steel tubular columns based on artificial neural networks. Front Comput Sci 2017;11:863–73. https://doi.org/10.1007/s11704-016-5113-6.

[165] Cakiroglu C, Islam K, Bekdaş G, Isikdag U, Mangalathu S. Explainable machine learning models for predicting the axial compression capacity of concrete filled steel tubular columns. Constr Build Mater 2022;356. https://doi.org/10.1016/j.conbuildmat.2022.129227.

[166] Duong TH, Le TT, Le M V. Practical Machine Learning Application for Predicting Axial Capacity of Composite Concrete-Filled Steel Tube Columns Considering Effect of Cross-Sectional Shapes. Int J Steel Struct 2023;23:263–78. https://doi.org/10.1007/s13296-022-00693-0.

[167] Zarringol M, Thai HT, Thai S, Patel V. Application of ANN to the design of CFST columns. Structures 2020;28:2203–20. https://doi.org/10.1016/j.istruc.2020.10.048.

[168] Lee S, Vo TP, Thai HT, Lee J, Patel V. Strength prediction of concrete-filled steel tubular columns using Categorical Gradient Boosting algorithm. Eng Struct 2021;238:112109. https://doi.org/10.1016/j.engstruct.2021.112109.

[169] Degtyarev V V., Thai HT. Design of concrete-filled steel tubular columns using data-driven methods. J Constr Steel Res 2023;200:107653. https://doi.org/10.1016/j.jcsr.2022.107653.

[170] de Carvalho AS, Rossi A, Morkhade SG, Martins CH. Machine Learning-Based Design Approach for Concrete-Filled Stainless Steel Tubular Columns. Arab J Sci Eng 2023;48:14105–18. https://doi.org/10.1007/s13369-023-08090-3.

[171] Chen K, Wang S, Wang Y, Wei J, Wang Q, Du W, et al. Intelligent design of limit states for recycled aggregate concrete filled steel tubular columns. Structures 2023;58:105338. https://doi.org/10.1016/j.istruc.2023.105338.

[172] Lusong Y, Yuxing Z, Li W, Qiren P, Yiyang W. Prediction of the Axial Bearing Compressive Capacities of CFST Columns Based on Machine Learning Methods. Int J Steel Struct 2024;24:81–94. https://doi.org/10.1007/s13296-023-00800-9.

[173] Nguyen HQ, Ly HB, Tran VQ, Nguyen TA, Le TT, Pham BT. Optimization of artificial intelligence system by evolutionary algorithm for prediction of axial capacity of rectangular concrete filled steel tubes under compression. Materials (Basel) 2020;13. https://doi.org/10.3390/MA13051205.

[174] Lyu F, Fan X, Ding F, Chen Z. Prediction of the Axial Compressive Strength of Circular Concrete-Filled Steel Tube Columns using Sine Cosine Algorithm-Support Vector Regression. Compos Struct 2021:114282. https://doi.org/10.1016/j.compstruct.2021.114282.

[175] Avci-Karatas C. Artificial Neural Network (ANN) Based Prediction of Ultimate Axial Load Capacity of Concrete-Filled Steel Tube Columns (CFSTCs). Int J Steel Struct 2022;22:1341–58. https://doi.org/10.1007/s13296-022-00645-8.

[176] Memarzadeh A, Sabetifar H, Nematzadeh M. A comprehensive and reliable investigation of axial capacity of Sy-CFST columns using machine learning-based models. Eng Struct 2023;284:115956. https://doi.org/10.1016/j.engstruct.2023.115956.

[177] Deng C, Xue X, Tao L. Prediction of ultimate bearing capacity of concrete filled steel tube stub columns via machine learning. Soft Comput 2024;28:5953–67. https://doi.org/10.1007/s00500-023-09343-x.

[178] Huang P, Dai K, Yu X. Data-driven shear strength prediction of steel reinforced concrete composite shear wall. Mater Today Commun 2024;38. https://doi.org/10.1016/j.mtcomm.2024.108173.

[179] Mirrashid M, Naderpour H, Kontoni DPN, Jakubczyk-Gałczyńska A, Jankowski R, Nguyen TN. Optimized Computational Intelligence Model for Estimating the Flexural Behavior of Composite Shear Walls. Buildings 2023;13. https://doi.org/10.3390/buildings13092358.

[180] Zhao W, Chen P, Liu X, Wang L. Impact response prediction and optimization of SC walls using machine learning algorithms. Structures 2022;45:390–9. https://doi.org/10.1016/j.istruc.2022.09.036.

[181] Li B, Qi P, Liu B, Di S, Liu J, Pei J, et al. Trustworthy AI: From Principles to Practices. ACM Comput Surv 2023;55:1–46. https://doi.org/10.1145/3555803.

[182] Chen SZ, Feng DC. Multifidelity approach for data-driven prediction models of structural behaviors with limited data. Comput Civ Infrastruct Eng 2022:1–16. https://doi.org/10.1111/mice.12817.

[183] Naser MZ, Kodur VK. Explainable machine learning using real, synthetic and augmented fire tests to predict fire resistance and spalling of RC columns. Eng Struct 2022;253:113824. https://doi.org/10.1016/j.engstruct.2021.113824.

[184] Chen J, Huang H, Cohn AG, Zhang D, Zhou M. Machine learning-based classification of rock discontinuity trace: SMOTE oversampling integrated with GBT ensemble learning. Int J Min Sci Technol 2021. https://doi.org/10.1016/j.ijmst.2021.08.004.

[185] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. Commun ACM 2020;63:139–44. https://doi.org/10.1145/3422622.

[186] Xu L, Skoularidou M, Cuesta-Infante A, Veeramachaneni K. Modeling tabular data using conditional GAN. Adv Neural Inf Process Syst 2019;32. https://doi.org/10.48550/arXiv.1907.00503.

[187] Marani A, Zhang L, Nehdi ML. Design of concrete incorporating microencapsulated phase change materials for clean energy: A ternary machine learning approach based on generative adversarial networks. Eng Appl Artif

Intell 2023;118:105652. https://doi.org/10.1016/j.engappai.2022.105652.
[188] Liao W, Lu X, Huang Y, Zheng Z, Lin Y. Automated structural design of shear wall residential buildings using generative adversarial networks. Autom Constr 2021;132:103931. https://doi.org/10.1016/j.autcon.2021.103931.
[189] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. ArXiv 2017:1–10.
[190] Goldstein A, Kapelner A, Bleich J, Pitkin E. Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. J Comput Graph Stat 2015;24:44–65. https://doi.org/10.1080/10618600.2014.907095.
[191] Razavi Setvati M, Hicks SJ. Machine learning models for predicting resistance of headed studs embedded in concrete. Eng Struct 2022;254:113803. https://doi.org/10.1016/j.engstruct.2021.113803.
[192] Li ZD, He WY, Ren WX, Li YL, Li YF, Cheng HC. Damage detection of bridges subjected to moving load based on domain-adversarial neural network considering measurement and model error. Eng Struct 2023;293:116601. https://doi.org/10.1016/j.engstruct.2023.116601.
[193] Pan Y, Hong R, Chen J, Feng J, Wu W. Performance degradation assessment of wind turbine gearbox based on maximum mean discrepancy and multi-sensor transfer learning. Struct Heal Monit 2021;20:118–38. https://doi.org/10.1177/1475921720919073.
[194] Luleci F, Necati Catbas F, Avci O. CycleGAN for undamaged-to-damaged domain translation for structural health monitoring and damage detection. Mech Syst Signal Process 2023;197:110370. https://doi.org/10.1016/j.ymssp.2023.110370.