OCEAN
ISSN 3008-1203
Open Access
Review Article

Machine learning-based techniques for marine structures: A state-of-the-art review

Xiaoguang Zhou[1], Chao Hou[1] ✉, Yantao Yu[2], and Yifan Zhou[3]

[1] Department of Ocean Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China
[2] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong 999077, China
[3] School of Engineering, The University of Western Australia, Perth 6009, Australia

Cite This: *Ocean*, **2025**, *1*, 9470005   Read Online

ABSTRACT: The ocean, as one of the vast territories on the Earth, holds abundant resources to be explored, while the high-quality construction and life-cycle maintenance of marine structures are fundamental for advancing this progress. Compared to inland engineering, more severe challenges are posed for marine structures, such as structural durability, variable loading, and harsh environments, making it challenging for traditional analysis and simulation techniques to meet the growing demands of this field. Artificial intelligence, representative by machine learning (ML) and deep learning (DL) algorithms, exhibits robust capability to handle highly nonlinear and complex problems, providing a promising alternative for addressing such challenges in the marine structures. The trend of implementing ML techniques in marine structures is rising, driven by updated algorithms, enhanced computing powers, and high-quality databases. This paper provides a systematic review of the application of ML in marine structures, paying attention to the typical ML and DL algorithms, the popular development platforms, the modeling process, and the application status in terms of design, construction, and maintenance stages. Given that ML algorithms heavily rely on data patterns while generally ignoring the mechanical principles underlying prediction problems, a novel structural mechanisms-based modeling process is proposed to achieve reliable and reasonable predictions. Based on the findings from the review, further research directions and challenges are highlighted and discussed. This paper aims to provide valuable resources for structural engineers and researchers to understand and engage in this promising domain.

KEYWORDS: machine learning, deep learning, artificial intelligence, marine structures, structural mechanism

1 Introduction

The ocean, covering approximately 360 million km^2, accounts for 71% of the Earth's surface, representing vast potential for development. As such, the ocean holds significant strategic importance in the economic growth, resources provision, ecological balance, etc., and the high-performance marine infrastructure is a critical foundation to exploit such potential. Compared to inland engineering, marine structures face more severe challenges across design, construction, and maintenance stages, such as durability issues, ever-changing loads, and harsh conditions [1]. As a result, the scientific construction and life-cycle maintenance of marine structures have become important supports for promoting the high-quality development of marine industry.

With the advent of the big data era, artificial intelligence (AI) technologies, represented by machine learning (ML) and deep learning (DL) algorithms, have garnered widespread attention from structural scholars, which are increasingly emerging as a promising alternative to traditional approaches [2–4]. ML method is able to explore hidden patterns within large amounts of data and construct complex mapping relationships between input parameters and output results, which is extensively utilized in prediction, clustering, and classification tasks [5]. As a crucial branch of ML, DL method exhibits the capability to automatically extract features from input data and analyze such effective features for decision-making across various fields such as image recognition, natural language processing, etc. [6]. Nowadays, these intelligent algorithms have been applied across the entire lifecycle of structural engineering domain in terms of design, construction, and maintenance stages, gradually demonstrating the powerful reasoning ability, remarkable computational efficiency, and exceptional scalability [2,7].

The current application status of ML in the field of structural engineering has been reviewed by several scholars. For specific ML algorithms, Adeli [8] provided a comprehensive review on the neural network (NN) algorithm applied in civil engineering domain during 1989–2000, among which the back-propagation NN (BPNN) was the most popular algorithm due to its simplicity. NN algorithms were also reviewed by Chojaczyk et al. [9] in the reliability analysis of steel structures. Çevik et al. [10] reviewed the application of support vector machine (SVM) in structural engineering problems and further verified its applicability through three scenarios including punching capacity prediction of fiber-reinforced polymer reinforced slabs, ultimate capacity prediction of steel fiber reinforced concrete corbels, and shear capacity prediction of reinforced concrete haunched beams. The advantages and

Received: December 26, 2024; **Revised:** February 23, 2025
Accepted: February 26, 2025

✉Address correspondence to Chao Hou, houc@sustech.edu.cn

disadvantages of such algorithm were also summarized and discussed. Zhang et al. [11] presented a systematic review of genetic programming technique and its applications in civil engineering over the last decade, with its features and potentials for solving various engineering problems discussed. Convolutional NN (CNN) exhibits the huge potential to solve computer vision problems, and its implementation for detecting structure cracks is reviewed by Ali et al. [12], revealing that CNN outperforms manual processes, image processing techniques, and ML methods.

For the specific application domain, Kumar et al. [13] and Zhao et al. [14] overviewed ML methods in predicting the strength of concrete and the property of recycled aggregate concrete, respectively. Mirrashid et al. [15] conducted a systematic review on the development trends from 2010 to 2020 regarding the behavior prediction of concrete materials along with reinforced concrete members through ML methods, concluding that ML methods were more powerful than traditional statistical methods to provide accurate solutions for engineering problems. Similarly, Thai [2] comprehensively reviewed the growing applications of ML algorithms in structural engineering, covering diverse areas such as structural analysis and design, structural health monitoring and damage detection, mechanical properties and mix design of concrete, fire resistance of structures, and resistance of structural members. Salehi et al. [3] summarized recent applications of ML in addressing structural engineering problems, and the advantages of employing such methods were discussed in detail. A scientometrics analysis of more than 4000 scholarly works within this domain was conducted by Tapeh et al. [4], identifying the best practices in terms of procedure, performance metric, database size, etc. Additionally, Afshari et al. [16] reviewed the development and application of ML models in the structural reliability analysis.

Although the aforementioned review papers have highlighted the applications of ML in structural engineering, their research targets primarily focus on the inland structures, ignoring marine structures with complex forms and harsh service conditions. Furthermore, previous reviews tend to concentrate on a specific application area or certain ML technique, lacking a comprehensive review and corresponding comparative analysis across the field. To fill such gaps, this paper presents a comprehensive review on various subtopics regarding the applications of ML techniques in marine structures, with details illustrated in Fig. 1. The paper is organized as follows: Section 2 presents a general introduction on ML and DL algorithms as well as relevant development platforms, with a structural mechanism-based modeling method of ML also proposed. The current research status of ML in marine structures is systematically reviewed in Section 3, ranging from design,

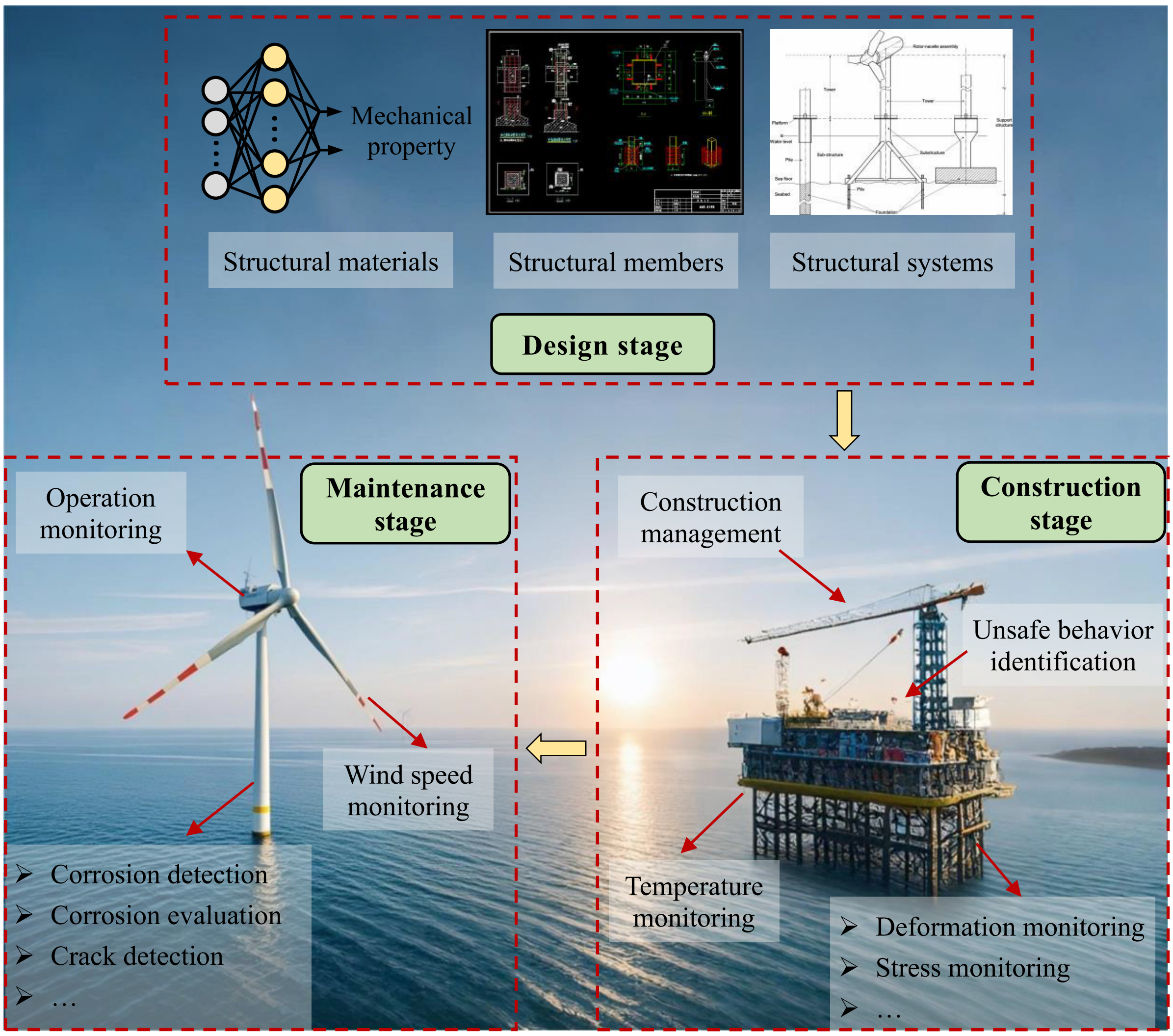

Figure 1 Typical applications of ML techniques in marine structures.

construction, and maintenance stages. Based on the findings from the reviewed literature, Section 4 highlights and discusses the further research directions and challenges in this domain. Finally, conclusions are summarized in Section 5.

2 Overview of ML and DL algorithms

This section provides the general introduction of ML and DL algorithms along with classic development platforms, which are widely adopted in the marine engineering domain. A structural mechanism-based modeling method is also proposed to ensure reliable and reasonable predictions, through which readers can develop their customized ML models.

2.1 ML algorithms

Nowadays, numerous advanced ML algorithms have been developed, and their unique merits and limitations make them suitable for addressing different types of problems. This section focuses on popular algorithms in marine engineering, with their fundamental concepts clearly introduced to assist researchers without ML background effectively understand them.

2.1.1 Single algorithms

2.1.1.1 Regression technique

Regression technique is a representative approach originally developed in statistics, which is designed to investigate the hidden relationships between input variables and output results. This method is then applied in ML domain as a supervised learning algorithm. Depending on the characteristics of problems in marine structures, different types of regression models are selected including linear regression, polynomial regression, lasso regression, ridge regression, logistic regression, etc.

2.1.1.2 NN

The artificial NN (ANN) was developed to mimic workings of biological neurons, which was firstly invented by Rosenblatt [17] in 1958 to address pattern recognition problems. With the advancement in computing power, ANN emerges as one of the most popular algorithms in the marine engineering domain, with various mainstream variants designed such as BPNN [17], radial basis function NN (RBFNN) [18], adaptive neuro-fuzzy inference system (ANFIS) [19], recurrent NN (RNN) [20], long short-term memory (LSTM) [21], gated recurrent unit (GRU) [22], CNN [6,23], etc.

BPNN is one of the mainstream NNs, whose typical architecture involves one input layer, one or more hidden layers, and one output layer [17], with its main calculation process illustrated in Fig. 2a. Specifically, inputs from the input layer are propagated forward to generate outputs, and errors between actual values and output results are then propagated backward to adjust biases and weights of BPNN. The above two processes continue iteratively until the predefined termination criterion is satisfied. Despite its widespread applications in the engineering domain, BPNN tends to suffer from relatively low training efficiency and local optimization issue. RBFNN can overcome such limitations to some extent, and its architecture is also composed of an input layer, a radial basis layer, and a linear layer [18]. ANFIS is a fuzzy inference system implemented within the framework of adaptive NN, effectively combing both advantages of NN and fuzzy logic [19], with its architecture shown in Fig. 2b.

RNN, as a NN variant, is designed for processing sequential data, which incorporates recurrent links to achieve the memory capability. The current computation at each time step depends on both the current input and the previous computations [20]. LSTM is an improved RNN for addressing limitations such as short-term memory, gradient vanishing, or explosion, etc. The memory cell is introduced to store and access information, while the flow of information is controlled through specialized gate units [21]. The key components of LSTM consist of the input gate, forget gate, and output gate, thereby achieving the management of long-term dependency information. Cho et al. [22] further optimized LSTM and proposed GRU in 2014, where the input gate and forget gate were combined into a single update gate, achieving a streamlined architecture. Consequently, GRU is more efficient than LSTM model, and increasingly gains widespread attentions and applications. Due to their excellent capability to capture temporal dependencies and sequential patterns, RNN and its variants are particularly suitable for dealing with time-dependent problems, such as the prediction of full-range responses of marine structures subjected to variable loading. Figure 2c presents the typical architecture of CNN, composing of a feature extractor and a classifier. CNN can automatically extract and learn features from the input data for decision-making, which is widely applied in computer vision-based tasks [6,23]. A detailed review of ANN along with its variants can be found in Ref. [24].

2.1.1.3 SVM

SVM is another powerful algorithm for both classification and regression tasks [25,26], as shown in Fig. 2d. For linear regression problems, SVM aims to identify an appropriate function that can accurately fit all data. However, the kernel function strategy is utilized for nonlinear regression problems by transforming it into a linear one through the dimension increase. The function of SVM can be illustrated as

$$f(x)=\sum_{i=1}^{n}w_iK(x,x_i)+b \tag{1}$$

and the corresponding optimization problem is formulated as

$$\min_{w,b,\xi,\xi^*}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}(\xi_i+\xi_i^*)$$

$$\text{subject to}\begin{cases} y_i-f(x_i)\leqslant\varepsilon+\xi_i \\ f(x_i)-y_i\leqslant\varepsilon+\xi_i^*, \quad i=1,2,\ldots,n \\ \xi_i,\xi_i^*\geqslant 0\end{cases} \tag{2}$$

where w is a weight vector; b is a bias vector; ξ and ξ^* are slack variables; C is a penalty parameter; y_i is the target output; $K(x, x_i)$ is a kernel function such as linear, sigmoid, polynomial, and radial basis functions. This is a convex optimization problem, which can be solved through the Lagrangian multiplier method [25].

2.1.1.4 Gaussian process regression (GPR)

The Gaussian process is a generalization of the Gaussian probability distribution [27], specified through its mean function and covariance function. GPR is a probabilistic and non-parametric supervised learning method, which is formulated as a Gaussian process. The kernel function, also known as covariance function, defines the measurement method of data similarity, which significantly impacts the prediction results. Since GPR is based on the probabilistic theory, an obvious advantage is directly quantifying the uncertainty of prediction results, well-suited for decision-making tasks related to risk assessments in marine structures.

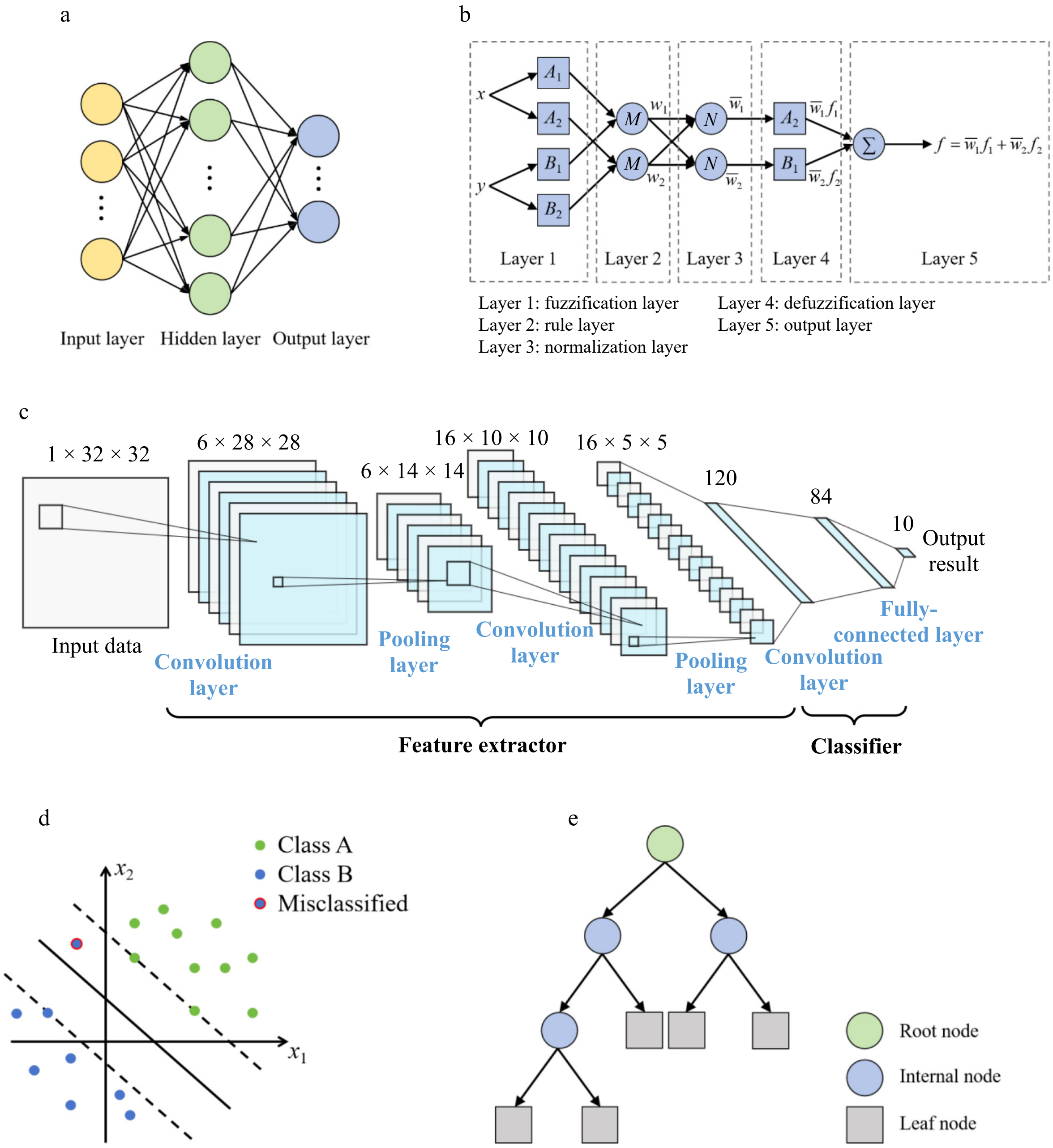

Figure 2 Architectures of classic single ML algorithms. (a) BPNN. (b) ANFIS. (c) CNN (LeNet). (d) SVM. (e) DT.

2.1.1.5 Decision tree (DT) and M5' model tree

DT is a representative tree algorithm, which is broadly used in the field of marine engineering due to its clear architecture, interpretable output, and efficient computation [2,4]. Compared with other ML algorithms, DT can effectively solve high-dimensional problems, which are often encountered in marine structures due to their highly nonlinear and complex nature. Depending on the data type, DT can be categorized into classification DT for discrete data and regression DT for continuous data. Figure 2e shows a typical structure of DT model, consisting of root nodes, internal nodes, leaf nodes, and directed edges. DT is mainly based on the iterative dichotomiser 3 and C4.5 algorithms proposed by Quinlan [28] and the classification and regression tree algorithm proposed by Breiman et al. [29]. In 1992, Quinlan [30] introduced M5 algorithm for handling continuous data, which innovatively combined traditional DT and multivariate linear functions used at leaf nodes. This algorithm can be used to solve multivariate and high-dimensional problems. Subsequently, Wang et al. [31] optimized the M5 algorithm and proposed the M5' algorithm, which significantly reduced the model size while only slightly affecting the model performance.

2.1.2 Ensemble algorithms

Ensemble algorithms employ specified strategies to effectively integrate multiple single models to generate a more powerful model, with bagging and boosting being two typical ensemble strategies [32]. Compared with single algorithms, the performance of the ensemble method is usually better than a single one it integrates.

https://doi.org/10.26599/OCEAN.2025.9470005

2.1.2.1 Bagging strategy

Figure 3a represents a general calculation process of the bagging strategy, which is implemented through the following four steps:

Step 1: Utilize the bootstrap sampling method to extract M samples from the training set.

Step 2: Repeat Step 1 p times to generate p different subsets of the training set.

Step 3: Train p different weak models using these p subsets generated in the previous step.

Step 4: Combine the prediction results of all weak models to obtain the final result of the strong model, where the majority voting is adopted for classification tasks while the average output is taken for regression tasks.

Random forest (RF) is a representative of bagging strategy, which integrates multiple DT models as base models [33]. In addition, RF introduces a mechanism for random feature selection during the training process of each DT model. Therefore, RF not only considers the randomness of samples but also the randomness of features. Such double randomness further increases the diversity of base model, enabling RF to avoid overfitting and exhibit stronger generalization ability.

2.1.2.2 Boosting strategy

Unlike the bagging strategy, the boosting strategy generates multiple weak models in sequence, and the subsequent model corrects the error based on the previous one. The general calculation process of the boosting strategy is shown in Fig. 3b, also including four steps:

Step 1: Assign initial weights to each sample in the training set, and train Weak model 1 using such set.

Step 2: Update the weights based on the training error, assigning higher weights to samples with larger errors. Train Weak model 2 using the updated training set.

Step 3: Repeat Step 2 q times to obtain q weak models.

Step 4: Assign larger weights to weak models with smaller errors, while assigning smaller weights to weak models with larger errors. Combine the prediction results of all weak models to obtain the final result of strong model.

Extreme gradient boosting (XGBoost), as a typical representative of boosting strategy, is developed based on the gradient boosting DT (GBDT), incorporating several enhancements [34]. Its objective function is expressed as

$$\text{Obj} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{m} \Omega(f_k) \tag{3}$$

where the former represents a loss function which measures the error between the predicted and real results, and the latter represents a regularization term that controls the model complexity, with specific expression shown as

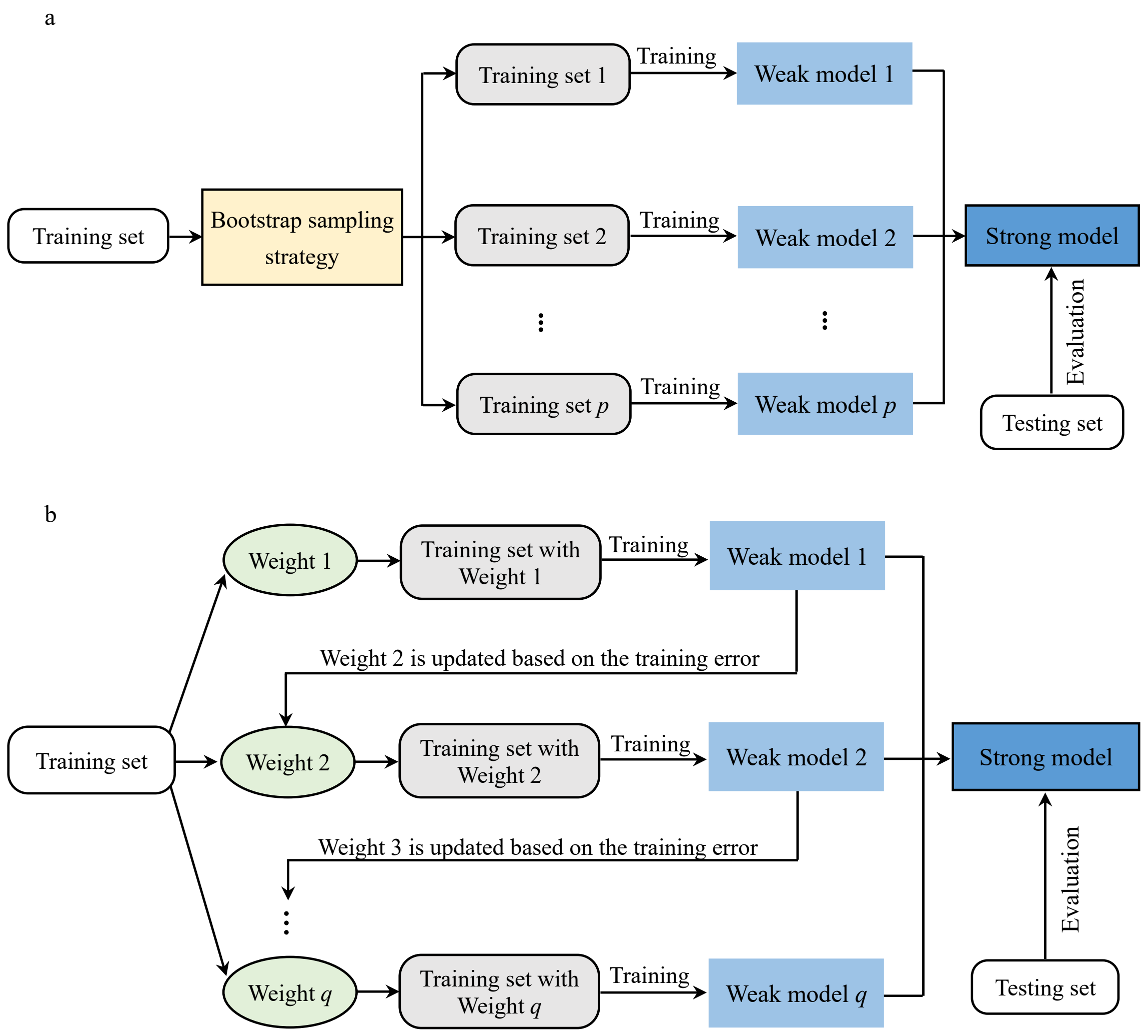

Figure 3 General calculation processes for ensemble algorithms. (a) Bagging strategy. (b) Boosting strategies.

$$\Omega(f_k) = \gamma T + \frac{1}{2}\alpha\sum_{j=1}^{T}|\omega_j| + \frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2 \tag{4}$$

where γ is a hyperparameter to control the number of leaf node; α and λ are coefficients for L_1 and L_2 regularization terms, respectively; T is the total number of leaf nodes for one tree; and ω_j is the predicted value for the jth leaf node.

2.2 DL algorithms

DL algorithm, as a crucial branch of ML method, has gained extensive applications across various fields, and this section introduces its architecture and concept from three levels in the computer vision, including the image classification, object detection, and semantic segmentation.

2.2.1 *Image classification algorithms*

CNN, as a prominent DL algorithm, has been extensively applied in image classification tasks, and LeNet proposed by LeCun et al. [23] in 1988 is one of the earliest CNNs. As illustrated in Fig. 2c, LeNet extracts image features through a series of convolutional layers and pooling layers, followed by the fully-connected layers, and finally performs classification through a softmax layer. Krizhevsky et al. [35] designed a deep and large CNN model named AlexNet, consisting of five convolutional layers and three fully-connected layers, with a total of 60 million parameters and 650,000 neurons. A novel regularization technique known as dropout is employed in the fully-connected layers, which can efficiently avoid overfitting by randomly deactivating neurons during training. The architecture of visual geometry group (VGG) model is similar to AlexNet, while a deeper network is adopted, proving that increasing the network depth can somewhat improve the model performance [36]. In addition, a large convolution kernel is replaced by multiple small ones in VGG, which not only enhances the model capability to extract image features through more nonlinear transformations, but also significantly reduces computational complexity with the receptive field unchanged.

Unlike the VGG model, GoogLeNet proposed by Szegedy et al. [37] improves model performance by increasing network width rather than depth. However, as the network depth increased, He et al. [38] found that the prediction accuracy on both training and testing sets gradually decreased, and this degradation could not be attributed to the overfitting. To address such issue, He et al. [38] devised the residual module and developed various residual networks (ResNets) with different depths. The skip connection is innovatively adopted to allow gradients to flow directly through the network, thereby mitigating the vanishing gradient problem and enabling the training of much deeper networks.

CNNs are gradually developing towards deeper and wider models with more complex architectures, and larger parameters, which limits their applications in mobile and embedded applications. Lightweight networks have attracted increasing attention due to their superiorities of small size and high efficiency, which are gradually being applied in devices with restricted computing resources and tasks requiring real-time decision. MobileNetV1 model is one of the prominent lightweight CNNs, utilizing the depthwise separable convolution to significantly reduce the number of parameters. Compared with the VGG model with 16 layers, the accuracy of MobileNetV1 has only reduced by 0.9% on the ImageNet, while its parameters are reduced by 32 times [39]. Sandler et al. [40] and Howard et al. [41] further optimized MobileNetV1 by employing linear bottlenecks and inverted residuals, developing V2 and V3 versions.

2.2.2 *Object detection algorithms*

Object detection algorithms can simultaneously classify and localize object, which are widely adopted to detect various defects in marine structures, such as cracks, corrosion, rebar exposure, and spalling. These algorithms are mainly divided into one-stage and two-stage methods. The two-stage algorithm generates a series of candidate boxes that may contain the targeted objects through different approaches, and then performs feature extraction, classification, and bounding box regression on each candidate box to refine and rectify it. Representative algorithms include regions with CNN features (R-CNNs) series. Comparatively, the one-stage algorithm directly performs classification and regression on dense anchor boxes, thereby avoiding the generation of candidate regions and improving the detection efficiency, with single shot multibox detector (SSD) and you only look once (YOLO) families being representatives.

Girshick et al. [42] developed the R-CNN algorithm in 2014, which achieved a 30% relative improvement compared with the previous best results on the PASCAL VOC 2012 database, with main steps illustrated in Fig. 4a. Multiple candidate regions are generated through selective search method, and features are then extracted using CNN. Finally, SVM is used for classification and bounding box regression. Although the detection accuracy is improved, R-CNN demands a large computation resource due to three independent training processes. Girshick [43] proposed Fast R-CNN to improve both computational efficiency and detection accuracy. As shown in Fig. 4b, Fast R-CNN extracts features from the entire image and then performs region of interest (RoI) pooling on each candidate region, streamlining the process, and reducing computational rescource. Ren et al. [44] further improved Fast R-CNN and proposed Faster R-CNN, which was composed of region proposal network (RPN) and Fast R-CNN, as illustrated in Fig. 4c. RPN proposals candidate bounding boxes, and Fast R-CNN then extracts features from these candidate boxes using RoI pooling, followed by classification and bounding box regression. Based on the Faster R-CNN, Mask R-CNN adds an additional branch for predicting the object mask (Fig. 4d), suitable for tasks that require precise segmentation [45].

Figure 5 presents the development timeline of YOLO family, with each version updated upon the previous ones to enhance performance. YOLO was firstly developed by Redmon et al. [46] in 2015, which quickly gained popularity due to its remarkable tradeoff between efficiency and accuracy. Unlike traditional object detection methods, YOLOV1 regards the detection task as a regression problem. A single NN model directly predicts the positions and categories of all targets on the entire image, achieving the end-to-end object detection. The loss function of YOLOV1 consists of localization loss, confidence loss, and classification loss, which is calculated by the sum-squared error between predictions and ground truths. YOLOV2 can detect over 9000 object categories, which thus is known as YOLO9000 [47]. Several improvements are adopted such as batch normalization, high resolution classifiers, dimension clusters, etc. To address the instability of YOLOV1 during bounding box prediction, YOLOV2 predicts location coordinates relative to grid-cell locations. Furthermore, the fully-connected layers of YOLOV1 are replaced with anchor boxes to predict bounding boxes, leading to a 7% increase in recall. Compared with YOLOV2, YOLOV3 employs a deeper Darknet-53 as its backbone network, which helps extract the richer feature information [48]. Multi-scale feature fusion is another major

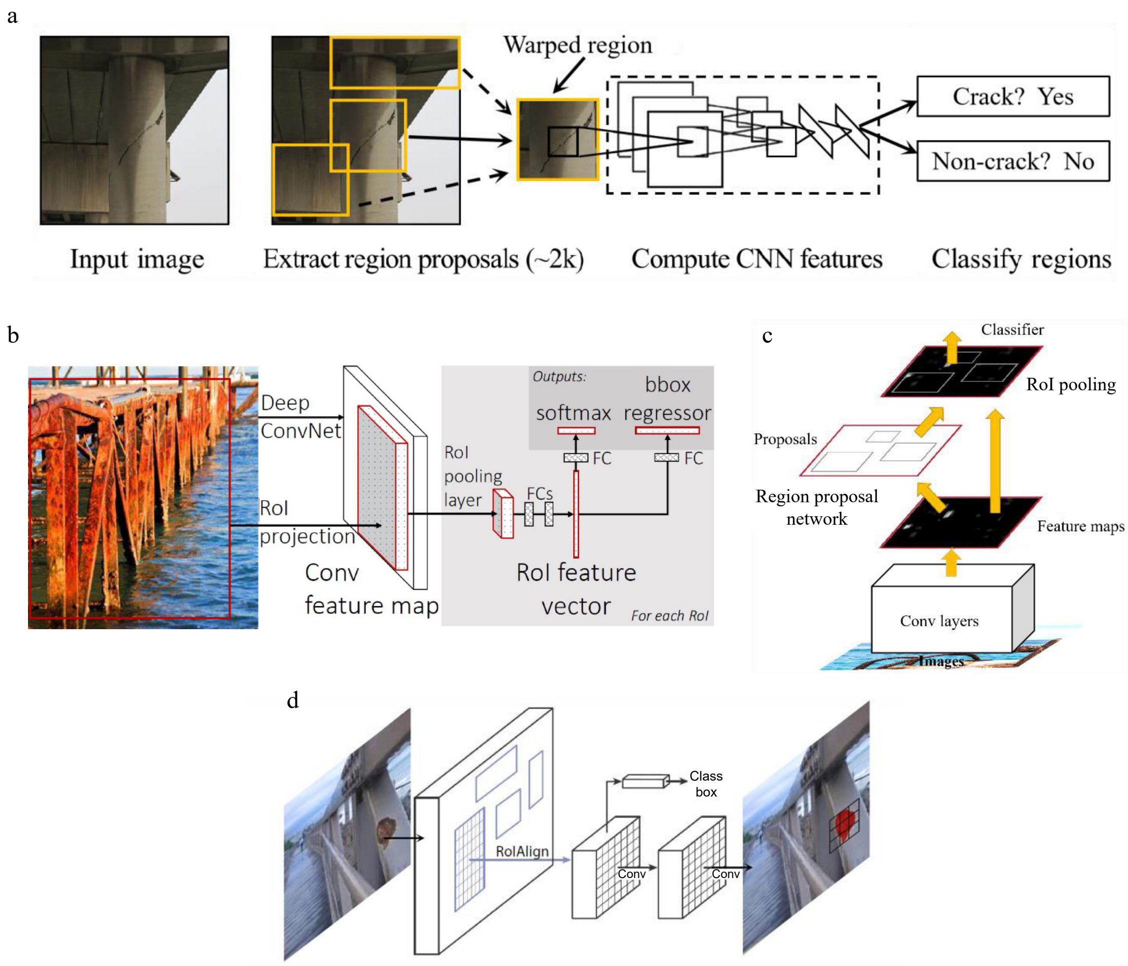

Figure 4 Typical architectures of R-CNN series algorithms. (a) R-CNN. (b) Fast R-CNN. (c) Faster R-CNN. (d) Mask R-CNN.

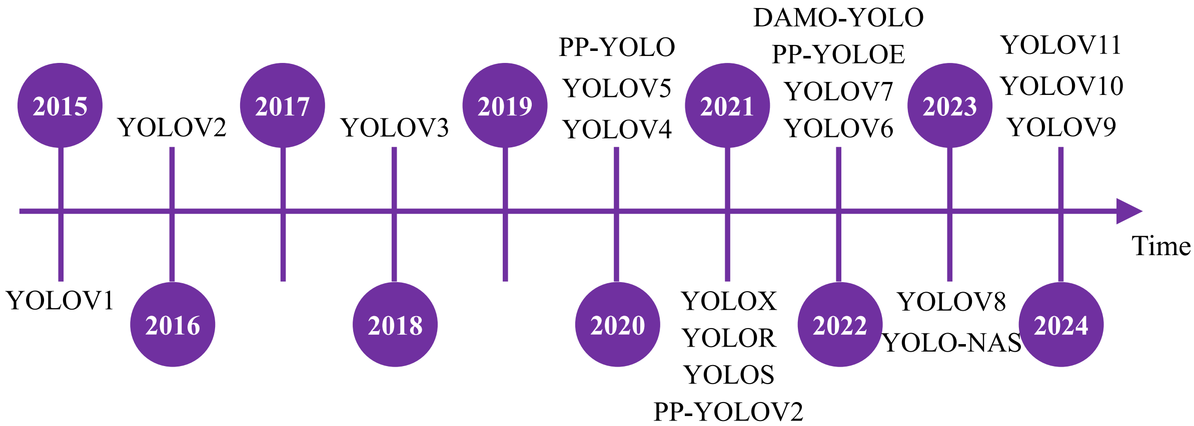

Figure 5 Development timeline of YOLO algorithms.

improvement in YOLOV3, with three feature maps of different scales adopted. Therefore, YOLOV3 can obtain fine-grained features and significantly improve the detection of small objects, which is one of the main weaknesses of the previous versions. YOLOV4 adopts self-adversarial-training and mosaic data augmentation during training [49]. Mish activation, cross stage partial (CSP) connections, and multi input weighted residual connections have also been added in the backbone to further enhance the feature extraction capability of YOLOV4. Both CSP backbone and path aggregation-network (PA-Net) neck are introduced in YOLOV5, enhancing computational efficiency without accuracy compromise. Specifically, CSP effectively addresses redundant gradient information, while PA-Net optimizes feature aggregation. Furthermore, YOLOV5 provides various variants (n, s, m, l, x), which are tailored to diverse application requirements of marine engineering tasks.

YOLOV6 was open sourced by Li et al. [50] in 2022, whose architecture was composed of an efficient backbone using RepBlock for small models or CSPStackRep for large models, a re-parameterized path aggregation network neck, and an efficient decoupled head using hybrid channel strategy. A self-distillation strategy was applied for both regression and classification tasks, which improved the model accuracy without introducing much computation cost. Wang et al. [51] designed YOLOV7, achieving three main innovations, namely extended efficient layer aggregation network, planned re-parameterized convolution, and dynamic label assignment. YOLOV8 introduces some modern features and improvements over the previous YOLO variants, such as a new loss function focal loss, a new data augmentation method Mixup, and a modern evaluation metric average precision across scales. To address the various changes required by deep networks for achieving multiple objectives, Wang et al. [52] proposed the concept of programmable gradient information (PGI), which could provide complete input information for the target task to calculate objective function, and the reliable gradient information could be obtained to update network weights. Generalized efficient layer aggregation network (GELAN), a new lightweight network architecture, was also designed according to the gradient path planning. Combining the proposed PGI and GELAN, YOLOV9 was developed in 2024 [52]. YOLOV10 was created by researchers from Tsinghua University using the Ultralytics Python package, achieving the efficient end-to-end detection through consistent dual assignments for non-maximum suppression-free training [53]. YOLOV11, as the latest in the YOLO family, supports a wide range of computer vision tasks, including object detection, instance segmentation, image classification, pose estimation, oriented object detection, and object tracking [54]. Compared with the previous versions, YOLOV11 exhibits remarkable versatility and enhanced performance across various applications in marine structures. A significant improvement in YOLOV11 is the incorporation of the CSP with kernel size 2 (C3k2) block, replacing the C2f block adopted in the previous versions, which significantly enhances the computational efficiency.

2.2.3 Semantic segmentation algorithms

Compared with the object detection algorithms, semantic segmentation approaches are able to achieve pixel-level localization of various defects in marine structures instead of bounding boxes, providing more detailed data for structural maintenance. The fully convolutional network (FCN), as a classic semantic segmentation algorithm, employs an encoder-decoder architecture [55], enabling pixel-level segmentation of arbitrary-sized images. The encoder module extracts features from the input image, while the decoder module transforms them into an output image. The skip architecture is introduced in FCN to fuse features from different depths, thereby enhancing the segmentation performance [55]. Depending on the upsampling magnifications, FCN can be categorized into FCN-32s, FCN-16s, and FCN-8s, which utilize factors of 32×, 16×, and 8× to restore the original image size, respectively. For achieving segmentation of biomedical images with limited training data, Ronneberger et al. [56] proposed the UNet model upon the design principles of FCN. However, UNet employs the concatenation to fuse features rather than the element-wise addition in FCN. Based on the architecture of UNet, Zhou et al. [57] further developed UNet++ model by redesigning skip pathways and adopting deep supervision strategy. Qin et al. [58] developed U2Net model by redesigning the UNet model, and incorporated multiple residual U-blocks to construct a two-level nested U-structure, whose architecture enabled the model to capture richer local and global information from both shallow and deep layers. Google Inc. devised a series of DeepLab models [59–62], which incorporated various advanced techniques such as atrous convolution, atrous spatial pyramid pooling, conditional random field, etc., to significantly improve both accuracy and speed. To achieve detection on mobile devices, Howard et al. [41] proposed a lightweight model lite reduced atrous spatial pyramid pooling in 2019, whose architecture only consisted of a backbone and a segmentation head.

2.3 Classic development platforms

Many companies and research teams have developed available platforms and cloud services to support scholars for coding their own ML models, such as Google, Baidu, Microsoft, Berkeley AI Research, etc. Several contributors have also participated in the subsequent maintenance and improvement. Table 1 reviews classic platforms developed for ML and DL algorithms, where Python and C++ are the mainstream programming languages, while Python universally supports all mentioned platforms. GitHub, as one of the largest communities for sharing ML codes, is widely used in academic and industrial communities across diverse research domains, and its user feedback thus serves as a valuable metric for evaluating popularity and applicability of different development

Table 1 Detailed information of classic platforms for ML and DL algorithms

Platform	Main programming language	Support language
TensorFlow	C++ (56.0%), Python (25.6%)	Python, C++, Java, JavaScript
PyTorch	Python (56.2%), C++ (35.6%)	Python, C++, Java
Keras	Python (99.9%), C++ (0.1%)	Python
Caffe	C++ (80.0%), Python (9.1%)	Python, Matlab
Sklearn	Python (92.5%), Cython (5.5%)	Python
Theano	Python (93.9%), C (5.0%)	Python
CNTK	C++ (54.7%), Python (35.9%)	Python
MXNet	C++ (48.6%), Python (34.7%)	Python, Java, C++, R, Scala, Clojure, Go, Javascript, Perl, and Julia
Chainer	Python (76.3%), C++ (20.0%)	Python
DL4J	Java (98.5%), Other (1.5%)	Java
Paddle	C++ (47.1%), Python (44.0%)	Python

Note: Sklearn, scikit-learn; DL4J, eclipse Deeplearning4J; CNTK, cognitive toolkit.

platforms, with a comprehensive statistic shown in Fig. 6 (until November 2024). It reveals that TensorFlow developed by Google Brain is currently the most popular platform, and PyTorch is another mainstream platform widely adopted in the academia. The main difference between both platforms lies in their computing paradigms. TensorFlow adopts a static computation graph, indicating that the model development should be completed before starting computation. However, PyTorch is a dynamic framework that allows input data for computation during the model development, thereby enabling for real-time improvements.

2.4 Structural mechanisms-based modeling method of ML

Structural engineering problems are governed by well-defined mechanical principles and theoretical foundations, and traditional numerical methods strictly follow such principles, enabling them widely applicable in engineering practice. However, the modeling processes of most ML algorithms are similar to a “black box”, which generally learn the hidden patterns in data while ignore the structural mechanisms behind prediction problems. The direct application of data-driven ML models in the marine engineering domain presents considerable challenges, particularly for decision-making tasks related to risk assessments. Therefore, it is essential to effectively integrate data-driven ML models with structural mechanisms behind the problems, ensuring that mechanical principles can accurately reflected in models and providing reliable design guidance. Based on structural mechanisms, this paper proposes a modeling method for ML, with basic process shown in Fig. 7, and the traditional modeling method is also provided for comparison. It can be observed that the proposed method combines the mechanical principles of engineering problems with data-driven intelligent algorithms through the following five steps:

Step 1: Establishment of comprehensive database. The performance of data-driven ML models heavily depends on the quantity and quality of training data, necessitating the establishment of a high-quality and large-scale database [63,64]. Literature review and internet search are two mainstream ways for collecting experimental data, while numerical simulation and generative algorithm can further supplement data to enrich database. Before training ML models, various pre-processing steps are implemented on the database, such as data normalization, database division, etc.

Step 2: Determination of input parameters for ML. ML methods explore the data patterns between input parameters and output results, with the selection of input parameters being crucial for the model performance. By combining correlation analysis and structural mechanisms, the input parameters can be reasonably determined.

Step 3: Development of initial ML models. Based on the various platforms introduced in Section 2.3, initial ML models can be developed using various mainstream single algorithms and ensemble algorithms, depending on the problem characteristics and the data nature.

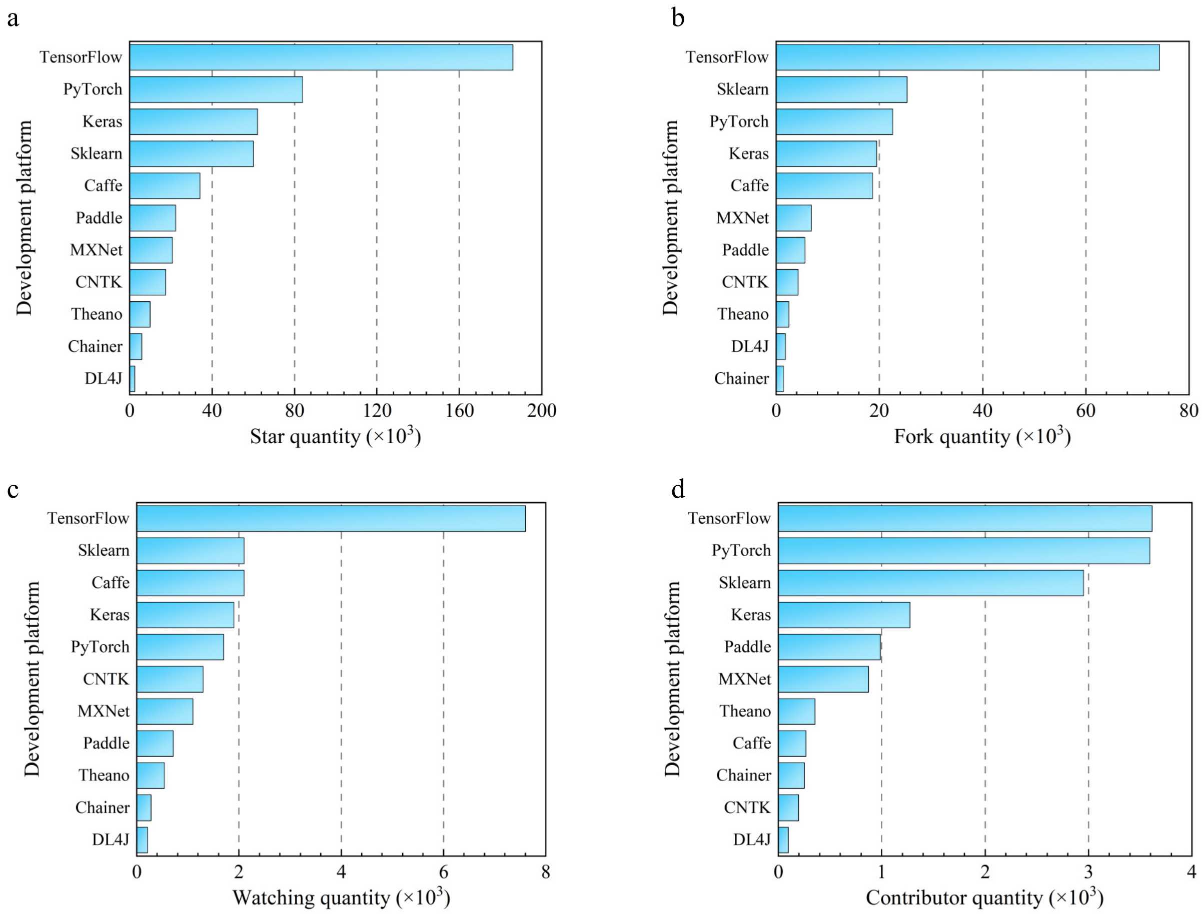

Figure 6 Statistic of classic development platforms based on GitHub. (a) Star quantity. (b) Fork quantity. (c) Watching quantity. (d) Contributor quantity.

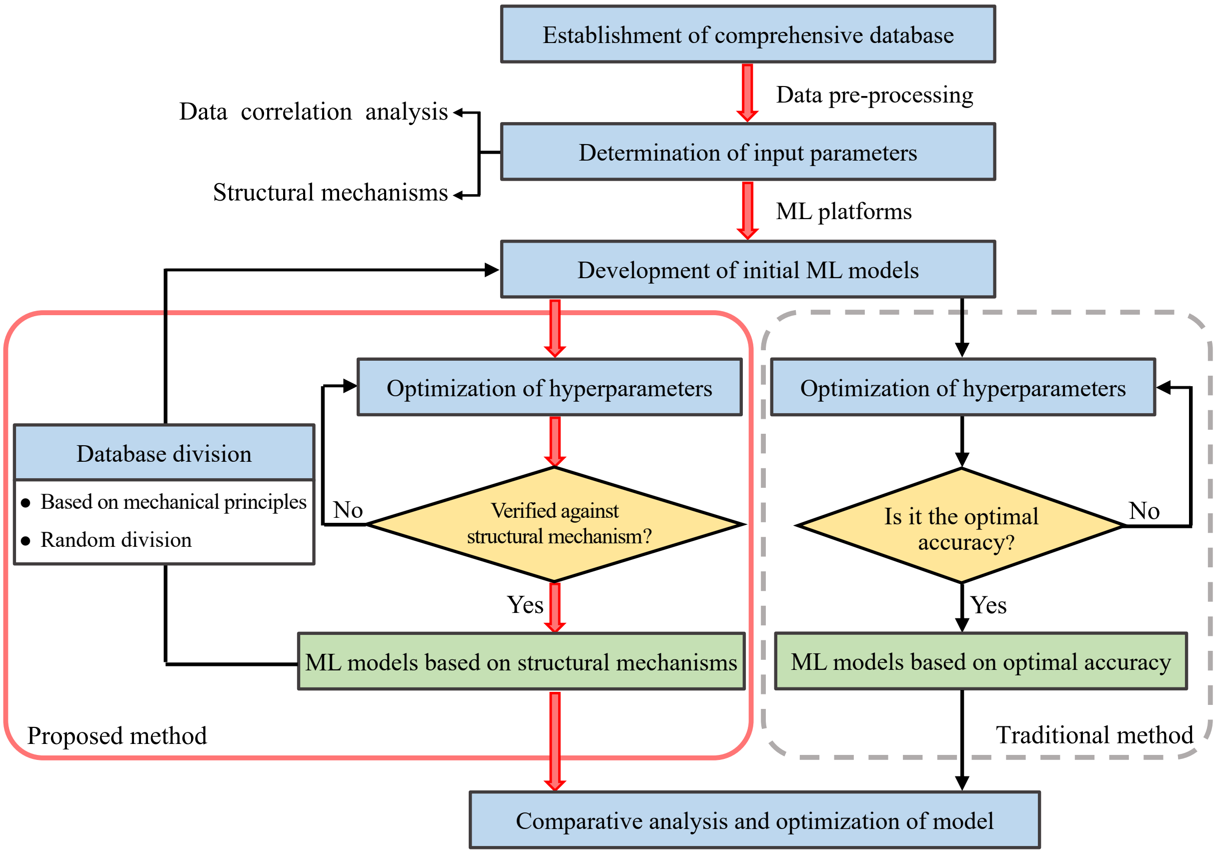

Figure 7 Flowchart of proposed modeling method for ML models based on structural mechanisms.

Step 4: Optimization of model hyperparameters. Hyperparameters play a decisive role in the performances of ML models. For traditional data-driven ML models, the optimization of hyperparameters is solely guided by the prediction accuracy, while structural mechanisms reflected by the model may fail due to overfitting, abnormal data, etc., particularly when the training data are too limited to reflect the actual laws. In view of this, this paper proposes an innovative optimization method that incorporates mechanical principles, with a verification module of structural mechanisms embedded. As such, the hyperparameter optimization not only is based on the prediction accuracy but also takes into account the mechanical principles. Each model with different hyperparameters is validated against structural mechanisms. Among the models that pass such validation, the one with the best accuracy is considered optimal. It should be mentioned that the verification method of mechanical principles is determined by the prediction problems. For the capacity prediction of structural members, the correlation between capacity and input parameters is examined to evaluate the rationality of mechanical principles [63], as shown in Fig. 8a. It can be seen in Fig. 8b that the seismic interstory drift ratio is selected as an indicator for the design of shear wall [64].

Step 5: Comparative analysis and optimization of model performances. Multiple evaluation indicators can be used to compare the established ML models, with in-depth analysis conducted such as the prediction stability, uncertainty, model interpretability, etc. Different models are evaluated from the engineering perspective to evaluate their practical applicability. Finally, optimization methods for ML models are explored in order to further improve the performance, such as database division methods, hyperparameter optimization strategies, etc.

The effectiveness of proposed modeling method is validated through the capacity prediction of structural members, demonstrating that mechanical principles can be reasonably reflected in ML models [63]. Therefore, the interpretability and reliability of ML models can be enhanced by reasonably incorporating domain-specific knowledge, which bridges the gap between data-driven approaches and established engineering principles, facilitating the application of ML methods in the marine structures to provide robust and informed decision-making.

3 Research status of ML methods in marine structures

This section systematically reviews the application status of ML techniques in the design, construction, and maintenance of marine structures, with details illustrated in Fig. 1.

3.1 Literature search and selection

For literature research, the Web of Science and Scopus databases are employed due to their reliability and extensive coverage, where identical literature search criteria are applied across both databases to ensure consistency. Table 2 outlines the specific search keywords corresponding to the design, construction, and maintenance stages. The search criteria are restricted to peer-reviewed journal articles published before December 2024, focusing on the marine engineering field. After conducting the initial search, the articles collected from both databases are merged and meticulously deduplicated to establish a comprehensive and reliable literature dataset. A preliminary screening is then conducted for all articles as per their titles, keywords, and abstracts, followed by an in-depth screening on their full text. Finally, a total of 142 highly relevant

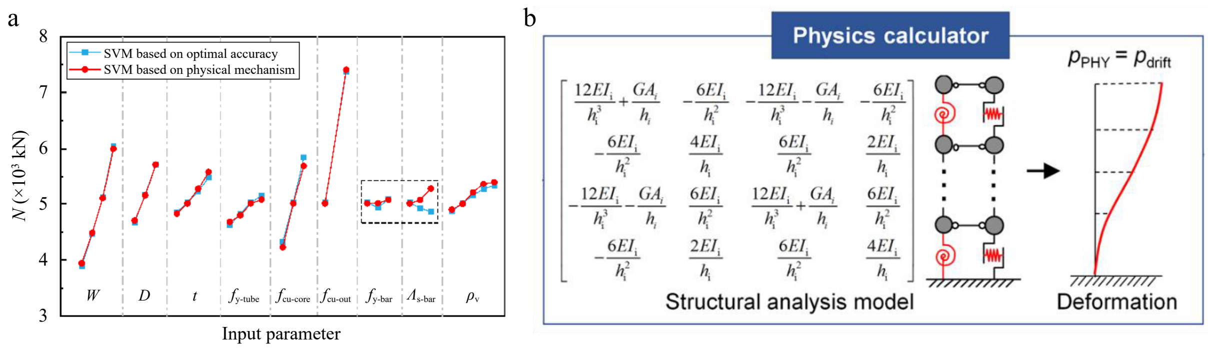

Figure 8 Verification methods of structural mechanisms for different tasks. (a) Prediction of member capacity. Reproduced with permission from Ref. [63] ©2023, Elsevier Ltd. (b) Design of shear wall. Reproduced with permission from Ref. [64] ©2022, John Wiley & Sons Ltd.

Table 2 Search keywords for different application domains in marine structures

Stage	Application domain	Search keyword
Design stage	Properties of structural materials	ML related terms AND concrete
	Behaviors of structural members	ML related terms AND CFST
	Designs of structural systems	Generative AI related terms AND (building OR structure)
Construction stage	PPE detection	DL related terms AND personal protective equipment
	Unsafe behavior identification	DL related terms AND behavior AND construction
	Physiological status monitoring	ML related terms AND worker AND (physiological OR fatigue)
Maintenance stage	Crack detection	DL related terms AND concrete AND crack
	Corrosion detection	DL related terms AND (corrosion OR rust) AND steel
	Corrosion degree evaluation	

Note: ML related terms refers to ML and its specific algorithms (e.g., NN, XGBoost, ensemble algorithm, etc.); Generative AI related terms refers to generative AI and its specific algorithms (e.g., GAN, VAE, etc.); DL related terms refers to ML, DL, and its specific algorithms (e.g., CNN, YOLO, R-CNN, etc.).

articles are collected for further reviewing the application status of ML techniques in marine structures.

3.2 Design stage

As presented in Fig. 1, the application of ML in the design stage can be mainly divided into the property prediction of structural materials, behavior evaluation of structural members, and design guidance of structural systems.

3.2.1 Properties of structural materials

Concrete is one of the most commonly used construction materials in marine structures, such as bridges, towers, platforms, etc., and accurate prediction of its properties is crucially important for scientific design and safe service of such structures. Considering the complicated mixtures and varied curing conditions, the property prediction of concrete is challenging. Significant efforts have been conducted to utilize ML techniques in predicting concrete properties, with related studies comprehensively reviewed in Refs. [13–15,65].

3.2.1.1 Compressive strength

One of the early applications of ML in predicting compressive strength of high-performance concrete (HPC) was conducted by Kasperkiewicz et al. [66], concluding that ML algorithm was able to effectively address the strength prediction problems. Since then, a huge number of ML models have been developed for different types of concrete, with details listed in Table 3. It is evident that NN is the most popular algorithm, while ensemble algorithms have gained increasing attention in recent studies. Specifically, Chou et al. [68] compared the performance of base models (ANN, SVM, and DT) and ensemble models (voting, bagging, and stacking techniques) in predicting strength of HPC, indicating that ensemble models could enhance the performances of base models. This conclusion was further confirmed by Feng et al. [69], where base models were integrated through adaptive boosting algorithm. More strength prediction models regarding various types of concrete are comprehensively reviewed in Refs. [13–15,65].

3.2.1.2 Tensile strength

Comparatively, the application of ML methods to predict tensile strength of concrete is relatively limited, which can be ascribed to the fact that the compressive performance of concrete significantly outperforms its tensile performance. The tensile strength of HPC was evaluated by Khan [67] using ANN algorithm, whilst Li et al. [70] employed various ensemble algorithms. Pakzad et al. [83] adopted thirteen ML algorithms to predict the tensile strength of steel fiber-reinforced concrete (SFRC), among which SVM achieved the highest prediction accuracy. Sun et al. [84] collected 313 tensile strengths of basalt fiber-reinforced coral aggregate concrete from fourteen papers, and an optimized XGBoost model was designed to predict their strength. Results indicate that the relative errors between the experimental and predicted strengths consistently are below 5%, highlighting the robust generalization and excellent accuracy of the developed model.

3.2.1.3 Flexural strength

SFRC exhibited superior mechanical performances than normal concrete due to the incorporation of discontinuous fibers, and Kang

Table 3 Application of ML methods in compressive strength prediction for different concrete

Concrete type	Database size	ML algorithm	Year	Reference
HPC	340	ANN	1995	[66]
	128	RBFNN	2012	[67]
	1133, 194, 144, 104, 100	ANN, SVM, DT, voting method, bagging method, stacking method	2014	[68]
	1030	ANN, SVM, AdaBoost	2020	[69]
	190	AdaBoost, GBDT, XGBoost, RF	2022	[70]
Ultra-HPC	363	BPNN, DT, RF, GBDT, XGBoost	2023	[71]
	810	RF, AdaBoost, GBDT	2024	[72]
Recycled aggregate concrete	470	ANN	2021	[73]
	721	GBDT, XGBoost, SVM	2022	[74]
	465	BPNN, SVM, AdaBoost, GBDT	2024	[75]
Coral aggregate concrete	420	BPNN	2024	[76]
Lightweight concrete	194	GPR, SVM, bagging method	2022	[77]
	232	ANN, GEP, GBDT	2022	[78]
Self-compacting concrete	327	RBFNN	2022	[79]
	255	SVM, RF	2023	[80]
Polymer concrete	15	ANN	2012	[81]
	1476	Ridge regression, Lasso regression, SVM, GPR, DT, RF, GBDT, ANN, KNN	2022	[82]

Note: GEP, gene expression programming.

et al. [85] established eleven ML models to predict its flexural strength. Zheng et al. [86] also predicted flexural strength of SFRC through three ensemble models, including gradient boosting, RF, and XGBoost, concluding that gradient boosting achieved the highest accuracy, with coefficient of determination (R^2) achieving 0.96 across all data. Based on ANN models, Barbuta et al. [81] evaluated the flexural strength of fly ash concrete, whilst Golafshani et al. [73] examined the flexural strength of recycled aggregate concrete.

3.2.1.4 Other properties

Besides the strength prediction of different types of concrete, ML techniques have also been applied to evaluate their other properties, such as the elastic module [85–88], Poisson's ratio [88], slump [80,89,90], coefficient of thermal expansion [88], durability property [67,91], stress–strain relationship [92], shrinkage and creep property [93], etc. On the other hand, optimizing concrete mixtures is another important application domain of ML. For example, Zhang et al. [94] proposed an efficient ML-based method to optimize concrete mixtures, considering the multiple objectives such as the strength, cost, slump, etc. Naseri et al. [95] developed ML models to design mixture for sustainable concrete, with environmental impacts taken into account, including embodied CO_2 emission, as well as energy and resource consumptions. These applications highlight the powerful capabilities of ML in concrete materials, enabling to provide efficient and accurate guidance.

3.2.2 Behaviors of structural members

Concrete-filled steel tube (CFST) member utilizes combined advantages of steel and concrete, demonstrating the superior mechanical property, convenient construction process, and excellent economic benefit, which is one of the preferred forms for marine structures [96,97]. Taking such typical structural members as an example, prediction of their performances through ML methods primarily focuses on the following conditions.

3.2.2.1 Static loading

Table 4 summarizes the application of ML methods for capacity prediction of CFST under static loading, among which BPNN is the most mainstream algorithm while ensemble algorithms are relatively limited. ML was employed earlier for the axial loading condition, where Hao et al. [98] in 2004 established a three-layer BPNN model to predict the capacity of 50 CFST components, providing valuable insights into the application of ML in composite structures. Due to the limited training data, the applicable range and prediction accuracy of model need further improvement. Zarringol et al. [104] and Naser et al. [108] established more comprehensive databases composed of 895 and 979 rectangular CFSTs, covering various types of components with different material properties, section types, and slenderness ratios. Based on such expanded databases, prediction accuracies and applicable ranges of ML models have been improved accordingly. Hou et al. [97] constructed a comprehensive database containing 2045 circular CFST samples under axial compression and investigated the impact of database subdivision on the model performances. It is concluded that subdividing database as per failure mechanisms can effectively enhance the model performance, whereas the random subdivision of database imposes negligible or even adverse impacts.

Comparatively, existing research on the prediction of flexural capacity is quite limited, with relevant information provided in Table 4. Basarir et al. [113] used ANFIS model to predict the flexural capacity of circular CFST under monotonic/cyclic loadings. Results reveal that root mean square error (RMSE) and R^2 provided by ANFIS are 0.096 and 0.81, respectively, higher than the current design standards (RMSE ranges from 0.224 to 0.238, and R^2 ranges from 0.30 to 0.53). Hanoon et al. [114] adopted ANN combined with particle swarm optimization to predict both flexural capacity and stiffness, whilst 224 circular and rectangular components jointly trained model, ignoring the influences of sectional shapes.

https://doi.org/10.26599/OCEAN.2025.9470005

Table 4 Summary of ML models for predicting CFST members subjected to static loading

Loading type	Reference	Database size	Shape	Algorithm
Axial loading	Hao et al. [98], 2004	50	R	BPNN
	Ahmadi et al. [99], 2014	272	C	BPNN
	Güneyisi et al. [100], 2016	314	C	GEP
	Du et al. [101], 2017	305	R	BPNN
	Tran et al. [102], 2019	300	R	BPNN
	Nour et al. [103], 2019	97	C	GEP
	Zarringol et al. [104], 2020	895	C	BPNN
		1305	R	
	Mai et al. [105], 2022	300	R	BPNN, RBFNN
	Le [106], 2022	314	R	GPR
	Tran et al. [107], 2020	768	C	BPNN
	Naser et al. [108], 2021	979	C	GA, GEP
		1245	R	
	Lee et al. [109], 2021	1245	C	CatBoost
		979	R	
	Vu et al. [110], 2021	1017	C	GTB, RF, SVM, DT, DL
	Hou et al. [97], 2022	2045	C	BPNN, RBFNN, GPR
	Faridmehr et al. [111], 2022	2963	R and C	CNN
	Zhou et al. [112], 2023	1641	R	BPNN, RBFNN, ANFIS, GPR, M5' model tree
Pure bending	Basari et al. [113], 2019	72	C	ANFIS
	Hanoon et al. [114], 2022	224	R and C	ANN
Eccentric loading	Gao [115], 2011	55	R	BPNN
	Saadoon et al. [116], 2012	111	R	BPNN
	Luat et al. [117], 2020	141	R	MARS
	Zarringol et al. [104], 2020	392	R	BPNN
		499	C	
	Naser et al. [108], 2021	394	R	GA, GEP
		485	C	
	Lee et al. [109], 2021	394	R	CatBoost
		485	C	
	Hou et al. [118], 2022	804	R	BPNN, RBFNN, GPR
	Nguyen et al. [119], 2022	241	C	ANN, MARS
	Hou et al. [120], 2023	899	C	BPNN, RBFNN, SVM, ANFIS, GPR, MARS, M5' model tree, Ensemble of M5' model trees, XGBoost
	Wang et al. [121], 2023	403	R	ANN, SVM, RF

Note: R and C, CFST members with rectangular and circular sections; GA, genetic algorithm; MARS, multivariate adaptive regression spline.

As illustrated in Table 4, scholars have begun to adopt ML algorithms for predicting eccentric compression capacity of CFSTs. Unlike axial loading condition, the eccentricity direction of loading imposes a significant impact on the capacity of rectangular components. However, only Naser et al. [108] and Hou et al. [118] considered the above impact when establishing their models, and more research was needed to explore such impact on ML models. Hou et al. [120] further proposed an intelligent prediction method for the axial load–bending moment (N–M) interaction curves of CFST, facilitating advanced structural design of composite members through ML methods. To facilitate the practical application of established models in engineering designs, some scholars have also proposed simplified formulas derived from ML models [104,108].

In addition to the capacity prediction, scholars have also started to employ ML methods for predicting full-range loading–deformation curves of axial compression components, and typical diagrams are outlined in Fig. 9, which are divided into two ways. Many studies adopt the characteristic point-based method, where the complete curve is divided into several characteristic points, and ML models are then adopted to predict such points, which are connected to form the complete curve. As shown in Fig. 9a, Yeong

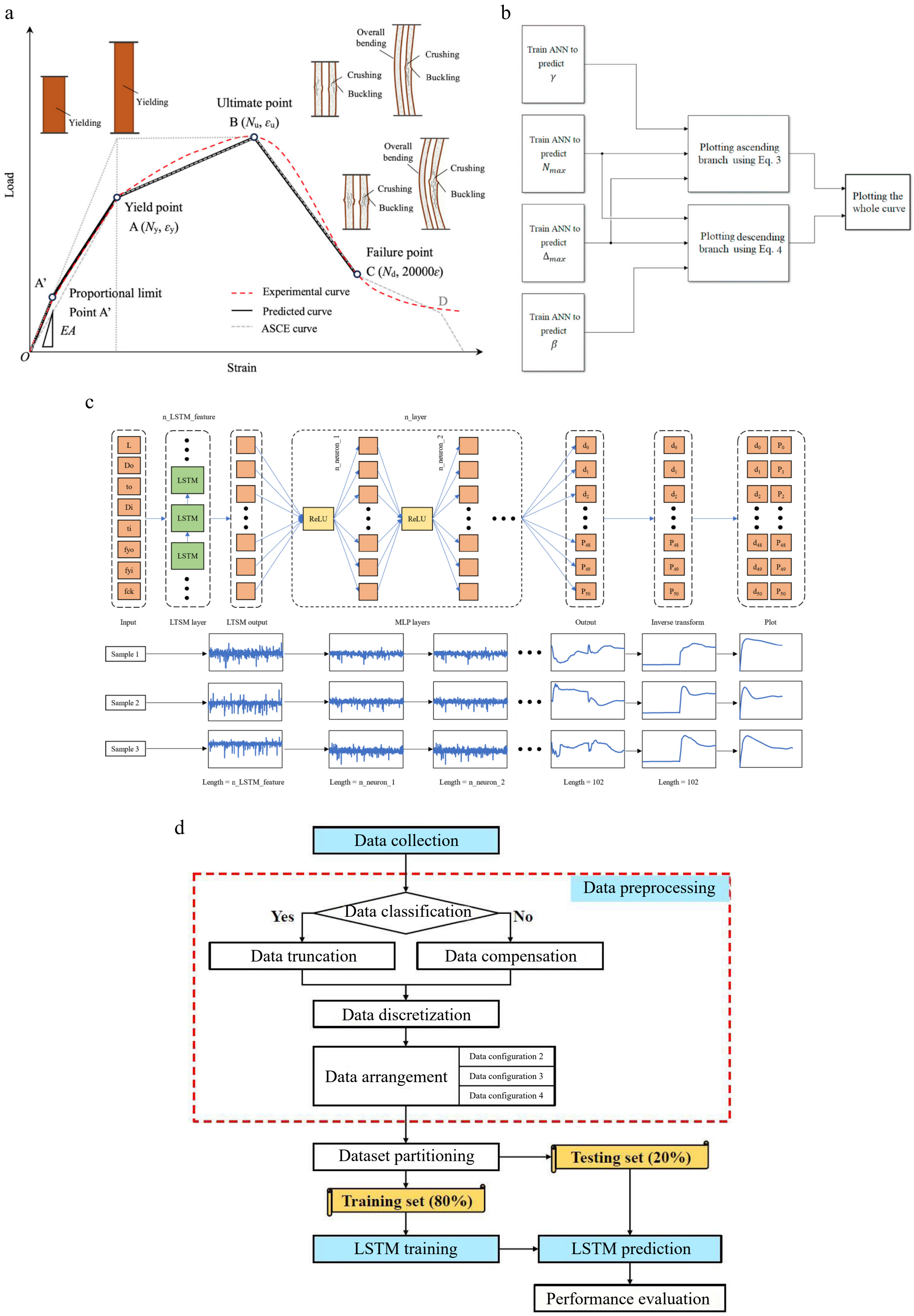

Figure 9 Process diagrams for predicting loading-deformation curves using ML method. Characteristic point-based methods: CFDST (a) and CFST (b). Reproduced with permission from Ref. [122] ©2024, Elsevier Ltd.; Ref. [123] ©2022, Elsevier Ltd. LSTM-based methods: CFDST (c) and CFST (d). Reproduced with permission from Ref. [124] ©2024, Elsevier Ltd.; under the CC BY 4.0 license from Ref. [125] ©2023, The Authors.

et al. [122] divided the load–strain curve of concrete-filled double-skin steel tubular (CFDST) column into the yield point, ultimate strength point, and failure point. Six ML models were adopted to predict these points and reconstruct the complete curve, among which SVM model exhibited the best fit to the test curves. Similarly, Zarringol et al. [123] used ANN to plot the curve of CFST columns, as illustrated in Fig. 9b. However, such approaches capture the general trends of curves, and the accuracy especially for the descending part is limited. In fact, this type of curve can be regarded as time series data, for which LSTM is one of the well-suited algorithms. As shown in Fig. 9c, Le et al. [124] predicted complete load-displacement curves of CFDSTs through an LSTM-based model, which adopted numerous LSTM cells to extract features from inputs, followed by the NN models to output curve. Due to the limited experimental curve, both finite element simulation and DoppelGANger algorithm were used to generate data for training model. Results demonstrate that R^2 values for both training and testing sets exceed 0.95, effectively validating the feasibility and accuracy of the proposed method for predicting load–displacement curves. Fan et al. [125] also applied the LSTM model to predict the loading-strain curves of circular CFSTs, and different input configurations are investigated. It can be found in Fig. 9d that the input configuration can be divided into two cases: (1) design parameters only; (2) the combination of design parameters and strain series. Results show that adding strain series can effectively improve the prediction accuracy, and LSTM algorithm achieves a better adaptability to handle two-dimensional prediction problems than ANN, RF, and SVM models.

3.2.2.2 Dynamic loading

Marine CFST structures in practice may suffer from accidental dynamic loadings, such as vehicle/ship collisions, impacts, and explosions, and a statistical report reveals that such loadings are responsible for approximately 40% of bridge failures [126]. As a result, the dynamic loading becomes a critical scenario for marine CFST structures, necessitating in-depth analysis to ensure structural safety. Based on 330 square CFST members, a GBDT model was established to predict their transverse impact capacity [127]. Yang et al. [128] evaluated the residual capacity of CFST columns under impact loading through the stacking ensemble method, where the base learner was selected as *K*-nearest neighbors (KNN), RF, and XGBoost models while the linear regression model was employed as the meta learner. Results show that the stacking ensemble approach exhibits higher accuracy than bagging and boosting methods, achieving $R^2 = 0.97$ in the testing set.

For CFST members after explosion, Li et al. [129] developed three XGBoost models to predict the residual axial capacity ratio (RCI). XGBoost3 model can provide efficient and accurate assessments of RCI without requiring blast loading parameter and axial load ratio, both of which are typically difficult to obtain in practice. Li et al. [130] further predicted the residual axial capacity of circular CFST columns through various advanced ML algorithms, and feature importance of input variables was thoroughly analyzed. The ensemble algorithm XGBoost exhibits the best performance, with R^2, mean absolute error, and RMSE on the testing set being 0.981, 0.018, and 0.026, respectively. Besides the capacity prediction, He et al. [131] adopted nine regression-based ML algorithms to estimate the peak displacement of CFST subjected to blast loading, with the variability of input parameters considered. Lai et al. [132] developed a novel probabilistic natural gradient boosting (NGBoost) model to predict the maximum displacement of CFST under lateral impact loading, demonstrating that the developed NGBoost not only achieved high accuracy but also provided detailed probabilistic predictions.

3.2.2.3 Corrosion condition

Corrosion is one of the severe challenges encountered by marine CFST structures, which not only reduces effective sectional area and stability of the steel tube but also weakens its confinement effect on the core concrete. However, limited research was conducted on the application of ML methods for the corrosion condition, and Zhou et al. [133] adopted seven ML methods to predict the residual strength of CFST under uniform corrosion and random localized corrosion. Given the limited test data and high costs for acquiring it, active learning (AL) approaches were adopted to assist in training ML models, with results presented in Fig. 10. For four AL methods, the ML model trained by only 63% data achieves comparable performances to the all data scenario. It can be concluded that AL can effectively identify valuable data, enabling ML models to achieve excellent performances even with fewer training data.

3.2.2.4 Fire condition

At present, the prediction of CFST performance under fire conditions primarily focuses on the fire resistance rating (FRR) and residual strength. ML was applied earlier in FRR prediction, and Al-Khaleefi et al. [134] in 2002 established NN model with one hidden layer and 14 neurons. Due to only 27 sets of test data, its generalization and accuracy need to be further enhanced. Zhao et

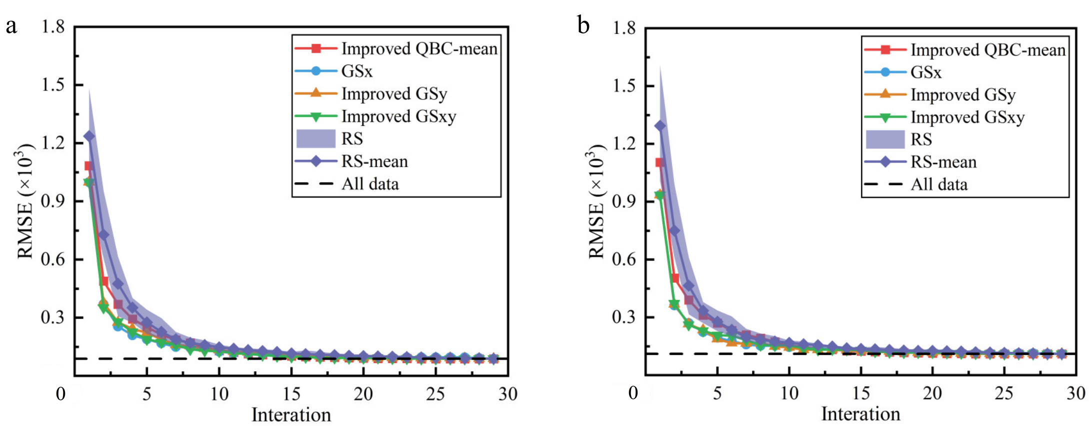

Figure 10 Performance comparison among different AL methods. (a) Training set. (b) Testing set. QBC, query-by-committee; GS*x*, greedy sampling on the inputs; GS*y*: greedy sampling on the outputs; GS*xy*: greedy sampling on the inputs and output; RS, random sampling. Reproduced with permission from Ref. [133] ©2023, Elsevier Ltd.

al. [135] adopted 172 samples to train ML model, with its hyperparameters optimized through balancing composite motion optimization algorithm. The established model outperformed existing design standards, achieving R^2 = 0.947 for all samples. Based on the principles of knowledge distillation and semi-supervised learning, Xu et al. [136] proposed a novel prediction model for FRR, whose training process effectively integrated the domain expertise through NN technology.

For residual strength prediction for CFDST columns subjected to eccentric loading, Wu et al. [137] modified the traditional Rankine method by considering the shear effect coefficient, and developed various ML models to predict such coefficient. Narang et al. [138] assessed the residual strength indexes of 56 circular CFST columns through mainstream ML algorithms, including GPR, bagging and boosting ensemble methods, SVM, and ANN. Quadratic SVM and bi-layered ANN models outperformed the other ML models, with their R^2 values on the training set being 0.93 and 0.98, respectively. Moradi et al. [139] adopted ANN to predict FRR and residual strength of circular CFST columns, respectively. Some researchers have also designed graphical user interface (GUI) to facilitate the practical implementation of established models, as shown in Fig. 11.

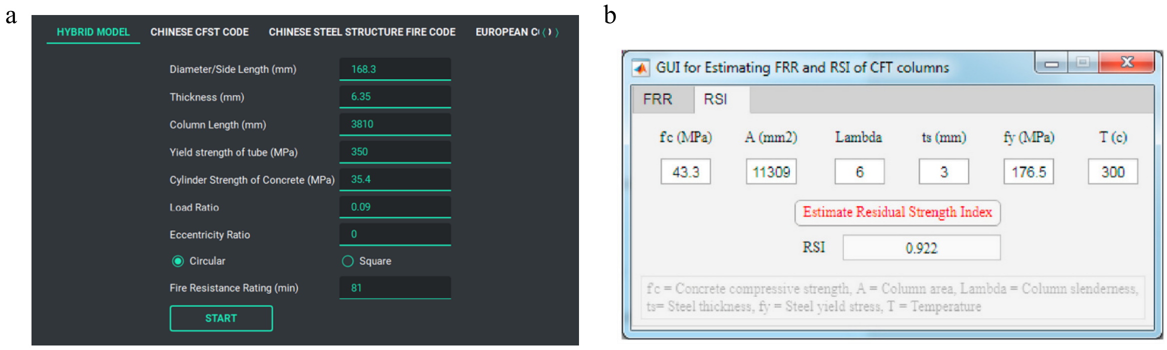

Figure 11 GUIs for the fire resistance prediction of CFST members. (a) FRR. Reproduced with permission from Ref. [135] ©2022, Elsevier Ltd. (b) Residual strength index. Reproduced with permission from Ref. [139] ©2021, Elsevier Ltd.

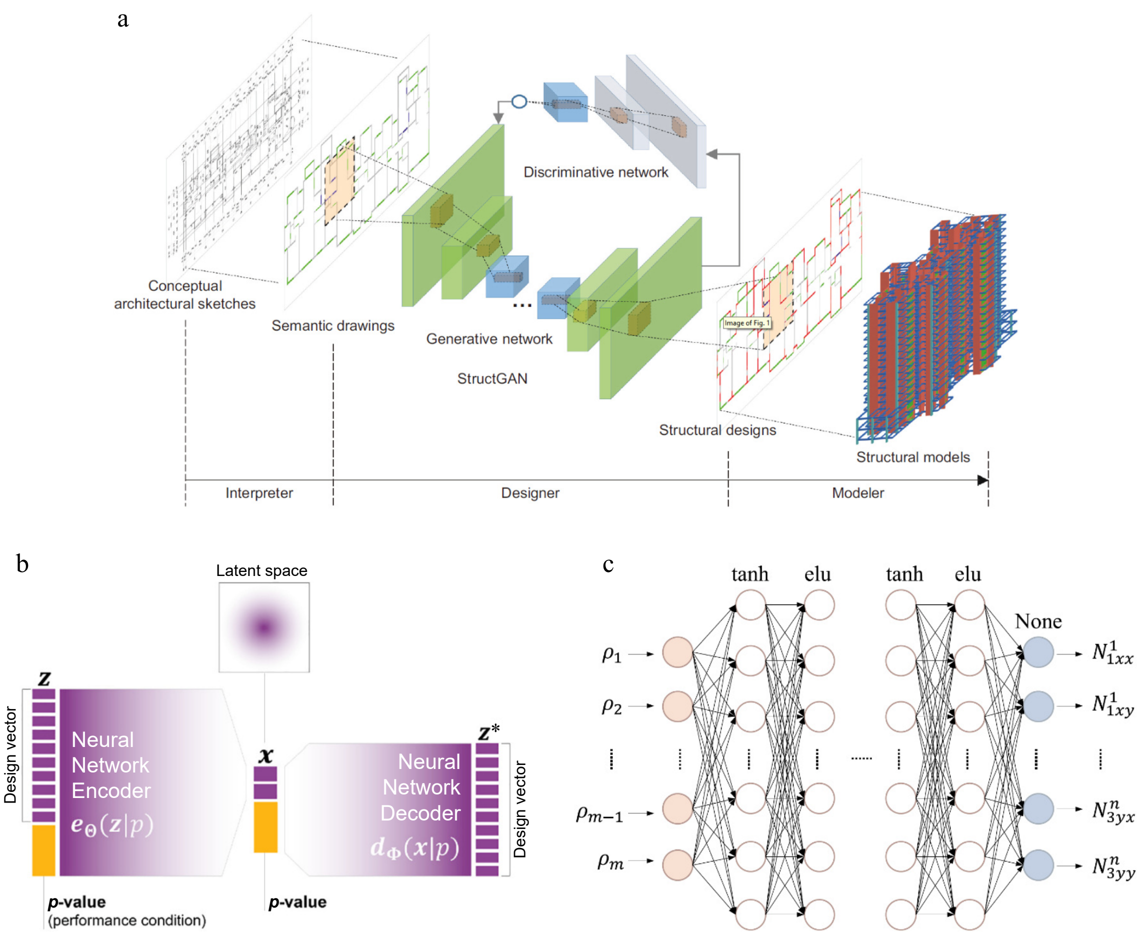

Figure 12 Typical generative AI methods for designing structural systems. (a) GAN for residential building structures. Reproduced with permission from Ref. [141] ©2021, Elsevier B.V. (b) VAE for spatial structures. Reproduced with permission from Ref. [142] ©2021, Elsevier B.V. (c) ANN for continuous structures. Reproduced with permission from Ref. [143] ©2022, Elsevier Ltd.

3.2.3 Designs of structural systems

For the intelligent design of buildings, expert system-based methods were proposed in the 1980s, spurring a series of research progress since then [140]. However, the insufficient computation performance and the limitations of algorithms have hampered their widespread applications. In recent years, AI technology, especially for generative AI algorithms, has emerged as a powerful tool for creatively generating new designs by learning from existing ones, which is gradually applied in the intelligent design of structural systems. Given the vast differences in design methods and rules across different types of structures, generative AI algorithms should be tailored to solve structural design problems with different characteristics. This section focuses on the application of generative AI algorithms in three representative structural forms.

3.2.3.1 Residential building structures

Generative AI has primarily used for designing residential building structures, among which CNN-based generative adversarial networks (GANs) and graph NNs are mainstream algorithms. For instance, Liao et al. [141] proposed an innovative shear wall design method (StructGAN) using GAN algorithm, which learned from existing shear wall designs and subsequently performed efficient and intelligent structural design. Figure 12a illustrates the main components of StructGAN, consisting of the interpreter, designer, and modeler, where the interpreter digests and semanticizes the architectural sketches, the designer analyzes the semantic drawings to perform structural design, and the modeler generates the design and corresponding structural model. Zhao et al. [144] further optimized StructGAN by introducing the attention mechanism to improve the local details in the generated results, with comprehensive evaluation indexes improved up to 30%. Wang et al. [145] adopted GAN and transfer learning to develop a layout generative design framework, which could efficiently design residential layouts with simple inputs and generate corresponding building information modeling models. Generative AI technology has also been applied to various structural forms, such as steel frame–brace structures [146], frame–core tube structures [147], etc.

3.2.3.2 Spatial structures

The shape of spatial structures usually exhibits a high degree of freedom and complexity, whilst variational autoencoder (VAE) algorithms have emerged as a powerful tool for intelligent design of such structure. By manipulating the latent space in VAE, the generation results can be effectively changed, offering valuable insights for various structural design schemes. Specifically, Mirra et

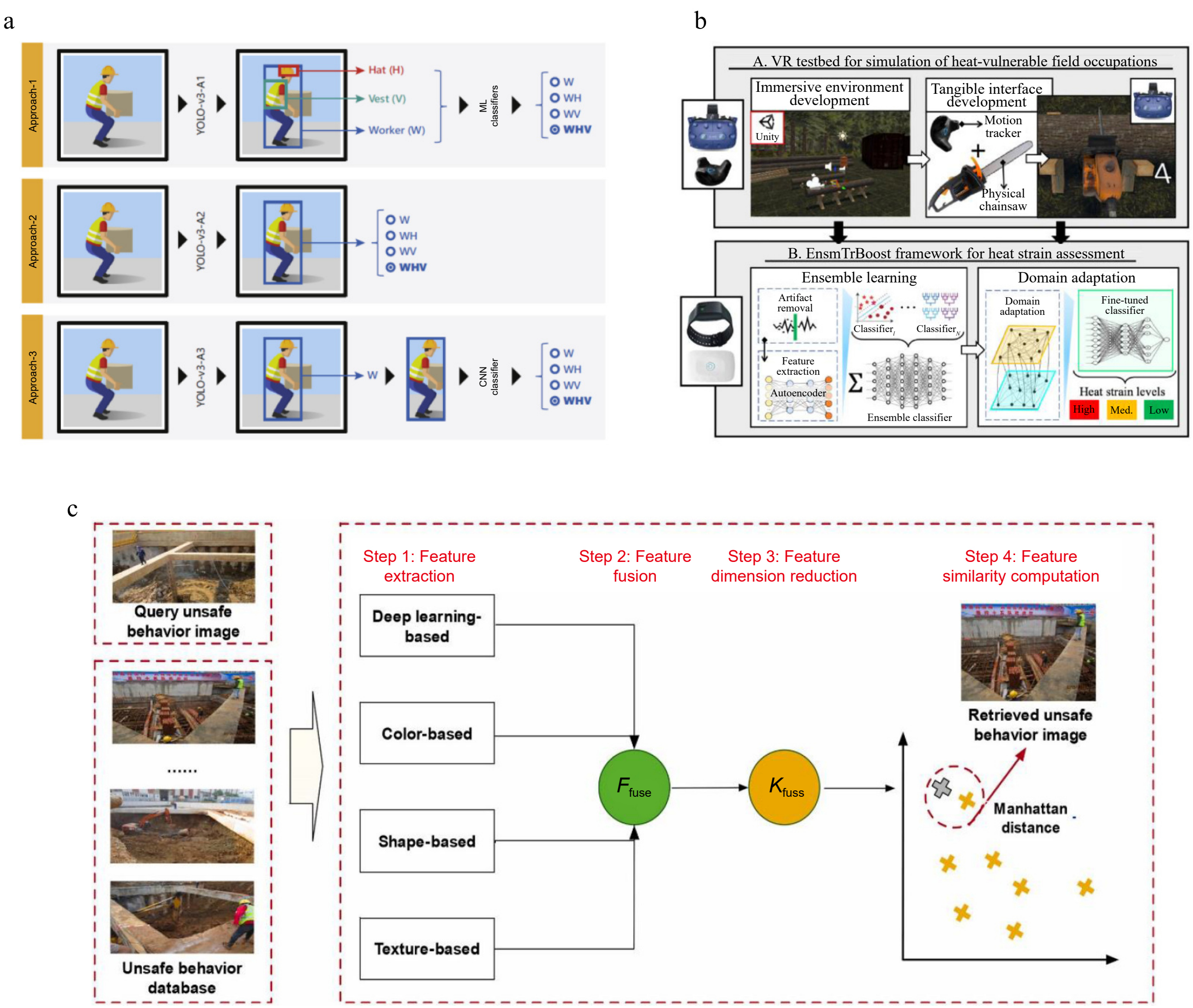

Figure 13 ML methods applied for detecting construction safety. (a) PPE detection. Reproduced with permission from Ref. [152] ©2020, Elsevier B.V. (b) Physiological status monitoring. Reproduced under the CC BY 4.0 license from Ref. [153] ©2024, The Authors. (c) Unsafe behavior identification. Reproduced under the CC BY 4.0 license from Ref. [154] ©2022, The Authors.

al. [148] trained VAE to construct a design space for shell structures, with human-defined design space following the superstructure approach compared. Result demonstrates that AI-generated design space exhibits a greater diversity of design outputs than the solutions predicted by the human-defined spaces. Danhaive et al. [142] also adopted VAE with a low-dimensional latent space for designing shell spatial structures, with its architecture shown in Fig. 12b.

3.2.3.3 Continuous structures

Comparatively, the application of generative AI in the topology design of continuous structures is relatively rare. Kallioras et al. [149] proposed a generative design methodology called DzAIN to optimize beam topology, where deep belief networks were adopted for diversification through randomness, and the input parameters consisted of the domain dimensions, the support and loading conditions, and the desired final volume. As shown in Fig. 12c, Huang et al. [143] adopted ANN to establish the hidden relationship between the shape function of extended multi-scale finite element method and the elementwise material density of a coarse resolution element, with a problem-independent ML technique proposed to improve the efficiency of topology optimization.

3.3 Construction stage

The complex structural forms and harsh service environments of marine structures present great difficulties during the construction stage, highlighting the importance of construction safety. ML has been acknowledged as a robust technique in managing construction safety [150,151], with its typical applications illustrated in Fig. 13.

3.3.1 Personal protective equipment (PPE) detection

Existing statistics reveal that construction workers without PPE are three times more likely to suffer injuries compared to counterparts who wear PPE [155]. Therefore, it is crucial to check whether workers wear PPE as required to protect their safety. Based on YOLOV3 algorithm, Nath et al. [152] established DL models to inspect whether workers wore hard hat and vest, achieving the mean average precision of 72.3% with the speed of 11 frames per second. Li et al. [156] further examined the proper and standardized use of hard hat through DL methods. Considering the huge noise at the construction site, Bangaru et al. [157] developed a smart system that could provide timely feedback to correct any mistake in wearing earplug.

3.3.2 Unsafe behavior identification

Timely detection and prevention of unsafe behaviors among workers can effectively avoid potential risk accidents on construction sites. Traditionally, monitoring unsafe behaviors heavily relies on the human supervision in the past, exhibiting several limitations like time-consuming, labor-intensive and subjectivity-strong. Recently, computer vision techniques have emerged as effective tools for identifying unsafe behaviors. Ding et al. [158] developed a hybrid DL model to detect unsafe behaviors, where a CNN model was adopted to extract visual features from videos, and an LSTM model was employed to analysis such features for unsafe behavior recognition. The developed model achieved excellent detection accuracy, with accuracy being 97% for safe behaviors and 92% for unsafe behaviors. Similarly, based on a transformer-based model developed by Yang et al. [159], both spatial and temporal features were extracted from videos for identifying unsafe behaviors. As shown in Fig. 13c, Fang et al. [154] inspected unsafe behaviors based on fused image features, which were extracted through DL, color, texture, and shaped-based methods.

3.3.3 Physiological status monitoring

Construction workers typically engage in physically demanding tasks and are exposed to harsh environments, which can cause poor physiological status such as fatigue, stress, etc., thereby leading to the reduction in productivity and increment in risk. To address such issues, some researchers have adopted advanced ML methods to monitor physiological status of workers by analyzing signals from wearable sensors [160]. For example, Aryal et al. [161] employed various ML algorithms to evaluate physical fatigue of workers, where several monitoring signals were compared including the heart rate, skin temperature, and electrical activity of brain. Results indicate that the boosted tree classifiers can accurately evaluate fatigue in workers, with best accuracy achieving 82% for combing both heart rate and temperature signals. Similarly, Anwer et al. [162] selected heart rate variability as a monitoring signal, achieving the highest accuracy of 93.5% through RF model. Considering that construction workers are frequently exposed to high temperature, Ojha et al. [153] developed a monitoring framework for worker-centered heat strain collected by wearable biosensors, effectively reducing the risk of heat stress.

3.4 Maintenance stage

The scientific maintenance of marine structures is crucial for preventing catastrophic accidents and ensuring safe service. However, manual visual inspection remains the mainstream methods for evaluating structural status, exhibiting huge limitations like time-consuming, subjective, and dangerous. As illustrated in Fig. 1, many scholars have turned to cutting-edge computer vision technologies to achieve intelligent and robust maintenance, with their applications reviewed herein.

3.4.1 Crack detection

Owing to the inherent properties of concrete materials, most concrete structures inevitably develop cracks during their service life, which imposes adverse effects on their performances. Therefore, the regular crack detection is one of the essential measures to ensure the safe service of such structures. According to the detection approaches and levels, DL-based crack detection can be mainly categorized into the image-based classification, region-based detection, and pixel-based segmentation, with their typical examples showcased in Fig. 14.

Image-based classification aims to determine the presence of cracks within an image, where the image containing cracks is classified as positive, otherwise as negative. This method has been widely employed for detecting cracks in complete images. Gopalakrishnan et al. [163] adopted the VGG16 model combined with transfer learning technology to detect crack in road images, with the classification accuracy achieving 90%. Based on the CNN model, Xu et al. [164] proposed an end-to-end crack detection method using atrous spatial pyramid pooling modules and atrous convolution, attaining an accuracy of 96.37% without pre-training. More recently, Rashid et al. [165] found that CNN could detect cracks in structures even using low-resolution images, achieving the comparable accuracy between low-resolution images with 5 × 5 pixel and high-resolution ones with 50 × 50 pixel. Apart from the

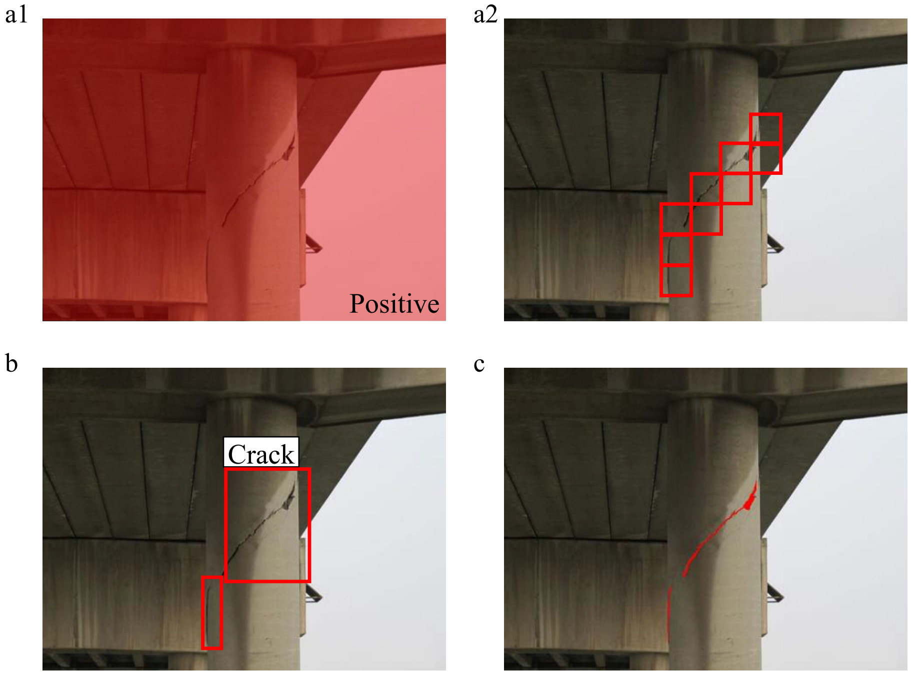

Figure 14 Visualization results of three crack detection methods. (a) Image-based classification: Image level (a1) and patch level (a2). (b) Region-based detection. (c) Pixel-based segmentation.

classification of crack images, research has also focused on quantifying crack details such as length, width, and direction [166].

Some researchers cut the complete images into multiple smaller images for patch-level classification. For example, Cha et al. [167] trained CNN models on 40,000 images, and then scanned a large image using the sliding window technique to localize cracks on structural surfaces. Han et al. [168] proposed a hybrid method based on CNN and image processing technique to detect cracks in concrete structures, achieving 98.26% accuracy. In order to enhance the detection accuracy of sliding window method, a probability map was developed using a softmax layer, and a parametric study was carried out to determine the optimal threshold [169]. Compared with the image-level classification, the patch-level classification can provide the approximate location information of cracks, while its computational efficiency is relatively low, and prediction results needs to be further refined.

Precise location information of target cracks is crucial for evaluating their hazard levels. Object detection algorithms can simultaneously classify and localize cracks, and several region-based detection models are developed using R-CNN, SSD, and YOLO series algorithms. For example, Cha et al. [170] proposed a Faster R-CNN method for inspecting structural defects, achieving real-time detection for five types of defects including concrete cracks. Jiang et al. [171] proposed a DL-based method for efficiently detecting and classifying concrete damages, where the object detection algorithm was optimized through various techniques such as the depthwise separable convolution, inverse ResNet and linear bottleneck structure. Compared with the original YOLO and SSD algorithms, the detection accuracy of their model was improved by 3.25% and 4.04%, respectively, while the detection speed also enhanced. To enable convenient detection, Maeda et al. [172] integrated a deep CNN model into smartphone, achieving high-precision identification for eight types of pavement damages.

Despite the improved efficiency, region-based detection methods fail to provide detailed information such as the specific shape and trajectory of cracks within the generated bounding boxes, potentially hampering the evaluation of crack hazards and the formulation of maintenance decisions. To address these challenges and provide more detailed insights into the crack morphology, recent research has primarily focused on the pixel-level segmentation, with encoder–decoder models being the most popular methods. Specifically, Yang et al. [173] implemented an FCN model to achieve pixel-wise crack detection, and the predicted cracks were represented by single-pixel width skeletons to obtain their quantitative information such as length, width, and topology. Based on the FCN, Li et al. [174] developed a crack segmentation model using transfer learning technique, with pixel accuracy and mean pixel accuracy being 98.61% and 91.59%, respectively. Ge et al. [175] applied parameter-efficient fine-tuning technologies to introduce the segment anything model (SAM) into crack segmentation, and proposed CrackSAM showed excellent performance on different scenes and materials. More recently, Lu et al. [176] developed an integrated detection method for microscopic cracks in concrete dams, where the YOLO-DEW was combined with UNet model. However, DL methods typically rely on a large number of training images, while obtaining such extensive data and labeling them are both time-consuming and resource-intensive. To address this issue, Tabernik et al. [177] developed a two-stage DL architecture for crack segmentation, achieving excellent performance with only 25–30 training images. Zhang et al. [178] proposed a self-supervised network architecture based on GAN, which can be effectively trained even without real labeled images.

3.4.2 Corrosion detection

The image holds abundant information, offering diverse perspectives on the features of described object. For instance, colors and textures of corroded steels can illustrate the corrosion degree, while spatial relationships of features provide the location information. Although certain features in the image are readily perceived by the human vision, a significant amount of which remains undetectable without advanced techniques, thereby promoting the application of ML and DL methods. Considering that corroded steels exhibit distinctive color differences compared to uncorroded ones, the color thus becomes a key feature for ML-based detection models. Lee et al. [179] developed a novel method to detect rust defects in steel bridges, where the mean in red channel, the difference in green channel, and the difference in blue channel were selected as input features for multivariate statistical analysis. Liao et al. [180] adopted the hue percentage combined with the coefficient of variation of gray levels to categorize images into three groups, implementing *K*-means and double-center-double-radius methods to detect corroded areas in each group. For non-uniformly illuminated images, Chen et al. [181] found that a*b* was the best color configuration for rust detection by comparing fourteen color spaces. Apart from the color feature, Hoang et al. [182] employed 78 features for SVM-based corrosion detection model, including statistical properties of colors, gray-level co-occurrence matrix, and gray-level run length. Shen et al. [183] employed color and texture features for rust detection, utilizing Fourier transform combined with color image processing techniques. Similarly, both features were also utilized as input for ANN model to effectively detect corrosion in steel structures [184].

Although ML-based methods can improve the detection accuracy and efficiency, their performances highly depend on the selected features. High-quality features typically require domain knowledges, technical experiences, and comparative experiments, consuming a significant amount of time and resources. However, DL-based algorithms can automatically extract suitable features from images, effectively avoiding manual feature selection and streamlining the model development. Similar to the crack detection reviewed in Section 3.4.1, DL-based corrosion detection can also be categorized into three main levels. For image-based level, Atha et al. [185] adopted various CNN models to achieve corrosion detection, comparing the influences of color spaces, sliding window sizes, and model architectures on the classification performances. Based on more than 140,000 pipeline images with different corrosion degrees, Bastian et al. [186] designed a customized CNN model with very few trainable parameters, achieving the classification accuracy of 98.8%. Forkan et al. [187] developed a corrosion detection framework composed of two DL models, where one identified targeted structures from the background and the other detected specific corrosion regions within such structures. Based on the sliding window algorithm, Yao et al. [188] applied the AlexNet while Papamarkou et al. [189] utilized the ResNet to detect corrosion regions. To further improve the detection accuracy, Zhang et al. [190] embedded the attention mechanism block into the deep ResNet. As mentioned above, such approaches fail to precisely localize corrosion, missing detailed information of corroded regions such as dimensions and shapes, which potentially hinders the accurate diagnosis and targeted maintenance of structures. Some researchers have attempted to employ object detection algorithms for region-based detection.

Based on the YOLOV5s algorithm, Jia et al. [191] developed an automated model to detect corrosion in facilities through bounding boxes, with the precision and recall achieving 90.2% and 90.8%, respectively. Yu et al. [192] proposed an improved YOLOV5 model to identify corroded metals in coastal regions, achieving 4% improvement over the original YOLOV5 model. Wang et al. [193] employed YOLOV7 to localize corroded carbon steels, where the convolution block attention module was introduced to enhance localization performance. Similarly, YOLOV7 was also utilized for localizing bridge corrosion, considering the impact of training database quality [194]. Li et al. [195] employed GAN to augment the database by generating images, and then established the corrosion detection model for steel bridge bolts, with precision and recall being 93.50% and 94.90%, respectively.

Compared with the object detection algorithms, semantic segmentation methods can offer pixel-level localization of corrosion rather than bounding boxes, providing more detailed data supports for structural maintenance. Based on the UNet algorithm and data projection scheme, Katsamenis et al. [196] proposed three-layered simultaneous precise localization and classification architecture employing a UNet model (SPLAC UNet) to localize corrosion and evaluate its grade, with the overall architecture illustrated in Fig. 15a. Compared with the traditional UNet model, their model achieves an error reduction of 14.58% in F_1 score and 13.54% in intersection over union (IoU). Nash et al. [198] developed a DL model to perform pixel-level localization of corrosion, and three Bayesian variants were employed to provide useful confidence levels for each pixel. Xu et al. [197] employed multiple ensemble CNN

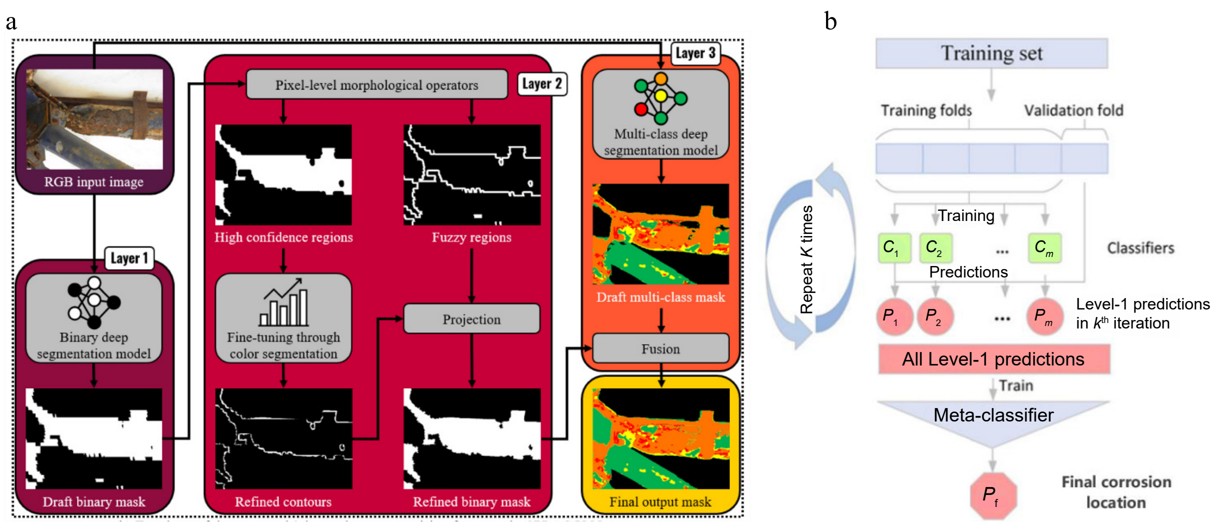

Figure 15 Architectures of pixel-level localization methods for corrosion. (a) Three-layered SPLAC UNet. Reproduced with permission from Ref. [196] © 2022, Elsevier B.V. (b) Ensembled CNN. Reproduced with permission from Ref. [197] © 2020, Computer-Aided Civil and Infrastructure Engineering.

 https://doi.org/10.26599/OCEAN.2025.9470005

Table 5 Comparison of corrosion evaluation methods in existing references

Reference	Application	Corrosion grade	Evaluation criterion
Wang et al. [201], 2018	Steel structure	—	Overall corrosion ratio
Xu et al. [197], 2020	Steel structure	4	Rust ratio and rust color
Sanchez et al. [202], 2020	M4140 steel	3	Corrosion time
Rahman et al. [203], 2021	Facility	3	Grayscale value
Bianchi et al. [204], 2022	Steel bridge	4	BIRM guidelines
Katsamenis et al. [196], 2022	Metal structure	4	ISO 8501-1
Jia et al. [191], 2023	Public facility	3	Color, texture, and other features of metal
Wang et al. [193], 2024	Q235 steel	5	Corrosion time
Guo et al. [205], 2024	Steel bridge	4	BIRM guidelines
Li et al. [195], 2024	Steel bridge bolt	3	USA, Japan, and ISO standards

Note: ISO, International Organization for Standardization.

models to localize corrosion in steel structures, with model structure shown in Fig. 15b. Results reveal that the ensemble model outperforms corresponding single model, and the learning method is more effective than voting method in integrating predictions of base models. Based on cycle GAN, Munawar et al. [199] established an end-to-end corrosion detector for civil infrastructures. Furthermore, CNN was also applied in combined vision and thermographic images to localize corrosion in steel bridges [200].

3.4.3 Corrosion degree evaluation

Comparatively, research on the corrosion degree evaluation remains relatively limited, with details summarized in Table 5. It is observed that the corrosion grades are generally categorized into three or four levels, according to the macroscopic and microscopic indicators such as the proportion of corroded areas, corrosion time, color, texture, etc. Based on the grayscale values of images, Rahman et al. [203] classified corrosion degrees of facilities into three levels. Bianchi et al. [204] established a four-grade assessment criterion following *Bridge Inspector's Reference Manual* (BIRM) guidelines, with both DeepLabV3 and DeepLabV3+ adopted for evaluating corrosion status of steel bridges. The F_1 score and Jaccard index were 88.57% and 83.17% for the optimized DeepLabV3+ model using spectrum loss, respectively. Xu et al. [197] employed ensemble CNN models while Jia et al. [191] adopted YOLOV5s to quantitatively evaluated corrosion degrees.

4 Further research directions and challenges

A comprehensive review is conducted in this paper for the application of various ML algorithms in marine structures in terms of design, construction, and maintenance stages, highlighting the potential of ML in this field. Based on the above review, research directions and challenges in the future are summarized and discussed.

4.1 Establishment of comprehensive databases

Depending on the architecture, each ML algorithm demands different quantity of training data to achieve excellent performance. Compared with ML algorithms, DL approaches like CNN typically requires larger databases due to their vast number of trainable parameters. Training model with limited data tends to result in overfitting or underfitting, leading to poor generalization and unreliable predictions [132,204]. Moreover, the data diversity is also crucial to ensure the model generalization, and the training data should encompass all scopes of potential application scenarios since ML is an excellent learner rather than an extrapolator. As a result, large-scale and high-quality databases are the foundation for effectively applying ML in marine engineering. Unfortunately, acquiring and labelling data generally require resource-consuming experiments or computation-demanding simulations, and the data scarcity is a prolonged challenge in the marine engineering domain. Building a cloud platform to share data is a promising direction for future development, and the performance of ML model can be expected to be further improved. AL methods can serve as valuable tools for researchers to develop targeted data generation plans, and data augmentation techniques, such as generative algorithms, image augmentation methods, etc., can be employed to further enrich the database.

4.2 Determination of appropriate ML algorithm with optimal hyperparameters

Determining the most appropriate ML algorithm for a given issue is both challenging and complicating, which requires comprehensive consideration of multiple factors such as the issue nature, data characteristics, computational efficiency, etc. Despite numerous comparative and benchmark studies that have been conducted in existing literatures to identify suitable algorithms for different topics, a unified selection criterion remains limited, and some findings are somewhat inconsistent. Further research is needed to reasonably select an appropriate algorithm tailored to the certain problem. Hyperparameter optimization is another essential process to achieve the excellent model performance, with several techniques developed like the grid search, Bayesian optimization, particle swarm optimization, etc. However, these methods generally depend on the prediction accuracy and ignore the structural mechanisms underlying the prediction problems [63]. How to reasonably consider the domain-specific principles during hyperparameter optimization is an issue to be addressed, thereby ensuring reliable and robust predictions.

4.3 Interpretation of ML models

The interpretability of ML models is as crucial as their accuracy, especially in safety-critical applications such as the decision-making for marine structures. Clarifying the basis of model predictions is fundamental to ensuring that they reasonably reflect the mechanical principles. Furthermore, interpretability builds trust in ML models, facilitating their integration into decision-making processes.

However, ML models, especially DL models with complex architectures, impose huge challenges to clarify underlying mechanisms behind the model, which hinders the direct application of ML models in marine structures. Although emerging techniques such as feature importance analysis, Shapley additive explanations, and local interpretable model-agnostic explanations can provide valuable insights into the factors influencing model decisions, further research is also needed to focus on developing interpretable and transparent algorithms to make reliable predictions. By bridging the gap between data-driven model and existing engineering principles, ML methods are able to make robust and informed decision-making.

4.4 Implementation of ML models in practice

The ultimate goal of developing ML models is their applications in practical engineering. However, most models developed in literature remains at the theoretical research stage, which have only been attempted as tools to solve certain engineering problems within limited scenarios. Successful implementation in practice requires collaboration between ML experts and structural engineers to bridge the gap between theory and practice, ensuring that ML solutions are both accurate and reliable in real-world settings. The deployment of ML models in practical scenarios encounters substantial challenges regarding variable conditions, limited computational resources, high reliability requirements, etc., which presents further research directions. To accelerate the implementation of ML models in practice, constructing a collaborative platform that facilitates close cooperation between experts from different disciplines is a promising alternative.

5 Conclusion

ML has become a promising technique applied in marine structures, offering a potential alternative to traditional analysis and simulation techniques. This paper provides an in-depth review of existing ML algorithms along with their associated development platforms, and their widespread applications in marine structures across the design, construction, and maintenance stages. An innovative framework is proposed for combining data-driven ML algorithms with mechanical principles in terms of input parameter selection, database subdivision, and hyperparameter determination, which facilitates addressing a major limitation of unclear structural mechanisms in ML models. In addition, the challenges and further research directions are also emphasized and discussed, among which the importance of data quantity, diversity, and accessibility is emphasized due to complex marine conditions. Given that safety-critical applications require models to provide clear and reliable insights, the interpretability of ML model is also highlighted. It is anticipated that this paper can assist structural engineers and scholars without ML backgrounds in developing their own ML models for practical applications.

Acknowledgements

The financial support from Shenzhen Science and Technology Program is highly appreciated.

Author contribution statement

Xiaoguang Zhou: methodology, investigation, data curation, writing—original draft. Chao Hou: conceptualization, methodology, supervision, writing—review & editing. Yantao Yu: investigation, validation, writing—review & editing. Yifan Zhou: investigation, validation, writing—review & editing. All the authors have approved the final manuscript.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declaration of competing interest

All the contributing authors report no conflicts of interest in this work.

Funding

This work is supported by Shenzhen Science and Technology Program (Grant No. RCYX20210706092044076).

Use of AI statement

None.

References

[1] Nie, J. G. (2020). Application of steel–concrete composite structure in ocean engineering. *Steel Constr* 35, 20–33.
[2] Thai, H. T. (2022). Machine learning for structural engineering: A state-of-the-art review. *Structures* 38, 448–491.
[3] Salehi, H., Burgueño, R. (2018). Emerging artificial intelligence methods in structural engineering. *Eng. struct* 171, 170–189.
[4] Tapeh, A. T. G., Naser, M. Z. (2023). Artificial intelligence, machine learning, and deep learning in structural engineering: A scientometrics review of trends and best practices. *Arch. Computat. Methods Eng* 30, 115–159.
[5] Jo, T. (2021). Machine Learning Foundations: Supervised, Unsupervised, and Advanced Learning. Cham: Springer.
[6] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. Cambridge: The MIT Press.
[7] Wang, C., Song, L. H., Yuan, Z., Fan, J. S. (2023). State-of-the-art AI-based computational analysis in civil engineering. *J. Ind. Inf. Integr* 33, 100470.
[8] Adeli, H. (2001). Neural networks in civil engineering: 1989–2000. *Comput.-Aided Civil Infrastruct. Eng* 16, 126–142.
[9] Chojaczyk, A. A., Teixeira, A. P., Neves, L. C., Cardoso, J. B., Guedes Soares, C. (2015). Review and application of artificial neural networks models in reliability analysis of steel structures. *Struct. Saf* 52, 78–89.
[10] Çevik, A., Kurtoğlu, A. E., Bilgehan, M., Gülşan, M. E., Albegmprli, H. M. (2015). Support vector machines in structural engineering: A review. *J. Civil Eng. Manage* 21, 261–281.
[11] Zhang, Q. Y., Barri, K., Jiao, P. C., Salehi, H., Alavi, A. H. (2021). Genetic programming in civil engineering: Advent, applications and future trends. *Artif. Intell. Rev* 54, 1863–1885.
[12] Ali, R., Chuah, J. H., Talip, M. S. A., Mokhtar, N., Shoaib, M. A. (2022). Structural crack detection using deep convolutional neural networks. *Automat. Constr* 133, 103989.
[13] Kumar, R., Althaqafi, E., Patro, S. G. K., Simic, V., Babbar, A., Pamucar, D., Singh, S. K., Verma, A. (2024). Machine and deep learning methods for concrete strength prediction: A bibliometric and content analysis review of research trends and future directions. *Appl. Soft Comput* 164, 111956.
[14] Zhao, Z. F., Liu, Y. J., Lu, Y. Y., Ji, C. Y., Lin, C., Yao, L., Pu, Z. Y., De Brito, J. (2024). Prediction of properties of recycled aggregate

concrete using machine learning models: A critical review. *J. Build. Eng* 90, 109516.

[15] Mirrashid, M., Naderpour, H. (2021). Recent trends in prediction of concrete elements behavior using soft computing (2010–2020). *Arch. Comput. Methods Eng* 28, 3307–3327.

[16] Afshari, S. S., Enayatollahi, F., Xu, X. Y., Liang, X. H. (2022). Machine learning-based methods in structural reliability analysis: A review. *Reliab. Eng. Syst. Saf* 219, 108223.

[17] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev* 65, 386–408.

[18] Moody, J., Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Comput* 1, 281–294.

[19] Jang, J. S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst., Man, Cybern* 23, 665–685.

[20] Elman, J. L. (1990). Finding structure in time. *Cognit. Sci* 14, 179–211.

[21] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Comput* 9, 1735–1780.

[22] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar, p 1724–1734.

[23] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324.

[24] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Netw* 61, 85–117.

[25] Vapnik, V. N. (2000). The Nature of Statistical Learning Theory. 2nd ed. New York: Springer.

[26] Chang, C. C., Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol* 2, 27.

[27] Rasmussen, C. E., Williams, C. K. I. (2006). Gaussian Processes for Machine Learning. Cambridge: MIT Press.

[28] Quinlan, J. R. (1992). C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann Publishers.

[29] Breiman, L., Friedman, J., Olshen, R. A., Stone, C. J. (2017). Classification and Regression Trees. New York: Hall/CRC.

[30] Quinlan, J. R. (1992). Learning with continuous classes. In Proceedings of the 5th Australian Joint Conference on Artificial Intelligence, Hobart, Tasmania, p 343–348.

[31] Wang, Y., Witten, I. H. (1996). Induction of Model Trees for Predicting Continuous Classes. Hamilton: University of Waikato, p 1–12.

[32] Zhou, Z. H. (2012). Ensemble Methods: Foundations and Algorithms. New York: CRC Press.

[33] Breiman, L. (2001). Random forests. *Mach. Learn* 45, 5–32.

[34] Chen, T. Q., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA, p 785–794.

[35] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, USA, p 1097–1105.

[36] Simonyan, K., Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, USA, p 1–14.

[37] Szegedy, C., Liu, W., Jia, Y. Q., Sermanet, P., Reed, S., Anguelov, D. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA, p 1–9.

[38] He, K. M., Zhang, X. Y., Ren, S. Q., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, p 770–778.

[39] Howard, A. G., Zhu, M. L., Chen, B., Kalenichenko, D., Wang, W. J., Weyand, T., Andreetto, M., Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.

[40] Sandler, M., Howard, A., Zhu, M. L., Zhmoginov, A., Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, p 4510–4520.

[41] Howard, A., Sandler, M., Chen, B., Wang, W. J., Chen, L. C., Tan, M. X. (2019). Searching for MobileNetV3. In Proceedings of 2019 IEEE/CVF International Conference on Computer Vision, Seoul, South Korea, p 1314–1324.

[42] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA, p 580–587.

[43] Girshick, R. (2015). Fast R-CNN. In Proceedings of 2015 IEEE International Conference on Computer Vision, Santiago, Chile, p 1440–1448.

[44] Ren, S. Q., He, K. M., Girshick, R., Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell* 39, 1137–1149.

[45] He, K. M., Gkioxari, G., Dollár, P., Girshick, R. (2017). Mask R-CNN. In Proceedings of 2017 IEEE International Conference on Computer Vision, Venice, Italy, p 2980–2988.

[46] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, p 779–788.

[47] Redmon, J., Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, p 6517–6525

[48] Redmon, J., Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv: 1804.02767.

[49] Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934.

[50] Li, C. Y., Li, L. L., Jiang, H. L., Weng, K. H., Geng, Y. F., Li, L., Ke, Z., D., Li, Q. Y., Cheng, M., Nie, W. Q., et al. (2022). YOLOv6: A single-stage object detection framework for industrial applications. arXiv:2209.02976.

[51] Wang, C. Y., Bochkovskiy, A., Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, p 7464–7475.

[52] Wang, C. Y., Yeh, I. H., Liao, H. Y. M. (2025). YOLOv9: Learning what you want to learn using programmable gradient information. In Proceedings of the 18th European Conference on Computer Vision, Milan, Italy, p 1–21.

[53] Wang, A., Chen, H., Liu, L. H., Chen, K., Lin, Z. J., Han, J. G., Ding, G. G. (2024). YOLOv10: Real-time end-to-end object detection. In Proceedings of the 38th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada.

[54] Khanam, R., Hussain, M. (2024). YOLOV11: An overview of the key architectural enhancements. arXiv:2410.17725.

[55] Shelhamer, E., Long, J., Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell* 39, 640–651.

[56] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, p 234–241.

[57] Zhou, Z. W., Rahman Siddiquee, M., Tajbakhsh, N., Liang, J. M. (2018). UNet++: A nested U-Net architecture for medical image

segmentation. In Proceedings of the 4th International Workshop Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Granada, Spain, p 3–11.

[58] Qin, X. B., Zhang, Z. C., Huang, C. Y., Dehghan, M., Zaiane, O. R., Jagersand, M. (2020). U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognit* 106, 107404.

[59] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K. Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv:1412.7062.

[60] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell* 40, 834–848.

[61] Chen, L. C., Papandreou, G., Schroff, F., Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587.

[62] Chen, L. C., Zhu, Y. K., Papandreou, G., Schroff, F., Adam, H. (2018). Encoder–decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, p 833–851.

[63] Zhou, X. G., Hou, C., Peng, J. H., Yao, G. H., Fang, Z. Z. (2023). Structural mechanism-based intelligent capacity prediction methods for concrete-encased CFST columns. *J. Constr. Steel Res* 202, 107769.

[64] Lu, X. Z., Liao, W. J., Zhang, Y., Huang, Y. L. (2022). Intelligent structural design of shear wall residence using physics-enhanced generative adversarial networks. *Earthq. Eng. Struct. Dyn* 51, 1657–1676.

[65] Ben Chaabene, W., Flah, M., Nehdi, M. L. (2020). Machine learning prediction of mechanical properties of concrete: Critical review. *Constr. Build. Mater* 260, 119889.

[66] Kasperkiewicz, J., Racz, J., Dubrawski, A. (1995). HPC strength prediction using artificial neural network. *J. Comput. Civ. Eng* 9, 279–284.

[67] Khan, M. I. (2012). Predicting properties of high performance concrete containing composite cementitious materials using artificial neural networks. *Automat. Constr* 22, 516–524.

[68] Chou, J. S., Tsai, C. F., Pham, A. D., Lu, Y. H. (2014). Machine learning in concrete strength simulations: Multi-nation data analytics. *Constr. Build. Mater* 73, 771–780.

[69] Feng, D. C., Liu, Z. T., Wang, X. D., Chen, Y., Chang, J. Q., Wei, D. F., Jiang, Z. M. (2020). Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. *Constr. Build. Mater* 230, 117000.

[70] Li, Q. F., Song, Z. M. (2022). High-performance concrete strength prediction based on ensemble learning. *Constr. Build. Mater* 324, 126694.

[71] Sun, C., Wang, K., Liu, Q., Wang, P. J., Pan, F. (2023). Machine-learning-based comprehensive properties prediction and mixture design optimization of ultra-high-performance concrete. *Sustainability* 15, 15338.

[72] Kashem, A., Karim, R., Malo, S. C., Das, P., Datta, S. D., Alharthai, M. (2024). Hybrid data-driven approaches to predicting the compressive strength of ultra-high-performance concrete using SHAP and PDP analyses. *Case Stud. Constr. Mater* 20, e02991.

[73] Golafshani, E. M., Behnood, A. (2021). Predicting the mechanical properties of sustainable concrete containing waste foundry sand using multi-objective ANN approach. *Constr. Build. Mater* 291, 123314.

[74] Tran, V. Q., Dang, V. Q., Ho, L. S. (2022). Evaluating compressive strength of concrete made with recycled concrete aggregates using machine learning approach. *Constr. Build. Mater* 323, 126578.

[75] Zhang, S. Y., Chen, W. G., Xu, J. J., Xie, T. Y. (2024). Use of interpretable machine learning approaches for quantificationally understanding the performance of steel fiber-reinforced recycled aggregate concrete: From the perspective of compressive strength and splitting tensile strength. *Eng. Appl. Artif. Intell* 137, 109170.

[76] Sun, Z., Li, Y. L., Li, Y. Q., Su, L., He, W. D. (2024). Investigation on compressive strength of coral aggregate concrete: Hybrid machine learning models and experimental validation. *J. Build. Eng* 82, 108220.

[77] Kumar, A., Arora, H. C., Kapoor, N. R., Mohammed, M. A., Kumar, K., Majumdar, A., Thinnukool, O. (2022). Compressive strength prediction of lightweight concrete: Machine learning models. *Sustainability* 14, 2404.

[78] Salami, B. A., Iqbal, M., Abdulraheem, A., Jalal, F. E., Alimi, W., Jamal, A., Tafsirojjaman, T., Liu, Y., Bardhan, A. (2022). Estimating compressive strength of lightweight foamed concrete using neural, genetic and ensemble machine learning approaches. *Cem. Concr. Compos* 133, 104721.

[79] Pazouki, G., Golafshani, E. M., Behnood, A. (2022). Predicting the compressive strength of self-compacting concrete containing Class F fly ash using metaheuristic radial basis function neural network. *Struct. Concr* 23, 1191–1213.

[80] Long, W. J., Cheng, B. Y., Luo, S. Y., Li, L. X., Mei, L. (2023). Interpretable auto-tune machine learning prediction of strength and flow properties for self-compacting concrete. *Constr. Build. Mater* 393, 132101.

[81] Barbuta, M., Diaconescu, R. M., Harja, M. (2012). Using neural networks for prediction of properties of polymer concrete with fly ash. *J. Mater. Civ. Eng* 24, 523–528.

[82] Sofos, F., Papakonstantinou, C. G., Valasaki, M., Karakasidis, T. E. (2022). Fiber-reinforced polymer confined concrete: Data-driven predictions of compressive strength utilizing machine learning techniques. *Appl. Sci* 13, 567.

[83] Pakzad, S. S., Ghalehnovi, M., Ganjifar, A. (2024). A comprehensive comparison of various machine learning algorithms used for predicting the splitting tensile strength of steel fiber-reinforced concrete. *Case Stud. Constr. Mater* 20, e03092.

[84] Sun, Z., Li, Y. L., Yang, Y. X., Su, L., Xie, S. J. (2024). Splitting tensile strength of basalt fiber reinforced coral aggregate concrete: Optimized XGBoost models and experimental validation. *Constr. Build. Mater* 416, 135133.

[85] Kang, M. C., Yoo, D. Y., Gupta, R. (2021). Machine learning-based prediction for compressive and flexural strengths of steel fiber-reinforced concrete. *Constr. Build. Mater* 266, 121117.

[86] Zheng, D., Wu, R. X., Sufian, M., Kahla, N. B., Atig, M., Deifalla, A. F., Accouche, O., Azab, M. (2022). Flexural strength prediction of steel fiber-reinforced concrete using artificial intelligence. *Materials* 15, 5194.

[87] Han, T. H., Siddique, A., Khayat, K., Huang, J., Kumar, A. (2020). An ensemble machine learning approach for prediction and optimization of modulus of elasticity of recycled aggregate concrete. *Constr. Build. Mater* 244, 118271.

[88] Koya, B. P., Aneja, S., Gupta, R., Valeo, C. (2022). Comparative analysis of different machine learning algorithms to predict mechanical properties of concrete. *Mech. Adv. Mater. Struct* 29, 4032–4043.

[89] Zou, B. P., Wang, Y. B., Amin, M. N., Iftikhar, B., Khan, K., Ali, M., Althoey, F. (2023). Artificial intelligence-based optimized models for predicting the slump and compressive strength of sustainable alkali-derived concrete. *Constr. Build. Mater* 409, 134092.

[90] Pal, A., Ahmed, K. S., Mangalathu, S. (2024). Data-driven machine learning approaches for predicting slump of fiber-reinforced concrete containing waste rubber and recycled aggregate. *Constr. Build. Mater* 417, 135369.

[91] Tran, V. Q. (2022). Machine learning approach for investigating chloride diffusion coefficient of concrete containing supplementary cementitious materials. *Constr. Build. Mater* 328, 127103.

[92] Dada, T. E., Gong, G. B., Xia, J., Di Sarno, L. (2024). Stress–strain behaviour of axially loaded FRP-confined natural and recycled

aggregate concrete using design-oriented and machine learning approaches. *J. Build. Eng* 95, 110256.

[93] Gedam, B. A. (2024). ANN-powered models for predicting shrinkage and creep properties of high-performance concrete using supplementary cementitious materials. *J. Comput. Civ. Eng* 38, 04024033.

[94] Zhang, J. F., Huang, Y. M., Wang, Y. H., Ma, G. W. (2020). Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms. *Constr. Build. Mater* 253, 119208.

[95] Naseri, H., Jahanbakhsh, H., Hosseini, P., Moghadas Nejad, F. (2020). Designing sustainable concrete mixture by developing a new machine learning technique. *J. Clean. Prod* 258, 120578.

[96] Han, L. H., Li, W., Bjorhovde, R. (2014). Developments and advanced applications of concrete-filled steel tubular (CFST) structures: Members. *J. Constr. Steel Res* 100, 211–228.

[97] Hou, C., Zhou, X. G. (2022). Strength prediction of circular CFST columns through advanced machine learning methods. *J. Build. Eng* 51, 104289

[98] Hao, Y. E., Zhai, Z. D. (2004). Evaluation of axial bearing capacity of filling steel tube concrete columns with rectangular section. *J. Chang'an Univ. (Arch. Envir. Sci. Ed.)* 21, 24–28.

[99] Ahmadi, M., Naderpour, H., Kheyroddin, A. (2014). Utilization of artificial neural networks to prediction of the capacity of CCFT short columns subject to short term axial load. *Arch. Civ. Mech. Eng* 14, 510–517.

[100] Güneyisi, E. M., Gültekin, A., Mermerdaş, K. (2016). Ultimate capacity prediction of axially loaded CFST short columns. *Int. J. Steel Struct* 16, 99–114.

[101] Du, Y. S., Chen, Z. H., Zhang, C. Q., Cao, X. C. (2017). Research on axial bearing capacity of rectangular concrete-filled steel tubular columns based on artificial neural networks. *Front. Comput. Sci* 11, 863–873.

[102] Tran, V. L., Thai, D. K., Kim, S. E. (2019). Application of ANN in predicting ACC of SCFST column. *Compos. Struct* 228, 111332.

[103] Nour, A. I., Güneyisi, E. M. (2019). Prediction model on compressive strength of recycled aggregate concrete filled steel tube columns. *Compos. Part B: Eng* 173, 106938.

[104] Zarringol, M., Thai, H. T., Thai, S., Patel, V. (2020). Application of ANN to the design of CFST columns. *Structures* 28, 2203–2220.

[105] Mai, S. H., El Amine Ben Seghier, M., Nguyen, P. L., Jafari-Asl, J., Thai, D. K. (2022). A hybrid model for predicting the axial compression capacity of square concrete-filled steel tubular columns. *Eng. Comput* 38, 1205–1222.

[106] Le, T. T. (2022). Practical machine learning-based prediction model for axial capacity of square CFST columns. *Mech. Adv. Mater. Struct* 29, 1782–1797.

[107] Tran, V. L., Thai, D. K., Nguyen, D. D. (2020). Practical artificial neural network tool for predicting the axial compression capacity of circular concrete-filled steel tube columns with ultra-high-strength concrete. *Thin-Walled Struct* 151, 106720.

[108] Naser, M. Z., Thai, S., Thai, H. T. (2021). Evaluating structural response of concrete-filled steel tubular columns through machine learning. *J. Build. Eng* 34, 101888.

[109] Lee, S., Vo, T. P., Thai, H. T., Lee, J., Patel, V. (2021). Strength prediction of concrete-filled steel tubular columns using categorical gradient boosting algorithm. *Eng. Struct* 238, 112109.

[110] Vu, Q. V., Truong, V. H., Thai, H. T. (2021). Machine learning-based prediction of CFST columns using gradient tree boosting algorithm. *Compos. Struct* 259, 113505.

[111] Faridmehr, I., Nehdi, M. L. (2022). Predicting axial load capacity of CFST columns using machine learning. *Struct. Concr* 23, 1642–1658.

[112] Zhou, X. G., Hou, C., Feng, W. Q. (2023). Optimized data-driven machine learning models for axial strength prediction of rectangular CFST columns. *Structures* 47, 760–780.

[113] Basarir, H., Elchalakani, M., Karrech, A. (2019). The prediction of ultimate pure bending moment of concrete-filled steel tubes by adaptive neuro-fuzzy inference system (ANFIS). *Neural Comput. Appl* 31, 1239–1252.

[114] Hanoon, A. N., Al Zand, A. W., Yaseen, Z. M. (2022). Designing new hybrid artificial intelligence model for CFST beam flexural performance prediction. *Eng. Comput* 38, 3109–3135.

[115] Gao, H. G. (2011). Calculation of load-carrying capacity of square concrete filled tube columns based on neural network. *Appl. Mech. Mater* 71–78, 847–850.

[116] Saadoon, A. S., Nasser, K. Z., Mohamed, I. Q. (2012). A neural network model to predict ultimate strength of rectangular concrete filled steel tube beam–columns. *Eng. Technol. J* 30, 3328–3340.

[117] Luat, N. V., Lee, J., Lee, D. H., Lee, K. (2020). GS-MARS method for predicting the ultimate load-carrying capacity of rectangular CFST columns under eccentric loading. *Comput. Concr* 25, 1–14.

[118] Hou, C., Zhou, X. G. (2022). Eccentric compression capacity prediction of rectangular CFST columns based on machine learning. *J. Build. Struct* 43, 155–166.

[119] Nguyen, M. S. T., Trinh, M. C., Kim, S. E. (2022). Uncertainty quantification of ultimate compressive strength of CCFST columns using hybrid machine learning model. *Eng. Comput* 38, 2719–2738.

[120] Hou, C., Zhou, X. G., Shen, L. M. (2023). Intelligent prediction methods for *N–M* interaction of CFST under eccentric compression. *Archiv. Civ. Mech. Eng* 23, 197.

[121] Wang, C., Chan, T. M. (2023). Machine learning (ML) based models for predicting the ultimate strength of rectangular concrete-filled steel tube (CFST) columns under eccentric loading. *Eng. Struct* 276, 115392.

[122] Yeong, C. H., Li, W. (2024). Prediction of load-deformation relations for CFDST columns through machine learning methods. *J. Constr. Steel Res* 223, 108998.

[123] Zarringol, M., Thai, H. T. (2022). Prediction of the load-shortening curve of CFST columns using ANN-based models. *J. Build. Eng* 51, 104279.

[124] Le, D. N., Pham, T. H., Pham, T. D., Kong, Z. Y., Papazafeiropoulos, G., Vu, Q. V. (2024). An efficient long short-term memory-based model for prediction of the load-displacement curve of concrete-filled double-skin steel tubular columns. *Constr. Build. Mater* 449, 138122.

[125] Fan, X. Y., Lyu, F., Fan, J. L., Ding, F. X. (2023). Prediction of axial compressive load–strain curves of circular concrete-filled steel tube columns using long short-term memory network. *Materials* 16, 3285.

[126] Zhang, G. J., Liu, Y. J., Liu, J., Lan, S. Y., Yang, J. (2022). Causes and statistical characteristics of bridge failures: A review. *J. Traffic Transp. Eng. (Engl. Ed.)* 9, 388–406.

[127] Yang, X. Q., Yang, H., Zhang, Z. Q., Zhu, Y., Lai, Z. C. (2022). Impact resistance and simplified evaluation method for square CFST members subjected to transverse impact. *J. Constr. Steel Res* 198, 107522.

[128] Yang, G. C., Yang, R., Zhang, J. (2023). Prediction and interpretation of residual bearing capacity of CFST columns under impact loads based interpretable stacking fusion modeling. *Buildings* 13, 2783.

[129] Li, J., Pang, Y. F., Wang, K. S., Zhang, X. J., Wang, N. (2024). Explainable machine-learning model for rapid damage assessment of CFST columns after close-in explosion. *J. Perform. Constr. Facil* 38, 04024010.

[130] Li, J., Pang, Y. F., Mu, Q. P., Zhang, X. J., Shi, Y. C., Wang, H. L. (2023). Post-blast capacity evaluation of concrete-filled steel tubular (CFST) column based on machine learning technique. *Adv. Struct. Eng* 26, 1953–1972.

[131] He, J. G., Jiang, L. Q., Jiang, L. Z., Wen, T. X., Hu, Y., Guo, W., Sun, J. S. (2023). Estimation of blast-induced peak response of concrete-filled double-skin tube columns by intelligence-based technique. *Thin-Walled Struct* 186, 110670.

[132] Lai, D. D., Demartino, C., Xiao, Y. (2024). Probabilistic machine leaning models for predicting the maximum displacements of concrete-filled steel tubular columns subjected to lateral impact loading. *Eng. Appl. Artif. Intell* 135, 108704.

[133] Zhou, X. G., Hou, C., Peng, J. H. (2023). Active learning methods for strength assessment of circular CFST under coupled long-term axial loading and random localized corrosion. *Thin-Walled Struct* 193, 111254.

[134] Al-Khaleefi, A. M., Terro, M. J., Alex, A. P., Wang, Y. (2002). Prediction of fire resistance of concrete filled tubular steel columns using neural networks. *Fire Saf. J* 37, 339–352.

[135] Zhao, X. Y., Chen, J. X., Wu, B. (2022). An interpretable ensemble-learning-based open source model for evaluating the fire resistance of concrete-filled steel tubular columns. *Eng. Struct* 270, 114886.

[136] Xu, L., Ren, M. Y., Li, X. (2024). Knowledge distillation for fire resistance of concrete-filled steel tubular columns: Teacher–student models. *J. Constr. Steel Res* 219, 108762.

[137] Wu, B. R., Dang, S. C., Zhu, Y. F., Yao, Y. (2023). A machine learning based interaction model to predict robustness of concrete-filled double skin steel tubular columns under fire condition. *Structures* 57, 105332.

[138] Narang, A., Kumar, R., Dhiman, A. (2024). Residual strength index prediction of circular concrete-filled steel tubular columns through advanced machine learning methods. *Asian J. Civ. Eng* 25, 747–760.

[139] Moradi, M. J., Daneshvar, K., Ghazi-Nader, D., Hajiloo, H. (2021). The prediction of fire performance of concrete-filled steel tubes (CFST) using artificial neural network. *Thin-Walled Struct* 161, 107499.

[140] Liao, W. J., Lu, X. Z., Fei, Y. F., Gu, Y., Huang, Y. L. (2024). Generative AI design for building structures. *Automat. Constr* 157, 105187.

[141] Liao, W. J., Lu, X. Z., Huang, Y. L., Zheng, Z., Lin, Y. Q. (2021). Automated structural design of shear wall residential buildings using generative adversarial networks. *Automat. Constr* 132, 103931.

[142] Danhaive, R., Mueller, C. T. (2021). Design subspace learning: Structural design space exploration using performance-conditioned generative modeling. *Automat. Constr* 127, 103664.

[143] Huang, M. C., Du, Z. L., Liu, C., Zheng, Y. G., Cui, T. C., Mei, Y., Li, X., Zhang, X. Y., Guo, X. (2022). Problem-independent machine learning (PIML)-based topology optimization—A universal approach. *Extreme Mech. Lett* 56, 101887.

[144] Zhao, P. J., Liao, W. J., Huang, Y. L., Lu, X. Z. (2023). Intelligent design of shear wall layout based on attention-enhanced generative adversarial network. *Eng. Struct* 274, 115170.

[145] Wang, L. F., Zhou, X. H., Liu, J. P., Cheng, G. Z. (2024). Automated layout generation from sites to flats using GAN and transfer learning. *Automat. Constr* 166, 105668.

[146] Fu, B. C., Gao, Y. Q., Wang, W. (2023). Dual generative adversarial networks for automated component layout design of steel frame–brace structures. *Automat. Constr* 146, 104661.

[147] Fei, Y. F., Liao, W. J., Huang, Y. L., Lu, X. Z. (2022). Knowledge-enhanced generative adversarial networks for schematic design of framed tube structures. *Automat. Constr* 144, 104619.

[148] Mirra, G., Pugnale, A. (2021). Comparison between human-defined and AI-generated design spaces for the optimisation of shell structures. *Structures* 34, 2950–2961.

[149] Kallioras, N. A., Lagaros, N. D. (2020). DzAIℕ: Deep learning based generative design. *Proc. Manuf* 44, 591–598.

[150] Liu, J. J., Luo, H. B., Liu, H. (2022). Deep learning-based data analytics for safety in construction. *Automat. Constr* 140, 104302.

[151] Akinosho, T. D., Oyedele, L. O., Bilal, M., Ajayi, A. O., Delgado, M. D., Akinade, O. O., Ahmed, A. A. (2020). Deep learning in the construction industry: A review of present status and future innovations. *J. Build. Eng* 32, 101827.

[152] Nath, N. D., Behzadan, A. H., Paal, S. G. (2020). Deep learning for site safety: Real-time detection of personal protective equipment. *Automat. Constr* 112, 103085.

[153] Ojha, A., Shayesteh, S., Sharifironizi, A., Liu, Y. Z., Jebelli, H. (2024). Worker-centric heat strain analysis: Integrating physiological signals with ensemble learning and domain adaptation. *Automat. Constr* 166, 105670.

[154] Fang, W. L., Love, P. E. D., Luo, H. B., Xu, S. J. (2022). A deep learning fusion approach to retrieve images of People's unsafe behavior from construction sites. *Dev. Built Environ* 12, 100085.

[155] Lette, A., Ambelu, A., Getahun, T., Mekonen, S. (2018). A survey of work-related injuries among building construction workers in southwestern Ethiopia. *Int. J. Ind. Ergon* 68, 57–64.

[156] Li, J. Q., Zhao, X. F., Zhou, G. Y., Zhang, M. Y. (2022). Standardized use inspection of workers' personal protective equipment based on deep learning. *Saf. Sci* 150, 105689.

[157] Bangaru, S. S., Wang, C., Zhou, X., Jeon, H. W., Li, Y. L. (2020). Gesture recognition-based smart training assistant system for construction worker earplug-wearing training. *J. Constr. Eng. Manage* 146, 04020144.

[158] Ding, L. Y., Fang, W. L., Luo, H. B., Love, P. E. D., Zhong, B. T., Ouyang, X. (2018). A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory. *Automat. Constr* 86, 118–124.

[159] Yang, M., Wu, C. K., Guo, Y. J., Jiang, R., Zhou, F. X., Zhang, J. L., Yang, Z. L. (2023). Transformer-based deep learning model and video dataset for unsafe action identification in construction projects. *Automat. Constr* 146, 104703.

[160] Fang, W. L., Wu, D. R., Love, P. E. D., Ding, L. Y., Luo, H. B. (2022). Physiological computing for occupational health and safety in construction: Review, challenges and implications for future research. *Adv. Eng. Inf* 54, 101729.

[161] Aryal, A., Ghahramani, A., Becerik-Gerber, B. (2017). Monitoring fatigue in construction workers using physiological measurements. *Automat. Constr* 82, 154–165.

[162] Anwer, S., Li, H., Umer, W., Antwi-Afari, M. F., Mehmood, I., Yu, Y. T., Haas, C., Wong, A. Y. L. (2023). Identification and classification of physical fatigue in construction workers using linear and nonlinear heart rate variability measurements. *J. Constr. Eng. Manage* 149, 04023057.

[163] Gopalakrishnan, K., Khaitan, S. K., Choudhary, A., Agrawal, A. (2017). Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Constr. Build. Mater* 157, 322–330.

[164] Xu, H. Y., Su, X., Wang, Y., Cai, H. Y., Cui, K. R., Chen, X. D. (2019). Automatic bridge crack detection using a convolutional neural network. *Appl. Sci* 9, 2867.

[165] Rashid, T., Mokji, M. M., Rasheed, M. (2024). Cracked concrete surface classification in low-resolution images using a convolutional neural network. *J. Opt.* in press, https://doi.org/10.1007/s12596-024-02080-w.

[166] Flah, M., Suleiman, A. R., Nehdi, M. L. (2020). Classification and quantification of cracks in concrete structures using deep learning image-based techniques. *Cem. Concr. Compos* 114, 103781.

[167] Cha, Y. J., Choi, W., Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Comput. -Aided Civ. Infrastruct. Eng* 32, 361–378.

[168] Han, X. J., Zhao, Z. C., Chen, L. K., Hu, X. L., Tian, Y., Zhai, C. C., Wang, L., Huang, X. M. (2022). Structural damage-causing concrete cracking detection based on a deep-learning method. *Constr. Build. Mater* 337, 127562.

[169] Kim, B., Cho, S. (2018). Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. *Sensors*. 18, 3452.

[170] Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based

deep learning for detecting multiple damage types. *Comput.-Aided Civ. Infrastruct. Eng* 33, 731–747.
[171] Jiang, Y. Q., Pang, D. D., Li, C. D. (2021). A deep learning approach for fast detection and classification of concrete damage. *Automat. Constr* 128, 103785.
[172] Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. *Comput.-Aided Civ. Infrastruct. Eng* 33, 1127–1141.
[173] Yang, X. C., Li, H., Yu, Y. T., Luo, X. C., Huang, T., Yang, X. (2018). Automatic pixel-level crack detection and measurement using fully convolutional network. *Comput.-Aided Civ. Infrastruct. Eng* 33, 1090–1109.
[174] Li, S. Y., Zhao, X. F., Zhou, G. Y. (2019). Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. *Comput.-Aided Civ. Infrastruct. Eng* 34, 616–634.
[175] Ge, K., Wang, C., Guo, Y. T., Tang, Y. S., Hu, Z. Z., Chen, H. B. (2024). Fine-tuning vision foundation model for crack segmentation in civil infrastructures. *Constr. Build. Mater* 431, 136573.
[176] Lu, X. C., Li, Q. Q., Li, J. Y., Zhang, L. (2025). Deep learning-based method for detection and feature quantification of microscopic cracks on the surface of concrete dams. *Measurement* 240, 115587.
[177] Tabernik, D., Šela, S., Skvarč, J., Skočaj, D. (2020). Segmentation-based deep-learning approach for surface-defect detection. *J. Intell. Manuf* 31, 759–776.
[178] Zhang, K. G., Zhang, Y. T., Cheng, H. D. (2020). Self-supervised structure learning for crack detection based on cycle-consistent generative adversarial networks. *J. Comput. Civ. Eng* 34, 04020004.
[179] Lee, S., Chang, L. M., Skibniewski, M. (2006). Automated recognition of surface defects using digital color image processing. *Automat. Constr* 15, 540–549.
[180] Liao, K. W., Lee, Y. T. (2016). Detection of rust defects on steel bridge coatings via digital image recognition. *Automat. Constr* 71, 294–306.
[181] Chen, P. H., Yang, Y. C., Chang, L. M. (2009). Automated bridge coating defect recognition using adaptive ellipse approach. *Automat. Constr* 18, 632–643.
[182] Hoang, N. D., Tran, V. D. (2019). Image processing-based detection of pipe corrosion using texture analysis and metaheuristic-optimized machine learning approach. *Comput. Intell. Neurosci* 2019, 8097213.
[183] Shen, H. K., Chen, P. H., Chang, L. M. (2013). Automated steel bridge coating rust defect recognition method based on color and texture feature. *Automat. Constr* 31, 338–356.
[184] Khayatazad, M., Honhon, M., de Waele, W. (2023). Detection of corrosion on steel structures using an artificial neural network. *Struct. Infrastruct. Eng* 19, 1860–1871.
[185] Atha, D. J., Jahanshahi, M. R. (2018). Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Struct. Health Monit* 17, 1110–1128.
[186] Bastian, B. T., Jaspreeth, N., Ranjith, S. K., Jiji, C. V. (2019). Visual inspection and characterization of external corrosion in pipelines using deep neural network. *NDT E Int* 107, 102134.
[187] Forkan, A. R. M., Kang, Y. B., Jayaraman, P. P., Liao, K. W., Kaul, R., Morgan, G., Ranjan, R., Sinha, S. (2022). CorrDetector: A framework for structural corrosion detection from drone images using ensemble deep learning. *Expert Syst. Appl* 193, 116461.
[188] Yao, Y., Yang, Y., Wang, Y. P., Zhao, X. F. (2019). Artificial intelligence-based hull structural plate corrosion damage detection and recognition using convolutional neural network. *Appl. Ocean Res* 90, 101823.
[189] Papamarkou, T., Guy, H., Kroencke, B., Miller, J., Robinette, P., Schultz, D., Hinkle, J., Pullum, L., Schuman, C., Renshaw, J., et al. (2021). Automated detection of corrosion in used nuclear fuel dry storage canisters using residual neural networks. *Nuclear Eng. Technol* 53, 657–665.
[190] Zhang, S., Deng, X. L., Lu, Y. M., Hong, S. Z., Kong, Z. Y., Peng, Y. L., Luo, Y. (2021). A channel attention based deep neural network for automatic metallic corrosion detection. *J. Build. Eng* 42, 103046.
[191] Jia, Z. T., Fu, M. J., Zhao, X. D., Cui, Z. D. (2023). Intelligent identification of metal corrosion based on corrosion-YOLOv5s. *Displays* 76, 102367.
[192] Yu, Q. F., Han, Y. D., Lin, W. G., Gao, X. J. (2024). Detection and analysis of corrosion on coated metal surfaces using enhanced YOLOv5 algorithm for anti-corrosion performance evaluation. *J. Mar. Sci. Eng* 12, 1090.
[193] Wang, Q. J., Gong, H. Y., Fu, Z. H., Zhang, D. W. (2024). Automatic detection of carbon steel corrosion degree based on image identification. *Comput. Mater. Sci* 233, 112717.
[194] Wang, J., Sun, H. B., Xiong, W., Han, G., Wu, D. (2024). Recognition of steel bridge corrosion considering the effect of training dataset quality. *Case Stud. Constr. Mater* 21, e03722.
[195] Li, Z. J., Shao, P., Zhao, M. H., Yan, K., Liu, G. X., Wan, L., Xu, X. L., Li, K. L. (2024). Optimized deep learning for steel bridge bolt corrosion detection and classification. *J. Constr. Steel Res* 215, 108570.
[196] Katsamenis, I., Doulamis, N., Doulamis, A., Protopapadakis, E., Voulodimos, A. (2022). Simultaneous precise localization and classification of metal rust defects for robotic-driven maintenance and prefabrication using residual attention U-Net. *Automat. Constr* 137, 104182.
[197] Xu, J., Gui, C. Q., Han, Q. H. (2020). Recognition of rust grade and rust ratio of steel structures based on ensembled convolutional neural network. *Comput.-Aided Civ. Infrastruct. Eng* 35, 1160–1174.
[198] Nash, W., Zheng, L., Birbilis, N. (2022). Deep learning corrosion detection with confidence. *npj Mater. Degrad* 6, 26.
[199] Munawar, H. S., Ullah, F., Shahzad, D., Heravi, A., Qayyum, S., Akram, J. (2022). Civil infrastructure damage and corrosion detection: An application of machine learning. *Buildings* 12, 156.
[200] Lim, H. J., Hwang, S., Kim, H., Sohn, H. (2021). Steel bridge corrosion inspection with combined vision and thermographic images. *Struct. Health Monit* 20, 3424–3435.
[201] Wang, D. L., Peng, B., Pan, Y., Chen, A. R. (2018). Segmentation and quantitative analysis of corrosion images based on deep neural networks. *J. South China Univ. Technol. (Nat. Sci. Ed.)* 46, 121–127, 146.
[202] Sanchez, G., Aperador, W., Cerón, A. (2020). Corrosion grade classification: A machine learning approach. *Indian Chem. Eng* 62, 277–286.
[203] Rahman, A., Wu, Z. Y., Kalfarisi, R. (2021). Semantic deep learning integrated with RGB feature-based rule optimization for facility surface corrosion detection and evaluation. *J. Comput. Civ. Eng* 35, 04021018.
[204] Bianchi, E., Hebdon, M. (2022). Development of extendable open-source structural inspection datasets. *J. Comput. Civ. Eng* 36, 04022039.
[205] Guo, J. R., Wang, L., Hua, L. (2024). Efficient metal corrosion area detection model combining convolution and transformer. *Appl. Sci* 14, 9900.