Engineering Science and Technology, an International Journal 70 (2025) 102159

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full Length Article

Transparent and reliable construction cost prediction using advanced machine learning and explainable AI

Lifei Chen [a], Changyong Xu [b], Wei Hong Lim [a,*], Abhishek Sharma [c], Sew Sun Tiang [a], Kim Soon Chong [a], El-Sayed M. El-kenawy [d,e], Amel Ali Alhussan [f], Marwa M. Eid [g,h], Doaa Sami Khafaga [f]

[a] Faculty of Engineering, Technology and Built Environment, UCSI University, Kuala Lumpur 56000, Malaysia
[b] Graduate Business School, UCSI University, Kuala Lumpur 56000, Malaysia
[c] Department of Computer Science and Engineering, Graphic Era Deemed to Be University, Dehradun 248002, India
[d] School of ICT, Faculty of Engineering, Design and Information & Communications Technology (EDICT), Bahrain Polytechnic, PO Box 33349 Isa Town, Bahrain
[e] Applied Science Research Center. Applied Science Private University, Amman, Jordan
[f] Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[g] Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura, Egypt
[h] Jadara Research Center, Jadara University, Irbid 21110, Jordan

ARTICLE INFO

Keywords:
Construction cost prediction
Machine learning
Confidence intervals
Explainable AI
Ensemble methods
SHAP

ABSTRACT

Accurate construction cost prediction is vital for project management, influencing budgeting, resource allocation, and overall success. This study proposes a comprehensive framework that combines machine learning models, uncertainty quantification through Confidence Intervals, and explainable AI techniques using SHAP (SHapley Additive exPlanations) to enhance transparency and decision-making. Ten machine learning models, including Ridge Regression, Lasso Regression, Elastic Net, K-Nearest Neighbor Regression, and advanced ensemble methods such as XGBoost, CatBoost, and HistGradient Boosting, were evaluated on the RSMeans dataset. Among these, HistGradient Boosting achieved the best performance on the testing dataset. Beyond traditional metrics, Confidence Intervals quantified prediction reliability, and SHAP identified critical cost drivers like "Formwork" and "Tributary Area," enabling interpretable and robust prediction. This study highlights the potential of machine learning models to revolutionize construction cost estimation by integrating predictive accuracy, uncertainty analysis, and explainability. The proposed framework supports resource efficiency and enables process innovation in cost management. It also contributes to the advancement of sustainable building practices, offering a strong foundation for future research and promoting the adoption of machine learning-based solutions with enhanced transparency and confidence.

1. Introduction

Accurate construction cost prediction is essential for the success of construction projects, as it facilitates effective budget control and provides reliable data to support informed decision-making [1]. For instance, during the expansion of Terminal 3 at San Francisco International Airport (SFO), the project team conducted a comprehensive cost estimation process, accounting for materials, labor, and time. By identifying potential risks and cost escalators early and implementing mitigation strategies, the project was completed below budget, achieving approximately 10 % cost savings [2]. Similarly, the 'Paris Eco-House' in France, designed as an environmentally friendly, low-energy building, prioritized cost estimation and savings from the outset. By utilizing renewable materials and energy-efficient equipment, the project successfully controlled construction costs and realized significant operational cost reductions [3]. In another example, the renovation of Amsterdam Central Station employed detailed capital and material estimates through the modeling and simulation techniques. This

* Corresponding author.
E-mail addresses: 1002372862@ucsiuniversity.edu.my (L. Chen), xcy123110@qq.com (C. Xu), limwh@ucsiuniversity.edu.my (W.H. Lim), tiangss@ucsiuniversity.edu.my (S.S. Tiang), ChongKS@ucsiuniversity.edu.my (K.S. Chong), skenawy@ieee.org (E.-S.M. El-kenawy), aaalhussan@pnu.edu.sa (A.A. Alhussan), mmm@ieee.org (M.M. Eid), dskhafga@pnu.edu.sa (D.S. Khafaga).

https://doi.org/10.1016/j.jestch.2025.102159
Received 9 January 2025; Received in revised form 7 July 2025; Accepted 23 July 2025
Available online 30 July 2025

approach reduced overall project costs by approximately 15 % compared to the initial budget, while maintaining high-quality standards and public satisfaction [4]. These case studies emphasize the critical role of accurate upfront cost estimation in ensuring cost control and establishing a solid foundation for successful project implementation. In a broader context, inaccurate cost predictions can contribute to housing unaffordability, delays in public infrastructure delivery, and inefficient use of public and private sector budgets. These consequences highlight the societal urgency and economic importance of advancing predictive capabilities, especially as construction activities continue to expand across both developing and developed nations.

Cost forecasting is a critical component of project management, essential for maintaining financial health and ensuring successful project completion. However, achieving highly accurate forecasts remains a significant challenge due to numerous complex factors. External uncertainties, such as economic fluctuations, changes of policies and regulations, market price volatility, and natural disasters, can significantly affect construction cost [5]. In addition, the project-specific complexities, including frequent design changes, complex construction techniques, and unforeseen workloads can distort the cost predictions [6]. The expertise and skills of the project team also play a pivotal role in forecasting accuracy. Inexperienced estimators may introduce errors due to limited understanding of cost components and influencing factors, often resulting in subjective judgments. Moreover, the absence of effective cost management tools and methodologies complicates the tracking and control of project expenses [7]. Given these challenges, accurate cost forecasting requires a comprehensive approach that considers multiple variables. Project managers should prioritize the optimization of forecasting models, leverage historical data for analytical insights, and implement robust cost control mechanisms from the project's inception. By adopting these strategies, project teams can manage financial resources more effectively, enhancing the likelihood of successful project outcomes.

Traditional construction cost prediction methods include the Delphi method, value engineering, target costing, work breakdown structure (WBS), and cost-benefit analysis. The Delphi method [8] employs a systematic process to gather expert insights through multiple rounds of anonymous questionnaires, enabling a convergence of opinions on cost estimates. It is particularly effective in addressing the scenarios with high uncertainty and limited historical data. The value engineering method [9] evaluates project functions and costs to identify opportunities for enhancing value while reducing expenses without compromising quality, ensuring efficient resource allocation. The target costing method [10] calculates the acceptable project costs based on market prices and desired profit margins, then works backward to ensure budget compliance, making it well-suited for competitive market environments. The WBS method [11] decomposes a project into the smaller and manageable components, facilitating detailed cost estimation for each segment. This approach enhances cost control by enabling continuous monitoring of expenditures across work packages. Lastly, the cost-benefit analysis method [12] evaluates the total costs of project against its expected benefits, enabling the decision-makers to assess its feasibility and economic viability. By considering both financial and qualitative outcomes, this method guides investment decisions and ensures resource optimization.

Although the Delphi method, value engineering, target costing, WBS, and cost-benefit analysis are valuable tools for construction cost prediction, they have notable limitations. These methods rely heavily on the availability and accuracy of data, making them vulnerable to errors in projects where historical data is scarce or unreliable, particularly for unique or innovative initiatives. Additionally, methods like the Delphi approach depend on expert judgment, which can introduce subjective biases and lead to inconsistent estimates, undermining reliability [13]. The complexity and time-intensive nature of approaches such as value engineering and WBS further reduce their practicality in fast-paced project environments that demand quick decision-making [14]. Moreover, these traditional methods often fail to adequately account for the dynamic risks and uncertainties inherent in modern construction projects, resulting in potential underestimations or inaccuracies in cost forecasts. These limitations highlight concerns about their effectiveness and adaptability in the rapidly evolving construction landscape. Given these shortcomings, there is growing interest in exploring alternative paradigms, such as machine learning, that can autonomously learn patterns from complex datasets and adapt to changing conditions. In this study, however, we do not attempt to merge traditional and modern approaches into a mixed-method strategy. Instead, we examine machine learning-based methods independently to better highlight their unique capabilities, allowing a clearer evaluation of how they address the limitations of conventional techniques.

With the advent of Industrial Revolution 4.0 [15], transformative technologies such as the Internet of Things (IoT), Big Data, and Artificial Intelligence (AI) are reshaping various aspects of society. Among these, machine learning, i.e., a critical subset of AI, has found extensive applications across diverse domains, from enhancing daily life to advancing complex scientific research. In everyday life, major e-commerce platforms, music services, and video streaming sites leverage machine learning models to personalize user experiences. By analyzing the historical behaviors and preferences of users, these machine learning models deliver tailored product, music, and video recommendations for improving the user satisfaction, while boosting company revenues. For instance, news recommendation systems and article classification engines have employed machine learning to handle large-scale imbalanced datasets and improve performance in text categorization [16]. In healthcare, machine learning models facilitate accurate disease diagnosis and tumor detection by analyzing medical images to identify subtle abnormalities, enabling faster and more precise clinical decisions. Additionally, machine learning can leverage genomic and medical history data to design personalized treatment plans, optimizing patient outcomes [17]. In agriculture, machine learning powers autonomous robots to perform tasks such as sowing, fertilizing, and weeding with precision, enhancing productivity while promoting resource efficiency and sustainable practices [18]. Beyond these areas, machine learning also plays pivotal roles in other sectors such as automation [19], non-destructive testing [20,21], modelling [22,23], predictive maintenance [24–27].

The rapid advancement of machine learning is also revolutionizing the construction industry, offering innovative solutions in design, construction, management, and beyond. Machine learning holds immense potential to enhance efficiency, quality, and sustainability across various construction processes. In optimal structural design, machine learning assists engineers in identifying solutions that meet performance requirements by analyzing historical data and models. This not only optimizes designs but also provides real-time feedback to refine decision-making [28]. Machine learning also plays a pivotal role in structural health monitoring [29], enabling early detection of damage, preventing catastrophic failures, and predicting the remaining service life of structures. These insights form the basis for effective maintenance and reinforcement strategies [30]. Beyond structural analysis, machine learning significantly impacts building materials research by predicting key properties such as strength and elasticity from material composition and preparation processes [31]. This capability aids engineers in selecting materials that ensure superior performance under various loads. Furthermore, machine learning predicts material durability under diverse environmental conditions, fostering the development of innovative, high-performance materials. In construction material recycling, the combination of machine learning and imagine recognition technology enables the efficient classification of waste materials, promoting sustainability in the construction industry [32]. Intelligent recycling systems powered by machine learning can enhance the waste processing efficiency, reduce resource wastage, and contribute in achieving the circular economy goals.

Machine learning has demonstrated remarkable potential for

processing large and complex datasets, making it particularly promising for construction cost prediction. By examining extensive historical project data, machine learning can deliver refined predictive models that serve as a scientific foundation for cost management [33,34]. However, several critical gaps persist, hindering the widespread adoption and effectiveness of machine learning in this domain:

- Firstly, most existing studies focus on a narrow range of machine learning models, i.e., often relying on standard approaches such as Linear Regression, Decision Trees, Random Forests or Neural Networks. This restricted scope may overlook the capabilities of more specialized or modern machine learning models that can better capture the complex and non-linear relationships that often present in construction cost data. Consequently, the potential of a broader suite of state-of-art machine learning models remains insufficiently explored.
- Secondly, while metrics such as Mean Square Error (MSE) and coefficient of determinations (R^2) are frequently employed to evaluate the predictive performance of machine learning models, they do not provide insight into the reliability or uncertainty of cost predictions. Construction projects involve high-stakes decisions, where budget overruns can lead to substantial financial and reputational damage. Thus, incorporating confidence intervals is critical for decision-makers to assess the level of certainty in each prediction and manage risks more effectively.
- Thirdly, the "black-box" nature of many advanced machine learning models (e.g., ensemble methods and deep neural networks) discourages practitioners from fully embracing these techniques [35]. Construction professionals that often more familiar with traditional cost estimation methods may be reluctant to trust the predictions generated by machine learning models without an interpretable rationale. This research gap emphasizes the needs for Explainable AI (XAI) methodologies that illuminate how machine learning models arrive at specific predictions, thereby enhancing the transparency and confidence of users.
- Finally, most existing research typically addresses the aforementioned three challenges (i.e., model selection, confidence level of prediction and model interpretability) in isolation. Rarely do studies concurrently implement a diverse portfolio of machine learning modes, quantify predictive uncertainties and provide transparent explanations of predictive results. The absence of this holistic approach constrains the potential impact of machine learning on the construction cost industry, limiting both the precision and trustworthiness of cost predictions.

The abovementioned deficiencies highlight an urgent need for a more comprehensive research framework that systematically compares multiple machine learning models in construction cost prediction, employs confidence interval analysis for reliability, and utilizes XAI techniques like SHAP to ensure transparency. Addressing these gaps is not just of academic value, it holds significant practical implications for a wide range of stakeholders, including construction firms, project owners, contractors, and government agencies. In real-world scenarios, especially in large-scale infrastructure or public housing projects, inaccurate or opaque cost estimates can result in severe budget overruns, delays, and reputational risks. By offering more transparent, robust, and adaptable prediction tools, this study supports better decision-making in high-stakes environments and contributes to the broader goals of cost efficiency, sustainability, and public accountability.

To address this challenge, the current study provides a comprehensive evaluation of ten advanced machine learning models, where many of which have yet to be thoroughly explored in the context of construction cost prediction, using a standardized dataset from RSMeans. By systematically comparing a broad spectrum of machine learning algorithms under the uniform conditions, this research aims to deliver deer insights into each model's predictive capabilities, reliability and interpretability. The technical contributions and novelties of this paper are explained as follows:

- A rigorous and side-by-side comparison of ten machine learning models, i.e., Ridge Regression, Lasso Regression, Elastic Net, K-Nearest Neighbor (KNN) Regression, Extremely Randomize Trees (Extra Trees) Regression, Gradient Boosting Regression, Adaptive Boosting (AdaBoost) Regression, Extreme Gradient Boosting (XGBoost) Regression, Categorical Boosting (CatBoost) and Histogram-based Gradient Boosting (HistGradientBoosting) Regression, is conducted on the same construction cost dataset. This extensive scope not only evaluates conventional methods (e.g., Ridge, Lasso) but also explore powerful ensemble and boosting approaches (e.g., XGBoost, CatBoost, HistGradientBoosting), thereby uncovering the strengths and limitations of each model.
- In contrast to most existing studies that rely solely on standard performance metrics (e.g., MSE and R^2), the present work incorporates confidence interval analysis to quantify the uncertainty associated with each prediction. This additional layer of information is critical for real-world construction cost decision-making, where the risk and reliability are as important as raw accuracy. By providing a statistical measure of model uncertainty, project managers and stakeholders can better gauge how much confidence to place in the forecasts.
- Recognizing the "black-box" concerns surrounding many advanced machine learning models, this study employs the SHapley Additive exPlanations (SHAP) technique to elucidate how input features influence the cost prediction made by each machine learning model. Such transparency is essential for fostering trust among practitioners, who often require clear and justifiable insights before integrating data-driven tools into critical budgeting processes. The explainability framework in this study thus bridges the gap between high predictive accuracy and practical usability.
- By combining a wide range of machine learning models, confidence interval estimation, and SHAP-based explainability, this study introduces a holistic evaluation framework for construction cost prediction. Unlike traditional benchmarking, the proposed pipeline jointly assesses predictive accuracy, uncertainty quantification, and interpretability, thus providing a more practical and trustworthy tool for decision-makers. This integrated approach delivers deeper insights into model behavior, fosters transparency, and enhances confidence in adopting machine learning for high-stakes construction budgeting scenarios.

The organization of this paper is as follows: Section 2 provides a comprehensive review of related work. Section 3 outlines the methodologies of the ten machine learning models used, including descriptions of the data sources, model training processes, and performance evaluation metrics. Section 4 presents the performance evaluation and analysis of the selected machine learning models in solving the construction cost prediction problem. Finally, Section 5 summarizes the key findings and discusses potential directions for future research.

2. Literature review

This section provides a comprehensive review of existing research in the field of construction cost forecasting, organized into three main thematic areas: (a) traditional statistical models, (b) single machine learning models, and (c) hybrid systems and ensemble strategies. The literature reveals a clear methodological progression, from classical regression-based techniques to advanced data-driven approaches, driven by the growing complexity of construction projects and the increasing availability of construction-related data. Recent developments in machine learning and ensemble modeling have led to notable improvements in prediction accuracy, model adaptability, and the ability to capture nonlinear relationships among cost-influencing factors.

2.1. Traditional statistical and regression-based models

Traditional statistical models, particularly regression-based techniques, have long served as foundational tools in construction cost estimation. These models typically rely on linear assumptions and structured variable relationships to generate cost predictions. While widely adopted due to their interpretability and simplicity, they often fall short in addressing the inherent complexity and nonlinearity embedded in real-world construction data.

Numerous studies have employed Linear Regression approaches to identify cost drivers and establish predictive relationships. For example, Yang et al. [36], Hai [37], and Lowe et al. [38] applied Multiple Linear Regression (MLR) and Stepwise Regression techniques across various construction datasets. Yang et al. [36] developed a BIM-integrated regression model that improved floor area estimation accuracy compared to traditional methods. Hai [37] utilized SPSS for weighted analysis of 16 influencing variables, ultimately narrowing down to four key factors while achieving a maximum budget deviation of 4.80 %. Similarly, Lowe et al. [38] analyzed 286 UK-based project datasets to identify significant cost drivers, such as gross internal floor area, function, and mechanical installation, highlighting the practical utility of regression in early-stage cost estimation. Despite their utility, traditional linear models often fall short when modeling complex, multivariate relationships. Lowe et al. [38] reported a Mean Absolute Percentage Error (MAPE) of approximately 25 %, illustrating their limited ability to fully capture intricate cost dynamics. Jafarzadeh [39] further emphasized that stepwise regression, while effective in identifying key variables, is highly sensitive to sample quality and lacks transferability across project types.

To overcome these limitations, researchers have turned to more flexible statistical methods. Petroutsatou et al. [40] applied Structural Equation Modeling (SEM) to tunnel construction cost estimation, showing superior accuracy and the ability to model latent interdependencies compared to standard regression and neural networks. Shahandashti and Ashuri [41] introduced a Vector Error Correction (VEC) model using 16 macroeconomic indicators to forecast the National Highway Construction Cost Index (NHCCI). Their model effectively captured long-term economic trends and temporal dependencies, outperforming univariate regression baselines. In a complementary direction, Zhang et al. [42] demonstrated the advantages of LASSO regression over Ordinary Least Squares (OLS), particularly in managing high-dimensional data and preventing overfitting through regularization.

While regression-based models continue to offer valuable insights in structured environments, their predictive performance degrades in high-dimensional, nonlinear, or volatile economic contexts. Methods like SEM, VEC, and LASSO represent important advancements, but they remain constrained by assumptions of linearity, static variable interactions, or lack of adaptability to shifting data distributions. These limitations underscore the growing need for more flexible, data-driven approaches such as machine learning, which are better suited to model heterogeneity and capture complex cost relationships in modern construction projects.

2.2. Single machine learning model innovations

The adoption of single machine learning models has gained substantial traction in construction cost estimation, primarily due to their ability to capture nonlinear dependencies and complex variable interactions, i.e., areas where traditional regression models often fall short. These models, ranging from neural networks to kernel-based methods, have demonstrated notable improvements in accuracy and flexibility across diverse project types.

Among the various ML approaches, Artificial Neural Networks (ANN) have consistently emerged as a top performer. Emsley et al. [43] assessed ANN performance across over 300 construction projects, reporting a MAPE of 16.6 %, significantly outperforming regression-based techniques which exhibited error rates between 20.8 % and 27.9 %. Cheng et al. [44] extended ANN capabilities by introducing the Evolutionary Fuzzy Neural Inference Model (EFNIM), integrating Fuzzy Logic and Genetic Algorithms to refine parameter optimization. This model was particularly effective in early-stage conceptual cost estimation, adding robustness to initial planning decisions.

Support Vector Machines (SVMs) offer strong generalization capabilities, especially in high-dimensional settings. Petruseva et al. [45] showed that SVMs outperformed linear regression in predictive accuracy, while Kim [46] found that ANNs maintained superiority even over SVMs in school construction contexts. To push accuracy further, Du et al. [47] implemented a Genetic Algorithm-optimized Backpropagation Neural Network (GA-BPNN), which improved coefficient calibration and yielded better investment forecasts. Beyond traditional ML architectures, Case-Based Reasoning (CBR) has also gained attention. Ahn et al. [48] proposed a normalized CBR framework, validated using Leave-One-Out Cross Validation (LOOCV) and Kernel Density Estimation (KDE). Xiao et al. [49] improved upon this by incorporating Modal Linear Regression (MODLR), which preserved knowledge stability and better handled dataset fluctuations, i.e., key concerns in dynamic construction environments.

Simić et al. [50] provided a noteworthy benchmark by comparing MRA, ANN, and XGBoost models. Their results showed that accurate cost predictions could be achieved with only three owner-based and five contractor-based cost drivers, suggesting that parsimony in input selection does not compromise predictive performance. Stakeholder analysis also revealed diverging priorities: environmental concerns dominated among owners, whereas contractors emphasized supply chain risks, especially those triggered by global disruptions. Yun [51] introduced a multi-output ANN model capable of estimating seven sub-construction cost components concurrently, enabling more granular forecasts tailored to specific cost categories. To improve model transparency, Wang et al. [52] combined Deep Neural Networks with SHAP explainability techniques to analyze 98 public school projects in Hong Kong. Their analysis revealed that economic indicators had a stronger impact on reducing prediction error than engineering factors. This integration of XAI tools significantly increased model interpretability, helping stakeholders understand the rationale behind cost estimates.

Despite their strengths, single machine learning models are not without limitations. While ANNs and SVMs excel in capturing complex patterns, their predictive power can degrade in the presence of noisy or sparse datasets. Moreover, most models reviewed remain isolated in design, optimizing within a single architecture rather than leveraging cross-model synergies. Additionally, many studies lack uncertainty quantification or confidence interval reporting, reducing their applicability in risk-sensitive decision-making environments. These limitations set the stage for hybrid and ensemble methods, which combine complementary algorithms, optimize parameter tuning, and integrate domain-specific knowledge to enhance accuracy, robustness, and explainability.

2.3. Hybrid systems and ensemble strategies

Hybrid systems and ensemble strategies have emerged as robust solutions for construction cost estimation, addressing the limitations of single-model approaches through model integration. These strategies aim to capture a broader spectrum of data relationships by combining diverse machine learning algorithms, each with complementary strengths, into unified predictive frameworks.

Among commonly used base learners, Random Forest (RF), XGBoost, and Light Gradient Boosting Machine (LightGBM) stand out for their ability to model nonlinear relationships, maintain computational efficiency, and scale to large datasets. Huang and Hsieh [53] combined RF with Linear Regression, achieving superior labor cost predictions in BIM environments by leveraging RF's nonlinear pattern recognition and LR's

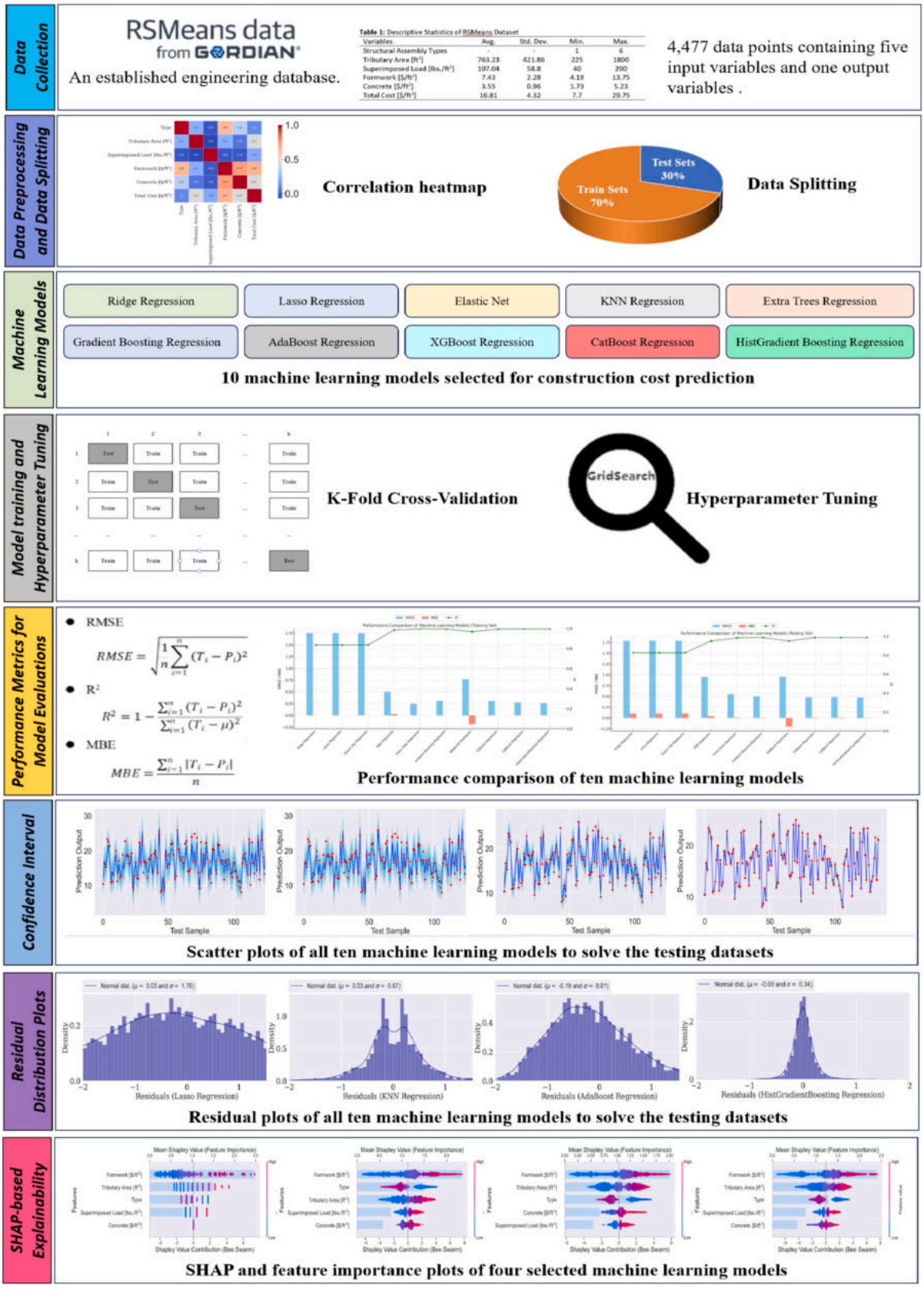


Fig. 1. Workflow of the proposed methodology for construction cost prediction.

strength in modeling linear trends. Similarly, Alshboul et al. [54] compared XGBoost, Deep Neural Networks, and RF in green building cost estimation, with XGBoost outperforming others with a prediction accuracy of 0.96, highlighting its effectiveness in high-dimensional settings. Shehadeh et al. [55] extended this line of inquiry to equipment asset valuation, using Modified Decision Trees (MDT), LightGBM, and XGBoost to predict the residual value of construction machinery. Their findings, validated via MAE, MSE, MAPE, and R^2, further confirmed the versatility of ensemble methods in automation tasks.

Hybrid models are increasingly coupled with metaheuristic optimization techniques to further enhance predictive performance. Kim et al. [56] applied Genetic Algorithms to optimize ANN hyperparameters, outperforming manual tuning and accelerating convergence. Cheng et al. [57] developed the ELSVM system by integrating Least Squares Support Vector Machines (LS-SVM) with Differential Evolution, achieving a MAPE below 1 % in modeling fluctuations in the Construction Cost Index (CCI). These examples underscore the importance of automated optimization for improving both accuracy and generalizability in construction forecasting pipelines.

In addition to algorithmic improvements, domain-specific knowledge has been strategically incorporated to increase relevance and interpretability. Alshboul et al. [58] and Ali et al. [59] integrated both soft and hard cost variables related to green building projects into ensemble models using XGBoost and LightGBM. Their studies emphasized the differential impact of public versus private investment on cost behavior, i.e., an often-overlooked factor in conventional models. Another key innovation in recent hybrid models is the inclusion of uncertainty quantification mechanisms. Cheng and Hoang [60] introduced the EAC-LSPIM model, which provides not only point estimates but also interval predictions, delivering confidence bounds essential for informed project planning and risk mitigation. This approach aligns more closely with decision-making practices in real-world construction management, where prediction reliability is just as critical as accuracy.

Across these studies, XGBoost consistently emerges as a top-tier performer, demonstrating superior adaptability across application domains, from green construction and labor cost estimation to equipment valuation. Yet, while these hybrid systems offer considerable gains in performance, challenges remain. Many models prioritize accuracy without systematically addressing interpretability, limiting their adoption by practitioners. Furthermore, few models offer mechanisms to dynamically adjust to evolving project conditions or data drift over time. These gaps highlight the need for frameworks that not only integrate multiple learning paradigms but also incorporate explainability and uncertainty handling as first-class design elements. The next section builds on these insights by proposing a novel hybrid framework that combines predictive strength with interpretive clarity and risk-aware outputs, positioning it as a comprehensive tool for modern construction cost forecasting.

3. Proposed methodology

The proposed methodology follows a systematic workflow illustrated in Fig. 1 to evaluate and compare the performance of ten advanced machine learning models for construction cost prediction. The process begins with data collection and preprocessing, where a standardized dataset is prepared through exploratory data analysis and feature engineering to ensure data quality and suitability for modeling. The dataset is then split into training and test sets for model validation. Next, a diverse set of machine learning models, namely Ridge Regression, Lasso Regression, Elastic Net, KNN Regression, Extra Trees Regression, Gradient Boosting Regression, AdaBoost Regression, XGBoost Regression, CatBoost and HistGradientBoosting Regression, are employed. Each machine model undergoes rigorous hyperparameter tuning using GridSearch to optimize its performance.

The performances of all selected machine learning models for construction cost predictions are evaluated using standard metrics such as R^2, RMSE and MBE, while confidence intervals are incorporated to enhance the reliability of predictions. The methodology also includes residual plot analysis to assess model fit and identify potential biases in predictions. Additionally, SHAP is utilized to interpret the behavior of the machine learning models, providing insights into the contribution of input features to prediction outcomes. This holistic evaluation framework goes beyond model benchmarking by incorporating uncertainty quantification via confidence intervals and interpretability via SHAP, offering a more robust and actionable tool for decision-making in construction cost prediction. By integrating accuracy, transparency, and risk-awareness, the proposed methodology bridges the gap between advanced analytics and real-world construction management.

Table 1
Descriptive Statistics of RSMeans Dataset.

Variables	Avg.	Std. Dev.	Min.	Max.
Structural Assembly Types	–	–	1	6
Tributary Area [ft^2]	763.23	421.86	225	1800
Superimposed Load [lbs./ft^2]	107.04	58.8	40	200
Formwork [$/ft^2]	7.43	2.28	4.19	13.75
Concrete [$/ft^2]	3.55	0.96	1.73	5.23
Total Cost [$/ft^2]	16.81	4.32	7.7	29.75

3.1. Data collection

The proposed methodology utilized a dataset compiled from the RSMeans Assemblies Books published between 1998 and 2018. This dataset served as the foundation for training, validating, and testing the ten selection machine learning models. The dataset includes one dependent variable, i.e., the total construction cost of structural assemblies (measured in $/ft^2), and four independent variables, namely structural assembly type, tributary area (ft^2), superimposed load (lbs./ft^2), unit cost of formwork ($/ft^2), and unit cost of concrete ($/ft^3). The year of the cost estimate was excluded as an independent variable, as data from different years were used solely to account for variations in unit costs of structural components.

RSMeans data is widely recognized as a trusted standard for cost estimation in the U.S. construction industry. Originally developed by Robert Snow Means in the 1940 s, RSMeans has become an essential resource for cost engineers and industry professionals. Maintained by Gordian since 2000, the RSMeans database now includes over 85,000-line items, with more than 22,000 h dedicated annually to cost research and validation. This extensive dataset is available in multiple formats, including printed books, CDs, and dynamic web-based estimating tools, making it the largest and most comprehensive cost database in the world. Its credibility and meticulous documentation make it ideal for construction cost prediction studies.

The dataset used in this study comprises 4,477 data points related to structural floor assemblies in medium-sized and high-rise buildings. The dataset covers various structural assembly types, including one-way and two-way slabs, flat slabs with or without drop panels, multi-span joist slabs, and waffle slabs. Independent variables were sourced from the RSMeans Unit Cost Data Book, while the dependent variable (total cost of the assembly) was obtained from the RSMeans Assemblies Book. Descriptive statistics for the dataset are summarized in Table 1, presenting the average (Avg.), standard deviation (Std. Dev.), minimum (Min.), and maximum (Max.) values for each variable. The structural assembly type, a categorical variable ranging from 1 to 6, represents different slab types, including one-way slabs, two-way slabs, flat plates with or without drop panels, multi-span joist slabs, and waffle slabs.

3.2. Data preprocessing and data splitting

Data preprocessing and data splitting are essential stages in machine learning workflows, as they significantly influence the accuracy and

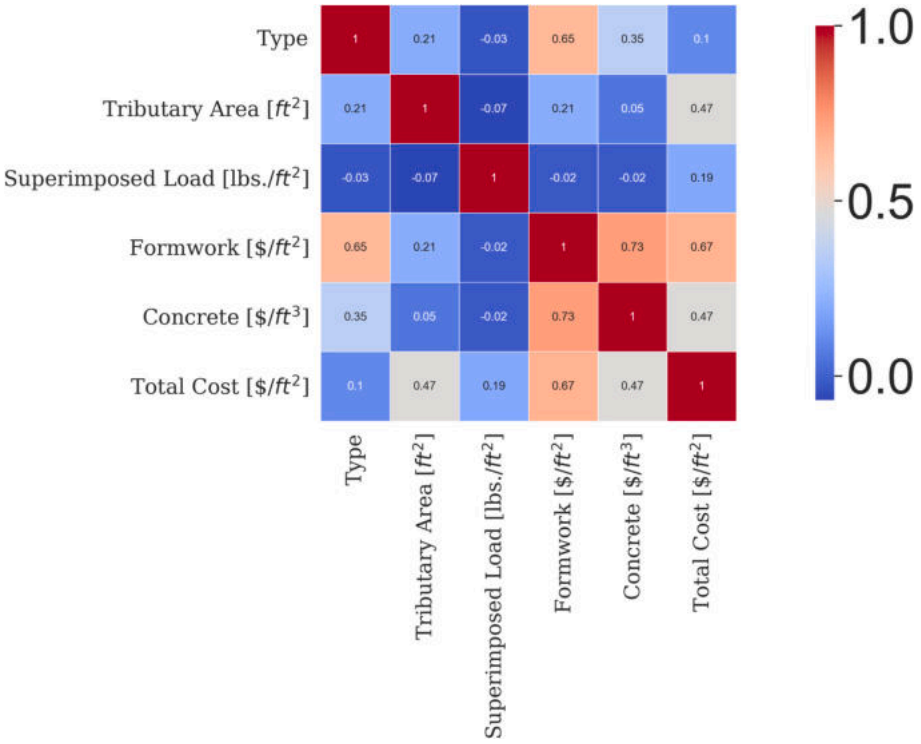


Fig. 2. Correlation heatmap showing the relationships between features and total construction cost ($/ft^2).

performance of predictive models. These steps ensure that the input data is of high quality, properly scaled, and prepared for training and testing.

Data preprocessing involves cleaning and transforming raw data into a format suitable for modeling. In this study, Pearson correlation coefficients were calculated to examine the linear relationships among features in the dataset, as visualized in the correlation heatmap in Fig. 2. The heatmap highlights the degree of correlation between independent variables and the target variable, Total Cost. Features such as Formwork ($/ft^2) and Concrete ($/ft^3) exhibit a strong positive correlation with Total Cost, suggesting their significant influence on the predictive model. Conversely, variables like Superimposed Load (lbs./ft^2) and Tributary Area (ft^2) demonstrate weaker correlations, providing insights into feature selection and model interpretation.

The dataset defines $\mathbf{x} = \{x_1, x_2, \cdots, x_i, \cdots, x_N\}$ as the feature matrix, containing independent variables, and $\mathbf{y} = \{y_1, y_2, \cdots, y_i, \cdots, y_N\}$ as the target variable, representing the Total Cost ($/ft^2). To further enhance the stability and efficiency of the models, feature scaling was applied. Scaling is particularly important for algorithms sensitive to the magnitude of features, such as KNN, where differences in scales can introduce bias. In this study, the *MinMaxScaler* function was employed to normalize the feature values of both the training and testing datasets, scaling them to a range between 0 and 1. This normalization process not only ensures balanced feature contributions but also accelerates convergence during model training, ultimately improving predictive performance.

Data partitioning is a critical step for evaluating the generalization capability of machine learning model on unseen data. The *train_test_split* function was utilized to divide the dataset into training and testing subsets. A *test_size* of 0.3 was specified, allocating 30 % of the data for testing and the remaining 70 % for training. Additionally, the parameter *shuffle* was set as "True" to ensure that the data was randomized before splitting.

3.3. Machine learning models

3.3.1. Ridge Regression

Ridge Regression [61] is a regularization technique designed to address multicollinearity, a prevalent issue in construction cost prediction where multiple input features are often highly correlated. Multicollinearity can lead to instability in the coefficient estimates of Ordinary Least Squares (OLS) regression, where small changes in the input data can result in large variations in the estimated coefficients. Ridge Regression mitigates this issue by introducing a regularization term to the OLS cost function, effectively penalizing the magnitude of the regression coefficients and reducing their variance.

Ridge Regression enhances model stability by adding a penalty term controlled by a regularization parameter (α), a positive scalar that determines the degree of shrinkage applied to the coefficients. The Ridge Regression cost function is defined as follows:

$$\widehat{\beta}_{ridge} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{P} \beta_j x_{ij} \right)^2 + \alpha \sum_{j=1}^{P} \beta_j^2 \right\} \quad (1)$$

where N denotes the total number of data points; P is the total number of features used for construction cost prediction; $i = 1, \cdots, N$ and $j = 1, \cdots, P$ refer to the indices of data points and features, respectively; y_i represents the actual construction cost of the i-th data sample; x_{ij} represents the j-th feature of i-th data sample; β_j are the regression coefficients; α is the regularization parameter that controls the penalty strengths.

The regularization term $\alpha \sum_{j=1}^{P} \beta_j^2$ shrinks the magnitude of the coefficients, reducing their sensitivity to variations in the input data. This shrinkage improves the generalization capability of model, making it particularly effective in handling noisy data or predicting costs for projects outside the training data range. The ridge parameter α plays a critical role in balancing the trade-off between bias and variance. A larger α increases the bias by applying stronger regularization but reduces the variance, resulting in a more robust model in complex or uncertain environments. Conversely, smaller values of α reduce the

penalty, allowing the model to better fit the training data but increasing the risk of overfitting. In practical applications, cross-validation techniques are commonly used to determine the optimal value of α that achieves the best predictive performance.

3.3.2. Lasso Regression

Lasso Regression [62], an abbreviation for Least Absolute Shrinkage and Selection Operator, is a regularization technique that extends OLS regression by incorporating an L_1 penalty term into the objective function of model. This penalty not only reduces overfitting by controlling model complexity but also facilitates automatic feature selection by shrinking some regression coefficients to exactly zero. This unique characteristic makes Lasso particularly valuable in scenarios involving a large number of features, where some predictors may be irrelevant or redundant.

The Lasso Regression cost function modifies the OLS loss function by adding the L_1 regularization term, which is defined as the sum of the absolute values of the coefficients:

$$\widehat{\beta}_{lasso} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{P} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{P} |\beta_j| \right\} \tag{2}$$

where λ is the regularization parameter that controls the strength of the L_1 penalty.

One of the defining characteristics of Lasso Regression is its ability to shrink some coefficients to exactly zero due to the nature of the L_1 penalty. This attribute effectively removes irrelevant or redundant features from the model, performing automatic feature selection during training. This capability is especially beneficial in complex prediction problems, where a wide range of potential predictors may exhibit high correlation or limited relevance. By adjusting the regularization parameter λ, Lasso Regression achieves a balance between bias and variance. Larger values of λ increase the penalty, resulting in higher bias and fewer features being retained, which can help improve generalization in noisy datasets or high-dimensional settings. Conversely, smaller values of λ allow the model to retain more features, reducing bias but increasing the risk of overfitting.

3.3.3. Elastic Net Regression

Elastic Net Regression [63] is a regularization technique that combines the strengths of Ridge Regression and Lasso Regression, making it particularly effective for handling high-dimensional data with strongly correlated features or a large number of redundant predictors. Such scenarios are frequently encountered in construction cost prediction, where predictors like material costs, labor costs, and project specifications often exhibit high intercorrelation.

Elastic Net extends the OLS loss function by incorporating both L_1 (Lasso) and L_2 (Ridge) regularization terms. Its objective function is defined as follows:

$$\widehat{\beta}_{elastic\_net} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \alpha_{overall} \left[\rho \sum_{j=1}^{p} |\beta_j| + \frac{(1+\rho)}{2} \sum_{j=1}^{p} \beta_j^2 \right] \right\} \tag{3}$$

where $\alpha_{overall}$ is an overall regularization strength parameter; ρ is a mixing parameter controlling the balance between L_1 and L_2 penalties, where $\rho = 1$ reduces Elastic Net to Lasso, and $\rho = 0$ reduces it to Ridge Regression.

Elastic Net's unique combination of L_1 and L_2 penalties addresses the limitations inherent in both Lasso Regression and Ridge Regression. The L_1 term enables feature selection by shrinking some coefficients to exactly zero, while the L_2 term ensures that highly correlated predictors are grouped together rather than arbitrarily excluded from the model. The parameter ρ allows practitioners to balance the trade-off between model interpretability and complexity. By adjusting $\alpha_{overall}$ and ρ, Elastic Net can capture the most relevant features while maintaining model stability and reducing the risk of overfitting. This flexibility makes it well-suited for datasets with numerous predictors, noisy observations, or strong multicollinearity.

3.3.4. KNN Regression

KNN Regression [64] is a non-parametric machine learning algorithm widely used for predicting continuous variables, such as construction costs. Unlike parametric models, KNN does not assume any predefined functional relationship between input features and the target variable. Instead, it bases predictions on the similarity between data points, making it particularly suitable for datasets with complex or unknown relationships among variables.

In KNN Regression, the predicted value for a new data point $\mathbf{x}$, denoted as y_{pred}, is computed as the average of the target values of the k nearest neighbors in the training set. The proximity between data points is typically determined using a distance metric, with Euclidean distance being one of the most commonly used. The prediction formula is as follows:

$$y_{pred} = \frac{1}{k} \sum_{a \in N_k(\mathbf{x})} y_a \tag{4}$$

where y_{pred} is the predicted value (e.g., construction cost) for the new data point $\mathbf{x}$; $N_k(\mathbf{x})$ is the set of the k nearest neighbors to $\mathbf{x}$, determined based on the chosen distance metrics; y_a is the target value (e.g., construction cost) of the a-th nearest neighbor in the training set; k is the number of neighbors considered in the prediction.

The parameter k plays a crucial role in balancing the trade-off between bias and variance. A smaller k focuses on fewer neighbors, resulting in low bias but high variance, as predictions are heavily influenced by local noise. Conversely, a larger k averages over more neighbors, reducing variance but potentially increasing bias by smoothing over finer details in the data. Selecting an appropriate value for k is therefore critical and is often determined through cross-validation.

In the context of construction cost prediction, KNN Regression is particularly useful for leveraging historical data to estimate the costs of new projects. By basing the prediction for a new project on the costs of similar past projects, KNN effectively captures local patterns and relationships within the data. This is especially advantageous in datasets encompassing diverse construction projects with varying characteristics, such as project scale, location, and material requirements. While KNN Regression offers simplicity and adaptability, it is sensitive to noisy data and computationally intensive, especially for large datasets or high-dimensional feature spaces. Consequently, it is most effective when applied to clean, well-curated datasets or when ample computational resources are available.

3.3.5. Extra Trees Regression

Extra Trees Regression [65] is a tree-based ensemble learning method that extends the principles of Random Forests by introducing additional randomness during tree construction. Unlike traditional Decision Trees or Random Forests, Extra Trees selects thresholds for node splitting randomly, rather than optimizing splits based on criteria such as Gini impurity or mean squared error. This added randomness makes Extra Trees faster to train and more robust against overfitting, trading a slight increase in bias for a significant reduction in variance, resulting in better generalization to unseen data.

Given a training dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \cdots, (x_N, y_N)\}$, Extra Trees constructs an ensemble of T randomized decision tress. Each tree is trained on the full dataset but splits nodes using randomly selected features and thresholds. For a given input $\mathbf{x}$, the

prediction of an individual tree, $f_t(\mathbf{x})$ is computed as the mean of the target value y_i in the leaf node $\mathscr{L}$ where $\mathbf{x}$ falls:

$$f_t(\mathbf{x}) = \frac{1}{|\mathscr{L}|} \sum_{(x_i, y_i) \in \mathscr{L}} y_i \tag{5}$$

The final prediction $\hat{y}_i$ for an input x_i is obtained by averaging the predictions of all T trees in the ensemble as:

$$\hat{y}_i = \frac{1}{T} \sum_{t=1}^{T} f_t(x_i) \tag{6}$$

where $f_t(x_i)$ is the prediction of the t-th tree for the input x_i; T is the total number of trees in the ensemble. Each tree is constructed by selecting a random subset of features at each node and splitting the data using a randomly chosen threshold for the selected feature. This process significantly reduces computational cost compared to Random Forests, where finding the optimal split requires evaluating all possible thresholds.

Extra Trees Regression offers several key advantages when it is used to tackle challenging prediction problems. By using random thresholds instead of optimal ones, Extra Trees significantly reduces training time, especially for high-dimensional datasets. The additional randomness in feature selection and thresholding increases bias slightly but reduces variance, resulting in better generalization of Extra Trees Regression to unseen data. Unlike Random Forests, Extra Trees does not rely on bootstrapped samples; all trees are trained on the full dataset, further simplifying computation. The hyperparameters of Extra Trees, such as the number of trees T, maximum tree depth, and the number of features considered for splitting, allow for fine-tuning to achieve optimal performance. Increasing T generally improves prediction stability but comes with a trade-off in computational cost. Despite its strengths, Extra Trees may struggle in scenarios where interpretability is critical, as the randomized nature of the model makes it less intuitive than simpler regression methods. However, its speed and generalization capability make it a strong choice for large-scale construction datasets.

3.3.6. Gradient Boosting Regression

Gradient Boosting Regression [66] is an ensemble learning technique that sequentially combines multiple weak learners, typically decision trees, to build a strong predictive model. Its core idea is to iteratively minimize a specified loss function by fitting the residuals of the current model with new base learners, progressively reducing prediction errors. This algorithm is widely used in complex regression tasks, due to its flexibility, robustness, and ability to handle both categorical and numerical data.

The Gradient Boosting Regression algorithm begins with an initial model, often set as the mean of the target values in the training set:

$$F_0(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{N} y_i \tag{7}$$

where N is the total number of training samples and y_i is the target value of the i-th sample. The algorithm iteratively refines its initial models by adding weak learners $h_m(\mathbf{x})$, which are fitted to approximate the residuals at each step. At each m-th iteration, the residuals are calculated as the negative gradient of the loss function $L(y, F(\mathbf{x}))$ with respect to the current model's predictions:

$$r_{i,m} = -\frac{\delta L(y_i, F_{m-1}(x_i))}{\delta F_{m-1}(x_i)} \tag{8}$$

A new weak learner $h_m(\mathbf{x})$ is then fitted to these residuals by minimizing the loss:

$$h_m(\mathbf{x}) = \underset{h}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \nu h(x_i)) \tag{9}$$

where x_i is the feature vector for the i-th data sample; y_i is the target value for the i-th data sample; ν is the learning rate, a hyperparameter that controls the contribution of each weak learner; $h_m(\mathbf{x})$ is the prediction of the m-th weak learner, which is constructed to fit the current residuals. The model is updated iteratively by adding the scaled predictions to the weak learner to the current model:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu h_m(\mathbf{x}) \tag{10}$$

where $F_m(\mathbf{x})$ is the model prediction after m iterations. This iterative process continues until the model converges or reaches the specified number of iterations M, where M is the total number of weak learners (e.g., decision trees).

Gradient Boosting Regression offers several advantages, including flexibility in optimizing for various loss functions (e.g., MSE, MAE), support for both categorical and numerical data, and robustness to missing values. The algorithm does not require explicit imputation for missing data, as it can handle missing values inherently during tree construction. The learning rate ν is a critical hyperparameter that balances the trade-off between model convergence and overfitting. A smaller ν leads to more gradual learning and requires more iterations but often results in a more robust model. Other hyperparameters, such as the number of trees M and tree depth, allow fine-tuning of the model to achieve optimal performance.

In the context of construction cost prediction, Gradient Boosting Regression is particularly advantageous due to its ability to model complex, non-linear relationships between input features and target variables. By leveraging its iterative approach, this algorithm effectively learns from residual errors, ensuring high accuracy and robustness even in datasets with intricate relationships. Despite its strengths, Gradient Boosting Regression can be computationally intensive, especially for large datasets or deep trees. Additionally, improper tuning of hyperparameters may lead to overfitting or slow convergence.

3.3.7. AdaBoost Regression

AdaBoost Regression [67] is another powerful ensemble learning algorithm that iteratively combines multiple weak learners to create a strong predictive model. While initially designed for classification tasks, AdaBoost has been successfully adapted for regression problems. Its core idea is to iteratively adjust the weights of data points, giving greater emphasis to samples that were falsely predicted in previous iterations. This adaptive weight adjustment ensures that subsequent weak learners focus on the challenging aspects of the data, progressively reducing the overall prediction error.

The AdaBoost algorithm starts by assigning equal weights to all data points in the training set. At each m-th iteration, a weak learner $h_m(\mathbf{x})$ is trained on the weighted dataset. The prediction error E_m of the weak learner is calculated as:

$$E_m = \frac{\sum_{i=1}^{N} w_i \mathbb{I}(y_i \neq h_m(x_i))}{\sum_{i=1}^{N} w_i} \tag{11}$$

where w_i is the weight assigned to the i-th data sample; $\mathbb{I}(y_i \neq h(x_i))$ is an indicator function that equals to 1 if the prediction is incorrect, and 0 otherwise. A coefficient α_m is then computed as the weight of the m-th weak learner based on its performance as follow:

$$\alpha_m = \frac{1}{2} \ln\left(\frac{1 - E_m}{E_m}\right) \tag{12}$$

This coefficient reflects the influence of each weak learner in the final model, with higher values assigned to more accurate learners. Next, the weights of the training samples are updated to emphasize the falsely

predicted points:

$$w_i \leftarrow w_i \exp(\alpha_m \mathbb{I}(y_i \neq h_m(x_i))) \tag{13}$$

The updated weights are then normalized to ensure they sum to 1, preparing the dataset for the next iteration. The final AdaBoost regression model is constructed as a weighted sum of the predictions from all weak learners:

$$F(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m h_m(\mathbf{x}) \tag{14}$$

where $F(\mathbf{x})$ is the final model prediction for the input data point $\mathbf{x}$,$h_m(\mathbf{x})$ is the prediction of the m-th weak learner.

AdaBoost Regression offers several key advantages when it is used to tackle challenging prediction problems. By increasing the weights of falsely predicted samples, AdaBoost ensures that subsequent weak learners concentrate on correcting the most challenging data points. AdaBoost is also flexible because it can handle both linear and non-linear relationships by combining weak learners, such as decision stumps or shallow trees. Furthermore, AdaBoost offers the interpretability characteristics as the weights assigned to weak learners provide insight into their relative importance in the final model. Despite its strengths, AdaBoost is sensitive to noisy data and outliers, as these samples can receive excessively high weights, potentially leading to overfitting. Careful preprocessing and parameter tuning, such as selecting the number of iterations M and the learning rate, are essential for optimal performance.

3.3.8. Xgboost Regression

XGBoost Regression [68] is a powerful machine learning algorithm that extends the traditional Gradient Boosting framework with several enhancements, making it particularly effective for large-scale datasets and complex regression tasks. XGBoost combines the outputs of multiple weak learners, typically decision trees, into a strong predictive model while addressing common challenges like overfitting and computational efficiency. XGBoost incorporates key improvements over traditional Gradient Boosting, including a regularization term in the objective function to control model complexity and a second-order Taylor expansion of the loss function to optimize accuracy and efficiency. These features enable XGBoost to handle complex and non-linear relationships within the datasets.

The prediction $\widehat{y}_i$ for an input x_i in XGBoost is computed as the sum of outputs from M decision trees as follows:

$$\widehat{y}_i = \sum_{m=1}^{M} f_m(x_i), f_m \in \mathscr{F}, i = 1, \cdots, N \tag{15}$$

where $\widehat{y}_i$ is the predicted construction cost for the i-th data sample of x_i; M is the total number of decision trees; $f_m(x_i)$ is the output of the m-th decision tree for input x_i; $\mathscr{F}$ is a space of all possible trees, defined as $\mathscr{F} = \{f(\mathbf{x}) = \omega_{q(\mathbf{x})}\}$, where $q(\mathbf{x})$ maps an input $\mathbf{x}$ to a leaf in the tree, and ω is the weight associated with that leaf.

XGBoost optimizes an objective function that combines the loss function L, which measures the difference between the true and predicted values, and a regularization term Ω, which penalizes model complexity:

$$\widehat{f} = \underset{f_1 f_M}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} L(y_i, \widehat{y}_i) + \sum_{m=1}^{M} \Omega(f_m) \right\} \tag{16}$$

where y_i is the actual construction cost for the i-th data sample of x_i; $L(y_i, \widehat{y}_i)$ is the loss function such as MSE or MAE used to measure the differences between the actual cost y_i and the predicted cost $\widehat{y}_i$; $\Omega(f_m)$ is the regularization term for the complexity of the m-th tree, defined as:

$$\Omega(f_m) = \gamma T_m + \frac{1}{2}\lambda \sum_{q=1}^{T_m} \omega_{mq}^2 \tag{17}$$

where T_m is the number of leaves in the m-th tree; ω_{mq} is the weight of the q-th leaf in m-th tree; γ and λ are regularization parameters to control the penalty for the number of leaves and leaf weights, respectively. Particularly, the first component of Eq. (17) refers the tree size penalty and it aims to penalize the large trees with more leaves, encouraging simpler models. Meanwhile, the second component of Eq. (17) represents leaf weight penalty, where it penalizes large weights assigned to the leaves of the tree, encouraging smoother predictions.

XGBoost Regression offers several key advantages when it is used to tackle challenging prediction problems. The inclusion of the regularization term $\Omega(f_m)$ helps to control the model complexity of XGBoost, thus reducing the risk of overfitting. The second-order Taylor expansion of the loss function allows for efficient optimization, making XGBoost suitable for large-scale datasets. XGBoost also supports parallel computation, further accelerating training. In addition, XGBoost supports a wide range of loss functions and handles both sparse and dense data effectively. Despite its strengths, XGBoost's complexity requires careful hyperparameter tuning and validation to avoid overfitting or underfitting. Cross-validation is often employed to select the optimal combination of hyperparameters (e.g., learning rate, maximum tree depth and the number of trees), ensuring robust performance.

3.3.9. CatBoost Regression

CatBoost Regression [69] is another powerful machine learning algorithm that built on the foundations of the Gradient Boosting framework like XGBoost. One of CatBoost's key advantages is its ability to natively handle categorical features without requiring preprocessing steps such as one-hot encoding. This capability reduces the dimensionality of the feature space, mitigating overfitting while improving computational efficiency. Additionally, CatBoost employs Ordered Boosting, a technique that enhances robustness by sorting training data and using only relevant subsets of samples to train each decision tree iteration. This approach reduces data leakage and further improves generalization.

Similar to other gradient boosting algorithms, CatBoost constructs an ensemble of decision trees sequentially, where each tree attempts to correct the residuals of the previous ensemble. The prediction for an input x_i after m iterations is given by:

$$F_m(x_i) = F_{m-1}(x_i) + h_m(x_i) \tag{18}$$

where $F_m(x_i)$ is the predicted output for the i-th data sample of x_i after m iterations; $F_{m-1}(x_i)$ is the cumulative predictions from the first $(m-1)$ decision trees; $h_m(x_i)$ is the output of the m-th decision tree, trained on the residuals from $F_{m-1}(x_i)$.

Similar to XGBoost, CatBoost optimizes an objective function that combines a loss function L and a regularization term Ω to control model complexity as follow:

$$\widehat{f} = \underset{f_1 f_M}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} L(y_i, F_m(x_i)) + \sum_{m=1}^{M} \Omega(h_m) \right\} \tag{19}$$

where $L(y_i, F_m(x_i))$ is the loss function such as MSE or MAE measure the differences between the actual y_i and the predicted $F_m(x_i)$; $\Omega(h_m)$ is a regularization term for the m-th tree, penalizing model complexity to prevent overfitting, i.e.,

$$\Omega(h_m) = \gamma T_m + \frac{1}{2}\lambda \sum_{q=1}^{T_m} \omega_{mq}^2 \tag{20}$$

Similar with Eq. (17), the regularization terms in Eq. (20) are defined to balance the model complexity and prediction accuracy by discouraging

overly complex trees that may overfit the training data.

CatBoost Regression offers several key advantages when it is used to tackle challenging prediction problems. By directly encoding categorical variables, CatBoost eliminates the need for preprocessing, reducing the risk of overfitting and computational overhead. The ordered boosting mechanisms of CatBoost enhances its robustness by ensuring unbiased training and improving generalization. CatBoost also supports GPU acceleration and optimized implementations, making it suitable for large-scale datasets. While CatBoost offers substantial benefits, it can be computationally intensive for extremely large datasets, particularly when leveraging its Ordered Boosting mechanism. Proper hyperparameter tuning and validation are essential to achieve optimal results without unnecessary computational overhead.

3.3.10. HistGradient Boosting Regression

HistGradient Boosting Regression [70] is an efficient extension of the Gradient Boosting framework, specifically designed for regression tasks involving large-scale datasets and high-dimensional features. A distinctive feature of HistGradient Boosting is its histogram-based optimization, which discretizes continuous feature values into bins. This significantly reduces computational complexity compared to traditional Gradient Boosting by approximating optimal split points efficiently, making the algorithm particularly well-suited for datasets with sparse or high-dimensional features.

Similar to other gradient boosting algorithms, HistGradient Boosting constructs an ensemble of decision trees sequentially. At each iteration m, the model refines its predictions by fitting a decision tree h_m to the residuals computed from the predictions of the previous model F_{m-1}. The prediction for an input x_i after m iterations is given by:

$$F_m(x_i) = F_{m-1}(x_i) + \eta h_m\left(x_i, \{r_i\}_{i=1}^N\right) \tag{21}$$

where $F_m(x_i)$ is the prediction for the i-th data sample x_i after m iterations; $F_{m-1}(x_i)$ is the prediction from the first $(m-1)$ iterations; η is the learning rate used to control the contribution of the m-th tree; $h_m\left(x_i, \{r_i\}_{i=1}^N\right)$ is the decision tree trained on the residuals r_i; r_i is the residuals from the i-th data sample, approximated as:

$$r_i = y_i - F_{m-1}(x_i) \tag{22}$$

The residual r_i represents the negative gradients of the loss function with respect to the model's predictions $F_{m-1}(x_i)$, guiding the algorithm to reduce errors in subsequent iterations. Similar to other boosting models, HistGradient Boosting optimizes an objective function that jointly minimizes the prediction loss while incorporating a regularization penalty to control model complexity.

HistGradient Boosting Regression introduces histogram-based optimization to improve training efficiency. Unlike traditional boosting algorithms, which evaluate all possible split points, HistGradient Boosting Regression groups continues feature values into B discrete bins and computes aggregated statistics for each bin. To discretize continuous features, two binning strategies can be employed, i.e., uniform binning and quantile-based binning. Let $X = \{x_1, x_2, \cdots, x_N\}$ denote the values of a continuous feature. In uniform binning, the range of X is divided into B equal-width intervals:

$$\Delta = \frac{\max(X) - \min(X)}{B}, q_k = \min(X) + k \bullet \Delta, k = 1, 2, \cdots, B \tag{23}$$

Each data sample x_i is then assigned to a bin index b_i using:

$$b_i = \min\{k | x_i \le q_k\} \tag{24}$$

In quantile-based binning, the bin boundaries are defined by quantile thresholds such that each bin contains approximately the same number of samples:

$$q_k = \text{Quantile}\left(X, \frac{k}{B}\right), k = 1, 2, \cdots, B \tag{25}$$

The Scikit-learn implementation of HistGradient Boosting Regression allows users to select the binning strategy using the *binning_strategy* parameter ('*quantile*' or '*uniform*'), with a default bin count of $B = 255$. Quantile binning is often preferred for skewed data distributions to ensure a more balanced representation of samples across bins.

Let bin$b$ represent the set of data samples whose bin index b_i equals b, i.e., bin$b = \{i | b_i = b\}$. For each bin, the algorithm accumulates the sum of gradients G_b and Hessians H_b as:

$$G_b = \sum_{i \in \text{bin}b} g_i \tag{26}$$

$$H_b = \sum_{i \in \text{bin}b} h_i \tag{27}$$

where $g_i = \partial \ell(y_i, \hat{y}_i)/\partial \hat{y}_i$ and $h_i = \partial^2 \ell(y_i, \hat{y}_i)/\partial \hat{y}_i^2$ represent the first- and second-order gradients of the loss function ℓ, respectively. These aggregated statistics significantly reduce the complexity of split selection from $O(N.P)$ to $O(B.P)$, where N is the number of data samples, P is the number of features, and B is the number of bins.

In addition to efficient binning, HistGradient Boosting Regression incorporates a native and adaptive mechanism for handling missing values during the tree-splitting process. Rather than relying on external imputation, HistGradient Boosting Regression evaluates the gain of assigning missing values to the left or right child node during each candidate split. The direction yielding the higher gain in loss reduction is selected as the default path:

$$\text{Gain}_{\text{missing}} = \max\left\{\text{Gain}_{\text{left}}, \text{Gain}_{\text{right}}\right\} \tag{28}$$

This routing decision is dynamically re-evaluated at each node and split, allowing the model to learn data-driven assignment strategies for missing values. Unlike static imputation methods that use global statistics (e.g., mean or median), this dynamic approach preserves local structure and heterogeneity in the data. As a result, HistGradient Boosting Regression can exploit informative missingness patterns and avoid biases introduced by arbitrary imputations, which is especially beneficial in real-world datasets with non-trivial or domain-specific missingness behaviors.

HistGradient Boosting Regression primarily employs L2 regularization to prevent overfitting by penalizing large leaf values. The regularized objective function can be expressed as:

$$\mathscr{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \lambda \sum_{j=1}^{T} \omega_j^2 \tag{29}$$

where $\ell(\bullet)$ is the loss function, ω_j is the weight of the j-th leaf, and λ is the L2 regularization coefficient. Unlike XGBoost, which leverages both L1 (Lasso) and L2 (Ridge) penalties to promote sparsity and control overfitting, HistGradient Boosting Regression relies solely on L2 regularization. As such, it does not perform automatic feature selection via sparsity constraints but focuses on smoother penalty control via squared weights. As a result, HistGradient Boosting Regression does not induce sparsity in feature weights and relies more heavily on early stopping and depth control for regularization.

Beyond leaf regularization, HistGradient Boosting Regression integrates structural regularization through tree hyperparameters such as *max_depth* and *min_samples_leaf*. The *max_depth* parameter limits the maximum depth of a tree, preventing overly complex models. The *min_samples_leaf* parameter ensures that a split must have at least a specified number of training samples in a leaf node, which reduces the risk of overfitting to noise or outliers. XGBoost also supports similar structural controls using the gamma parameter (minimum loss reduction

to make a split) and *min_child_weight*. While XGBoost offers more fine-grained regularization, HistGradient Boosting Regression benefits from reduced tuning complexity and computational overhead due to its histogram-based strategy.

3.4. Model training and hyperparameter tuning

Model training is a critical process in machine learning, where the model learns patterns from input features and corresponding target outputs to make accurate predictions. The objective of training is to optimize the model parameters, such as weights and biases, by minimizing the error between the predicted outputs and the actual values. This is achieved by minimizing a predefined loss function $L(\mathbf{y}, \widehat{\mathbf{y}})$. The training process often uses algorithms like Gradient Descent, which iteratively adjust the model parameters to minimize the loss function. Effective training not only ensures accurate predictions on the training data but also enhances the model's ability to generalize to unseen data.

To evaluate models' performance and mitigate the risk of overfitting, this study employs K-Fold Cross-Validation (CV) during training. In K-Fold CV, the dataset is partitioned into k equally sized folds. The ten selected machine models are trained on $(K-1)$ folds and validated on the remaining fold, with the process repeated K times. This ensures that each subset of the data is used for validation exactly once, and the overall performance is averaged across all folds for a robust evaluation. The cross-validation score is calculated as:

$$\text{CV Score} = \frac{1}{K}\sum_{i=1}^{K}\text{Metric}(\textit{Validation}_i) \tag{30}$$

where Metric(•) represents the evaluation metric used, such as RMSE or R^2; $\textit{Validation}_i$ denotes the performance of machine learning model in the i-th validation fold. This approach provides a reliable estimate of the machine learning model's ability to generalize, ensuring fairness and consistency in the training process.

Hyperparameter tuning is another essential aspect of model training. It involves optimizing hyperparameters, such as the learning rate, tree depth, and regularization terms, which govern the learning process but are not learned directly during training. In this study, the *GridSearchCV* function is utilized to systematically evaluate all possible combinations of predefined hyperparameter values. Each hyperparameter combination is evaluated using K-Fold Cross-Validation, ensuring that the tuning process is both rigorous and comprehensive. The objective of hyperparameter tuning is to identify the hyperparameter combination that minimizes the chosen evaluation metric across all validation folds. Mathematically, this can be expressed as:

$$\text{Optimal Hyperparameters} = \underset{\text{Hyperparameter Sets}}{\text{argmin}}\frac{1}{K}\sum_{i=1}^{K}\text{Metric}(\textit{Validation}_i) \tag{31}$$

The training and hyperparameter tuning workflow in this study begins with splitting the dataset into training (70 %) and testing (30 %) subsets, reserving the testing set for final evaluation. Within the training set, K-Fold Cross-Validation is applied to evaluate the performance of various hyperparameter combinations for each selected machine learning model. GridSearchCV iterates through the predefined hyperparameter grid, selecting the combination that yields the best cross-validation performance. Once the optimal hyperparameters are identified, the machine learning model is retrained on the entire training dataset using these hyperparameters to maximize its predictive accuracy. This process is repeated for all ten machine learning models explained in the previous subsections until their optimal hyperparameters are obtained.

3.5. Performance metrics for model evaluations

The performance of the ten machine learning models for construction cost prediction is evaluated using three key metrics, namely the coefficient of determination (R^2), root mean square error (RSME), and mean bias error (MBE). The detailed definitions of these metrics are as follows.

The R^2 metric quantifies how well the regression model fits the data. It measures the proportion of variance in the dependent variable explained by the model and is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \mu)} \tag{32}$$

where y_i is the i-th actual data sample; $\widehat{y}_i$ is the i-th predicted data sample; N is the total number of data samples; i is the index of data sample; μ is the mean of actual values. Higher R^2 values indicate a better model fit, as they signify a greater proportion of variance explained by the model.

The RMSE metric measures the average magnitude of error between the predicted and actual values. It evaluates the model's prediction accuracy and is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2} \tag{33}$$

Lower RMSE values are preferred, as they indicate smaller prediction errors and higher overall model accuracy.

The MBE metric quantifies the average bias in the model's predictions. It reflects whether the model tends to overestimate or underestimate the actual values on average and is calculated as:

$$MBE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i) \tag{34}$$

A smaller absolute MBE value indicates reduced bias and improved prediction accuracy. An MBE value close to zero signifies minimal average deviation between predicted and actual values, representing a well-calibrated model.

3.6. Confidence interval

Confidence intervals (CIs) are a statistical tool used to quantify the uncertainty associated with predictions in machine learning regression tasks. They complement standard evaluation metrics such as R^2, RSME, and MBE, providing additional insight into the reliability of the predicted results. For construction cost prediction, presenting CIs alongside predicted means offers a comprehensive view of the model's performance and uncertainty.

A confidence interval represents the range within which the true value of a target variable is likely to fall, given a specified confidence level. Mathematically, the CI for a prediction $\widehat{y}_i$ is defined as:

$$CI_i = \widehat{y}_i \pm \delta\sqrt{\text{Var}(\widehat{y}_i)} \tag{35}$$

where $\widehat{y}_i$ is the Predicted value for the i-th data sample; $\text{Var}(\widehat{y}_i)$ is the prediction variance, accounting for model uncertainty and data variability; δ is a constant derived from the standard normal distribution, corresponding to the desired confidence level (e.g., $\delta = 1.96$ for a 95 % confidence interval).

The qualitative and quantitative interpretation of confidence intervals provides deeper insights into the model's predictive behavior. For qualitative interpretations, Cls are typically visualized as shaded bands around the predicted mean. A narrow band of Cls reflects high certainty in the model's predictions, while a wide band indicates greater uncertainty. Overlaying CIs on actual test data points allows for intuitive

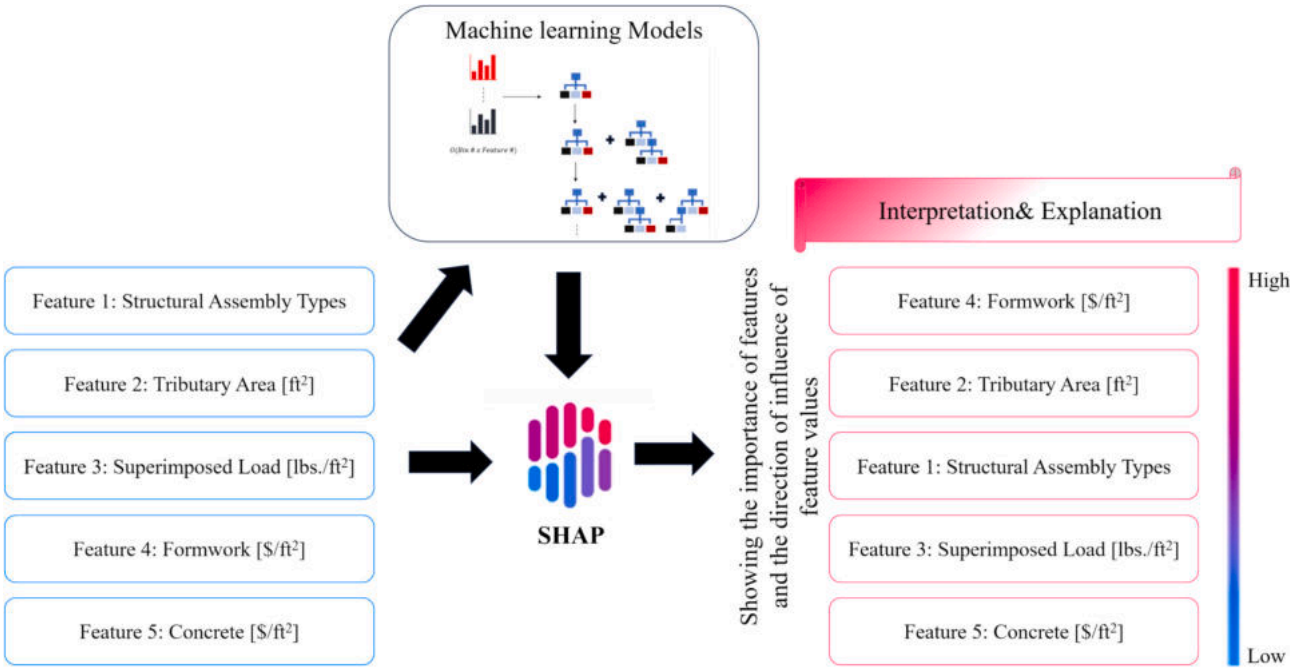


Fig. 3. Conceptual diagram illustrating the integration of SHAP in machine learning models for construction cost prediction.

assessment of prediction reliability and uncertainty. For quantitative interpretation, a CI captures the range where the true target value is likely to lie for a given confidence level. For instance, a 95 % CI implies that, in repeated predictions under similar conditions, approximately 95 % of the intervals would contain the true value. Wider CIs may indicate challenges in the data (e.g., noise, outliers) or limitations of the model, while narrower CIs suggest stable and consistent predictions.

In the context of construction cost prediction, confidence intervals can serve different purposes. For instance, CIs explicitly communicate the level of uncertainty in the machine learning model's predictions, helping stakeholders understand the reliability of the results. By visualizing CIs alongside predicted means and actual data points, stakeholders can better evaluate the prediction performances of different machine learning models qualitatively. Finally, incorporating CIs also enables informed decision-making, especially in scenarios where high prediction reliability is crucial.

3.7. Shap-based explainability

In machine learning applications, explainability plays a critical role in building trust and enabling stakeholders to understand the rationale behind model predictions. Explainability refers to the degree to which humans can comprehend the decision-making process of a machine learning model. Greater explainability facilitates better understanding of a model's internal mechanisms and its results. This is especially important in construction cost prediction, where decisions often involve significant financial and resource commitments. During the modeling phase, explainability helps developers understand model behavior, select and optimize models, and make necessary adjustments. In the operational phase, it provides stakeholders with insights into the model's reasoning, fostering confidence and enabling informed decision-making.

SHAP [71] is a widely adopted method for interpreting the outputs of machine learning models, as shown in Fig. 3. SHAP is based on Shapley values, a concept from cooperative game theory introduced by Lloyd Shapley. Shapley values were designed to fairly distribute gains or contributions among players in a cooperative game. In the context of machine learning, this principle is adapted to allocate the contribution of each feature to the model's prediction. SHAP offers several advantages, including model-agnostic applicability, instance-level interpretability, and the ability to provide global insights by aggregating feature contributions across the dataset.

Mathematically, the SHAP value for a feature z_j quantifies its contribution to the model's prediction by considering all possible subsets $\mathscr{S}$ of features that exclude z_j. The SHAP value is calculated as:

$$\varnothing_j = \sum_{S \subseteq \mathscr{Z} \setminus \{z_j\}} \frac{|\mathscr{S}|! \bullet (|\mathscr{Z}| - |\mathscr{S}| - 1)!}{|\mathscr{Z}|!} \bullet \left[f(\mathscr{S} \cup \{z_j\}) - f(\mathscr{S}) \right] \tag{36}$$

where $\mathscr{Z}$ is the set of all features; $\mathscr{S}$ is any subset of features excluding z_j; $f(\mathscr{S})$ represents the model prediction using only the features in subset $\mathscr{S}$; $f(\mathscr{S} \cup \{z_j\}) - f(\mathscr{S})$ denotes the marginal contribution of z_j when added to the subset $\mathscr{S}$. Note that the factorial terms in Eq. (29) weight the contributions of feature z_j based on all possible feature orderings, ensuring a fair distribution of important values.

Integrating SHAP into this study significantly enhances the interpretability of the machine learning models used for construction cost prediction. SHAP not only ensures accurate predictions but also provides valuable insights into how each feature influences the model's output. For example, it quantifies the impact of factors such as structural assembly type, tributary area, superimposed load, and the unit cost of formwork on predicted costs. This level of interpretability is crucial for identifying key drivers of cost variability, detecting potential model biases, and explaining individual predictions to stakeholders. By enabling stakeholders to make more informed and transparent decisions, this approach bridges the gap between predictive performance and practical applicability, addressing the critical need for explainable models in high-stakes domains like construction.

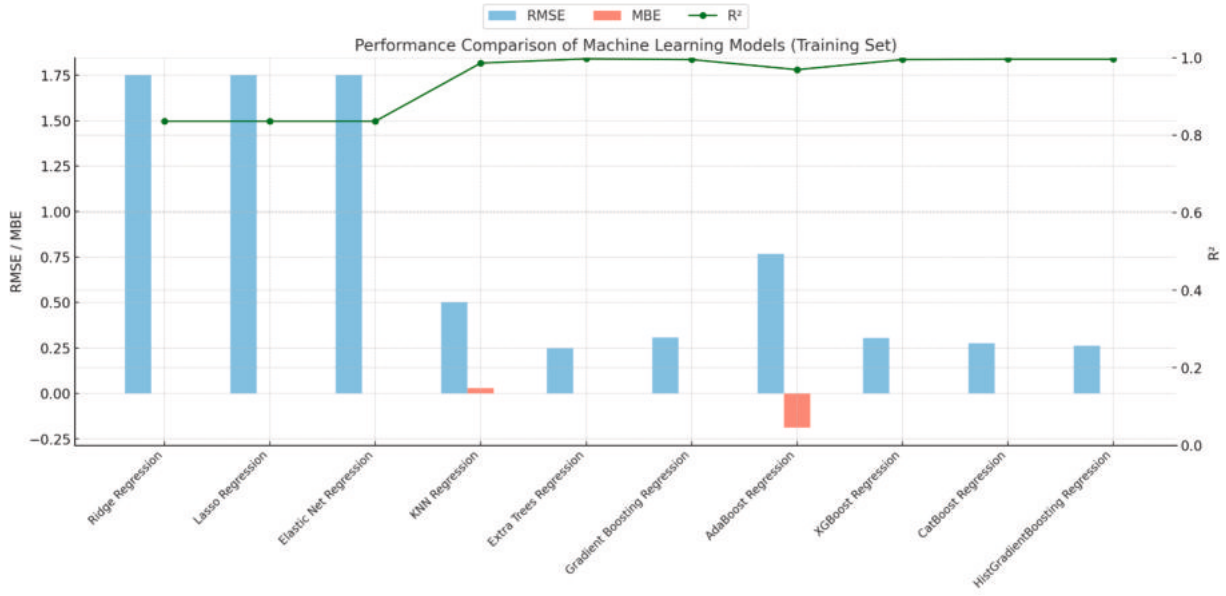
4. Performance evaluation and analysis

4.1. Comparisons of standard metrics

The performance of ten machine learning models (i.e., Ridge Regression, Lasso Regression, Elastic Net Regression, KNN Regression, Extra Trees Regression, Gradient Boosting Regression, AdaBoost, XGBoost, CatBoost, and HistGradientBoosting Regression) on the

Table 2
Performance comparison of ten machine learning models on standard metrics.

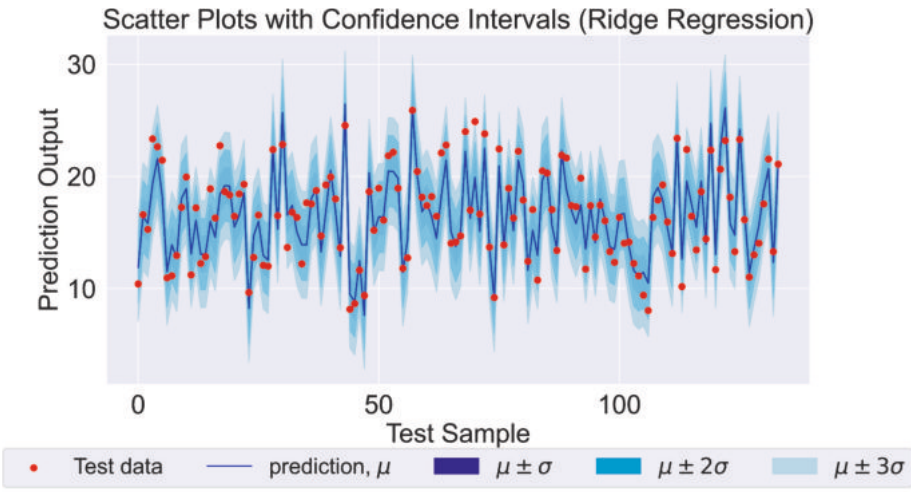
Machine Learning Models	Training Dataset			Testing Dataset		
	R^2	RSME	MBE	R^2	RSME	MBE
Ridge Regression	0.836	1.750	0.000	0.827	1.790	0.102
Lasso Regression	0.836	1.750	0.000	0.827	1.792	0.102
Elastic Net Regression	0.836	1.750	0.000	0.827	1.792	0.102
KNN Regression	0.986	0.503	0.029	0.951	0.951	0.042
Extra Trees Regression	0.997	0.249	0.000	0.984	0.551	−0.010
Gradient Boosting Regression	0.995	0.307	0.000	0.987	0.500	−0.008
AdaBoost Regression	0.969	0.768	−0.188	0.951	0.954	−0.201
XGBoost Regression	0.995	0.306	0.000	0.988	0.478	−0.010
CatBoost Regression	0.996	0.275	0.000	0.987	0.485	−0.010
HistGradientBoosting Regression	0.996	0.262	0.000	0.988	0.477	−0.009



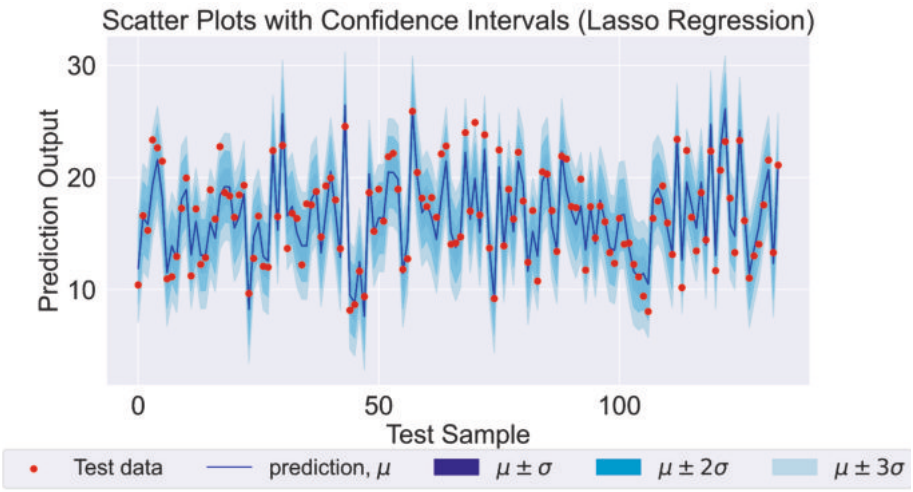
(a)

(b)

Fig. 4. Performance comparison of ten machine learning models using the three metrics (i.e., R^2, RSME and MBE) on: (a) training dataset and (b) testing dataset.



(a)



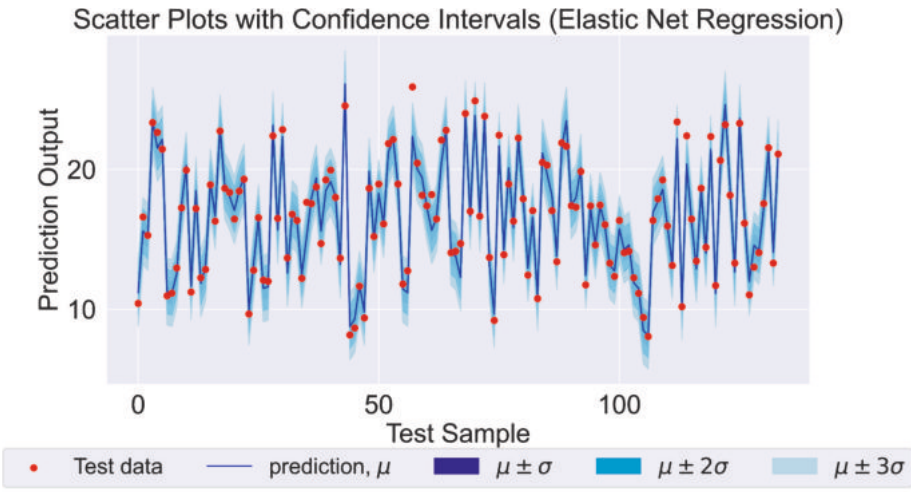
(b)

Fig. 5. Scatter plots of all ten machine learning models to solve the testing datasets, i.e., (a) Ridge Regression, (b) Lasso Regression, (c) Elastic Net Regression, (d) KNN Regression, (e) Extra Trees Regression, (f) Gradient Boosting Regression, (g) AdaBoost, (h) XGBoost, (i) CatBoost, and (j) HistGradientBoosting Regression.

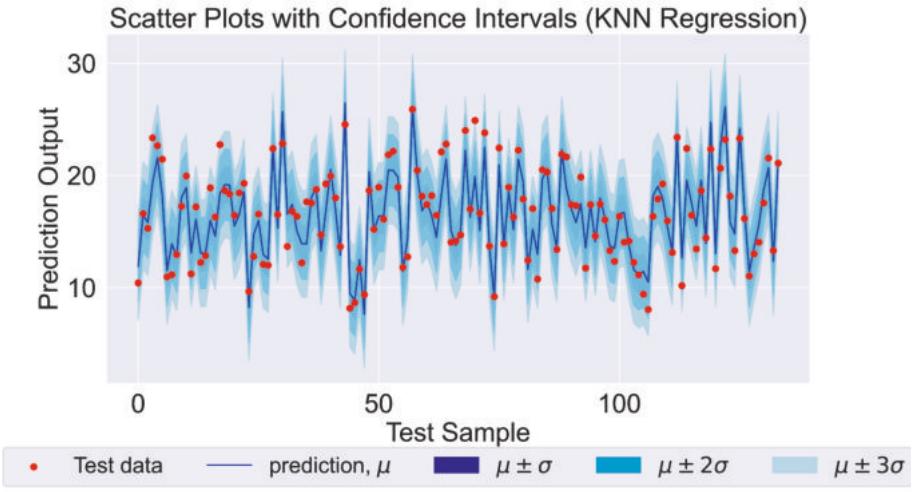
construction cost prediction is presented in Table 2 and Fig. 4. These machine learning models were evaluated using three metrics, i.e., R^2, RMSE, and MBE on both the training dataset (70 %) and testing dataset (30 %).

On the training dataset, Ridge Regression, Lasso Regression, and Elastic Net Regression demonstrate identical performance, with R^2 values of 0.836 and RMSE values of 1.750. Their MBE values of 0.000 suggest no systematic bias in predictions. While these models effectively capture basic patterns in the data, their relatively moderate R^2 and high RMSE indicate limited precision compared to more advanced machine learning models. KNN Regression exhibits a notable improvement, achieving an R^2 of 0.986 and a much lower RMSE of 0.503, highlighting its strong ability to fit the training data. However, its slight positive MBE of 0.029 suggests a minor tendency to overestimate. Among ensemble methods, AdaBoost performs the worst with an R^2 of 0.969 and an RMSE of 0.768. Its negative MBE of −0.188 reveals a tendency to underestimate predictions. Gradient Boosting Regression achieves an R^2 of 0.995 and an RMSE of 0.307, with an MBE of 0.000, highlighting its excellent performance. XGBoost and CatBoost also perform outstandingly, with R^2 values of 0.995 and 0.996, respectively, and RMSE values of 0.306 and 0.275, demonstrating high precision and predictive capability. Both models exhibit near-zero MBEs, indicating their unbiased predictions. HistGradientBoosting stands out with an R^2 of 0.996 and an RMSE of 0.262, showcasing exceptional precision and minimal bias, further confirming its strong predictive power. Extra Trees Regression achieves exceptional results with an R^2 of 0.997 and the lowest RMSE at 0.249, indicating near-perfect performance in capturing data patterns. Its MBE of 0.000 reflects an absence of bias.

On the testing dataset, Ridge Regression, Lasso Regression, and Elastic Net Regression maintain consistent performance, each achieving an R^2 of 0.827. However, their RMSE values increase slightly to approximately 1.792, and their positive MBE of 0.102 suggests a small overestimation bias, reflecting their limited adaptability to new data. KNN Regression continues perform better than Ridge Regression, Lasso Regression and Elastic Net Regression on the testing dataset, achieving an R^2 of 0.951, though its RMSE increases to 0.951, indicating slight overfitting compared to its training results. Its MBE of 0.042 shows a reduced tendency to overestimate predictions. The six ensemble models continue to excel on the testing dataset. AdaBoost continues to perform the worst among all ensemble models with an R^2 of 0.951 and an RMSE of 0.954, though its negative MBE of −0.201 indicates a consistent underestimation bias. Despite having the best performance to solve training datasets, some performance degradations were observed from Extra Trees Regression when it is used to solve the unseen testing datasets, implying its limited generalization capability. Among all ensemble models used to solve the testing datasets, Extra Trees



(c)



(d)

Fig. 5. (*continued*).

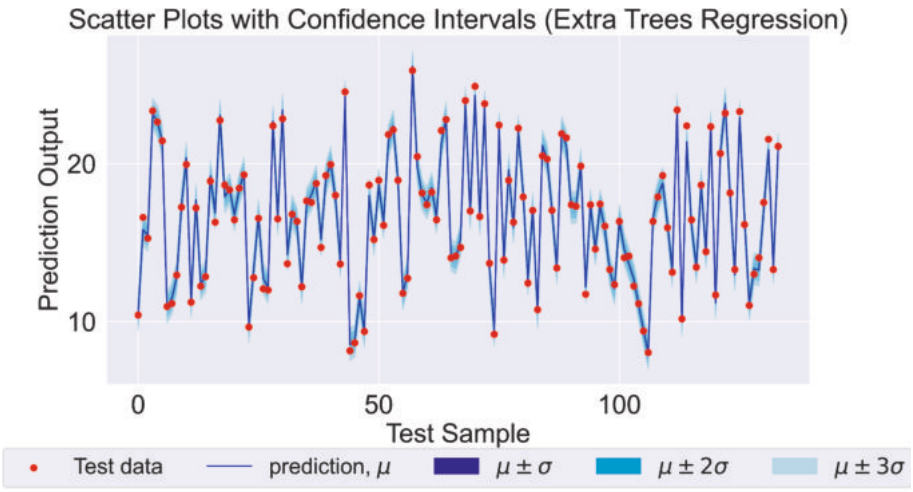
Regression only outperforms AdaBoost by achieving an R^2 of 0.984, coupled with a RMSE of 0.551. Its near-zero MBE of −0.010 reflects minimal bias. This is followed by the Gradient Boosting Regression that performs impressively, achieving an R^2 of 0.987 and an RMSE of 0.500. Its MBE of −0.008 signifies negligible bias, reinforcing its predictive reliability. Both of the XGBoost and CatBoost remain as competitive performers when solving the testing datasets, with R^2 values of 0.988 and 0.987, respectively, and RMSE values of 0.478 and 0.485. Both models exhibit near-zero MBEs, emphasizing their accuracy and reliability. Finally, HistGradientBoosting emerges as the best performer, achieving an R^2 of 0.988 and the lowest RMSE of 0.477 on the testing dataset. Its near-zero MBE of −0.009 confirms minimal bias, making it the most reliable model for construction cost prediction tasks.

4.2. Comparison of scatter plots with confidence intervals

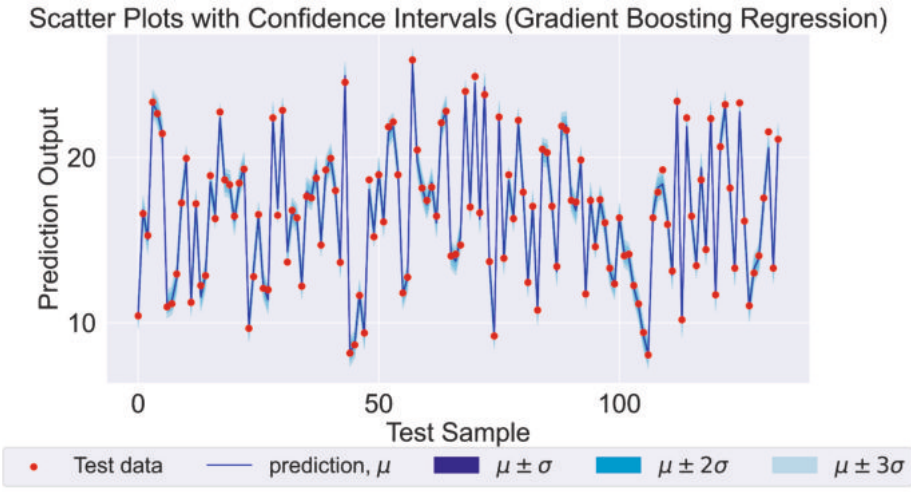
Fig. 5 presents the scatter plots and confidence intervals for the ten machine learning models (i.e., Ridge Regression, Lasso Regression, Elastic Net, KNN Regression, Extra Trees Regression, Gradient Boosting Regression, AdaBoost Regression, XGBoost Regression, CatBoost Regression, and HistGradientBoosting Regression) used in construction cost prediction. These visualizations capture the relationship between predicted and actual values, along with the prediction uncertainty represented by confidence intervals. The red scatter points in Fig. 5 indicate the true test data values, the blue lines depict the predicted means (μ), and the shaded regions correspond to $\mu \pm \sigma$, $\mu \pm 2\sigma$, and $\mu \pm 3\sigma$. This analysis allows for a comprehensive evaluation of both predictive accuracy and reliability.

Referring to Fig. 5, it is observed that the scatter plot for KNN Regression exhibits the widest confidence intervals and significant variability in predictions. The predicted values deviate notably from the actual values, reflecting the model's sensitivity to noise and localized data distributions. This limits KNN's suitability for complex prediction tasks, such as construction cost estimation. Ridge Regression, Lasso Regression, and Elastic Net demonstrate narrower confidence intervals compared to KNN Regression but still exhibit higher variability than the ensemble-based models. These linear models provide relatively consistent alignment between predicted and actual values, with Elastic Net striking a balance between Ridge Regression and Lasso Regression in handling correlated features. However, their moderate prediction accuracy and wider intervals highlight their limitations in capturing complex relationships.

Meanwhile, AdaBoost Regression shows a clear improvement over the KNN Regressions and linear models (i.e., Ridge Regression, Lasso Regression, and Elastic Net), with narrower confidence intervals and better alignment between predicted and actual values. Although slight underestimation is observed, indicated by consistent bias, AdaBoost Regression effectively captures the main factors influencing construction costs and demonstrates robust predictive performance. Extra Trees Regression refines this further, exhibiting moderate confidence intervals



(e)



(f)

Fig. 5. (*continued*).

and strong predictive accuracy. Its scatter plot shows minimal systematic bias, making it a reliable option for moderately complex datasets. Gradient Boosting Regression narrows the confidence intervals even further, demonstrating its higher prediction reliability. Its predicted values are closely aligned with the actual data points, reflecting the model's capability to capture complex, non-linear relationships in construction cost prediction.

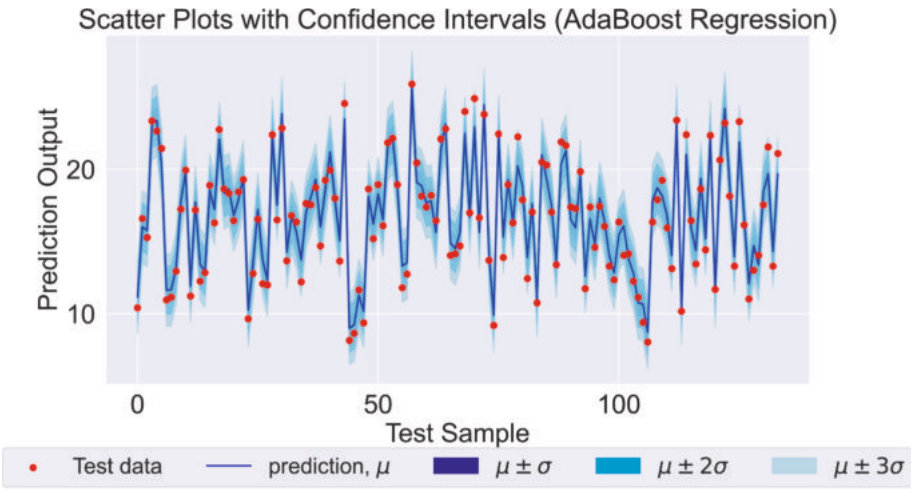
XGBoost Regression builds upon the strengths of Gradient Boosting Regression, with slightly narrower confidence intervals and consistently accurate predictions, showcasing its robustness across diverse data distributions. CatBoost Regression performs similarly to XGBoost Regression, offering equally narrow confidence intervals and highly stable predictions. Its native handling of categorical features and efficient processing of large-scale datasets ensure its practical utility. HistGradientBoosting Regression emerges as the best performer among all machine learning models. Its scatter plot demonstrates the narrowest confidence intervals and nearly perfect alignment between predicted and actual values. This reflects its exceptional stability, minimal bias, and unparalleled predictive accuracy, particularly in handling complex real-world datasets.

The scatter plots and confidence intervals presented in Fig. 5 reveal a clear progression in machine learning model performance. KNN Regression exhibits the widest intervals and greatest variability, followed by Ridge Regression, Lasso Regression, and Elastic Net with moderate intervals and consistent predictions. AdaBoost Regression improves further with narrower intervals and reduced bias. Extra Trees Regression transitions to higher accuracy, while Gradient Boosting Regression, XGBoost Regression, and CatBoost Regression deliver superior performance. HistGradientBoosting Regression stands out as the most reliable model, offering the highest accuracy and stability for construction cost prediction.

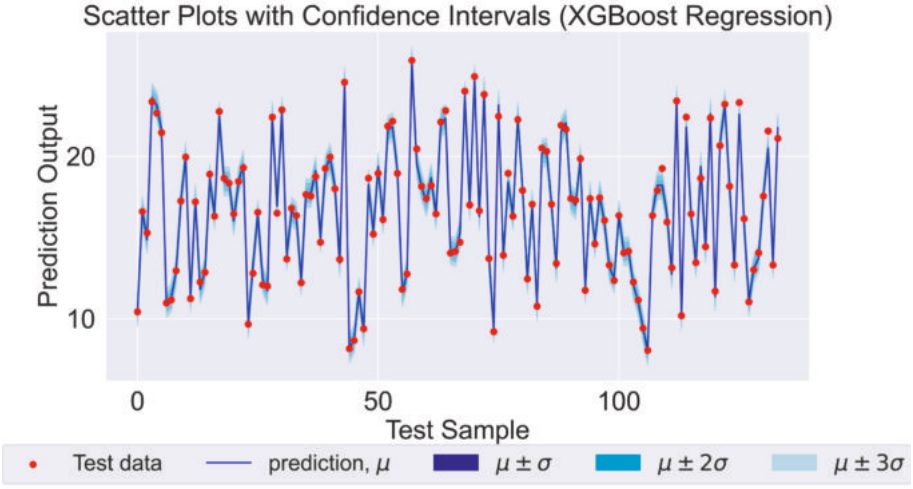
4.3. Comparison of residual distribution plots

In Fig. 6, the residual distribution plots for the ten machine learning models (Ridge Regression, Lasso Regression, Elastic Net, KNN Regression, Extra Trees Regression, Gradient Boosting Regression, AdaBoost Regression, XGBoost Regression, CatBoost Regression, and HistGradientBoosting Regression) provide insights into their predictive performance and error patterns. These plots reveal the degree to which each model minimizes prediction errors and the extent of their alignment with a normal distribution.

The residuals of Ridge Regression and Lasso Regression exhibit wider spreads and deviations from a normal distribution. The residual distributions suggest that these linear models are less effective in capturing the complexity of construction cost prediction data, as they fail to adequately minimize errors and exhibit higher variability. Elastic Net performs slightly better, with a narrower spread and residuals closer to a normal distribution. However, its performance remains limited compared to more advanced models. KNN Regression also shows a wide



(g)



(h)

Fig. 5. (*continued*).

residual distribution with noticeable asymmetry, reflecting its sensitivity to noise and local variations in the dataset. This highlights its limitations in handling complex, large-scale data, as it struggles to generalize beyond local patterns.

AdaBoost Regression demonstrates an improved residual distribution compared to the four earlier machine learning models, with residuals more concentrated and closer to a normal shape. However, the slight skewness indicates some prediction bias and sensitivity to outliers. Extra Trees Regression achieves a much tighter residual distribution, with most residuals concentrated around zero and fewer deviations, though a slight dispersion remains. This indicates better generalization and predictive accuracy, although it may still occasionally overfit certain features. Gradient Boosting Regression and XGBoost Regression both produce residuals that closely resemble a normal distribution, with minimal deviations and spikes. Their concentrated residuals signify strong predictive performance and the ability to model complex relationships effectively.

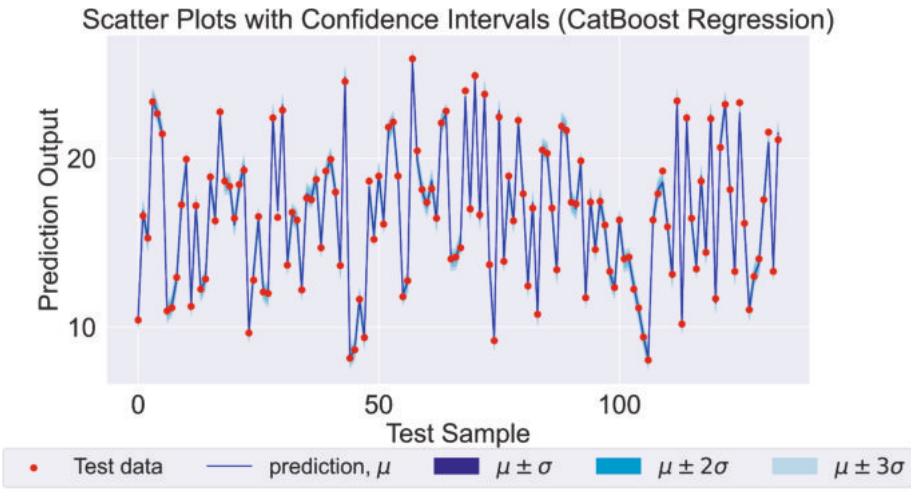
CatBoost Regression achieves one of the most concentrated and symmetric residual distributions, underscoring its robustness in handling complex datasets with minimal bias. HistGradientBoosting Regression stands out as the best performer, with residuals exhibiting the tightest clustering and the closest alignment to a normal distribution. This demonstrates its superior predictive accuracy, adaptability, and robustness in dealing with complex construction cost data.

Based on the findings observed from Fig. 6, it can be concluded that Ridge Regression, Lasso Regression, Elastic Net, and KNN Regression tend to show broader residual distributions and limited predictive accuracy, reflecting their inability to handle the complexity of the data effectively. AdaBoost Regression and Extra Trees Regression perform better, but their residuals still exhibit some dispersion or bias. The best-performing models, namely Gradient Boosting Regression, XGBoost Regression, CatBoost Regression, and HistGradientBoosting Regression, demonstrate the tightly concentrated, near-normal residual distributions, with HistGradientBoosting Regression achieving the highest predictive accuracy and reliability. These observations highlight the importance of selecting advanced machine learning models for construction cost prediction to achieve greater accuracy and robustness.

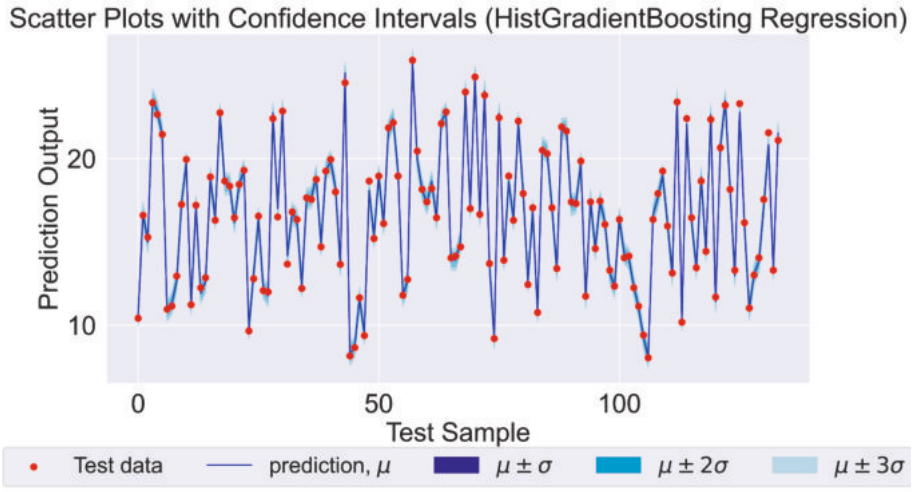
4.4. Comparison of SHAP plots and feature importance

Fig. 7 provides a comparative analysis of the SHAP plots and feature importance for four selected machine learning models (i.e., Lasso Regression, KNN Regression, AdaBoost Regression, and HistGradientBoosting Regression) applied to construction cost prediction. These SHAP analyses offer insights into the unique strengths and limitations of each model by examining their feature attributions.

In Fig. 7(a), the SHAP plot for Lasso Regression reveals a primary reliance on two features, namely “Formwork” and “Tributary Area.” Among these, “Formwork” exhibits the highest mean SHAP value, signifying its critical role in influencing predictions. Its SHAP values are



(i)



(j)

Fig. 5. (*continued*).

tightly distributed with predominantly positive contributions, indicating consistent importance across various samples. "Tributary Area," ranking second in importance, displays a balanced distribution of SHAP values across positive and negative contributions, highlighting its context-dependent role. The remaining features (i.e., "Type," "Superimposed Load," and "Concrete") exhibit minimal contributions, with their SHAP values narrowly centered around zero. This pattern reflects Lasso Regression's inherent feature selection capability, effectively prioritizing dominant features while suppressing noise.

The SHAP plot for KNN Regression in Fig. 7(b) highlights its sensitivity to local data variations, as evidenced by the wide distribution of SHAP values across features. "Formwork" is the most influential feature, with the highest mean SHAP value and contributions spanning both positive and negative ranges, demonstrating KNN's sensitivity to variations in formwork costs across local data neighborhoods. The categorical feature "Type" shows moderate importance, with SHAP values concentrated within a smaller range, reflecting its localized but meaningful impact on predictions. "Tributary Area" also plays a significant role, with substantial SHAP value variability, indicating its influence depends on specific configurations in the data. In contrast, the contributions of "Superimposed Load" and "Concrete" are relatively minor, with smaller mean SHAP values, suggesting these features play a secondary role in KNN's predictions.

Fig. 7(c) showcases the SHAP plot for AdaBoost Regression, emphasizing its adaptive nature in capturing feature importance. "Formwork" emerges as the most critical feature, with high and consistently positive SHAP values. The compact distribution of its SHAP values reflects AdaBoost's ability to focus on dominant features while minimizing noise. "Tributary Area" ranks second, with SHAP values spread across positive and negative ranges, demonstrating AdaBoost's ability to capture the varying influence of this feature on construction cost predictions. "Type" and "Concrete" also contribute meaningfully, with moderately distributed SHAP values, indicating a broader utilization of features compared to Lasso Regression and KNN Regression. However, "Superimposed Load" shows minimal impact, with SHAP values tightly concentrated around zero, signifying its limited relevance in this model's predictions.

Finally, Fig. 7(d) presents the SHAP plot for HistGradientBoosting Regression, which demonstrates the most robust feature importance distribution among the four models. "Formwork" is again the most influential feature, with the highest mean SHAP value and a broad yet consistent range of positive contributions. This reflects HistGradientBoosting's ability to effectively capture nuanced variations in formwork's impact across samples. "Tributary Area" is the second most significant feature, with balanced SHAP value contributions that vary depending on context, highlighting the model's capability to account for complex feature interactions. The categorical feature "Type" also plays a significant role, with a concentrated SHAP value distribution, indicating its use as a secondary predictor to refine predictions. The contributions of "Superimposed Load" and "Concrete" are minimal, as indicated by

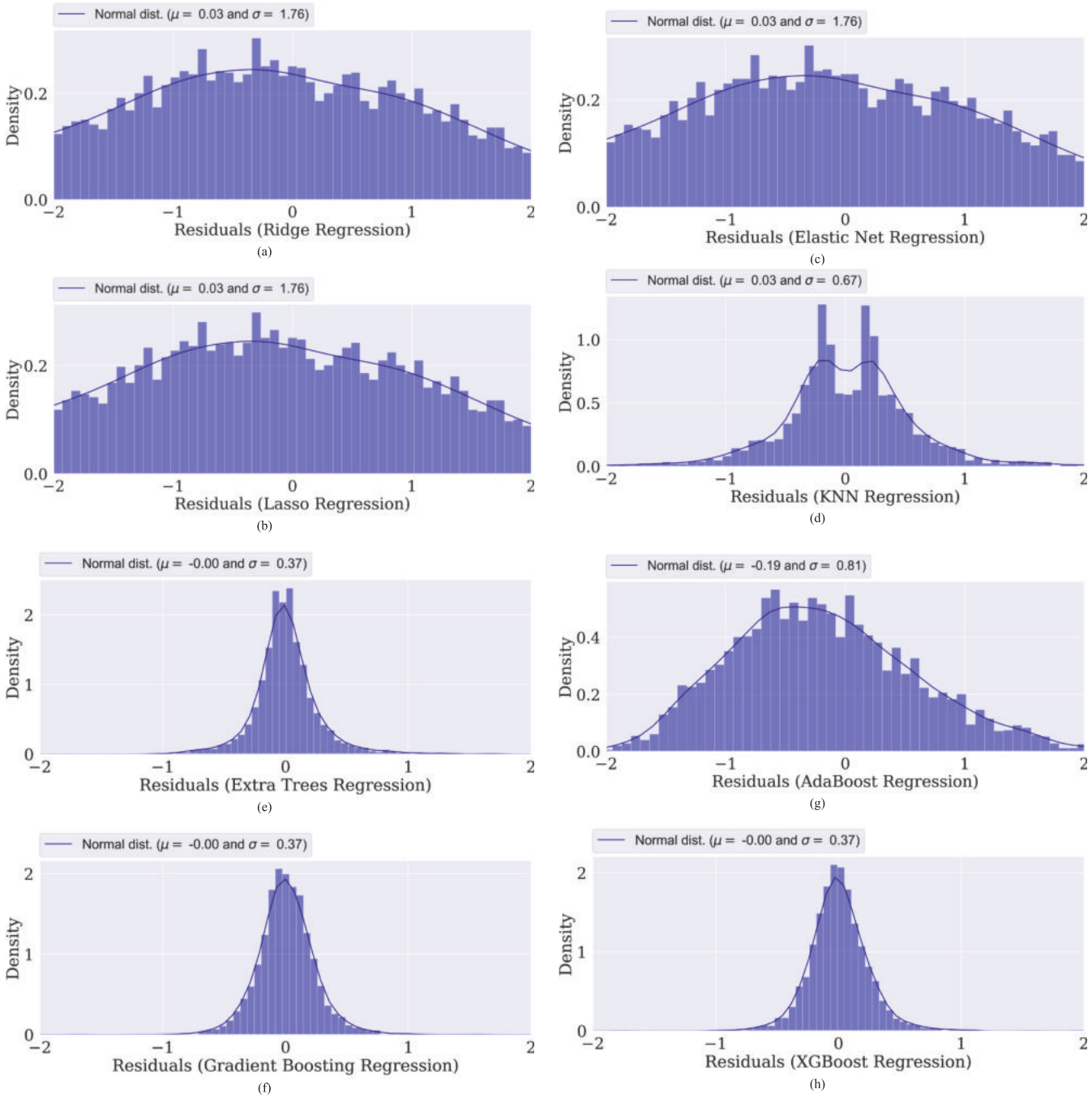


Fig. 6. Residual plots of all ten machine learning models to solve the testing datasets, i.e., (a) Ridge Regression, (b) Lasso Regression, (c) Elastic Net Regression, (d) KNN Regression, (e) Extra Trees Regression, (f) Gradient Boosting Regression, (g) AdaBoost, (h) XGBoost, (i) CatBoost, and (j) HistGradient Boosting Regression.

their narrow SHAP value distributions centered near zero, showcasing HistGradientBoosting's ability to deprioritize less significant features while maintaining predictive accuracy.

To further quantify the ability of HistGradientBoosting Regression to model nonlinear feature interactions, a SHAP interaction value heatmap is presented in Fig. 8. This heatmap decomposes the total SHAP values into main effects (diagonal values) and pairwise interaction effects (off-diagonal values). Notably, "Formwork" exhibits the highest main effect (2.362), confirming its standalone predictive strength, which aligns with the SHAP summary plot in Fig. 7(d). Beyond individual importance, significant pairwise interactions are observed, i.e., most prominently between "Type" and "Formwork" (0.300), as well as "Formwork" and "Concrete" (0.225). These elevated interaction values indicate that the contribution of one feature is influenced by the value of another, highlighting nonlinear dependencies. For example, the predictive influence of "Formwork" may vary depending on the structural "Type" or the composition of "Concrete." The magnitude of these interactions affirms that HistGradientBoosting Regression does not treat features independently but learns joint contributions through hierarchical partitioning, thus demonstrating its inherent capacity to model complex, nonlinear relationships. In contrast to linear models such as Lasso or ensemble methods like AdaBoost that primarily capture additive relationships, HistGradientBoosting leverages tree-based recursive partitioning, enabling the discovery of conditional and interactive effects among

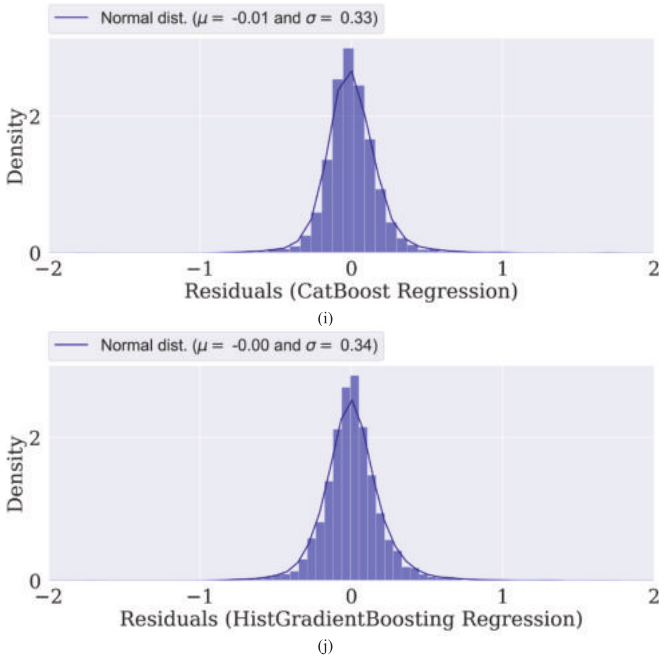


Fig. 6. (*continued*).

features. This provides a direct quantification of the nonlinear feature interactions modeled by HistGradientBoosting, addressing the need for interpretability and fulfilling the goal of understanding higher-order effects in construction cost prediction.

Across the four selected machine learning models, "Formwork" consistently emerges as the most influential feature, revealing its critical role in construction cost predictions. While Lasso Regression and KNN Regression focus heavily on a limited number of features, AdaBoost and HistGradientBoosting show greater adaptability in capturing broader feature contributions. HistGradientBoosting, in particular, outperforms the others by balancing feature importance effectively, maintaining robustness, and deprioritizing irrelevant features. These findings provide valuable insights into selecting the appropriate model for construction cost prediction tasks based on feature attribution patterns.

4.5. Comparison of runtime and computational efficiency

In practical applications such as construction cost estimation, computational efficiency is a critical consideration, particularly when models are deployed in real-time systems, iterative design processes, or embedded within digital twin environments. To assess the computational characteristics of the evaluated models, we measured three types of computing times for each machine learning model: (a) training time, which denotes the duration required to fit the model to the training dataset; (b) training set prediction time, indicating how long the model takes to generate predictions on the same dataset it was trained on; and (c) testing set prediction time, which reflects the time required to generate predictions for previously unseen data. These three metrics offer a holistic view of model efficiency during both development and deployment phases.

Table 3 presents a comparative summary of the runtime performance for all ten models. As expected, linear models such as Ridge Regression, Lasso Regression, and Elastic Net Regression recorded the fastest training and prediction times, completing operations in milliseconds. Their simplicity, stemming from closed-form solutions or convex optimization with limited parameters, makes them ideal for small-scale or resource-constrained applications. Notably, their testing set prediction time is recorded as 0.0 s due to the limitations of Python's timing resolution when handling operations that complete within microseconds. This does not imply a zero-runtime cost, but rather reflects the negligible latency at such a fine-grained resolution.

In contrast, ensemble models inherently involve more computational complexity. Among them, HistGradientBoosting Regression exhibited a strong balance between speed and predictive power. It completed training in 82.04 s, which is 4.5 times faster than XGBoost (369.29 s), demonstrating its superior efficiency in handling medium-scale datasets like RSMeans. Compared to CatBoost (12.28 s), HistGradientBoosting Regression is slower, but this trade-off is justified by its greater stability, better generalization, and more robust performance in sparse and high-dimensional data settings. This advantage is attributed to its histogram-based split-finding mechanism and native handling of missing values, which enable consistent learning even in noisy or incomplete construction datasets.

HistGradientBoosting Regression also maintains low prediction latency, requiring only 0.0363 s to generate results on the test set. This represents less than 0.05 % of its total training time, highlighting the model's responsiveness in real-time scenarios where frequent updates and low-latency outputs are required. While CatBoost achieved even faster training and prediction times, HistGradientBoosting Regression consistently delivered superior uncertainty quantification and model interpretability, making it more suitable for applications that require explainable, risk-aware cost predictions. Moreover, Gradient Boosting Regression and AdaBoost Regression, while comparable in architecture, required longer training times (126.27 s and 109.60 s respectively) and exhibited lower prediction accuracy and interpretability in this study. These results further reinforce the suitability of HistGradientBoosting Regression in balancing performance and transparency.

Extra Trees Regression also deserves mention. Although it achieved faster training than XGBoost (196.29 s vs. 369.29 s), it still lagged behind HistGradientBoosting Regression. Unlike boosting models that construct trees sequentially based on gradient optimization, Extra Trees Regression builds all trees in parallel using random splits. This architectural difference removes the overhead of iterative loss minimization, but also limits the model's capacity to capture subtle patterns, leading to slightly lower predictive performance.

To further substantiate the efficiency claims of HistGradientBoosting Regression, we reference benchmark results from the Scikit-learn documentation [72], which indicate that for datasets with N samples

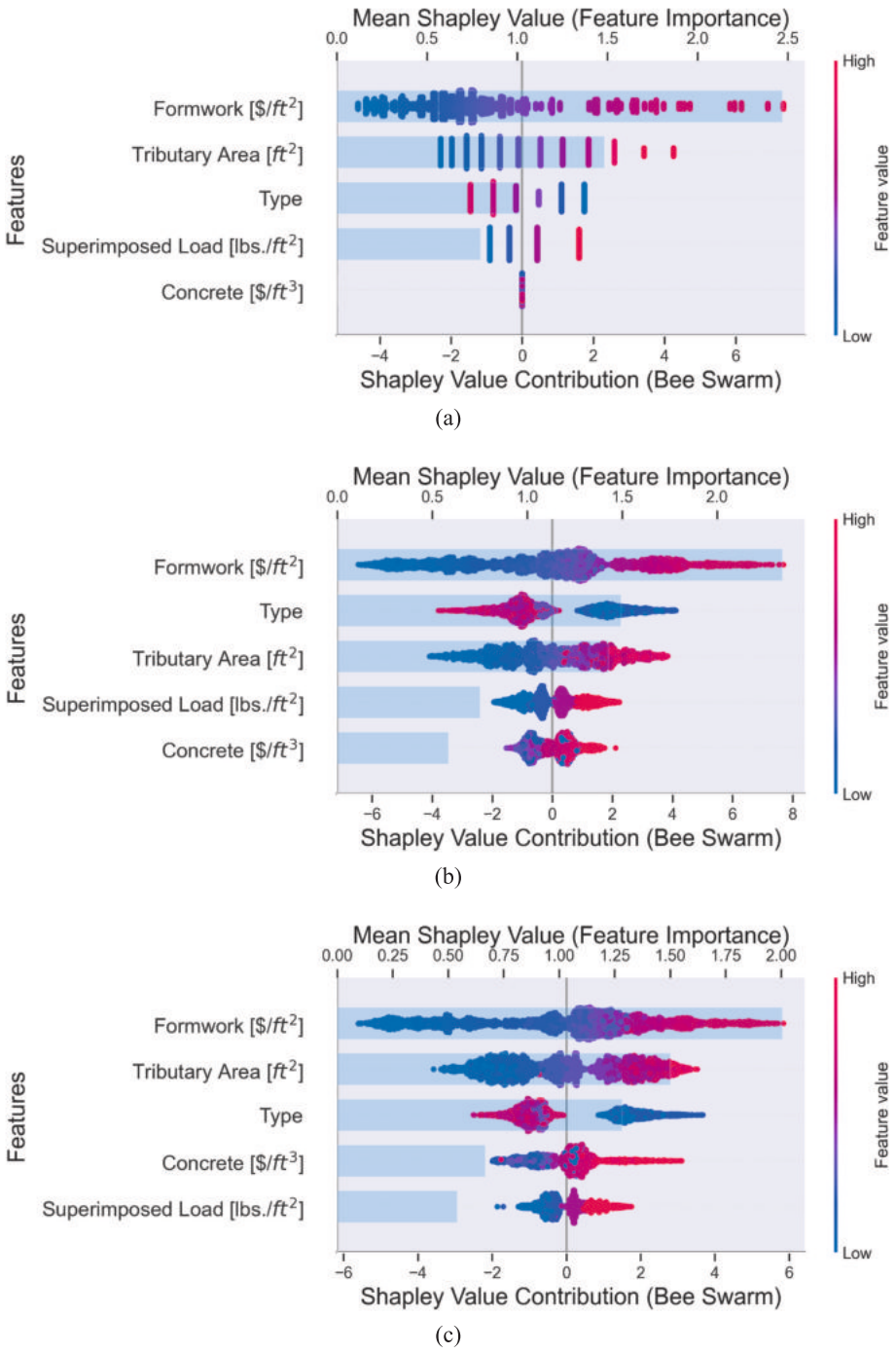


Fig. 7. SHAP and feature importance plots of four selected machine learning models to solve the testing datasets, i.e., (a) Lasso Regression, (b) KNN Regression, (c) AdaBoost, and (d) HistGradientBoosting Regression.

and P features, HistGradientBoosting Regression requires only approximately $0.4NP$ floating-point units of memory, in contrast to the $1.2NP$ requirement for XGBoost due to pre-sorted feature value storage. When tested on the Higgs Boson dataset (10 million samples × 28 features), HistGradientBoosting Regression completed training in just 42 s, while XGBoost required 189 s, i.e., a 4.5 times speed-up. These findings affirm the scalability and suitability of HistGradientBoosting Regression for large-scale industrial tasks.

In summary, HistGradientBoosting Regression stands out as a well-rounded model that combines computational efficiency with strong predictive performance and transparency. Although it is not the fastest ensemble model in absolute terms, its balanced runtime behavior, ability to quantify uncertainty, and explainable outputs make it highly suitable for real-world construction cost modeling where both speed and reliability are crucial.

5. Discussions

5.1. Comparative Insights: HistGradientBoosting versus other ensemble models

To justify the adoption of HistGradient Boosting Regression, it is

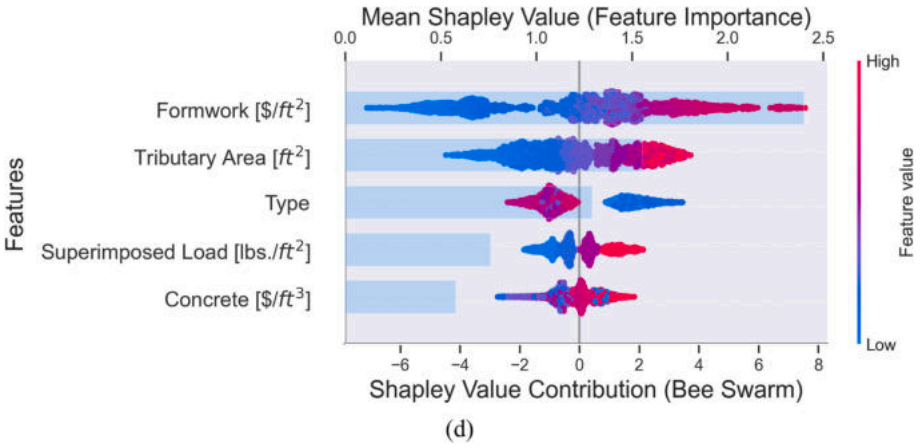


Fig. 7. (*continued*).

crucial to compare its performance and structural characteristics against three widely used gradient boosting models, i.e., the XGBoost, CatBoost, and LightGBM. While all are grounded in the gradient boosting framework, their implementations diverge significantly in computational strategy, regularization mechanisms, and data handling capabilities.

Compared to XGBoost, HistGradient Boosting Regression offers a more streamlined training process through its histogram-based split finding and native integration in Scikit-learn. XGBoost, though highly flexible, relies on exact greedy or approximate algorithms that can be computationally intensive. Moreover, while XGBoost supports both L1 (Lasso) and L2 (Ridge) regularization, enabling automatic feature selection through sparsity, it typically demands extensive hyperparameter tuning. In contrast, HGB adopts a more restrained L2-only regularization, which reduces the risk of overfitting while promoting smoother model convergence and interpretability.

Relative to CatBoost, HistGradient Boosting Regression lacks native support for categorical variables but compensates with faster training times and lower memory usage. CatBoost's ordered boosting and categorical encoding strategies offer clear advantages in datasets rich in categorical features, but these come with additional algorithmic complexity. In contrast, HistGradient Boosting Regression assumes numerical inputs, which aligns well with the nature of our dataset, where features are predominantly continuous or have been numerically encoded. Furthermore, HistGradient Boosting Regression benefits from full compatibility with Scikit-learn's ecosystem, which simplifies preprocessing, validation, and interpretability workflows.

In comparison with LightGBM, both models employ histogram-based learning for speed and scalability. However, LightGBM utilizes a leaf-wise tree growth strategy with depth constraints, often resulting in deeper trees and faster loss reduction but higher susceptibility to overfitting, especially on smaller datasets. HistGradient Boosting Regression, by contrast, grows trees level-wise rather than leaf-wise, which is a key distinction that mitigates overfitting risks in small or noisy datasets. This level-wise approach ensures balanced growth across the tree structure and contributes to more stable generalization. It is also easier to configure due to fewer sensitive hyperparameters. This characteristic makes it attractive in practical scenarios where computational resources or tuning bandwidth are limited.

However, a known limitation of HistGradient Boosting Regression is its handling of sparse data in high-dimensional feature spaces. Unlike LightGBM, which employs an advanced technique known as Exclusive Feature Bundling (EFB) to reduce feature dimensionality by combining mutually exclusive sparse features, of HistGradient Boosting Regression does not implement any built-in bundling strategy. As a result, its performance on highly sparse datasets may not be as optimized, especially in domains like text mining or recommender systems where EFB can significantly compress feature space. This trade-off highlights HGB's emphasis on simplicity and memory efficiency at the cost of aggressive sparse feature handling. Given that our dataset is of moderate size and largely numerically encoded, the limitations of HistGradient Boosting Regression in handling extreme sparsity or categorical complexity did not pose a significant disadvantage in this context.

These structural advantages are reflected in our empirical results in Section 4, where HistGradient Boosting Regression achieved competitive or superior performance across multiple evaluation metrics, including RMSE and R^2. Its consistent accuracy, fast training time, and efficient memory usage contributed to its top-tier ranking in both predictive accuracy and computational efficiency. Moreover, its compatibility with SHAP-based explainability tools further strengthens its interpretability in real-world deployment. In summary, HistGradient Boosting Regression strikes a pragmatic balance between performance, ease of use, and scalability. While it may lack some of the fine-grained controls offered by other models, including advanced sparse feature handling or categorical encoding, its robust out-of-the-box performance and lower tuning complexity make it a compelling choice for regression problems involving high-dimensional, numerical, or partially missing data.

5.2. Comparative analysis with recent studies

Recent advancements in construction cost prediction have increasingly explored the application of machine learning techniques. However, as summarized in Table 4, the majority of these studies remain limited in scope, often focusing on a narrow subset of models, such as RFs, Deep Neural Networks, or basic ensemble methods, and primarily evaluating their performance using conventional accuracy metrics like the *R*2, MSE, MAE, or MAPE. While these metrics are important, they only offer a partial picture of a model's practical utility. In high-stakes construction environments, especially for large infrastructure or public-sector projects, decision-makers require not only accurate forecasts but also reliable measures of uncertainty and transparent explanations of how cost estimates are derived.

For instance, Wang et al. [52] employed Deep Neural Networks to predict the construction costs of public-school projects in Hong Kong and applied SHAP to improve model interpretability. While this approach partially addressed the explainability gap, it fell short of providing any uncertainty quantification, leaving stakeholders without insight into the confidence level or risk margin associated with each prediction. Similarly, the study by Huang and Hsieh [53], which modeled cost data from BIM projects using RF and Linear Regression, focused exclusively on predictive accuracy, neglecting both interpretability and predictive uncertainty. Alshboul et al. [54,55] offered a broader comparative study involving XGBoost, LightGBM, RF, and Deep Neural Networks to estimate costs of green building projects. However,

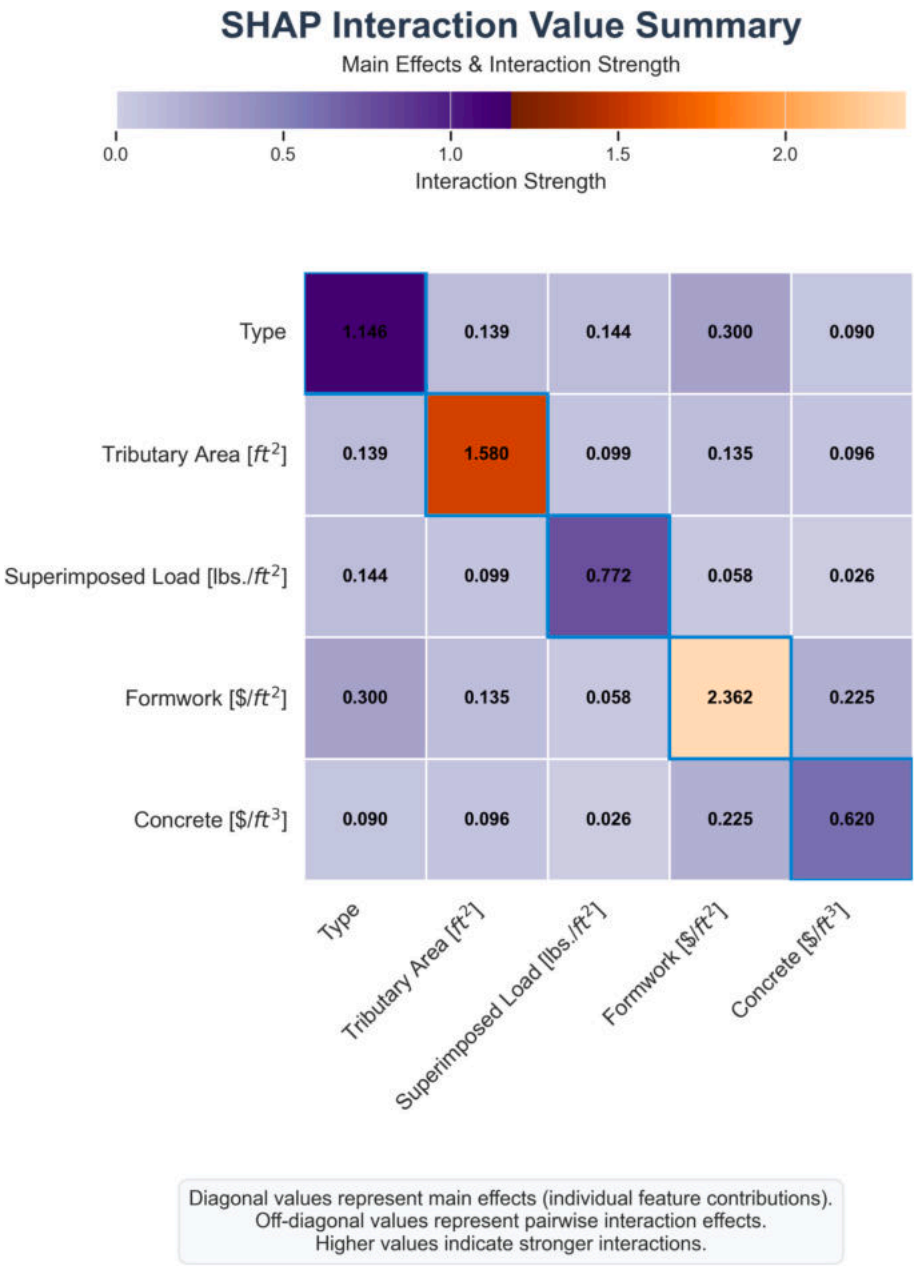


Fig. 8. SHAP interaction value heatmap for the HistGradientBoosting Regression model.

Table 3
Runtime Performance Comparison of Ten Machine Learning Models Based on Training and Prediction Times (in Seconds).

Machine Learning Model	Training Time (s)	Training Set Prediction Time (s)	Testing Set Prediction Time (s)
Ridge Regression	0.0010	0.0010	0.0000
Lasso Regression	0.0352	0.0010	0.0000
Elastic Net Regression	0.0302	0.0016	0.0000
KNN Regression	0.1727	0.0066	0.0031
Extra Trees Regression	196.2884	0.3025	0.1476
Gradient Boosting Regression	126.2696	0.0339	0.0146
AdaBoost Regression	109.6031	0.3069	0.1638
XGBoost Regression	369.2867	0.0121	0.0035
CatBoost Regression	12.2809	0.0412	0.0085
HistGradientBoosting Regression	82.0437	0.0680	0.0363

their analysis also lacked confidence intervals and explainable AI techniques, limiting the transparency and risk-awareness of their predictive outputs.

Compared to these existing studies, the current research introduces a holistic and methodologically rigorous evaluation framework that addresses these key deficiencies. First, the study expands the range of machine learning models beyond typical selections by including ten advanced algorithms, ranging from regularized linear regressors (e.g., Ridge, Lasso, Elastic Net) to distance-based (KNN), ensemble bagging (Extra Trees), and a comprehensive suite of boosting models (Gradient Boosting, AdaBoost, XGBoost, CatBoost, and HistGradientBoosting Regression). This diverse portfolio allows for a more nuanced analysis of how different algorithmic families perform under a consistent dataset and evaluation protocol.

Second, the study systematically incorporates confidence interval analysis, enabling the quantification of prediction uncertainty at both the global and individual levels. This is crucial in construction contexts

Table 4
Comparative Summary of Recent Studies on Construction Cost Prediction in Terms of Model Diversity, Evaluation Metrics, Uncertainty Quantification, and Interpretability.

Authors	Year	Dataset Description	Machine Learning Models Used	Evaluation Metrics	Uncertainty Quantification	Interpretability Techniques
Wang et al. [52]	2022	98 public school projects in Hong Kong	Deep Neural Network	R^2	No	SHAP
Huang & Hsieh [53]	2020	19 BIM projects from a Taiwanese firm	Random Forest, Linear Regression	MSE, MAE	No	No
Alshboul et al. [54]	2021	3,578 green projects in North America	LightGBM, XGBoost	RMSE, MAE, MAPE, R^2	No	No
Alshboul et al. [55]	2022	283 LEED-certified green buildings	XGBoost, Deep Neural Network, Random Forest	RMSE, MAE, MAPE, R^2	No	No
This Study	2025	4,477 RSMeans construction cost entries (1998 to 2018)	Ten models including Ridge Regression, Lasso Regression, Elastic Net, KNN, Extra Trees Regression, Gradient Boosting, AdaBoost, XGBoost, CatBoost, Histogram-based Gradient Boosting	RMSE, MBE, R^2, Residual Plot	Yes	SHAP

where estimates must be accompanied by margins of error to support contingency planning, budget allocation, and contract negotiation. Unlike prior work, this study does not treat uncertainty as an afterthought but rather integrates it directly into the performance evaluation pipeline.

Third, model transparency is ensured through the use of SHAP, a state-of-the-art explainable AI method that quantifies the contribution of each input feature to the final prediction. This component addresses a persistent barrier in industry adoption of machine learning, i.e., the so-called "black-box" problem, by offering interpretable justifications behind cost estimations. By visualizing how factors such as floor area, material type, or region impact the predicted cost, SHAP aids practitioners in validating model behavior and justifying recommendations to stakeholders.

Furthermore, this study employs a robust multi-perspective evaluation strategy that includes residual analysis, R^2, RMSE, and MBE across all models. This enables a deeper understanding of not only model accuracy but also variance patterns, over- or under-estimation tendencies, and generalization performance. These insights go beyond those offered by single-metric evaluations found in most existing literature, allowing for more informed model selection and deployment readiness.

It is noteworthy that while prior studies have made meaningful contributions in terms of accuracy benchmarking, they have not yet converged on a fully integrated framework that combines accuracy, uncertainty quantification, and explainability. This study bridges that gap, establishing a comprehensive pipeline that better reflects the multifaceted demands of real-world construction cost prediction. As such, it contributes both methodologically and practically to the advancement of data-driven cost estimation in the construction domain.

6. Conclusions

This paper presents a comprehensive evaluation of ten machine learning models applied to construction cost prediction, leveraging the RSMeans dataset of 4,477 data points. Advanced ensemble methods, including HistGradientBoosting, XGBoost, CatBoost, and Extra Trees Regression, emerged as the most effective approaches, significantly outperforming traditional linear models (Ridge Regression, Lasso Regression, and Elastic Net) and non-parametric methods (KNN Regression). Among these ensemble-based methods, HistGradientBoosting Regression demonstrated exceptional predictive accuracy, achieving an R^2 of 0.988 and an RMSE of 0.477, with robust generalization across unseen data.

Beyond standard metrics, this study introduced confidence intervals as a key element to evaluate the reliability of model predictions. The visualization of confidence intervals highlighted the superior consistency of ensemble methods, particularly HistGradientBoosting Regression, in minimizing prediction uncertainty. These intervals provided valuable insights into the range of potential costs, enabling more informed decision-making for construction professionals. Furthermore, the study also leveraged SHAP analyses to enhance model interpretability by quantifying feature contributions to predictions. SHAP analyses revealed the dominant role of features like "Formwork" and "Tributary Area" in influencing construction costs, while identifying the nuanced impact of secondary features. This dual focus on reliability and transparency bridges the gap between advanced machine learning models and their real-world usability.

By integrating predictive accuracy with interpretability and uncertainty quantification, this study contributes a practical framework for cost estimation in high-stakes construction environments. The results directly support critical applications such as public infrastructure budgeting, private development planning, and automated cost estimation systems. Such capabilities are especially valuable in an era of rising construction costs and increased demand for accountability in both public and private sector projects. Given the strong performance and explainability of models like HistGradientBoosting, the proposed approach also shows promise for integration into digital construction tools, including commercial cost estimation software and digital twin platforms. These integrations would enable practitioners, such as cost estimators, engineers, and project managers, to leverage data-driven insights through familiar interfaces, thus facilitating wider adoption across the construction industry.

In conclusion, the integration of confidence intervals and SHAP analyses alongside traditional metrics underscores the strengths of ensemble methods like HistGradientBoosting, XGBoost, and CatBoost. These findings have significant implications for the construction industry, where precise and interpretable cost estimation is crucial for budgeting, resource allocation, and risk mitigation. Looking forward, future research could explore user-centric design of web-based tools that embed these predictive engines into intuitive interfaces suitable for both technical and non-technical stakeholders. Further studies could also investigate the inclusion of domain-specific features, hybrid modeling strategies, and broader deployment in construction-related applications.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT in order to improve readability and language of manuscript. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

CRediT authorship contribution statement

Lifei Chen: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Changyong Xu:** Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Wei Hong Lim:** Writing – original draft, Supervision, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Abhishek Sharma:** Writing – review & editing, Validation, Methodology, Formal analysis, Data curation. **Sew Sun Tiang:** Writing – review & editing, Software, Investigation, Data curation, Conceptualization. **Kim Soon Chong:** Writing – review & editing, Validation, Methodology, Investigation, Formal analysis, Data curation. **El-Sayed M. El-kenawy:** Writing – review & editing, Validation, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Amel Ali Alhussan:** Writing – review & editing, Validation, Investigation, Funding acquisition. **Marwa M. Eid:** Writing – review & editing, Validation, Investigation, Formal analysis, Data curation. **Doaa Sami Khafaga:** Data curation, Formal analysis, Funding acquisition, Investigation, Project administration, Validation, Writing – review & editing.

Declaration of competing interest

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

Acknowledgement

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R308), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

References

[1] Wang, Yali, Jian Zuo, Min Pan, Bocun Tu, Rui-Dong Chang, Shicheng Liu, Feng Xiong, and Na Dong. “Cost Prediction of Building Projects Using the Novel Hybrid RA-ANN Model.” Eng., Construct. Architect. Manage., January 24, 2023. doi: 10.1108/ecam-07-2022-0666.

[2] G.C. Eads, M.R. Kiefer, S. Mehndiratta, Short-term delay mitigation strategies for San Francisco international airport, Transport. Res. Record: J. Transport. Res. Board 1744 (1) (January 2001) 44–51, https://doi.org/10.3141/1744-06.

[3] Eco-Sustainable House / Djuric Tardio Architectes“ 18 Apr 2012. ArchDaily. Accessed 3 Jan 2025. .

[4] Alajmi, Ali F. “Implementing the Integrated Design Process (IDP) to Design, Construct and Monitor an Eco-House in Hot Climate.” Int. J. Sustain. Eng. 14, no. 4 (June 20, 2021), pp. 630 646. doi:10.1080/19397038.2021.1934183.

[5] D. Blomberg, P. Cotellesso, W. Sitzabee, A.E. Thal, Discovery of internal and external factors causing military construction cost premiums, J. Constr. Eng. Manag. 140 (3) (March 2014) 04013060, https://doi.org/10.1061/(asce)co.1943-7862.0000810.

[6] Ahn, Seungjun, Samin Shokri, SangHyun Lee, Carl T. Haas, and Ralph C. G. Haas. “Exploratory Study on the Effectiveness of Interface-Management Practices in Dealing with Project Complexity in Large-Scale Engineering and Construction Projects.” J. Manage. Eng. 33, no. 2 (March 2017): 04016039. doi:10.1061/(asce)me.1943-5479.0000488.

[7] Elmousalami, Haytham H. “Artificial Intelligence and Parametric Construction Cost Estimate Modeling: State-of-The-Art Review.” J. Construct. Eng. Manage. 146, no. 1 (January 2020): 03119008. doi:10.1061/(asce)co.1943-7862.0001678.

[8] W Timothy Weaver, and Syracuse University. Research Corporation. Educational Policy Research Center. The Delphi Method. Syracuse, N.Y.: Educational Policy Research Center, Syracuse University Research Corp, 1970.

[9] Cooper, Robin, and Regine Slagmulder. Target Costing and Value Engineering. Portland, Or.: Productivity Press ; Montvale, N.J, 1997.

[10] Everaert, Patricia, Stijn Loosveld, Tom Van Acker, Marijke Schollier, and Gerrit Sarens. “Characteristics of Target Costing: Theoretical and Field Study Perspectives.” Qualit. Res. Account. Manage. 3, no. 3 (September 2006): pp. 236–263. Doi:10.1108/11766090610705425.

[11] A.C. Narváez, A.P. Fernández, M.O. Mateo, P.B. Pérez, Integration of cost and work breakdown structures in the management of construction projects, Appl. Sci. 10 (4) (2020) 1386, https://doi.org/10.3390/app10041386.

[12] E. Ikpe, F. Hammon, D. Oloke, Cost-benefit analysis for accident prevention in construction projects, J. Constr. Eng. Manag. 138 (8) (August 2012) 991–998, https://doi.org/10.1061/(asce)co.1943-7862.0000496.

[13] A. Ishikawa, et al., The max-min delphi method and Fuzzy Delphi method via fuzzy integration, Fuzzy Set. Syst. 55 (3) (May 1993) 241–253, https://doi.org/10.1016/0165-0114(93)90251-c.

[14] Xiaojuan Li, Chen Wang, and Ali Alashwal, “Case Study on BIM and Value Engineering Integration for Construction Cost Control,” ed. Jian Ji, Adv. Civil Eng. 2021 (February 4, 2021): 1–13, doi:10.1155/2021/8849303.

[15] Dalenogare, Lucas Santos, Guilherme Brittes Benitez, Néstor Fabián Ayala, and Alejandro Germán Frank. “The Expected Contribution of Industry 4.0 Technologies for Industrial Performance.” Int. J. Product. Econom. 204, no. 1 (October 2018): 383–94. doi:10.1016/j.ijpe.2018.08.019.

[16] Khan Md. Hasib, Nurul Akter Towhid, Kazi Omar Faruk, Jubayer Al Mahmud, & Mridha, M. F. (2023). Strategies for enhancing the performance of news article classification in Bangla: handling imbalance and interpretation. Eng. Appl. Artif. Intellig., 125, pp. 106688–106688. Doi: 10.1016/j.engappai.2023.106688.

[17] K. Rasheed, A. Qayyum, M. Ghaly, A. Al-Fuqaha, A. Razi, J. Qadir, Explainable, trustworthy, and ethical machine learning for healthcare: a survey, Comput. Biol. Med. 149 (October 2022) 106043, https://doi.org/10.1016/j.compbiomed.2022.106043.

[18] Cai, Bingyu, Mahmud Iwan Solihin, Chaoran Chen, Xujin Lu, Zhigang Xie, and Defu Yang. “Modeling of a Nonlinear Coupled Compliant Mechanism via Developed Kinematics-Integrated Neural Network Algorithm.” Microsyst. Technol., August 1, 2024. doi:10.1007/s00542-024-05733-9.

[19] Ze Han Ang et al., “Development of an Artificial Intelligent Approach in Adapting the Characteristic of Polynomial Trajectory Planning for Robot Manipulator,” Int. J. Mechan. Eng. Robot. Res., January 1, 2020, pp. 408–414, .

[20] Dheya Galal Abdullah Al-Sanabani et al., “Development of Non-Destructive Mango Assessment Using Handheld Spectroscopy and Machine Learning Regression,” J. Phys.: Conf. Ser. 1367, no. 1 (November 1, 2019) pp. 012030, Doi:10.1088/1742-6596/1367/1/012030.

[21] Mohamed Yasser Mohamed et al., “Food Powders Classification Using Handheld Near-Infrared Spectroscopy and Support Vector Machine,” J. Phys.: Conf. Ser.es 1367, no. 1 (November 1, 2019) pp. 012029, Doi: 10.1088/1742-6596/1367/1/012029.

[22] Gasim Hayder, Mahmud Iwan Solihin, Hauwa Mohammed Mustafa, “Modelling of River Flow Using Particle Swarm Optimized Cascade-Forward Neural Networks: A Case Study of Kelantan River in Malaysia,” Appl. Sci. 10, no. 23 (December 4, 2020) pp. 8670, Doi: 10.3390/app10238670.

[23] Gasim Hayder, Mahmud Iwan Solihin, M. R. N. Najwa, “Multi-Step-Ahead Prediction of River Flow Using NARX Neural Networks and Deep Learning LSTM,” H2Open J. 5, no. 1 (January 25, 2022) pp. 43–60, doi: 10.2166/h2oj.2022.134.

[24] Tarek Berghout et al., “A Neural Network Weights Initialization Approach for Diagnosing Real Aircraft Engine Inter-Shaft Bearing Faults,” Machines 11, no. 12 (December 14, 2023): 1089–89, doi: 10.3390/machines11121089.

[25] Tarek Berghout et al., “Federated Learning for Condition Monitoring of Industrial Processes: A Review on Fault Diagnosis Methods, Challenges, and Prospects,” Electronics 12, no. 1 (December 29, 2022): 158, doi: 10.3390/electronics12010158.

[26] Moath Alrifaey et al., “Hybrid Deep Learning Model for Fault Detection and Classification of Grid-Connected Photovoltaic System,” IEEE Access 10 (January 1, 2022): pp. 13852–13869, doi: 10.1109/access.2022.3140287.

[27] Moath Alrifaey, Wei Hong Lim, and Chun Kit Ang, “A Novel Deep Learning Framework Based RNN-SAE for Fault Detection of Electrical Gas Generator,” IEEE Access 9 (2021): 21433–42, doi:10.1109/access.2021.3055427.

[28] Chen, Chi, Yunxing Zuo, Weike Ye, Xiangguo Li, Zhi Deng, Shyue Ping Ong. A Critical Review of Machine Learning of Energy Materials. Adv. Energy Mater. 10, no. 8 (January 29, 2020): 1903242. doi:10.1002/aenm.201903242.

[29] Davis, Peter, Fayeem Aziz, Mohammad Tanvi Newaz, Willy Sher, Laura Simon. “The Classification of Construction Waste Material Using a Deep Convolutional Neural Network.” Automat. Construct. 122 (February 2021) pp. 103481. doi: 10.1016/j.autcon.2020.103481.

[30] M.J. Esfandiari, G.S. Urgessa, Progressive collapse design of reinforced concrete frames using structural optimization and machine learning, Structures 28 (December 2020) 1252–1264, https://doi.org/10.1016/j.istruc.2020.09.039.

[31] Flah, Majdi, Itzel Nunez, Wassim Ben Chaabene, and Moncef L. Nehdi. “Machine Learning Algorithms in Civil Structural Health Monitoring: A Systematic Review.” Arch. Computat. Method. Eng. 28, no. 4 (July 29, 2020) pp. 2621–2643. Doi: 10.1007/s11831-020-09471-9.

[32] Mahmud, None, Chan Jin Yuan, Wan Siu Hong, Liew Phing Pui, Ang Chun Kit, Wafa Hossain, and Affiani Machmudah. “Spectroscopy Data calibration using stacked ensemble machine learning.” IIUM Eng. J. 25, no. 1 (January 1, 2024) pp. 208–224. doi:10.31436/iiumej.v25i1.2796.

[33] H.G. Melhem, Y. Cheng, Prediction of remaining service life of bridge decks using machine learning, J. Comput. Civ. Eng. 17 (1) (January 2003) 1–9, https://doi.org/10.1061/(asce)0887-3801(2003)17:1(1).

[34] Ze Han Ang, Chun Kit Ang, Wei Hong Lim, Lih Jiun Yu, Mahmud Iwan Solihin. “Development of an Artificial Intelligent Approach in Adapting the Characteristic of Polynomial Trajectory Planning for Robot Manipulator.” Int. J. Mechan. Eng. Robot. Res., January 1, 2020, pp. 408–414. doi:10.18178/ijmerr.9.3.408-414.

[35] N.V. Dharwadkar, S.S. Arage, Prediction and estimation of civil construction cost using linear regression and neural network, Int. J. Intellig. Syst. Design Comput. 2 (1) (2018) 28, https://doi.org/10.1504/ijisdc.2018.092554.

[36] Yang, Seung-Won, Seong-Wan Moon, Hangyeol Jang, Seungyeon Choo, and Sung-Ah Kim. “Parametric Method and Building Information Modeling-Based Cost Estimation Model for Construction Cost Prediction in Architectural Planning.” Appl. Sci. 12, no. 19 (September 23, 2022): 9553. doi:10.3390/app12199553.

[37] Hai, Chen. "Construction and Application of Multiple Linear Regression Model for Construction Project Cost," June 1, 2021. doi:10.1109/aeis53850.2021.00017.
[38] D.J. Lowe, M.W. Emsley, A. Harding, Predicting construction cost using multiple regression techniques, J. Constr. Eng. Manag. 132 (7) (July 2006) 750–778, https://doi.org/10.1061/(asce)0733-9364(2006)132:7(750).
[39] Jafarzadeh, R., J. M. Ingham, K. Q. Walsh, N. Hassani, G. R. Ghodrati Amiri. Using Statistical Regression Analysis to Establish Construction Cost Models for Seismic Retrofit of Confined Masonry Buildings. J. Construct. Eng. Manage. 141, no. 5 (May 2015): 04014098. doi:10.1061/(asce)co.1943-7862.0000968.
[40] Petroutsatou, Kleopatra, Sergios Lambropoulos. Road tunnels construction cost estimation: a structural equation model development and comparison Operat. Res. 10, no. 2 (July 25, 2009): 163–73. doi:10.1007/s12351-009-0061-7.
[41] S.M. Shahandashti, B. Ashuri, Highway Construction cost forecasting using Vector Error Correction Models, J. Manag. Eng. 32 (2) (March 2016) 04015040, https://doi.org/10.1061/(asce)me.1943-5479.0000404.
[42] Zhang, Chi, Jinsong Zhu, Teng Shi, Xingtian Li. "Influence Line Estimation of Bridge Based on Elastic Net and Vehicle-Induced Response." Measurement 202 (October 1, 2022): pp. 111883–83. Doi:10.1016/j.measurement.2022.111883.
[43] M.W. Emsley, D.J. Lowe, A. Roy Duff, A. Harding, A. Hickson, Data Modelling and the Application of a Neural Network Approach to the Prediction of Total Construction costs, Constr. Manag. Econ. 20 (6) (September 2002) 465–472, https://doi.org/10.1080/01446190210151050.
[44] M.-Y. Cheng, H.-C. Tsai, W.-S. Hsieh, Web-based Conceptual cost estimates for Construction Projects using Evolutionary Fuzzy Neural Inference Model, Autom. Constr. 18 (2) (March 2009) 164–172, https://doi.org/10.1016/j.autcon.2008.07.001.
[45] P. Silvana, "Construction costs forecasting: Comparison of the Accuracy of Linear Regression and support Vector Machine Models." Tehnicki Vjesnik - Technical, Gazette 24, no. 5 (October 2017), https://doi.org/10.17559/tv-20150116001543.
[46] G.-H. Kim, J.-M. Shin, S. Kim, Y. Shin, Comparison of School Building Construction costs Estimation Methods using Regression Analysis, Neural Network, and support Vector Machine, J. Build. Construct. Plann. Res. 01 (01) (2013) 1–7, https://doi.org/10.4236/jbcpr.2013.11001.
[47] Du, Zeyan, and Binyong Li. "Construction Project Cost Estimation Based on Improved BP Neural Network." 2017 International Conference on Smart Grid and Electrical Automation (ICSGEA), May 1, 2017. doi:10.1109/icsgea.2017.162.
[48] Ahn, Joseph, Sae-Hyun Ji, Sung Jin Ahn, Moonseo Park, Hyun-Soo Lee, Nahyun Kwon, Eul-Bum Lee, and Yonggu Kim. "Performance Evaluation of Normalization-Based CBR Models for Improving Construction Cost Estimation." Automation in Construction 119 (November 2020): 103329. .
[49] Xiao, Xue, Martin Skitmore, Weixin Yao, Yousuf Ali. "Improving Robustness of Case-Based Reasoning for Early-Stage Construction Cost Estimation." Automa. Construct. 151 (July 1, 2023): 104777. doi:10.1016/j.autcon.2023.104777.
[50] Simić, Nevena, Nenad Ivanišević, Đorđe Nedeljković, Aleksandar Senić, Zoran Stojadinović, and Marija Ivanović. "Early Highway Construction Cost Estimation: Selection of Key Cost Drivers." Sustainability 15, no. 6 (March 22, 2023): 5584. doi:10.3390/su15065584.
[51] Yun, Seokheon. "Performance Analysis of Construction Cost Prediction Using Neural Network for Multioutput Regression." Appl. Sci. 12, no. 19 (September 24, 2022): 9592.doi:10.3390/app12199592.
[52] Wang, Ran, Vahid Asghari, Clara Man Cheung, Shu-Chien Hsu, Chia-Jung Lee. Assessing Effects of Economic Factors on Construction Cost Estimation Using Deep Neural Networks. Automat. Construct. pp. 134 (February 1, 2022): 104080. doi:10.1016/j.autcon.2021.104080.
[53] C.-H. Huang, S.-H. Hsieh, Predicting BIM Labor cost with Random Forest and simple Linear Regression, Autom. Constr. 118 (October 2020) 103280, https://doi.org/10.1016/j.autcon.2020.103280.
[54] A. Shehadeh, O. Alshboul, R.E.A. Mamlook, O. Hamedat, Machine Learning Models for predicting the Residual Value of Heavy Construction Equipment: an Evaluation of Modified Decision tree, LightGBM, and XGBoost Regression, Autom. Constr. 129 (September 2021) 103827, https://doi.org/10.1016/j.autcon.2021.103827.
[55] Alshboul, Odey, Ali Shehadeh, Ghassan Almasabha, and Ali Saeed Almuflih. "Extreme Gradient Boosting-Based Machine Learning Approach for Green Building Cost Prediction." Sustainability 14, no. 11 (May 29, 2022): 6651. doi:10.3390/su14116651.
[56] G.-H. Kim, J.-E. Yoon, S.-H. An, H.-H. Cho, K.-I. Kang, Neural Network Model Incorporating a Genetic Algorithm in estimating Construction costs, Build. Environ. 39 (11) (November 2004) 1333–1340, https://doi.org/10.1016/j.buildenv.2004.03.009.
[57] M.-Y. Cheng, N.-D. Hoang, Wu. Yu-Wei, Hybrid Intelligence Approach based on LS-SVM and Differential Evolution for Construction cost Index Estimation: a Taiwan Case Study, Autom. Constr. 35 (November 2013) 306–313, https://doi.org/10.1016/j.autcon.2013.05.018.
[58] Alshboul, Odey, Ali Shehadeh, Ghassan Almasabha, Rabia Emhamed Al Mamlook, and Ali Saeed Almuflih. "Evaluating the Impact of External Support on Green Building Construction Cost: A Hybrid Mathematical and Machine Learning Prediction Approach." Buildings 12, no. 8 (August 16, 2022): 1256. doi:10.3390/buildings12081256.
[59] Ali, Zainab Hasan, Abbas M. Burhan, Murizah Kassim, and Zainab Al-Khafaji. "Developing an Integrative Data Intelligence Model for Construction Cost Estimation." Edited by Haitham Abdulmohsin Afan. Complexity 2022 (September 29, 2022): 1–18. doi:10.1155/2022/4285328.
[60] Cheng, Min-Yuan, Nhat-Duc Hoang. "Interval estimation of construction cost at completion using least squares support vector machine. J. Civil Eng. Manage. 20, no. 2 (March 10, 2014) pp. 223–236. doi: 10.3846/13923730.2013.801891.
[61] A.E. Hoerl, R.W. Kennard, Ridge regression: applications to nonorthogonal problems, Technometrics 12 (1) (February 1970) 69–82, https://doi.org/10.1080/00401706.1970.10488635.
[62] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Roy. Stat. Soc.: Ser. B (Methodol.) 58 (1) (January 1996) 267–288, https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.
[63] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B (Stat Methodol.) 67 (2) (April 2005) 301–320, https://doi.org/10.1111/j.1467-9868.2005.00503.x.
[64] T. Cover, Estimation by the nearest neighbor rule, IEEE Trans. Inf. Theory 14 (1) (January 1968) 50–55, https://doi.org/10.1109/tit.1968.1054098.
[65] Pierre Geurts, Damien Ernst, Louis Wehenkel, "Extremely Randomized Trees," Machine Learn. 63, no. 1 (March 2, 2006): 3–42, doi:10.1007/s10994-006-6226-1.
[66] J. Friedman, T. Hastie, R. Tibshirani, Additive Logistic Regression: a Statistical View of Boosting (with Discussion and a Rejoinder by the Authors), Ann. Statist. 28 (2) (April 2000) 337–407, https://doi.org/10.1214/aos/1016218223.
[67] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139, https://doi.org/10.1006/jcss.1997.1504.
[68] Tianqi Chen, Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, 2016, pp. 785–794, doi:10.1145/2939672.2939785.
[69] Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin, "CatBoost: Gradient Boosting with Categorical Features Support," arXiv.org, October 24, 2018, doi:10.48550/arXiv.1810.11363.
[70] Aleksei Guryanov, "Histogram-Based Algorithm for Building Gradient Boosting Ensembles of Piecewise Linear Decision Trees," Lect. Not. Comput. Sci., July 17, 2019, 39–50, doi:10.1007/978-3-030-37334-4_4.
[71] Scott M Lundberg, Su-In Lee, "A Unified Approach to Interpreting Model Predictions," Neural Information Processing Systems (Curran Associates, Inc., 2017), https://papers.nips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.
[72] Scikit-learn Benchmarks. (2023). Scikit-learn benchmarks. https://scikit-learn.org/scikit-learn-benchmarks/#regressions?sort=3&dir=desc.