Contents lists available at **SCCE**

Journal of Soft Computing in Civil Engineering

Journal homepage: www.jsoftcivil.com

Comparison of Various Machine Learning Models for Estimating Construction Projects Sales Valuation Using Economic Variables and Indices

Yazan Alzubi[1*]

1. Associate Professor, Civil Engineering Department, Faculty of Engineering Technology Al-Balqa Applied University, 11134 Amman, Jordan
Corresponding author: *yazan.alzubi@bau.edu.jo*

https://doi.org/10.22115/SCCE.2023.364221.1536

ARTICLE INFO

Article history:
Received: 02 October 2022
Revised: 25 April 2023
Accepted: 19 May 2023

Keywords:
Real estate valuation;
Construction project sales valuation;
Economic variables and indices;
Machine learning;
Rapid estimation.

ABSTRACT

The capability of various machine learning techniques in predicting construction project profit in residential buildings using a combination of economic variables and indices (EV&Is) and physical and financial variables (P&F) as input variables remain uncertain. Although recent studies have primarily focused on identifying the factors influencing the sales of construction projects due to their significant short-term impact on a country's economy, the prediction of these parameters is crucial for ensuring project sustainability. While techniques such as regression and artificial neural networks have been utilized to estimate construction project sales, limited research has been conducted in this area. The application of machine learning techniques presents several advantages over conventional methods, including reductions in cost, time, and effort. Therefore, this study aims to predict the sales valuation of construction projects using various machine learning approaches, incorporating different EV&Is and P&F as input features for these models and subsequently generating the sales valuation as the output. This research will undertake a comparative analysis to investigate the efficiency of the different machine learning models, identifying the most effective approach for estimating the sales valuation of construction projects. By leveraging machine learning techniques, it is anticipated that the accuracy of sales valuation predictions will be enhanced, ultimately resulting in more sustainable and successful construction projects. In general, the findings of this research reveal that the extremely randomized trees model delivers the best performance, while the decision tree model exhibits the least satisfactory performance in predicting the sales valuation of construction projects.

How to cite this article: Alzubi Y. Comparison of various machine learning models for estimating construction projects sales valuation using economic variables and indices. J Soft Comput Civ Eng 2024;8(1):1-32. https://doi.org/10.22115/scce.2023.364221.1536

1. Introduction

Different investment decisions are associated with high returns, such as real estate, which is considered one of the most profitable and sustainable choices [1,2]. The evaluation of real estate in any region is based on the assessment of multiple factors, including the ongoing condition of the economy and the value of money [3]. In addition to that, the prevalence of the application of real estate is significantly governed by the expansion of the population and the prompt urbanization due to the necessity of investigating the obtainability, supply, and demand of housing in order to provide the requirements caused by the growth of urbanization and population [4,5]. Hence, the need for adequate and accurate housing price estimation is crucial for various aspects, including demand, development, investment, evaluations, and tax inspections of housing prices [6,7]. The existence of real estate valuation in many aspects caused the development of diverse methods for forecasting fluctuating housing prices [8–10]. To overcome the undesirability of this type of inaccurate prediction, Ibisola et al. [11] suggested the need for precise, safe, and objective identification of the real estate values for the social economy of any nation. As a result, the interest in predicting housing prices has increased remarkably over the last decades where different estimation models were developed to close the information gap, enhance the performance and effectiveness of the real estate market, and establish specific standards and clear processes for providing far better comprehension of the complex mechanisms of the housing market [12]. Nonetheless, establishing a general model for predicting housing prices is still challenging or even unachievable to the difficulty in determining the interaction between the social, economic, and financial parameters [13]. The recent use of various computational methods and optimized algorithms, such as mathematical and automated valuation models in the real estate industry, to predict prices have increased considerably [14–17]. In general, the applications of regression, stochastic, and neural network approaches in estimating housing prices have recently gained popularity [18,19]. Rafiei and Adeli [20] studied using a neural network, particularly the Deep Belief Restricted Boltzmann Machine (DRBM) with a dataset with a sample size of 500 training and testing points to estimate the real estate sale price evaluation. The limitation of the study is the narrow evaluation of the model in one locality; hence, the model needs to be evaluated in other localities. Moreover, Kim et al. [21] predicted the construction cost of residential buildings using applied back-propagation neural networks (BPNNs) incorporating genetic algorithms (GAs) with collected data for 530 residential buildings. The limitations of the study include the computational complexity and overfitting of the data. However, the capability of various machine learning in predicting construction project profit in residential buildings using various economic variables and indices (EV&Is) as well as physical and financial variables (P&F) as input variables, are still unclear. Moreover, the main focus of recent studies has been on identifying the parameters that impact the sales of construction projects. This is because these parameters significantly influence any country's economy in the short term. Predicting these parameters is essential for ensuring the sustainability of the project. While techniques such as regression and ANN have been used to estimate construction project sales, very few studies have been conducted on this topic. The use of machine learning techniques has several advantages over traditional methods, including cost, time, and effort reduction. Therefore, this paper aims to predict the sales valuation of construction projects using different machine learning approaches. Various economic variables and indices will be used as inputs to the machine learning models to generate the sales valuation as the output. The paper will compare and investigate the efficiency of these machine learning models to determine the most efficient one for estimating the sales valuation of construction projects. By using machine learning techniques, the accuracy of sales valuation predictions is expected to increase, ultimately leading to more sustainable and successful construction projects. The structure of the paper is divided as follows: Section 2 provides a literature review of the paper; Section 3 discusses the research methodology of the study; Section 4 provides the results and discussions of the study; Section 5 provides the main conclusions of the paper.

2. Literature review

Machine learning is a subfield of artificial intelligence that focuses on developing algorithms and statistical models that can automatically learn and improve from data without being explicitly programmed. The use of machine learning has become increasingly popular in various industries, such as healthcare, finance, and transportation, due to its ability to discover hidden patterns and make accurate predictions from large datasets. There are different types of machine learning techniques, such as supervised, unsupervised, and reinforcement learning, each with its own strengths and weaknesses [22]. Supervised learning involves training the model on labeled data to predict new outputs, while unsupervised learning involves finding hidden patterns without needing labeled data. Reinforcement learning involves learning through trial and error by rewarding the model for making the correct decision. Some of the commonly used machine learning models include linear regression, logistic regression, decision trees, random forests, neural networks, Naive Bayes, K-Nearest Neighbors, gradient boosting, clustering, and dimensionality reduction. However, there are various important and popular machine learning models, including Stochastic Gradient Descent Regression (SGD), Support Vector Regressor (SVR), Decision Tree (DT), Random Forest (RF), Extremely Randomized Trees (ETR), Adaptive Boosting (Ada), Stochastic Gradient Boosting (GB), Histogram-Based Gradient Boosting (HGB), and eXtreme Gradient Boosting (XGB). Stochastic Gradient Descent Regression (SGD) is a popular optimization algorithm used in machine learning for training linear models, particularly in large-scale settings such as Finance, Healthcare, and Robotics. Manogaran and Lopez [23] investigated the implementation of SGD in developing scalable sensor data for healthcare applications, and they concluded that the accuracy is approximately 82%. Moreover, Chakraborty et al. [24] compared six machine learning algorithms and concluded hybrid light gradient boosting and natural gradient boosting models reflected the best performance in predicting construction cost. Support Vector Regressor (SVR) is a type of machine learning algorithm used for regression tasks. Additionally, SVR is commonly used in various applications such as finance, engineering, and environmental science, where it is used for tasks such as predicting stock prices, estimating engineering properties, and modeling environmental variables. A study performed by Raghavendra and Deka [25] regarding the utilization of SVR in the field of hydrology concluded that SVR showed adequate efficiency in different applications, including the prediction of rainfall, water level, and flood. Zahariev et al. [26] studied the relationship between macroeconomic factors and indicators related to bank profitability using Support Vector Regressor. Another machine learning model is the Decision Tree (DT), an algorithm used for classification and regression tasks by constructing a tree-like model of decisions based on the features of the input data. The applications of DT are numerous such as finance, marketing, and healthcare, for tasks such as credit scoring, customer segmentation, and disease diagnosis. DT model was incorporated into medical diagnostics to assist experts in making critical decisions with satisfactory accuracy [27]. Furthermore, Höppner et al. [28] proposed a new churn model named ProfTree, which utilizes advanced DT for optimizing the expected maximum profit measure for customer churn (EMPC). Random Forest (RF) is a machine learning algorithm used for classification and regression tasks. RF works by constructing multiple decision trees, each trained on a randomly selected subset of the input data and features. Various applications of the RF model can be listed, including bioinformatics, marketing, and environmental science. A study investigating the performance of the RF model in forecasting the stock market price was conducted, and the results reflected the remarkable ability of RF [29]. Zhu et al. [30] investigated the performance of the RF algorithm based on fuzzy mathematics to develop a primary investment strategy portfolio for the VR industry. Extremely Randomized Trees (ETR) is an ensemble learning technique used in machine learning for classification, regression, and other tasks. Overall, ETR is a powerful and flexible ensemble learning algorithm that can be used for various machine learning tasks, from classification and regression to outlier and anomaly detection. There are widespread applications of the ETR model, such as image classification, drug discovery, and anomaly detection. Shang et al. [31] studied the performance of ETR in estimating the latent heat flux for evaluating surface water and energy balance. Egwim et al. [32] developed hyperparameter-optimized predictive models, including ETR, and showed adequate performance in estimating construction project delay. Adaptive Boosting, commonly known as AdaBoost, is a machine learning algorithm that is used for classification and regression problems. AdaBoost is an ensemble learning method that combines multiple weak learners to create a strong learner. Furthermore, AdaBoost is a popular algorithm for binary classification problems, and it has been used successfully in a variety of applications such as face detection, text classification, and bioinformatics. One of the applications of the Ada algorithm is the improvement of the detection accuracy of structural members based on sensitivity analysis [33]. Ding et al. [34] use the Ada model to evaluate the sustainability of photovoltaic projects. It concluded that the Ada model is a tool for developing photovoltaic projects. Stochastic Gradient Boosting (GB) is a machine learning algorithm used for regression and classification tasks. It is a variant of Gradient Boosting that introduces additional randomness to the training process, making it more robust to overfitting and better at handling noisy data. Stochastic GB is widely used in online advertising, recommendation systems, and credit risk assessment applications. Guelman [35] evaluated the performance of GB in modeling and predicting loss cost for auto-insurance. Xiao et al. [36] proposed a technique for predicting forward contract costs using the GB model with RMSE equal to 0.1391. Histogram-Based Gradient Boosting (HGB) is a machine learning algorithm that is used for classification and regression tasks. It is a variant of Gradient Boosting that uses histogram-based algorithms for efficient feature binning and split finding. HGB has been used successfully in a variety of applications, such as online advertising, credit scoring, and customer segmentation. In a study performed by Marvin et al. [37] to assess the effectiveness of HGB in detecting the location of water leakage where the results reflected high accuracy in achieving that. Tamim Kashifi and Ahmad [38] studied the effectiveness of the HGB model in estimating the severity of car accidents with superior results and overall accuracy of 82.5%. Lastly, eXtreme Gradient Boosting (XGB) is a popular machine learning algorithm used for regression and classification tasks. It is an extension of Gradient Boosting that includes additional features and optimization techniques, making it more efficient and accurate. Chang et al. [39] conducted a study by deploying XGB to construct a credit risk assessment model for financial institutions where the results showed superior performance. Hou and Qin [40] developed 15 significant parameters related to the growth of Chinese construction enterprises using the XGB algorithm.

3. Research methodology

This section is dedicated to comprehensively describing the various machine learning models where the mathematical equations of these models will be discussed. ANN was commonly implemented to predict housing prices in the real estate industry. However, the performance and efficiency of these machine learning models in estimating the sales profit of real estate projects through comparative assessment of the findings of these models in order to indicate the best one.

3.1. Utilized database

All machine learning models were developed using a previous database by Rafiei and Adeli [20] for various factors impacting the residential construction project profit. Table 1 and Table 2 show details about the P&F and EV&Is factors that were used as input parameters to the machine learning. Physical properties refer to the tangible characteristics of the property, such as its size, location, condition, layout, and amenities. These physical properties can significantly impact the property's value and potential for generating rental income. For example, a property in a desirable location with modern amenities and a functional layout may be more valuable and attractive to potential renters or buyers than a similar property in a less desirable location with outdated features. In addition, financial properties refer to the economic aspects of the property, including its rental income, operating expenses, cash flow, and potential for appreciation. These financial properties are critical for determining the property's potential return on investment and evaluating its performance compared to other investment opportunities. For example, a property with a high rental income and low operating expenses may generate higher cash flow and be more financially attractive than a property with a lower rental income and higher expenses. Accordingly, the project locality variable refers to the geographical area of a real estate project, including the neighborhood, surroundings, and amenities of the area where the project is being developed. As the name implies, the total floor area of a building refers to the sum of the floor area of all its floors, including the ground floor, mezzanine floors, and upper floors. Lot area refers to the total area of a land parcel on which a building or structure is built or planned to be built. Total preliminary estimated construction cost refers to the approximate total cost of constructing a building or structure, as estimated during the preliminary design phase. It includes all the direct and indirect costs associated with the construction project, such as labor, materials, equipment, permits, fees, and overhead expenses. Equivalent preliminary estimated construction cost refers to the estimated cost of constructing a real estate unit with similar specifications and features as the unit being evaluated or compared. This estimation is made using the cost of materials, labor, and other expenses needed to construct a unit with similar features, size, and location. Duration of construction refers to the estimated time required to complete the construction of a building or structure, from the start of the construction phase until its completion. The price of the unit at the beginning of the project refers to the estimated selling price of a real estate unit when it is first introduced to the market or at the start of the project's development phase. This initial price is often based on market research, including factors such as the location, size, features, and amenities of the unit, as well as current market conditions, demand, and competition. On the other hand, real estate units are affected by a range of economic variables and indices that can impact their value, demand, and performance. These variables and indices are typically influenced by broader macroeconomic conditions and trends, such as interest rates, inflation, economic growth, and employment levels. The number of building permits issued refers to the total number of permits issued by a government agency or department authorizing the construction, alteration, or renovation of a building or structure within a given jurisdiction or area. The total subcontractor's amount of contracts refers to the amount paid to subcontractors for work on a construction project during a specified base year. This figure is often used as an economic indicator to measure the construction industry's health and the subcontractor activity level in a particular region or market. The Producer Price Index (PPI) for building materials is an economic indicator that measures the average changes in prices received by domestic producers for their output of building materials. The PPI is calculated by measuring the price changes of a basket of goods and services that are commonly used in the construction industry, such as lumber, cement, steel, and other building materials. The total floor area of building permits issued refers to the total amount of floor space approved for construction or renovation under the building permits issued during a specified time period. This indicator is often used as an economic indicator to measure the level of construction activity in a particular area or region. Cumulative liquidity refers to the total amount of liquid assets that a company has available over a specified time period. Liquid assets are those that can be easily converted into cash, such as cash on hand, short-term investments, and accounts receivable. Private sector investment in new buildings refers to the amount of money invested by private companies or individuals to construct new buildings or to undertake significant renovations of existing buildings. This includes investments in residential, commercial, industrial, and institutional buildings, and can be a key driver of economic growth and development. A land price index for the base year is a measure of the relative change in the prices of land over a specified time period, with the base year typically serving as the reference point. This index is commonly used by real estate professionals, developers, and investors to track trends in land prices and assess the value of real estate investments. A land price index for the base year is a measure of the relative change in the prices of land over a specified time period, with the base year serving as the reference point. It is used to track trends in land prices and assess the value of real estate investments. The number of loans extended by banks in a time resolution refers to the total number of loans approved and disbursed by banks during a specified time period. This can be used as an indicator of the level of credit activity and financial liquidity in the economy. The number of loans extended by banks can be measured in various time resolutions, such as daily, weekly, monthly, quarterly, or yearly. The time resolution chosen depends on the specific purpose for which the data is required. The interest rate for a loan can also be measured in various time resolutions, such as daily, weekly, monthly, quarterly, or yearly. The time resolution chosen depends on the specific purpose for which the data is required. The average construction cost of buildings by the private sector at the time of completion of construction can vary based on various factors, such as the type of building, location, size, materials used, and other construction-related expenses. Therefore, it can be measured in various time resolutions, such as quarterly, annually, or bi-annually. The official exchange rate with respect to dollars refers to the value of one country's currency in relation to the US dollar, as set by the country's government or central bank. This exchange rate is typically used for official transactions such as trade, government payments, and financial reporting. The nonofficial or street market exchange rate with respect to dollars refers to the value of a currency in relation to the US dollar, as determined by market forces such as supply and demand outside of the official foreign exchange market. This type of exchange rate is also known as the black market, parallel, or unofficial exchange rate. The Consumer Price Index (CPI) in the base year is a measure of the average price level of a basket of goods and services consumed by households in a specific year relative to a designated base year. The base year is typically chosen as a reference point for comparison purposes, and the CPI in the base year is set to a value of 100. The CPI of housing, water, fuel, and power in the base year is a measure of the average price level of a basket of goods and services related to housing, water, fuel, and power consumed by households in a specific year relative to a designated base year. The base year is typically chosen as a reference point for comparison purposes, and the CPI in the base year is set to a value of 100. A stock market index is a weighted average of the prices of a basket of stocks that are traded on a stock exchange. Stock market indices are used to track the performance of a particular stock market segment, such as the entire market, a specific industry, or a group of companies with similar characteristics.

3.2. Machine learning models

3.2.1. Stochastic gradient descent regression (SGD)

Stochastic gradient descent (SGD) is one of the most prevalent optimization methods of machine learning which provides the best fitting between predicted and exact outputs by means of correlating the factors of the model. In addition, the linear relationship between a single dependent parameter and two or more independent parameters is performed using the multiple linear regression approach (MLR). An illustration of the utilized mathematical model is provided in Eq.1.

$$Y = \beta X + \varepsilon \tag{1}$$

where $Y = [y_1, \dots, y_n]^T$ is the dependent variable vector, $X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,k} \\ \vdots & \ddots & \vdots \\ x_{n,k} & \cdots & x_{n,k} \end{bmatrix}$ are the Variable independence $\beta = [\beta_1, \dots, \beta_k]^T$ is the model's coefficients vector to be estimated; and n number of observations; $\varepsilon = [\varepsilon_1, \dots, \varepsilon_k]^T$ is a random error vector for k number of inputs.

Table 1
List of the physical and financial variables (P&F) highlighted by Rafiei and Adeli [20].

Project locality	$\boldsymbol{N/A}$
Total floor area of the building	m^2
Lot area	m^2
Total preliminary estimated construction cost	\$
Preliminary estimated construction cost	$\frac{\$}{m^2}$
Equivalent preliminary estimated construction cost	$\frac{\$}{m^2}$
Duration of construction	Quarter, month, or week
Price of the unit at the beginning of the project	$\frac{\$}{m^2}$

Table 2
List of the economic variables and indices (EV&Is) highlighted by Rafiei and Adeli [20].

Number of building permits issued	$\boldsymbol{N/A}$
total subcontractor's amount of contracts (BSI for a preselected base year)	N/A
producer price index (WPI of building materials for the base year)	N/A
Total floor areas of building permits issued	m^2
Cumulative liquidity	Millions of dollars
Private sector investment in new buildings	Millions of dollars
Land price index for the base year	Millions of dollars
Number of loans extended by banks in a time resolution	N/A
Amount of loans extended by banks in a time resolution	Millions of dollars
Interest rate for loan in a time resolution	%
Average construction cost of buildings by private sector at the time of completion of construction	$\frac{\text{Millions of dollars}}{m^2}$
Average of construction cost of buildings by private sector at the beginning of the construction	$\frac{\text{Millions of dollars}}{m^2}$
Official exchange rate with respect to dollars	%
Nonofficial (street market) exchange rate with respect to dollars	%
Consumer price index (CPI) in the base year	N/A
CPI of housing, water, fuel, and power in the base year	N/A
Stock market index	N/A
Population of the city	N/A
Gold price per ounce	\$

It is considered a direct and efficient method used solely as an optimization approach to fit the linear and machine learning models where no association to any specific numerical model is present. Although SGD is one of the oldest machine learning approaches, the incorporation of this approach has recently increased in large-scale MLR modeling due to its superior performance in the case of large data. In general, the computation of the gradient of loss is conducted for each specimen in succession, where a lowering strength schedule is followed for real-time improvement of these specimens. The loss function can consist of ElsticNet's absolute norm, the squared Euclidean norm, or a combination of the two for reaching the model's factors to zero-vector. In this study, the ElsticNet is utilized to calculate the safety factor of a road embankment.

The general steps for SGD are as follows:

- Data preparation: Collect and preprocess the data, including cleaning, normalization, and splitting into training and testing sets.
- Model initialization: Choose the SGD regressor model and set the hyperparameters such as the learning rate, regularization strength, and number of iterations.
- Model training: Train the model on the training data using the stochastic gradient descent algorithm, which updates the model weights after each iteration based on a random subset of the training data.
- Model evaluation: Evaluate the trained model on the testing data using appropriate metrics such as mean squared error or R^2.
- Hyperparameter tuning: Adjust the hyperparameters of the model using techniques such as cross-validation to optimize performance.

The effective parameters for SGD include:

- Learning rate: This parameter determines the step size of the gradient descent algorithm and affects the speed and stability of the model training process.
- Regularization strength: This parameter controls the balance between fitting the training data well and avoiding overfitting to noise. Regularization techniques such as L1 or L2 regularization can be used.
- Number of iterations: This parameter determines the maximum number of times the model weights are updated during training.
- Loss function: This parameter specifies the objective function used to evaluate the model performance during training. Common loss functions for regression problems include mean squared error and mean absolute error.

3.2.2. Support vector regressor (SVR)

The support vector regressor (SVR) was implemented in the engineering field primarily for application in regression issues as a supervised learning method.

For a space of input variables χ, a sample size n, and a given training dataset $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \chi \times R$. The objective of SVR is to perform a full training phase to determine a function f(x) with a maximum deviation ε from the target y_i. For linear functions, *f*(x) can be expressed as shown in Eq. 2.

$$f(x) = \langle w, x \rangle + b \text{ with } w \in \chi, b \ \in R \tag{2}$$

Reduction of the norm is comparable to determining a solution to a convex optimization issue which is used to achieve a small value for w under the case the equation is flat as in Eq. 1. However, the soft margin loss function can be deployed sometimes due to the inappropriateness of the convex optimization issue in order to overcome the complication of the limitations in the optimization issue. Hence, it is demonstrated mathematically in Eq. 3.

$$\begin{aligned}&\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ &\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \le \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \le \varepsilon + \xi_i^* \\ \quad\quad \xi_i, \xi_i^* \quad \ge 0 \end{cases}\end{aligned} \tag{3}$$

Equation 4 represents the linear ε-insensitive loss function$|\xi|_\varepsilon$.

$$|\xi|_\varepsilon := \begin{cases} 0 & \text{if } |\xi| \le \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases} \tag{4}$$

Moreover, Eq. 5 shows the solution to the optimization issue by means of transformation to a dual issue.

$$f(x) = \sum_{i=1}^{n}(a_i - a_i^*)k(x_i, x) + b \quad \text{subject to} \ \begin{smallmatrix} 0 \le a_i \le C \\ 0 \le a_i^* \le C \end{smallmatrix} \tag{5}$$

Along the linear SVR (SVR-L) [4], other kernel types exist, such as RBF, Eq. 6, radial basis function (RBF), Eq. 7, and sigmoid, Eq. 8.

$$\mathrm{k(x, x')} = (\langle \mathrm{x, x'} \rangle + \mathrm{c})^{\mathrm{p}} \tag{6}$$

$$k(x, x') = e^{-\frac{\|x - x'\|}{2\sigma^2}} \tag{7}$$

$$k(x, x') = \tanh(\gamma \langle x, x' \rangle + \vartheta) \tag{8}$$

Thus, this study will examine various kernels to identify the best one.

Here are the general steps for SVR:

- Data Collection: Collect the data from various sources.
- Data Cleaning: Clean the data to remove any missing or erroneous values.
- Data Preparation: Prepare the data for modeling by splitting it into training and testing sets.
- Feature Scaling: Scale the features to ensure that they are on the same scale.
- Model Training: Train the SVR model on the training data.
- Model Evaluation: Evaluate the performance of the model on the testing data.
- Model Tuning: Tune the model's hyperparameters to improve its performance.
- Model Deployment: Deploy the model to make predictions on new data.

Here are some of the effective parameters for SVR:

- kernel: This parameter determines the type of kernel function used in the SVR model. The most common options are "linear", "polynomial", and "radial basis function (RBF)".
- C: This parameter controls the tradeoff between achieving a low training error and a low testing error. Increasing the value of C can result in a more complex model that fits the training data better, but may not generalize well to new data.

- epsilon: This parameter controls the width of the margin around the regression line. Increasing the value of epsilon can result in a wider margin, which can improve the model's robustness to noise.
- gamma: This parameter controls the width of the RBF kernel. Increasing the value of gamma can result in a more complex model that fits the training databetter, but may not generalize well to new data.
- degree: This parameter is only applicable when using a polynomial kernel. It controls the degree of the polynomial used in the kernel function.
- coef0: This parameter is only applicable when using a polynomial or sigmoid kernel. It controls the constant term in the kernel function.
- shrinking: This parameter determines whether or not to use the shrinking heuristic. Setting this parameter to True can speed up training, but may result in a slightly less accurate model.

3.2.3. Decision tree (DT)

The decision tree (DT) is similar to SVR, a familiar learning approach utilized for categorization and regression issues in the data mining industry. One of the significant merits of DT is the full evaluation of all possible outcomes as well as the detection of the paths to the end. In fact, this approach conducts an extensive examination and review of the outcomes as well as each path to perform further analysis for the decision nodes. Furthermore, the decision tree is created from the mixture of these predicting models as the dataset is usually separated iteratively, where each separation is appointed to form the estimation model. A specific training dataset $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \chi \times R$ describes the input variable space χ, and the sample size is n. The comparable results or similar labels for all elements in this approach are classified in the case of iterative separation of the feature space.

The data is demonstrated using Q_m and N_m samples. Subsequently, $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$ are the two generated subsets which are composed of the elements j features t_m threshold for each candidate split. These subsets are stated in Eqs. 9 and 10 accordingly.

$$Q_m^{left}(\theta) = \{(x, y) | x_i \leq t_m\} \tag{9}$$

$$Q_m^{right}(\theta) = Q_m / Q_m^{left}(\theta) \tag{10}$$

The loss function H () is implemented to indicate the quality of the candidate split at specific node m.

$$G(Q_m, \theta) = \frac{N_m^{left}}{N_m} H\left(Q_m^{left}(\theta)\right) + \frac{N_m^{right}}{N_m} H\left(Q_m^{right}(\theta)\right) \tag{11}$$

For the case of decreasing the loss, Eq. 11 will be used to specify the factors needed. Afterwards, the maximum permitted depth of $N_m < min_{samples}$ or $N_m = 1$ is reached using the same recurrent procedure for $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$.

$$\theta^* = argmin_\theta G(Q_m, \theta) \tag{12}$$

Currently, the literature is filled with different decision tree approaches. Thus, the DT model used in this paper will utilize the classification and regression tree (CART) to manage numerical target parameters effectively. Lastly, here are the general steps for building a DT:

- Data collection: Gather data related to the problem you are trying to solve.
- Data preparation: Preprocess the data to clean it up and convert it into a format that can be used by the algorithm. This may include handling missing values, encoding categorical variables, and normalizing numerical variables.
- Splitting the dataset: Split the data into a training set and a testing set. The training set will be used to build the decision tree, while the testing set will be used to evaluate its performance.
- Choosing the splitting criterion: Choose a splitting criterion to use when building the decision tree. Common splitting criteria include Gini index and Information Gain.
- Building the decision tree: Use the chosen splitting criterion to build the decision tree by recursively splitting the data into subsets based on the values of different features.
- Pruning the decision tree: Prune the decision tree to prevent overfitting. This can be done by removing branches that do not improve the performance of the tree on the testing set.
- Testing the decision tree: Evaluate the performance of the decision tree on the testing set. This can be done by calculating metrics such as accuracy, precision, and recall.

Some effective parameters for Decision Tree include:

- Max depth: This parameter sets the maximum depth of the decision tree. A deeper tree can capture more complex relationships in the data, but may also be more prone to overfitting.
- Min samples split: This parameter sets the minimum number of samples required to split an internal node. Setting this parameter too low can lead to overfitting.
- Min samples leaf: This parameter sets the minimum number of samples required to be at a leaf node. Setting this parameter too high can lead to underfitting.
- Max leaf nodes: This parameter sets the maximum number of leaf nodes allowed in the tree. Setting this parameter too low can lead to underfitting.
- Splitting criterion: As mentioned earlier, the choice of splitting criterion can have a significant impact on the performance of the decision tree. Gini index and Information Gain are commonly used criteria.

3.2.4. Random forest (RF)

Random forest (RF) is considered one of the most common models incorporated in businesses and is referred to as a "black box". The importance of this mode stems from its capability to precisely estimate over different datasets with small configurations. It consists of many tree variables, where each tree is associated with the values of a random vector arranged exclusively and distributed identically. Over the last decades, RF has gained noticeable popularity in the domain of civil engineering for establishing functional models. A comparison between DT and RT approaches points to a fundamental distinction since the DT model includes only one tree, whereas the RT model is composed of several trees and a random sample of the training data, which controls each tree accordingly [41]. Therefore, multiple CARTs are built where the basics of bootstrapping and aggregation are deployed to perform RF. The general steps for building a RF model are as follows:

- Prepare the data: Random Forest requires a labeled dataset with both input features and output labels. The dataset should be split into training and testing sets.
- Build multiple decision trees: Random Forest builds multiple decision trees on different sub-samples of the training dataset. Each decision tree is trained on a different subset of the features and data.
- Split nodes based on feature importance: At each node of each decision tree, the algorithm selects a random subset of features and chooses the best one to split the node. The best feature is chosen based on the information gain or Gini impurity criteria.
- Build the forest: After building all the decision trees, the algorithm combines the predictions of each tree to make a final prediction. For classification tasks, the algorithm takes the majority vote of the predictions, while for regression tasks, the algorithm takes the average of the predictions.
- Evaluate the model: The performance of the Random Forest model is evaluated using metrics such as accuracy, precision, recall, and F1 score. The model can be fine-tuned by adjusting the hyperparameters such as the number of trees, the maximum depth of each tree, and the number of features to consider at each node.
- Use the model: Once the Random Forest model is trained and evaluated, it can be used to make predictions on new data.

Here are the effective parameters for RF:

- Number of trees: This parameter sets the number of decision trees to use in the random forest. A larger number of trees can improve the accuracy of the model, but may also increase the computation time.
- Maximum depth: This parameter sets the maximum depth of each decision tree. A deeper tree can capture more complex relationships in the data, but may also be more prone to overfitting.
- Minimum samples split: This parameter sets the minimum number of samples required to split an internal node. Setting this parameter too low can lead to overfitting.
- Minimum samples leaf: This parameter sets the minimum number of samples required to be at a leaf node. Setting this parameter too high can lead to underfitting.
- Maximum features: This parameter sets the maximum number of features to consider when splitting a node. Setting this parameter too low can lead to underfitting, while setting it too high can lead to overfitting.
- Bootstrap sampling: This parameter controls whether or not to use bootstrap sampling to randomly sample the data when building each decision tree. Using bootstrap sampling can help to reduce the variance of the model.
- Feature importance: This parameter can be used to calculate the importance of each feature in the model. This can be useful for feature selection and understanding the relationships between different features.

- Random state: This parameter sets the seed used by the random number generator. Setting a fixed seed can help to ensure that the model produces consistent results.

3.2.5. Extremely randomized trees (ETR)

The extremely randomized trees (ETR) approach is based on the random determination of the thresholds for each candidate property as well as on the random determination of the splits within the tree's nodes to select the most suitable candidate as a splitting criterion in contrast to RF model which is based on the most critical thresholds. The bias is marginally increased, and the variance is slightly reduced in this model. Moreover, the most crucial distinction between ETR and RF model is that ETR examines the whole actual sample while RF deploys bootstrap duplicates where it substitutes the input data using down-sampling. ETR with sklearn utilization includes the potential to use bootstrap duplicates. Nonetheless, this approach includes the entire input sample where the variance is increased due to bootstrapping. Here are the general steps for ETR:

- Collect and prepare data: As with any machine learning model, the first step in using Extremely Randomized Trees is to collect and prepare the data. This involves selecting the features and labels to use for training and testing, cleaning and processing the data, and splitting it into training and testing sets.
- Initialize the ETR model: The next step is to initialize the ETR model and set the parameters. This involves specifying the number of trees to use, the maximum depth of the trees, the minimum number of samples required to split an internal node, the minimum number of samples required to be at a leaf node, the maximum number of features to consider when looking for the best split, whether or not to use bootstrap sampling, and whether or not to use extra randomization.
- Train the ETR model: Once the model has been initialized and the parameters have been set, the next step is to train the ETR model using the training data. This involves constructing the trees by recursively splitting the data based on the selected features and labels, and using a measure of impurity (such as Gini index or entropy) to determine the best split at each internal node.
- Evaluate the ETR model: After the ETR model has been trained, the next step is to evaluate its performance on the testing data. This involves using the trained model to predict the labels for the testing data, and comparing these predictions to the actual labels. Metrics such as accuracy, precision, recall, and F1 score can be used to evaluate the performance of the model.
- Tune the ETR model: Finally, the performance of the ETR model can be further improved by tuning the parameters. This involves adjusting the values of the parameters to find the optimal combination that produces the best performance on the testing data. Various techniques such as grid search or randomized search can be used to automate the parameter tuning process.

The effective parameters for ETR are:

- Number of trees: The number of trees to be used in the ETR model. Increasing the number of trees can improve the accuracy of the model, but also increase the computational time.
- Maximum depth: The maximum depth of each tree in the ETR model. Setting the maximum depth too high can lead to overfitting, while setting it too low can result in underfitting.
- Minimum samples split: The minimum number of samples required to split an internal node. Increasing this parameter can help to control the complexity of the trees and prevent overfitting.
- Minimum samples leaf: The minimum number of samples required to be at a leaf node. Increasing this parameter can help to control the complexity of the trees and prevent overfitting.
- Maximum features: The maximum number of features to consider when looking for the best split. Setting this parameter too low can result in poor performance due to lack of diversity, while setting it too high can result in overfitting.
- Bootstrap sampling: A boolean parameter that determines whether or not to use bootstrap sampling. Setting this parameter to True can help to reduce the variance of the model.
- Extra randomization: A boolean parameter that determines whether or not to use extra randomization. Setting this parameter to True can increase the diversity of the trees and improve the performance of the model.
- Random state: A parameter that sets the random seed for the ETR model. Setting this parameter can help to ensure that the model produces consistent results.

1.1.1. Adaptive Boosting (Ada)

Adaptable boosting (Ada) is a meta-algorithm implemented in a wide range of diverse learning methods, as described in the literature, to improve performance. The algorithm generally depends on the iterative process where the weights are altered when the previous trial fails. Later, the same actual training dataset and the chosen regressor are utilized to fit multiple cases of the regression model. As a result, the model adopts Drucker's guidelines for handling problematic cases. In addition to that, Ada model was one-level DT regressor reduced, and the mathematical representation is concisely presented. Using a predetermined dataset $\{(x_1, y_1), \dots, (x_n, y_n)\} \subset \chi \times R$ a base predictor (weak learner) $f(x)$ is trained. Where n indicates the sample size, χ space of input variables, and e_i is the error obtained for the whole set. Afterward, using the technique in Eq. 13, a series of weak learners $f_k(x), k = 1,2, \dots, N$ are created and grouped to form a strong model $H(x)$.

$$H(x) = v \sum_{k=1}^{N} \left(ln \frac{1}{\alpha_k} \right) g(x) \tag{13}$$

According to Eq. 14, $g(x)$ is the median of all $\alpha_k f_k(x)$, v is the learning rate, and α_k is the weight of the base estimators. The most significant merit of Ada is the capability of grouping strong base learners, including deep decisions as well as merging weak base learners leading to far more accurate models.

$$\alpha_k = \frac{e_i}{1 - e_i} \tag{14}$$

The general steps for Ada are:

- Initialize sample weights: Each sample in the training set is assigned an equal weight.
- Train a weak learner: A weak learner is trained on the training set using the current weights.
- Evaluate the weak learner: The performance of the weak learner is evaluated on the training set.
- Update sample weights: The weights of the samples are updated based on their classification error. Samples that were classified correctly are assigned a lower weight, while misclassified samples are assigned a higher weight.
- Train another weak learner: A new weak learner is trained on the updated weights.
- Evaluate the new weak learner: The performance of the new weak learner is evaluated on the training set.
- Repeat steps 4-6 for a predetermined number of iterations, or until a stopping criterion is met.
- Combine the weak learners: The weak learners are combined to create a strong learner, which is used to make predictions on new data.

The effective parameters for Ada are:

- Base estimator: The type of weak learner to be used, such as decision trees or linear models.
- Learning rate: The contribution of each weak learner to the final prediction. A smaller learning rate will result in a slower learning process, but can help to prevent overfitting.
- Number of estimators: The number of weak learners to be used in the Ada model. Increasing the number of estimators can improve the accuracy of the model, but also increase the computational time.
- Loss function: The function used to measure the difference between the predicted and actual values. Common loss functions include binary cross-entropy and mean squared error.
- Random state: A parameter that sets the random seed for the Ada model. Setting this parameter can help to ensure that the model produces consistent results.

3.2.6. Stochastic gradient boosting (GB)

Stochastic gradient boosting (GB) is an enhanced version of the conventional gradient boosting method used for regression and categorization functions. This approach is consistent with Ada in terms of combining learners in succession to form a new model. On the other hand, the most noticeable distinction between the two approaches is Ada aims to minimize the learner's loss function. Additionally, the weak estimator of GB possesses a higher DT than the reduced one-level regressor of Ada model. The stochastic gradient boosting (GB) model does not require training for the whole dataset; only the training is performed for the base learner with a fraction of $f < 1$ via arbitrary choosing where no replacement is needed. Hence, the advantage of this approach is the prevention of overfitting and diminishing the trees' correlation.

The general steps for GB are:

- Initialize the model: The first estimator is trained on the training set.
- Predictions: The model is used to make predictions on the training set.
- Residuals: The difference between the predicted and actual values is calculated.
- Train a new model: A new model is trained on the residuals of the previous model.
- Update predictions: The predictions of the previous models are updated by adding the predictions of the new model, multiplied by a learning rate.
- Repeat steps 3-5 for a predetermined number of iterations, or until a stopping criterion is met.
- Combine the models: The models are combined to create a strong learner, which is used to make predictions on new data.

The effective parameters for GB are:

- Number of estimators: The number of models to be used in the GB model. Increasing the number of estimators can improve the accuracy of the model, but also increase the computational time.
- Learning rate: The contribution of each model to the final prediction. A smaller learning rate will result in a slower learning process, but can help to prevent overfitting.
- Subsample: The fraction of samples to be used for each model. A smaller subsample can help to prevent overfitting.
- Maximum depth: The maximum depth of each tree in the GB model. Increasing the maximum depth can improve the accuracy of the model, but also increase the risk of overfitting.
- Loss function: The function used to measure the difference between the predicted and actual values. Common loss functions include binary cross-entropy and mean squared error.
- Random state: A parameter that sets the random seed for the GB model. Setting this parameter can help to ensure that the model produces consistent results.

3.2.7. Histogram-based gradient boosting (HGB)

Histogram-based gradient boosting (HGBoost) differs from other machine learning techniques since it assigns permanent attribute values into bins forming attribute histograms deployed during training. Accordingly, this approach displays superiorities in terms of accelerating the training stage, immediately enhancing the quality, and minimizing the memory requirements of the model. Regardless, the current orientation of the research is toward gradient boosting algorithms instead of conventional base learners to produce machine learning applications with superior quality and reduced outcome period. Here are the general steps for HGB:

- Initialize the model: Start by initializing the HGB model with the desired hyperparameters such as the number of estimators, learning rate, maximum depth, number of bins, L2 regularization, and random state.

- Fit the model: Train the HGB model on the training data. The model will iteratively add new trees to the ensemble, each time focusing on the residuals (difference between predicted and actual values) of the previous model.
- Predict on test data: Use the trained HGB model to predict the target variable for the test data.
- Evaluate the model: Assess the performance of the HGB model using appropriate evaluation metrics such as mean squared error (MSE), root mean squared error (RMSE), and coefficient of determination (R-squared).
- Tune the hyperparameters: Adjust the hyperparameters of the HGB model to optimize its performance on the given task.
- Repeat steps 2-5: Iterate through steps 2-5 until the desired level of model performance is achieved.
- Finalize the model: Once the optimal hyperparameters have been identified, train a final HGB model on the entire dataset (including training and validation data) using these hyperparameters.
- Deploy the model: Use the final HGB model to make predictions on new, unseen data.

Here are the effective parameters for Histogram-Based Gradient Boosting (HGB):

- Number of estimators: The number of models to be used in the HGB model. Increasing the number of estimators can improve the accuracy of the model, but also increase the computational time.
- Learning rate: The contribution of each model to the final prediction. A smaller learning rate will result in a slower learning process, but can help to prevent overfitting.
- Maximum depth: The maximum depth of each tree in the HGB model. Increasing the maximum depth can improve the accuracy of the model, but also increase the risk of overfitting.
- Number of bins: The number of bins to be used in the histogram-based algorithm. Increasing the number of bins can improve the accuracy of the model, but also increase the computational time.
- L2 regularization: A parameter that penalizes large weights in the model. Increasing the L2 regularization can help to prevent overfitting.
- Random state: A parameter that sets the random seed for the HGB model. Setting this parameter can help to ensure that the model produces consistent results.

3.2.8. Extreme gradient boosting (XGB)

Extreme gradient boosting (XGB) is an effective and versatile machine learning approach capable of producing consecutive decision trees where a weight classification for each independent parameter is made and then assigned to the decision tree for predicting outcomes. Another classification is performed for the wrongly estimated parameters where a larger weight is assigned in the second decision tree. Finally, an accurate and resilient model is generated by combining various forecasters and classifiers. Thus, the XGB algorithm is primarily based on the influence of the weights. The similarity between XGB and gradient boosting is that both depend on the gradient boosting principle, leading to discrete modeling characteristics where XGB is implemented to overcome the overfitting issues and eventually yield surpassing outcomes.

Here are the general steps for XGB and the effective parameters:

- Initialize the model: Start by initializing the XGB model with the desired hyperparameters such as the learning rate, maximum depth, number of trees, and random state.
- Train the model: Train the XGB model on the training data by iteratively adding decision trees to the ensemble, each time focusing on the residuals (difference between predicted and actual values) of the previous model.
- Evaluate the model: Assess the performance of the XGB model using appropriate evaluation metrics such as mean squared error (MSE), root mean squared error (RMSE), and coefficient of determination (R-squared).
- Tune the hyperparameters: Adjust the hyperparameters of the XGB model to optimize its performance on the given task.
- Finalize the model: Once the optimal hyperparameters have been identified, train a final XGB model on the entire dataset (including training and validation data) using these hyperparameters.
- Deploy the model: Use the final XGB model to make predictions on new, unseen data.

Here are the effective parameters include:

- Learning rate: Controls the contribution of each tree to the final prediction. Lower values can help prevent overfitting but may require more trees for sufficient model performance.
- Maximum depth: Limits the depth of each tree, which can help prevent overfitting.
- Number of trees: Determines the number of trees to be added to the ensemble. Increasing the number of trees can improve the model's performance, but may also increase the risk of overfitting.
- Subsample: Controls the fraction of observations to be randomly sampled for each tree. This can help to reduce overfitting by increasing the diversity of the ensemble.
- Colsample bytree: Controls the fraction of features to be randomly sampled for each tree. This can help to reduce overfitting and increase the diversity of the ensemble.
- Alpha: L1 regularization term on weights, which can help to prevent overfitting.
- Lambda: L2 regularization term on weights, which can help to prevent overfitting.

3.3. Model development and hyperparameters tunning

The optimization of the hyperparameters was conducted using the grid search technique coupled with 10-fold cross-validation in the training process. This approach aims to fine-tune the machine learning models, ensuring they perform optimally on unseen data. In general, the grid search is a comprehensive technique for hyperparameter optimization that involves evaluating all possible combinations of specified hyperparameter values for a given machine learning model. The primary advantage of grid search lies in its exhaustive exploration of the hyperparameter space, ensuring that the optimal combination is identified. However, this also makes it computationally expensive, especially for models with a large number of hyperparameters or a wide range of possible values. Cross-validation is a method for evaluating the performance of machine learning models by partitioning the dataset into smaller subsets or 'folds.' In k-fold cross-validation, the data is divided into 'k' equally sized folds, where one fold is held out as the validation set while the remaining k-1 folds are used for training. This process is repeated 'k' times, ensuring that each fold is used once as the validation set. The model's performance is then assessed by averaging the results obtained from the 'k' iterations, providing a reliable and robust estimation of its performance on unseen data. In the case of 10-fold cross-validation, the dataset is partitioned into ten equally sized folds. Each of the ten iterations holds out one fold as the validation set and uses the remaining nine folds for training. By implementing 10-fold cross-validation, it is possible to ensure that the model is evaluated on multiple subsets of the data, thus mitigating the risk of overfitting and providing a more reliable estimate of the model's performance on new data. The combination of grid search with 10-fold cross-validation presents a robust method for hyperparameter optimization. This technique is applied to the training process, where various machine learning models are constructed using different combinations of hyperparameter values. The performance of each model is then assessed using 10-fold cross-validation, producing an averaged performance score. The combination of hyperparameters that yields the highest average score is selected as the optimal configuration for the given model. The current study examined an extensive range of hyperparameter values to identify the most suitable model configuration. A flowchart summarizing the process of constructing and evaluating models using the grid search technique with 10-fold cross-validation in the Scikit-learn library of Python is illustrated in Fig. 1. By adopting this approach, the study aims to yield machine learning models that exhibit strong performance on both the training data and previously unseen data, thus maximizing the models' predictive capabilities and overall reliability.

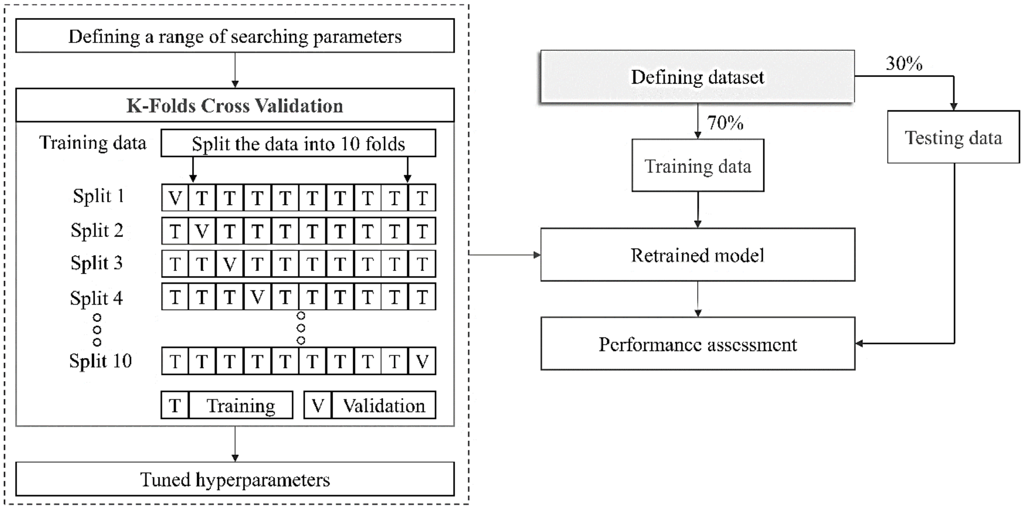
Defining a range of searching parameters
K-Folds Cross Validation
Training data
Split the data into 10 folds
Split 1 V T T T T T T T T T
Split 2 T V T T T T T T T T
Split 3 T T V T T T T T T T
Split 4 T T T V T T T T T T
Split 10 T T T T T T T T T V
T Training
V Validation
Tuned hyperparameters
Defining dataset
30%
70%
Training data
Testing data
Retrained model
Performance assessment

Fig. 1. Flowchart of the adopted approach of the machine learning model's development.

3.4. Models' performance assessment

The goodness of fit of the linear regression model deployed the coefficient of determination (R^2) where the numerator of the R^2 fraction depends on the unidentified dissimilarities by the response independent parameters, whereas the denominator the R^2 fraction depends on the total dissimilarities in the response, Eq. 15 [42]. The range of R^2 values are between 0 and 1 where 1 represents the strongest linear relationship. In fact, root-mean-square error (RMSE) is an error analysis used to measure the difference between the estimated and observed values, Eq. 16. Additionally, the mean absolute error (MAE) is an error analysis used to measure the difference between the absolute estimated and observed values, Eq .17.

$$R^2 = 1 - \frac{\sum(x_i - y_i)^2}{\sum(x_i - \bar{x}_i)^2} \tag{15}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}} \tag{16}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \bar{y}_i| \tag{17}$$

where x_i is the measured value, $\bar{x}_i$ is the mean of the measured values, y_i is the predicted value, $\bar{y}_i$ is the mean of the predicted values, and n is the number of observations.

4. Results and discussions

In this section, we will delve into the implemented machine learning models used in the study to forecast project sales valuation. We will discuss the efficiency of these models and compare their performance and accuracy to determine the best model. Our analysis of the training stage for all the machine learning models demonstrated superior performance and greater accuracy. Specifically, the fitting rate of training data was concentrated near the equity line, which suggests that the models were able to fit the training data well. However, the results of the testing stage showed some variation across the different models, as illustrated in Fig. 2. To determine the best machine learning model for both the training and testing phases, we conducted a comparative analysis. Our results showed that the ERT model outperformed all other models, exhibiting the highest level of accuracy and performance. On the other hand, the SGD model performed the worst among all the models used in the study. Overall, our findings indicate that machine learning models can be used effectively to forecast project sales valuation. The ERT model, in particular, can provide the best results when compared to other commonly used models. These findings may have significant implications for industries that rely on accurate sales valuation forecasts to make strategic decisions.

During the training stage of our study, we evaluated the predicted profit values computed using machine learning models against the observed profit values. The results of this analysis are presented in Fig. 3.

Our findings show that the ERT model achieved the highest profit value, which was marked at 860 $\frac{\$}{m}$. This indicates that the ERT model was able to predict sales valuation accurately, resulting in higher profits for the project. On the other hand, the GB model recorded the lowest profit value, approximately 740 $\frac{\$}{m}$. This suggests that the GB model may not be the best choice for accurately predicting sales valuation and maximizing project profits

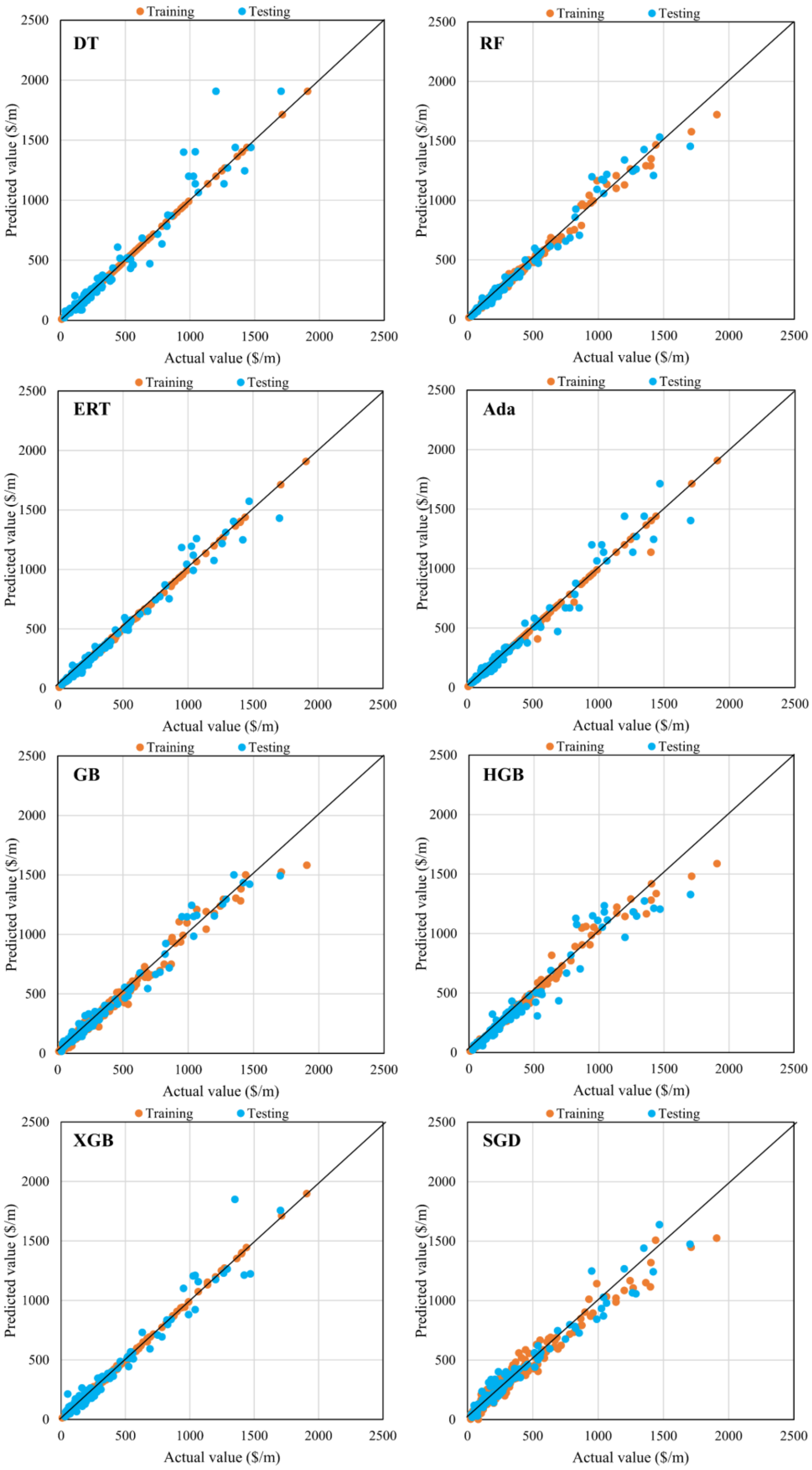

Fig. 1. Training and testing results of the selected results.

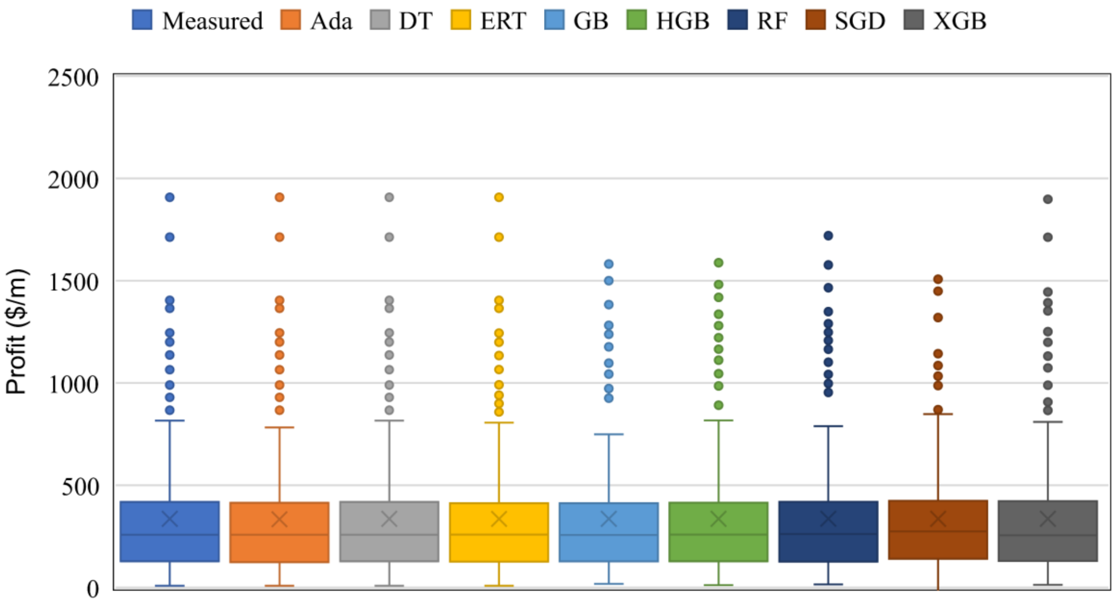

Fig. 2. Comparative of machine learning models' outcomes in the training dataset.

These results have significant implications for businesses that rely on accurate profit forecasts to make informed decisions about their projects. By using machine learning models such as the ERT model, these businesses can improve their profitability and make better decisions based on accurate sales valuation forecasts. The GB model, on the other hand, may not provide the desired level of accuracy and may result in suboptimal profits for the project.

In our study, we evaluated the observed and estimated profit values of the testing dataset using different machine learning models. The results of this analysis are presented in Fig. 4.

Our findings show that the GB model achieved the highest profit result, which was approximately 530 $\frac{\$}{m}$. This suggests that the GB model was able to accurately predict sales valuation, resulting in higher profits for the project. On the other hand, the Ada model exhibited the lowest profit result, which was approximately 496 $\frac{\$}{m}$. This indicates that the Ada model may not be the best choice for accurately predicting sales valuation and maximizing project profits.

In our study, we conducted a residual analysis to validate the accuracy of our machine learning models. Residual analysis is a widely used technique that measures the difference between the observed and predicted values of a model. The residual is calculated by subtracting the predicted value from the actual observed value, as shown in Eq. 18.

The residual analysis helps to identify the accuracy of the model by evaluating how well it fits the data. Specifically, it measures the deviation of the predicted values from the actual observed values. A residual value of zero indicates that the predicted value is exactly the same as the observed value. Positive residual values indicate that the predicted value is higher than the observed value, while negative residual values indicate that the predicted value is lower than the observed value.

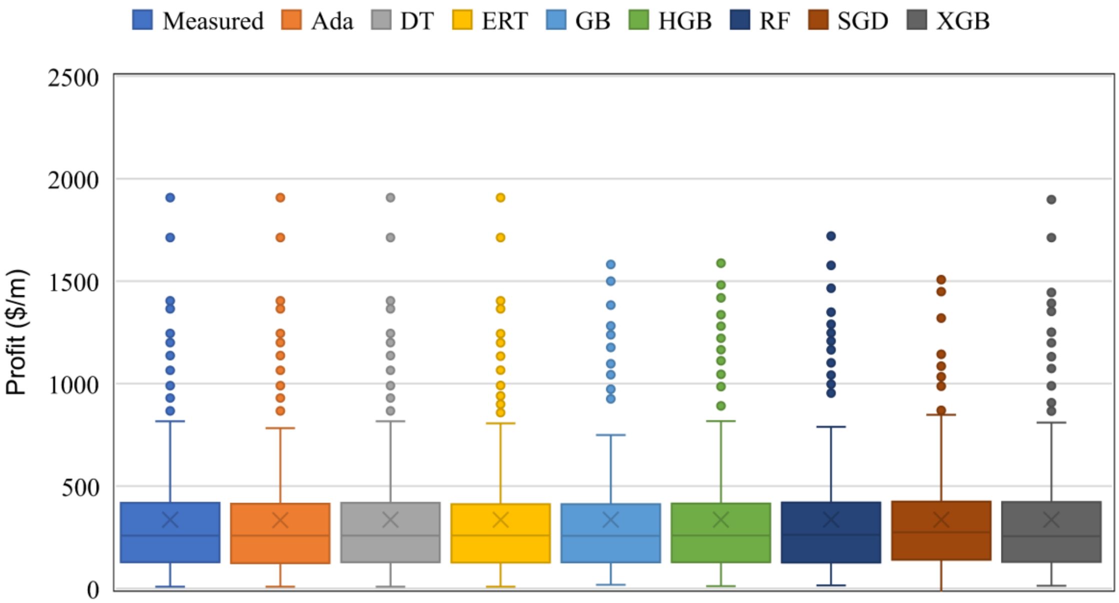

Fig. 4. Comparative of machine learning models' outcomes in the testing dataset.

$$e_i = (y_i - \hat{y}_i) \tag{18}$$

To visualize the residual values, we plotted them on the vertical axis against one variable on the horizontal axis. This graph helps identify any patterns or trends in the residual values and determine whether the model is systematically overestimating the observed values.

In our study, we conducted a residual analysis of the training dataset using various machine learning models to assess their accuracy in forecasting sales valuations. We plotted the residual values over the entire observation numbers, as shown in Fig. 5.

The residual values for all the machine learning models exhibited typical patterns and concentrated values over the entire observation period. This suggests that the models are accurate in their predictions, and there are no significant outliers or errors in the training data.

However, it is worth noting that some models performed better than others in terms of minimizing the residual values. The HGB model, for example, recorded the lowest residual value, nearly at -170 $\frac{\$}{m}$. This indicates that the model is making accurate predictions and is a good fit for the training data. On the other hand, the SGD model recorded the highest residual value, approximately at 400 $\frac{\$}{m}$. This suggests that the model may be overestimating or underestimating the observed values and may need to be adjusted or retrained to improve its accuracy.

Overall, the residual analysis provides valuable insights into the accuracy of machine learning models in forecasting sales valuations. By carefully analyzing the residual values and identifying any patterns or trends, businesses can fine-tune their models to improve their accuracy and make more informed decisions about their projects.

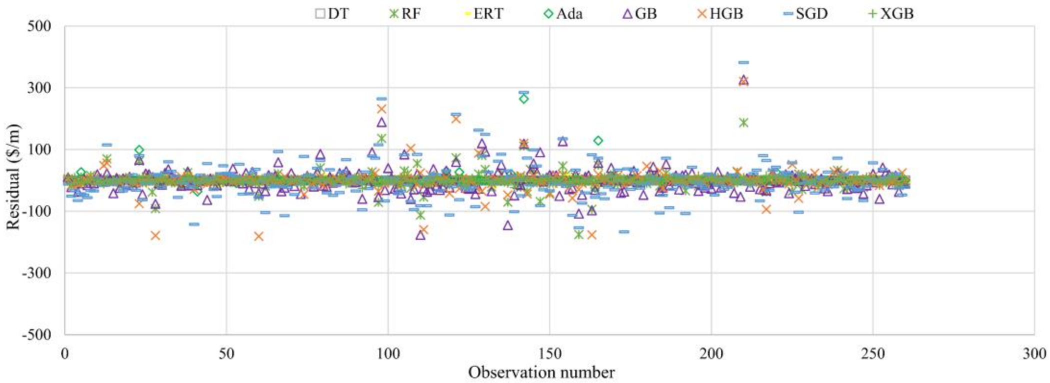

Fig. 3. Residuals of the machine learning models for the training dataset.

In our study, we conducted a residual analysis of the testing dataset using different machine learning models to assess their accuracy in forecasting sales valuations. We plotted the residual values over the observation numbers, as shown in Fig. 6.

The residual values for all the machine learning models exhibited a scattered pattern over the entire observation period. This indicates that the models may not be as accurate in their predictions as they were during the training phase. However, it is worth noting that some models performed better than others in terms of minimizing the residual values.

The ST model, for example, recorded the lowest residual value, nearly at -420 $\frac{\$}{m}$. This suggests that the model is making accurate predictions and is a good fit for the testing data. On the other hand, the HGB model recorded the highest residual value, approximately at 398 $\frac{\$}{m}$. This indicates that the model may be overestimating or underestimating the observed values, and it may need to be adjusted or retrained to improve its accuracy.

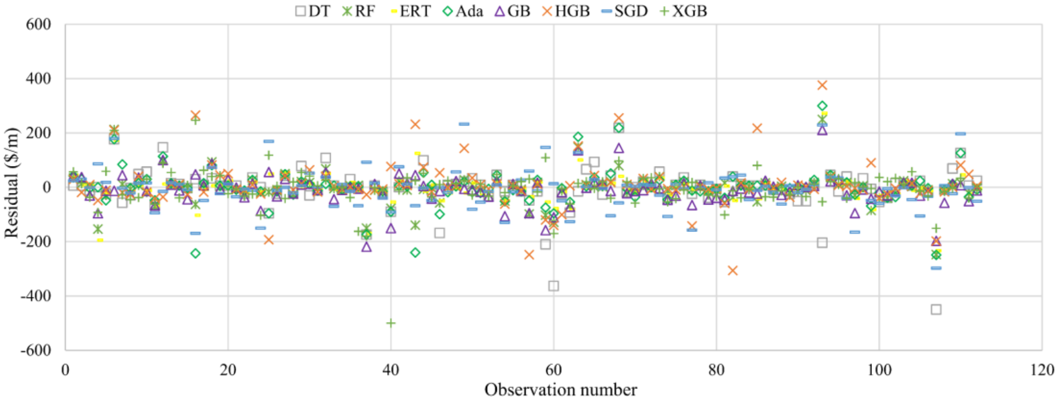

Fig. 4. Residuals of the machine learning models for the testing dataset.

In our study, we evaluated the performance of different machine learning models in forecasting sales valuations by analyzing their R^2 values. We plotted the R^2 values for both the training and testing stages using the tested machine learning models in Fig. 7 and provided the values in Table 3. We observed that the R^2 values for both the training and testing stages were quite comparable and similar, indicating that the models were performing well in predicting the sales valuations.

During the training stage, most of the machine learning models demonstrated R^2 values of 1, indicating that they fit the training data perfectly. However, the GB, HGB, RF, and SGD models recorded slightly lower R^2 values of 0.97, 0.974, 0.99, and 0.95, respectively. During the testing stage, the ERT model exhibited the highest R^2 value, whereas the DT model showed the lowest R^2 value, at 0.996 and 0.925, respectively. This indicates that the ERT model performed the best in predicting sales valuations on the testing dataset, while the DT model may need further optimization or adjustments to improve its accuracy. Overall, analyzing the R^2 values provides important insights into the performance of machine learning models in forecasting sales valuations. By comparing the R^2 values of different models, businesses can select the best model that fits their specific needs and requirements. However, it is essential to keep in mind that the R^2 values should not be the only factor considered in selecting a model, as other metrics such as accuracy, precision, and recall also play crucial roles in determining the model's effectiveness.

Table 1
Performance metrics of the developed models for both training and testing cases.

	R^2		MAE		MSE		RMSE		Max Error	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Ada	1	0.96	19.8	70.41	8	1567	2.84	39.58	264	300
GB	0.98	0.97	41.85	57.72	669	1431	25.87	37.83	327	218
HGB	0.98	0.94	40.32	86.98	258	2280	16.05	47.75	320	376
XGB	1	0.96	7.34	75.66	33	2010	5.77	44.83	25	500
RF	0.99	0.97	28.76	58.76	220	1144	14.82	33.83	188	250
ERT	1	0.98	5.96	54.82	16	920	3.96	30.33	24	273
DT	1	0.92	0	104.98	0	2574	0	50.73	0	708
SGD	0.96	0.96	59.63	74.93	1581	2366	39.76	48.64	382	298

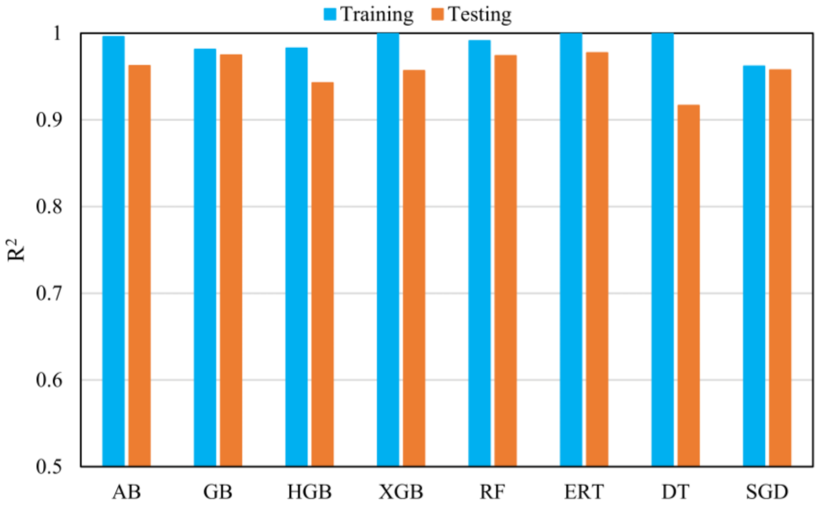

Fig. 5. R^2 of the investigated machine learning models.

The RMSE values for both the training and testing stages using various machine learning models are important metrics that evaluate the performance of the models. The graphs of RMSE and MSE values for both datasets are shown in Fig. 8 and Fig. 9. As expected, the RMSE values of all machine learning models for the testing dataset were significantly higher than those of the training dataset. During the training stage, the SGD model had the highest RMSE value at 40 $\frac{\$}{m}$, while the AB model had the lowest RMSE value at 3 $\frac{\$}{m}$. However, during the testing stage, the DT model exhibited the highest RMSE value at 51 $\frac{\$}{m}$, while the ERT model had the lowest RMSE value at 30 $\frac{\$}{m}$. These results suggest that the ERT model performs better than the other models in terms of forecasting the project sales valuation. It is worth noting that although the training dataset showed superior performance and greater accuracy than the testing dataset for all the machine learning models, the RMSE values of the testing dataset were still relatively low, indicating good predictive power. In addition, the RMSE values of the training dataset were relatively low, indicating that the models were well-fitted to the training data.

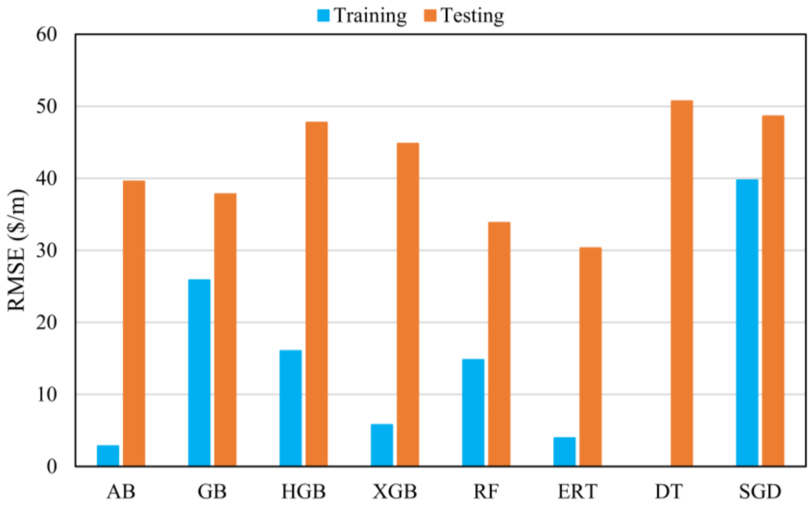

Fig. 6. RMSE of the investigated machine learning models.

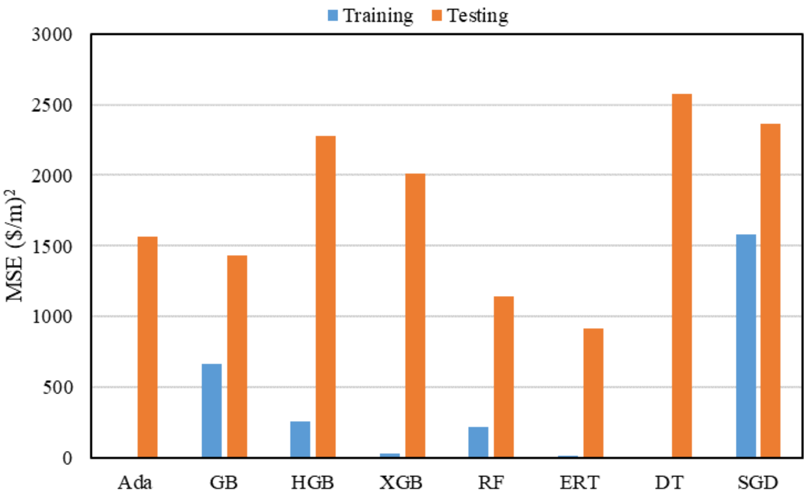

Fig. 7. MSE of the investigated machine learning models.

The MAE values for the different machine learning models were analyzed for both the training and testing datasets, as shown in Fig. 10. It was observed that the SGD model had the highest MAE value during the training stage, with a value of 60 $\frac{\$}{m}$. In contrast, the ERT model had the lowest MAE value during the training stage, with a value of 7 $\frac{\$}{m}$. During the testing stage, the DT model had the highest MAE value of 104 $\frac{\$}{m}$, while the ERT model had the lowest MAE value of 55 $\frac{\$}{m}$. The MAE metric measures the absolute difference between the predicted values and the observed values, making it a useful tool for evaluating the accuracy of machine learning models. The results of this analysis indicate that the ERT model performed the best overall, with the lowest MAE values for both training and testing datasets. In contrast, the DT model performed poorly, with the highest MAE value for the testing dataset. These findings suggest that the ERT model may be the most appropriate machine learning model for predicting project sales valuation in this study.

Similar to the results of the MAE, the maximum error in Fig. 11 also provides the same conclusion on the models' performances. The proposed method appears to have performed well in the available data evaluation metrics. However, it is important to consider how the method might cope with external disturbances, modeling errors, and uncertainties that are common in practical applications. External disturbances, such as changes in operating conditions, can affect the accuracy of the machine learning models. The performance of the models may deteriorate if the data distribution changes significantly. Therefore, it is essential to monitor the model's performance in real-time and retrain it when necessary using new data that accounts for any changes in the operating conditions. Modeling errors may arise due to various reasons, including measurement errors, missing data, and incorrect assumptions. These errors can lead to inaccurate predictions and affect the model's overall performance. To cope with modeling errors, it is important to validate the data and model assumptions and ensure that the model is trained on high-quality data. Uncertainties are inherent in any real-world application, and the proposed method needs to account for them. Understanding the sources of uncertainties and their impact on the system's performance is essential. One way to account for uncertainties is by using probabilistic modeling techniques such as Bayesian modeling, which can estimate the probability distribution of the predicted outputs.

In summary, while the proposed method shows promise in the evaluation metrics, it is important to consider how it will cope with external disturbances, modeling errors, and uncertainties in practical applications. Ongoing monitoring and retraining of the model using new data that accounts for changes in operating conditions, validation of data and model assumptions, and the use of probabilistic modeling techniques are all strategies that can help address these issues. On the other hand, the computational burden of machine learning models can vary depending on factors such as the dataset's size, the model's complexity, and the hardware used for training and inference. Generally, more complex models with larger datasets require more computational resources, such as processing power and memory. For example, deep learning models, such as convolutional neural networks and recurrent neural networks, can be computationally intensive and require high-end GPUs or TPUs to train efficiently. On the other hand, simpler models, such as linear regression or decision trees, may have lower computational requirements. In the context of the specific study being discussed, the models were implemented using Python libraries such as Scikit-learn and XGBoost, which are known for their efficiency and scalability. Additionally, the models were optimized using a computer with an Intel Core i7 CPU and 16 GB of RAM, which suggests that the models may have moderate computational requirements based on the adopted hyperparameter optimization strategy and the range of parameters being optimized.

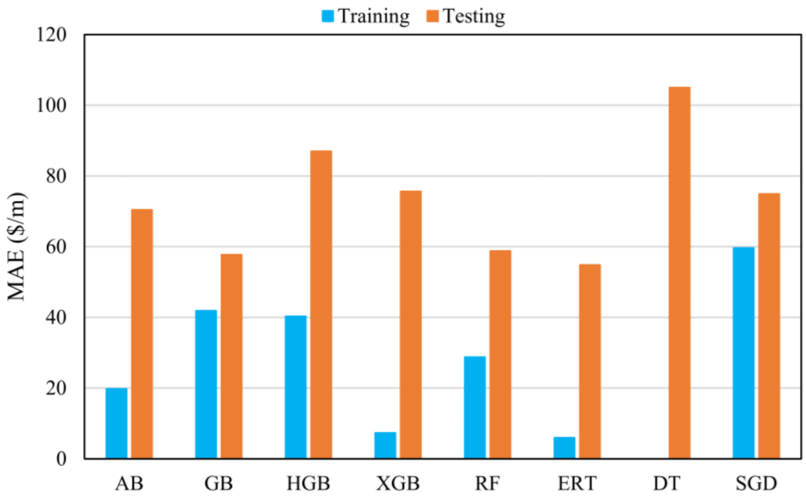

Fig. 8. MAE of the investigated machine learning models.

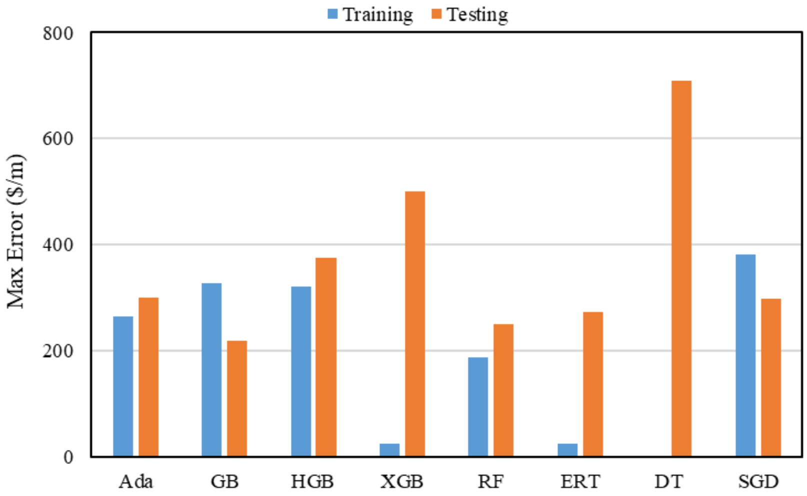

Fig. 9. Maximum Error of the investigated machine learning models.

5. Conclusions

This study aimed to evaluate the efficiency of a wide variety of different machine learning models in estimating the sales profit of projects. In addition, the inputs of machine learning models will be selected using various economic variables and indices to generate the outputs. Finally, the outputs of these machine learning approaches were investigated and compared to the observed measurements. This paper represents a comparison between the diverse machine learning approaches, which contributes to the literature review in defining the best performance of the machine learning model in predicting the sales profit of projects.

Based on the above-mentioned statement, the following conclusions are made:

- A wide range of various machine learning approaches was utilized where they showed suitable and adequate performance in predicting the sales profit of projects.
- The ERT model showed the lowest error in both the RMSE and MAE cases.
- The DT model achieved the highest error in both the RMSE and MAE cases.
- During the training phase, the residual results of the tested machine learning models demonstrated concentrated values and similar patterns over the entire observation numbers. On the other hand, the residual results in the testing dataset illustrated scattered values.

This study is limited to profit value prediction and did not went into other parameters. Additionally, it mainly focuses on ensemble machine learning models and does not go into details of other techniques such as genetic programing or regularized regression models. On the other hand, future efforts in this field can include testing other soft computing techniques including the regularized regression methods and investigating the sensitivity of various indices on the profit value by employing some artificial intelligence techniques that can handle large sets of inputs such as deep learning.

Funding

This research received no external funding.

Conflicts of interest

The authors declare no conflict of interest.

Authors contribution statement

YA: Conceptualization; YA: Formal analysis; YA: Investigation; YA: Methodology; YA: Roles/Writing – original draft.

References

[1] Peter NJ, Okagbue HI, Obasi EC, Akinola A. Review on the Application of Artificial Neural Networks in Real Estate Valuation. Int J Adv Trends Comput Sci Eng 2020;9:2918–25. https://doi.org/10.30534/ijatcse/2020/66932020.

[2] Alfaro-Navarro J-L, Cano EL, Alfaro-Cortés E, García N, Gámez M, Larraz B. A Fully Automated Adjustment of Ensemble Methods in Machine Learning for Modeling Complex Real Estate Systems. Complexity 2020;2020:1–12. https://doi.org/10.1155/2020/5287263.

[3] Chiu S-M, Chen Y-C, Lee C. Estate price prediction system based on temporal and spatial features and lightweight deep learning model. Appl Intell 2022;52:808–34. https://doi.org/10.1007/s10489-021-02472-6.

[4] Peter NJ, Fateye OB, Oloke CO, Iyanda P. Changing urban land use and neighbourhood quality: evidence from Federal Capital Territory (FCT), Abuja, Nigeria. Int J Civ Eng Technol 2018;9:23–36.

[5] Pinter G, Mosavi A, Felde I. Artificial Intelligence for Modeling Real Estate Price Using Call Detail Records and Hybrid Machine Learning Approach. Entropy 2020;22:1421. https://doi.org/10.3390/e22121421.

[6] Frew J, Jud G. Estimating the Value of Apartment Buildings. J Real Estate Res 2003;25:77–86. https://doi.org/10.1080/10835547.2003.12091101.

[7] Limsombunchao V. House price prediction: hedonic price model vs. artificial neural network 2004.

[8] Ayuthaya NP na, Swierczek FW. Factors influencing variation in value and investors confidence. IOSR J Bus Manag 2014;16:41–51.

[9] Skitmore M, Irons J, Armitage L. Valuation accuracy and variation: a meta analysis. Proc. from PRRES Conf. 2007, Pacific Rim Real Estate Society; 2007, p. 1–19.

[10] Tchuente D, Nyawa S. Real estate price estimation in French cities using geocoding and machine learning. Ann Oper Res 2022;308:571–608. https://doi.org/10.1007/s10479-021-03932-5.

[11] Durodola OD, Oluwatobi AO, Oni AA, Peter NJ. Factors Contributing to the Valuation of Arts and Artifacts in Ogun State, Nigeria. Int J Civ Eng Technol 2019;10:2224–31.

[12] Calhoun CA. Property valuation models and house price indexes for the provinces of Thailand: 1992-2000. Hous Financ Int 2003;17:31–41.

[13] Ćetković J, Lakić S, Lazarevska M, Žarković M, Vujošević S, Cvijović J, et al. Assessment of the Real Estate Market Value in the European Market by Artificial Neural Networks Application. Complexity 2018;2018:1–10. https://doi.org/10.1155/2018/1472957.

[14] Fan G-Z, Ong SE, Koh HC. Determinants of house price: A decision tree approach. Urban Stud 2006;43:2301–15.

[15] Gao L, Guo Z, Zhang H, Xu X, Shen HT. Video Captioning With Attention-Based LSTM and Semantic Consistency. IEEE Trans Multimed 2017;19:2045–55. https://doi.org/10.1109/TMM.2017.2729019.

[16] Demetriou D. A spatially based artificial neural network mass valuation model for land consolidation. Environ Plan B Urban Anal City Sci 2017;44:864–83. https://doi.org/10.1177/0265813516652115.

[17] Park B, Bae JK. Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert Syst Appl 2015;42:2928–34. https://doi.org/10.1016/j.eswa.2014.11.040.

[18] Gribniak V, Mang HA, Kupliauskas R, Kaklauskas G, Juozapaitis A. Stochastic Tension-Stiffening Approach for the Solution of Serviceability Problems in Reinforced Concrete: Exploration of Predictive Capacity. Comput Civ Infrastruct Eng 2016;31:416–31. https://doi.org/10.1111/mice.12183.

[19] El Hajj B, Schoefs F, Castanier B, Yeung T. A Condition-Based Deterioration Model for the Stochastic Dependency of Corrosion Rate and Crack Propagation in Corroded Concrete Structures. Comput Civ Infrastruct Eng 2017;32:18–33. https://doi.org/10.1111/mice.12208.

[20] Rafiei MH, Adeli H. A Novel Machine Learning Model for Estimation of Sale Prices of Real Estate Units. J Constr Eng Manag 2016;142. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001047.

[21] Kim G-H, Yoon J-E, An S-H, Cho H-H, Kang K-I. Neural network model incorporating a genetic algorithm in estimating construction costs. Build Environ 2004;39:1333–40. https://doi.org/10.1016/j.buildenv.2004.03.009.

[22] Brnabic A, Hess LM. Systematic literature review of machine learning methods used in the analysis of real-world data for patient-provider decision making. BMC Med Inform Decis Mak 2021;21:1–19.

[23] Manogaran G, Lopez D. Health data analytics using scalable logistic regression with stochastic gradient descent. Int J Adv Intell Paradig 2018;10:118–32.

[24] Chakraborty D, Elhegazy H, Elzarka H, Gutierrez L. A novel construction cost prediction model using hybrid natural and light gradient boosting. Adv Eng Informatics 2020;46:101201. https://doi.org/10.1016/j.aei.2020.101201.

[25] Raghavendra. N S, Deka PC. Support vector machine applications in the field of hydrology: A review. Appl Soft Comput 2014;19:372–86. https://doi.org/10.1016/j.asoc.2014.02.002.

[26] Estimation of Bank Profitability Using Vector Error Correction Model and Support Vector Regression. Econ Altern 2022;28:157–70. https://doi.org/10.37075/EA.2022.2.01.

[27] Schetinin V, Fieldsend JE, Partridge D, Coats TJ, Krzanowski WJ, Everson RM, et al. Confident interpretation of Bayesian decision tree ensembles for clinical applications. IEEE Trans Inf Technol Biomed 2007;11:312–9.

[28] Höppner S, Stripling E, Baesens B, Broucke S vanden, Verdonck T. Profit driven decision trees for churn prediction. Eur J Oper Res 2020;284:920–33. https://doi.org/10.1016/j.ejor.2018.11.072.

[29] Khaidem L, Saha S, Dey SR. Predicting the direction of stock market prices using random forest. arXiv 2016. ArXiv Prepr ArXiv160500003 n.d.

[30] Zhu J-M, Geng Y-G, Li W-B, Li X, He Q-Z. Fuzzy decision-making analysis of quantitative stock selection in VR industry based on random forest model. J Funct Spaces 2022;2022:1–12.

[31] Shang K, Yao Y, Li Y, Yang J, Jia K, Zhang X, et al. Fusion of Five Satellite-Derived Products Using Extremely Randomized Trees to Estimate Terrestrial Latent Heat Flux over Europe. Remote Sens 2020;12:687. https://doi.org/10.3390/rs12040687.

[32] Egwim CN, Alaka H, Toriola-Coker LO, Balogun H, Sunmola F. Applied artificial intelligence for predicting construction projects delay. Mach Learn with Appl 2021;6:100166. https://doi.org/10.1016/j.mlwa.2021.100166.

[33] Tsiapoki S, Bahrami O, Häckell MW, Lynch JP, Rolfes R. Combination of damage feature decisions with adaptive boosting for improving the detection performance of a structural health monitoring framework: Validation on an operating wind turbine. Struct Heal Monit 2021;20:637–60. https://doi.org/10.1177/1475921720909379.

[34] Ding W, Zhao X, Meng W, Wang H. Smart Evaluation of Sustainability of Photovoltaic Projects in the Context of Carbon Neutrality Target. Sustainability 2022;14:14925. https://doi.org/10.3390/su142214925.

[35] Guelman L. Gradient boosting trees for auto insurance loss cost modeling and prediction. Expert Syst Appl 2012;39:3659–67. https://doi.org/10.1016/j.eswa.2011.09.058.

[36] Xiao H, Liu Y, Du D, Lu Z. An Approach for Predicting the Costs of Forwarding Contracts using Gradient Boosting. 2022 17th Conf. Comput. Sci. Intell. Syst., IEEE; 2022, p. 451–4.

[37] Marvin G, Grbčić L, Družeta S, Kranjčević L. Water distribution network leak localization with histogram-based gradient boosting. J Hydroinformatics 2023. https://doi.org/10.2166/hydro.2023.102.

[38] Tamim Kashifi M, Ahmad I. Efficient Histogram-Based Gradient Boosting Approach for Accident Severity Prediction With Multisource Data. Transp Res Rec J Transp Res Board 2022;2676:236–58. https://doi.org/10.1177/03611981221074370.

[39] Chang Y-C, Chang K-H, Wu G-J. Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. Appl Soft Comput 2018;73:914–20. https://doi.org/10.1016/j.asoc.2018.09.029.

[40] Hou Y, Qin C. Contribution Analysis of Factors Affecting the Growth of Chinese Construction Enterprises Based on the XGBOOST Algorithm. Highlights Business, Econ Manag 2023;5:681–92. https://doi.org/10.54097/hbem.v5i.5258.

[41] Shi D, Zhang H, Guan J, Zurada J, Chen Z, Li X. Deep Learning in Predicting Real Estate Property Prices: A Comparative Study 2023.

[42] Renaud O, Victoria-Feser M-P. A robust coefficient of determination for regression. J Stat Plan Inference 2010;140:1852–62. https://doi.org/10.1016/j.jspi.2010.01.008.