International Journal of Construction Management

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjcm20

Efficient estimation and optimization of building costs using machine learning

T. Q. D. Pham, T. Le-Hong & X. V. Tran

To cite this article: T. Q. D. Pham, T. Le-Hong & X. V. Tran (2021): Efficient estimation and optimization of building costs using machine learning, International Journal of Construction Management, DOI: 10.1080/15623599.2021.1943630

To link to this article: https://doi.org/10.1080/15623599.2021.1943630

Published online: 02 Jul 2021.

INTERNATIONAL JOURNAL OF CONSTRUCTION MANAGEMENT
https://doi.org/10.1080/15623599.2021.1943630

Efficient estimation and optimization of building costs using machine learning

T. Q. D. Pham, T. Le-Hong and X. V. Tran

Institute for Development Strategies, Thu Dau Mot University, Binh Duong, Vietnam

ABSTRACT

This study provides a fast and accurate Machine learning (ML) and optimization framework, which allows a quick estimate for building costs, hence improving operational efficiency and competitiveness of a construction company. A dataset composed of 10,000 parametric building configurations, collected from end-to-end real-world activities in our partner company, was used to train and validate the ML models to perform multiple tasks. Among the 13 ML regression algorithms used, the Artificial Neural Network (ANN), Gradient Boosting, and XGBoost models appear to be the most suitable to estimate the building costs and the required resources with an accuracy of 99% within less than a second of the training time. The ANN models are also developed to identify available options of the building features under a given budget. The optimization problem under constraints is solved, helping clients determine the optimal building costs according to their preferences. Besides, the optimized building costs obtained by this study are 7% smaller than those of the actual data, hence to improve the company's competitiveness. This study showcases that ML models can be efficiently used in the construction sector to optimize the workflow for cost savings and provide some practical implications for data-driven management.

KEYWORDS

Building cost; maintenance cost; machine learning; regression analysis; neural networks; operational efficiency

Introduction

Optimizing operations to save engineering and design costs play a vital role for the company to propose the proposals with the least costs based on the information at the very early stage of project (Rardin and Rardin 1998). In the construction sector, most of large projects are procured through competitive contracting process (Tepeli et al. 2019), where clients issue request for proposals and contractors interested submit proposals with project solutions and the cost tender (Rardin and Rardin 1998; Tepeli et al. 2019). During the proposal preparation, many iterations between different design, engineering, resource estimation and costing departments occur (Sha'ar et al. 2017; García de Soto et al. 2019; Tepeli et al. 2019). In practice, only large size companies can have sufficient resources to prepare the proposals for bidding and the clients' requirements are often different from different projects, so it is not easy to standardize the data and share experience to apply in new projects (Sha'ar et al. 2017; García de Soto et al. 2019; Tepeli et al. 2019).

In contrast, our medium-sized partner company used to build for thousands of quite similar small projects such as few storey buildings where the owner often directly requests for proposal without a need to go through the formal procurement process. Also, the building is normally co-designed between the owner and the builder company to finalize the building features. Once an initial request for proposal is issued by the client, the key information on the planned building features, such as ground surface area, number of floors, number of windows, construction materials, etc., are collected. These data are then transferred to the design team, who will sketch plans of the future building, and to the engineering team to ensure safety in design. Later, required resources, such as materials and labour, are estimated by the company's internal services to prepare for storage and estimate associated costs that are construction and maintenance costs per unit surface. This total bare cost of construction and maintenance is then added with a margin rate, leading to the final cost estimation that is communicated to the client. Normally, the house is co-designed between the house owner and the building company, the process is repeated until the client approves the design for a given budget. During this process, the client often continues to change the requests, sometimes with minor modifications. One such change usually takes a lot of time and requires many interfaces between different company services, leading to complicated coordination work, inefficient operations as well as long waiting time for clients.

After years of services, our partner company has gathered a quite comprehensive database of about 10000 building configurations. This motivates them to seek for opportunities to save operating costs to improve efficiency and to reduce the waiting time during the co-design phase by exploiting these data. Our partner company identifies the following four tasks as ambitions:

1. The company needs to estimate the building costs quickly and accurately so that they can quickly communicate with the clients while assuring that only profitable cost estimations are proposed. This task will help to save engineering and design costs, to achieve competitive offered prices, and improve the customer relationship.
2. The company needs to be able to quickly identify the most realistic options of the planned building for a given budget to assure a profitable and competitive price and manage the client's expectations efficiently.
3. Once a set of building features is confirmed by the client, the required resource estimate needs to be made as quickly as possible by different services to prepare for adequate storage and efficient supply chain management. This task will

CONTACT X. V. Tran xuantv@tdmu.edu.vn

help to streamline operations, improve design and engineering cost savings as well as risk management efficiency.

4. Finally, in order to take into account the very diverse preferences of customers, an optimization tool under constraints needs to be developed to estimate the building costs in different scenarios.

It should be noted that these four tasks can be done and continuously improved by the current practices. This paper aims to demonstrate how a construction company can exploit the available data to perform the above-mentioned tasks with better efficiency.

Supervised machine learning (ML) models, based on self-learning algorithms using a set of training data, are widely used in different technology sectors (Kim et al. 2004; 2004; 2013). As discussed in detail in Section literature review, the application of ML for the building cost prediction using the available real-world engineering data from a construction company has not yet extensive in the construction management and operation applications (Kim et al. 2004; 2004; 2013). Also, it appears that a few papers that reported a systematic deployment of a dozen ML models to perform all the above-mentioned four tasks to identify their suitability and to provide some practical implications. More precisely, solving an inverse prediction model to determine the range of feature options under a given budget and optimization problems with constraints to save costs has not gained extensive interest. These problems are investigated in this paper in order to provide fast and accurate ML-based models that can be used to predict and optimize the building costs under different scenarios.

In this study, based on the actual data obtained from a long-established construction company, an ML-based estimation tool is developed to provide a fast and efficient estimation of the building and maintenance costs directly from the customer requests (end-to-end services). More precisely, the laborious and time-consuming steps performed by the back-end offices are to be integrated into an automatic tool that the front-end company can employ to quickly estimate the building costs to better discuss with customers. The ML-based model is expected to link the client requests and the final cost estimation within a short time, improving the client – builder relationship. Besides, the repeated designs, as well as resource estimations, can be avoided, leading to significantly enhanced work efficiency. In addition, for a set of feature constraints required by the customer preferences, the building cost needs to be minimized to increase the company's profits. Thus, the tool developed in this study will help the company save costs, increase revenues, and improve customer satisfaction.

In this paper, Section literature review is dedicated to a literature review. The data description, data processing techniques, and ML models are presented in Section dataset and methodology. In Section results and discussions, the results obtained from ML models performing the four tasks mentioned previously, including the cost prediction for a given set of building features, the identification of the possible building features for a given budget, the prediction of the required resources for a given set of building features, and the optimization of the building costs with and without constraints on building features are discussed in detail prior to conclusions in the last section.

Literature review

It should be noted that we focus on medium size construction company that is working on many relatively small projects such as few-storey buildings and there is constant and open interaction between the house owner and builder to finalize the building features. The purpose of the building cost prediction is to identify the end-to-end relationship between the building features (i.e., number of floors) requested by the customer and the building costs computed by the company to set the offered cost estimation. Focussing on the four tasks described in Section introduction, this section aims to review articles related to the building cost estimation using ML models.

Note that the objective of this study is to predict the building cost from the construction company's perspectives rather than the house price from the clients' ones. However, as the building costs and the house price are somehow interlinked, both these perspectives are reviewed. In the housing sector, ML is used to predict the house prices from the mortgage data based on house features, for example, locations, state, useful surface, garden … (Limsombunchai 2004; Park and Bae 2015; Gao et al. 2019; Madhuri et al. 2019), and mainly for the demand side once the houses are already built (Limsombunchai 2004; Gao et al. 2019). From the supply side of house building companies, advanced data analytics has recently gained interest. Le et al. (2019) performed an in-depth literature review of the civil integrated management data. They concluded that data sharing and integration were critical for successfully implementing integrated civil infrastructure management.

Ghosalkar and Dhage (2018) predicted the building cost using multiple linear regression ML algorithms. The three most important factors influencing the building cost were physical conditions, concepts, and location. Their research proved that the linear regression model predicted the building cost with a mean squared error of 0.3713, ensuring a good predictive model. Besides, Bhagat et al. (2016) developed a purely statistical framework to predict the efficient house pricing for real estate customers with respect to their budgets and priorities using a linear regression algorithm. Based on a dataset composed of 286 building configurations collected in the United Kingdom, Lowe et al. (2006) predicted the total client costs given the construction and client costs with an R^2 (coefficient of determination) value of 0.661 using multiple linear regression. Recently, Huang and Hsieh (2020) proposed a simple linear regression framework to predict the labour cost using 19 completed Building Information Modelling (BIM) projects from a leading construction company in Taiwan. Their results indicated that the linear regression model was the most stable, compared to the Random Forest model, with increasing the number of projects.

Regarding the nonlinear ML methods, Chen et al. (2017) adopted a novel approach based on the Support Vector Machine (Suykens and Vandewalle 1999) algorithm to predict residential house prices. Their results showed that the maximum accuracy (R^2 value) could reach 0.72. Phan (2018) developed a framework using a polynomial regression method to forecast house prices in Australia with an accuracy (R^2) of 0.84. Besides, the Artificial Neural Network (ANN) was also used in the construction sector, for example, to predict engineering service costs (Matel et al. 2019) or classify the delay risk assessment in tall building projects (Sanni-Anibire et al. 2020). These studies suggested that the ANN model can present a high accuracy of more than 93%.

As for the boosting-based ML algorithms, Čeh et al. (2018) estimated Random Forest's performance versus multiple linear regression's one when predicting the apartment price of 7407 data from 2008-2013. The authors concluded that the Random Forest outperformed the other regression method at all performance measures. The Random Forest has also been used to predict

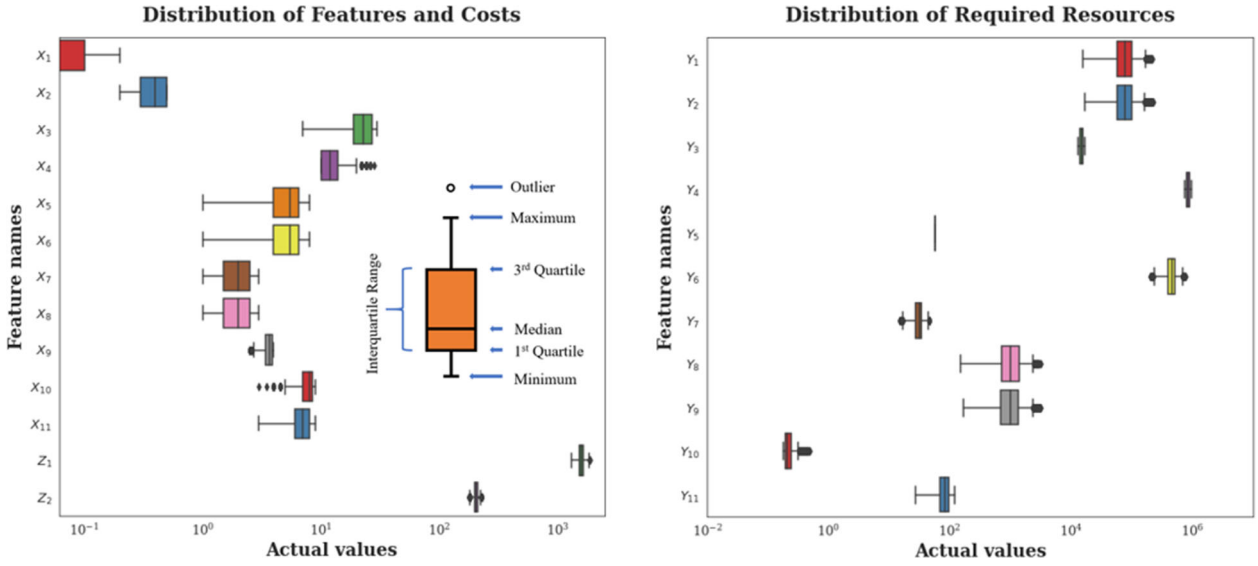

Figure 1. Statistical distributions of the variables *X*, *Z*, and *Y* from the actual dataset.

the building cost in many studies as in (Afonso et al. 2019; Hong et al. 2020; Jui et al. 2020). Several frameworks related to the building cost prediction were recently developed using the XGBoost method (Chen et al. 2015). Zhen Peng et al. (2019) used this technique to analyze 35,417 pieces of data captured by the Chengdu HOME LINK network. The results indicated that XGBoost prediction accuracy was the highest, with an R^2 value reaching about 0.93. In a similar study, Truong et al. (2020) concluded that XGBoost appeared to be the best algorithm to predict the house price.

It should be noted that the above-mentioned studies (Suykens and Vandewalle 1999; Lowe et al. 2006; Chen et al. 2015; Bhagat et al. 2016; Chen et al. 2017; Čeh et al. 2018; Ghosalkar and Dhage 2018; Phan 2018; Afonso et al. 2019; Matel et al. 2019; Peng et al. 2019; Hong et al. 2020; Huang and Hsieh 2020; Jui et al. 2020; Sanni-Anibire et al. 2020; Truong et al. 2020) mainly focussed on the estimation of the building costs using data from one or a few internal services without going through an end-to-end workflow from customer requests to the building price offered. Regarding cost optimization used to improve profits, few studies have been reported in the literature. Kravanja and Žula (2010) presented a simultaneous cost, topology, and standard cross-section optimization of industrial steel building structures using the mixed-integer nonlinear programming approach. Similarly, Michael J. Risbeck et al. (2015) developed a framework to optimize the combined building heating/cooling equipment cost. Besides, Vinko Lešić et al. (2017) proposed a modular energy cost optimization for buildings with an integrated microgrid. The results showed a considerable cost-saving of modular energy in different configurations.

Based on this review, it appears that the use of advanced data analytics for the real-world data collected from end-to-end activities of a construction company to perform the four tasks described in Section introduction has not been extensively studied. Also, most of the published works focussed on predictive models to estimate the costs from the building features. However, limited attention has been given to an inverse approach that identifies the features under a given budget. Besides, the optimization of the building costs performed in the literature was usually subjected to constraints on the input, which will be considered in this paper. Additionally, our study aims to demonstrate the suitability of different ML regression methods and optimization under constraints using real-world data from the end-to-end activities of a construction company in order to provide some practical implications for construction management.

Dataset and methodology

Data description

In this study, a dataset composed of 10,000 parametric building configurations collected from the end-to-end activities of a construction company was used to train, validate, and optimize ML models in order to estimate building costs and required resources. The dataset consists of 24 variables divided into three specific groups, including building features, costs, and required resources. The building features, denoted by the vector X, have 11 features named $X_1, X_2, \ldots, X_{11}$. They consist of key characteristics of a building, which are provided by the client, such as the number of floors, the building depth, the width of windows, the floor height, etc. The costs, denoted by the vector Z, have 2 components, Z_1 and Z_2, corresponding to the building and maintenance cost per unit surface, respectively. The required resources, denoted by the vector Y, are composed of 11 parameters, named $Y_1, Y_2, \ldots, Y_{11}$. Internal to the company, these variables represent the amount of cement and sand, etc., estimated by the company, which serve to calculate costs and prepare necessary resources for construction.

The actual dataset is included in the Appendix. This dataset is a matrix of 10,000 lines and 24 columns formed by the standardized data. The first 11 columns represent the independent variables of the building features X. The next two columns represent the costs Z, which are functions of the building features X. The last 11 columns are the required resources Y estimated by different company departments, which are also functions of X. Each line in the data represents a configuration of the building to be built and associated costs estimated beforehand and required resources. They were obtained across internal services and departments of the company.

Figure 1 shows the distributions of the variables X, Z, and Y from the actual dataset. Note that Figure 1 is presented in the logarithmic scale without normalization, and the negative values of X_1 are excluded. As can be seen in the figure, each feature

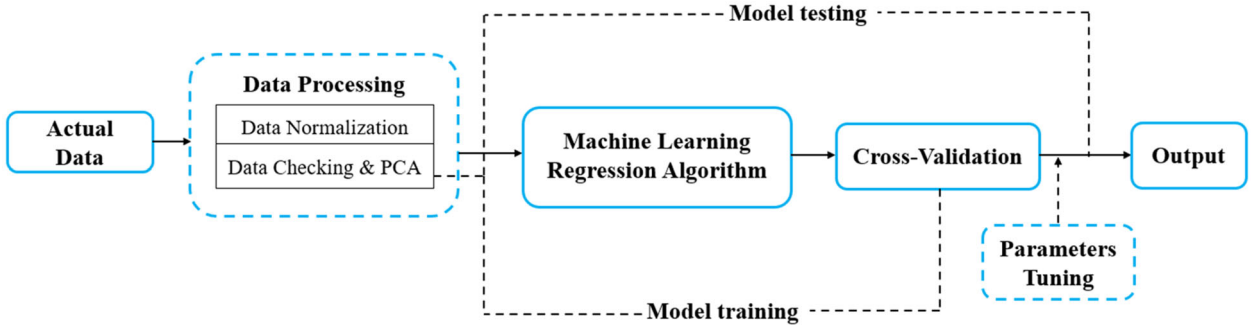

Figure 2. Workflow for the ML regression framework.

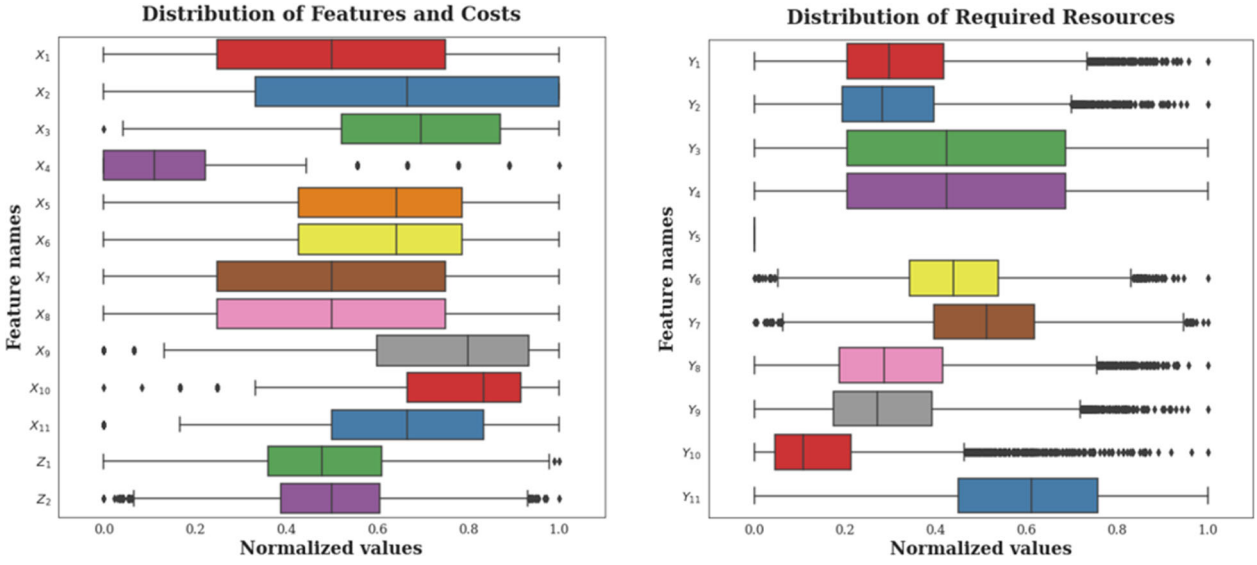

Figure 3. Distributions of the normalized values of *X*, *Z*, and *Y*.

exhibits a different distribution. These differences could be very sensitive and considerably affect the algorithms that consider the Euclidean distance, such as the K-means algorithm. Additionally, this significantly increases the ML model's training time, in particular that of the ANN.

Regression frameworks

Figure 2 represents the workflow for the ML regression framework used in this study. It includes data processing, ML regression algorithm, and model evaluation. The actual data, defined as those collected from the company without any modification, were first analyzed using several processing methods, such as data normalization, data checking, and Principal Component Analysis (PCA) (Wold et al. 1987). They were then split into training and test datasets with a ratio of 80% and 20%, respectively. The training dataset was used as the input for the ML models that were then validated using the test dataset. During this process, cross-validation and parameter tuning were performed to improve/maximize the model accuracy.

Data processing

- *Data normalization*
  As shown in Figure 1, due to the diverse intrinsic characteristics and properties of the features *X* from the actual data, their scales are relatively large with different orders of magnitudes and statistical distributions. This significantly slows down the computation process and negatively affects the model performance. In order to deal with these issues, the data of each column in the actual dataset were normalized in the range of 0 and 1 using the following equation:

$$X^{scaled} = \frac{X - X^{min}}{X^{max} - X^{min}}, \quad (1)$$

  where X^{scaled} is the normalized value of each variable of X from the actual dataset. X^{max} and X^{min} represent its actual minimum and maximum values. Similar normalization was performed on Y and Z variables. The distributions of the normalized data are shown in Figure 3. In this paper, the normalized values were used for training, testing, and validation processes, as well as for result illustration, except the optimization task under constraints presented in Section optimization of the costs Z under constraints on the building features X.
- *Data checking and Principal Component Analysis*
  Figure 4 shows the correlation coefficients between the building features X before and after applying the Principal Component Analysis (PCA) using Pearson correlation analysis. It can be seen that without the PCA, some features are strongly correlated, in particular between X_9 or X_{10} and other features with absolute values of the correlation coefficient of 0.4 or 0.5, as shown in Figure 4. The correlations

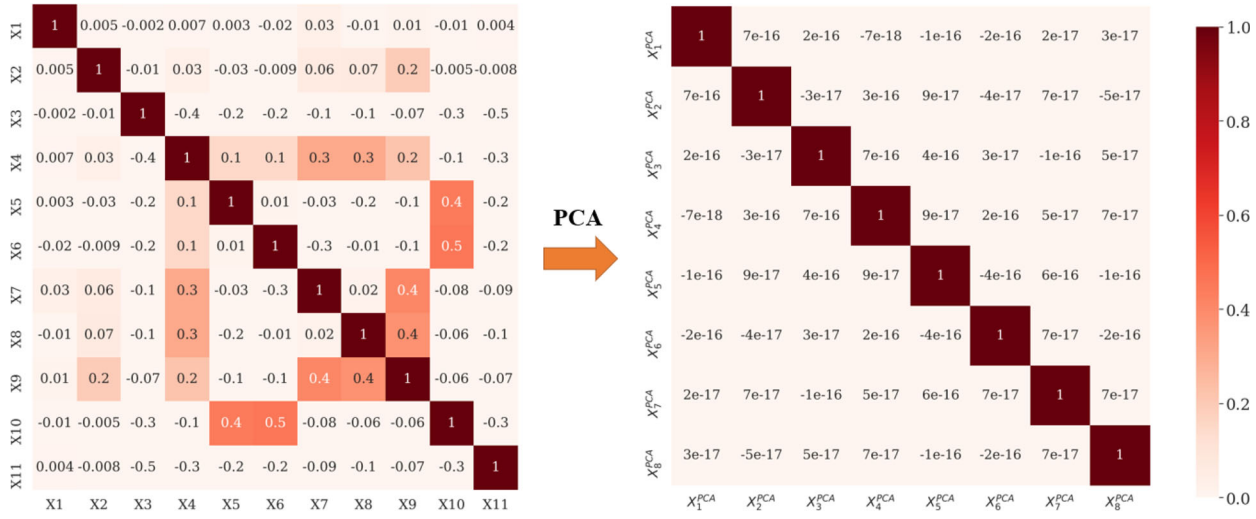

Figure 4. Correlation matrices for the building features X before (left) and after (right) applying the PCA by Pearson correlation analysis.

Table 1. Analysis results of the PCA method for the dataset (the values higher than 0.5 are marked in bold).

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
X_1	7e-5	2.1e-4	8.9e-4	1.1e-3	−8.1e-5	6.9e-3	1.9e-3	−1.2e-3
X_2	2.5e-4	7.8e-4	1.5e-3	−7.6e-4	5.1e-3	1.1e-3	0.02	−0.012
X_3	**−0.94**	0.23	−0.012	−5.6e-3	−0.12	0.01	0.10	0.15
X_4	0.28	**0.87**	0.021	−4.7e-3	−0.13	5e-3	0.044	0.28
X_5	0.073	0.086	**−0.55**	**0.7**	−0.37	−0.051	0.15	−0.11
X_6	0.079	0.077	**−0.55**	**-0.7**	−0.38	0.074	0.14	−0.12
X_7	0.014	0.055	0.082	0.061	0.19	**0.74**	0.49	−0.21
X_8	0.015	0.053	0.074	−0.065	0.21	**−0.66**	**0.61**	−0.19
X_9	5.9e-3	0.024	0.043	1.7e-5	0.11	0.042	0.37	−0.20
X_{10}	0.054	−0.049	−0.46	−4.1e-3	**0.52**	0.027	0.15	**0.68**
X_{11}	0.10	−0.039	0.32	3.2e-3	**-0.55**	0.033	0.38	**0.52**
Variance explained (%)	62	18	9	7	2	0.7	0.5	0.3

between features appear to be plausible as in the building design (i.e., a higher number of floors is strongly correlated with the lower floor depth). Note that for the predictive tasks using ANN and regression models, the features are not required to be non-correlated. Besides, each feature's contribution can be estimated via the linear regression coefficients.

Nevertheless, it is more difficult for an optimization process with correlated features to converge (Wold et al. 1987). In practice, the PCA (Wold et al. 1987) is employed to convert all the features into orthogonal ones with zero covariance by performing a change of basis on the data, reducing the number of correlated features. In this study, the PCA was only applied in the optimization task under constraints (see Section optimization of the costs Z under constraints on the building features X) in order to eliminate the correlations between features. Besides, the PCA is also capable of eliminating noise and improving the processing speed. As shown in Figure 4, no correlation (indicated by correlation coefficients of nearly 0) is observed amongst the building features after applying the PCA. Furthermore, the number of features is reduced from 11 to 8, which can significantly decrease the computing time.

In the PCA method, the converted feature $t \in R^L$ is generated from the actual data $x \in R^p$ via a linear transformation function $\Phi(x)$. This feature can be formulated as follows:

$$t = \Phi(x) = Wx + b, \tag{2-a}$$

where $W \in R^{pxL}$, $WW^T = I$ (*Identity matrix*), *and* $b \in R^p$. W denotes the eigenvectors of the covariance matrix corresponding to the first largest values. On the contrary, a reverse map is defined as $\psi: t \rightarrow x$ so that the actual data can be reconstructed from the PCA-converted data as follows:

$$\psi(t) = W^T(t - b), \tag{2-b}$$

The analysis results of the 8 principal components (PCs) from the PCA are shown in Table 1. Each principal component (PC1 to PC8) is the direction vector that best fits the data, and the variance explained represents the total variability of the data. Note that the values higher than 0.5 are marked in bold in the table. It can be seen that PC1 has a high impact on X_3 (number of floors). This component explains up to 62% of the total variance and contains the features that directly affect the building costs Z. PC2 has a high impact on X_4 (floor depth) and explains 18% of the total variance. As shown in the table, X_5 and X_6, corresponding to the building properties (i.e., floor height), are the features that contribute the most to PC3 and PC4. This suggests that PC3 and PC4 can represent the building properties. Besides, the number of facilities in the house (i.e., number of doors) X_{10}, X_{11}, show the highest impact on PC5 and PC8. Finally, PC6 and PC7 have a significant effect on

X_8. A more detailed analysis of each feature's influence on the building costs will be shown in Section prediction of the building and maintenance costs Z as a function of the features X and Section prediction of the required resources Y as a function of the building features X.

Machine learning regression algorithms

In this study, 13 ML algorithms were applied to identify the most suitable models to estimate the costs Z and the required resources Y with respect to the building features X. As a basic ML algorithm, regression is a training process to label the input independent variables with the output dependent variables. It becomes a supervised learning problem and can be solved using various ML algorithms. The next section will provide a brief description of three families of the ML regression algorithms.

- *Linear regression-based ML models*
  Linear regression-based ML model is a parametric and supervised learning algorithm that uses a linear approach for a prediction problem. In the context of this study, the values of the independent features X are associated with those of the dependent variables Z by a linear relationship as follows:

$$Z = \ b_0 + \ b_1X_1 + b_2X_2 + \ \ldots \ + \ b_{11}X_{11} \tag{3}$$

  As mentioned above, Z may be either the building cost per unit surface Z_1 or the maintenance cost per unit surface Z_2. The total budget (Z_1+Z_2) is also considered. Four linear regression-based algorithms, namely Linear Regression (Seber and Lee 2012), Lasso Regression (Tibshirani 1996), Ridge Regression (Hoerl and Kennard 1970), and Elastic-Net Regression (Zou and Hastie 2005), were tested in this study.
- *Nonlinear regression-based ML models*
  The nonlinear regression-based ML models θ are extended versions of the linear regression-based ML models, which can be expressed as follows:

$$Z = \theta(X_1, X_2, \ \ldots, X_{11}) \tag{4}$$

  In this method, the data are fitted using a method of successive approximation for the model's hyperparameters. Decision Tree (Safavian and Landgrebe 1991), K-Nearest Neighbour (Keller et al. 1985), Support Vector Machine (Suykens and Vandewalle 1999), and Artificial Neural Network (Jain et al. 1996) were tested in this study.
- *Boosting regression-based ML models*
  Boosting regression-based ML models or Ensemble regression models aim to obtain a simple function with small residuals at almost every point in a sufficiently large sample. In this study, four models, namely AdaBoost (Roe et al. 2005), Extra Tree (Maier et al. 2015), Random Forest (Aldous 1993), Gradient Boosting (Friedman 2002), and XGBoost (Chen et al. 2015), were tested.

Evaluation metrics

In order to assess the performance of the building cost prediction model using ML, two error metrics, including Mean Squared Error (MSE) and coefficient of determination R^2, were used in this study. The MSE measures how close the estimated data are relative to the actual data by calculating the average of the squared deviation between the estimated values (obtained from the ML model) and the actual values by the following formula:

$$MSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i \ - \ y_i)}{n}}, \tag{5}$$

where n, $\hat{y}$, and y are the number of samples, the predicted values from the regression, and the actual values, respectively. Hence, the lower the value of MSE, the better the model predicts.

The coefficient of determination R^2 represents a measure of how much the model replicates the actual values on the basis of the set of various errors:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{6-a}$$

$$SST = \sum_{i=1}^{n} (y_i - \overline{y})^2, \tag{6-b}$$

$$R^2 = 1 - \frac{SSE}{SST}, \tag{6-c}$$

where SSE, SST, and $\overline{y}$ represent the residual sum of squares, the total sum of squares, and the mean of the actual values, respectively. Thus, the closer to 1 the value of R^2 is, the better the model predicts and the higher its accuracy.

Genetic algorithm for optimization models

A Genetic Algorithm (Goldberg 2006) is a computational method for solving constrained and unconstrained optimization problems, which is based on the natural selection process that mimics biological evolution. This technique repeatedly modifies a population of individual solutions, which is the input that minimizes the optimization function, to search for the optimal one created through the propagation of positive traits and mutation. The evaluation of each solution is rated by a target function named fitness function; only the solutions with the highest fitness function score are allowed to move on to the next generation after a generation. The whole algorithm can be defined as a 4-step process: population representation and initialization, objective and fitness functions, selection and crossover, and mutation, as shown in Figure 5.

Results and discussions

In this section, the results of the ML models for different tasks are presented. First, an overview of the four tasks described in Section introduction and inspired by real-world scenarios is discussed to remind the ML models' goals. Then, the results for these four tasks are detailed. Finally, practical implications and contributions to construction management are discussed.

To develop ML-based models in this study, Python script codes were implemented using several Python libraries, such as Numpy (numerical computing tool) (Van Der Walt et al. 2011), Scipy (scientific and technical computing) (Jones et al. 2001), Matplotlib (visualization) (Hunter 2007), Pandas (data loading and data-driven) (McKinney 2011), Scikit-learn (library for ML-based regression models) (Pedregosa et al. 2011), and Keras (library for the artificial neural network) (Gulli and Pal 2017). The training time was obtained from an Intel Core I5 Laptop equipped with 6 GB RAM and 2 CPU cores.

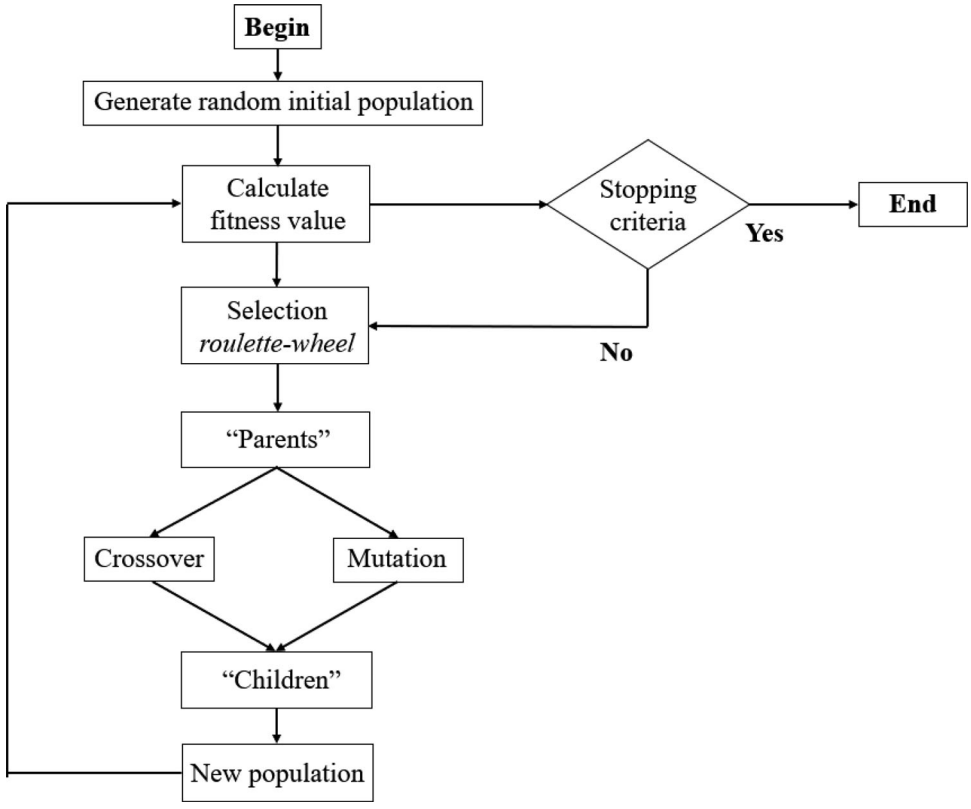

Figure 5. Genetic algorithm framework for the building cost optimization with and without constraints.

Overview of tasks performed by machine learning models

In this section, four main tasks, previously described in Section introduction, were performed using ML-based models. The first task was to predict the building cost per unit surface (Z_1) and maintenance cost per unit surface (Z_2) for a given set of features X. This task aims to provide the company with a fast and accurate tool to estimate the building costs based on the actual dataset. The second task was to identify potential options of the building features X that a client could afford for a given cost (budget) Z ($= Z_1 + Z_2$). This task aims to guide the company to the available building options to communicate with the client effectively. The third task was to predict the required resources Y for a given set of features X so that all necessary materials and labours can be stocked and planned before construction. This task aims to ensure that the risks of delays due to resource shortage can be limited, and the costs can be accurately estimated beforehand. The fourth task was to minimize the costs Z under a given set of constrained features X according to the clients' preferences. This task aims to ensure that the company's competitive but profitable prices are always offered to the client.

Prediction of the building and maintenance costs Z as a function of the features X

In order to evaluate the performance of the selected ML-based models described in Section machine learning regression algorithms, the dataset was split into two parts for training and testing. The test dataset was used for the model validation using the R^2 metrics in Eq. (6-c). For the Boosting method, 2000 estimators were used during aggregation to maximize the model performance. The ANN model was built from three layers with 11 nodes in the first layer corresponding to the input features, five nodes in the second layer named hidden layer, and one node in the last layer representing the costs. The ANN model architecture is shown in Figure 6. The model hyperparameters, such as the number of neurons in the hidden layers, the number of hidden layers, the activation function, were selected using the Bayesian optimization algorithm from the scikit-learn library (Aldous 1993).

Table 2 lists the R^2 metrics obtained from 13 ML regression algorithms mentioned in Section machine learning regression algorithms. It can be seen that the Boosting regression-based models and ANN model exhibit the best accuracy when performing the cost prediction, in particular Gradient Boosting, XGBoost, and ANN with a very high-value R^2 of 0.99. In other words, the building and maintenance costs, as well as the total cost, can be predicted very well for a given set of building features X. This observation is expected and can be explained by the fact that the Boosting regression techniques and ANN model use the gradient descent to minimize the error function. However, these methods need much more time to converge as compared to the others. Regarding the linear regression-based models, except the multiple linear regression that can reach 0.94 of R^2 after only about 0.002 s of the training time, the Lasso, Ridge, and Elastic-Net regression models show a relatively low value of R^2. This lack of accuracy may be due to the high correlation between the variables. In this case, these algorithms will only retain one variable and set the other correlated variables to zero. Thus, the dataset will lose information resulting in lower model performance. It can be noticed that all the linear regression models can converge rapidly after only 0.002-0.003s. This can be straightforwardly explained by the fact that solving a linear equation is much easier and less time-consuming than a nonlinear Boosting equation. In terms of accuracy and training time, the ANN model appears to be the

most suitable to predict the building costs, as shown in Table 2.

Figure 7 shows the actual and predicted results using the linear regression and ANN models with cross-validation. Note that cross-validation was only performed for the linear regression and ANN models due to their best quality in both training time and accuracy. In the figure, the colour bar on the right side represents the density of points. Since the linear regression shows a lower R^2 than the ANN model, cross-validation presents the same trend in the 3 cases Z_1, Z_2, and $Z_1 + Z_2$, but with more scattering around the $y = x$ line (dashed red line in Figure 7) where x and y represent actual and predicted values of Z, respectively.

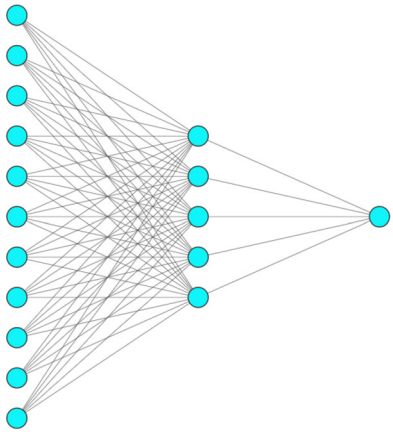

Figure 6. Architecture of the Artificial Neural Network for task 1.

Table 2. Values of R2 for different ML-based regression models.

	R^2			Training time (s)		
Regression model	Z_1	Z_2	(Z_1+Z_2)	Z_1	Z_2	(Z_1+Z_2)
Multiple Linear	0.94	0.94	0.94	0.002	0.003	0.003
Lasso	0.13	0.12	0.13	0.003	0.003	0.004
Ridge	0.12	0.13	0.13	0.003	0.003	0.004
Elastic-Net	0.15	0.13	0.13	0.003	0.003	0.003
Decision Tree	0.95	0.83	0.9	0.03	0.03	0.03
KNN	0.82	0.79	0.8	0.3	0.3	0.4
SVM	0.93	0.91	0.97	0.09	0.1	0.25
AdaBoost	0.92	0.86	0.89	14.87	14.9	16.12
Gradient Boosting	0.99	0.99	0.99	12.08	12.0	12.86
Extra Tree	0.97	0.92	0.95	48.9	48.2	49.5
Random Forest	0.97	0.92	0.95	44.94	46.3	47.7
XGBoost	0.99	0.99	0.99	16.87	18.21	18.62
ANN	0.99	0.99	0.99	2.79	2.83	2.94

As shown in Figure 7, the ANN model's predicted results are in excellent agreement with the actual data. As a result, the ANN was used to perform the remaining studies in this paper, thanks to its accuracy and fast training time. This result helps to establish the link between client requests and final cost described in Figure 2 within a few seconds instead of many days of work performed by different services of a construction company as in current practice. This can result in a more efficient and smoother interaction between company and clients, enhancing customer satisfaction. Simultaneously, the costs incurred by the internal services can be avoided; the company's experience and knowledge are streamlined and exploited, leading to a better estimate for the building costs.

In order to investigate the influence of each building feature on the building and maintenance costs, the linear regression model was used to find the parameters with the highest coefficients b_i as shown in Table 3. Note that the actual data were normalized using Eq. (1); thus, the coefficients b_i are called as the normalized regression coefficients. According to statistical theory, a coefficient with the highest value is the most influential parameter. Also, a positive coefficient indicates an increasing contribution of the feature to the output, while a negative coefficient tends to decrease the output as the input feature increases.

As listed in Table 3, among 11 features, X_4 and X_9 exhibit the highest regression coefficient values in absolute terms, leading to the most influence on the building cost per unit surface Z_1. Indeed, the X_4 and X_9 correspond to the floor depth and the floor height, respectively. Thus, it is reasonable that these key geometrical building parameters can directly increase the building cost. Whereas, in the case of the maintenance cost per unit surface Z_2, according to the regression coefficient values shown in Table 3, the two most important features are X_9 (floor height) and X_{10} (kitchen width). Similar to the building cost Z_1, it appears that X_4 and X_9 are the two most influential features in the case of the total cost (Z_1+Z_2). These results are beneficial for the sales representatives when choosing the appropriate features to meet as much as possible the client's requests with a

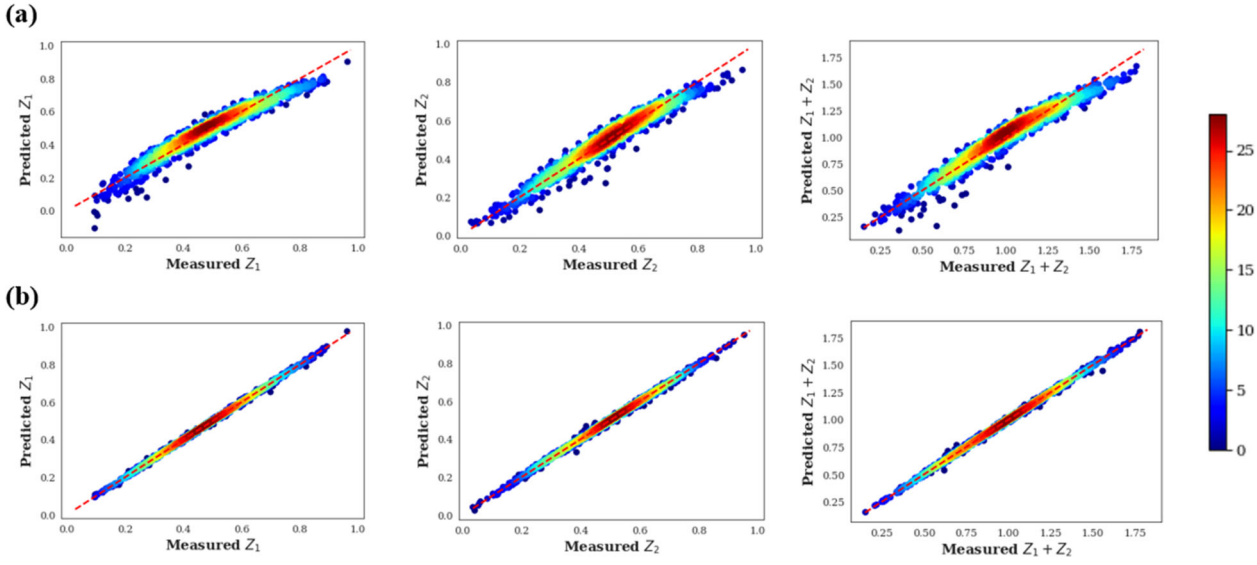

Figure 7. Scatter plot of the cross-validation result for (a) the linear regression models, and (b) the ANN models (the dashed red lines represent the $y = x$ line).

Table 3. Normalized regression coefficients of the linear regression models.

Case ($Z=$)	Normalized regression coefficients b_i ($i=1 \ldots 11$)										
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}
Z_1	−0.007	0.14	0.28	−0.82	0.11	0.10	0.07	0.07	0.43	−0.009	0.26
Z_2	0.06	−0.02	−0.21	−0.22	0.26	0.05	0.16	0.02	0.57	−0.39	−0.24
Z_1+Z_2	0.05	0.12	−0.33	−1.65	0.37	0.16	0.23	0.09	1.01	−0.40	−0.18

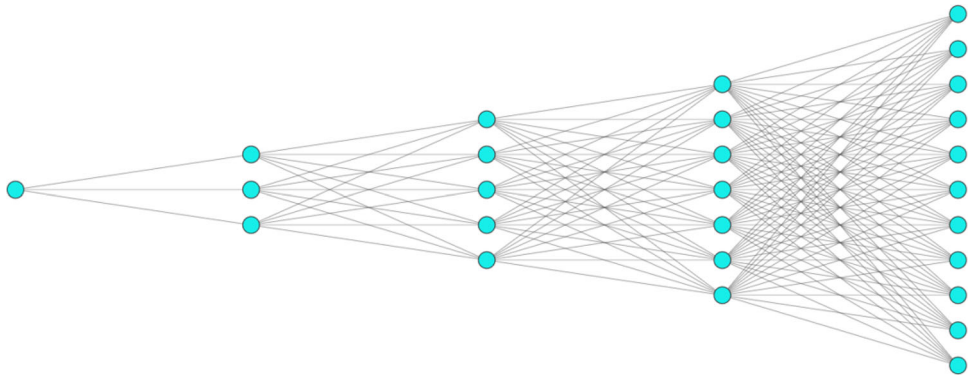

Figure 8. Architecture of the Artificial Neural Network used for Task 2.

reasonable price. Also, the client and seller can quickly identify the critical features that need to be modified, but always under a given budget. The customer's waiting time to receive the quotes from the changes can be minimized, improving customer satisfaction.

Identification of the features X for a given value of the cost (budget) Z

This task aims to perform a multi-output regression with only one independent variable (input). Three types of input variables were investigated, including the building cost per unit surface Z_1, the maintenance cost per unit surface Z_2, and the total cost (Z_1+Z_2). The ANN regression model was employed for this task because of its high quality in training time and accuracy, as shown in Section prediction of the building and maintenance costs Z as a function of the features X. In addition, it appears that the ANN model is the most suitable for this task since there is only one input variable with multiple outputs. The ANN architecture was selected as 1-3-5-7-11, and it contains five layers densely connected to one another, as shown in Figure 8. The first layer has one node representing the single input. The final layer represents the 11 features corresponding to the input features of Task 1.

The Relu activation function (Li and Yuan 2017) and Adam optimizer (Kingma and Ba 2014) were used in the inverse regression model for this task. The loss computed by the Mean Squared Error (MSE) for training and validation (see Eq. (5)) is shown in Figure 9. Note that an epoch is defined as one cycle when the network looks through the full training dataset. It can be seen that the models rapidly converge after about 150 epochs. Additionally, no significant difference in the loss can be observed for three types of costs Z_1, Z_2, and (Z_1 + Z_2) after model convergence.

Table 4 shows that the values of MSE obtained by the ANN regression model can reach only about 0.0577, 0.058, and 0.0638 for Z_1, Z_2, and (Z_1+Z_2), respectively. For the training time, the models can quickly converge after only a few tenths of seconds. Besides, there is no significant difference among the three cases. The result indicates that the ANN model is capable of predicting the inverse task consisting of a set of building features (X_1 to X_{11}) as the outputs with one input in a short period of time. In practice, this information can allow the company to quickly select each feature's value for a given budget, then communicate it to the client. For example, under a given amount of budget, the model is able to claim that the client can afford a building with a specific number of floors and windows, etc.

Prediction of the required resources Y as a function of the building features X

The required resources Y are critical information for company operation, which can be used to prepare needed resources and estimate the costs for a construction project. Table 5 lists the regression results for two cases. Similar to Tasks 1 and 2 shown in Sections prediction of the building and maintenance costs Z as a function of the features X and identification of the features X for a given value of the cost (budget) Z for the costs Z, the first case consists of predicting the required resources Y from the input features X. The inverse case aims to determine the building features X as a function of the input variables corresponding to the required resources Y. Contrary to Task 1 (see Section prediction of the building and maintenance costs Z as a function of the features X) in which the output contains only one variable, the first problem investigated in this part consists of a multi-output regression task, knowing the number of the required resources is 11, similar to the input features X. The first problem can be summarized as follows:

$$+ \text{Train the dataset: } \{(X_1, X_2, \ldots, X_{11}), (Y_1, Y_2, \ldots, Y_{11})\}, Y_i \in R^{10000\times 1}$$

$$+ \text{Predict the vector } \boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{11}) \textit{ for a given } X.$$

It is observed that the R^2 values, describing the model accuracy, obtained by the ANN models for the two problem cases can reach 0.99 and 0.985, as shown in Table 5. This result is similar to that of the one-output regression in Task 1 for the costs Z (see Section prediction of the building and maintenance costs Z as a function of the features X). Due to the complicated structure

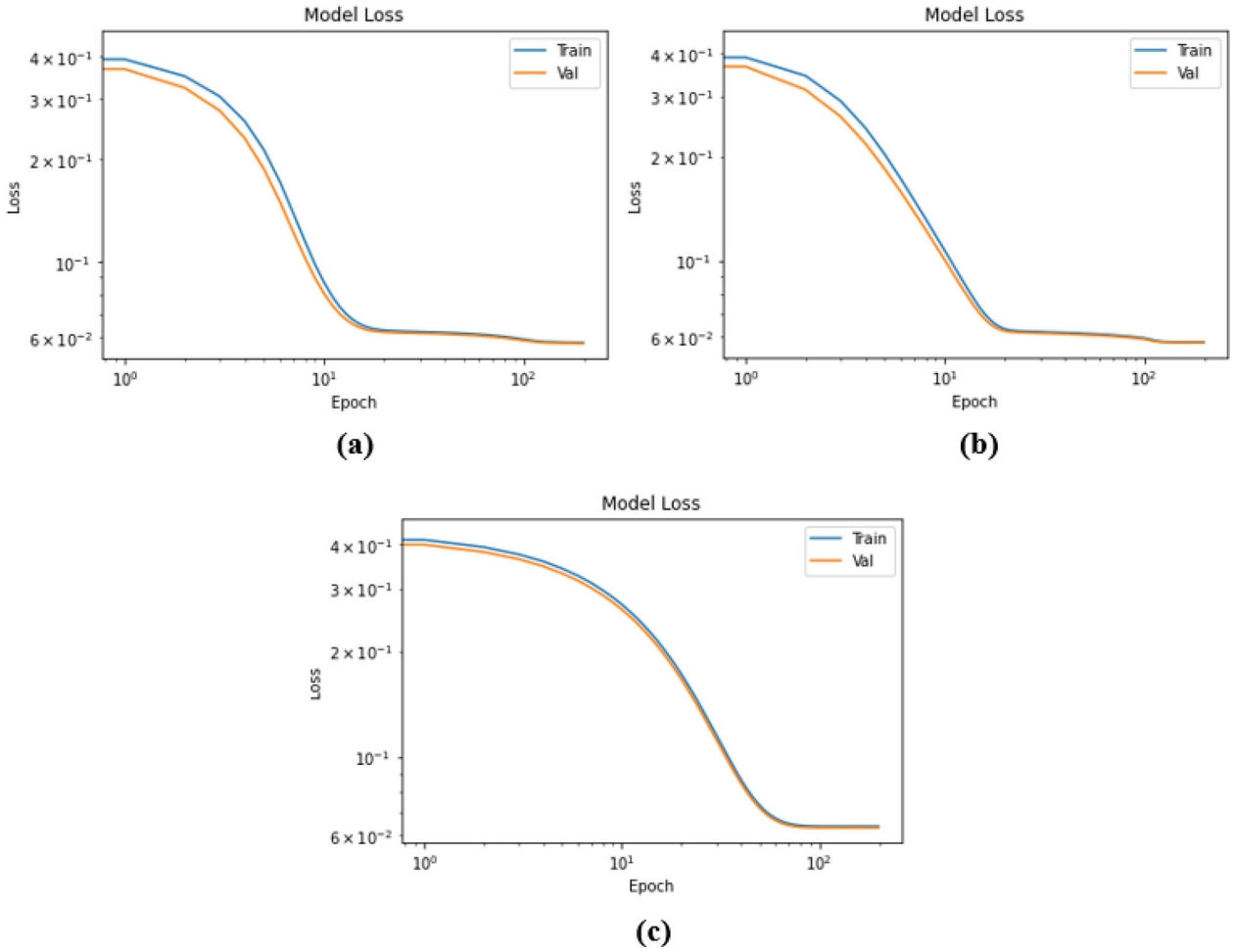

Figure 9. Epoch-dependent MSE loss of the training and validation for (a) Z_1, (b) Z_2, (c) (Z_1+Z_2).

Table 4. MSE and training time obtained by the ANN model for task 2: identification of the features X as a function of the costs Z.

Input case	Mean squared error (MSE)	Training time (s)
Z_1	0.0577	19.7
Z_2	0.0580	20.3
Z_1+Z_2	0.0638	17.6

Table 5. ANN model's results for two regressions tasks: required resources *Y* vs. building features *X*, and the inverse problem.

	R^2	Training time (s)
Prediction of the required resources *Y*	0.990	21.2
Prediction of the building features *X*	0.985	23.4

of the ANN, this method significantly increases the training time, i.e., 21.2 s and 23.4 s (see Table 5). In practice, these results are expected to help the engineers reduce the time for calculating the number of resources needed to build the corresponding for a given set of features, improving operational efficiency.

Optimization of the costs Z under constraints on the building features X

For this task, the building cost per unit surface Z_1 and the maintenance cost per unit surface Z_2 were minimized under constraints on certain building features using the ANN model and the Genetic Algorithm (GA) described in Section regression frameworks. These constraints represent the features imposed by clients' preferences, which may generally increase the minimum costs as compared to the case without constraints. As discussed previously in the literature review section, the GA, which is currently one of the most popular optimization methods used to solve the global minimum problem, was chosen for this task.

In the GA, the population contains 11 components with a total size of 100. In practice, the total size strongly depends on the complexity of the dataset and the problem/model to be solved; it can vary from a few tens for a simple linear regression to a few thousand for ML-based models with highly complicated structure. Note that, in this study, the total size was chosen by following a trial-and-error process: (i) starting from a low population; (ii) increasing the population progressively; (ii) monitoring the improvement of the GA algorithm in both convergence time and performance. The minimization algorithm was then performed under 100 generations in order to obtain the global minimum of *Z*. For this particular purpose, as discussed in Section data processing, the features *X* were standardized with unit-variance and zero-mean using the following formula:

$$X^{scaled} = \frac{X - \overline{X}}{\sigma} \tag{7}$$

where $\overline{X}$ and σ are the mean and standard deviation of the actual data before standardization. The maximum and minimum values of the building features after standardization are listed in Table 6. As discussed previously in Section data processing,

Table 6. Maximum and minimum values of the building features X after standardization.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}
Max	1.55	1.21	1.47	5.42	1.69	1.68	1.52	1.53	1.05	1.10	1.28
Min	−1.54	−1.97	−3.27	−0.93	−2.46	−2.47	−1.92	−1.94	−3.24	−4.41	−2.81

Table 7. Optimal building features X for three types of costs.

			X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	Z^{min}
Case 1	$Z=Z_1$	Real value	1.55	−1.97	−1.13	1.89	1.40	1.38	−0.19	−0.20	−2.09	1.10	−0.79	0
		Prediction	0.8	−1.49	−1.75	3.44	2.23	0.92	−0.53	−0.95	−2.64	−0.45	−0.85	−0.12
	$Z=Z_2$	Real value	−1.15	1.21	1.47	−0.93	0.50	1.38	−1.92	−1.07	−3.24	1.10	−0.76	0
		Prediction	−0.11	−0.20	−0.16	−0.02	0.25	1.02	−1.50	−0.48	−1.59	0.19	0.04	−0.06
	$Z=Z_1+Z_2$	Real value	1.16	−1.97	−2.24	1.89	1.69	1.38	−0.19	−0.21	−2.1	1.1	0.59	0
		Prediction	0.3	−1.62	−2.70	3.14	4.04	−0.70	−0.45	−1.50	−2.89	−0.29	0.25	−0.08
Case 2	$Z=Z_1$	Real value	−0.77	−0.91	0.02	0.48	0.21	0.19	−1.06	−1.07	−2.67	−1.19	0.59	0.11
		Prediction	0.22	−0.47	−0.74	1.71	1.67	0.62	−0.39	−1.10	−1.80	0.04	−0.73	0.04
	$Z=Z_2$	Real value	−0.77	−0.91	0.02	0.48	0.21	0.19	−1.06	−1.07	−2.67	−1.19	0.59	0.04
		Prediction	0.21	−0.86	−0.76	1.33	1.63	0.61	−0.18	−1.36	−1.74	0.25	0.53	0.03
	$Z=Z_1+Z_2$	Real value	−0.77	−0.91	0.02	0.48	0.21	0.19	−1.06	−1.07	−2.67	−1.19	0.59	0.09
		Prediction	0.08	−0.67	−1.67	1.96	2.24	0.46	−0.01	−1.12	−1.58	0.41	−0.24	0.03
Case 3	$Z=Z_1$	Real data	0.01	0.15	−0.17	0.48	−0.38	−0.10	−1.06	−0.21	−2.09	−1.19	−0.08	0.17
		Prediction	0.23	−0.77	−0.65	1.53	1.68	0.05	0.24	−0.39	−0.70	0.34	−0.22	0.16
	$Z=Z_2$	Real data	0.01	0.15	0.43	0.48	−0.38	0.19	−0.19	−1.07	−2.09	−1.19	−0.08	0.14
		Prediction	0.08	−0.67	−1.67	1.96	2.24	0.46	−0.01	−1.12	−1.58	0.41	−0.24	0.08
	$Z=Z_1+Z_2$	Real value	0.01	0.15	−0.17	0.48	−0.38	−0.10	−1.06	−0.21	−2.09	−1.19	−0.08	0.15
		Prediction	0.03	−0.48	−1.57	1.91	1.93	0.18	0.34	−0.28	−0.59	0.49	−0.35	0.13

some building features are highly correlated to one another, the Principal Component Analysis (PCA) was used in this task.

In this study, three ranges of features X were selected, corresponding to 100% (case 1, $X^{min} \leq X \leq X^{max}$), 80% (case 2, $1.1\times X^{min} \leq X \leq 0.9\times X^{max}$), and 60% (case 3, $1.2\times X^{min} \leq X \leq 0.8\times X^{max}$) of the minimum and maximum values of the actual dataset, as listed in Table 7. Case 1 can be considered equivalent to a non-constrained optimization problem. Note that this section aims to demonstrate that an optimization problem under constraint can help to take into account customers' preferences. For the sake of simplification, the building features were obtained by eliminating the top and bottom 10% quantiles from the dataset for case 2, and 20% for case 3, to facilitate discussions. For customers' world-real requirements, the constraints may be different, and solving these real requirements is straightforward.

For this task, the fitness function was obtained by the output of the ANN model that has been developed in Section prediction of the building and maintenance costs Z as a function of the features X. The ANN model took 11 building features as the input (X_1 to X_{11}). The output consists of three types of building costs Z. As a consequence, three Genetic Algorithm frameworks were separately developed to consider three types of building costs, including the building cost per unit surface Z_1, the maintenance cost per unit surface Z_2, and the total cost (Z_1+Z_2). The GA algorithm was performed for 100 generations corresponding to the initial population's size. Besides, the crossover is defined as the average of each individual, and the mutation is the addition of a random value from the range [-0.1, 0.1] to each individual.

Using the PCA, the number of building features was reduced from 11 to 8, as described previously in Section data processing, leading to a new matrix of 10,000 rows with 8 columns. This matrix was then multiplied by the inversion of the conjunction matrix of 11 rows × 8 columns, transforming the building features back into the original data matrix of 10,000 rows × 11 columns. Note that the lowest value of the actual cost after normalization is 0; thus, the obtained actual cost after optimization is expected to be negative in Case 1.

For case 1, the optimized values are equal to −0.12, −0.06, and −0.08 for Z_1, Z_2, and (Z_1+Z_2), respectively. Besides, the actual normalized minimum costs are 0 for all three cases. After reconversion into their original space of 11 dimensions, the minimum costs are reduced by 7%, 1%, and 1% for Z_1, Z_2, and Z_1+Z_2, respectively. Regarding the optimal values of the building features for the costs, the value of the feature X_3 is reduced as compared to its actual value. Besides, the feature X_4 increases in comparison with its actual value. Note that X_3 denotes the number of floors that can be considered as one of the most important building features since a lower value of X_3 can lead to a lower building cost. In addition, when the number of floors X_3 decreases, the depth of the building X_4 should increase to ensure its balance (negative correlation between these two features as shown in Figure 6). That is why the value of X_4 increases compared to its actual value in three types of costs. In brief, using the ML-based optimization, the minimum costs can be reduced by 7% at best as compared to its actual value. This finding helps the company identify the optimal features for more profitable prices.

For case 2, the range of the features was reduced by removing the 10% of maximum and minimum values from the dataset. In this case, the actual normalized minimum costs become 0.11, 0.04, and 0.09 for these three types of costs, respectively. The optimized values are 0.04, 0.03, and 0.03 for three costs Z_1, Z_2, and (Z_1+Z_2), respectively. As listed in Table 7, the optimization process can reduce up to 6%, 1%, and 4% of the minimum costs for Z_1, Z_2, and (Z_1+Z_2) as compared to the actual values. Similar to case 1, a lower value of X_3 and a higher value of X_4 can lower the costs. Different from case 1, in case 2, X_8 and X_{11} are decreased compared to their real data, while the others are increased. This may be due to the high device range to maintain in the building, knowing that X_8 and X_{11} correspond to the number of facilities. As a consequence, the building cost will be decreased.

Another range of the boundary value was considered in case 3, in which the building features were limited to 80% of the maximum and 120% of the minimum. Similar to the other cases,

the optimization result shows that a decrease in X_3, X_8, and X_{11} can reduce the three types of costs, as listed in Table 7. Besides, the optimization algorithm reduces 1%, 5%, and 2% of the actual minimum costs for Z_1, Z_2, and (Z_1+Z_2). It is noted that the (Z_1+Z_2) value is the total budget that the client needs to pay for owning the building. The result obtained by the optimization process will help the client select the most affecting variables to reduce the building cost according to his/her budget. Similarly, the company can quickly identify the critical features to change according to the client's feature constraints. All these functions will help to improve operational efficiency and customer satisfaction, and save costs.

In summary, about 1 to 7% of three types of costs can be reduced in comparison with the actual values using the Genetic Algorithm with the fitness function calculated from the Artificial Neural Network. Furthermore, it shows that the more the features are constrained, the less the minimum cost reduction can be achieved. This result may be explained by the fact that the minimum value is searched for within a lower space in the case of constraints than in the space without constraints. In case 1, the highest reduction in the minimum cost is observed in Z_1. However, in case 2 and case 3, Z_2 is the most reduced by the optimization process. Practical constraints by an expert will be provided in the implementation phase of this project.

Practical implications and contributions to construction management

As discussed in the previous sections, this paper aims to demonstrate how ML could be used in a construction company to exploit real-world operational data. Some practical implications can be drawn upon this study as follows:

- In this paper, four tasks are identified by asking the right and interesting questions stemmed from real-world activities. This process is of great importance to assure that the dataset and suitable algorithms can be selected to deliver the expected values of the data projects. Identifying regularly the pain points and opportunities for improvement can help the company figure out the tasks with potential good return of investment.
- It is shown in this paper that once a sizable dataset of high quality from end-to-end activities is available, ML is a powerful tool to streamline operations, save design and engineering costs, and improve customer experience. Also, the self-improvement of the ML-based models will help to integrate skills and knowledge of different services within the company to better predict the building costs when the models are implemented and more data are collected. Data collection plays a vital role in any data-driven project. The company should adopt a centralized and integrated approach to assure that both quality and quantities of data across different services are collected, shared, and exploited.
- The paper provides a systematic application of 13 ML models and employs different optimization techniques in order to perform the four tasks stemmed from real-world activities. Not only the hyperparameters of the ML models should be carefully computed, but also the accurate interpretation of the prediction and performance of the results obtained from ML models is essential.
- The fourth task involving the optimization of the costs (building and maintenance) Z under constrained building features X can be extended to the constraints on the required resources Y as both variables Y and Z are functions of X. This task represents an important mission to plan potential scenarios for risk management, especially when facing the world of uncertainty such as the labour shortage due to the pandemic or the lack of certain materials resulting from supply chain disruption.

It should be noted that the above implications may only be valid for medium sized partner company who used to build for thousands of relatively similar small projects such as few storey buildings where the owner often directly contacts for a proposal without a need to go through the formal procurement process and where the building is co-designed by the owner and the building company.

Conclusions

This study provides a fast and accurate Machine learning (ML) and optimization framework, which allows a quick estimate for building costs, hence improving operational efficiency and competitiveness of construction company.

A dataset composed of 10,000 parametric building configurations, collected from end-to-end real-world activities in the company, was used to train and validate the ML models to perform multiple tasks. First, the costs and required resources were predicted for a given set of building features. Then, the possible building features were identified for a given budget. Finally, the optimization of the building costs with feature constraints was conducted. Numerical data was carefully processed, and the Principal Component Analysis method was applied to remove the correlation of data features. Multiple algorithms were tested to explore the suitability of ML to estimate the building costs. Among the 13 ML regression algorithms used, the Artificial Neural Network, Gradient Boosting, and XGBoost models appear to be the most suitable to estimate the building costs and the required resources with an accuracy of 99% within less than a second of the training time. The number of floors, the floor depth, the floor height, and the kitchen's width are found to be the most influential in estimating the building costs. Artificial Neural Network models were also developed to identify options that a client can afford under a given budget. The optimization problem under constraints was successfully solved, helping clients determine the optimal building costs according to their preferences. The optimized building costs obtained by the optimization framework using ML-models are 7% smaller than those of the actual data, hence improving the company's competitiveness.

This study showcases that ML models can be efficiently used in the construction sector to optimize the workflow for cost savings and provide some practical implications for data-driven construction management.

For future study, the optimization with the constrained set of the required resources will be performed in order to exploit in-depth the datasets. Also, further investigation on the sensitivity of ML models will be performed.

Acknowledgement

The support of Thu Dau Mot University for this project is greatly appreciated.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Afonso B, Melo L, Oliveira W, Sousa S, Berton L. 2019. Housing prices prediction with a deep learning and random forest ensemble. In Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional. SBC; p. 389–400.

Aldous D. 1993. The continuum random tree 2I. Ann Probab. 167:248–289.

Bhagat N, Mohokar A, Mane S. 2016. House price forecasting using data mining. IJCA. 152(2):23–26.

Čeh M, Kilibarda M, Lisec A, Bajat B. 2018. Estimating the performance of random forest versus multiple regression for predicting prices of the apartments. IJGI. 7(5):168.

Chen JH, Ong CF, Zheng L, Hsu SC. 2017. Forecasting spatial dynamics of the housing market using support vector machine. Int J Strategic Prop Manag. 21(3):273–283.

Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H. 2015. Xgboost: extreme gradient boosting. R package version 0.4-2, 1(4).

Friedman JH. 2002. Stochastic gradient boosting. Comput Stat Data Anal. 38(4):367–378.

Gao G, Bao Z, Cao J, Qin AK, Sellis T, Wu Z. 2019. Location-centered house price prediction: a multi-task learning approach. arXiv preprint arXiv: 1901.01774.

García de Soto B, Agustí-Juan I, Joss S, Hunhevicz J. 2019. Implications of construction 4.0 to the workforce and organizational structures. Int J Constr Manag. 1–13. DOI: 10.1080/15623599.2019.1616414

Ghosalkar NN, Dhage SN. 2018. Real estate value prediction using linear regression. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE; p. 1–5.

Goldberg DE. 2006. Genetic algorithms. USA: Springer.

Gulli A, Pal S. 2017. Deep learning with Keras. Birmingham, England: Packt Publishing Ltd.

Hoerl AE, Kennard RW. 1970. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics. 12(1):55–67.

Hong J, Choi H, Kim WS. 2020. A house price valuation based on the random forest approach: the mass appraisal of residential property in South Korea. Int J Strategic Prop Manag. 24(3):140–152.

Huang CH, Hsieh SH. 2020. Predicting BIM labor cost with random forest and simple linear regression. Autom Constr. 118:103280.

Hunter JD. 2007. Matplotlib: A 2D graphics environment. IEEE Ann Hist Comput. 9(03):90–95.

Jain AK, Mao J, Mohiuddin KM. 1996. Artificial neural networks: A tutorial. Computer. 29(3):31–44.

Jones E, Oliphant T, Peterson P. 2001. SciPy: Open source scientific tools for Python.

Jui JJ, Molla MI, Bari BS, Rashid M, Hasan MJ. 2020. Flat price prediction using linear and random forest regression based on machine learning techniques. In Embracing Industry 4.0. Singapore: Springer; p. 205–217.

Keller JM, Gray MR, Givens JA. 1985. A fuzzy k-nearest neighbor algorithm. IEEE Trans Syst, Man Cybern. SMC-15(4):580–585.

Kim GH, An SH, Kang KI. 2004. Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning. Build Environ. 39(10):1235–1242.

Kim GH, Shin JM, Kim S, Shin Y. 2013. Comparison of school building construction costs estimation methods using regression analysis, neural network, and support vector machine. JBCPR. 01(01):1–7.

Kim GH, Yoon JE, An SH, Cho HH, Kang KI. 2004. Neural network model incorporating a genetic algorithm in estimating construction costs. Build Environ. 39(11):1333–1340.

Kingma DP, Ba J. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kravanja S, Žula T. 2010. Cost optimization of industrial steel building structures. Adv Eng Softw. 41(3):442–450.

Le T, Hassan F, Le C, Jeong HD. 2019. Understanding dynamic data interaction between civil integrated management technologies: a review of use cases and enabling techniques. Int J Constr Manag. 1–22. DOI: 10.1080/15623599.2019.1678863.

Lešić V, Martinčević A, Vašak M. 2017. Modular energy cost optimization for buildings with integrated microgrid. Appl Energy. 197:14–28.

Li Y, Yuan Y. 2017. Convergence analysis of two-layer neural networks with relu activation. In Advances in neural information processing systems; p. 597–607.

Limsombunchai V. 2004. House price prediction: hedonic price model vs. artificial neural network. In New Zealand agricultural and resource economics society conference; p. 25–26.

Lowe DJ, Emsley MW, Harding A. 2006. Predicting construction cost using multiple regression techniques. J Constr Eng Manage. 132(7):750–758.

Madhuri CR, Anuradha G, Pujitha MV. 2019, March. House price prediction using regression techniques: A comparative study. In 2019 IEEE International Conference on Smart Structures and Systems (ICSSS); p. 1–5.

Maier O, Wilms M, von der Gablentz J, Krämer UM, Münte TF, Handels H. 2015. Extra tree forests for sub-acute ischemic stroke lesion segmentation in MR sequences. J Neurosci Methods. 240:89–100.

Matel E, Vahdatikhaki F, Hosseinyalamdary S, Evers T, Voordijk H. 2019. An artificial neural network approach for cost estimation of engineering services. Int J Constr Manag. 1–14. DOI: 10.1080/15623599.2019.1692400

McKinney W. 2011. pandas: a foundational Python library for data analysis and statistics. Python High Performance Scientific Comput. 14(9):1–9.

Park B, Bae JK. 2015. Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert Syst Appl. 42(6):2928–2934.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, … Duchesnay E. 2011. Scikit-learn: Machine learning in Python. J Machine Learn Res. 12:2825–2830.

Peng Z, Huang Q, Han Y. 2019. Model research on forecast of second-hand house price in chengdu based on xgboost algorithm. In 2019 IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT). IEEE; p. 168–172.

Phan TD. 2018. Housing price prediction using machine learning algorithms: The case of Melbourne city, Australia. In 2018 International Conference on Machine Learning and Data Engineering (iCMLDE). IEEE; p. 35–42.

Rardin RL, Rardin RL. 1998. Optimization in operations research (Vol. 166). Upper Saddle River, NJ: Prentice Hall.

Risbeck MJ, Maravelias CT, Rawlings JB, Turney RD. 2015. Cost optimization of combined building heating/cooling equipment via mixed-integer linear programming. In 2015 American Control Conference (ACC). IEEE; p. 1689–1694.

Roe BP, Yang HJ, Zhu J, Liu Y, Stancu I, McGregor G. 2005. Boosted decision trees as an alternative to artificial neural networks for particle identification. Nucl Instrum Methods Phys Res, Sect A. 543(2-3):577–584.

Safavian SR, Landgrebe D. 1991. A survey of decision tree classifier methodology. IEEE Trans Syst, Man, Cybern. 21(3):660–674.

Sanni-Anibire MO, Zin RM, Olatunji SO. 2020. Machine learning model for delay risk assessment in tall building projects. Int J Constr Manag. 1–10. DOI: 10.1080/15623599.2020.1768326.

Seber GA., Lee AJ. 2012. Linear regression analysis. Vol. 329. New Jersey, USA: John Wiley & Sons.

Sha'ar KZ, Assaf SA, Bambang T, Babsail M, Fattah AAE. 2017. Design–construction interface problems in large building construction projects. Int J Constr Manag. 17(3):238–250.

Suykens JA, Vandewalle J. 1999. Least squares support vector machine classifiers. Neural Process Lett. 9(3):293–300.

Tepeli E, Taillandier F, Breysse D. 2019. Multidimensional modelling of complex and strategic construction projects for a more effective risk management. Int J Constr Manag. 1–22. DOI: 10.1080/15623599.2019.1606493.

Tibshirani R. 1996. Regression shrinkage and selection via the lasso. J Stat Soc Ser B Methodol. 58(1):267–288.

Truong Q, Nguyen M, Dang H, Mei B. 2020. Housing Price Prediction via Improved Machine Learning Techniques. Procedia Comput Sci. 174: 433–442.

Van Der Walt S, Colbert SC, Varoquaux G. 2011. The NumPy array: a structure for efficient numerical computation. Comput Sci Eng. 13(2):22–30.

Wold S, Esbensen K, Geladi P. 1987. Principal component analysis. Chemometrics Intell Lab Syst. 2(1-3):37–52.

Zou H, Hastie T. 2005. Regression shrinkage and selection via the elastic net, with applications to microarrays. J Royal Statistical Soc B. 67(2):301–320.

Appendix

Due to competitive advantage constraints and confidentiality concerns, the attached dataset was standardized, and no explicit labels of the variables are listed. The interested readers are invited to contact the authors for access to the actual dataset.