IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 10, No. 2, April, 2023, pp. 405-411
Journal homepage: http://iieta.org/journals/mmep

# Estimation and Analysis of Building Costs Using Artificial Intelligence Support Vector Machine

Zahra Salahaldain[1], Sepanta Naimi[1], Riyadh Alsultani[2*]

[1] Department of Civil Engineering, Altinbas University, Istanbul 212, Turkey
[2] Department of Building and Construction Techniques Engineering, Al-Mustaqbal University College, Hilla 51001, Iraq

Corresponding Author Email: dr.riyadh.abdulabbas@uomus.edu.iq

https://doi.org/10.18280/mmep.100203

**Received:** 20 October 2022
**Accepted:** 20 December 2022

***Keywords:***
*conceptual cost estimation, building cost, cross-validation, support vector machines, sustainable economic model*

## ABSTRACT

An essential component of the project feasibility assessment is the conceptual cost estimate. In actuality, it is carried out based on the estimator's prior expertise. However, budgeting and cost control are planned and carried out ineffectively as a result of inaccurate cost estimates. The purpose of this article is to introduce an intelligent model to improve modeling approaches accuracy throughout early phases of a project's development in the construction sector. A support vector machine model, which is computationally effective, is created to calculate the conceptual costs of building projects. To get accurate estimates, the suggested neural network model is trained using a cross-validation method. Through the research of the literature and interviews with experts, the cost estimate's influencing elements are determined. As training instances, the cost information from 40 structures is used. Two potent intelligence methods-Nonlinear Regression (NR) and Evolutionary Fuzzy Neural Interface Model (EFNIM)-are offered to illustrate how well the suggested model performs. Based on the readily accessible dataset from the relevant literature in the construction business, their results are contrasted. The computational findings show that the intelligent model that is being provided outperforms the other two potent methods. During the planning and conceptual design phase, the inaccuracy is satisfied for a project's conceptual cost estimate. Case studies demonstrate how SVMs may help planners anticipate the cost of construction in an effective and precise manner.

## 1. INTRODUCTION

Building cost conceptual estimates give planners a foundation for assessing the project's viability at the conceptual planning stage. The feasibility and profitability of a project are both significantly impacted by erroneous cost estimation. Low feasibility prevents clients from selling their existing projects because of overestimated expenses. On the other side, an underestimated cost can deceive planners into believing it is very feasible, which would result in extra expenditures for the customer at the building stage. Overestimated or underestimated expenses, therefore, have an impact on clients' earnings, necessitating the use of an accurate measurement technique [1].

Experience is the main focus of the conceptual cost estimate. Cost estimators are limited to basing their construction cost estimates during the conceptual planning phase on the initial design and project concepts. Cost estimators consult past situations when there is little information, and they then assess the conceptual cost in light of their prior experiences [2]. Nevertheless, a wide range of factors affects building costs. Some of these characteristics, including geological property and decorative class, are rife with ambiguity [3]. Estimators cannot correctly estimate building costs using a simple linear method since the evaluation process is so complicated and imprecise. Because of this, current building cost estimates are imprecise.

The related literature contains a variety of research methods to calculate the conceptual costs of building projects. Neural networks (NNs) have received a lot of attention in this sector, especially in the previous 10 years. An evolutionary conceptual cost estimation model for building was created using the "Evolutionary Fuzzy Neural Inference Model (EFNIM)". In the model, fuzzy input-output mapping is handled by neural networks, fuzzy logic is utilized to represent uncertainty and approximate reasoning, and genetic algorithms are predominantly employed for optimization. However, it takes a very considerable amount of time to conduct the computation to find the best answer [4]. Support vector machines (SVMs), a kind of artificial intelligence, were used to forecast the conceptual cost of building [5]. The SVM model was introduced for assessing the accuracy of conceptual cost estimates, and its use in the construction industry was looked at the research [6]. It was suggested to anticipate the cost of the construction project using an intelligent strategy based on the SVM. The outcomes demonstrated that the least square SVM's prediction accuracy was superior to the NN [7].

The appropriateness of any given estimating method depends typically on the purpose for which it is employed, the quantity of information available at the time of the estimation, and the person utilizing it. Although customers and contractors rely on existing cost prediction and forecasting techniques, real final building project prices still differ significantly from initial projections [8]. To suggest a general copula-based Monte Carlo simulation approach for forecasting the overall costs of building projects with dependent cost components. It was discovered that various dependency structures might produce various total cost probability distributions. Moreover, it is discovered that the current goodness of fit tests may be used to choose the copula that performs the best. The article concluded that the copula-based Monte Carlo simulation approach may reasonably anticipate the overall cost of construction projects [9, 10]. Firouzi et al. [11] contrast the precision of several estimation methodologies. By completing construction cost estimations using linear regression analysis (LRA) and support vector machine techniques (SVMs), the results highlight the urgent need for a solution to the problem of cost overruns.

To enhance the decision-making process in feasibility studies, a novel intelligent model for building projects based on an SVM model is implemented in this study. The suggested approach may be successfully used to anticipate conceptual costs over the long run in the building sector. In the SVM model, a cross-validation procedure is also utilized on the preparation dataset to minimize overfitting and to generate accurate results. The available information from the relevant literature in the construction sector is utilized to evaluate the suggested intelligent model's precision in estimating the expenses of these projects. The applicability and usefulness of the suggested approach are further demonstrated by comparisons with other well-known methodologies. The suggested model exhibited improved generalization performance and produced lower estimation errors. Given that these results pertain to actual situations from practice. An expansion of this research suggests improving the recommended model to boost yields and address this issue.

## 2. PROPOSED MODEL

It is usually hard to accurately calculate a model using cost time series data produced by linear methods. In the construction industry, the use of linear estimation techniques is not feasible [12-16]. Cost information is used in building projects in a complex and nonlinear way. Therefore, as conventional linear estimate approaches are inappropriate for these projects, it is crucial to develop cutting-edge methodologies like artificial intelligence for estimating time series data. In order to address the weaknesses of the widely-used strategies, this research reports on the cost estimate of these projects by suggesting a new, intelligent model built on two effective methodologies.

### 2.1 Support vector machine

Vapnik built the first SVMs on the back of SLT in the late 1960s. However, from the middle of the 1990s, as more computer power became available, the algorithms used for SVMs began to emerge, opening the door for several real-world applications with significant outcomes [16]. As shown in Figure 1 [17], the basic SVM works with two-class issues where the data are divided by a hyperplane that is specified by a set of support vectors. For the sake of thoroughness, a basic introduction to SVM is provided below. Readers can refer to the SVM tutorials [18] for detailed descriptions. A novel training method based on the SVM called least squares support vector machines (LS-SVM) only needs the answers to a few linear equations as opposed to the regular SVM's lengthy and computationally challenging quadratic programming issue. The least squares cost function is used by the LS-SVM [19].

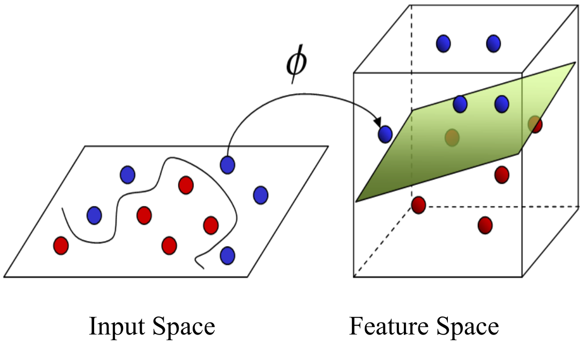

**Figure 1.** Nonlinear SVM [11]

**Classification:** The SVM is designed to learn a unique function that categorises training instances into different groups according to the class labels that have been provided to them. This viewpoint asserts that SVMs are a category of supervised learning model that are mostly employed to address classification and regression problems [20]. By converting the input vector x into a high-dimensional feature space, SVM models developed in the new space can represent a linear or nonlinear decision boundary in the old space. A linear hyperplane can be used to partition the instances if they are linearly separable; if not, the case is nonlinearly separable [20].

Consider a given set S with n labeled training instances: $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ for the linearly separable case. Each training instance $x_i \in R^k$, for i=1, …, n, belongs to one of the two classes in comparison to its label $y_i \in \{-1, +1\}$, where k is the input dimension. The following equation may be used to explain the greatest margin hyperplane:

$$y = b + \sum w_i y_i x(i) x \tag{1}$$

where denotes a dot product, a test example is represented by the vector x, and support vectors are represented by the vectors x(i)s. The parameters, *b* and $w_i$ in this equation, which determines the hyperplane, must be learned by the SVM. To create a perfect hyperplane, the following sequential quadratic programming (QP) problem must be solved:

$$\begin{gathered} \text{Minimize } \tfrac{1}{2}\|w\|^2 \\ \text{Subject to } y_i(wx_i + b) \geq 1 \qquad i = 1, \ldots, n \end{gathered} \tag{2}$$

where, the kernel function is defined as $K(x(i), x)$. There are several kernels available for creating the inner products needed to construct SVMs with various nonlinear decision surfaces. The most often used kernel functions are the Gaussian radial basis function $K(x, y) = \exp(-1/\delta^2(x - y)^2)$ and the polynomial kernel $K(x, y) = (xy + 1)^d$, where, d is the degree of the polynomial kernel and $\delta^2$ is the bandwidth of the Gaussian radial basis function [20].

**Regression:** A version of the SVM for regression has been included in the SVMs, which have been created for general estimation and prediction applications (LS-SVM). By reducing the prediction error, LS-SVM seeks to identify a function that accurately approximates the training examples. By simultaneously attempting to maximize the flatness of the function, the danger of over-fitting is reduced when the error is minimized. One must once again resolve the following quadratic programming issue to identify an ideal hyperplane:

$$\begin{gathered}\text{Minimize } \frac{1}{2}\|\mathrm{w}\|^{2} \\ \text{Subject to } \|\mathrm{y_i}\ (\mathrm{w.x_i}+\mathrm{b})\| \ \leq \varepsilon,\end{gathered} \tag{3}$$

where, the value ε ≥ 0 is the prediction error bound. If $\mathrm{f} = \langle \mathrm{w,x} \rangle + \mathrm{b}$ genuinely exists and approximates all pairs $(\mathrm{x_i}, \mathrm{y_i})$ with ε precision, then the aforementioned convex optimization problem will be achievable. The following optimization problem has restrictions that are ordinarily impossible to satisfy, thus one inserts slack variables $\varphi_i, \varphi_i^*$ to deal with them:

$$\begin{gathered}\text{Minimize } \frac{1}{2}\|\mathrm{w}\|^{2} + \mathrm{C}\sum_{i=1}^{l}(\varphi_i, \varphi_i^*) \\ \text{Subject to } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \varphi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \varphi_i^* \\ \varphi_i, \varphi_i^* \geq 0 \end{cases}\end{gathered} \tag{4}$$

The trade-off between f flatness and the maximum allowed deviation, *ε*, is calculated using the constant *C*. This optimization problem may be built as a dual problem by building the Lagrangian function:

$$\begin{gathered}\mathrm{L} = \frac{1}{2}\|\mathrm{w}\|^{2} + \mathrm{C}\sum_{i=1}^{l}(\varphi_i, \varphi_i^*) - \sum_{i=1}^{l}\lambda_i(\varepsilon + \varphi_i - \\ y_i + \langle w, x_i \rangle + b) - \sum_{i=1}^{l}\lambda_i^*(\varepsilon + \varphi_i^* - y_i + \langle w, x_i \rangle - \\ b) \text{and } \lambda_i, \lambda_i^* \geq 0\end{gathered} \tag{5}$$

Solving the Lagrangian, one provides the optimal solutions $\mathrm{w}^*$and $\mathrm{b}^*$:

$$\begin{gathered}b^* = y_i - \langle w, x_i \rangle - \varepsilon, \quad 0 \leq \lambda_i \leq C,\ i = 1, \dots, l, \\ b^* = y_i - \langle w, x_i \rangle + \varepsilon, \quad 0 \leq \lambda_i^* \leq C,\ i = 1, \dots, l\end{gathered} \tag{6}$$

For nonlinear issues, the inner products can be changed by appropriate kernels, much like in classification. Enforcing the upper limit *C* on the absolute value of the coefficients $\mathrm{w_i}$s allows us to keep an eye on the trade-off between reducing prediction error and maximizing the flatness of the regression function. The function may match the data more closely the greater *C*. The approach essentially does the least absolute-error regression with the coefficient size restriction in the degenerate situation where *ε*=0, regardless of the amount of *C*. In contrast, if is big enough, the error decreases to zero, and the algorithm returns the flattest curve that contains the data, regardless of the amount of *C* [20-22].

### 2.2 Cross-validation technique

Cross-validation [23] is a well-known technique for calculating generalisation mistakes. Among various cross-validation techniques, the k-Fold cross-validation approach is taken into consideration in this study. K-Fold cross-validation is one way to enhance the holdout strategy. Each of the k subgroups of the dataset uses the holdout method. Every time, one of the k subsets is used as the test set, and the remaining k-1 subsets are used to create the training set. Next, the average error over all k trials is calculated. The main benefit of this method is that it doesn't matter how the data is split up.

Each data set appears k times in the training set and exactly once in the test set. The variability of the resulting estimate diminishes as k is increased. By randomly dividing the data into test and training sets k times, this strategy may be changed. An example of how neural network architectures may be utilised to get the best generalisation is the k-fold cross-validation approach. In the pertinent literature, 3 and ten-fold cross-validation approaches were advised to estimate practical applications [24].

### 2.3 Intelligent proposed model

The proposed model is based on the LS-SVM and cross-validation, two effective techniques. The input-output mapping in this model is managed by the LS-SVM, which focuses on the features of cost data in building projects. The LS-SVM is trained using k-fold cross-validation to provide accurate findings and to enable a more realistic evaluation of the accuracy by splitting the whole dataset into numerous training and test sets. The suggested intelligent model may be a mix that is both suitable and efficient computationally for cost estimation of construction projects. The recommended model's steps are as follows: dividing the data into training and test sets is **Stage 1:** The test set is used to evaluate the model's performance after the training set has been used to create the LS-SVM model. **Stage 2:** Instructional data the model uses sequential data as training data. To help prevent numerical difficulties, the training data are now normalised into the same domain (0, 1) and the sequential data represents the desired attributes. The function used it to normalise data is demonstrated in the following illustration:

$$\mathrm{x_{sca}} = \frac{\mathrm{x_i} - \mathrm{x_{min}}}{\mathrm{x_{max}} - \mathrm{x_{min}}} \tag{7}$$

All the variables have no dimensions as a result of this change. **Stage 3:** Instruction in LS-SVM This step manages input-output mapping using the LS-SVM. The Gaussian radial basis function kernel is employed as a logical alternative [12, 20]. Using the k-fold cross-validation approach on the training data set, the LS-SVM training is done to build the prediction model.

Fitness definition: Chen and Wang [21] assert that although the training dataset's fitness may be easily calculated, it is prone to over-fitting.

This problem is solved using the k-fold cross-validation technique. This technique divides the training dataset into k subsets at random, then uses the k-1 subset as the training set to construct the regression function with the specified set of parameters $(\mathrm{C_i}, \delta_i)$. The final set is taken into account for validation. The preceding procedure is repeated k times. As a result, the fitness function is defined by the MAPECV of the k-fold trans procedure on the training dataset:

$$\text{Fitness} = \min \mathrm{f} = \mathrm{MAPE_{cv}}, \tag{8}$$

$$\mathrm{MAPE_{cv}} = \frac{1}{l}\sum_{j=1}^{l}\left|\frac{y_j - \hat{y}_j}{y_j}\right| \times 100\% \tag{9}$$

where, $y_j$ is the actual value; $\hat{y}_j$ are the validation value and l is the number of subsets. The solution with a smaller $\mathrm{MAPE_{cv}}$ of the training, the dataset has a smaller fitness value. **Stage 4:** The LS-SVM parameters are supplied in this stage so that the estimations may be made. Choose the best parameter settings to construct the LS-SVM model in **Stage 5:** LS-SVM model with the test dataset substituted to achieve the estimation values. Through testing performance, it is possible to validate the LS-SVM model's estimated capacity by taking into account performance criteria to compute the error between real and estimates values. The structure of the suggested intelligent model is shown in Figure 2. To reduce the MAPECV during prediction iterations, this model is used to try to optimize the LS-parameter SVM's combinations.

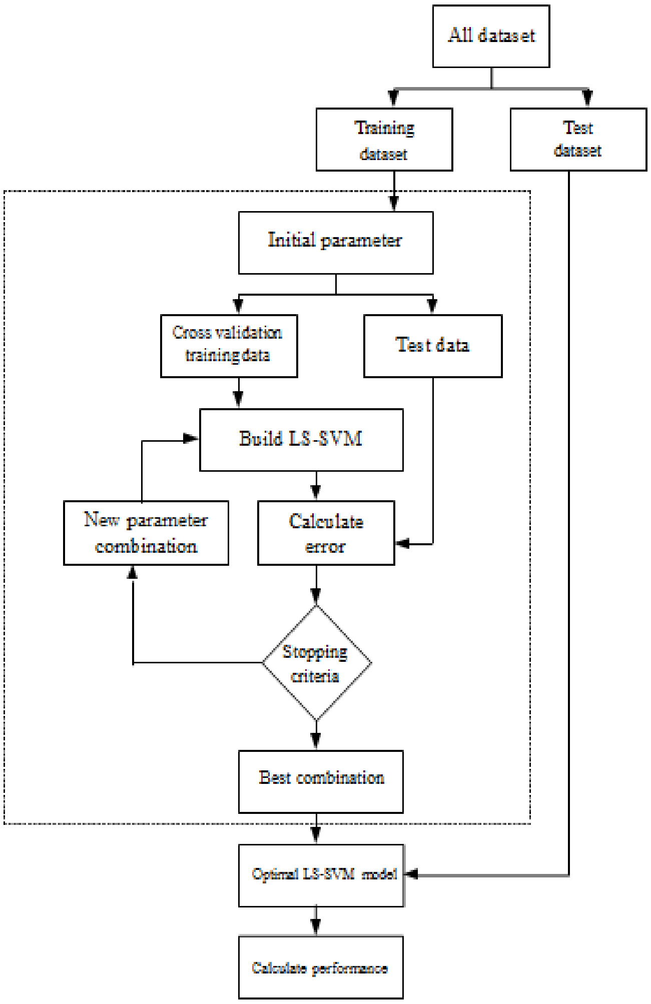

**Figure 2.** The proposed LS-SVM model

## 3. EVOLUTIONARY FUZZY NEURAL INTERFACE MODEL

The majority of issues in construction management are complicated, unclear, and environment-dependent. Neural networks, fuzzy logic, and genetic algorithms have all been effectively used in construction management to address a variety of issues. These three computing techniques make up for one paradigm's shortcomings with those of the other two. By combining the aforementioned three techniques and taking into consideration each method's advantages, Cheng [1] develops an "Evolutionary Fuzzy Neural Inference Model (EFNIM"). Since GAs is primarily concerned with optimization, FL with imprecision and approximation reasoning, and NNs with learning and curve fitting, the best adaptation mode is automatically found in the model. Figure 3 depicts the EFNIM's architecture.

The FL, NN, and GA paradigms are combined to form the proposed EFNIM. The benefits of one paradigm were balanced out by the others when FL, NNs, and GAs were used together. FL, NN, and GAs are largely focused on learning and curve fitting in the formulated model, whereas FL, NN, and GAs are mostly focused on imprecision and approximation reasoning. Technologies FL and NNs complement one another.

A possible first step toward the development of intelligent machines that can replicate the operations of the human brain appears to be the combination of two techniques into a single system. The NN in Figure 3 takes the role of the fuzzy rule base and fuzzy inference engine of the conventional fuzzy logic system. The NN is used to overcome difficulties in collecting fuzzy rules and figuring out composition operators as well as to provide the integrated system a learning capability. A neural network with both fuzzy inputs and fuzzy outputs is referred to as a "neuro with fuzzy input-output," and it combines the FL and NN. The FNN, a general word to denote the fusion or union of FL and NN, is used in this study to initialise the "neuro with fuzzy input-output" for convenience of usage.

Although the FNN is more reasonable than conventional FL to mimic the qualities and process of human reasoning, it has trouble choosing an acceptable topology and sufficient parameters for human inference. Furthermore, selecting a suitable distribution for the MFs to address different problems takes time, and the difficulty increases with problem complexity. GA is an effective tactic for addressing the drawbacks of FNN. The EFNIM makes advantage of GA to simultaneously determine the best FNN topology, best FNN variables, and fittest MF forms.

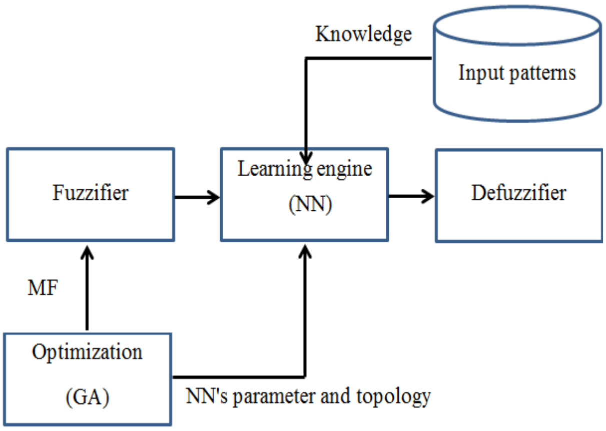

**Figure 3.** EFNIM Architecture [1]

## 4. MODEL VALIDATION

In order to evaluate the proposed model's effectiveness, the accessible dataset based on an actual data presented in the study [3] is applied to the LS-SVM model with a cross-validation strategy. The conceptual phase of a construction project's cost is affected by 10 input patterns in this dataset. Site studies and the owner's preliminary needs can be divided into two primary categories. Table 1 shows eight simulated validation of multi-story buildings and 32 multi-story buildings. These models are actual community housing complexes with reinforced concrete buildings that will be built in the Central region of Iraq between 2015 and 2022. These input patterns are from actual group initiatives in Taiwan between 1997 and 2001 [3].

The descriptions of these patterns are as follows: (1) Site area (in square meters); (2) Geology property; (3) Influencing householder number; (4) Earthquake impact; (5) Planning householder number; (6) Total floor area (in square meters); (7) Floor over ground (in stories); (8) Floor underground (in stories); (9) Decoration class; (10) Facility class; and (11) Normalized cost of the building project.

The cost of building projects was in Iraqi Dinar per square meter. In Table 1, factors 1, 3, 5, 6, 7, and 8 are quantitative, whereas factors 2, 4, 9, and 10 are qualitative.

In Table 2, the qualitative characteristics are outlined. The dataset contains 80 rows of data, of which 20 rows are chosen to evaluate the suggested intelligent model as well as two well-known methods, namely Nonlinear Regression (NR) and Evolutionary Fuzzy Neural Interface Model (EFNIM). The sequential data is then used by the model as training data. In the training phase, a cross-validation approach is used to address the issues caused by the short training dataset.

To avoid numerical difficulties and attributes with greater numeric ranges predominating those with smaller numeric ranges, training data are normalised into a (0, 1) range and sequential data reflects the suggested qualities. Data normalisation using a function (7). Three metrics are used to evaluate the performance of the proposed model: mean absolute average error (MAPE), mean square error (MSE), and R-squared (R2), as indicated by (10) to (12):

$$\mathrm{MAPE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left|\frac{\mathrm{y_i} - \widehat{\mathrm{y_i}}}{\mathrm{y_i}}\right| \times 100\% \tag{10}$$

$$\mathrm{MSE} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \widehat{\mathrm{y_i}})^2}{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \bar{\mathrm{y}})^2} \tag{11}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \widehat{\mathrm{y_i}})^2}{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \bar{\mathrm{y}})^2} \tag{12}$$

where, $y_i$ and $\widehat{y_i}$ represent the actual and estimated values of the i-th data, respectively. *y* is the average of actual data and *N* is the number of data. The overall comparative results based on the MAPE, MSE and $R^2$ indices are illustrated for the proposed model and two famous techniques in Table 3.

**Table 1.** Patterns for conceptual estimation of building cost

| Input patterns | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **No.** | **Building cost** | **Input** | | | | | | | | | | **Output** |
| | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| 1 | 643,458,474 | 909.3438 | 1 | 38 | 1 | 60 | 7137.7 | 14 | 2 | 1 | 2 | 0.643458 |
| 2 | 704,478,078 | 1715.448 | 2 | 2 | 1 | 29 | 6848.01 | 15 | 1 | 3 | 1 | 0.704478 |
| 3 | 972,456,659 | 2259.313 | 1 | 20 | 1 | 16 | 6934.26 | 19 | 3 | 1 | 2 | 0.972457 |
| 4 | 614,184,888 | 1851.813 | 1 | 120 | 1 | 95 | 9238.73 | 10 | 2 | 2 | 2 | 0.614185 |
| 5 | 917,898,343 | 2989.521 | 1 | 44 | 1 | 175 | 19275.83 | 12 | 2 | 1 | 3 | 0.917898 |
| 6 | 980,104,713 | 3913.427 | 2 | 18 | 1 | 104 | 6629.14 | 14 | 2 | 3 | 3 | 0.980105 |
| 7 | 561,769,346 | 3144.719 | 1 | 23 | 1 | 241 | 24185.31 | 22 | 2 | 2 | 2 | 0.561769 |
| 8 | 465,773,082 | 927.7292 | 1 | 124 | 1 | 14 | 10606.56 | 16 | 3 | 1 | 1 | 0.465773 |
| 9 | 400,665,726 | 6019.646 | 1 | 147 | 1 | 272 | 43767.47 | 19 | 3 | 1 | 1 | 0.400666 |
| 10 | 727,554,103 | 2982.083 | 1 | 12 | 1 | 94 | 16934.4 | 16 | 2 | 2 | 3 | 0.727554 |
| 11 | 557,318,970 | 1928.323 | 2 | 0 | 1 | 218 | 13233.82 | 19 | 3 | 2 | 2 | 0.557319 |
| 12 | 490,002,908 | 2238.24 | 2 | 33 | 1 | 55 | 17399 | 19 | 2 | 2 | 1 | 0.490003 |
| 13 | 405,050,170 | 3360.313 | 2 | 78 | 1 | 154 | 17667.02 | 12 | 2 | 1 | 1 | 0.40505 |
| 14 | 622,854,881 | 2902.823 | 1 | 22 | 1 | 160 | 36542.68 | 33 | 4 | 2 | 2 | 0.622855 |
| 15 | 647,084,707 | 867.75 | 1 | 175 | 1 | 151 | 8638.9 | 19 | 3 | 2 | 2 | 0.647085 |
| 16 | 631,327,078 | 1370.49 | 3 | 224 | 1 | 12 | 7665.56 | 8 | 2 | 3 | 2 | 0.631327 |
| 17 | 843,956,166 | 2397.938 | 1 | 88 | 1 | 70 | 20517.38 | 19 | 3 | 3 | 3 | 0.843956 |
| 18 | 790,419,788 | 840.6146 | 1 | 45 | 1 | 86 | 8727.04 | 19 | 3 | 3 | 3 | 0.79042 |
| 19 | 892,811,407 | 4557.667 | 2 | 12 | 1 | 59 | 15588.21 | 15 | 2 | 3 | 3 | 0.892811 |
| 20 | 485,025,080 | 822.9792 | 1 | 0 | 1 | 38 | 6190.73 | 16 | 2 | 2 | 1 | 0.485025 |
| 21 | 810,924,484 | 1619.24 | 2 | 125 | 1 | 233 | 13191.88 | 16 | 2 | 3 | 3 | 0.810924 |
| 22 | 706,456,023 | 2896.927 | 1 | 29 | 1 | 68 | 6629.14 | 14 | 1 | 2 | 2 | 0.706456 |
| 23 | 682,456,957 | 7924.792 | 1 | 78 | 1 | 283 | 28735.09 | 23 | 3 | 1 | 2 | 0.682457 |
| 24 | 762,860,421 | 1968.719 | 1 | 74 | 1 | 235 | 12098.87 | 18 | 2 | 3 | 3 | 0.76286 |
| 25 | 405,280,931 | 3042.625 | 1 | 147 | 1 | 161 | 17222.73 | 11 | 2 | 1 | 1 | 0.405281 |
| 26 | 401,588,767 | 1415.594 | 1 | 23 | 1 | 96 | 9051.38 | 12 | 2 | 1 | 1 | 0.401589 |
| 27 | 833,538,989 | 1821.75 | 1 | 36 | 1 | 103 | 15263.04 | 16 | 2 | 3 | 3 | 0.833539 |
| 28 | 746,872,032 | 2412.438 | 2 | 124 | 1 | 100 | 13371.74 | 15 | 2 | 1 | 3 | 0.746872 |
| 29 | 527,979,452 | 3278.052 | 2 | 78 | 1 | 173 | 20768.76 | 12 | 2 | 1 | 2 | 0.527979 |
| 30 | 747,795,073 | 1828.01 | 1 | 21 | 1 | 106 | 23942.59 | 19 | 2 | 1 | 2 | 0.747795 |
| 31 | 614,712,340 | 2560.063 | 2 | 63 | 1 | 89 | 2393.2 | 23 | 2 | 1 | 3 | 0.614712 |
| 32 | 869,867,245 | 3487.479 | 1 | 77 | 1 | 154 | 28498.2 | 12 | 1 | 1 | 2 | 0.869867 |
| 33 | 655,161,316 | 3840.885 | 1 | 82 | 1 | 115 | 8773.19 | 10 | 1 | 3 | 1 | 0.655161 |
| 34 | 664,128,000 | 2962.875 | 1 | 64 | 1 | 40 | 13616.38 | 18 | 3 | 3 | 2 | 0.664128 |
| 35 | 447,608,954 | 1941.635 | 3 | 29 | 1 | 90 | 2832.2 | 23 | 2 | 2 | 1 | 0.447609 |
| 36 | 589,922,096 | 3278.052 | 2 | 33 | 1 | 152 | 23840.39 | 14 | 3 | 1 | 1 | 0.589922 |
| 37 | 647,315,467 | 2963.188 | 1 | 47 | 1 | 89 | 27368.39 | 10 | 1 | 2 | 2 | 0.647315 |
| 38 | 589,922,096 | 3557.167 | 1 | 86 | 1 | 118 | 25145.72 | 15 | 2 | 3 | 1 | 0.589922 |
| 39 | 809,737,717 | 2015.969 | 2 | 98 | 1 | 82 | 6087.76 | 23 | 1 | 2 | 3 | 0.809738 |
| 40 | 717,301,754 | 2788.802 | 1 | 122 | 1 | 151 | 13719.02 | 22 | 2 | 1 | 2 | 0.717302 |

**Table 2.** Description of qualitative factors for conceptual cost estimating

| Influencing factor | Qualitative option | Value |
|---|---|---|
| Geology property | Soft | 1 |
| | Medium | 2 |
| | Hard | 3 |
| Earthquake impact | Low | 1 |
| | High | 2 |
| Decoration class | Basic type | 1 |
| | Normal type | 2 |
| | Luxurious type | 3 |
| Facility class | Basic type | 1 |
| | Normal type | 2 |
| | Luxurious type | 3 |

**Table 3.** Overall comparative results

| Intelligent techniques | MAPE | MSE | $R^2$ |
|---|---|---|---|
| Proposed model | 0.625 | 0.1687 | 0.9147 |
| EFNIM | 2.214 | 0.3187 | 0.6687 |
| NR | 3.935 | 0.4987 | 0.5214 |

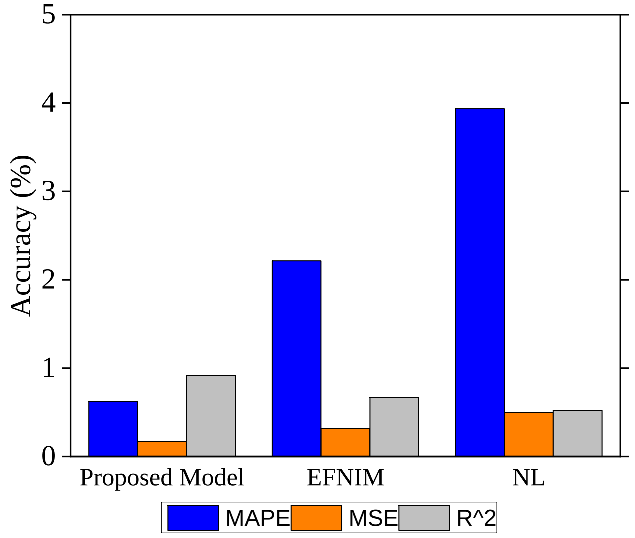

**Figure 4.** Comparison of accuracy for proposed model, EFNIM, and NL cost estimation models

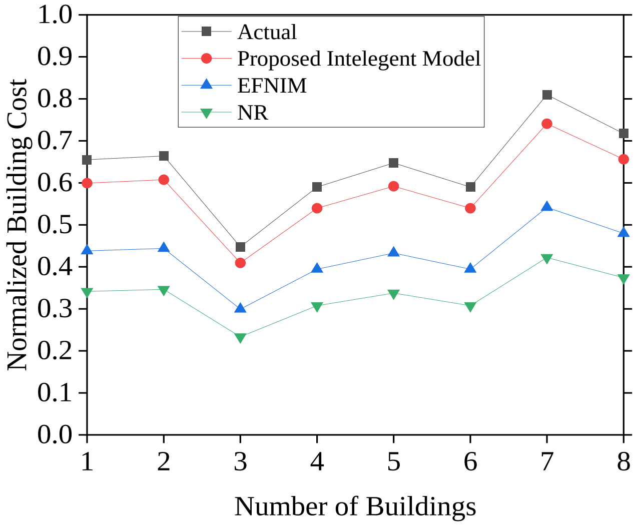

**Figure 5.** Examples illustrate the suggested model and two more well-known methods

MAPE and MSE often reflect accuracy as a percentage in statistics. According to this model, MAPE = 0.625 for LS-SVM, 2.214 for EFNIM, and 3.935 for NL indicating that the LS-SVM has a lower and greater percentage error than EFNIM and NR, respectively. The wide applicability of the prediction model is measured by the coefficient of determination R2, which quantifies how well data points fit a curve or a line. When represented as a percentage, R2 computes how much of the variance in the output response can be attributed to the model's predictors. R2 values fall within the range [0,1]. As shown in Figure 4, the value R2= 0.9147 for our cleverly developed model may be understood as follows: about 91% of the variation in the response is attributable to the selected predictors, and the remaining 9% is related to unknown variability.

Here, it is important to emphasize that since various types of predictive models may be appropriate depending on the type of relationship between the predictors and the dependent variable, it is best to test them all out to find the one that best fits our data and provides the most accurate predictions. The authors confirmed that the cross-validation approach was well suited for trustworthy estimation after obtaining pleasing results with LS-SVM with an error of less than 9%.

As shown in Figure 5, the prediction made using the suggested model yields results that are adequate. The minimum three indices in the final prediction results of the suggested model show strong performance and adaptability to the conceptual building cost in the construction sector. Project managers may assess construction projects appropriately, particularly in feasibility studies, by utilizing the suggested model. Additionally, it raises the likelihood that building projects will be successful.

## 5. CONCLUSION

In order to evaluate the conceptual cost of building projects, a new intelligent model created with cross-validation and support vector machine approaches was suggested in this study.

With the help of the suggested model, the LS-SVM was given a stronger ability to generalize the input-output connection it had learned throughout its training phase to make accurate predictions for brand-new input data. The Evolutionary Fuzzy Neural Interface Model and Nonlinear Regression were put up against the suggested model for a conceptual cost estimate of construction projects.

It was determined through comparison that the cross-validation approach was suitable for accurate estimation. Using the data from the building projects, the suggested model exhibited improved generalization performance and produced lower estimation errors. Given that these results pertain to actual situations from practice, the results MAPE=0.625, MSE=0.1687, and R2=0.9147 (91.5%) demonstrate the high predictive capabilities of the suggested LS-SVM.

The values of the input parameters, which were taken for Taiwan for the period from 1997-2001, have an effect on the performance of the proposed intelligent model for conceptual cost estimation in an overall way, although it matches the Iraqi conditions, but this matter must be reconsidered. As an extension of this research, optimizing the suggested model is suggested to increase yields and overcome some of the challenges.

## ACKNOWLEDGMENT

The corresponding author extends his thanks to Al-Mustaqbal University for the financial support for the research.

## REFERENCES

[1] Cheng, M.Y., Tsai, H.C., Sudjono, E. (2010). Conceptual cost estimates using evolutionary fuzzy hybrid neural network for projects in construction industry. Expert Systems with Applications, 37: 4224-4231. https://doi.org/10.1016/j.eswa.2009.11.080

[2] Cheng, M.Y., Wu, Y.W. (2005). Construction conceptual cost estimates using support vector machine. In 22nd International Symposium on Automation and Robotics in Construction, ISARC, pp. 1-5.

[3] Hsieh, W.S. (2002). Construction conceptual cost estimates using evolutionary fuzzy neural inference model. M. S. thesis, Dept. of Construction Engineering, National Taiwan University of Science and Technology, Taiwan.

[4] Cheng, M.Y., Wu, Y.W. (2005). Construction conceptual cost estimates using support vector machine. In 22nd International Symposium on Automation and Robotics in Construction ISARC, pp. 1-5. https://doi.org/10.22260/ISARC2005/0064

[5] Chang, C.C., Lin, C.J. (2001). Training nu-support vector classifiers: Theory and algorithms. Neural Computation, 13(9): 2119-47. https://doi.org/10.1162/089976601750399335

[6] An, S.H., Park, U.Y., Kang, K.I., Cho, M.Y., Cho, H.H. (2007). Application of support vector machines in assessing conceptual cost estimates. Journal of Computing in Civil Engineering, 21(4): 259-264. https://doi.org/10.1061/(ASCE)0887-3801(2007)21:4(259)

[7] Kong, F., Wu, X.J., Cai, L.Y. (2008). A novel approach based on support vector machine to forecasting the construction project cost. Proc. of the International Symp. on Computational Intelligence and Design (ISCID 2008), Wuhan, China. https://doi.org/10.1109/ISCID.2008.13

[8] Kim, G.H., Shin, J.M., Kim, S., Shin, Y. (2014). Comparison of school building construction costs estimation methods using regression analysis, neural network, and support vector machine. Journal of Building Construction and Planning Research, 1(1): 1-7. https://doi.org/10.4236/jbcpr.2013.11001

[9] Mačková, D., Bašková, R. (2014). Applicability of Bromilow´s time-cost model for residential projects in Slovakia. SSP - Journal of Civil Engineering, 9(2): 5-12. https://doi.org/10.2478/sspjce-2014-0011

[10] El-Sawalhi, I.N. (2015). Support vector machine cost estimation model for road projects. Journal of Civil Engineering and Architecture, 9: 1115-1125. https://doi.org/10.17265/1934-7359/2015.09.012

[11] Firouzi, A., Yang, W., Li, C.Q. (2016). Prediction of total cost of construction project with dependent cost items. Journal of Construction Engineering and Management, 142(12): 04016072. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001194

[12] Sapankevych, N.I., Sankar, R. (2009). Time series prediction using support vector machine: A survey. IEEE Computational Intelligence Magazine, 4(2): 24-38. https://doi.org/10.1109/MCI.2009.932254

[13] Mousavi, S.M., Iranmanesh, S.H. (2011). Least squares support vector machines with genetic algorithm for estimating costs in NPD projects. The IEEE International Conference on Industrial and Intelligent Information (ICIII 2011), Indonesia, Xi'an, China, pp. 127-131. https://doi.org/10.1109/ICCSN.2011.6014864

[14] Tavakkoli-Moghaddam, R., Mousavi, S.M., Hashemi, H., Ghodratnama, A. (2011). Predicting the conceptual cost of construction projects: A locally linear neuro-fuzzy model. The First International Conference on Data Engineering and Internet Technology (DEIT), pp. 398-401. https://doi.org/10.22094/JOIE.2016.231

[15] Vahdani, B., Iranmanesh, S.H., Mousavi, S.M., Abdollahzade, M. (2011). A locally linear neuro-fuzzy model for supplier selection in cosmetics industry. Applied Mathematical Modelling, 36(10): 4714-4727. https://doi.org/10.1016/j.apm.12.006

[16] Salunkhe, A.A., Patilin, P.A. (2020). Comparative analysis of construction cost estimation model using SVM and regression analysis. https://www.researchgate.net/publication/343065757_Comparative_Analysis_of_Construction_Cost_Estimation_Model_Using_SVM_and_Regression_Analysis.

[17] Pisner, D.A., Schnyer, D.M. (2020). Support vector machine. Machine Learning, Academic Press, 101-121.

[18] Wang, Y.R., Yu, C.Y., Chan, H.H. (2012). Predicting construction cost and schedule success using artificial neural networks ensemble and support vector machines classification models. International Journal of Project Management, 30(4): 470-478. https://doi.org/10.1016/j.ijproman.2011.09.002

[19] Khanday, A.M.U.D., Khan, Q.R., Rabani, S.T. (2021). SVMBPI: Support vector machine-based propaganda identification. In: Mallick, P.K., Bhoi, A.K., Marques, G., Hugo C. de Albuquerque, V. (eds) Cognitive Informatics and Soft Computing. Advances in Intelligent Systems and Computing, vol. 1317. Springer, Singapore. https://doi.org/10.1007/978-981-16-1056-1_35

[20] Huang, C.F. (2012). A hybrid stock selection model using genetic algorithms and support vector regression. Applied Soft Computing, 12(2): 807-818. https://doi.org/10.1016/j.asoc.2011.10.009

[21] Chen, K.Y., Wang, C.H. (2007). Support vector regression with genetic algorithms in forecasting tourism demand. Tourism Management, 28(1): 215-226. http://dx.doi.org/10.1016/j.tourman.2005.12.018

[22] Salgado, D.R., Alonso, F.J. (2007). An approach based on current and sound signals for in-process tool wear monitoring. International Journal of Machine Tools and Manufacture, 47(14): 2140-2152. https://doi.org/10.1016/j.ijmachtools.2007.04.013

[23] Tanveer, M., Rajani, T., Rastogi, R., Shao, Y.H., Ganaie, M.A. (2022). Comprehensive review on twin support vector machines. Annals of Operations Research, 1-46. https://doi.org/10.48550/arXiv.2105.00336

[24] Tanveer, M., Tiwari, A., Choudhary, R., Ganaie, M.A. (2022). Large-scale pinball twin support vector machines. Machine Learning, 111(10): 3525-3548. https://doi.org/10.1007/s10994-021-06061-z