

Original Article

# Comparative Study of Neural Networks and Support Vector Regression for Final Project Duration Prediction using Simulated Scheduling Scenarios

Hicham Hassi<sup>1</sup>, Mouna El Mkhale<sup>2</sup>, Nouzha Lamdouar<sup>3</sup>

<sup>1,2,3</sup>Civil Engineering and Construction structure GCC laboratory, Mohammadia School of Engineers,  
Mohammed V University, Rabat, Morocco.

<sup>1</sup>Corresponding Author : [hhasi2020@gmail.com](mailto:hhasi2020@gmail.com)

Received: 05 December 2025

Revised: 21 September 2026

Accepted: 04 October 2026

Published: 28 October 2026

**Abstract** - The prediction of the duration of a construction project is a highly complex task influenced by multiple factors. To tackle the heterogeneity and noise of historical datasets, this paper proposes the use of a highly controlled, simulation-generated dataset comprising 157 realistic schedules of school construction projects. The schedules were generated using six core planning variables. A distinctive contribution of this research is the systematic evaluation of the combinatorial performance between data normalization techniques (Min-Max and Z-score) and learning algorithms (Support Vector Regression and Artificial Neural Networks). The models were globally optimized using cross-validation to prevent data leakage. Comparison between the results of the SVR model with an RBF kernel under Min-Max normalization, and a compact ANN with one hidden layer using the Levenberg-Marquardt training function, demonstrated high theoretical accuracy within the simulated boundaries. The results indicate that SVR yields superior generalization accuracy across the analyzed scenarios. Furthermore, a conceptual reliability-aware framework, ScheduleRelia, is introduced. ScheduleRelia serves as a proof-of-concept application designed to estimate project duration and assess schedule health using critical path metrics. While broad generalization requires future external validation with real-world inputs, this study establishes a rigorous baseline for algorithm selection and data formatting in schedule forecasting.

**Keywords** - Construction management, Project duration prediction, Artificial Neural Networks (ANN), Support Vector Regression (SVR), ScheduleRelia, Project scheduling.

## 1. Introduction

Successful prediction of the end-of-project results is still a critical factor for the success of construction projects, controlling cost performance, resource efficiency, and stakeholder satisfaction. For many years, the industry has had to grapple with deterministic scheduling approaches such as Program Evaluation and Review Technique (PERT) or Critical Path Method (CPM). These static approaches, however, fail to address the stochastic nature of construction sites. Most of the time, they cannot accurately handle such complex and non-linear parameters like variability on execution, resource bottleneck and unforeseen site constraints, resulting in large discrepancies between planned and actual baselines [1, 2]. As a result, traditional monitoring systems often do not have the predictive capabilities to indicate risks in advance, therefore leading to significant monetary and time overruns.

To overcome this capability shortfall, the incorporation of Artificial Intelligence (AI) and Machine Learning (ML) has

increasingly become a game-changing breakthrough in transforming from reactive monitoring to proactive prediction. There has been significant progress in the management of cost; for example, Ding and Lu [3] have proved that the combination of Building Information Modeling (BIM) with a Neural Network can make up for the defects of traditional price management, so as to achieve better estimation accuracy. Concurrently, progress on the schedule automation has been made by Hong et al. [4], who introduced a Graph-based Schedule Mining (GSM) approach. They fill the “tacit knowledge” gap in leveraging history schedules to automate preliminary construction planning, not just by relying on manual heuristics. To complement these planning tools, Lagos et al. [5] demonstrate that when the Last Planner System is augmented with machine learning, it provides early warning indicators with much better performance than the widely used Schedule Performance Index (SPI) for the construction industry. The predictive models evolved in these studies have had the inevitable consequence that more sophisticated dynamic control models have come into focus of



the research work. The study of hybrid systems is a theme also evolving in actual research work, for instance the study by Yu and Zuo [6] proposed a fuzzy logic neural network for the optimal construction scheduling under uncertainty conditions, and the non-linear relationships that may occur in project logic need to be transformed into the network via the non-linear mapping.

The choice of the ‘optimal’ predictive model is still a topic of debate in the literature. A study by Chandanshive and Kambekar [7] utilised Support Vector Machines (SVM) and considered it to be a viable alternative for cost prediction for small-to-medium length of construction records. In a more empirical contribution to the literature, a study by Tamayo and Famadico [8] carried out the modelling of project success rating of infrastructure projects in Mindanao, using the Artificial Neural Networks (ANN) model and compared it with the traditional method of ordinary regression. They concluded that the rating of projects that comply with the standard parameters can be accurately identified using ANN, as it is capable of portraying the relationship among variables in a unique way.

However, the huge efforts so far [1-11] have left serious gaps. The vast majority of existing work uses heterogeneous historical sets, which, while useful for calibration, are contaminated by uncontrolled site-specific noise that masks the true properties of the algorithms. There are a few studies that use generated scheduling scenarios with the aim of benchmarking in a highly controlled environment. In addition, only a small number of comparisons have been made to explore working sensitivities in ANN and SVR by using different data normalization techniques (Min-Max vs. Z-Score) for the project scheduling problems [10, 11]. I. 4 E The “coupling effect” between the pre-processing procedure and model kernels is still far from being completely explored. Compared to current studies that tend to investigate cost or risk with static data [3, 9, 11], this study conducts a comparative analysis on 157 simulated schedules of a realistic school construction project. The aim is to provide an empirical guideline to determine the optimal prediction model by covering different types of normalization with various algorithm configurations.

### 1.1. Novelty and Objectives

The main contributions of this work are:

- **Controlled Simulation-Generated Dataset:** Creation of a unique, highly controlled dataset derived from 157 simulated, realistic scheduling scenarios (including compression, overlapping, and scope changes). This novel approach rigorously benchmarks predictive models while eliminating the uncontrolled, site-specific noise inherently present in standard historical field data.
- **Systematic Normalization-Kernel Coupling:** A thorough and systematic empirical investigation into how specific data scaling techniques (Min-Max vs. Z-Score)

intrinsically couple with ANN topologies and SVR kernels. This establishes a clear guideline for optimal data formatting in construction forecasting.

- **Reliability-Aware Framework (ScheduleRelia):** The development of an integrated decision-support application that translates theoretical high-accuracy ML performance into a pragmatic project governance tool, offering both predictive capabilities and dynamic schedule-health diagnostics.

### 1.2. Research Gaps

Although there are many attempts to introduce AI in construction forecasting, there are many critical issues that are not yet addressed in the body of knowledge of the subject. Most studies have relied on uncontrolled historical data for training and validating the developed models. Although these data sets can be used for broad calibration of a forecasting model, they contain site-specific noise and hidden errors, mostly resulting from undocumented subjective decisions made by human schedulers. Hence, there is a need to develop and validate forecasting models using a large set of highly controlled, virtually generated scheduling scenarios, which allow for evaluation of the effects of specific planning variables [5, 9].

**Neglected Pre-processing and Algorithmic Combinations:** Most research compares core algorithms independently and is highly sensitive to data pre-processing. To efficiently employ ML algorithms for scheduling, the deep sensitivities of various models to potential data scaling methods, and their specific performance when combined with distinct model kernels, require thorough investigation. [26, 28].

**Lack of Translation into Decision Making:** Most models currently are ‘black boxes’ that produce a number of predictions of duration that are relatively accurate but do not in any way provide a basis for a project manager to be able to make an active governance decision about a project [10, 16].

The literature lacks sufficient exploration of using models for the analysis of probabilistic and dynamic scheduling systems. The existing foundations focus on the analysis using static, deterministic models for point predictions of the expected duration of a scheduling problem, using traditional models such as ANN and SVR. There is a growing need to compare a set of advanced ensemble models, such as XGBoost, and also to investigate the use of models that allow uncertainty quantification, such as using Monte Carlo simulations.

The results from ML models need to be mapped into a dynamic, real-time scheduling system that can be updated based on a variety of data coming from various sources, such as IoT and from Digital Twins of the physical and process assets [11, 15, 40].

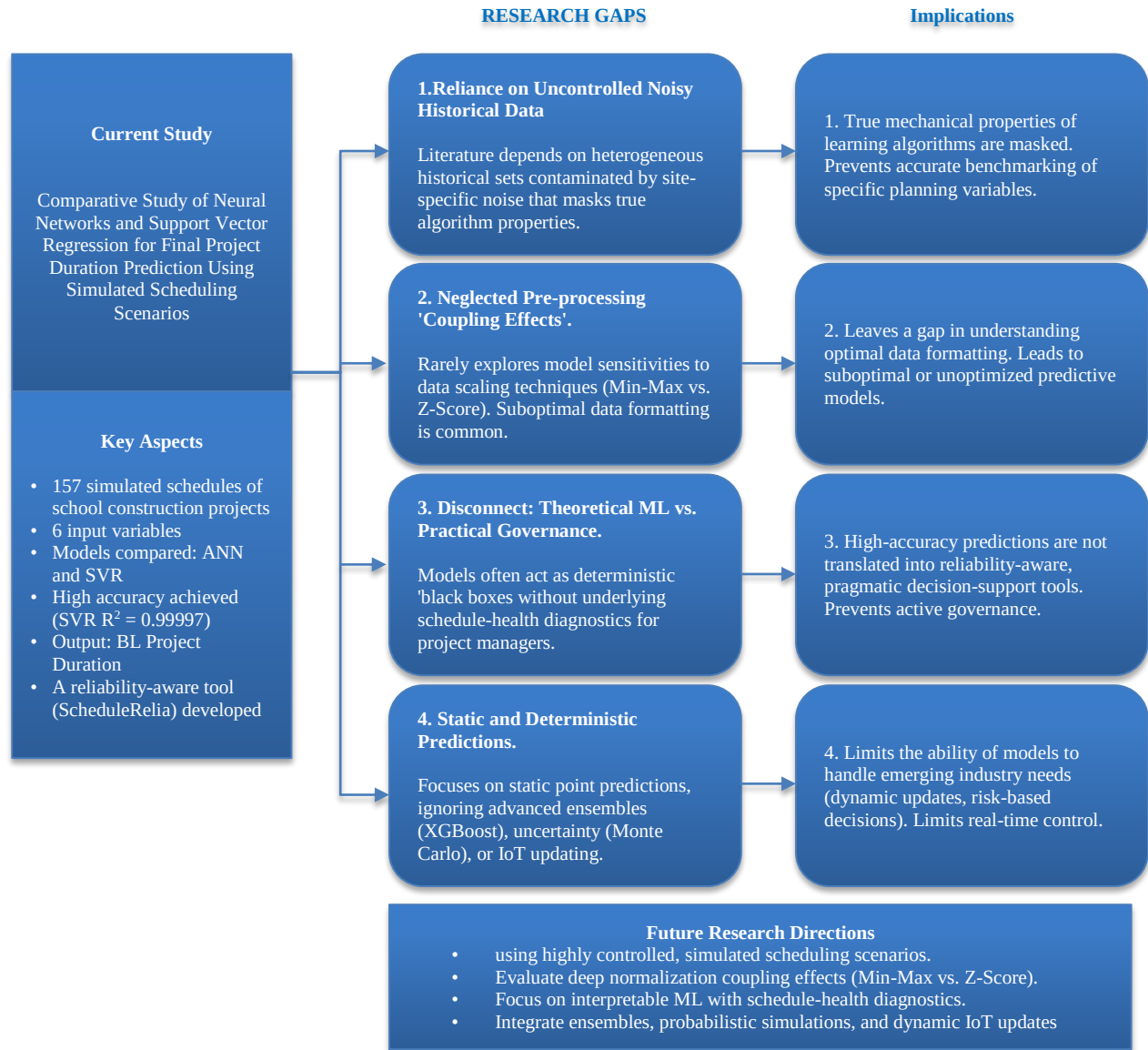


Fig. 1 Overview of current methodological gaps in machine learning for project duration forecasting

## 2. Overall Research Methodology

To address the issues formulated in this research and to compare the performance of ANN and SVR in terms of prediction, a novel framework for comparing the performance of two different methodologies has been designed. The methodology is divided into five sequential phases in order to transform raw data into meaningful information for project management. A flowchart that depicts the overall research framework has been graphically illustrated in Figure 2.

### 2.1. Phase 1: Data Generation and Acquisition

157 test cases covering a number of different realistic scenarios (e.g. data for 4 different school projects plus additional variation within each project to test for different scheduling conditions) generated 157 scheduling scenarios.

For each of the 157 test cases, 6 planning variables were recorded as input features. These 6 planning variables are: Total Activities, Sum of Total Float, Total Count of Relationships, Sum of Lag, Average Original Duration, and Maximum Original Duration.

### 2.2. Phase 2: Data Pre-Processing

All 157 development scheduling scenarios were 'unwound', and for each respective scenario, all 6 input planning variables were extracted and fed into the learning algorithm(s) after first having passed through appropriate scaling to eliminate scale bias. In this regard, two common scaling techniques, Min-Max Normalization and Z-Score Standardization, were both individually applied to the raw data and each isolated treatment caused to produce its own 'coupling' with the learning algorithms.

### 2.3. Phase 3: Model Development and Hyperparameter Tuning

The processed data was fed into two different approaches to implement a machine learning paradigm:

- The approach of the ANN design was to test different numbers of hidden layers and neurons within the hidden layers. An attempt was made to increase the capacity of the network while trying to avoid overfitting the data by using different optimization algorithms, such as Levenberg-Marquardt.
- SVR Development: In addition to different setups of the SVR for the prediction of the project completion time, various kernel functions (Linear, Quadratic, Cubic, as well as Radial Basis Function) were examined. As for the ANN, the grid-search for the optimal hyperparameters (C, epsilon, Kernel Scale) was implemented automatically, and evaluated alongside automated grid-search optimization for critical hyperparameters ( $C$ ,  $\epsilon$ , Kernel Scale).

### 2.4. Phase 4: Performance Evaluation and Validation

All models have been tested using k-fold cross-validation in order to ensure a high generalization capability of the models. The quality of the prediction of the models has been quantified by common statistical indicators (e.g. Mean Squared Error, Root Mean Squared Error, Coefficient of Determination) and by an inverse standardization in order to get back to the physical scale of days by the previously applied normalization.

### 2.5. Phase 5: Practical Implementation (ScheduleRelia)

The final phase translates the theoretical machine learning outcomes into a pragmatic decision-support tool. By integrating the top-performing algorithms, a reliability-aware framework, ScheduleRelia, was proposed to help project stakeholders dynamically assess schedule health and proactively predict final project durations.

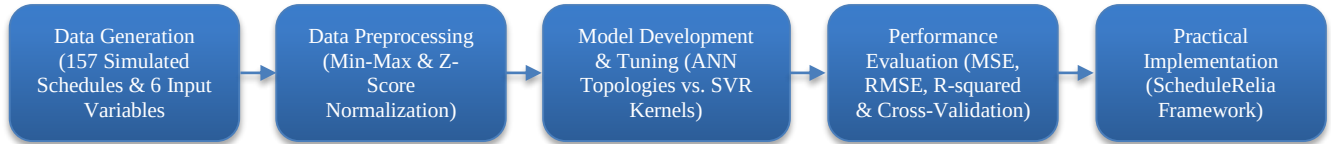


Fig. 2 Flowchart of the proposed research methodology for project duration forecasting

## 3. State of the Art

### 3.1. Artificial Neural Networks (NN/ANN)

#### 3.1.1. Principle

Artificial Neural Networks approximate complex non-linear mappings through layered compositions of linear transformations and non-linear activations trained by gradient-based optimization to minimize predictive error on project features such as scope, resources, and market indices

[12, 13]. In construction analytics, feed-forward backpropagation networks and hybrid deep models combining dense layers with temporal encoders have been used to forecast costs, schedules, risk indices, and prediction intervals for material price dynamics and project performance [14, 15]. Integration with BIM enables continuous data harvesting across the life cycle, supporting risk evaluation, cost control, and early warnings by coupling structured project information with ANN predictors [3, 13].

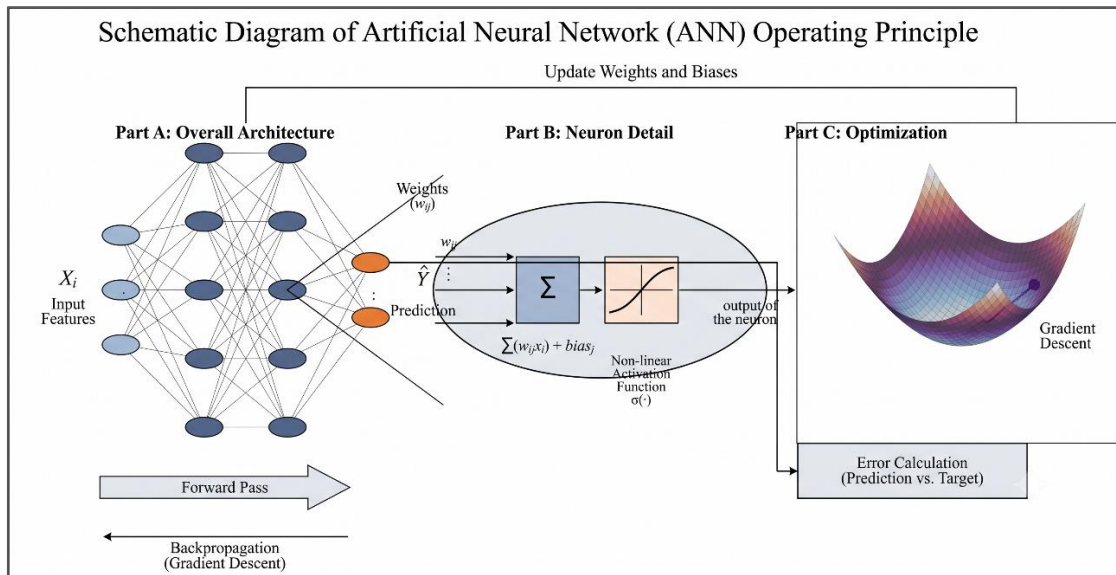


Fig. 3 Operational principle of an Artificial Neural Network (ANN)

By making use of the above-mentioned approach, a fundamental architecture and operational flow for the decision-support tool have been identified and are depicted in Figure 3. Figure 3 Operational Principle of an Artificial Neural Network (ANN). schematic diagram illustrating (A) the overall layered architecture and information flow, (B) the

detailed mathematical operations within a single hidden neuron demonstrating linear transformation ( $\sum wx + b$ ) and non-linear activation ( $\sigma$ ), and (C) the cyclic gradient-based optimization process via backpropagation to iteratively minimize predictive error.

### 3.1.2. Advantages and Disadvantages

**Table 1. Advantages and Disadvantages of Support Vector Regression (SVR)**

| Advantages                                                                                                                                                                                                                                                                      | Disadvantages                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MLQP has a high expressive power for non-linear, high-dimensional relationships. Thus, MLQP can make very accurate predictions of costs and schedules when enough training data is available [12, 16].                                                                          | These models are particularly sensitive to the quality of input data and tend to require a large amount of data. They can overfit even modestly-sized datasets, and may see a loss of generalizability when not properly regularized, tested, etc. [7, 12]. |
| Deterministic predictive models can be complemented by assessing the predictive uncertainty via techniques such as interval forecasting, named as LUBE (Lower and Upper Bound Estimation), which delivers much more useful decision support than mere point forecasts [15, 16]. | Black-box nature hinders interpretability and governance, complicating adoption in mission-critical control processes [10, 16].                                                                                                                             |
| BIM and Lifecycle Data Integration for Proactive Cost and Risk Management using Automated Inference Pipelines [3, 13].                                                                                                                                                          | Computationally intensive for architecture search and training, especially with deep or hybrid models and large BIM-integrated datasets (Cheng et al., 2025; Huang et al., 2024) [13, 14].                                                                  |
| Hybrid deep learning, which integrates static and sequential features, achieves superior accuracy in cost and schedule estimation for mixed data types. [14, 17].                                                                                                               | Case-specific modeling may limit external validity if not retrained or recalibrated when project contexts shift (Ding, 2020; El Abidi et al., 2025) [3, 18].                                                                                                |

### 3.1.3. Operating Mode of Artificial Neural Networks

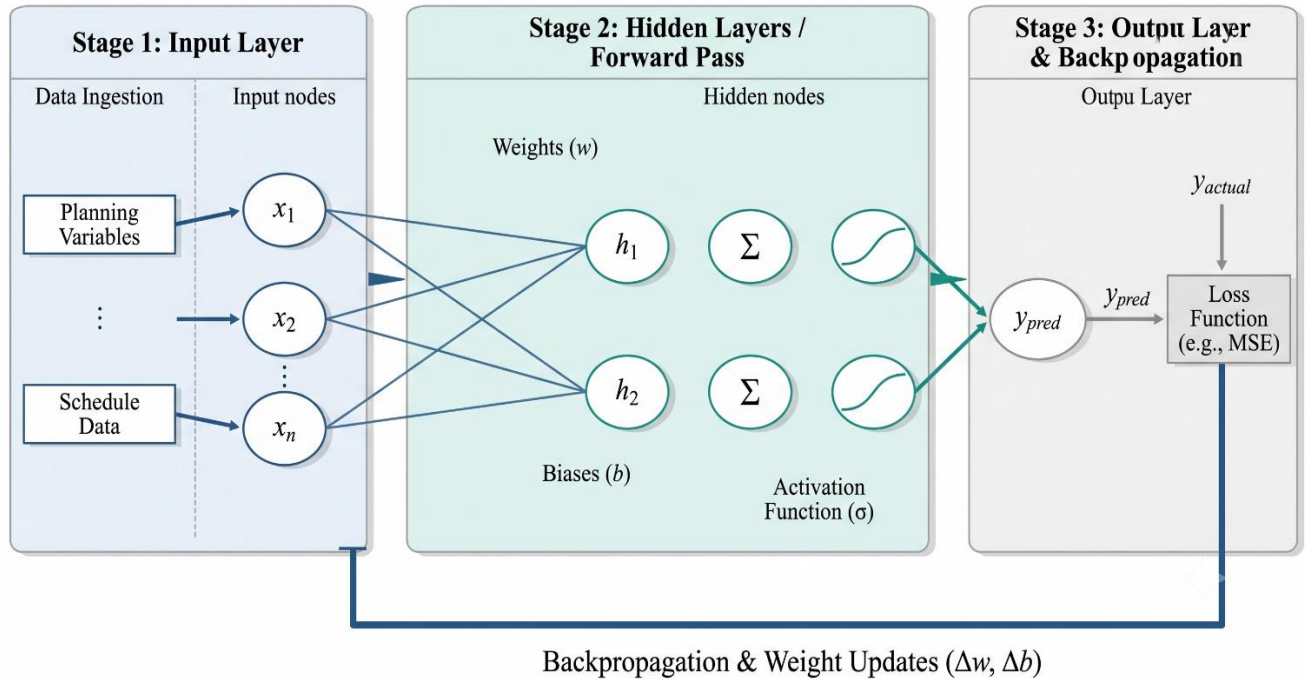


**Fig. 4 Operating mode of an Artificial Neural Network (ANN)**

Feed-Forward Neural Networks operate in two continuous loops of processing: forward pass and backpropagation of errors. The forward pass is the processing

of inputs in a forward sequence through the layers of a hidden structure to arrive at a predicted output. In a neural network, this processing involves the inputs being passed through

layers of neurons, where each neuron multiplies its inputs by the corresponding weights and sums them up with the neuron's bias to produce a linear summation. This linear summation is then passed through a non-linear activation function to produce a non-linear transformation or mapping. The output from each neuron in a hidden layer is passed to subsequent neurons in subsequent layers until the output layer is reached, where the network's predicted output is produced and compared with the target output to calculate an error. Backpropagation of errors then occurs by computing the error's gradients as they travel back through the hidden layers of a network. The weights of the connections between processing elements and between the elements and their inputs are then changed in a manner that minimizes the network's error, with the method of changing the weights typically depending on the network's optimization algorithm.

### 3.1.4. Mathematical Equations

- Feed-forward neuron input and activation [13]:

$$X_j = \sum \omega_{ij} \beta_j ; Y = \frac{1}{1+e^{-x_j}} \quad (1)$$

Which are:

- $X_j$ : Input to the  $jj$ -th neuron.
- $\omega_{ij}$ : Weight from the  $ii$ -th input to the  $jj$ -th neuron.
- $\beta_j$ : Bias for the  $j$ -th neuron.
- $Y_j$ : Output of the  $j$ -th neuron after activation.

- Output-layer gradient for backpropagation [13]:

$$\delta k = \frac{\partial E_n}{\partial a_k} = y_k - t_k \quad (2)$$

Where:

- $\delta k$ : Output error gradient.
- $E_n$ : Sample error.
- $a_k$ : Pre-activation.
- $y_k$ : Predicted value.
- $t_k$ : Target value.
- Hidden-layer gradient for backpropagation [13]:

$$\delta j = h'(a_j) \sum \delta_k w_{kj} \quad (3)$$

In this formulation:

- $\delta j$ : Hidden error gradient.
- $h'(a_j)$ : Activation derivative.
- $\delta k$ : Next-layer gradient.
- $w_{kj}$ : Forward weight.
- Weight updates (output layer) [13]:

$$\frac{\partial E_n}{\partial w_{kj}} = (y_k - t_k) \quad (4)$$

Which are:

- $E_n$ : Sample error.
- $w_{kj}$ : Output weight.
- $y_k - t_k$ : Output error.
- Weight updates (hidden layer) [13]:

$$\frac{\partial E_n}{\partial w_{kj}} = h'(a_j) x_i \sum \delta_k w_{kj} \quad (5)$$

In this expression:

- $w_{kj}$ : Output weight.
- $h'(a_j)$ : Derivative scale.
- $x_i$ : Input value.
- $\sum \delta_k w_{kj}$ : Propagated error.

- Mean Squared Error [14]:

$$MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2 \quad (6)$$

- MSE: Average squared error.
- $y_i$ : Actual value.
- $\hat{y}_i$ : Predicted value.

- Root Mean Squared Error [14]:

$$RMSE = \sqrt{MSE} \quad (7)$$

- Mean Absolute Error [14]:

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i| \quad (8)$$

- Mean Absolute Percentage Error [16]:

$$MAPE = \frac{1}{n} \sum \frac{|y_i - \hat{y}_i|}{y_i} \quad (9)$$

- Coefficient of determination [14]:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2} \quad (10)$$

Where

- $y_i$ : Actual value.
- $\hat{y}_i$ : predicted value,
- $\bar{y}_i$ : mean of the Actual value,
- $n$ : number of data points.

## 3.2. Support Vector Regression (SVR)

### 3.2.1. Principle

Support Vector Regression (SVR) tries to find a function  $f(x)$  that balances between a simple model and the training data by fitting the data points into an  $\epsilon$ -insensitive tube and penalizing them with slack variables. SVR can also make non-linear regression using kernel mappings like the RBF kernel.

So, for construction prediction, it is necessary to choose the right values of hyperparameters  $C$ ,  $\epsilon$ , and kernel width. These parameters can be chosen via cross-validation grid search or via some metaheuristics. The hyperparameters chosen via these methods result in a good generalization performance for

the SVR. For construction prediction, it is shown that it outperforms the linear regression and is competitive with Artificial Neural Network (ANN) for cost and schedule prediction tasks [9].

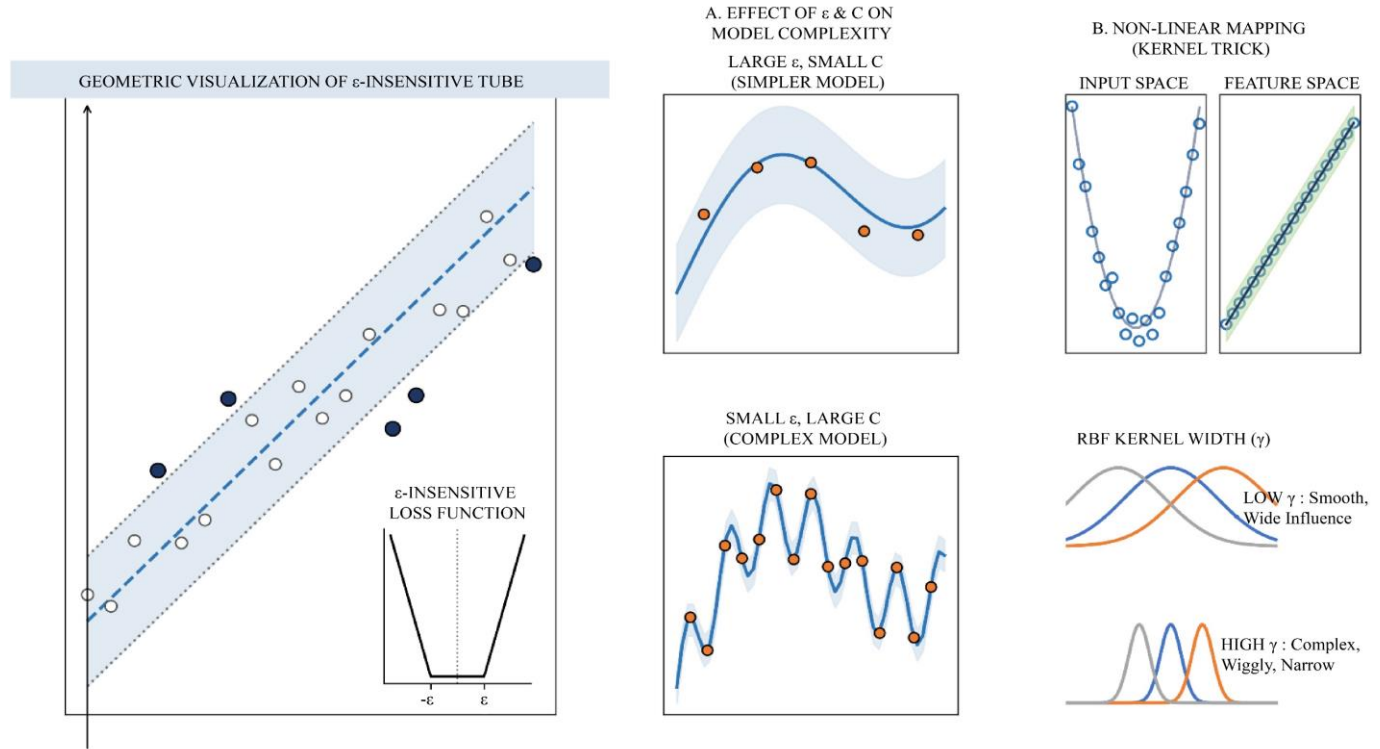


Fig. 5 Operational principle and optimization of Support Vector Regression (SVR)

The underlying mechanisms of the optimization process and the geometric interpretation of the mechanism for generating the model are shown in Figure 5. As illustrated by the construction of the epsilon-insensitive tube around the

training data, the non-linear mapping of the input space created by the kernel, and the delicate task of tuning the model's complexity by adjusting the values of its hyperparameters.

### 3.2.2. Advantages and Disadvantages

Table 2. Advantages and Disadvantages of Artificial Neural Networks (ANN)

| Advantages                                                                                                                                                                                                                              | Disadvantages                                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Highly generalizes on small-to-medium-sized datasets because the margin is maximized and the number of support vectors is sparsened, which tends to reduce overfitting compared to higher capacity classifiers [2, 20].                 | Performance may be sensitive to kernel choice and hyperparameters, which can be computationally very expensive to search over, especially for large searches [6, 19].                                                                                         |
| $\epsilon$ -Sensitivity SVM with a kernel-based approach that can robustly handle small errors in training data and simultaneously afford nonlinearity in classification surfaces without requiring large numbers of parameters [6, 9]. | Models may not perform well when faced with a distribution shift between training and deployment environments, and, therefore, it is important to validate the robustness of the models to such variations or to perform specific domain adaptations [6, 20]. |
| Exhibited strong accuracy for early-stage cost estimation and schedule planning ahead for the future in the construction industry [6, 9].                                                                                               | Scalability limitations for very large data sets, due to the kernel matrix operations and the support vector growth [2, 22].                                                                                                                                  |
| It can be optimized for better performance on nonlinear economic and engineering time series using hyperparameters with swarm or grid strategies [19, 21].                                                                              | Requires normalization and careful feature engineering to stabilize performance across heterogeneous project attributes [9, 22].                                                                                                                              |

### 3.2.3. Operating Mode of Support Vector Regression (SVR)

The operating mode of Support Vector Regression (SVR) focuses on constructing a predictive model that adheres strictly to a specified margin of tolerance ( $\epsilon$ ), rather than trying to minimize every training error to zero. Conceptually, SVR transforms input data (e.g., project features) into a high-dimensional feature space using non-linear kernel functions to handle complex, non-linear relationships. Within this space, it performs quadratic programming optimization to find a regression hyper-plane  $f(x)$  that is as flat as possible while

encompassing most data points within an  $\epsilon$ -insensitive tube. Deviations outside this tube are penalized using slack variables ( $\xi, \xi^*$ ), with the overall tolerance for error controlled by the regularization hyperparameter  $C$ . The optimal solution is uniquely defined by a subset of critical data points located on or outside the tube's boundary—the “Support Vectors.” This systematic algorithmic process is visually summarized in the flowchart below (Figure 6) and is subsequently defined by rigorous mathematical formulations.

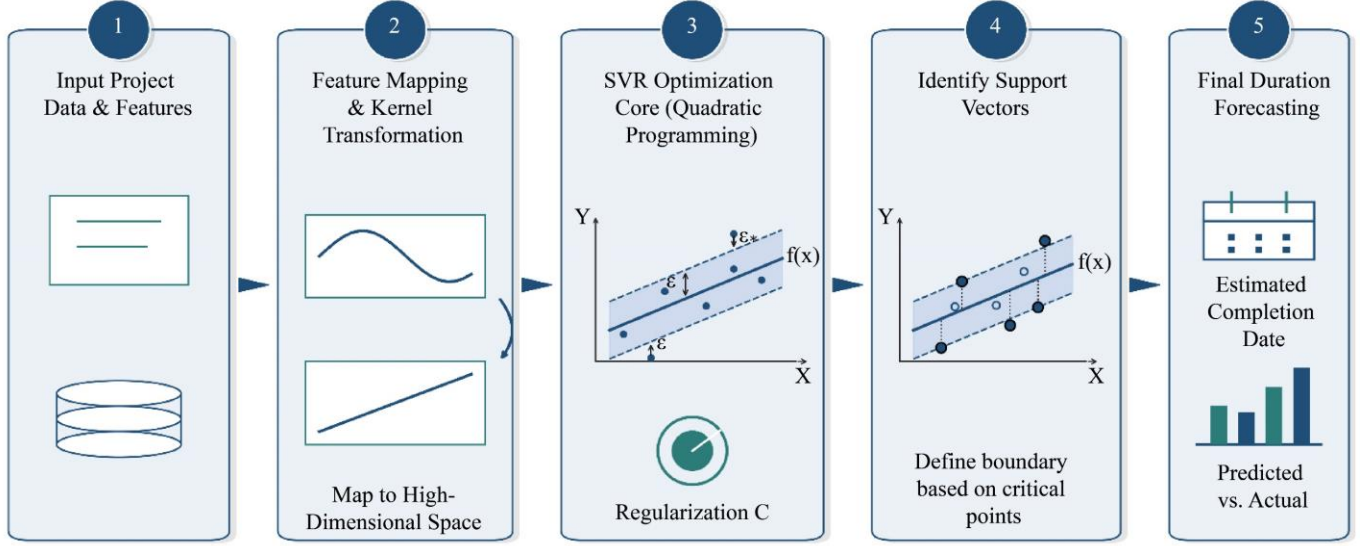


Fig. 6 Operational workflow of Support Vector Regression (SVR)

### 3.2.4. Mathematical Equations

- Linear SVR model [2]:

$$y = w^T + b \quad (11)$$

- Primal objective [2]:

$$\min_{w, \xi_i, \xi_i^*} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (12)$$

Subject to  $\epsilon$ -insensitive constraints:

$$\begin{cases} y_i - (w^T x_i + b) \leq \epsilon + \xi_i \\ (w^T x_i + b) - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

Which are:

- $\|w\|^2$ : squared norm of the weight vector, representing model complexity
- $C$ : regularization parameter, balances model complexity and estimation error
- $\xi_i, \xi_i^*$ : slack variables (tolerated deviations beyond the  $\epsilon$ -margin)

- $\epsilon$ : insensitive margin, a zone around the prediction where errors are not penalized
- $y_i$ : true value of the  $i$ -th data point
- $n$ : number of samples

- Dual predictor [2]:

$$f(x) = \sum_{i=1}^n (a_i - a_i^*) k(x_i, x) + b \quad (13)$$

In this expression:

- $a_i, a_i^*$ : Lagrange multipliers obtained during the dual problem optimization
- $k(x_i, x)$ : kernel function measuring similarity between  $x_i$  and  $x$
- $b$ : bias term
- $f(x)$ : prediction for input  $x$

- RBF kernel [6]:

$$K(x_i, x) = e^{(-\gamma \|x_i - x\|^2)} \quad (14)$$

Where

- $\gamma$ : kernel parameter controlling the width of the RBF kernel

- $\|x_i - x\|^2$ : squared Euclidean distance between the data points  $x_i$  and  $x$ .

### 3.3. Comparison and Hybridization between NN and SVR

Comparative evidence indicates SVR often yields lower errors than ANN on compact construction datasets, particularly for early-stage cost prediction and EVM-based schedule forecasting, due to its margin-based generalization and kernel flexibility [6, 9]. ANN-based approaches can be considered in case of large datasets, with complex non-linear relationships, and when the dataset has sequential correlations, because hybrid models based on dense ANNs and temporal encoders can reveal more complex relationships in execution dynamics and mixed input data [14, 17]. Hybridization strategies that pair SVR or ANN with metaheuristics or ensembles such as PSO-driven hyperparameter tuning, EMD-based decomposition plus PSO for input optimization, or AdaBoost-enhanced pipelines have demonstrated further accuracy gains and robustness on non-stationary or heterogeneous signals relevant to construction forecasting [21, 23].

### 3.4. Data Standardization Methods

#### 3.4.1. Min-Max Normalization

Min-Max Normalization preserves the relationships among the data while transforming it into a fixed range, generally [0, 1] or [-1, 1]. As mentioned by Gökhan Aksu, Cem Oktay Güzeller and Mehmet Taha Eser in the year 2019, Min-Max Normalization transforms the features into a given range (e.g. [0, 1] or [-1, 1]) while maintaining the relations among the values [24]. In a similar fashion, in 2022, Peshawa J. Muhammad Ali defines Min-Max Normalization as a Data Normalization technique that scales the data to a fixed range, typically [0, 1] or [-1, 1] [25]. In 2022, Thanh Ngoc Tran and others similarly defined it as a technique that scales the data to a fixed range, which is typically [0, 1] [26, 27]. According to Rahmad Ramadhan Laska and Anne Mudya Yolanda (2024), Min-Max Normalization is the process of scaling the data to a fixed range, generally between (0 and 1) (Laska & Yolanda, 2024) [28]. Mohammed Z. Al-Faiz, Ali A. Ibrahim and Sarmad M. Hadi (2018) confirm it scales the data to a fixed range, typically [0, 1] [29].

#### Purpose

The most important reason to use Min-Max Normalization in prediction models like Neural Networks and SVR is to remove biases of input features, with each feature on a different scale. Gökhan Aksu et al. (2019) state that its role is needed to make the contributions of all input features to the model learning of the same order by removing scale bias. Peshawa J. Muhammad Ali (2022) further claims that it removes the effect of different scales of features with respect to machine learning task performance [24, 25].

Thanh Ngoc Tran et al. (2022) explain that this is done for the sake of the input data being on a comparable level of

measurement to improve model performance [26]. Yasemin Onal (2021) and Mohammed Z. Al-Faiz et al. (2018) support this by stating that it avoids all input features to contribute equally when calculating the distance in machine learning [27, 29]. Rahmad Ramadhan Laska and Anne Mudya Yolanda (2024) describe it as employed to improve the machine-learning performance by scaling features down into a similar range [28].

#### Advantages and Disadvantages

Min-Max Normalization offers several benefits when applied to forecasting models; however, despite these advantages, it still presents certain limitations.

**Table 3. Advantages and Disadvantages (Min-Max Normalization)**

| Advantages                                                                                                                                                                                   | Disadvantages                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| It preserves the relationships among data values and is easy to understand and implement, which makes it efficient to use with algorithms that are sensitive to data magnitude [24, 26, 29]. | It is sensitive to outliers; extreme values can have a substantial effect [24, 27, 28].                                    |
| All of the attributes shall have the same bearing on the similarity computations. Makes the training of the model efficient [25].                                                            | It does not address dimensionality or the complexity of the data, nor does it cater for the new data point value [24, 26]. |
| Easy to use and understand [26, 27, 28].                                                                                                                                                     | The performance could degrade if the dataset contains outliers; it is sensitive to differences in feature range [25].      |

#### Mathematical Equations

The formula for Min-Max Normalization, as provided by Peshawa J. Muhammad Ali (2022), is [25]:

$$x' = \frac{x - \text{oldMin}}{\text{oldMax} - \text{oldMin}} \times (\text{newMax} - \text{newMin}) + \text{newMin} \quad (15)$$

Where:

- $x'$ : normalized value.
- $x$ : original value.
- $\text{oldMin}$ : minimum value of the original dataset.
- $\text{oldMax}$ : maximum value of the original dataset.
- $\text{newMin}$  and  $\text{newMax}$ : desired minimum and maximum values for the normalized data (0 and 1).

Mohammed Z. Al-Faiz et al. (2018) provide a simplified version for the range 0,1 [29]:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (16)$$

### 3.4.2. Z-Score Normalization

The popular Z-Score Normalization, also known in literature as standardization, transforms input features into a standard normal distribution with mean 0 ( $\mu = 0$ ) and unit variance ( $\sigma = 1$ ) [27, 30]. In this method, the data is rescaled by subtracting the overall mean of the population from each observation and then dividing by its standard deviation [28].

#### Purpose

The main purpose of this normalization in the forecasting models is to center the data and scale variance, making sure that the distributional assumptions behind a great part of the statistical algorithms that are employed for forecasting [27, 30] hold.

Compared to Min-Max scaling, Z-Score standardization makes no change in the shape of the original data distribution and copes better with outliers (making regression-based models such as SVR [28] more robust to outliers). Moreover, in the context of ANN, this algorithm is very useful to speed up learning, converging faster, especially with complex or binary data sets [29].

#### Advantages and Disadvantages

Z-Score Normalization provides distinct advantages for forecasting models; however, the method presents the following drawbacks.

**Table 4. Advantages and Disadvantages (Z-Score Normalization)**

| Advantages                                                                                                    | Disadvantages                                                                                                  |
|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| It is less sensitive to outliers compared to Min-Max [27, 30].                                                | The assumption of normal distribution of the data may not always hold. [27, 28, 30].                           |
| It “improves convergence speed and facilitates comparison across different datasets [29].                     | It is affected by extreme values, whereby prior knowledge of the mean and standard deviation is required [29]. |
| It is resistant to outliers and preserves the distribution shape, allowing for better model performance [28]. | It may not perform well when the data is non-Gaussian [30].                                                    |

#### Mathematical Equations

The formula for Z-Score Normalization, as provided by Mohammed Z. Al-Faiz et al. (2018), is [29]:

$$z_{ij} = \frac{a_{ij} - \mu}{\sigma} \quad (17)$$

Where:

- $z_{ij}$  is the normalized value.
- $a_{ij}$  is the original value.

- $\mu$  is the mean of the dataset.
- $\sigma$  is the standard deviation.

Yasemin Onal (2021) provides a variant adjusted by sample size [27]:

$$D_{norm} = \frac{D - m}{s/\sqrt{n}} \quad (18)$$

Where:

- $D_{norm}$  is the normalized value.
- $D$  is the original value.
- $m$  is the mean.
- $s$  is the standard deviation.
- $n$  is the sample size.

### 3.5. Results

#### 3.5.1. Cost Prediction

- An SVR with RBF kernel achieved  $R^2 \approx 0.943$  and MAPE  $\approx 8.96\%$  on 78 residential building projects, outperforming linear, polynomial, and sigmoid kernels, and highlighting Typical Floor Area as the most influential feature, demonstrating early-stage applicability in practice [9, 23].
- ANN-based cost estimation for engineering services reached MAPE  $\approx 13.65\%$  with selected key variables, illustrating accuracy with limited data but highlighting interpretability challenges [3, 16].

#### 3.5.2. Duration/Schedule Forecasting

- SVR trained on EVM-derived indicators outperformed traditional control formulas in simulated portfolios of 900 networks, with lower MAPE for time and cost and robust performance when training and test distributions were aligned, supporting its use as an early warning predictor in project control [5, 6].
- Genetic algorithm scheduling under resource and supply constraints reduced deviations on PSPLIB benchmarks and shortened real-case completion under delays, demonstrating optimization's role alongside predictive models [4, 31].

#### 3.5.3. Global Performance and Risk Prediction

- In success rating models using 30 critical factors, SVR outperformed ANN and MLR with  $R^2 \approx 0.881$  and MAE  $\approx 2.172$ , indicating superior accuracy on modest-sized datasets [10, 11].
- BIM-integrated BP neural networks enabled full life-cycle risk prediction with deviations below  $\approx 3.57\%$  in specific phases, supporting proactive control in large public buildings [13].

#### 3.5.4. Dynamic Schedule and Resource Management

- Integrating a BP-ANN within a fuzzy-EVM framework reduced MAPE by about 32% versus traditional methods

in a field case, demonstrating ANN's value for proactive and uncertainty-aware schedule control [7, 8].

- RBF-FLNN resource scheduling minimized variance by  $\approx 70\%$  across operations spaces, improving balance and lowering idle time in prefabricated projects [31, 32].

### 3.5.5. Specialized Forecasting and Domain Applications

- Interval forecasting via ANN with optimal LUBE produced reliable prediction intervals for material prices, outperforming point estimates and quantifying uncertainty for asphalt and steel [1, 15].
- Gaussian Process Regression, SVR, and DT for tunnel geology/time/cost forecasting showed that updating with short segments of constructed data substantially improves accuracy, with GPR achieving  $R^2 \approx 0.99$  post-update [20, 33].
- An SVR optimized by PSO combined with signal decomposition is proposed for predicting landslide displacement and proved to be efficient in dealing with the non-stationary characteristic of landslide movement velocity series [19, 21].
- By leveraging smart grid DRNN Bi-LSTM technique to forecast the output power of microgrids, combining them with PSO/SPSO power optimization algorithms, the power losses and costs can be reduced, and instability caused by renewable energy fluctuations in microgrids can be solved, as shown in [17, 22].
- Improved operational reliability by performing predictive maintenance at optimal times using a scheduling system integrated with the Deep Neural Network (DNN)/Recurrent Neural Network (RNN) that accurately estimated the Remaining Useful Lifetime (RUL) of assets, resulting in less unplanned shutdown time [34, 35].
- Construction cost indices were modeled using time series and NN approaches, where autoregressive models gave lower average errors than pure regression, which assisted in cost budgeting [1, 23].
- The proposed Graph-based Schedule Mining method was shown to automate pre-construction scheduling through the mining of frequent activity sequences to obtain higher time productivity in the analysed case studies [4, 36].
- Soil organic carbon mapping demonstrated SVR's precision with environmental predictors, offering transferable lessons on model selection and validation for construction-related geospatial analytics [2, 20].
- Pavement performance prediction via ANN achieved high accuracy for PCI with sensitivity identifying critical distresses, supporting PMS decisions [10, 18].

### 3.6. Conclusion (of the State of the Art)

ANN and SVR are now reference approaches for predicting costs and schedules in construction, with SVR recommended when data is limited, and robustness is a priority, while ANN is essential for capturing complex nonlinearities and leveraging sequential or lifecycle data at scale [9, 10]. The recommended selection protocol is

empirical: evaluate both families under consistent cross-validation, compare  $R^2$ , MAE, RMSE, and MAPE, and assess sensitivity to distribution shift to justify the best approach for deadline prediction in the target context, potentially integrating hybrid strategies for further gains [6, 21].

## 4. Case Study

### 4.1. Explanation of the Context

#### 4.1.1. Project Location and Regional Significance

The empirical data and simulated scheduling scenarios utilized in this research are deeply grounded in the context of large-scale educational infrastructure development within the Marrakesh-Safi region of Morocco. Over the past decade, this region has emerged as a premier hub for academic excellence and sustainable urban development, prominently featuring advanced academic centers and smart-city campuses. The construction of modern school infrastructure in this specific climate and geographic location presents unique logistical and engineering challenges. High temperature variations, specific soil geomechanics typical of the Haouz plain, and strict urban planning regulations necessitate highly rigorous project management and scheduling frameworks. The educational facilities targeted in this study are not standard structures; they are designed as integrated, high-performance learning environments. These projects encompass multiple multi-story academic blocks, specialized technical laboratories, administrative hubs, and extensive external landscaping. Delivering such multifaceted projects within stringent deadlines—often dictated by the immutable start of the academic school year—requires zero tolerance for critical path delays. Consequently, predicting the final project duration with absolute precision becomes a critical success factor for regional development authorities and investment stakeholders.



Fig. 7 Aerial view or Google Earth Map of the Marrakesh region, highlighting the educational district development zone

#### 4.1.2. Construction Process and Scheduling Workflow

To understand the intricate variables present in the dataset (such as total relationships, sum of lag, and total float), it is essential to visualize the physical construction process that the Principal Scheduler manages. One of the most critical and time-sensitive sequences in school construction is the

Reinforced Concrete (RC) Structural Works, specifically the cycle of pouring structural slabs and columns.

This process is governed by strict physical dependencies and mandatory curing times, which translate directly into the “Lag” and “Relationship” variables in the dataset.

The workflow follows a highly sequential, yet overlapping, logic:

- **Formwork Installation:** Erection of timber or steel molds for columns and slabs.
- **Steel Rebar Fixing:** Placement of structural reinforcement. This activity often overlaps with formwork installation (Start-to-Start relationship with a lag).
- **MEP First Fix (Embedment):** Integration of electrical conduits and plumbing sleeves within the rebar before concrete placement.
- **Concrete Pouring:** The critical physical activity, requiring continuous logistical supply of ready-mix concrete.

- **Curing Period (Lag):** A mandatory technical lag (typically 7 to 14 days for partial load-bearing) during which no subsequent load-bearing activities can commence. This introduces significant “Total Float” for non-structural parallel tasks while locking the critical path.
- **Formwork Stripping:** Removal of molds, allowing the cycle to repeat for the next floor.

Any disruption in this micro-process—such as delayed rebar delivery or extended curing due to weather—propagates through the entire schedule, altering the final project duration. The machine learning models developed in this study (ANN and SVR) are specifically designed to capture and predict the compounding effects of these localized process variations on the macroscopic project deadline.

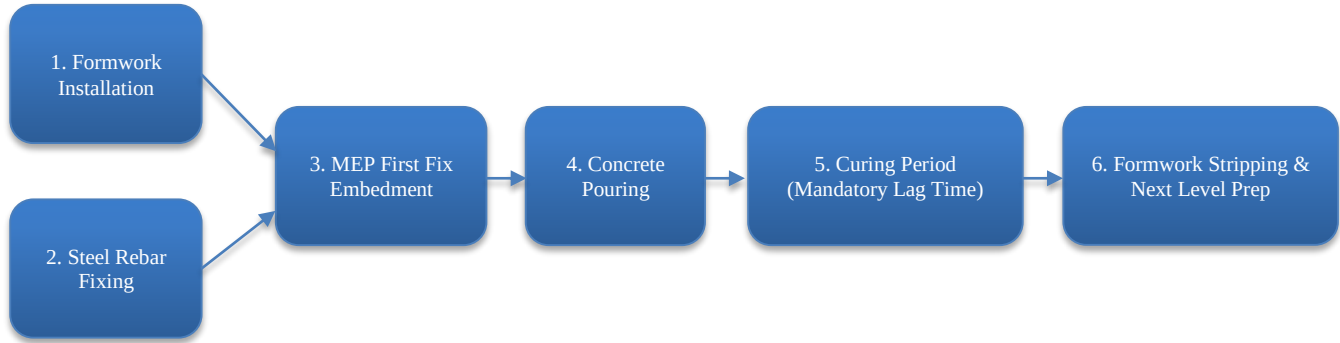


Fig. 8 Reinforced concrete structural workflow

## 4.2. Data Description

### 4.2.1. Dataset Overview

The School Construction dataset consists of information about a real school construction project, covering every stage of the construction cycle, including community engagement, clearing and grading the land, foundation, ground floor, first and second floors, roofing works, external works including architectural & MEP/technical drawings, as well as final inspection and hand over.

Multiple scenarios were modeled to produce outcomes such as total project duration and critical-path behavior. Each scenario uses logical dependencies to simulate a unique critical path for every run. The scenarios include a baseline plan as well as variations such as task compression, task extension, activity overlapping, and reduced-scope schedules. The 157 schedules were run in this exercise. A small data excerpt is shown below.

Table 5. Input variables and output

|        |                                           |        |        |        |        |         |         |         |         |         |
|--------|-------------------------------------------|--------|--------|--------|--------|---------|---------|---------|---------|---------|
| Input  | Total Activities                          | 51.00  | 51.00  | 51.00  | 51.00  | 51.00   | 51.00   | 51.00   | 51.00   | 51.00   |
|        | Sum of Total Float                        | 765.00 | 829.00 | 905.00 | 969.00 | 1043.00 | 1104.00 | 1178.00 | 1244.00 | 1318.00 |
|        | Total Count of Relationship               | 67.00  | 67.00  | 67.00  | 67.00  | 67.00   | 67.00   | 67.00   | 67.00   | 67.00   |
|        | Sum of Lag                                | 0.00   | 0.00   | 0.00   | 0.00   | 0.00    | 0.00    | 0.00    | 0.00    | 0.00    |
|        | Average of Original Duration per Activity | 8.98   | 9.39   | 10.41  | 11.22  | 12.29   | 12.67   | 13.69   | 14.49   | 15.12   |
|        | Max of Original Duration per Activity     | 22.00  | 24.00  | 26.00  | 28.00  | 30.00   | 32.00   | 34.00   | 36.00   | 38.00   |
| Output | BL Project Duration                       | 196    | 214    | 231    | 249    | 267     | 285     | 303     | 320     | 338     |

### 4.2.2. Input Variables

Each simulation is characterized by six key planning parameters:

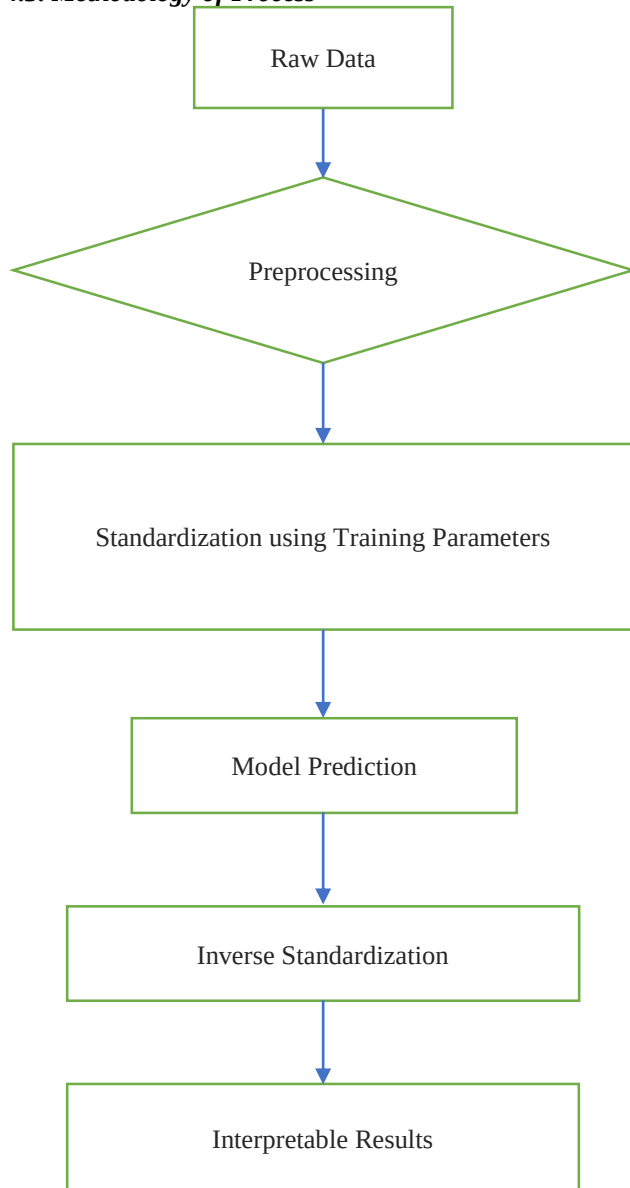
**Table 6. Input data description**

| Variable Name                      | Description                                                  |
|------------------------------------|--------------------------------------------------------------|
| Total Activities                   | Number of activities in the project schedule                 |
| Sum of Total Float                 | Aggregate float (slack) across all activities                |
| Total Count of Relationship        | Number of logical relationships (dependencies) between tasks |
| Sum of Lag                         | Total lag time across all relationships                      |
| Average Original Duration/Activity | Average planned duration per activity                        |
| Max Original Duration/Activity     | Maximum planned duration among all activities                |

#### 4.2.3. Output Variable

Project Duration (BL Project Duration): The total duration of the simulated project schedule.

#### 4.3. Methodology of Process

**Fig. 9 Data processing and prediction workflow**

#### 4.3.1. Raw Data

Raw data is the original information created during a real simulation. Multiple simulation scenarios are called data description. All variables, input features and output target are included in the raw data. It is the base for further data analysis and modeling.

#### 4.3.2. Preprocessing

Preprocessing is the process of preparing the raw data in a consistent manner that can be appropriate for modeling. This can involve filling in missing values, cleaning outliers, converting variables to a suitable data type, standardizing the data format and selecting a subset of the variables that can be included in a model, all in an effort to remove sources of inherent or induced bias and error in the data.

#### 4.3.3. Standardization using Training Parameters

Standardization via training parameters refers to the process of changing the features to the same scale via statistical calculations that are based only on the training data.

This has important implications as it ensures that different variables with different units and scales do not receive different weights during the modeling process. Popular normalization techniques that fall under standardization via training parameters are Min-Max and Z-Score normalization.

#### 4.3.4. Model Prediction

Model prediction is the process of applying a trained model to a dataset of preprocessed input features and target values in order to generate the predicted output. Model prediction utilizes the knowledge extracted from the training data and learned by the model during the training phase to produce estimated values for new, unseen data.

#### 4.3.5. Inverse Standardization

Inverse standardization is the process of taking the model predictions from the standardized form and bringing them back to the original data scale. This process uses the same standardization data as was used for the feature engineering and scaling in the forward standardization.

This step employs in the initial standardization process so that comparisons can be made directly with field data.

#### 4.3.6. Interpretable Results

The interpretable predictions are the last outputs of the modeling pipeline in the unit of the original data. These outputs provide actionable insights that reflect real-world behaviors. To evaluate the quality of these predictions, a measure of the goodness of fit between predictions and reality is required. Several statistical metrics are relevant for this, such as Mean Squared Error (MS, or RMS or RMSE), Coefficient of Determination ( $R^2$ ), which are all related to quantifying the errors or variations between the true and the simulated values and can be used to compare model

performances or to characterize the differences between different results from the models.

#### 4.4. Forecasting Model

##### 4.4.1. ANN Model Structure

Architecture of the proposed Artificial Neural Network (ANN), featuring a six-node input layer representing planning variables, one or two hidden layers (depending on the simulation topology) with optimized neurons, and a linear output layer for duration forecasting.

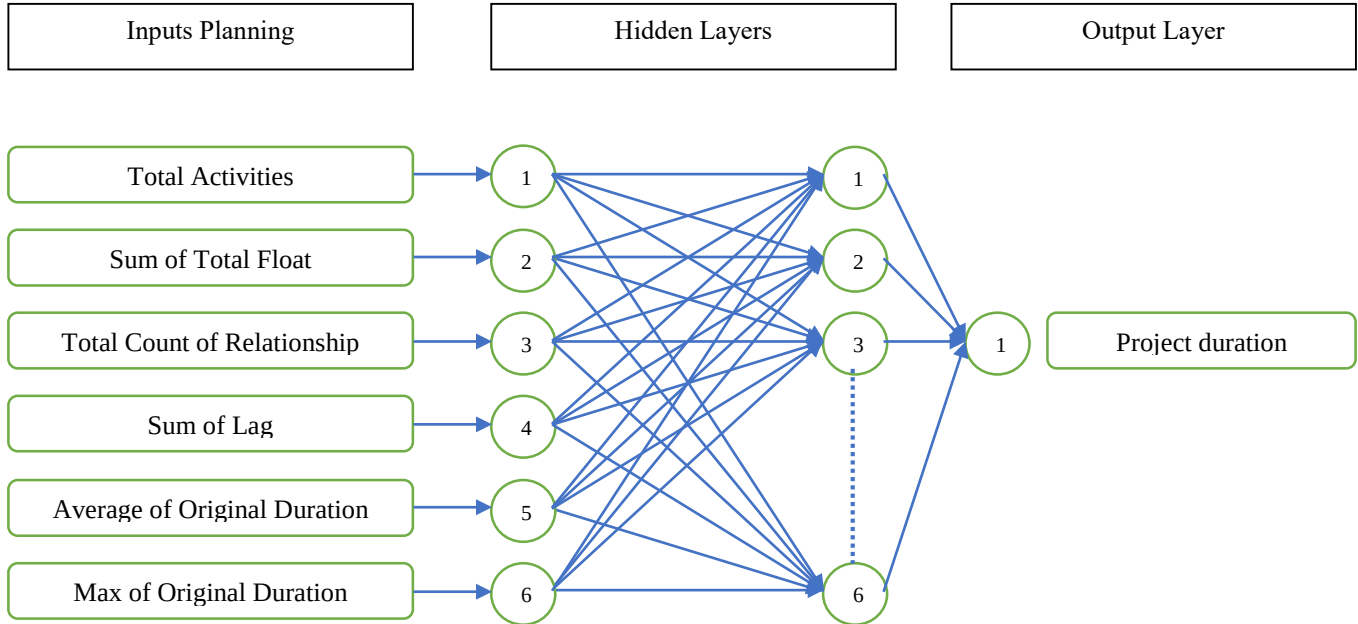


Fig. 10 Artificial Neural Network model structure

- Input Layer: 6 neurons (one for each planning variable)
- Hidden Layers: Multiple topologies tested, with hidden neurons ranging from 10 to 20 (in steps of 2)
- Output Layer: 1 neuron (predicted project duration)
- Network type: Feed-forward backprop
- Training Algorithm: Backpropagation with trainlm, traingda or trainoss optimizers, as appropriate for each topology tested.
- Transfer Function: TANSIG, PURELIN or LOGSIG, as appropriate for each topology tested.
- Loss Function: Mean Squared Error (MSE)

##### 4.4.2. SVR Model Structure

- Input Features: The model uses 6 input features corresponding to planning variables.
- Output Feature: predicted project duration.
- Kernel Function: A linear kernel is employed for the SVR model.
- Hyperparameters: Key hyperparameters include:
  - Box Constraint: Controls the trade-off between achieving a low training error and a low testing error.

- Epsilon: Defines the margin of tolerance for errors in the predictions.
- Kernel Scale: Affects the scaling of the input data in the kernel function.

##### 4.5. Data Partitioning

The data is read from the Excel file named Data.xlsx, where any six columns are the input variables and the seventh is the target variable.

MATLAB is used for all processing involved in pre-processing and model simulation, is performed in MATLAB.

##### 4.5.1. Neural Network Model

The neural network model divides the dataset as follows:

- The dataset is divided into training, validation, and test sets, with 70% of the data used for training.
- 15% of the data used is for the validation set.
- Fifteen percent of the data is used for testing.

Each split has a fair representation of all scenario types in the data split.

#### 4.5.2. SVR Model

This sentence states that a holdout method is applied to the support vector regression model:

- 70% of the data is used in the training set.
- 30% of the data is used for testing.

This partitioning guarantees that both sets preserve a representative distribution of all variety scenarios.

#### 4.6. Model Training and Evaluation

##### 4.6.1. Hyperparameter Tuning

Hyperparameters were sped up on a systematic basis to minimize the error of prediction while also preventing overfitting. Refinements can be carried out automatically (e.g. Bayesian search) where relevant and appropriate. For Neural Networks (NN), the number of neurons per layer, learning rate, number of epochs, activation function, and training algorithm were varied. The methodology starts by creating a search space that includes architectures with one to two hidden layers and 10 to 20 neurons per layer.

$$N_H = 0.5(N_I + N_O) + N_T^{1/2} \quad (19)$$

- $N_H$ : Number of hidden layers
- $N_I$ : Number of inputs, which is 6
- $N_O$ : Number of outputs, which is 1
- $N_T$ : Number of training data 157 x 70%

A grid search with k-fold cross-validation is applied to the models to find the combination of parameters leading to the

smallest average validation MSE on the normalized dataset obtained with Min-Max or Z-Score scaling. The optimal configurations are then used to train new models with early stopping and regularization applied when needed. Otherwise, the Support Vector Regression (SVR), the explored hyperparameters include kernel type (linear, quadratic, cubic and Radial Basis Function). The parameters BoxConstraint (regularization strength), Epsilon (insensitive loss margin), and KernelScale are launched automatically via multiple iterations throughout the MATLAB code to find the Best Hyperparameters, which are used on the kernel type.

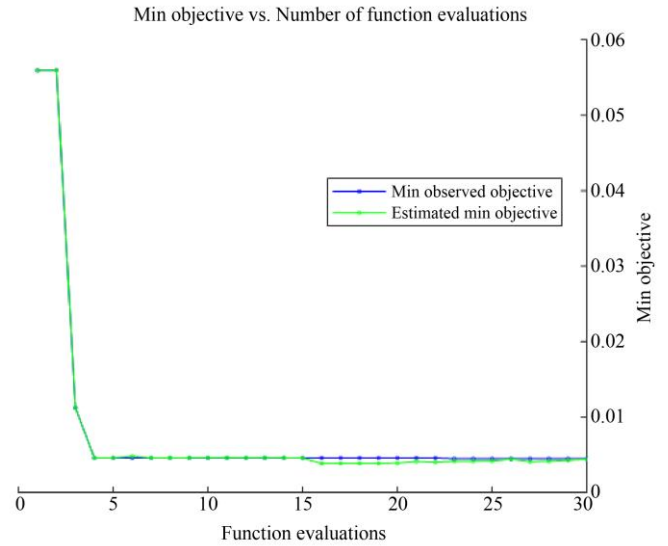


Fig. 11 Function evolution with minimum objective

Table 7. SVR hyperparameter optimization in MATLAB (screenshot from MATLAB)

| Iter | Eval result | Objective log(1+loss) | Objective runtime | BestSoFar observed | BestSoFar estim. | BoxConstraint | KernelScale | Epsilon    |
|------|-------------|-----------------------|-------------------|--------------------|------------------|---------------|-------------|------------|
| 21   | Accept      | 0.0046087             | 0.1574900         | 0.0045843          | 0.0041278        | 292.880000    | 13.333000   | 0.00093910 |
| 22   | Accept      | 0.0046750             | 0.0639630         | 0.0045843          | 0.0040178        | 883.220000    | 151.240000  | 0.02079900 |
| 23   | Best        | 0.0044746             | 0.1561800         | 0.0044746          | 0.0041514        | 529.840000    | 32.928000   | 0.01003300 |
| 24   | Accept      | 0.0054173             | 0.0773390         | 0.0044746          | 0.0041545        | 0.001199      | 0.044950    | 0.00027743 |
| 25   | Accept      | 0.0046686             | 1.7427000         | 0.0044746          | 0.0041695        | 17.918000     | 0.806050    | 0.00032000 |
| 26   | Accept      | 0.0045460             | 0.1543600         | 0.0044746          | 0.0045770        | 0.024039      | 0.095227    | 0.00030736 |
| 27   | Accept      | 0.0071986             | 0.0880060         | 0.0044746          | 0.0040901        | 599.130000    | 347.860000  | 0.00555160 |
| 28   | Accept      | 0.0046051             | 0.3524600         | 0.0044746          | 0.0041482        | 65.279000     | 3.617700    | 0.00065091 |
| 29   | Accept      | 0.0046184             | 0.1751700         | 0.0044746          | 0.0042880        | 779.290000    | 23.055000   | 0.00029452 |
| 30   | Accept      | 0.0046449             | 0.9901800         | 0.0044746          | 0.0044370        | 239.590000    | 3.877400    | 0.00028126 |

Optimization completed.

Max Objective Evaluations of 30 reached.

Total function evaluations: 30

Total elapsed time: 138.1324 seconds.

Total objective function evaluation time: 101.8839

Best observed feasible point:

| BoxConstraint | KernelScale | Epsilon  |
|---------------|-------------|----------|
| 529.84        | 32.928      | 0.010033 |

The MATLAB optimization results show that the SVR model achieved its best performance at iteration 23, where the minimum objective value reached 0.0044746 using a BoxConstraint of 529.84, a KernelScale of 32.928, and an Epsilon value of 0.010033. The objective function decreased sharply during the first evaluations, then became almost stable, indicating that the optimization process quickly identified a promising region of hyperparameters. The relatively high BoxConstraint suggests that the model gives more importance to reducing training errors, while the selected KernelScale allows the model to capture nonlinear relationships in the dataset without excessive complexity. In addition, the small Epsilon value indicates a narrow insensitive margin, which helps improve prediction precision. After 30 function evaluations, MATLAB stopped the optimization because the maximum number of evaluations was reached, with a total elapsed time of 138.1324 seconds. Overall, these results confirm that the optimized SVR configuration provides a stable and efficient compromise between prediction accuracy and generalization ability.

The second step, including NN and SVR, the procedure begins with grid search and k-fold cross-validation conducted for the best NN topology and each kernel SVR type, followed by automated refinement of ranges around the top-performing result. Evaluation is performed on data normalized with Min-Max and Z-Score scaling, with inverse scaling applied to predictions to enable comparison on the original measurement scale. Configurations are selected based on the minimization of average validation error, ensuring a balance between bias and variance. Crucially, to strictly prevent data leakage during the k-fold cross-validation process, all normalization parameters (minimum, maximum, mean, and standard deviation) were computed exclusively on the training subset of each fold. These isolated parameters were subsequently applied to transform the corresponding validation subset.

#### 4.6.2. Performance Metrics

Performance is reported on normalized and denormalized data (inverse Min-Max applied) to ensure interpretability in

the original physical scale. Metrics are computed for training, validation/test, and the combined dataset:

- Root Mean Square Error (RMSE)
- Mean Squared Error (MSE)
- Absolute Error per Simulation (instance-wise absolute deviation)
- Coefficient of Determination ( $R^2$ )

#### 4.6.3. Scenario Analysis

This section discusses the analysis of errors in the context of three specific scenarios, which are compression, overlap and reduction of scope. In this section, the predictions are compared against the normalized and denormalized data of the training, validation/test sets, as well as the combined sets. The denormalization of the normalized data is carried out by using the inverse of Min-Max scaling, such that the predictions are obtained in the original scale.

### 4.7. Results

#### 4.7.1. Predictive Performance

##### Neural Network

In this study, five different feedforward neural network models were built, trained and tested in order to compare the performance of different numbers of neurons in the hidden layer (depth and width), different activation/transfer functions and different optimization methods in a prediction experiment for various angles of attack. All models were trained and validated with the same training set, and the same test set was used for testing. The input data were preprocessed using two well-known methods in the literature: Min-Max scaling and Z-score normalization.

The performance of the model was evaluated by calculating the Mean Squared Error (MSE) and Coefficient of Variation ( $R^2$ ) metrics in the validation split. The characteristics of the models used for neural networks are indicated in the table below: number of hidden layers and neurons, activation/transfer functions and training algorithms. The values of MSE and  $R^2$  are shown for each of the two normalization methods.

**Table 8. Performance comparison of Neural Network topologies and training parameters for forecasting**

| Forecasting Type | Topology   | No of Hidden Layers | Neurons per Layer | Training Algorithm | Transfer Function | MSE (Min-Max) | R2 (Min-Max) | MSE (Z-Score) | R2 (Z-Score) |
|------------------|------------|---------------------|-------------------|--------------------|-------------------|---------------|--------------|---------------|--------------|
| NN               | Topology 1 | 1                   | 10                | trainlm            | TANSIG            | 6.95E-06      | 0.99993      | 0.01039       | 0.99494      |
| NN               | Topology 2 | 2                   | 15                | traingda           | PURELIN           | 0.01149       | 0.90990      | 0.19457       | 0.93090      |
| NN               | Topology 3 | 3                   | 10                | trainoss           | TANSIG            | 0.00108       | 0.98985      | 0.58687       | 0.67485      |
| NN               | Topology 4 | 1                   | 15                | trainlm            | PURELIN           | 0.00322       | 0.96813      | 0.06214       | 0.96839      |
| NN               | Topology 5 | 2                   | 20                | traingda           | LOGSIG            | 0.04135       | 0.84639      | 0.78125       | 0.83209      |

The funding emphasizes the advantage of compact structures with optimal Levenberg–Marquardt training. Topology 1: a single hidden layer with 10 neurons and TANSIG achieved the highest scores for both cases of normalization, being extremely well fitted to the observed data in Min–Max (MSE of  $\sim 6.95 \times 10^{-6}$ ;  $R^2 = 0.99993$ ) and highly robust under Z-score (with an  $R^2$  of 0.99494). Topology 4 also trained with Levenberg–Marquardt ensured competitive accuracy; however, with a higher Min–Max error (MSE = 0.00322;  $R^2 = 0.96813$ ) and similar Z-score performance (MSE = 0.06214,  $R^2 = 0.96839$ ), indicating that networks benefit from nonlinear hidden activations when the capacity is limited, as was the case here. An under-performance was observed in architectures trained with optimizers that do not use a full gradient descent style approach. Topology 2 (2 layers, PURELIN, traingda) and Topology 5 (2 layers, LOGSIG, traingda) presented considerably lower accuracy and  $R^2$  values compared to other topologies, especially for the Z-score normalization model (e.g., Topology 5: MSE =

0.78125;  $R^2 = 0.83209$ ), meaning that convergence was slower or sub-optimal.

Topology 3 (3 layers, TANSIG, trainoss) showed intermediate outcomes better than traingda in Min–Max (MSE = 0.00108;  $R^2 = 0.98985$ ) yet substantially degraded under Z-score (MSE = 0.58687;  $R^2 = 0.67485$ ), consistent with increased optimization difficulty and sensitivity to scaling in deeper networks. After evaluating the data in their normalized form, an Inverse Standardization step was adopted for the performance evaluation. For the highest-performing Topology 1, the MSE (Original Data) values were Training 0.68375 (RMSE 0.82689), Validation 1.11942 (RMSE 1.05803), Test 3.90688 (RMSE 1.97658), and All 1.24306 (RMSE 1.11493). In the context of the actual data, this MSE of 1 represented an error of a day, showing the model’s accuracy in practice, particularly in relation to the training and validation sets, where it is below this level, while the test set shows slightly more variance.

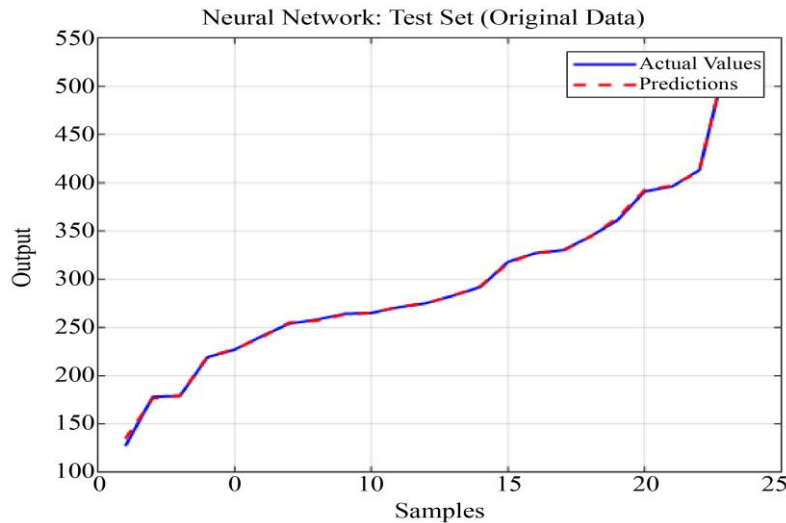


Fig. 12 NN performance on test set (original data) for topology 1

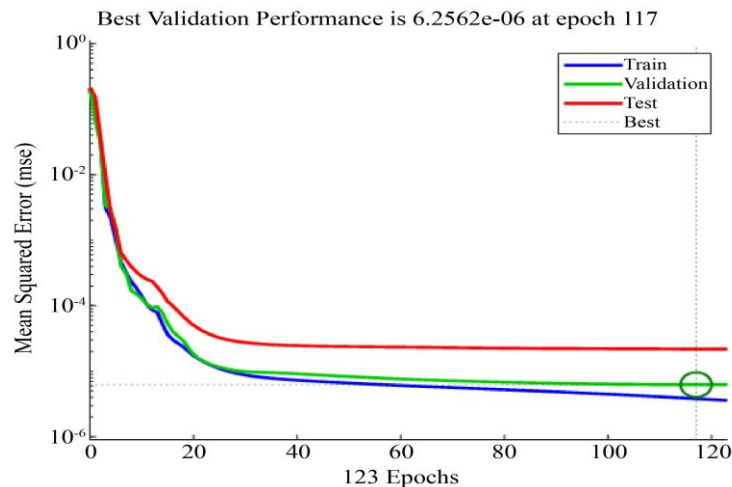


Fig. 13 Training, validation, and test MSE over Epochs for top 1

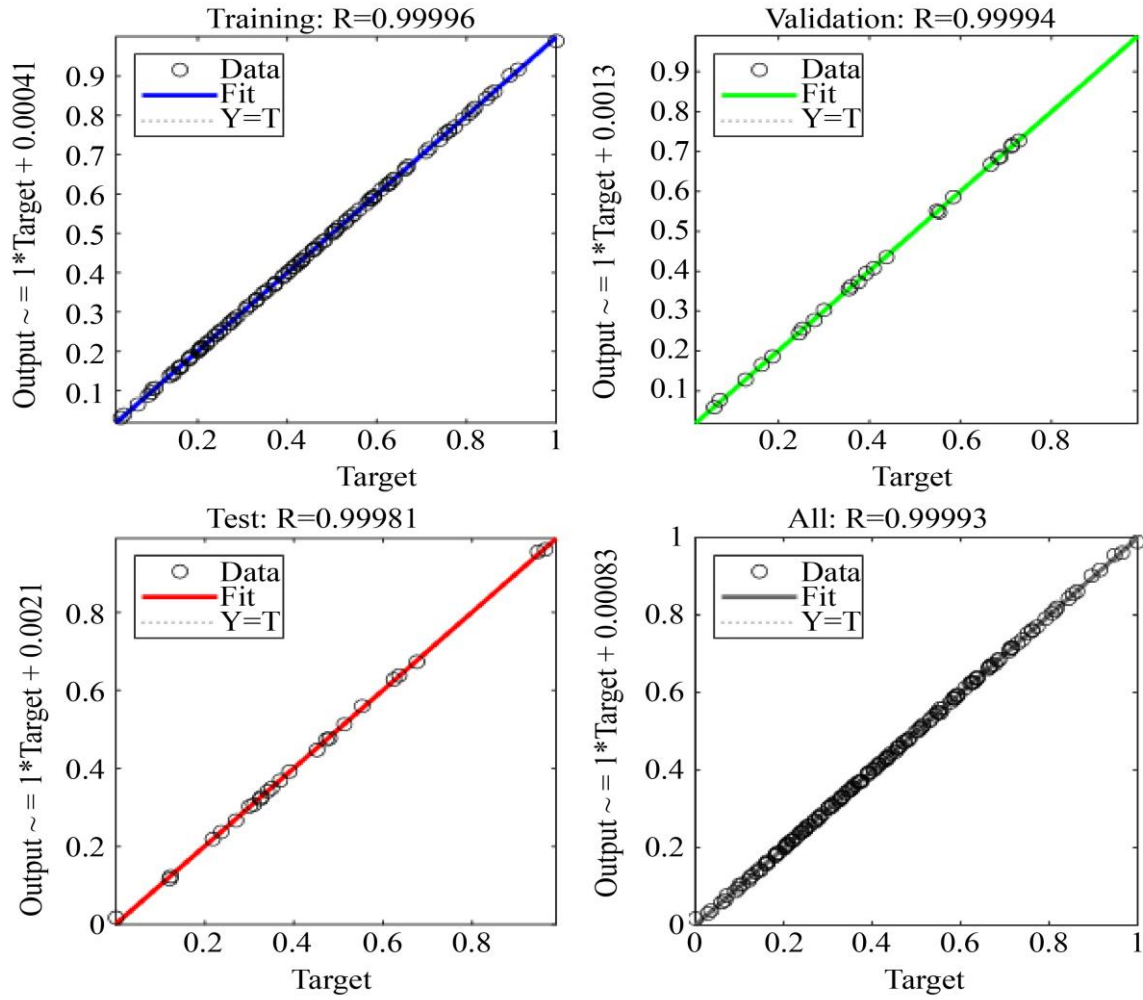


Fig. 14 Neural Network regression performance for topology 1

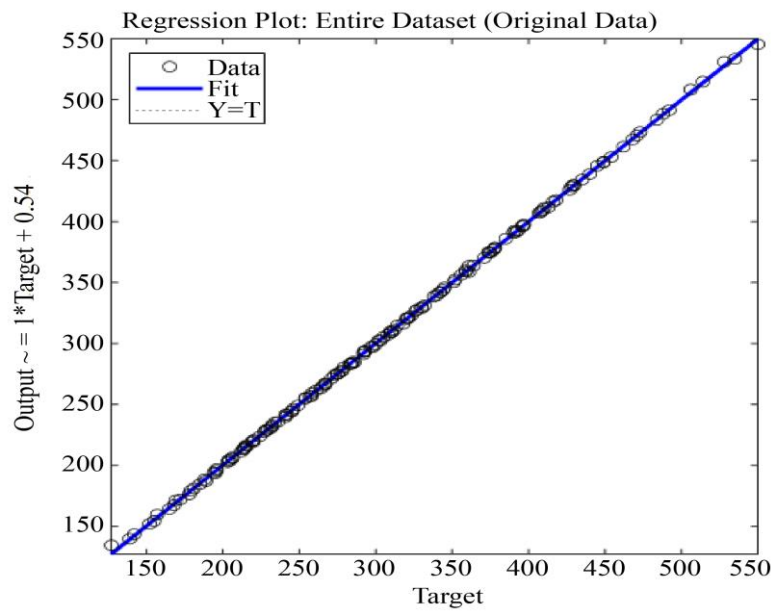


Fig. 15 NN regression plot for entire dataset (original data) for topology 1

Overall, the evidence supports using a single hidden layer with modest width and a nonlinear hidden transfer (e.g., TANSIG) paired with a linear output, trained via Levenberg–Marquardt when dataset size permits. Generalization appears driven more by appropriate capacity and optimizer choice than by additional depth.

#### Support Vector Regression

A comparative evaluation was run across four Support Vector Regression (SVR) kernels: linear, quadratic

(polynomial-2), cubic (polynomial-3), and RBF for the same forecasting tasks. Two input normalizations were considered: Min–Max scaling and Z-score standardization. Performance was measured on a held-out validation split using Mean Squared Error (MSE) and the Coefficient of Determination ( $R^2$ ).

Hyperparameter Tuning: Please refer to Table 3 and Figure 11 above. The parameters of the three kernels were optimally tuned by a loop of iterative MATLAB scripts.

**Table 9. Performance comparison of SVR topologies using the best hyperparameter and min-max normalization**

| Forecasting Type | Topology   | Kernel Type | BoxConstraint | KernelScale | Epsilon   | MSE (Min-Max) | R2 (Min-Max) |
|------------------|------------|-------------|---------------|-------------|-----------|---------------|--------------|
| SVR              | Topology 1 | Linear      | 42.113        | 0.5433      | 0.0003037 | 0.00363       | 0.93388      |
| SVR              | Topology 2 | RBF         | 771.36        | 4.4101      | 0.0006255 | 1.32E-06      | 0.99997      |
| SVR              | Topology 3 | Quadratic   | 755.52        | 5.94        | 0.0003160 | 0.00008       | 0.99853      |
| SVR              | Topology 4 | Cubic       | 672.22        | 70.554      | 0.0020829 | 0.00096       | 0.98132      |

The SVR model with a nonlinear kernel (Topology 2) is actually surprisingly robust to Min-Max scaling. Moreover, clearly, the RBF kernel is the way to go here. The performance is: MSE:  $1.32 \times 10^{-6}$   $R^2$ : 0.99997. You cannot really get closer to a perfect fit than that. The BoxConstraint is very large at

771.36, and the KernelScale is not too low at 4.41, suggesting the model is not being constrained too tightly by the large value, and the smaller value is large enough not to lose the fit to the fine structure of the nonlinearity that it has identified.

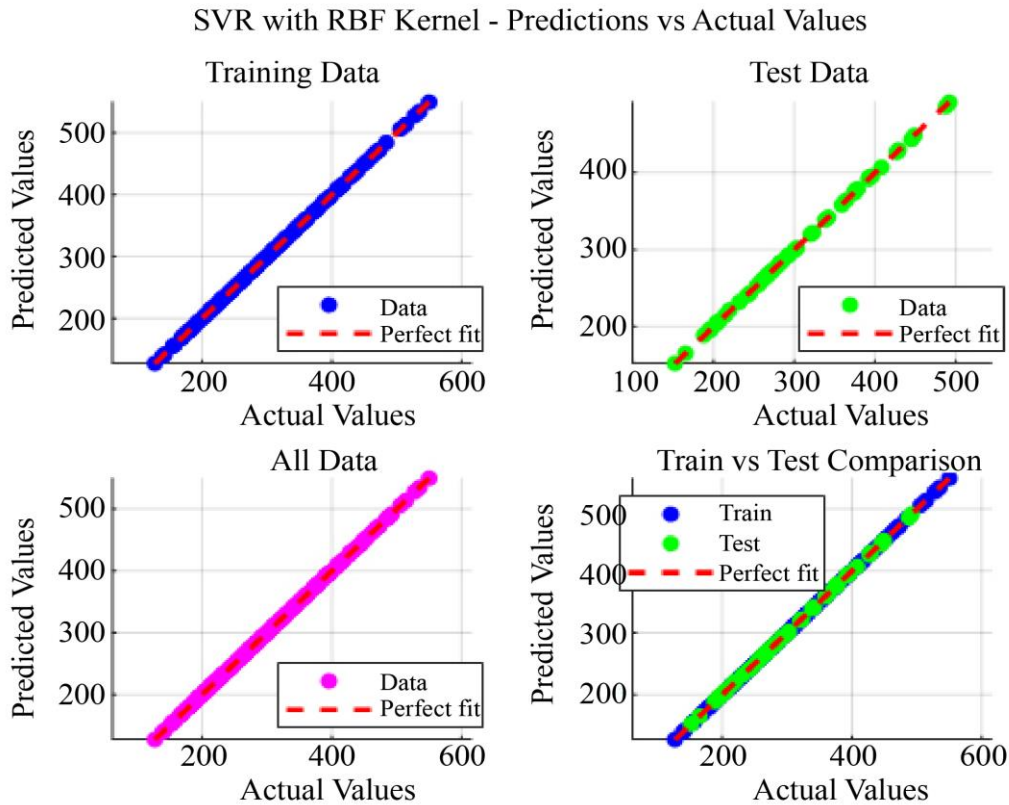


**Fig. 16 NN SVR with RBF kernel: Predictions vs actual values for training, test, and all data**

The quadratic kernel (Topology 3) ranked second, and achieved excellent but slightly inferior performance ( $MSE = 0.00008$ ,  $R^2 = 0.99853$ ). The performance decreased further for the cubic kernel ( $MSE = 0.00096$ ,  $R^2 = 0.98132$ ), indicating a combination of reasonable accuracy and over-complexity or less optimal fitting to the data structure under the applied normalization.

On the other hand, the linear kernel performed worst ( $MSE = 0.00363$ ,  $R^2 = 0.93388$ ), which indicates the existence of a non-linear relationship between inputs and target using Min–Max scaling. Hence, overall, Min–Max normalization preferred the most flexible option of the RBF kernel, and it

provided the best generalization performance for the scaled feature space.

After running the normalization, an Inverse Standardization was performed to run the performance metrics on the original scale of the data. For the best Topology 2 (RBF kernel), the MSE (Original Data) results were:  $MSE_{Training} = 0.06727$ ,  $RMSE_{Training} = 0.25936$ ;  $MSE_{Test} = 0.63164$ ,  $RMSE_{Test} = 0.79476$ ;  $MSE_{All} = 0.23622$ ,  $RMSE_{All} = 0.48603$ . The results are quite accurate for practical purposes, noting that there is very little error for the training set, and there is only a bit more variability with the test set.

**Table 10. Performance comparison of SVR topologies using the best hyperparameter and Z-Score normalization**

| Forecasting Type | Topology   | Kernel Type | BoxConstraint | KernelScale | Epsilon   | MSE (Z-Score) | $R^2$ (Z-Score) |
|------------------|------------|-------------|---------------|-------------|-----------|---------------|-----------------|
| SVR              | Topology 1 | Linear      | 118.93        | 8.6845      | 0.0012374 | 0.0695        | 0.93224         |
| SVR              | Topology 2 | RBF         | 929.85        | 6.0662      | 0.026073  | 0.00114       | 0.99876         |
| SVR              | Topology 3 | Quadratic   | 626.31        | 2.6674      | 0.014506  | 0.00599       | 0.99314         |
| SVR              | Topology 4 | Cubic       | 0.17788       | 0.47238     | 0.002887  | 0.00008       | 0.99992         |

After Z-score normalization, the ranking in performance changes greatly. The cubic kernel fits the best.  $MSE = 0.00008$ ,  $R^2 = 0.99992$  (Topology 4). Thus, a polynomial regression to the third power correlates very well with centered or standardized (by variance) data.

The results show that, although the RBF kernel is still the best performing kernel ( $MSE = 0.00114$ ,  $R^2 = 0.99876$ ), it dominates less than it did in the Min–Max scaling case. The optimised parameters for the RBF kernel are close to those found for Min–Max scaling. The BoxConstraint is decreased to 929.85, and the KernelScale is increased to 6.07, indicating a smoother decision boundary than in the Min–Max-based case.

Running the same test with a quadratic kernel still only yielded second-best results ( $MSE = 0.00599$ ,  $R^2 = 0.99314$ ), while the linear kernel, as before, was nearly useless ( $MSE = 0.0695$ ,  $R^2 = 0.93224$ ). So, linear SVR still is not expressive enough to compete in either of the preprocessing scenarios. As a result, due to standard variance compatibility and symmetric feature distribution, Z-score normalization steers the optimal performance toward polynomial modeling (in particular, cubic).

In conclusion, the comparison demonstrates that SVR is much more sensitive to the selected normalization. Min–Max scaling shows a clear preference for the RBF kernel, as it achieves the highest accuracy by exploiting powerful local nonlinear flexibility. On the other hand, the Z-score standardization moves the best performance to the cubic kernel, since the polynomial structure could work better when the variance lies in a feature-space-centered manner. In this

square, the linear SVR still does not perform very well, which indicates that the forecasting scheduling problem is not linear. Hence, the pre-processing and kernel selection should not be considered independently since RBF is preferred with Min–Max normalization, while cubic SVR is more effective when Z-score normalized, which underlines that scaling of the data has a significant effect on the overall model performance.

#### 4.8. ScheduleRelia: Technical Architecture and Validation

In order to bridge the huge gap between results achieved by machine learning for the prediction of construction projects and their practical application by project governance in construction, in this study, a reliability-aware decision-support tool called ScheduleRelia for construction project scheduling is introduced.

##### 4.8.1. ScheduleRelia System Architecture

ScheduleRelia is a tool written in modular code to allow for easy modification and the use of existing project management software. ScheduleRelia has a front-end that reads in the standard outputs of scheduling plans from existing project management software.

The back-end is a prediction engine that uses the trained SVR model with an RBF kernel for its regressions, which was found to be the most stable and with the highest accuracy of  $R^2 \approx 0.99997$  in the testing of this study. It must be noted that ScheduleRelia is currently presented as a theoretical proof-of-concept framework. While it demonstrates high fidelity on the simulated dataset, independent external validation using real-world, uncontrolled project data is required before it can be deployed as a fully reliable, practical decision-support system in the field.

#### 4.8.2. Metric Computation and Weighting Approach

The six key planning measures are automatically extracted from a project plan by ScheduleRelia. These are: Total Activities; Sum of Total Float; Total Count of Relationships; Sum of Lag; Average Original Duration; and Maximum Original Duration.

All of these six measures are then Min-Max normalized (on a project-by-project basis, using the Min-Max scaling technique established during training with the SVR-RBF, Relia engine) prior to making a prediction.

Schedule health: a new risk metric. A new “schedule health” risk metric is defined and assessed using a weighted sensitivity-based approach. The key variables were assessed for their impact on schedule performance in the prior study.

Thus, variables that cause greater variation in output (e.g., excessive Sum of Lag or Maximum Original Duration) will receive greater penalization weight. Thus, if the ratio of total float to predicted project duration drops below a historically “safe” value for a given number of activities in prior projects, then a proactive “schedule health” warning will be issued to the user.

#### 4.8.3. Worked Demonstration

In the following section, the tool is applied to a baseline case study for a school construction project, which was already included in the dataset.

The input variables and the respective values for the ScheduleRelia tool are listed as follows: 51 activities in total, 765 days of total float, 67 relationships between the activities, 0 days of lag, average duration 8.98 days, and maximum original duration 22 days.

- Ingestion & Scaling: ScheduleRelia normalizes these values strictly between 0 and 1 using the pre-established Min-Max training parameters.
- Prediction: The data is passed through the SVR-RBF prediction engine.
- Output & Diagnostics: Undo standardization to arrive at the final Inverse standardized value. The calculated Baseline Project Duration for the provided data set equals 196 days. The same data set had the following properties regarding the risk for critical path delay, given by ‘Sum of Lag’ = 0, ‘Sum of Total Float’ = 765, which is proportional to the number of activities (51 in this case): ScheduleRelia flags this schedule health as ‘Optimal / Low Risk’. Note that a schedule with the very same 51 activities would immediately receive a risk for critical path delay by a manager, who tries to force his schedule by artificially setting more concurrent tasks, i.e. by setting ‘Sum of Lag’ to 30 while at the same time reducing average duration. In such a case, the tool would give a new duration and indicate the risk for the manager.

## 5. Discussions

### 5.1. Comparison between NN and SVR Topologies by Normalization

#### 5.1.1. Comparative Performance of Neural Networks and SVR under Min-Max Normalization

According to the Min-Max normalized results in Tables 4 and 5, both models have a very high prediction accuracy, but SVR outperforms Neural Networks slightly in terms of power and stability.

According to the NN Min-Max Table 4, the optimal configuration is NN Topology 1 (1 hidden layer, 10 neurons, TANSIG for hyperbolic tangent sigmoid and trainlm for Levenberg-Marquardt backpropagation algorithm), which yields an MSE of  $6.95 \times 10^{-6}$  and  $R^2$  of 0.99993.

If this does demonstrate a great predictive power, the other NN topologies result in significant performance degradation, especially for more complex architectures (like Topologies 3 and 6), showing the sensitivity to architecture complexity and training set.

Based on the SVR Min-Max Table 5, the SVR Topology 2 with RBF kernel performs the best overall performance scoring at  $MSE = 1.32 \times 10^{-6}$  and  $R^2 = 0.99997$ , which makes it the most accurate model from all configurations tested. This works better than the top NN plot and yields somewhat more stable behavior over kernel variation than the range observed across NN architectures.

**Table 11. Comparison between NN and SVR using min-max normalization**

| Forecasting Type      | MSE (Min-Max) | $R^2$ (Min-Max) |
|-----------------------|---------------|-----------------|
| NN (10 neurons/Layer) | 6.9472E-06    | 0.99993         |
| SVR (RBF Kernel)      | 1.3202E-06    | 0.99997         |

In general, the observed results suggest that the SVR is more robust and stable (and thus more predictive accurate) to build regression models using Min-Max normalization than NN models when dealing with this dataset.

The best prospective Neural Network (Topology 1) reduces the MSE without lower than approximately equal to  $6.95 \times 10^{-6}$  but achieves less in terms of  $R^2$  at 0.99993, while the SVR model using RBF kernel attains a correspondingly lower MSE at about  $1.32 \times 10^{-6}$  and a higher  $R^2$  at about 0.99997, respectively.

This proves that even though the better performance of Neural Networks when optimally-tuned, SVR is more consistent and accurate, especially when quantisation using Min-Max scaling is adopted, further supporting it as the best predictive model for this investigation.

### 5.1.2. Comparative Performance of Neural Networks and SVR under Z-Score Normalization

According to the Z-Score Normalization results given in Tables 4 and 6, SVR performs much better than any of the NN models. The NN model, topology 1, performed the best with 0.01039 MSE and  $R^2$  0.99494. These results indicate that it gave good predictions. On the other hand, the remaining NN topologies had greater fluctuations, and it was noted that MSE and  $R^2$  markedly declined, showing the NN dependency on these two components.

On the other hand, SVR models reflect better and superior performance in the kernel. As per the same table, the optimum SVR configuration is Topology 4, which uses the Cubic kernel. It records the lowest MSE of 0.00008 and the highest  $R^2$  of 0.99992. This is extremely high precision and a very strong agreement between predictions and observations.

**Table 12. Comparison between NN and SVR using Z-Score normalization**

| Forecasting Type      | MSE (Z-Score) | $R^2$ (Z-Score) |
|-----------------------|---------------|-----------------|
| NN (10 neurons/Layer) | 0.01039       | 0.99494         |
| SVR (Cubic Kernel)    | 0.00008       | 0.99992         |

Based on the tabled results, it is demonstrated that SVR, along with Z-Score normalization, leads to more robustness, stability and prediction accuracy than NN, making it a more reliable choice for this dataset. The best Neural Network approach, Topology 1, yields an MSE of about 0.01039 and  $R^2 = 0.99494$ ; also, Cubic Kernel SVR matrices outperform greatly with MSE about 0.00008 and  $R^2 = 0.99992$ . This clearly argues that, although with a good predictive capacity of the optimum structure Neural Network performs, still SVR is better in largely consistent and accurate prediction under the Z-Score normalization method, proving it as an effective and reliable forecasting technique for this study.

## 5.2. Comparison between NN and SVR Topologies by the Best Performance, Sensitivity Analysis and Cross-validation

### 5.2.1. Best Performance

After analyzing all tested topologies, a comparative evaluation was carried out to assess the influence of normalization methods on model performance, based on the Mean Squared Error (MSE) criterion. Table 9 summarizes the best results obtained for each model under Min-Max and Z-Score normalization.

**Table 13. Best performance comparison based on MSE**

| Forecasting Type | MSE (Min-Max) | MSE (Z-Score) |
|------------------|---------------|---------------|
| NN               | 6.9472E-06    | 0.01039       |
| SVR              | 1.3202E-06    | 0.00008       |

For neural networks, Min-Max normalization performs better than Z-Score normalization. It yields a much lower MSE ( $6.9472 \times 10^{-6}$  against 0.01039), which also facilitates

learning smoother and faster. SVR follows the same trend as Min-Max normalization performs better than Z-Score normalization (MSE =  $1.3202 \times 10^{-6}$  and MSE = 0.00008, respectively).

Error levels and accuracy are similar for both models using the two pre-processing techniques. However, Min-Max scaling produces less error and is more accurate. Overall, Min-Max normalization is better for scheduling forecasting since it increases the stability and predictability of the model.

### 5.2.2. Sensitivity Analysis

Table 10 below presents the response of the Neural Network (NN) and Support Vector Regression (SVR) models (best-performing topologies under Min-Max normalization) to incremental percentage changes in the input variables.

This analysis evaluates how variations in input data influence the corresponding changes in model outputs, thereby reflecting the relative sensitivity and robustness of each approach.

**Table 14. Sensitivity analysis of NN and SVR**

| Percentage change in input | Output change for NN | Output change for SVR |
|----------------------------|----------------------|-----------------------|
| 1%                         | 0.31837%             | 2.74515%              |
| 3%                         | 0.74537%             | 2.46808%              |
| 5%                         | 0.36122%             | 2.34344%              |
| 7%                         | 1.48392%             | 1.78365%              |
| 9%                         | 0.83602%             | 2.76284%              |

The Neural Network can show stability when the input perturbation factor is applied to the basic test functions in the range between 1%-5 %, the output varies by about the 1% methodology.

On the other hand, SVR shows greater output variation for the same input changes, indicating a higher sensitivity to input variation at a low level.

With a 7% and 9% variation in input, the NN outputs vary slightly but remain controlled and proportional. However, the output of SVR is higher and fluctuates irregularly because its output may change by more than 2.7% for 9% of input variation. This implies that SVR is more responsive to input data changes than NN, which is relatively smooth and stable in response.

As a whole, the Neural Network model shows higher stability under input perturbations. The NN produces a controlled and consistent output that displays a high level of reliability against minor data variations.

### 5.2.3. Cross-Validation and Generalization Performance Analysis

Following the application of both modeling approaches, the results of the 5-fold cross-validation procedure are summarized in Tables 11 and 12 below.

These tables report the Mean Squared Error (MSE) obtained for each fold, as well as the aggregated training, testing, and cross-validation errors for the Neural Network (NN) and the Support Vector Regression (SVR) models, thereby providing a robust assessment of their generalization capability and stability.

Interestingly enough, performance consistency between the two models fluctuates greatly. For 5 of the folds, the Network's MSE was higher than that of the previous models. MSE values were around 0.00234 to 0.00739.

Because of this fluctuation, it is expected that the models generated based on data partitions with different characteristics will differ significantly. The validation error for the SVR model is much lower than the Network, varying between 0.00010 and 0.00147 for all folds.

**Table 15. Fold-wise MSE comparison between NN and SVR**

| Forecasting type | Fold's number | MSE Validation |
|------------------|---------------|----------------|
| NN               | 1             | 0.00714        |
| NN               | 2             | 0.00237        |
| NN               | 3             | 0.00351        |
| NN               | 4             | 0.00739        |
| NN               | 5             | 0.00234        |
| SVR              | 1             | 0.00083        |
| SVR              | 2             | 0.00147        |
| SVR              | 3             | 0.00127        |
| SVR              | 4             | 0.00010        |
| SVR              | 5             | 0.00044        |

**Table 16. Global performance metrics for NN and SVR**

| Forecasting type      | MSE training | MSE Test  | Cross-Validation for 5 folds |
|-----------------------|--------------|-----------|------------------------------|
| NN (10 neurons/layer) | 1.196E-04    | 3.240E-04 | 4.550E-03                    |
| SVR (RBF)             | 3.288E-07    | 7.997E-05 | 8.220E-04                    |

This observation is further corroborated by the aggregated performance metrics. Table 12 indicates that the Neural Network achieves an MSE of  $1.20 \times 10^{-4}$  in training,  $3.24 \times 10^{-4}$  in testing, and 0.00455 in cross-validation.

Despite these results indicating a satisfactory predictive performance, the disparity between training and validation errors suggests a limited generalization efficiency.

On the other hand, the SVR model (RBF kernel) has a low error rate because it has an MSE of  $3.29 \times 10^{-7}$  during training,  $7.99 \times 10^{-5}$  during testing, and only 0.000822 during 5-fold cross-validation.

Hence, the SVR model (RBF kernel) demonstrates superior generalization robustness and predictive accuracy when exposed to new, unseen data.

### 5.3. Contribution

The research adds significant value to the field of construction project management through the comparison of using Artificial Neural Networks (ANN) and Support Vector Regression (SVR) for estimating project duration.

The research used a simulated data set of 157 school building projects in order to generate a realistic yet controlled data set to enable a number of variables that are used in the planning of the projects to be isolated and benchmarked individually.

The study is also the first to systematically investigate and quantify the combinatorial performance and interdependence between different data normalization methods (e.g., Min-Max scaling vs. Z-Score scaling) and the corresponding algorithmic settings (e.g., SVR with RBF kernel).

Importantly, the study's findings from the empirical-based selection processes provide strong evidence that Min-Max scaling optimizes SVR with an RBF kernel for the best possible performance, whereas Z-Score scaling maximally optimizes models that follow a cubic polynomial structure.

This study also translates its findings into practice by developing a functional framework called ScheduleRelia, which forecasts the project duration with high accuracy, while at the same time actively checks the schedule's quality by measuring the critical path metrics, thus allowing for timely replanning of the schedule as well as for making of risk-informed and wise decisions.

### 5.4. Comparison with Other Studies

Table 13 below presents a comparative analysis between the current study and other studies.

**Table 17. Comparative studies**

| Study                              | Application Domain                         | Forecasting Type                                                    | Normalization Type             | Data to Predict (Output)       | Performance (MSE / R <sup>2</sup> )                                                                                                                                                                                          | Findings                                                                                                                                                                                                              |
|------------------------------------|--------------------------------------------|---------------------------------------------------------------------|--------------------------------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Current study                      | Construction Project Management            | Artificial Neural Network (ANN) and Support Vector Regression (SVR) | Min-Max and Z-Score            | Project Duration               | NN/Min-Max: MSE: 6.95E-06 / R <sup>2</sup> : 0.99993<br>NN/Z-Score: MSE: 0.01039 / R <sup>2</sup> : 0.99494<br>SVR/Min-Max: MSE: 1.32E-06 / R <sup>2</sup> : 0.99997<br>SVR/Z-Score: MSE: 0.00008 / R <sup>2</sup> : 0.99992 | Compact ANN is sensitive to Z-Score but works well with Min-Max. SVR with Cubic leads under Z-Score, emphasizing pre-processing and kernel influence, whereas SVR with RBF performs exceptionally well under Min-Max. |
| Kamil (2025) [37]                  | Urban flexible pavement (Roads)            | Artificial Neural Network (ANN)                                     | Min-Max scaling                | Delay risk / Time overrun      | R <sup>2</sup> = 0.847 / MAPE = 7.10%                                                                                                                                                                                        | A closed-form ANN model using Min-Max scaling provides a mathematically validated and robust framework for forecasting time overruns in urban road networks.                                                          |
| Park et al. (2025) [38]            | High-rise buildings (Curtain walls)        | SVR, DNN, and Random Forest                                         | Z-score (Standardization)      | Construction duration (Days)   | R <sup>2</sup> = 0.970 / RMSE = 74 days / +4.23% (Correlation increase)                                                                                                                                                      | Combining Z-score with data augmentation (GANs) significantly enhances SVR and DNN generalization and accuracy on limited construction datasets.                                                                      |
| El Abidi Oumaima et al., 2025 [18] | Pavement Performance Prediction in Morocco | Artificial Neural Network (ANN)                                     | Min-Max (normalized to [0, 1]) | Pavement Condition Index (PCI) | RMSE = 4.3691x10 <sup>-3</sup> , R <sup>2</sup> = 0.95153 (overall)                                                                                                                                                          | ANN model shows low error levels and high accuracy, suitable for Moroccan PMS development                                                                                                                             |
| Velumani et al. (2021) [39]        | Highway road projects                      | ANN and Multiple Linear Regression                                  | Min-Max Normalization          | Construction duration          | RMSE: 0.128 / R <sup>2</sup> : 0.979 / MAPE: 1.2% (best model)                                                                                                                                                               | The ANN and smoothing architectures showed superior predictive capabilities compared to Bromilow's time-cost model and standard regression for highway durations.                                                     |

### 5.5. Limitations of this Work

While the comparative analysis yields highly accurate and promising results for project duration forecasting using ANN and SVR, this study is subject to several limitations. Acknowledging these limitations is essential for contextualizing the findings and defining the trajectory for

future applied research in construction management. The primary constraints of the current modeling approach, their impacts, and the corresponding mitigation pathways are summarized in Table 14 and conceptually visualized in Figure 17.

## Limitations of a Predictive Machine Learning Model in Construction

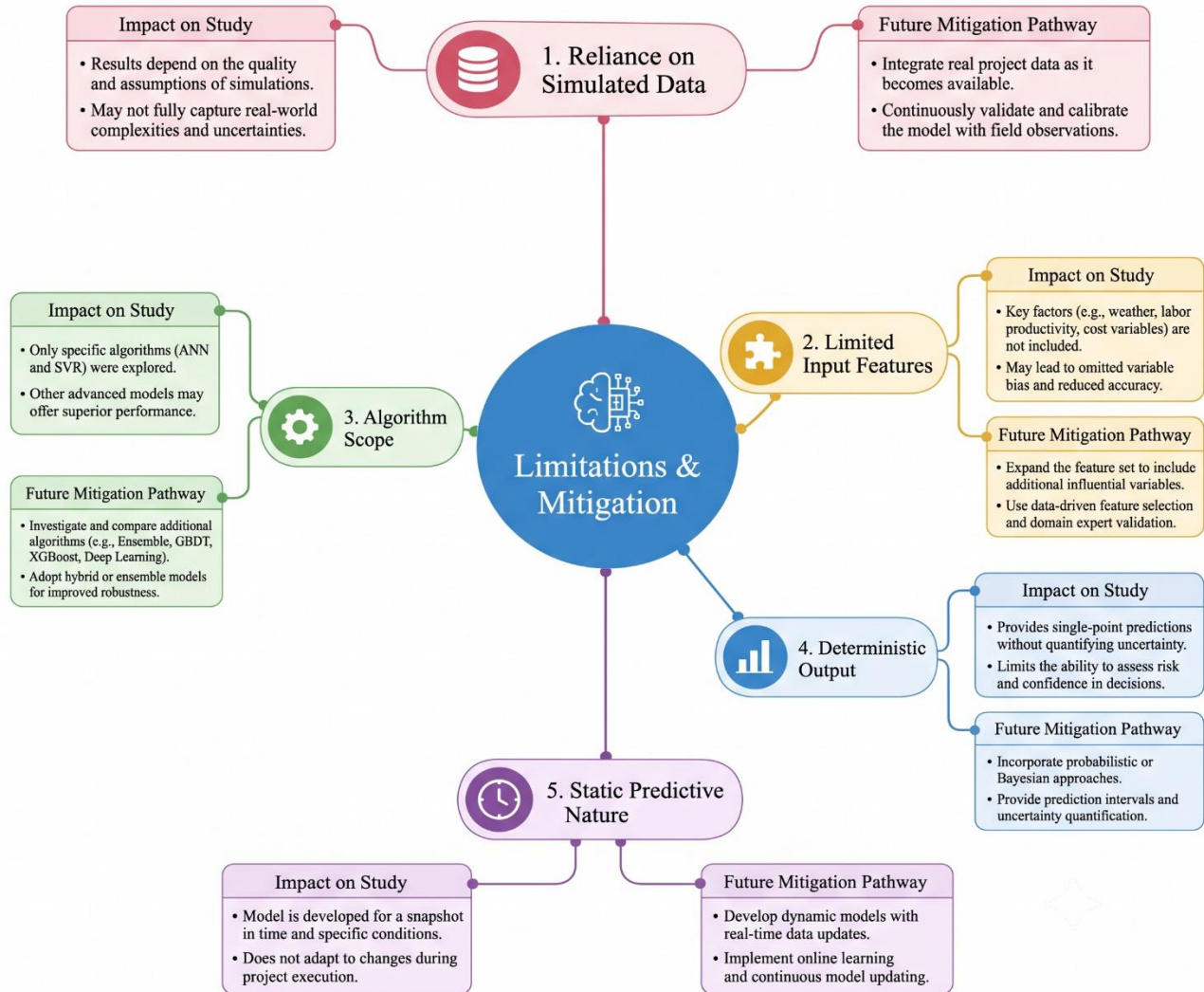


Fig. 17 Limitations of a predictive machine learning model in construction

Table 18. Summary of the primary limitations of the current study and proposed future research directions

| Limitation Category        | Description of the Constraint                                                                                                       | Impact on the Study                                                                                                                                               | Pathway for Future Mitigation                                                                                      |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Reliance on Simulated Data | The models are trained and tested on 157 highly controlled, simulated scheduling scenarios rather than noisy historical field data. | Restricts the immediate generalization of the findings to highly volatile, real-world construction environments affected by unforeseen site-specific disruptions. | Validation of the proposed topologies using real-world multi-project datasets from diverse geographical locations. |

|                           |                                                                                                                       |                                                                                                                                                                                        |                                                                                                                                     |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Limited Input Feature Set | The prediction relies strictly on six foundational scheduling variables (e.g., float, lag, duration, activities).     | Fails to capture other critical real-world factors influencing project duration, such as weather severity, labor productivity variations, cost dynamics, and specific site conditions. | Incorporation of a broader, heterogeneous set of features, integrating BIM data, economic indicators, and environmental conditions. |
| Algorithm Scope           | The comparative analysis is strictly limited to Artificial Neural Networks (ANN) and Support Vector Regression (SVR). | The selected models, while effective, may not represent the absolute best available predictive performance compared to modern ensemble methods.                                        | Evaluation of more advanced hybrid models or ensemble learning techniques (e.g., XGBoost, Random Forest, deep LSTM networks).       |
| Deterministic Output      | The current ANN and SVR models deliver point predictions without confidence intervals.                                | Does not quantify the uncertainty of the prediction, limiting its utility for rigorous risk management and probabilistic stakeholder decision-making.                                  | Integration of uncertainty quantification mechanisms, such as Lower and Upper Bound Estimation (LUBE) or Monte Carlo simulations.   |
| Static Predictive Nature  | The framework forecasts the final duration based on a static baseline input without continuous recalibration.         | The system cannot adapt to the dynamic, evolving nature of a construction site where daily progress constantly shifts the critical path.                                               | Development of dynamic updating mechanisms fueled by real-time data ingestion from site IoT sensors and daily progress tracking.    |

### 5.6. Perspectives

The paper presents a few future trends that could be relevant for the development of AI-based prediction systems for the Construction industry. Specifically, it is suggested that future hybrid models based on ANN and SVR are integrated with metaheuristics such as PSO and ensemble methods, aiming at improving the accuracy and robustness of the predictions for non-stationary or even heterogeneous data. Furthermore, uncertainty-aware improvements such as interval forecasting and drift tracking should be explored in order to ensure the usefulness of the models for projects that evolve over time. Utilizing BIM for a consistent information platform will ensure that the model remains accurate over time. Further studies can explore if these models scale well with larger data and various types of projects. Additionally, one must solve interpretability problems before they are used for mission-critical control processes. For operations, the target regions of deployment-oriented AI technologies may include making retention time corrective recommendations and providing an interpretable output.

Although SVR and ANN are successful for predicting duration according to the results, the research is based on a static modelling method. The predictions are performed according to a set of planning parameters at baseline. However, construction processes are complex and dynamic systems that evolve with time and are subject to stochastic interruptions. Thus, future research should attempt to shift from static forecasting to active dynamic schedule management.

Their task would be accomplished through the following instructions:

- Further studies should incorporate Deep Learning models such as Long Short-Term Memory networks or Gated Recurrent Units. These models are specifically intended to capture the temporal dependencies and sequential mechanisms embedded in time-series data, which may be ignored by Feed-Forward Models employed in this analysis.
- Real-Time Monitoring: Developing a real-time monitoring architecture where it is possible to update the prediction model with real-time information from the site (e.g., IoT sensors or daily progress reports). Through this technique, it becomes possible to track the drift and recalibrate predictions as shifts in the project or setting occur.
- The construction of interval-valued forecasts is very useful for risk management purposes. Future models need to present risk-adjusted error bars rather than point estimates to be able to make more rigorous decisions under uncertainty.

## 6. Conclusion

Forecasting project duration is a critical aspect of construction project planning and management. For this purpose, it is desirable to have good predictions as well as an assessment of their reliability. Using a set of relatively small project planning datasets, this study reports on the application of Support Vector Regression (SVR), specifically with both Radial Basis Function (RBF) kernels and Min-Max scaling

versus cubic kernels and Z-score scaling. Results reveal exceptional accuracy and cross-validation stability, as well as performance that approaches state-of-the-art for well-tuned compact Artificial Neural Networks (ANNs) with error at the level of one day for training/validation for a single project. The choice of pre-processing methodology and kernel/architecture for SVR must thus be considered as a single coupled decision. Applying these techniques, a neural network-centric application, supported by an SVR layer for robustness, has been developed to estimate duration from scope and baseline planning data, as well as perform a rigorous quality check against the planned schedule using the critical-path, levels of float, lag, relationship density and nominal or standardized durations that reflect current productivity, WBS depth, technology and risk. The result is a

more reliable schedule realism and improved schedule governance, enabling better predictive ability and more effective replanning. It is recommended to test the ANN and SVR approaches in an empirical and practical way, based on normalisation of data and a proper evaluation of the results ( $R^2$ , MAE, RMSE, MAPE) based on cross-validation techniques. Feature attributions and uncertainty intervals should also be evaluated. It is also important to monitor time drifts and to optimise the model to correct its possible errors to ensure correct results. With the data flows generated from BIM and proper model management, it is possible to move from a scheduling approach based on experience and skills to a data-driven planning approach, which is clear, explainable and rational, and that provides risk assessment and ensures accuracy along the whole lifecycle of construction projects.

## References

- [1] Yasser Elfahham, "Estimation and Prediction of Construction Cost Index Using Neural Networks, Time Series, and Regression," *Alexandria Engineering Journal*, vol. 58, no. 2, pp. 499-506, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Fan Zhang, and Lauren J. O'Donnell, *Support Vector Regression*, Machine Learning, pp. 123-140, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Xiaojing Ding, and Qiulan Lu, "Construction Cost Management Strategy based on BIM Technology and Neural Network Model," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 40, no. 4, pp. 6669-6681, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] Ying Hong, Hanyan Xie, and Ioannis Brilakis, "Automate Construction Scheduling at the Pre-construction Stage – A Case Study," *Apollo - University of Cambridge*, pp. 1-10, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Camilo I. Lagos et al., "Predicting Construction Schedule Performance with Last Planner System and Machine Learning," *Automation in Construction*, vol. 167, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Xiaobing Yu, and Hengzhong Zuo, "Intelligent Construction Optimization Control of Construction Project Schedule Based on the Fuzzy Logic Neural Network Algorithm," *Mathematical Problems in Engineering*, vol. 2022, no. 1, pp. 1-11, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Viren B. Chandanshive, and Ajaykumar R. Kambekar, "Prediction of Building Construction Project Cost Using Support Vector Machine," *Industrial Engineering and Strategic Management*, vol. 1, no. 1, pp. 31-42, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Clyde L. Tamayo, and Jerome Jordan F. Famadico, "Predictive Models of Construction Project Success Rating Using Regression and Artificial Neural Network," *International Journal of Multidisciplinary: Applied Business and Education Research*, vol. 5, no. 7, pp. 2733-2745, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] Shen Zhang, and Xuechun Li, "A Comparative Study of Machine Learning Regression Models for Predicting Construction Duration," *Journal of Asian Architecture and Building Engineering*, vol. 23, no. 6, pp. 1980-1996, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Mathieu Wauters, and Mario Vanhoucke, "Support Vector Machine Regression for Project Control Forecasting," *Automation in Construction*, vol. 47, pp. 92-106, 2014. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] Fangfang Yu et al., "An Active Construction Dynamic Schedule Management Model: Using the Fuzzy Earned Value Management and BP Neural Network," *KSCE Journal of Civil Engineering*, vol. 25, no. 7, pp. 2335-2349, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Ayaz Ahmad et al., "Compressive Strength Prediction via Gene Expression Programming (GEP) and Artificial Neural Network (ANN) for Concrete Containing RCA," *Buildings*, vol. 11, no. 8, pp. 1-18, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] Shihong Huang, Chengye Liang, and Jiao Liu, "Research on Neural Network Prediction Model of Whole Process Risk Management Based on Building Information Model," *Advances in Civil Engineering*, vol. 2024, no. 1, pp. 1-18, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Min-Yuan Cheng, Quoc-Tuan Vu, and Frederik Elly Gosal, "Hybrid Deep Learning Model for Accurate Cost and Schedule Estimation in Construction Projects Using Sequential and Non-Sequential Data," *Automation in Construction*, vol. 170, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Mostafa Mir et al., "Neural Network-based Interval Forecasting of Construction Material Prices," *Journal of Building Engineering*, vol. 39, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [16] Erik Matel et al., "An Artificial Neural Network Approach for Cost Estimation of Engineering Services," *International Journal of Construction Management*, vol. 22, no. 7, pp. 1274-1287, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Fatma Yaprakdal et al., "A Deep Neural Network-Assisted Approach to Enhance Short-Term Optimal Operational Scheduling of a Microgrid," *Sustainability*, vol. 12, no. 4, pp. 1-27, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] El Abidi Oumaima et al., "Performance Prediction of Moroccan Pavements Using Artificial Neural Networks," *International Journal of Engineering Trends and Technology*, vol. 73, no. 4, pp. 70-80, 2025. [[CrossRef](#)] [[Publisher Link](#)]
- [19] Putu Manik Sesa Arianti, Miftakhul Jannah, and Noviyanti Santoso, "A Metaheuristic-Based Approach to Inflation Prediction in Indonesia with Support Vector Regression (SVR)," *Journal of Computer Engineering, Electronics and Information Technology*, vol. 4, no. 1, pp. 1-10, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [20] Kennedy Were et al., "A Comparative Assessment of Support Vector Regression, Artificial Neural Networks, and Random Forests for Predicting and Mapping Soil Organic Carbon Stocks Across an Afromontane Landscape," *Ecological Indicators*, vol. 52, pp. 394-403, 2015. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Junrong Zhang et al., "Input-Parameter Optimization using a SVR Based Ensemble Model to Predict Landslide Displacements in a Reservoir Area – A Comparative Study," *Applied Soft Computing*, vol. 150, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Yuexin Wang, Chao Wen, and Ping Huang, "Predicting the Effectiveness of Supplement Time on Delay Recoveries: A Support Vector Regression Approach," *International Journal of Rail Transportation*, vol. 10, no. 3, pp. 375-392, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Wenhui Feng, and Yafeng Zou, "Construction Cost Prediction based on Adaptive Boosting and Artificial Neural Networks," *Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction*, vol. 178, no. 1, pp. 1-9, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [24] Gökhan Aksu, Cem Oktay Güzeller, and Mehmet Taha Eser, "The Effect of the Normalization Method Used in Different Sample Sizes on the Success of Artificial Neural Network Model," *International Journal of Assessment Tools in Education*, vol. 6, no. 2, pp. 170-192, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [25] Peshawa J. Muhammad Ali, "Investigating the Impact of Min-Max Data Normalization on the Regression Performance of K-Nearest Neighbor with Different Similarity Measurements," *ARO-The Scientific Journal of Koya University*, vol. 10, no. 1, pp. 85-91, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [26] ThanhNgoc Tran et al., "Load Forecasting with Support Vector Regression: Influence of Data Normalization on Grid Search Algorithm," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 4, pp. 3410-3420, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [27] Yasemin Onal, "Gaussian Kernel Based SVR Model for Short-Term Photovoltaic MPP Power Prediction," *Computer Systems Science and Engineering*, vol. 41, no. 1, pp. 141-156, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [28] Rahmad Ramadhan Laska, and Anne Mudya Yolanda, "A Comparative Study of Z-Score and Min-Max Normalization for Rainfall Classification in Pekanbaru," *Journal of Data Science*, vol. 2024, no. 4, pp. 1-8, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [29] Mohammed Z. Al-Faiz, Ali A. Ibrahim, and Sarmad M. Hadi, "The Effect of Z-Score Standardization (Normalization) on Binary Input Due the Speed of Learning in Back-Propagation Neural Network," *Iraqi Journal of Information & Communications Technology*, vol. 1, no. 3, pp. 42-48, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [30] Pei-Yu Wu et al., "Predictive Modeling and Estimation of Moisture Damages in Swedish Buildings: A Machine Learning Approach," *Sustainable Cities and Society*, vol. 118, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [31] Linlin Xie et al., "Scheduling Optimization of Prefabricated Construction Projects by Genetic Algorithm," *Applied Sciences*, vol. 11, no. 12, pp. 1-21, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [32] Xiaobing Yu, "A RBF Fuzzy Logic Neural Network Algorithm for Construction Resource Scheduling," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 41, no. 4, pp. 4937-4945, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [33] Arsalan Mahmoodzadeh et al., "Forecasting Tunnel Geology, Construction Time and Costs Using Machine Learning Methods," *Neural Computing and Applications*, vol. 33, pp. 321-348, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [34] Tiago Zonta et al., "A Predictive Maintenance Model for Optimizing Production Schedule Using Deep Neural Networks," *Journal of Manufacturing Systems*, vol. 62, pp. 450-462, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [35] Oleg A. Stepanov, and Oleg S. Amosov, "The Comparison of the Monte-Carlo Method and Neural Networks Algorithms in Nonlinear Estimation Problems," *IFAC Proceedings Volumes*, vol. 40, no. 13, pp. 392-397, 2007. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [36] Molusiwa Stephan Ramabodu, "Harnessing Regression Models to Optimize Project Duration Estimation and Reduce Delays in South African Infrastructure Development," *Preprint*, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [37] Ban Ali Kamil, "An Empirical Evaluation of the Performance of Deep Neural Networks on Delay Risk Prediction in Urban Flexible Pavement Projects in Iraq," *Engineering, Technology & Applied Science Research*, vol. 15, no. 5, pp. 28343-28349, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [38] Sang-Jun Park et al., “Data Augmentation Approaches for Estimating Curtain Wall Construction Duration in High-Rise Buildings,” *Buildings*, vol. 15, no. 4, pp. 1-21, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [39] P. Velumani, N. V. N. Nampoothiri, and M. Urbański, “A Comparative Study of Models for the Construction Duration Prediction in Highway Road Projects of India,” *Sustainability*, vol. 13, no. 8, pp. 1-13, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [40] Farras Nadhif Danedrata Herman, and Mochamad Gaharu Dida Devedo, “Integrated Risk Analysis and Delay Prediction in Construction Projects using XGBoost, Monte Carlo Simulation, and Rule-Based Reasoning,” *International Conference on Tropical Studies and Its Applications*, vol. 5, no. 1, pp. 1-10, 2026. [[Google Scholar](#)] [[Publisher Link](#)]