Automated machine learning pipeline for robust project cost and duration forecasting

Automation in Construction 178 (2025) 106426

Contents lists available at ScienceDirect

Automation in Construction

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

Automated machine learning pipeline for robust project cost and duration forecasting

Filippo Maria Ottaviani [a,*], Pablo Ballesteros-Pérez [b], Timur Narbaev [a]

[a] Department of Management and Production Engineering, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Turin, 10129, Italy

[b] Escuela Técnica Superior de Ingeniería Industrial, Universitat Politècnica de València, Camino de Vera, s/n, València, 46022, Spain

ARTICLE INFO

Keywords:
Project management
Machine Learning
Regression
Monitoring and control
Earned Value Management
Estimate at completion

ABSTRACT

Projects often face issues that trigger project controls, where Estimates at Completion (EACs) play a crucial role in determining the scope of corrective actions. Recent studies have applied supervised Machine Learning (ML) regression techniques to develop EAC models, utilizing features derived from Earned Value Management (EVM) and Earned Schedule Management (ESM) methodologies. However, these studies overlook several underfitting and overfitting issues that could compromise model robustness, leading to biased results. This paper introduces an ML pipeline designed to address these issues through automated procedures for data balancing and augmentation, feature engineering, and model training and evaluation. The pipeline was tested with 30 ML techniques on a dataset of 50 real-world construction projects. Results show that the EAC models developed through the pipeline achieve superior accuracy, precision, and timeliness to EVM and ESM ones. These findings validate the pipeline and offer practitioners an automated framework for developing robust, ML-based EAC models.

1. Introduction

On average, over 90% of construction projects experience cost or schedule overruns because of planning and execution issues [1,2]. Planning issues are due to inaccurate estimates [3], unrealistic schedules [4], and scope gaps [5]. Execution issues stem from unforeseen internal or external risks that impact project activities and resources [6].

Estimates at Completion (EACs) play a key role in project control. They are forecasts of the revised project cost and duration, calculated during the execution phase. Comparing EACs to the project baseline helps determine whether and to what extent corrective actions are needed [7,8]. These actions are aimed at recovering cost and schedule variances to ensure that the project meets the contractual budget and duration constraints.

The state-of-the-art approach to calculating EACs integrates Earned Value Analysis (EVA) with supervised Machine Learning (ML) regression techniques [9]. EVA methodologies, including Earned Value Management (EVM) [10] and Earned Schedule Management (ESM) [11], rely on three metrics: the budgeted cost of work scheduled, the budgeted cost of work performed, and the actual cost of work performed. ML techniques leverage EVA data from completed projects to infer the relationships between these metrics and the actual cost and duration at completion, and then use this information to build EAC models.

For practical use, EAC models must be robust—i.e., forecasts must be accurate, precise, and timely both during training and when applied to new data. This requires balancing the trade-off between underfitting and overfitting [12]. Underfitting occurs when models fail to capture underlying relationships in the project data, resulting in inaccurate forecasts. Conversely, overfitting occurs when models are excessively tuned to the training data and consequently unable to generalize to new, unseen data [13–15].

This study aims to improve project control by proposing an automated ML pipeline for building robust EAC models. The pipeline incorporates specific procedures to minimize underfitting and overfitting: data balancing and augmentation through interpolation to address sparsity and irregularity across project data; nested cross-validation to prevent overfitting during model training and evaluation; and the use of indirect regression that calculates forecasts by modeling intermediate variables instead of EAC values. All procedures are fully automated, requiring no manual intervention during runtime. The study evaluates the pipeline by comparing 30 ML techniques with standard EVM and ESM models using a dataset of 50 real-world construction projects. Performance is assessed in terms of accuracy and precision across the full dataset, at different progress stages, and at the individual project level.

* Corresponding author.
E-mail addresses: filippo.ottaviani@polito.it (F.M. Ottaviani), pabbalpe@dpi.upv.es (P. Ballesteros-Pérez), timur.narbaev@polito.it (T. Narbaev).

https://doi.org/10.1016/j.autcon.2025.106426
Received 13 December 2024; Received in revised form 15 July 2025; Accepted 18 July 2025
Available online 30 July 2025

Glossary

EAC	Estimate at Completion;
EVA	Earned Value Analysis;
EVM	Earned Value Management;
ESM	Earned Schedule Management; and
PMB	Performance Measurement Baseline.

Abbreviations

AdaBoost	Adaptive Boosting;
ARD	Automatic Relevance Determination;
CV	Cross-validation;
DE	Differential Evolution;
DT	Decision Tree;
DR	Direct Regression;
EN	Elastic Net;
ERT	Extremely Randomized Tree;
ERTs	Extremely Randomized Trees;
fSFS	Forward Sequential Feature Selection;
GB	Gradient Boosting;
GP	Gaussian Process;
HGB	Histogram-based Gradient Boosting;
HPT	Hyperparameters Tuning;
IR	Indirect Regression;
k-NN	k-Nearest Neighbors;
KR	Kernel Ridge;
LOGO	Leave-One-Group-Out;
ML	Machine Learning;
MLP	Multilayer Perceptron;
MLR	Multiple Linear Regression;
NN	Neural Network;
OLS	Ordinary Least Squares;
OMP	Orthogonal Matching Pursuit;
PA	Passive Aggressive;
PF	Performance Factor;
PSO	Particle Swarm Optimization;
RANSAC	Random Sample Consensus;
RF	Random Forest;
SGD	Stochastic Gradient Descent;
SVM	Support Vector Machine; and
SVR	Support Vector Regression.

The paper is organized as follows. Section 1 introduced EACs, described their role in project control, and briefly outlined the challenges associated with applying ML techniques. Section 2 reviews standard EVA methodologies and advanced EVA approaches, highlights their limitations, and compares previous studies' ML pipelines to define the research gap. Section 3 details the proposed ML pipeline, the causes of underfitting and overfitting when building EAC models, and the procedures implemented to address these issues. Section 4 compares the performance of the ML, EVM, and ESM models. Section 5 discusses the main results of the study and their theoretical and practical implications. Finally, Section 6 summarizes the key findings of the study, exposes its (de)limitations, and suggests future avenues of research.

2. Literature review

This section comprises four subsections. Section 2.1 reviews standard EVA methodologies and their limitations. Section 2.2 discusses advanced EVA approaches, highlighting their respective strengths and weaknesses, while Section 2.3 compares prior ML pipelines to identify key considerations for EAC model development. Building on these analyses, Section 2.4 delineates the research gap.

2.1. Standard EVA methodologies

EVA involves comparing project monitoring data with the Performance Measurement Baseline (PMB). Monitoring data include Actual Time (AT), Earned Value (EV), and Actual Cost (AC), where AT denotes the elapsed time, and EV and AC denote the budgeted and actual cost of work performed, respectively. The PMB comprises Planned Value (PV) data from the project start to its Planned Duration (PD) when it reaches the Budget at Completion (BAC), where PV denotes the budgeted cost of work scheduled.

Standard EVA methodologies (i.e., EVM and ESM) use index-based models to forecast EACs. EVM calculates the cost Estimate at Completion (EAC) using BAC, EV, and AC, and calculates the duration Estimate at Completion (EAC^t) using PD, EV, and PV. Conversely, ESM calculates the EAC^t using PD, AT, and the Earned Schedule (ES) metric, which denotes the time when the current EV was planned to be achieved according to the PMB. Appendix A further details on EVM and ESM features and their cost and duration EAC models.

Recent studies continue to use EVM and ESM models as benchmarks for EAC forecasting. For cost forecasting, Zwikael et al. (2000) [16] evaluated five EAC models across 12 real-world projects, reporting Mean Absolute Percentage Error ($MAPE$) values ranging from 0.11 to 0.27. Barrientos-Orellana et al. (2023) [17] tested EVM and ESM models on 4100 simulated projects, reporting mean $MAPE$ values between 0.06 and 0.20. For duration forecasting, both de Andrade et al. (2019) [18] and Ottaviani et al. (2024) [19] analyzed 57 and 65 real-world projects, respectively, using the same database employed in this study. De Andrade et al. found that Eq. (A.16) produced improved accuracy with mean $MAE = 0.1180$ and mean $RMSE = 0.1670$. Ottaviani et al. reported that the same equation achieved a Mean Absolute Error (MAE) of 0.2418 and a Root Mean Squared Error ($RMSE$) of 0.4825, whereas Eq. (A.13) yielded $MAE = 0.1180$ and $RMSE = 0.1670$.

Despite their continued use, standard EVA methodologies present several limitations that undermine their reliability. First, they depend on the accuracy of the monitoring data and the PMB; inconsistencies in either can bias the EVA metrics and, consequently, the EACs [3,4]. Second, neither EVM nor ESM accounts for the relationship between cost and schedule performances [20], nor do they consider interactions with other critical project variables such as scope, resources, or quality [21,22]. Finally, both methodologies base EAC calculations solely on current values, disregarding historical performance trends [23,24] and the current phase [19].

2.2. Advanced EVA approaches

Multiple studies have addressed the limitations of standard EVA methodologies by integrating EVM and ESM with advanced analytical techniques. These approaches include alternative performance factors, time series analysis, nonlinear regression, and Bayesian inference.

2.2.1. Alternative performance factors

Alternative performance factors (PFs) involve the use of composite, average-based, or progress-based PFs within EAC models. Composite PFs capture the relationship between cost and schedule performances using products or weighted sums of standard PFs [25,26]. Average-based PFs capture performance trends using cumulative, moving, or exponential moving averages of standard PFs [24,27–29]. Progress-based PFs adjust EACs based on the current progress stage, employing either physical progress (i.e., EV/BAC) or "temporal" progress (i.e., ES/PD) in their calculation [19].

Empirical studies show that no single PF consistently outperforms others, underscoring that the most appropriate PF depends on the specific characteristics of each project. This limitation stems from the context-dependent assumptions embedded in PF calculations. For instance, composite PFs may become unreliable when cost and schedule performance are weakly correlated, while average-based PFs can underperform in long-duration projects or those impacted by major risks—situations where recent performance data offer stronger predictive value.

2.2.2. Time series analysis

Time series analysis studies involve applying time series techniques to capture trends in project performance. Time series techniques can be used to analyze the work performed and actual expenditure curves [30], or trends in cost and schedule performance indices [22,31,32].

Similar to alternative PFs, studies applying time series techniques have produced mixed results, with the most effective technique varying by project context. However, time series models introduce additional limitations. They require explicit modeling of both trend — the overall direction of performance — and seasonality — recurring temporal patterns — which increases model complexity and reduces adaptability. Moreover, these models overlook external factors that affect project outcomes but are not reflected in historical performance data alone.

2.2.3. Nonlinear regression

Nonlinear regression studies capture performance trends and current progress by fitting a theoretical profile to the PMB and using it to model and project the EV and AC curves. Projections of EV are used to calculate the EAC^t [33,34], while AC projections are used to calculate the EAC [35–38].

Several studies have validated nonlinear regression in forecasting both project duration [39] and cost [20,40]. On the one hand, these approaches highlight the importance of analyzing project S-curves. On the other hand, EAC accuracy depends on the geometric characteristics of the selected theoretical profile. These characteristics include the weights of input data, the optimization function, and the method for determining the parameters of the fitted model—factors that can lead to inconsistent results across different projects.

2.2.4. Bayesian inference

Bayesian inference studies account for additional features influencing project cost and schedule performances by integrating internal project data with external information [41]. This approach has been used to model the probability distributions of project S-curves [42,43], fluctuations in cost and schedule performances [44], EACs [33], and the likelihood of risk events [45], by incorporating data from similar past projects.

Although these studies report the highest performance compared to other approaches, Bayesian methods present notable limitations. First, they rely on prior assumptions — i.e., the choice of prior distribution and the estimation of its parameters — which can introduce bias if not properly specified. Second, their implementation is mathematically complex, often requiring advanced statistical expertise. Finally, Bayesian models must continuously update the posterior distribution as new project data become available, increasing both computational and modeling overhead.

2.3. Machine learning studies

Recent advances in ML have prompted its application in project control. In theory, ML techniques can overcome several core limitations of EVA methodologies. First, they can capture planning inaccuracies by adjusting bias parameters. Second, they can analyze cost, duration, and other project variables simultaneously, identifying patterns and relationships from training data without requiring explicit model formulations [46,47]. Finally, ML techniques can account for performance trends and current progress by incorporating physical or temporal progress indicators and time-based features [22].

Table 1 compares the ML pipelines used by previous studies to develop EAC models. It outlines the forecasting targets (cost, duration, or both), the type and quantity of data used (real or synthetic projects along with the number of projects), and whether the pipeline incorporated data scaling, cross-validation (CV), hyperparameter tuning (HPT), and data balancing or augmentation procedures, specifying the techniques applied. All acronyms appearing in Table 1 are defined in the Glossary.

Table 1 highlights both minor and major considerations for applying ML techniques to build EAC models. Minor considerations include using data from multiple real projects to better reflect the real relationship between cost and schedule performances [51,52,56,59,62,69–71], and automating HPT to avoid reliance on manual intervention for optimizing model performance [56,57,59,67,69,74]. Major considerations involve scaling project data, applying rigorous CV procedures, and balancing data—all of which significantly affect model learning. Scaling prevents large projects from dominating model training due to their higher numerical values, thereby reducing the risk of underfitting smaller projects [49,56,58,59,61,62,66,69]. Rigorous CV is critical for assessing model generalizability, as model performance evaluation can vary substantially depending across different train-test splits, increasing the risk of overfitting and undermining reproducibility [51,52,55–57,67,69]. Data balancing further reduces bias toward dominant outcomes, a common cause of overfitting to frequent patterns and underfitting to underrepresented cases [48,55]. Finally, the lack of data augmentation can impair model performance on unseen progress stages — such as early or late stages — increasing the risk of overfitting to known conditions.

2.4. Gap

This section reviewed standard EVA methodologies, advanced EVA approaches, and ML-based studies for project performance forecasting. Standard methodologies are widely used and easy to implement but struggle to capture dynamic project behavior or external influences. Advanced approaches extend EVA's analytical scope but often rely on restrictive assumptions, added complexity, or context-specific calibration. ML techniques model complex, nonlinear relationships, overcoming the limitations of EVM and ESM without requiring continuous intervention, as Bayesian methods do. However, their effectiveness depends on careful model building. If not properly addressed, overfitting or underfitting issues can compromise accuracy and reduce their reliability. To mitigate these risks, this study proposes an ML pipeline that automates key steps to generate robust EAC models.

3. Research method

This section outlines the proposed ML pipeline, which consists of four stages: data collection, data preprocessing, feature engineering, and model training and evaluation. Each stage comprises several steps. The following subsections explain the purpose of each step, identify potential sources of underfitting and overfitting, and describe the procedures implemented to mitigate these issues.

3.1. Data collection

Data collection involves gathering EVA data from completed projects into a structured dataset for training the ML models. Project selection should prioritize those similar to the intended application context, as higher similarity increases the likelihood that ML techniques will detect the underlying relationships between EVA data and project outcomes, thereby reducing underfitting.

EVA data include BAC, PD, AD, AT, PV, EV, and AC.

3.2. Data preprocessing

Data preprocessing involves formatting the EVA data to prepare it for model training. This stage includes the data transformation and data balancing and augmentation steps.

Table 1
Comparison of studies using ML techniques for EAC forecasting.

Study	Target		ML Method(s)	Project data		Scaling	Cross validation	Hyperparameters tuning	Balancing/Augmentation
	Cost	Time		Number	Type				
[48]	✓		NN	2	Synthetic			Manual	Balancing
[49]		✓	DT, NN	1	Real				
[50]		✓	NN, OLS	Undef.	Synthetic				
[51]	✓		GP	15	Real	✓	Train-Test	PSO	
[52]	✓		SVM	13	Real		Train-Test	fMGA	
[53]	✓		NN						
[54]	✓	✓	*k*-NN	3	Undef.		Train-Test	Manual	
[55]	✓	✓	SVM	900	Synthetic		*k*-fold	Grid Search CV	Balancing
[56]	✓		SVM	13	Real	✓	*k*-fold	DE CV	
[57]		✓	DT, RF, GB, SVM	90	Synthetic		*k*-fold	Grid Search CV	
[58]		✓	DT, RF, GB, SVM, *k*-NN	10	Synthetic	✓	Train-Test		
[59]		✓	NN	11	Real	✓	*k*-fold		
[60]	✓		NN	11	Real			Manual	
[61]	✓		NN	15	Real	✓		Manual	
[62]	✓		NN, MLR	50	Real	✓		Manual	
[63]	✓	✓	NN	Undef.	Real		Train-Test	Manual	
[64]	✓	✓	NN	Undef.	Real		Train-Test	Manual	
[65]	✓	✓	NN	Undef.	Real		Train-Test		
[66]	✓		NN	Undef.	Undef.	✓		Manual	
[67]		✓	GB, RF, XGBoost, AdaBoost	1	Real		*k*-fold	Grid Search	
[68]		✓	AdaBoost, GB, *k*-NN, RF, SVM	1	Synthetic				
[69]	✓		SVM, RF, NN, DT, *k*-NN, MLR	10	Real	✓	*k*-Fold CV	Grid Search CV	
[70]	✓		XGBoost, RF, SVR	110	Real		Train-Test	Manual	
[71]	✓		NN	13	Real		Train-Test	Manual	
[72]	✓		NN, GP, SVM	1	Real				
[73]	✓		MLR, SVR, DT, RF	11	Real		*k*-fold	Manual	
This study	✓	✓	See Table 4	50	Real	✓	LOGO CV	Grid Search (LOGO) CV	Balancing, Augmentation

3.2.1. Data transformation

Data transformation involves scaling EVA data. This procedure reduces underfitting by limiting the amount of information that ML techniques analyze to infer relationships within EVA data and by preventing ML techniques that rely on distance-based calculations or regularization techniques (e.g., L1 and L2) from being biased by different data scales.

The pipeline scales EVA data by dividing cost metrics by BAC and time metrics by PD, as follows: $\mathrm{BAC_s} = \mathrm{PD_s} = 1$, $\mathrm{AD_s} = \mathrm{AD}/\mathrm{PD}$, $AT_s = AT/\mathrm{PD}$, $PS = PV/\mathrm{BAC}$, $PC = EV/\mathrm{BAC}$, and $AC_s = AC/\mathrm{BAC}$. The subscript "s" denotes scaled metrics, except for PV and EV, whose scaled versions are referred to as Percentage Scheduled (PS) and Percentage Completed (PC), respectively.

3.2.2. Data balancing and augmentation

Data balancing and augmentation are interrelated steps. Data balancing involves equalizing the number of records across projects and, within each project, across progress stages. Data augmentation provides the means to achieve this balance by generating additional synthetic records. These steps reduce underfitting by preventing projects with more (or fewer) records from disproportionately influencing model training and by ensuring models are trained across all progress stages uniformly.

The pipeline implements both data balancing and augmentation using a fixed number of synthetic records generated through linear interpolation of AT_s, PS, and AC_s at equidistant PC values. Linear interpolation assumes a constant rate of change between consecutive records, which aligns with the assumption adopted by Lipke (2003) [11] in the calculation of the ES metric, and by more recent studies when augmenting project data [75,76].

Let x denote the metric to interpolate, $\tilde{x}$ the interpolated value, and z a predetermined PC value at which interpolation is performed. Then, $\tilde{x}(PC = z)$ is determined as per

$$\tilde{x}(PC = z) = x_i + (z - PC_i)/(PC_{i+1} - PC_i) \cdot (x_{i+1} - x_i), \quad (1)$$

where $PC_i \leq z \leq PC_{i+1}$.

Alternatively, nonlinear interpolation could be used [77], provided it preserves the monotonic and cumulative nature of AT_s, PS, and AC_s. In this case, the interpolation rate would follow a theoretical model characterized by non-uniform growth. However, when PC values are sufficiently dense (e.g., in 0.01 to 0.05 increments), the added complexity of nonlinear interpolation rarely justifies its marginal gain in accuracy.

To mitigate the risk of distortion introduced by interpolation, projects with PC value variations greater than 0.50 across consecutive time periods were excluded from the dataset. Discontinuities are susceptible to interpolation inaccuracies that can affect the quality of the augmented data disproportionately. Although this approach does not eliminate all potential sources of error, it is a practical strategy for preventing error accumulation.

3.3. Feature engineering

Feature engineering involves transforming EVA data into additional features that enhance the learning capability of ML models. These features are used either as inputs (independent variables) or targets (dependent variables) during model training. This stage generates features that capture latent relationships in project behavior, thereby reducing underfitting by enriching the input space with more informative patterns. It also helps mitigate overfitting by enabling subsequent feature selection techniques to discard irrelevant or redundant features, thereby limiting model complexity [14,78].

3.3.1. Input features

Input features comprise EVA scaled metrics and EVM and ESM features. The EVA scaled metrics are those evaluated in Section 3.2.1. The EVM features include the scaled versions of Eq. (A.5) (CV_s), Eq. (A.6) (CPI), Eq. (A.10) (SV_s), and Eq. (A.9) (SPI). The ESM features include the scaled Eq. (A.12) (ES_s), Eq. (A.14) (SV_s^t), and Eq. (A.15) (SPI^t).

3.3.2. Target features

Target features depend on both the forecasting target and the regression method. The forecasting target is either $\mathrm{AC_s}(\mathrm{AD_s})$ (cost forecasting) or $\mathrm{AD_s}$ (duration forecasting). The method is either direct regression (DR) or indirect regression (IR). While DR sets the target variable to the forecasting target itself, IR sets the target variable to an intermediate

Table 2
Model development features.

Type	Method	Symbol	Name
Input	Both	AT_s	(Scaled) Actual Time
		$\mathrm{BAC_s/PD_s}$	(Scaled) Budget at Completion/Planned Duration
		PS	Percentage Scheduled
		PC	Percentage Completed
		AC_s	(Scaled) Actual Cost
		CPI	Cost Performance Index
		CV_s	(Scaled) Cost Variance
		SPI	EVM Schedule Performance Index
		ES_s	(Scaled) Earned Schedule Metric
		SV_s^t	(Scaled) Earned Schedule Variance
		SPI^t	Earned Schedule Performance Index
Target	DR	$\mathrm{AC_s(AD_s)}$	(Scaled) Actual Cost at Completion
		$\mathrm{AD_s}$	(Scaled) Actual Duration at Completion
	IR	cPF^*	Target Cost Performance Factor
		sPF^*	Target Schedule Performance Factor

variable, which is then used to calculate the forecasting target through a specific formula.

The pipeline implements both regression methods for the two forecasting targets. Let y denote the real value of the target feature, $\hat{y}$ denote the forecast, and $\mathbf{X}$ denote the set of input features. The DR method evaluates forecasts as per

$$\hat{y} = f^{\mathrm{DR}}(\mathbf{X}), \tag{2}$$

where f^{DR} denotes the regression models developed through DR using $\mathrm{AC_s(AD_s)}$ or $\mathrm{AD_s}$ as target features. In contrast, the IR method evaluates cost forecasts as per

$$\hat{y} = AC_s + (1 - PC)/\widehat{cPF} \tag{3}$$

with

$$\widehat{cPF} = f^{\mathrm{IR}}(\mathbf{X}) \tag{4}$$

where f^{IR} denotes the regression models developed through IR using cPF^* as target feature, which is defined as the value such that

$$\mathrm{AC_s(AD_s)} = AC_s + (1 - PC)/cPF^*. \tag{5}$$

Similarly, the IR method evaluates duration forecasts as per

$$\hat{y} = AT_s + (1 - ES_s)/\widehat{sPF} \tag{6}$$

with $\widehat{sPF} = g^{\mathrm{IR}}(\mathbf{X})$ where g^{IR} denotes the regression models developed through IR using sPF^* as target feature, which is defined as the value such that

$$\mathrm{AD_s} = AT_s + (1 - ES_s)/sPF^*. \tag{7}$$

3.3.3. Summary

Table 2 summarizes the model development features, specifying their use (input or target), the method in which they are employed (DR, IR, or both), and formula.

3.4. Model training and evaluation

Model training and evaluation are interrelated steps. Model training involves feature selection, model training, and HPT steps. Model evaluation involves benchmarking the trained models against standard EVM and ESM models to assess their performance.

The pipeline implements the Leave-One-Group-Out (LOGO) CV procedure for both model evaluation and internal training steps. LOGO CV splits the dataset P times, where P is the number of projects in the dataset. In each split, the records of the pth project serve as the validation set, while the remaining projects records serve as the training set. This procedure prevents both time-series and group leakage, which would otherwise bias model evaluation and distort performance results. Moreover, repeating the train-validate split for each project in the dataset ensures the consistency and reproducibility of results.

3.4.1. Feature selection

Feature selection involves identifying a subset of input features to be used for model development. This step reduces overfitting by limiting the number of relationships ML techniques must analyze, accelerates model training by eliminating irrelevant or redundant features, and enhances model interpretability by focusing on the most informative features.

The pipeline uses the forward Sequential Feature Selection (fSFS) procedure with LOGO CV. The procedure initializes the model without features. At each LOGO CV iteration, the procedure identifies the feature that, if added, would yield the greatest improvement in a CV scorer. The feature is incorporated into the model if the improvement in performance exceeds a predefined threshold. This process is repeated until no further improvement or all input features are included.

3.4.2. Hyperparameters tuning

HPT involves selecting the optimal values for the hyperparameters to maximize model performance.

The pipeline employs the Grid Search procedure with LOGO CV to tune the models' hyperparameters. The procedure involves searching through a specified parameter grid and evaluating all possible combinations of the hyperparameter values on a given dataset using LOGO CV. For each combination, the model is trained and validated multiple times, and the combination that results in the best CV scorer is selected.

3.4.3. Benchmarking

The study benchmarks the ML techniques by comparing their performance to that of EVM and ESM models.

Performance is evaluated using two metrics: MAE and $RMSE$. MAE measures the forecasting accuracy of the model, while $RMSE$ captures its precision by assigning greater weight to larger errors. Both metrics are expressed in the same units as the forecasting target, making them directly interpretable and suitable for comparative analysis.

Let i denote the ith record in the dataset, where $i = 1, 2, \ldots, n$, and let E_i denote the forecast residual for the ith record, as per

$$E_i = y_i - \hat{y}_i. \tag{8}$$

Then, MAE is determined as per

$$MAE = 1/n \sum_{i=1}^{n} |E_i|. \tag{9}$$

Instead, $RMSE$ is determined as per

$$RMSE = \sqrt{1/n \sum_{i=1}^{n} E_i^2}. \tag{10}$$

The EVM models include EVM(CPI) (Eq. (A.3)), EVM(1) (Eq. (A.3)), and EVM(SPI) (Eq. (A.8)). The ESM models include

Table 3
Proposed ML pipeline workflow.

Stage/Step	Procedure	Description
Data collection		Retrieve projects BAC, PD, AT, PV, EV, and AC
Data preprocessing		
Data transformation	Scaling	Evaluate BAC$_s$, PD$_s$, AT_s, PS, PC, and AC_s
Data balancing and augmentation		Remove projects where $PC_i - PC_{i-1} \geq 0.50$
	Interpolation	Evaluate $\widetilde{AT}_s$, $\widetilde{PS}$, and $\widetilde{AC}_s$
Feature engineering		
Input features		Evaluate ES_s, CPI, CV_s, SPI, SV_s, SPI^t, and SV^t_s
Target features		Evaluate AC$_s$(AD$_s$), AD$_s$, cPF^*, and sPF^*
		for Target ∈ [Cost, Duration]:
		for Method ∈ [DR, IR]:
Model training and evaluation	LOGO CV	for $p \in [0..P]$:
		Train-Validate Split
Feature Selection	fSFS (LOGO) CV	Select model features
Hyperparameters Tuning	Grid Search (LOGO) CV	Select model hyperparameters
		Evaluate pth project forecasts
		for Performance Metric ∈ [MAE, $RMSE$]:
		Evaluate Target-Method-Performance Metric

Table 4
Machine learning methods tested.

Category/Subcategory	Method	Abbreviation
Linear		
Linear	Ordinary Least Squares	OLS
	Ridge	Ridge
	Least Absolute Shrinkage and Selection Operator	Lasso
	Elastic Net	EN
	Least Angle Regression	Lars
	Lasso Least Angle Regression	Lasso Lars
	Orthogonal Matching Pursuit	OMP
	Passive Aggressive	PA
Bayesian	Bayesian Ridge	BR
	Automatic Relevance Determination	ARD
Generalized Linear Model	Tweedie	Tweedie
Stochastic Gradient Descent	Stochastic Gradient Descent	SGD
	One-Class Support Vector Machine using SGD	SGD1cSVM
Robust Regression	Random Sample Consensus	RANSAC
	Huber	Huber
Nonlinear		
Kernel Ridge	Kernel Ridge	KR
Support Vector Machine	Support Vector Regression	SVR
	Nu Support Vector Regression	NuSVR
k-Nearest Neighbors	k-Nearest Neighbors	k-NN
Gaussian Process	Gaussian Process	GP
Decision Trees	Decision Tree	DT
	Extremely Randomized Tree	ERT
Ensemble Methods	Random Forest	RF
	Extremely Randomized Trees	ERTs
	Adaptive Boosting	AdaBoost
	Gradient Boosting	GB
	Histogram-based GB	HGB
	XGBoost	XGB
	XGBoost RF	XGB RF
Neural Network	Multilayer Perceptron	MLP

ESM(SPI^t) (Eq. (A.16)) and ESM(1) (Eq. (A.13)). Cost and duration forecasts generated by the above models were scaled using BAC and PD, respectively.

The study considers ML models robust if their MAE and $RMSE$ values, as evaluated using the LOGO CV procedure, are close to or lower than those of the EVM and ESM models. Without the LOGO iterator, models may appear accurate and precise for a specific train-test split, but their performance could vary significantly across different splits — undermining the reliability of the evaluation.

Given the inherent variability in evaluating EAC model performance — and the absence of established thresholds for acceptable MAE or $RMSE$ values in EAC forecasting — it is currently infeasible to determine which model is objectively "better" based solely on these metrics. Instead, practitioners should prioritize models that demonstrate consistent performance across multiple project types and progress stages. Such models can be used to generate a range of forecast estimates, each accompanied by a confidence level derived from the model's observed behavior under comparable conditions.

This study compares EAC models at the dataset, progress stage, and project level. Dataset-level comparison provides a summary view of model accuracy and precision. Progress stage comparison offers insights into model timeliness by assessing performance at various stages of project advancement. Project-level comparison allows for an analysis of model robustness and highlights variations in forecasting difficulty across different projects.

3.5. Pipeline summary

Table 3 outlines the ML pipeline stages/steps, procedures, and description. This format clarifies the sequential and nested logic of the workflow.

Table 5
Project dataset characteristics.

Code	Type	Activities	BAC [€]	PD [Days]	PD [TPs]	$AC_s(AD_s)$	AD_s
C2011–10	Residential	32	484,398	195	39	1.0218	1.0513
C2011–12	Commercial	49	3,027,133	442	7	1.0249	1.0000
C2011–13	Industrial	135	21,369,836	525	105	1.2203	1.1429
C2012–13	Civil	71	336,410	125	25	1.0419	1.1200
C2012–17	Residential	33	241,015	145	29	1.3064	1.4138
C2013–01	Civil	42	1,069,532	152	6	1.2291	1.0000
C2013–02	Civil	181	1,236,604	403	17	0.9271	1.0000
C2013–03	Institutional	55	15,440,866	425	18	1.0581	1.0000
C2013–04	Institutional	252	2,113,684	333	7	1.1887	1.5714
C2013–07	Residential	46	180,476	170	10	0.9698	1.1000
C2013–08	Residential	42	501,030	216	10	1.1509	1.3000
C2013–09	Commercial	71	1,537,399	291	8	1.1038	1.2500
C2013–12	Institutional	27	818,440	115	3	1.0750	1.6667
C2014–04	Industrial	24	62,385,598	522	24	1.0504	1.5000
C2014–05	Residential	25	532,410	228	11	1.1108	1.1818
C2014–06	Residential	29	3,486,375	547	17	1.0323	1.1176
C2014–07	Residential	25	1,102,537	353	12	1.1698	1.1667
C2014–08	Residential	39	1,992,222	233	11	1.1948	1.1818
C2015–01	Institutional	23	612,769	131	6	1.0550	1.5000
C2015–02	Civil	217	1,121,317	417	8	0.8633	1.1250
C2015–03	Industrial	135	2,244,091	257	9	0.8328	1.1111
C2015–05	Residential	64	2,524,765	120	4	1.0154	1.2500
C2015–08	Commercial	186	467,297	191	8	0.9885	1.0000
C2015–27	Civil	18	22,704	68	5	1.1149	1.2000
C2015–29	Institutional	204	1,874,497	284	8	1.0067	1.0000
C2015–30	Residential	40	440,941	244	14	1.0000	1.0000
C2015–31	Residential	29	1,310,723	271	16	0.9782	1.3125
C2015–33	Civil	12	214,418	50	3	1.0484	1.6667
C2015–34	Civil	13	511,326	120	4	0.8613	1.7500
C2015–35	Residential	10	14,956,314	850	38	1.0744	1.0789
C2016–01	Civil	28	671,384	225	12	1.0481	1.1667
C2016–02	Civil	23	962,182	229	12	1.0106	1.0833
C2016–03	Civil	25	926,888	203	10	0.9826	1.1000
C2016–07	Commercial	110	930,179	224	8	1.0028	1.3750
C2016–11	Residential	55	162,472	241	5	1.0044	1.0000
C2016–12	Residential	59	222,858	291	5	1.0154	1.0000
C2016–13	Residential	51	367,952	306	4	1.0308	1.2500
C2016–14	Residential	48	218,366	321	5	1.0167	1.0000
C2016–15	Residential	13	95,694	126	4	1.0530	1.0000
C2016–27	Residential	16	813,663	78	3	1.0812	1.3333
C2016–28	Residential	19	569,178	71	4	1.0297	1.0000
C2016–29	Residential	19	1,797,874	129	4	1.0347	1.0000
C2016–30	Residential	23	1,319,736	85	3	1.0255	1.3333
C2016–31	Residential	23	488,936	105	3	1.0195	1.3333
C2016–32	Residential	22	477,381	89	4	1.0411	1.0000
C2016–33	Residential	23	377,282	116	3	1.0465	1.3333
C2016–34	Residential	23	362,476	83	3	1.0590	1.0000
C2018–10	Civil	42	115,115	99	20	1.0000	1.2000
C2019–01	Residential	86	1,292,979	533	8	1.0177	1.2500
C2019–02	Residential	18	734,602	352	9	1.0190	1.0000

4. Results

The following subsections compare the performance of 30 ML techniques, implemented through the pipeline, against EVM and ESM models using a real project dataset.

The 30 ML techniques included in this study were selected over more advanced alternatives for three main reasons. First, they have been widely adopted in prior EAC studies, as summarized in Table 1, whereas more complex methods are rarely applied consistently across multiple works. Second, the dataset is limited in size, and complex nonlinear models tend to introduce substantial overhead with only marginal improvements in accuracy and precision—thereby increasing the risk of overfitting. Third, the selected methods are publicly available and straightforward to implement, making them well-suited for practical use in project management contexts where transparency and reproducibility are essential. Table 4 lists the ML techniques tested, along with their category (linear or nonlinear), subcategory, and abbreviation.

The pipeline was implemented as a Python 3.9.19 script using Scikit-learn 1.4.2 for the LOGO CV, fSFS CV, and Grid Search CV procedures, and for training all ML models except XGB and XGB RF, which were trained using XGBoost 2.1.1. Records were interpolated at 0.05 PC increments — from 0.05 to 0.95 — the 0.05 interval was chosen arbitrarily to ensure full coverage of project stages without generating an excessive number of data points for progress-stage-level analyses. The LOGO CV scorer within the fSFS CV and Grid Search CV procedures was set to MAE. The fSFS CV threshold value was set arbitrarily to 1e-6 (based on the range of the target variables). The hyperparameters values for the Grid Search CV procedure are listed in Appendix B.

The real project dataset comprises EVA data from 50 construction and construction engineering projects sourced from the OR&S database [7], selected for its relevance to EAC research [19,79]. Table 5 summarizes key characteristics, including project code, construction type, number of tracking periods, number of activities, budget (i.e., BAC), contracted duration (i.e., PD), and values at completion. Project codes and construction types were defined by the database creators. BAC, PD (in days), activity count, and actual completion values were extracted from individual project files. To enable cross-project comparison, cost and duration at completion were scaled relative to BAC

Table 6
Dataset-level EAC models regression metrics.

Model	Cost				Duration			
	MAE		$RMSE$		MAE		$RMSE$	
	DR	IR	DR	IR	DR	IR	DR	IR
EVM								
EVM(CPI)	0.0530		0.1030					
EVM(1)	0.0417		0.0688					
EVM(SPI)					0.2291		0.5048	
ESM								
ESM(SPI^t)					0.2567		0.6094	
ESM(1)					**0.1266**		0.1785	
ML								
OLS	0.0421	0.0445	0.0675	0.0720	0.1339	0.1289	0.1680	0.1812
Ridge	0.0482	0.0445	0.0715	0.0720	0.1344	0.1288	0.1672	0.1802
Lasso	0.0421	0.0445	0.0675	0.0720	0.1339	0.1289	0.1680	0.1812
EN	0.0609	0.0445	0.0888	0.0719	0.1626	0.1285	0.2028	0.1806
Lars	0.0414	0.0445	0.0697	0.0720	0.1333	0.1289	0.1665	0.1812
Lasso Lars	0.0420	0.0445	0.0683	0.0720	0.1339	0.1289	0.1680	0.1812
OMP	0.0419	0.0445	0.0686	0.0720	0.1292	0.1289	0.1644	0.1812
PA	0.0477	0.0436	0.0702	0.0724	0.1282	0.1261	**0.1630**	0.1695
BR	0.0422	0.0445	0.0675	0.0719	0.1340	0.1288	0.1678	0.1812
ARD	0.0421	0.0445	0.0675	0.0719	0.1322	0.1291	0.1659	0.1816
Tweedie	0.0603	0.0431	0.0882	0.0705	0.1592	0.1283	0.1974	0.1786
SGD	0.0602	0.0430	0.0926	0.0739	0.1432	0.1280	0.1896	0.1725
SGD1CSVM	0.0689	0.0417	0.0980	0.0688	0.1903	0.1266	0.2752	0.1785
RANSAC	**0.0395**	0.0411	0.0680	0.0725	0.1408	0.1390	0.1850	0.1928
Huber	0.0396	**0.0385**	**0.0662**	0.0688	0.1317	0.1250	0.1675	**0.1630**
KR	0.0477	0.0449	0.0708	0.0733	0.1339	0.1287	0.1668	0.1800
SVR	0.0549	0.0436	0.0820	0.0688	0.1420	1.0981	0.1805	29.3463
NuSVR	0.0413	0.0397	0.0679	**0.0646**	0.1407	0.1412	0.1806	0.2221
k-NN	0.0550	0.0458	0.0874	0.0669	0.1555	0.1290	0.1998	0.1666
GP	0.0457	0.0467	0.0766	0.0744	0.3260	0.1283	2.1544	0.1803
DT	0.0480	0.0412	0.0759	0.0682	0.1492	0.1325	0.1981	0.1762
ERT	0.0480	0.0412	0.0759	0.0682	0.1492	0.1325	0.1981	0.1762
RF	0.0483	0.0412	0.0759	0.0682	0.1492	0.1334	0.1981	0.1768
ERTs	0.0517	0.0408	0.0797	0.0673	0.1466	**0.1247**	0.1883	0.1648
AdaBoost	0.0520	0.0477	0.0794	0.0734	0.1374	0.1331	0.1729	0.1865
GB	0.0506	0.0414	0.0796	0.0681	0.1429	0.1298	0.1839	0.1707
HGB	0.0569	0.0390	0.0862	0.0656	0.1595	0.1269	0.2072	0.1671
XGB	0.0578	0.0397	0.0875	0.0666	0.1602	0.1271	0.2081	0.1673
XGB RF	0.0609	0.0444	0.0888	0.0719	0.1625	0.1285	0.2027	0.1806
MLP	0.0525	0.0465	0.0781	0.0789	0.1429	0.1319	0.1754	0.1807
DR	9		13		2		15	
IR	21		17		28		15	

and PD, respectively. Although the database does not explicitly report project locations, most projects were executed in Belgium, the Netherlands, and Italy. Additional project details are available in Vanhoucke (2023) [7].

4.1. Dataset level

Table 6 presents the regression metrics values for the EVM models, ESM models, and ML techniques at the dataset level. Rows DR and IR indicate the number of ML techniques where DR models outperform IR models, and vice versa. The models with their respective metrics values in bold produced the best estimate within the column estimates. EVM and ESM models metrics values are the same for both DR and IR methods.

In cost forecasting, Huber IR shows the lowest MAE (0.0385), while NuSVR IR shows the lowest $RMSE$ (0.0646). Comparing DR and IR methods, IR models show lower MAE than DR models in 21/30 cases and show lower $RMSE$ in 17/30 cases. In duration forecasting, ERTs IR shows the lowest MAE (0.1247), while both PA DR and Huber IR tie for the lowest $RMSE$ (0.1630). Comparing DR and IR methods, IR models show lower MAE than DR models in 28/30 cases and show lower $RMSE$ in 15/30 cases.

4.2. Progress stage level

Table 7 presents the best-performing models and their regression metrics values at each progress stage. Rows EVM/ESM and ML indicate the number of progress stages where EVM/ESM models and ML techniques show the lowest values of the regression metrics, respectively. Rows DR and IR indicate the number of progress stages where DR models outperform IR models, and vice versa. If EVM or ESM models show the lowest values in both DR and IR methods for a given target-metric scenario, they are excluded from the count.

Overall, Table 7 shows that no single model consistently dominates across all progress stages. This is consistent across all target-metric-method scenarios. Comparing ML techniques and EVM/ESM models, ML techniques outperform EVM and ESM models across all scenarios except for the Duration-MAE. The best-performing cost EAC models include Huber DR (11 occurrences in MAE and 8 in $RMSE$) and NuSVR IR (6 occurrences in MAE and 8 in $RMSE$). The best-performing duration EAC models include ESM(1) (6 occurrences in MAE-DR), ERTs (7 occurrences in MAE-IR), OMP (7 occurrences in $RMSE$-DR), and Huber (6 occurrences in $RMSE$-IR). Comparing DR and IR methods, DR and IR models perform similarly in Cost-MAE (10 vs 9) and Duration-$RMSE$ (8 vs 9), while IR models outperform DR models in Cost-$RMSE$ and Duration-MAE scenarios.

4.3. Project level

Table 8 presents the best-performing models and their regression metrics values for each project. Rows EVM/ESM, ML, DR, and IR are analogous to those in Table 7.

Table 7
Progress stage-level best EAC models and corresponding regression metrics.

PC	Cost								Duration							
	MAE				$RMSE$				MAE				$RMSE$			
	DR		IR		DR		IR		DR		IR		DR		IR	
	Model	Value	Model	Value	Model	Value	Model	Value	Model	Value	Model	Value	Model	Value	Model	Value
.05	NuSVR	0.0549	SVR	0.0569	NuSVR	0.0806	SVR	0.0829	ERTs	0.1544	ERTs	0.1571	AdaBoost	0.1979	Huber	0.2051
.10	Huber	0.0533	Huber	0.0507	NuSVR	0.0784	Huber	0.0779	OMP	0.1488	ERTs	0.1489	PA	0.1899	Huber	0.1919
.15	Huber	0.0522	Huber	0.0493	Huber	0.0801	Huber	0.0774	AdaBoost	0.1430	Huber	0.1453	AdaBoost	0.1778	Huber	0.1831
.20	Huber	0.0515	Huber	0.0504	NuSVR	0.0820	HGB	0.0815	OMP	0.1440	Huber	0.1419	AdaBoost	0.1796	Huber	0.1792
.25	Huber	0.0509	GB	0.0511	NuSVR	0.0805	GB	0.0765	OMP	0.1445	Huber	0.1421	AdaBoost	0.1789	Huber	0.1795
.30	Huber	0.0480	HGB	0.0487	Huber	0.0756	GB	0.0748	OMP	0.1371	ERTs	0.1335	OMP	0.1694	ERTs	0.1677
.35	Huber	0.0474	HGB	0.0477	Huber	0.0744	HGB	0.0739	ESM(1)	0.1332	ERTs	0.1296	OMP	0.1639	ERTs	0.1629
.40	Huber	0.0459	HGB	0.0467	Huber	0.0729	HGB	0.0724	OMP	0.1300	ERTs	0.1254	OMP	0.1613	ERTs	0.1606
.45	Huber	0.0451	HGB	0.0457	Huber	0.0724	HGB	0.0717	ESM(1)	0.1298	ERTs	0.1268	OMP	0.1625	ERTs	0.1635
.50	Huber	0.0430	HGB	0.0431	Huber	0.0722	NuSVR	0.0688	ESM(1)	0.1315	ERTs	0.1301	OMP	0.1668	ERTs	0.1706
.55	Huber	0.0426	NuSVR	0.0408	Huber	0.0718	NuSVR	0.0633	ESM(1)	0.1317	ESM(1)	0.1317	OMP	0.1656	ERTs	0.1714
.60	Huber	0.0404	NuSVR	0.0360	Huber	0.0706	NuSVR	0.0594	ESM(1)	0.1322	ESM(1)	0.1322	OMP	0.1668	Huber	0.1748
.65	RANSAC	0.0351	NuSVR	0.0321	EVM(1)	0.0624	NuSVR	0.0528	EVM(SPI)	0.1217	EVM(SPI)	0.1217	EVM(SPI)	0.1595	EVM(SPI)	0.1595
.70	RANSAC	0.0303	NuSVR	0.0280	NuSVR	0.0533	NuSVR	0.0455	ESM(1)	0.1114	ESM(1)	0.1114	EVM(SPI)	0.1493	EVM(SPI)	0.1493
.75	RANSAC	0.0253	NuSVR	0.0243	NuSVR	0.0432	NuSVR	0.0389	ESM(1)	0.1059	Huber	0.1051	PA	0.1460	NuSVR	0.1419
.80	NuSVR	0.0204	NuSVR	0.0201	NuSVR	0.0340	NuSVR	0.0320	ESM(1)	0.0985	NuSVR	0.0961	PA	0.1380	NuSVR	0.1323
.85	EVM(CPI)	0.0156	EVM(CPI)	0.0156	NuSVR	0.0266	NuSVR	0.0262	ESM(1)	0.0910	NuSVR	0.0895	PA	0.1318	NuSVR	0.1259
.90	EVM(CPI)	0.0112	EVM(CPI)	0.0112	EVM(CPI)	0.0191	EVM(CPI)	0.0191	ESM(1)	0.0799	NuSVR	0.0742	ESM(1)	0.1182	NuSVR	0.1072
.95	EVM(CPI)	0.0075	EVM(CPI)	0.0075	EVM(CPI)	0.0143	EVM(CPI)	0.0143	ESM(1)	0.0600	NuSVR	0.0533	ESM(1)	0.0857	NuSVR	0.0764
EVM/ESM	3		3		3		2		12		4		4		2	
ML	16		16		16		17		7		15		15		17	
DR	10				1				3				8			
IR	9				16				12				9			

Table 8
Project-level best EAC models and corresponding regression metrics.

Code	Cost								Duration							
	MAE				$RMSE$				MAE				$RMSE$			
	DR		IR		DR		IR		DR		IR		DR		IR	
	Model	Value	Model	Value	Model	Value	Model	Value	Model	Value	Model	Value	Model	Value	Model	Value
C2011–10	SGD	0.0023	Tweedie	0.0063	SGD	0.0027	Tweedie	0.0096	ESM(1)	0.0335	Tweedie	0.0307	ESM(1)	0.0436	Tweedie	0.0428
C2011–12	SGD	0.0040	RANSAC	0.0073	SGD	0.0047	RANSAC	0.0095	SGD1CSVM	0.0000	AdaBoost	0.0795	SGD1CSVM	0.0000	AdaBoost	0.0970
C2011–13	SVR	0.0608	Huber	0.0242	SVR	0.0684	Huber	0.0319	HGB	0.0118	k-NN	0.0662	HGB	0.0119	XGB RF	0.0901
C2012–13	Tweedie	0.0026	k-NN	0.0086	XGB RF	0.0032	k-NN	0.0099	HGB	0.0178	k-NN	0.0415	HGB	0.0203	k-NN	0.0463
C2012–17	EVM(CPI)	0.0543	EVM(CPI)	0.0543	EVM(CPI)	0.0617	EVM(CPI)	0.0617	ESM(SPI^t)	0.0760	ESM(SPI^t)	0.0760	ESM(SPI^t)	0.1001	ESM(SPI^t)	0.1001
C2013–01	Huber	0.1031	Huber	0.0975	PA	0.1361	k-NN	0.1294	SGD1CSVM	0.0000	AdaBoost	0.0093	SGD1CSVM	0.0000	AdaBoost	0.0142
C2013–02	EVM(CPI)	0.0150	EVM(CPI)	0.0150	EVM(CPI)	0.0224	EVM(CPI)	0.0224	SGD1CSVM	0.0000	AdaBoost	0.0446	SGD1CSVM	0.0000	AdaBoost	0.0518
C2013–03	EN	0.0133	k-NN	0.0246	EN	0.0133	k-NN	0.0287	SGD1CSVM	0.0000	AdaBoost	0.0450	SGD1CSVM	0.0000	AdaBoost	0.0580
C2013–04	EVM(CPI)	0.0751	EVM(CPI)	0.0751	EVM(CPI)	0.0929	EVM(CPI)	0.0929	EVM(SPI)	0.2121	EVM(SPI)	0.2121	EVM(SPI)	0.2503	EVM(SPI)	0.2503
C2013–07	Huber	0.0049	Tweedie	0.0049	Huber	0.0073	Tweedie	0.0065	SGD	0.0220	Tweedie	0.0306	SGD	0.0270	SGD	0.0392
C2013–08	k-NN	0.0986	SVR	0.1032	AdaBoost	0.1000	NuSVR	0.1042	EN	0.1119	Huber	0.1734	EN	0.1119	NuSVR	0.1843
C2013–09	SVR	0.0337	EVM(CPI)	0.0418	SVR	0.0383	SVR	0.0481	EN	0.0609	Huber	0.1607	EN	0.0609	NuSVR	0.1646
C2013–12	PA	0.0126	Huber	0.0150	PA	0.0149	Huber	0.0176	EVM(SPI)	0.2364	EVM(SPI)	0.2364	EVM(SPI)	0.2892	EVM(SPI)	0.2892
C2014–04	EN	0.0054	SVR	0.0193	EN	0.0054	SVR	0.0204	AdaBoost	0.2663	EVM(SPI)	0.2835	AdaBoost	0.2839	EVM(SPI)	0.3204
C2014–05	Huber	0.0107	GB	0.0133	Huber	0.0136	GB	0.0179	EN	0.0087	RANSAC	0.0233	EN	0.0087	RANSAC	0.0269
C2014–06	HGB	0.0011	DT	0.0060	HGB	0.0016	DT	0.0067	ERTs	0.0144	RANSAC	0.0260	ERTs	0.0168	RANSAC	0.0302
C2014–07	NuSVR	0.0392	NuSVR	0.0340	SVR	0.0465	NuSVR	0.0459	MLP	0.0113	k-NN	0.0379	MLP	0.0137	k-NN	0.0407
C2014–08	EVM(CPI)	0.0441	EVM(CPI)	0.0441	EVM(CPI)	0.0662	EVM(CPI)	0.0662	XGB RF	0.0087	NuSVR	0.0239	XGB RF	0.0087	NuSVR	0.0279
C2015–01	MLP	0.0074	HGB	0.0190	MLP	0.0093	HGB	0.0252	OLS	0.0903	MLP	0.0935	AdaBoost	0.1334	MLP	0.1327
C2015–02	SGD1CSVM	0.1367	NuSVR	0.1609	SGD1CSVM	0.1367	NuSVR	0.1714	XGB	0.0322	SVR	0.2517	XGB	0.0334	ESM(1)	0.2738
C2015–03	RANSAC	0.0414	RANSAC	0.0333	RANSAC	0.0578	RANSAC	0.0428	ERTs	0.0163	XGB	0.0549	ERTs	0.0183	XGB	0.0676
C2015–05	DT	0.0013	DT	0.0059	DT	0.0013	XGB	0.0070	EN	0.0609	ESM(SPI^t)	0.1233	EN	0.0609	ESM(SPI^t)	0.1319
C2015–08	SGD1CSVM	0.0115	k-NN	0.0182	SGD1CSVM	0.0115	k-NN	0.0205	SGD1CSVM	0.0000	GP	0.0166	SGD1CSVM	0.0000	SGD1CSVM	0.0219
C2015–27	MLP	0.0331	Tweedie	0.0524	MLP	0.0376	Tweedie	0.0585	EN	0.0099	GB	0.0912	EN	0.0099	Huber	0.1030
C2015–29	EVM(CPI)	0.0014	EVM(CPI)	0.0014	EVM(CPI)	0.0024	Tweedie	0.0020	SGD1CSVM	0.0000	SGD1CSVM	0.0009	SGD1CSVM	0.0000	SGD1CSVM	0.0015
C2015–30	SGD1CSVM	0.0000	MLP	0.0036	SGD1CSVM	0.0000	MLP	0.0041	SGD1CSVM	0.0000	SGD1CSVM	0.0819	SGD1CSVM	0.0000	MLP	0.0918
C2015–31	SGD1CSVM	0.0218	k-NN	0.0082	SGD1CSVM	0.0218	k-NN	0.0092	MLP	0.0624	DT	0.1254	MLP	0.0956	Huber	0.1424
C2015–33	Tweedie	0.0021	MLP	0.0549	XGB RF	0.0033	MLP	0.0655	ESM(SPI^t)	0.1611	ESM(SPI^t)	0.1611	ESM(SPI^t)	0.1692	ESM(SPI^t)	0.1692
C2015–34	NuSVR	0.1243	SVR	0.1123	SGD1CSVM	0.1387	NuSVR	0.1385	ARD	0.1739	k-NN	0.1859	Lars	0.2259	k-NN	0.2157
C2015–35	EN	0.0299	HGB	0.0611	EN	0.0299	HGB	0.0677	ESM(SPI^t)	0.0173	SGD	0.0098	ESM(SPI^t)	0.0212	SGD	0.0119
C2016–01	Tweedie	0.0029	SGD1CSVM	0.0087	XGB RF	0.0031	HGB	0.0159	MLP	0.0237	NuSVR	0.0317	EN	0.0241	NuSVR	0.0406
C2016–02	RANSAC	0.0045	MLP	0.0030	RANSAC	0.0061	MLP	0.0036	SGD	0.0311	PA	0.0308	SGD	0.0371	PA	0.0409
C2016–03	EVM(CPI)	0.0145	k-NN	0.0071	EVM(CPI)	0.0149	k-NN	0.0092	NuSVR	0.0105	DT	0.0272	NuSVR	0.0165	DT	0.0315
C2016–07	EVM(CPI)	0.0026	MLP	0.0017	EVM(CPI)	0.0027	MLP	0.0020	AdaBoost	0.1188	DT	0.1415	AdaBoost	0.1378	DT	0.1514
C2016–11	SGD1CSVM	0.0044	MLP	0.0026	SGD1CSVM	0.0044	MLP	0.0034	SGD1CSVM	0.0000	MLP	0.0444	SGD1CSVM	0.0000	MLP	0.0532
C2016–12	SGD	0.0009	Tweedie	0.0037	SGD	0.0010	MLP	0.0051	SGD1CSVM	0.0000	SGD1CSVM	0.0025	SGD1CSVM	0.0000	EVM(SPI)	0.0058
C2016–13	HGB	0.0014	GB	0.0100	HGB	0.0017	DT	0.0127	Tweedie	0.0604	EVM(SPI)	0.0801	Tweedie	0.0608	EVM(SPI)	0.0909
C2016–14	SGD	0.0007	PA	0.0032	SGD	0.0011	Tweedie	0.0046	SGD1CSVM	0.0000	AdaBoost	0.0129	SGD1CSVM	0.0000	AdaBoost	0.0231
C2016–15	PA	0.0057	SVR	0.0109	PA	0.0068	SVR	0.0132	SGD1CSVM	0.0000	SGD1CSVM	0.0151	SGD1CSVM	0.0000	SGD1CSVM	0.0333
C2016–27	EVM(CPI)	0.0168	EVM(CPI)	0.0168	EVM(CPI)	0.0221	SVR	0.0205	OLS	0.1103	Huber	0.1080	OLS	0.1197	Huber	0.1206
C2016–28	HGB	0.0008	NuSVR	0.0049	HGB	0.0009	SVR	0.0057	SGD1CSVM	0.0000	MLP	0.0631	SGD1CSVM	0.0000	MLP	0.0910
C2016–29	HGB	0.0041	NuSVR	0.0048	XGB	0.0044	NuSVR	0.0054	SGD1CSVM	0.0000	SGD1CSVM	0.1245	SGD1CSVM	0.0000	ESM(1)	0.1358
C2016–30	ERTs	0.0023	HGB	0.0077	ERTs	0.0023	HGB	0.0087	AdaBoost	0.0884	k-NN	0.0811	AdaBoost	0.1002	k-NN	0.0923
C2016–31	SGD	0.0019	MLP	0.0065	SGD	0.0026	MLP	0.0068	Tweedie	0.1443	Huber	0.1606	Tweedie	0.1445	Huber	0.1704
C2016–32	EVM(CPI)	0.0020	EVM(CPI)	0.0020	EVM(CPI)	0.0023	EVM(CPI)	0.0023	SGD1CSVM	0.0000	ESM(1)	0.1154	SGD1CSVM	0.0000	ESM(1)	0.1304
C2016–33	EN	0.0015	SVR	0.0052	EN	0.0015	SVR	0.0061	PA	0.0414	k-NN	0.0554	PA	0.0562	k-NN	0.0701
C2016–34	Ridge	0.0050	MLP	0.0055	Ridge	0.0061	MLP	0.0090	SGD1CSVM	0.0000	ESM(1)	0.1298	SGD1CSVM	0.0000	ESM(1)	0.1610
C2018–10	EVM(CPI)	0.0000	EVM(CPI)	0.0000	EVM(CPI)	0.0000	EVM(CPI)	0.0000	EN	0.0099	SVR	0.0865	EN	0.0099	SVR	0.0926
C2019–01	DT	0.0023	ERTs	0.0026	DT	0.0023	HGB	0.0032	EN	0.0609	DT	0.1031	EN	0.0609	DT	0.1157
C2019–02	SGD	0.0033	GB	0.0037	SGD	0.0035	GB	0.0055	SGD1CSVM	0.0000	SGD1CSVM	0.0229	SGD1CSVM	0.0000	MLP	0.0375
EVM/ESM	10		9		10		6		6		9		6		12	
ML	40		41		40		44		44		41		44		38	
DR	31				31				41				41			
IR	11				13				5				6			

Overall, Table 8 shows that no single model consistently dominates across all projects. This is consistent across all target-metric-method scenarios. Comparing ML techniques and EVM/ESM models, ML techniques outperform EVM and ESM models across all target-metric-method scenarios. Comparing DR and IR methods, DR outperforms IR across all target-metric scenarios; however, considering the results from Tables 6–7, the difference in performance between DR and IR variants of the same ML method is limited.

5. Discussion

This section discusses the main findings of the study, explores their theoretical and practical implications, and reflects on how they compare to existing research.

5.1. Main and secondary results

The analysis of the results obtained by applying the ML pipeline to a dataset of 50 real-world construction projects reveals several key findings. First, the pipeline can build EAC models that match or exceed the performance of standard EVM and ESM models in terms of accuracy (measured by MAE), precision (measured by $RMSE$), and timeliness (assessed via MAE and $RMSE$ across different progress stages). Second, IR models consistently outperform their DR counterparts, highlighting the advantage of using a dynamic feature as the target variable rather than a static one. These findings hold for both forecasting targets — cost and duration — and are consistent across dataset-level (Table 6), progress-stage-level (Table 7), and individual project-level analyses (Table 8).

5.2. Theoretical implications

On the theoretical level, this study validates the effectiveness of the procedures implemented within the proposed ML pipeline in reducing underfitting and preventing overfitting.

From a numerical perspective, the performance comparison at the overall dataset level (Table 6) shows a relatively narrow gap between the best ML models and the best-performing EVM and ESM models. This is due to the strict LOGO CV procedures nested within the pipeline, as well as the limited size of the dataset used. However, the performance gap widens at the progress stage level (Table 7) and individual project level (Table 8), where ML model performance varies significantly across projects. These findings underscore that no ML technique should be dismissed prematurely; rather, multiple ML techniques should be tested.

Focusing on the comparison between DR and IR models, IR models consistently outperform DR models across all progress stages. This gap can be attributed to two main factors. First, IR models predict stage-specific performance factors (cPF, sPF), which provide the learning algorithm of the ML technique in use with multiple intermediate targets. In contrast, DR models are trained to predict a single, static EAC value for each project. Second, in IR, cPF and sPF are dimensionless ratios characterized by narrow and relatively stable variance, while the scaled cost and duration targets used in DR models vary widely across projects. This smoother target surface facilitates convergence for gradient-based optimizers, thereby reducing forecast error.

Compared to prior studies, the findings confirm that cost forecasts are generally more accurate than duration forecasts, and that ML techniques can outperform standard EVM and ESM models. However, the performance gap between ML and EVA models is narrower than reported in earlier studies. This difference can be attributed to the use of real-world project data and the nesting of LOGO CV procedures, which produce more conservative and realistic performance assessments. In contrast to earlier work, this study reveals that complex ML models do not consistently imply superior performance—likely due to their greater susceptibility to overfitting and the added complexity of HPT.

The study tested the pipeline on 50 construction projects to evaluate whether ML techniques can detect and leverage the relationship between cost and schedule performance—an especially relevant dynamic in construction, where these variables are often tightly interdependent. However, the pipeline is designed to be applicable across other industries, provided that monitoring data are available. It is based on standard EVA metrics and progress-based features, which are commonly used in various project environments. Rather than aiming to identify a single best model, the findings highlight the importance of selecting models that demonstrate consistent performance across different projects and progress stages.

5.3. Practical implications

From a practical standpoint, this study presents a clear and replicable procedure for developing EAC models using ML techniques and rigorously evaluating their effectiveness. The proposed pipeline is fully automated and requires no manual input once configured, making it well-suited for integration into routine project control workflows. At the same time, it offers practitioners substantial flexibility: users can customize key components such as the interpolation method and progress step size, the feature selection threshold, and the hyperparameter tuning strategy to fit their specific data and project context. The use of LOGO CV ensures that these customizations do not result in overfitting, thereby preserving the generalizability of the models. Once deployed, the pipeline enables project teams to identify the most effective ML models for their datasets, generate reliable EAC forecasts, and integrate these with outputs from standard EVM and ESM models.

6. Conclusions

This paper addressed key challenges in developing robust ML models for project performance forecasting and proposed a structured pipeline to mitigate underfitting and overfitting. Compared to previous literature, the proposed pipeline implements the following procedures:

- Data scaling: using cost and time metrics on a relative scale, prevents projects with metrics of different magnitudes from biasing model training.
- Augmentation through interpolation: using a fixed number of synthetic records at equidistant progress intervals allows training the models on records describing projects throughout all their phases and prevents projects with more (or fewer) records from having higher (or lower) weight on model training. Training the models using records that describe all stages of project execution ensures that the models will be effective at each stage of the new projects to which they are applied.
- LOGO CV: replicating P times the CV split, where P is equal to the number of projects in the dataset, prevents records of the same project from being divided between the training and validation sets. All other non-LOGO procedures involve Train-Test splits that depend on the seed used to assign project records to the two sets, which does not guarantee the reproducibility of the study nor the ability to analyze the difficulty of making forecasts at the individual project level.
- IR: setting the performance factor as the target feature and using it in a predetermined formula to calculate the EACs allows ML techniques to better capture the relationships that determine the cost and schedule performance factors. This is because while cost and project duration are fixed values in all records belonging to the same project, cost and schedule performance factors vary from record to record.

Overfitting and underfitting were addressed theoretically throughout the design of the pipeline. Underfitting was assessed by comparing the performance of ML models against EVM and ESM models. Lower MAE and $RMSE$ values in ML models suggest that relevant patterns were captured, suggesting no underfitting. Although overfitting is typically evaluated by comparing training and validation performance, this was not feasible given the extensive configuration—which included nested LOGO CV, 30 models, 50 projects, and numerous hyperparameter combinations. To mitigate overfitting, the pipeline employed nested validation to prevent data leakage and implemented data balancing to maintain representative training distributions.

The pipeline demonstrated substantial improvements over standard EVM and ESM models. At the dataset level (Table 6), the best-performing IR models reduced cost forecast errors by 0.27 in MAE and 0.37 in $RMSE$. For duration forecasts, improvements were even greater, with reductions of 46% in MAE and 68% in $RMSE$. Compared to constant-factor baselines — EVM(1) and ESM(1) — the best IR models still achieved error reductions of approximately 7% for cost and 9% for duration. In head-to-head comparisons, IR variants of the same ML models outperformed their DR counterparts, improving cost forecasts by up to 39% in MAE and 30% in $RMSE$, and duration forecasts by up to 33% in MAE and 35% in $RMSE$.

Limitations of this study mostly correspond to the dataset and hyperparameters values used, the parameters chosen for the CV methods, and the use of synthetic data.

Different datasets or hyperparameters could have yielded different outcomes. However, the focus of the study was not to identify the best-performing ML model but rather to compare the performance of all ML models developed through the pipeline against the EVM and ESM methodologies.

A second problem related to the dataset is data availability. When the sample of projects to be analyzed is limited, the likelihood of ML models overfitting the records of the few available projects increases. To address this issue, additional procedures could be implemented, such as adding perturbation to augmentation, using L1 or L2 regularization.

Another factor influencing the ML models' performance are the regression metrics used within the LOGO CV procedures, which determine whether the models prioritize mean forecasts (MAE) vs. mitigating outliers ($RMSE$).

The use of synthetic data and LOGO CV may raise concerns about scalability and data distortion. However, computational cost is unlikely to be a practical issue as the pipeline only needs to be rerun when new projects are completed and the number of tracking periods per project is limited compared to common ML applications. Data distortion is also minimized, as synthetic records are generated through dense (i.e., 0.05) linear interpolation between actual data points, ensuring that no new patterns are introduced and that the statistical properties relevant to performance metric calculations remain intact.

Furthermore, there is no objective method to verify the accuracy of interpolated values in real-world projects. Mitigation measures were adopted by excluding projects whose absolute change in PC exceeded a preset threshold; however, this safeguard remains inherently subjective. The choice of which projects to exclude — and therefore which data are deemed potentially distorted — varies with the performance metric examined (e.g., PC, PS, or AC), the change measure applied (e.g., absolute, relative, percentile, z-score), and the arbitrarily selected cut-off value (0.50 in this study), each of which can influence the final results.

ML techniques for performance forecasting offer several avenues for further improvement. The study could include a broader range of algorithms and focus on optimizing the hyperparameters of these models. While the study only considered EVM and ESM and standard metrics and indicators, future research could incorporate other features influencing cost or schedule performance. Regarding feature selection, the study focused exclusively on the fSFS procedure. In contrast, exploring alternative feature selection methods (e.g., backward elimination, regularized feature selection) would be beneficial to identify different subsets of relevant features and improve model performance. Lastly, the IR method was developed based on the EVM and standard formulae and used cPF and sPF as target features. Future research could replicate this procedure but explore different formulae for EAC and EAC^t.

CRediT authorship contribution statement

Filippo Maria Ottaviani: Writing – original draft, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Data curation, Conceptualization. **Pablo Ballesteros-Pérez:** Writing – review & editing, Supervision. **Timur Narbaev:** Writing – review & editing, Supervision.

Declaration of competing interest

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

Acknowledgments

This work was supported by a grant from Project Management Institute, Inc, United States.

Appendix A. EVM and ESM

Let PS denote the project's Percentage Scheduled, PC the Percentage Completed, and AD the Actual Duration. Then, EV is determined as per

$$EV = \text{BAC} \cdot PC, \tag{A.1}$$

and PV is determined as per

$$PV = \text{BAC} \cdot PS, \tag{A.2}$$

with $PV(\text{PD}) = EV(\text{AD}) = \text{BAC}$.

EVM calculates the EAC as the sum of AC and the cost Estimate to Complete (ETC), where the ETC is determined as the ratio of the cost associated with the remaining work ($\text{BAC} - EV$) to a Cost Performance Factor (cPF), resulting in

$$\begin{aligned} EAC[cPF] &= AC + ETC\,[cPF] \\ &= AC + (\text{BAC} - EV)/cPF. \end{aligned} \tag{A.3}$$

Standard cPFs include 1 and the Cost Performance Index (CPI). Using $cPF = 1$ assumes that the current cost variance (CV) will remain constant until project completion, leading to

$$\begin{aligned} EAC\,[1] &= AC + (\text{BAC} - EV)/1 \\ &= AC + \text{BAC} - EV, \end{aligned} \tag{A.4}$$

where

$$CV = EV - AC. \tag{A.5}$$

Alternatively, using CPI, defined as

$$CPI = EV/AC, \tag{A.6}$$

assumes that cost overruns will continue at the current rate until completion, leading to

$$EAC\,[CPI] = \text{BAC}/CPI. \tag{A.7}$$

EVM calculates the EAC^t as the ratio of PD to the EVM Schedule Performance Index (SPI), as per

$$EAC^t = \text{PD}/SPI, \tag{A.8}$$

where

$$SPI = EV/PV. \tag{A.9}$$

EVM also includes the Schedule Variance (SV^t) indicator, defined as

$$SV = EV - PV. \tag{A.10}$$

Table B.9
Hyperparameters grid.

Model	Hyperparameter	Values
OLS	fit_intercept	TRUE
Ridge	alpha	1e–2, 1e–1, 1, 1e1, 1e2
	fit_intercept	TRUE
	max_iter	1e3
	tol	1e–4
	solver	auto
	positive	FALSE
Lasso	alpha	1e–6, 1e–5, 1e–4, 1e–3, 1e–2
	fit_intercept	TRUE
	max_iter	1e3
	tol	1e–4
	positive	FALSE
	selection	cyclic
Elastic Net	alpha	1e–6, 1e–5, 1e–4, 1e–3, 1e–2
	l1_ratio	0, .25, .50, .75, 1
	fit_intercept	TRUE
	max_iter	1e3
	tol	1e–4
	positive	FALSE
Lars	fit_intercept	TRUE
Lasso Lars	alpha	1e–6, 1e–5, 1e–4, 1e–3, 1e–2
	fit_intercept	TRUE
	max_iter	1e3
OMP	fit_intercept	TRUE
	tol	None
Passive Aggressive	C	1e–2, 1e–2, 1, 1e1, 1e2
	fit_intercept	TRUE
	max_iter	1e3
	tol	1e–3
	early_stopping	FALSE
	validation_fraction	1e–1
	n_iter_no_change	5
	loss	epsilon_insensitive
	average	TRUE
Bayesian Ridge, ARD	alpha_1	1e–8, 1e–7, 1e–6, 1e–5, 1e–4
	alpha_2	1e–8, 1e–7, 1e–6, 1e–5, 1e–4
	lambda_1	1e–8, 1e–7, 1e–6, 1e–5, 1e–4
	lambda_2	1e–8, 1e–7, 1e–6, 1e–5, 1e–4
	tol	1e–3
	max_iter	None
Tweedie	power	0, 1, 2, 3
	alpha	1
	fit_intercept	TRUE
	solver	lbfgs
	max_iter	1e2
	tol	1e–4
SGD	l1_ratio	0, .25, .50, .75, 1
	alpha	1e–6, 1e–5, 1e–4, 1e–3, 1e–2
	loss	squared_error
	penalty	l2
	fit_intercept	TRUE
	max_iter	1e3
	tol	1e–3
	epsilon	1e–1
	learning_rate	invscaling
	eta0	1e–2
	power_t	25e–2
	validation_fraction	1e–1
	average	TRUE
SGD1CSVM	nu	0.05, .25, .5, .75, 1
	fit_intercept	TRUE
	max_iter	1000
	tol	1e–3
	learning_rate	optimal
	eta0	0
	power_t	0.25
	average	TRUE
RANSAC	estimator	LinearRegression
	max_trials	100
	stop_probability	0.99
	loss	absolute_error
	max_skips	inf
	stop_n_inliers	inf
	stop_score	inf
Huber	epsilon	1.005, 1.35
	alpha	1e–5, 1e–4, 1e–3
	max_iter	1e3
	fit_intercept	TRUE
	tol	1e–04
Kernel Ridge	alpha	1e–2, 1e–1, 1, 1e1, 1e2
	kernel	linear, rbf, poly
	degree	1, 2, 3
	gamma	None
	coef0	1

(*continued on next page*)

However, since the indicator is expressed in cost units, it is not used for calculating the EAC^t.

ESM calculates the EAC^t as the sum of AT and the time Estimate to Complete (ETC^t), where the ETC^t is determined as the ratio of the duration associated with remaining work (PD – ES) to a Schedule Performance Factor (sPF), resulting in

$$EAC^t\,[sPF] = AT + ETC^t\,[sPF] = AT + (\mathrm{PD} - ES)/sPF, \tag{A.11}$$

where ES denotes the Earned Schedule metric, determined as per

$$ES = AT_i + (EV - PV_i)/(PV_{i+1} - PV_i) \cdot (AT_{i+1} - AT_i), \tag{A.12}$$

with $PV_i \leq EV \leq PV_{i+1}$. Standard sPFs include 1 and SPI^t. Using $sPF = 1$ assumes that the current Earned Schedule variance (SV^t) will remain constant until project completion, leading to

$$EAC^t\,[1] = AT + \mathrm{PD} - ES, \tag{A.13}$$

Table B.9 (*continued*).

Model	Hyperparameter	Values
SVR	C	1e–3, 1e–2, 1e–1, 1
	epsilon	1e–3, 1e–2, 1e–1
	gamma	scale, auto
	kernel	rbf
	tol	1e–1
	coef0	0
	tol	1e–2
NuSVR	nu	.50, .25, .50, .75, 1
	C	1e–3, 1e–2, 1e–1, 1, 1e1
	gamma	scale, auto
	kernel	rbf
	tol	1e–2
	max_iter	–1
k-NN	n_neighbors	1, 3, 5, 7, 10, 20, 50
	weights	uniform, distance
	leaf_size	10, 30, 50
	p	1, 2, 3
	algorithm	auto
	metric	minkowski
	metric_params	None
Gaussian Process	kernel	None
	alpha	1e–10
	optimizer	fmin_l_bfgs_b
	n_restarts_optimizer	0
	n_targets	None
	normalize_y	TRUE
DT, ERT	max_depth	1, 3, 5
	splitter	best, random
	criterion	absolute_error
	min_samples_split	2
	min_samples_leaf	1
	min_weight_fraction_leaf	0
	max_features	None
	max_leaf_nodes	None
	min_impurity_decrease	0
	ccp_alpha	0
RF, ERTs	max_depth	1, 3, 5
	n_estimators	1e1
	criterion	absolute_error
	min_samples_split	2
	min_samples_leaf	1
	min_weight_fraction_leaf	0
	max_features	None
	max_leaf_nodes	None
	min_impurity_decrease	0
	ccp_alpha	0
	boostrap	FALSE
	warm_start	FALSE
	max_samples	None
AdaBoost	learning_rate	1e–2, 1e–1, 1, 1e1
	estimator	None
	n_estimators	1e1
	loss	linear
GB	learning_rate	1e–2, 1e–1, 1, 1e1
	max_depth	1, 3, 5
	loss	absolute_error
	n_estimators	1e1
	criterion	friedman_mse
	min_samples_split	2
	min_samples_leaf	1
	min_weight_fraction_leaf	0
	max_features	None
	init	None
	tol	1e–4
	validation_fraction	1e–1
	alpha	9e–1
	warm_start	FALSE
	ccp_alpha	0
HGB	learning_rate	1e–3, 1e–2, 1e–1, 1
	max_depth	1, 3, 5
	loss	absolute_error
	quantile	None
	max_iter	1
	max_leaf_nodes	None
	min_samples_leaf	1
	max_bins	255
	early_stopping	auto
	scoring	loss
	validation_fraction	1e–1
	l2_regularization	0
	warm_start	FALSE
	early_stopping	auto
	n_iter_no_change	1e1
	tol	1e–4
XGB	learning_rate	1e–3, 1e–2
	max_depth	1, 3, 5
	objective	reg:absoluteerror
XGB RF	learning_rate	1e–3, 1e–2
	max_depth	1, 3, 5
	objective	reg:absoluteerror
MLP	hidden_layer_sizes	(2,), (4,), (6,)
	activation	identity, logistic, tanh, relu
	solver	adam, lbfgs
	alpha	1e–4
	batch_size	auto
	learning_rate	adaptive
	tol	1e–4
	warm_start	None
	early_stopping	TRUE
	validation_fraction	1e–1
	epsilon	1e–1
	n_iter_no_change	1
	beta_1	9e–1
	beta_2	999e–3

where

$$SV^t = ES - AT. \tag{A.14}$$

Alternatively, using SPI^t, defined as

$$SPI^t = ES/AT, \tag{A.15}$$

assumes that schedule overruns will continue at the current rate until completion, leading to

$$EAC^t\left[SPI^t\right] = \mathrm{PD}/SPI^t. \tag{A.16}$$

Appendix B. Hyperparameters grid

See Table B.9.

Data availability

Data will be made available on request.

References

[1] S. Changali, A. Mohammad, M. van Nieuwland, The construction productivity imperative, 2015, https://www.mckinsey.com/capabilities/operations/our-insights/the-construction-productivity-imperative. (Accessed 14 July 2025).

[2] P. Ballesteros-Pérez, E. Sanz-Ablanedo, R. Soetanto, M.C. González-Cruz, G. Larsen, A. Cerezo-Narváez, Duration and cost variability of construction activities: An empirical study, J. Constr. Eng. Manag. 146 (1) (2020) 04019093, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0001739.

[3] J.M. Davila Delgado, L. Oyedele, M. Bilal, A. Ajayi, L. Akanbi, O. Akinade, Big data analytics system for costing power transmission projects, J. Constr. Eng. Manag. 146 (1) (2020) 05019017, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0001745.

[4] P. Ballesteros-Pérez, G.D. Larsen, M.C. González-Cruz, Do projects really end late? On the shortcomings of the classical scheduling techniques, J. Technol. Sci. Educ. 8 (1) (2018) 17–33, http://dx.doi.org/10.3926/jotse.303.

[5] A. Cerezo-Narváez, A. Pastor-Fernández, M. Otero-Mateo, P. Ballesteros-Pérez, Integration of cost and work breakdown structures in the management of construction projects, Appl. Sci. 10 (4) (2020) 1386, http://dx.doi.org/10.3390/app10041386,

[6] D. Hillson, How risky is your project—And what are you doing about it?, 2014, URL: https://www.pmi.org/learning/library/risky-project-doing-it-9351, Presented at PMI® Global Congress 2014 – North America, Phoenix, AZ.

[7] M. Vanhoucke, The Illusion of Control: Project Data, Computer Algorithms and Human Intuition for Project Management and Control, first ed., Springer, ISBN: 978-3-031-31784-2, 2023, p. 330, http://dx.doi.org/10.1007/978-3-031-31785-9.

[8] F.M. Ottaviani, A. De Marco, C. Rafele, G. Castelblanco, Risk perception-based project contingency management framework, Systems 12 (3) (2024) 93, http://dx.doi.org/10.3390/systems12030093.

[9] T.V. Fridgeirsson, H.T. Ingason, H.I. Jonasson, H. Jonsdottir, An authoritative study on the near future effect of artificial intelligence on project management knowledge areas, Sustainability 13 (4) (2021) 2345, http://dx.doi.org/10.3390/su13042345.

[10] PMI, Practice Standard for Earned Value Management, second ed., Project Management Institute, Newton Square, PA, ISBN: 1935589350, 2012, p. 135.

[11] W. Lipke, Schedule is different, Meas. News (2003) 31–34, URL: https://www.earnedschedule.com/docs/schedule%20is%20different.pdf, Published by the College of Performance Management.

[12] I. Karaca, D.D. Gransberg, H.D. Jeong, Improving the accuracy of early cost estimates on transportation infrastructure projects, J. Manag. Eng. 36 (5) (2020) 04020063, http://dx.doi.org/10.1061/(ASCE)ME.1943-5479.0000819.

[13] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second ed., Springer New York, NY, ISBN: 978-0-387-84857-0, 2009, http://dx.doi.org/10.1007/978-0-387-84858-7.

[14] M. Kuhn, K. Johnson, Applied Predictive Modeling, Springer New York, New York, NY, ISBN: 978-1-4614-6848-6, 2013, http://dx.doi.org/10.1007/978-1-4614-6849-3.

[15] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, first ed., MIT Press, ISBN: 978-0262035613, 2016.

[16] O. Zwikael, S. Globerson, T. Raz, Evaluation of models for forecasting the final cost of a project, Proj. Manag. J. 31 (1) (2000) 53–57, http://dx.doi.org/10.1177/875697280003100108.

[17] A. Barrientos-Orellana, P. Ballesteros-Pérez, D. Mora-Melià, A. Cerezo-Narváez, J.H. Gutiérrez-Bahamondes, Comparison of the stability and accuracy of deterministic project cost prediction methods in earned value management, Buildings 13 (5) (2023) 1206, http://dx.doi.org/10.3390/buildings13051206.

[18] P.A. de Andrade, A. Martens, M. Vanhoucke, Using real project schedule data to compare earned schedule and earned duration management project time forecasting capabilities, Autom. Constr. 99 (2019) 68–78, http://dx.doi.org/10.1016/j.autcon.2018.11.030.

[19] F.M. Ottaviani, A. De Marco, T. Narbaev, M. Rebuglio, Improving project estimates at completion through progress-based performance factors, Buildings 14 (3) (2024) 643, http://dx.doi.org/10.3390/buildings14030643.

[20] A. De Marco, T. Narbaev, F.M. Ottaviani, M. Vanhoucke, Influence of cost contingency management on project estimates at completion, Int. J. Constr. Manag. 24 (9) (2024) 935–945, http://dx.doi.org/10.1080/15623599.2023.2239487.

[21] H. Khamooshi, H. Golafshani, EDM: Earned duration management, a new approach to schedule performance management and measurement, Int. J. Proj. Manage. 32 (6) (2014) 1019–1041, http://dx.doi.org/10.1016/j.ijproman.2013.11.002.

[22] A. De Marco, F.M. Ottaviani, F. Bolognesi, Time series-based project cost forecasting framework, Procedia Comput. Sci. 239 (2024) 105–113, http://dx.doi.org/10.1016/j.procs.2024.06.152.

[23] J. Batselier, M. Vanhoucke, Improving project forecast accuracy by integrating earned value management with exponential smoothing and reference class forecasting, Int. J. Proj. Manage. 35 (1) (2017) 28–43, http://dx.doi.org/10.1016/j.ijproman.2016.10.003.

[24] A. Martens, M. Vanhoucke, Integrating corrective actions in project time forecasting using exponential smoothing, J. Manag. Eng. 36 (5) (2020) 04020044, http://dx.doi.org/10.1061/(ASCE)ME.1943-5479.0000806.

[25] J. Batselier, M. Vanhoucke, Empirical evaluation of earned value management forecasting accuracy for time and cost, J. Constr. Eng. Manag. 141 (11) (2015) 05015010, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0001008.

[26] J.H. Kim, Multicollinearity and misleading statistical results, Korean J. Anesthesiol. 72 (6) (2019) 558–569, http://dx.doi.org/10.4097/kja.19087.

[27] F. Anbari, Earned value project management method and extensions, IEEE Eng. Manag. Rev. 32 (3) (2004) 97–110, http://dx.doi.org/10.1109/EMR.2004.25113.

[28] J. Batselier, M. Vanhoucke, Project regularity: Development and evaluation of a new project characteristic, J. Syst. Sci. Syst. Eng. 26 (1) (2017) 100–120, http://dx.doi.org/10.1007/s11518-016-5312-6.

[29] M. Zhao, X. Zi, Using earned value management with exponential smoothing technique to forecast project cost, J. Phys.: Conf. Ser. 1955 (1) (2021) 012101, http://dx.doi.org/10.1088/1742-6596/1955/1/012101.

[30] B.C. Kim, Y.H. Kwak, Improving the accuracy and operational predictability of project cost forecasts: an adaptive combination approach, Prod. Plan. Control 29 (9) (2018) 743–760, http://dx.doi.org/10.1080/09537287.2018.1467511.

[31] Y. Cao, B. Ashuri, Predicting the volatility of highway construction cost index using long short-term memory, J. Manag. Eng. 36 (4) (2020) 04020020, http://dx.doi.org/10.1061/(ASCE)ME.1943-5479.0000784.

[32] S. Kim, C.-Y. Choi, M. Shahandashti, K.R. Ryu, Improving accuracy in predicting city-level construction cost indices by combining linear ARIMA and nonlinear ANNs, J. Manag. Eng. 38 (2) (2022) 04021093, http://dx.doi.org/10.1061/(ASCE)ME.1943-5479.0001008.

[33] B.-C. Kim, S.J. Kim, Credibility evaluation of project duration forecast using forecast sensitivity and forecast-risk compatibility, J. Constr. Eng. Manag. 141 (8) (2015) 04015023, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0001000.

[34] R.D. Warburton, D.F. Cioffi, Estimating a project's earned and final duration, Int. J. Proj. Manage. 34 (8) (2016) 1493–1504, http://dx.doi.org/10.1016/j.ijproman.2016.08.007.

[35] T. Narbaev, A. De Marco, An earned schedule-based regression model to improve cost estimate at completion, Int. J. Proj. Manage. 32 (6) (2014) 1007–1018, http://dx.doi.org/10.1016/j.ijproman.2013.12.005.

[36] A. De Marco, M. Rosso, T. Narbaev, Nonlinear cost estimates at completion adjusted with risk contingency, J. Mod. Proj. Manag. 4 (2) (2016) 24–33, http://dx.doi.org/10.19255/JMPM01102.

[37] R.D. Warburton, A. De Marco, F. Sciuto, Earned schedule formulation using nonlinear cost estimates at completion, J. Mod. Proj. Manag. 5 (1) (2017) 75–81, http://dx.doi.org/10.19255/JMPM01307.

[38] Q.T. Huynh, T.A. Le, T.H. Nguyen, N.H. Nguyen, D.H. Nguyen, A method for improvement the parameter estimation of non-linear regression in growth model to predict project cost at completion, in: Proceedings - 2020 RIVF International Conference on Computing and Communication Technologies, RIVF 2020, 2020, http://dx.doi.org/10.1109/RIVF48685.2020.9140765.

[39] R.D. Warburton, F.M. Ottaviani, A. De Marco, Critical analysis of linear and nonlinear project duration forecasting methods, J. Mod. Proj. Manag. 11 (1) (2023) 187–199, http://dx.doi.org/10.19255/JMPM03113.

[40] T. Narbaev, A. De Marco, Earned value and cost contingency management: A framework model for risk adjusted cost forecasting, J. Mod. Proj. Manag. 4 (3) (2017) 12–19, http://dx.doi.org/10.19225/JMPM01202.

[41] B. Zafari, J. Kettunen, Bayesian methods in project management, in: Wiley StatsRef: Statistics Reference Online, Wiley, 2017, pp. 1–5, http://dx.doi.org/10.1002/9781118445112.stat07900.

[42] B.-C. Kim, K.F. Reinschmidt, Probabilistic forecasting of project duration using Bayesian inference and the beta distribution, J. Constr. Eng. Manag. 135 (3) (2009) 178–186, http://dx.doi.org/10.1061/(ASCE)0733-9364(2009)135:3(178).

[43] A. Firouzi, M. Khayyati, Bayesian updating of copula-based probabilistic project-duration model, J. Constr. Eng. Manag. 146 (5) (2020) 04020046, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0001822.

[44] F. Caron, F. Ruggeri, B. Pierini, A Bayesian approach to improving estimate to complete, Int. J. Proj. Manage. 34 (8) (2016) 1687–1702, http://dx.doi.org/10.1016/j.ijproman.2016.09.007.

[45] K. Mostafa, T. Hegazy, Potential of Bayesian networks for forecasting the ripple effect of progress events, in: Proceedings of the Canadian Society for Civil Engineering Annual Conference 2019, 2019-June, 2019, pp. 1–9, URL: https://legacy.csce.ca/elf/apps/CONFERENCEVIEWER/conferences/2019/pdfs/PaperPDFversion_74_0604104637.pdf.

[46] Y. Huang, Q. Shi, J. Zuo, F. Pena-Mora, J. Chen, Research status and challenges of data-driven construction project management in the big data context, Adv. Civ. Eng. 2021 (2021) 674980, http://dx.doi.org/10.1155/2021/6674980.

[47] I. Taboada, A. Daneshpajouh, N. Toledo, T. de Vass, Artificial intelligence enabled project management: A systematic literature review, Appl. Sci. 13 (8) (2023) 5014, http://dx.doi.org/10.3390/app13085014.

[48] S.H. Iranmanesh, M. Zarezadeh, Application of artificial neural network to forecast actual cost of a project to improve earned value management system, Int. J. Educ. Pedagog. Sci. 2 (6) (2008) 658–661, URL: https://publications.waset.org/13246.pdf.

[49] S.H. Iranmanesh, Z. Mokhtari, Application of data mining tools to predicate completion time of a project, Int. J. Comput. Syst. Eng. 2 (6) (2008) 652–657, URL: https://publications.waset.org/2412.pdf.

[50] S.H. Iranmanesh, G.H. Mirseraji, S. Shahmiri, An emotional learning based fuzzy inference system (ELFIS) for improvement of the completion time of projects estimation, in: 2009 International Conference on Computers & Industrial Engineering, 2009, pp. 470–475, http://dx.doi.org/10.1109/ICCIE.2009.5223748.

[51] H. Chin-Chi, C. Min-Yuan, Estimate at completion for construction projects using evolutionary Gaussian process inference model, in: 2011 International Conference on Multimedia Technology, IEEE, 2011, pp. 4414–4417, http://dx.doi.org/10.1109/ICMT.2011.6003217.

[52] M.Y. Cheng, N.D. Hoang, A.F. Roy, Y.W. Wu, A novel time-depended evolutionary fuzzy SVM inference model for estimating construction project at completion, Eng. Appl. Artif. Intell. 25 (4) (2012) 744–752, http://dx.doi.org/10.1016/j.engappai.2011.09.022.

[53] M.R. Feylizadeh, A. Hendalianpour, M. Bagherpour, A fuzzy neural network to estimate at completion costs of construction projects, Int. J. Ind. Eng. Comput. 3 (3) (2012) 477–484, http://dx.doi.org/10.5267/j.ijiec.2011.11.003.

[54] M.T. Hajiali, M.R. Mosavi, K. Shahanaghi, Estimation of project completion time-based on a mixture of expert in an interactive space, Mod. Appl. Sci. 8 (6) (2014) 229–237, http://dx.doi.org/10.5539/mas.v8n6p229.

[55] M. Wauters, M. Vanhoucke, Support vector machine regression for project control forecasting, Autom. Constr. 47 (2014) 92–106, http://dx.doi.org/10.1016/j.autcon.2014.07.014.

[56] M.Y. Cheng, N.D. Hoang, Interval estimation of construction cost at completion using least squares support vector machine, J. Civ. Eng. Manag. 20 (2) (2014) 223–236, http://dx.doi.org/10.3846/13923730.2013.801891.

[57] M. Wauters, M. Vanhoucke, A comparative study of artificial intelligence methods for project duration forecasting, Expert Syst. Appl. 46 (2016) 249–261, http://dx.doi.org/10.1016/j.eswa.2015.10.008.

[58] M. Wauters, M. Vanhoucke, A nearest neighbour extension to project duration forecasting with artificial intelligence, European J. Oper. Res. 259 (3) (2017) 1097–1111, http://dx.doi.org/10.1016/j.ejor.2016.11.018.

[59] M.-Y. Cheng, Y.-H. Chang, D. Korir, Novel approach to estimating schedule to completion in construction projects using sequence and nonsequence learning, J. Constr. Eng. Manag. 145 (11) (2019) 04019072, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0001697.

[60] E.F.T. Al Hares, C. Budayan, Estimation at completion simulation using the potential of soft computing models: Case study of construction engineering projects, Symmetry 11 (2) (2019) 190, http://dx.doi.org/10.3390/sym11020190.

[61] K.R. Kareem Kamoona, C. Budayan, Implementation of genetic algorithm integrated with the deep neural network for estimating at completion simulation, Adv. Civ. Eng. 2019 (1) (2019) 7081073, http://dx.doi.org/10.1155/2019/7081073.

[62] A. Balali, A. Valipour, J. Antucheviciene, J. Šaparauskas, Improving the results of the earned value management technique using artificial neural networks in construction projects, Symmetry 12 (10) (2020) 1745, http://dx.doi.org/10.3390/sym12101745.

[63] I.A. Aidan, D. Al-Jeznawi, F.M. Al-Zwainy, Predicting earned value indexes in residential complexes' construction projects using artificial neural network model, Int. J. Intell. Eng. Syst. 13 (4) (2020) 248–259, http://dx.doi.org/10.22266/IJIES2020.0831.22.

[64] S.J. Mohammed, H.A. Abdel-Khalek, S.M. Hafez, Predicting performance measurement of residential buildings using an artificial neural network, Civ. Eng. J. (Iran) 7 (3) (2021) 461–476, http://dx.doi.org/10.28991/cej-2021-03091666.

[65] S.J. Mohammed, H.A. Abdel-khalek, S.M. Hafez, Predicting performance measurement of residential buildings using machine intelligence techniques (MLR, ANN and SVM), Iran. J. Sci. Technol. - Trans. Civ. Eng. 46 (4) (2022) 3429–3451, http://dx.doi.org/10.1007/s40996-021-00742-4.

[66] S.R. Dastgheib, M.R. Feylizadeh, M. Bagherpour, A. Mahmoudi, Improving estimate at completion (EAC) cost of construction projects using adaptive neuro-fuzzy inference system (ANFIS), Can. J. Civ. Eng. 49 (2) (2022) 222–232, http://dx.doi.org/10.1139/cjce-2020-0399.

[67] J.I. Santos, M. Pereda, V. Ahedo, J.M. Galán, Explainable machine learning for project management control, Comput. Ind. Eng. 180 (2023) 109261, http://dx.doi.org/10.1016/j.cie.2023.109261.

[68] A. Liang, L. Tao, H. Lei, Combined machine-learning and EDM to monitor and predict a complex project with a GERT-type network: A multi-point perspective, Comput. Ind. Eng. 180 (December 2022) (2023) 109256, http://dx.doi.org/10.1016/j.cie.2023.109256.

[69] M.-Y. Cheng, R.R. Khasani, Least square moment balanced machine: A new approach to estimating cost to completion for construction projects, J. Inf. Technol. Constr. 29 (2024) 503–524, http://dx.doi.org/10.36680/j.itcon.2024.023.

[70] T. Narbaev, Ö. Hazir, B. Khamitova, S. Talgat, A machine learning study to improve the reliability of project cost estimates, Int. J. Prod. Res. 62 (12) (2024) 4372–4388, http://dx.doi.org/10.1080/00207543.2023.2262051.

[71] A. Abo Mhady, C. Budayan, A.P. Gurgun, Estimate-at-completion (EAC) prediction using archimedes optimization with adaptive fuzzy and neural networks, Autom. Constr. 166 (2024) 105653, http://dx.doi.org/10.1016/j.autcon.2024.105653.

[72] G. Yalçı n, S. Bayram, H. Çı takoğlu, Evaluation of earned value management-based cost estimation via machine learning, Buildings 14 (12) (2024) 3772, http://dx.doi.org/10.3390/buildings14123772.

[73] A.H. Turkyilmaz, G. Polat, Risk-based completion cost prediction approach in construction projects utilizing machine learning, J. Inf. Technol. Constr. 30 (June 2024) (2025) 375–396, http://dx.doi.org/10.36680/j.itcon.2025.016.

[74] S. He, J. Du, J.Z. Huang, Singular-value decomposition feature-extraction method for cost-performance prediction, J. Comput. Civ. Eng. 31 (5) (2017) 04017043, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000694.

[75] J. Colin, M. Vanhoucke, Setting tolerance limits for statistical project control using earned value management, Omega 49 (2014) 107–122, http://dx.doi.org/10.1016/j.omega.2014.06.001.

[76] M. Akhbari, Project time and cost forecasting using Monte Carlo simulation and artificial neural networks, Int. J. Ind. Eng. Prod. Res. 29 (2) (2018) 231–239, http://dx.doi.org/10.22068/IJIEPR.29.2.231.

[77] Y. Li, L. Liu, Hybrid artificial neural network and statistical model for forecasting project total duration in earned value management, Int. J. Netw. Virtual Organ. 10 (3–4) (2012) 402–413, http://dx.doi.org/10.1504/IJNVO.2012.046460.

[78] J. Murel, E. Kavlakoglu, What is feature engineering?, 2025, https://www.ibm.com/think/topics/feature-engineering. (Accessed 4 May 2025).

[79] P. Ballesteros-Pérez, A. Cerezo-Narváez, M. Otero-Mateo, A. Pastor-Fernández, M. Vanhoucke, Performance comparison of activity sensitivity metrics in schedule risk analysis, Autom. Constr. 106 (2019) 102906, http://dx.doi.org/10.1016/j.autcon.2019.102906.