Machine Learning Pipeline for Project Performance Forecasting

Filippo Maria Ottaviani[1], Alberto De Marco[2], and Pablo Ballesteros-Pérez[3]

[1]Department of Management and Production Engineering, Politecnico di Torino, Turin, 10129 Italy. Email: filippo.ottaviani@polito.it

[2]Department of Management and Production Engineering, Politecnico di Torino, Turin, 10129 Italy. Email: alberto.demarco@polito.it

[3]Departamento de Proyectos de Ingeniería, Universitat Politècnica de València, Valencia, 46022 Spain. Email: pabbalpe@dpi.upv.es

ABSTRACT

Forecasting a project's cost and duration at completion requires assessing the current performance. This process is crucial for taking action and bringing the project back on track when it suffers cost or schedule deviations. Although machine learning (ML) models are a powerful tool for project performance forecasting, their development must be contingent on the characteristics of project monitoring data. Previous studies have prioritized models' accuracy over generalizability, hindering their practical implementation. This study aims to improve project monitoring and control by proposing an ML pipeline for developing project performance forecasting models. Each step of the pipeline addresses a specific issue of the monitoring data to reduce underfitting and overfitting. The study tests the ML pipeline on a dataset of 46 real projects and benchmarks the performance of 27 ML models against the Earned Value Management (EVM) and Earned Schedule (ES) techniques. Results demonstrate the ML models' superiority over EVM and ES, focusing on handling outliers while maintaining accurate mean predictions. Project managers can employ the proposed pipeline to develop enhanced regression models that minimize underfitting and overfitting, thus, ensuring their models' effectiveness and robustness.

PRACTICAL APPLICATIONS

This paper explores the use of machine learning (ML) models to predict whether a running project will meet its cost and contracted duration. Knowing this information is crucial for making informed decisions about project actions. However, developing accurate ML models is not without challenges, and overlooking these issues can lead to unreliable predictions. This study highlights the problems that can affect model performance and identifies past research that has not adequately addressed them. The paper presents practical solutions to overcome these challenges, ensuring the ML models are accurate, precise, and outperform traditional forecasting methods. The proposed solutions involve three key steps: Scaling project variables to address differences in magnitude and simplify relationship inference. Interpolating monitoring observations to enable predictions at any project stage. Calculating revised cost or duration directly or indirectly. By following these steps, practitioners and decision-makers can develop robust ML models that provide valuable insights into project outcomes. These models have significant real-world applications, enabling organizations to make better-informed decisions and improve project management success rates.

INTRODUCTION

Monitoring and control are crucial processes for project success (Demachkieh and Abdul-Malak 2018). In large-scale and complex projects, planning alone can not anticipate all risks that might arise (Rezakhani 2020). Hence, one must monitor project progress and proactively take measures to mitigate potential threats (Kwon and Kang 2019).

The effectiveness of response measures depends on the appraisal of project performance (Caron et al. 2016). Proper performance assessment ensures reliable forecasts of the project cost and duration. These, in turn, determine whether to intervene and to which extent (Iranmanesh et al. 2007; Chen et al. 2016). However, project performance assessment is not an exact science but depends on the many relationships between project constraints (Muriana and Vizzini 2017; Maylor et al. 2023).

Researchers have recognized Earned Value Management (EVM) as the prevailing methodology for project monitoring (Gedi et al. 2018). EVM provides a systematic approach to evaluate the project status by integrating cost, schedule, and scope factors (PMI 2005). This methodology relies on three metrics: Planned Value (PV), Earned Value (EV), and Actual Cost (AC). The PV represents the budgeted cost of the work that should have been completed according to the baseline schedule until a particular tracking period (t). The EV represents the budgeted cost of the work completed until t. The AC represents the "actual cost" of performing such work.

Classical EVM quantifies project cost and schedule performances through two indicators, namely, Cost Performance Index (CPI) and Schedule Performance Index (SPI^{EVM}). The CPI is determined by the ratio of EV to AC, as per Eq. 1:

$$CPI(t) = EV(t)/AC(t). \tag{1}$$

Whereas the SPI^{EVM} is given by the ratio of EV to PV, as per Eq. 2:

$$SPI^{\mathrm{EVM}}(t) = EV(t)/PV(t). \tag{2}$$

An indicator above one indicates cost savings ($CPI > 1$) or progress ahead of schedule ($SPI^{\mathrm{EVM}} > 1$). Conversely, an indicator below one indicates cost overruns ($CPI < 1$) or progress behind schedule ($SPI^{\mathrm{EVM}} < 1$). When the indicator is equal to one, the project is exactly on budget ($CPI = 1$) or on schedule ($SPI^{\mathrm{EVM}} = 1$).

EVM uses the Cost and Schedule Performance Indices as projection factors to evaluate the Cost Estimate at Completion ($cEAC$) and Duration Estimate at Completion ($tEAC^{\mathrm{EVM}}$), respectively. The $cEAC$ is determined by the sum of the AC to date and the Cost Estimate to Complete ($cETC$). The latter is evaluated as the ratio of the budgeted cost of the remaining work (BAC $-$ EV) to CPI, as per Eq. 3,

$$cEAC(t) = AC(t) + cETC(t) = AC(t) + [\mathrm{BAC} - EV(t)]/CPI(t). \tag{3}$$

Instead, the $tEAC^{\mathrm{EVM}}$ is evaluated through the ratio of the project Planned Duration (PD) to SPI^{EVM}, as per Eq. 4:

$$tEAC^{\mathrm{EVM}}(t) = \mathrm{PD}/SPI^{\mathrm{EVM}}(t). \tag{4}$$

EVM is a simple and effective methodology but also has significant limitations. Firstly, neither the indicators nor the estimates at completion consider any correlation between cost and schedule performances (Khamooshi and Golafshani 2014). However, numerous studies have proven the existence of this relationship (Karimi et al. 2018), emphasizing the need to analyze both factors jointly (Mamghaderi et al. 2022). Secondly, EVM overlooks the project phase we are in (Ottaviani and De Marco 2022) and any past trends in performance (Du et al. 2016). In contrast, several works demonstrated that performance indicators must have reached stability to be reliable (Kim 2016; De Koning and Vanhoucke 2016; Ballesteros-Pérez et al. 2019; Barrientos-Orellana et al. 2021; Barrientos-Orellana et al. 2023). Ignoring these issues might compromise project performance assessment and lead to sub-optimal response actions.

Researchers have explored the application of machine learning (ML) methods to address some of the limitations of EVM (Fridgeirsson et al. 2021). These methods can analyze data, identify relationships among variables, and leverage such information to enhance project forecasts. Some of these studies have developed machine and deep learning regression models to substitute EVM formulas when forecasting the project cost and duration at completion (Chin-Chi Huang and Min-Yuan Cheng 2011).

Yet, developing ML models must also contemplate two phenomena, namely underfitting and overfitting (Karaca et al. 2020). Underfitting occurs when models do not fully capture the interrelationships of the analyzed data, leading to inaccurate forecasts. Conversely, overfitting occurs when models are excessively tailored to specific project records, generating inaccurate forecasts for slightly different projects.

There is no standard approach to developing ML models for project performance forecasting. ML models are developed through pipelines of sequenced phases, each comprising multiple steps. Although most of these phases are well-known and determined (i.e., data collection and pre-processing, feature engineering, model selection and evaluation), their steps depend on the data under analysis, forecasting priorities (i.e., mean accuracy or mitigating outliers), as well as other issues influencing underfitting or overfitting (e.g., data scaling, panel data).

This study aims to improve project monitoring by proposing an ML pipeline to deliver enhanced project performance forecasting models. Each pipeline step addresses a specific issue associated with the project monitoring data to improve the models' effectiveness and robustness. The study involves testing the pipeline on a dataset of 46 real projects and evaluating 27 ML models that benchmark their performance against the standard project forecasting method: the EVM methodology.

The paper is organized as follows. Section 2 introduced project performance analysis and forecasting and briefly discussed the challenges of the EVM methodology current ML methods. Section 3 outlines the research gap of EVM and ES (two standard methodologies for performance forecasting) and describes the pipelines followed by the studies that developed ML models for performance forecasting. Section 4 describes the steps of the proposed ML pipeline, providing solutions to the issues related to underfitting and overfitting. Section 5 evaluates ML models and compares their performance against standard methodologies. Section 6 explores research implications, while Section 7 summarizes key findings and the limitations of ML methods and identifies potential areas for future research.

LITERATURE REVIEW

This section first introduces the two standard methodologies for project cost and duration forecasting, highlights their limitations, and reviews previous works that attempted to solve them. Then, the following subsection introduces some ML methods, compares the pipelines adopted by ML models for project performance forecasting, and highlights some of their issues.

Background

EVM is arguably the most widespread project monitoring and control technique nowadays (Santos et al. 2023). Initially introduced by the U.S. federal government as a crucial component of the Cost/Schedule Control System Criteria (Kwak and Anbari 2012), the methodology was later formalized by Fleming and Koppelman (1997), leading to its recognition as both an ANSI standard (PMI 2005) and an ISO standard (BS 2018). Several studies have demonstrated the effectiveness of the EVM within various industries (Marshall et al. 2008; Chen and Zhang 2012; Netto et al. 2020; Mayo-Alvarez et al. 2022).

Regarding cost performance, studies have confirmed that the CPI can provide accurate forecasts (Batselier and Vanhoucke 2015b; Kim et al. 2019; Ballesteros-Pérez et al. 2019). Concerning schedule performance, though, SPI^{EVM} has faced criticism primarily due to its reliance on cost metrics (i.e., EV and PV) to quantify schedule delay (Borges and Mário 2017). This can lead to unreliable estimates when the relationship between cost and schedule is not linear (Khamooshi and Golafshani 2014; Warburton et al. 2017). Additionally, the SPI^{EVM} converges to a value of one as the project approaches completion, hinting that may be no delay no matter how delayed the project may be (De Marco et al. 2017; Chang et al. 2020).

To address the issues of SPI^{EVM}, Lipke (2003) proposed the Earned Schedule (ES) methodology. ES introduces the homonym metric (Earned Schedule, ES), which represents the time z at which the current EV should have been reached according to the baseline schedule, as per Eq. 5:

$$ES(t) = \{z : PV(z) = EV(t)\}\,. \tag{5}$$

The ES, divided by the elapsed time since the project start (t), yields the ES Performance Index (SPI^{ES}), as per Eq. 6:

$$SPI^{\mathrm{ES}}(t) = ES(t)/t. \tag{6}$$

The project duration estimate at completion ($tEAC^{\mathrm{ES}}$) is then obtained by adding t to the Time Estimate to Complete ($tETC$), which is the ratio of the difference between the Planned Duration and the Earned Schedule (PD $-$ ES) to SPI^{ES}, as per Eq. 7,

$$tEAC^{\mathrm{ES}}(t) = t + tETC(t) = t + [\mathrm{PD} - ES(t)]/SPI^{\mathrm{ES}}(t). \tag{7}$$

Compared to SPI^{EVM} from Eq. 2, SPI^{ES} relies on time metrics (i.e., ES and t) and does not converge to one, retaining representative information about whether the project schedule is ahead or behind schedule. Numerous studies have proven the ES approach more accurate than EVM for duration forecasting (Colin et al. 2015; Ballesteros-Pérez et al. 2019).

For clarity, Fig. 1 displays the EVM and ES metrics and indicators described so far. The value t denotes the current time. The PV (black line) reaches the project planned cost (BAC) at the planned duration (PD). The EV (dark gray line) reaches BAC at the forecasted time at completion ($tEAC$). Instead, AC (light gray line) reaches the forecasted cost at completion ($cEAC$) at $tEAC$. The $tEAC$ can be calculated following either the EVM approach (as per Eq. 4) or the ES approach (as per Eq. 7). In the latter case, ES is determined by projecting the to-date EV onto the PV curve, as per Eq. 5. Instead, the $cEAC$ is determined as the sum of the to-date AC and $cETC$, as per Eq. 3.

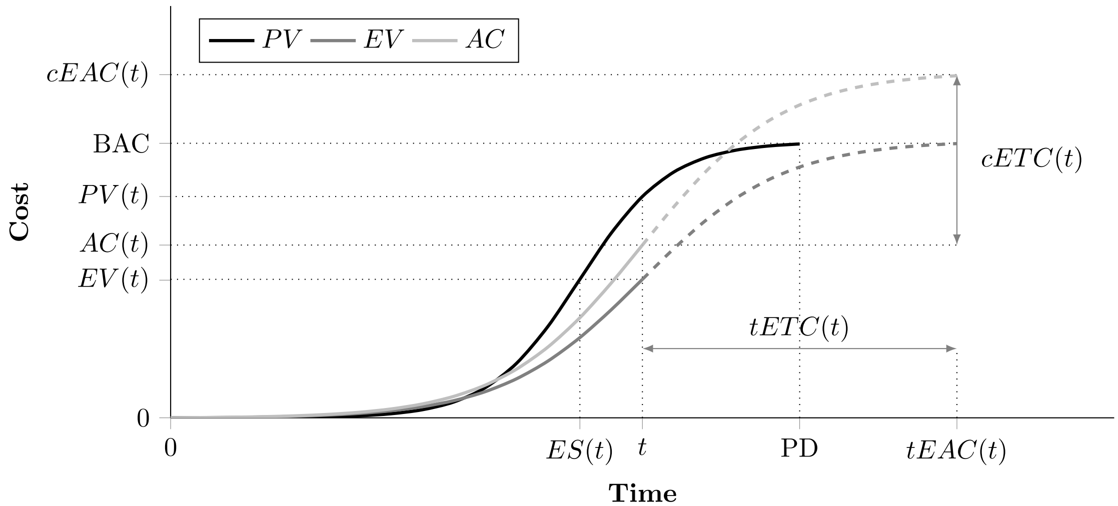

Fig. 1. Example project EVM and ES metrics

Despite their widespread use among practitioners, EVM and ES have some limitations. Firstly, they assess cost and schedule performances separately (i.e., SPI^{ES} does not incorporate AC, and CPI does not incorporate PV, ES, or t anyhow). Secondly, they forecast the project $cEAC$ and $tEAC$ by exclusively relying on the current values of CPI and SPI^{ES}, respectively, disregarding their trends (Du et al. 2016; Ottaviani and De Marco 2022; Ngo et al. 2022). Thirdly, they do not consider the possibility of corrective actions or external factors influencing the project execution (Narbaev and De Marco 2011; Willems and Vanhoucke 2015; Tariq et al. 2020). Lastly, they do not account for the project network features, such as subcritical paths (Vanhoucke 2010; Narbaev and De Marco 2011), level of parallelism between paths (Gálvez et al. 2015; Gálvez et al. 2017), and other relationships between activities (Vanhoucke 2012; Elshaer 2013). These limitations have prompted researchers to develop alternative solutions.

The first stream of studies used alternative projection factors to better evaluate the project $cEAC$ and $tEAC$. One group of studies evaluated the product of the cost and schedule indices (i.e., $CPI \cdot SPI^{\mathrm{EVM}}$ or SPI^{ES}) or their weighted sum [i.e., $\mathrm{w} \cdot CPI + (1 - \mathrm{w}) \cdot SPI^{\mathrm{EVM}}$ or SPI^{ES}] (Narbaev and De Marco 2011; Cândido et al. 2014; Batselier and Vanhoucke 2015c; Simion and Marin 2018; Kim et al. 2019) to consider both cost and schedule performances jointly. Others have used the moving average of the performance indicators (Christensen 1993; Christensen et al. 1995; Anbari 2003) or their exponential moving average (Batselier and Vanhoucke 2017a; Martens and Vanhoucke 2020; Zhao and Zi 2021) to account for past performance trends. However, these studies have provided discordant results. On the one hand, they confirmed CPI and SPI^{ES} to be the most robust cost and duration forecasting indicators, respectively. On the other hand, the most accurate approach within each study varied depending on project characteristics beyond the information prompted by the EVM and ES metrics.

The second stream of studies employed nonlinear regression to analyze the performance trends and evaluate the project $cEAC$ and $tEAC$ (Warburton 2011; Narbaev and De Marco 2014b; Narbaev and De Marco 2014a; Warburton and Cioffi 2016; Warburton et al. 2017; Warburton et al. ress). These studies fit the project EV and AC data to a theoretical model and determined both $tEAC$ and $cEAC$ geometrically [i.e., $tEAC$ is determined as the t at which $\widehat{EV}(t)$ = BAC while $cEAC$ is determined as $\widehat{EV}(tEAC)$]. The performance trend is considered by the curve fitting procedure, influencing the projection trajectory. Although this method proved accurate, its accuracy heavily relies on the geometrical properties of the theoretical profiles and the mathematical procedure utilized to evaluate their parameters.

ML Models

Recent advancements in ML have prompted researchers to explore its application in many domains, including project monitoring and control. Elmousalami (2020), Awada et al. (2021), and Santos et al. (2023) reviewed studies that utilized ML methods to improve the accuracy of cost and duration estimates, highlighting the predominance of machine and deep learning models. Compared to previous approaches, machine, and deep learning models rely on external data to infer the relationships between project metrics. Analyzing both internal factors (i.e., project metrics) and external factors (metrics relationships) can lead to improved forecasts (Huang et al. 2021; Taboada et al. 2023).

Table 1 presents a series of studies that used ML methods to develop project performance forecasting models. The table outlines the target forecasts (i.e., cost, duration, or both), models employed, and the data used (i.e., real or synthetic projects and their number). Additionally, the table indicates if the study involved data scaling, cross-validation (CV), and hyperparameter tuning, specifying the approaches taken. The acronyms are described in Section 10.

Table 1. Comparison of studies using ML methods for project cost and duration forecasting

Study	Target		ML Models	Data		Scaling	Cross Validation	Hyperparameters Tuning
	Cost	Duration		Number	Type			
Iranmanesh and Zarezadeh (2008)	✓		ANN	2	Synthetic			Manual
Iranmanesh and Mokhtari (2008)		✓	DT, NN, AR	1	Real			
Iranmanesh et al. (2009)		✓	NN, ANFIS, ELFIS, OLS	Undef.	Synthetic			
Cheng et al. (2010)	✓		SVM	11	Real	✓	Train-Test	fMGA
Cheng and Roy (2010)	✓		SVM	15	Real	✓	Train-Test	
Cheng et al. (2012)	✓		SVM	13	Real		Train-Test	fMGA
Feylizadeh et al. (2012)	✓		ANN					
Hajiali et al. (2014)	✓	✓	k-NN	3	Undef.		Train-Test	Manual
Wauters and Vanhoucke (2014)	✓	✓	SVM	900	Synthetic		k-fold	Grid Search CV
Cheng and Hoang (2014)	✓		SVM	13	Real	✓	k-fold	DE CV
Acebes et al. (2015)	✓	✓	GAMs, Natural Splines, Local Regression	1	Real		k-fold	
Wauters and Vanhoucke (2016)		✓	DT, Bagging, RF, GB, SVM	90	Synthetic		k-fold	Grid Search CV
Wauters and Vanhoucke (2017)		✓	DT, Bagging,RF, GB, SVM, k-NN	10	Synthetic	✓	Train-Test	
He et al. (2017)	✓		SVD	1	Real	✓	k-fold	
Cheng et al. (2019)		✓	ANN	11	Real	✓	k-fold	
Al Hares and Budayan (2019)	✓		ELM, ANN	11	Real			Manual
Kareem Kamoona and Budayan (2019)	✓		ANN	15	Real	✓		Manual
Balali et al. (2020)	✓		NN, MLR	50	Real	✓		Manual
Aidan et al. (2020)	✓	✓	ANN	Undef.	Real		Train-Test	Manual
Mohammed et al. (2021a)	✓	✓	ANN	Undef.	Real		Train-Test	
Mohammed et al. (2021b)	✓	✓	ANN	Undef.	Real		Train-Test	Manual
Ottaviani and De Marco (2022)	✓		MLR	29	Real	✓	Train-Test	Manual
Dastgheib et al. (2022)	✓		ANN	Undef.	Undef.	✓		Manual
Inan et al. (2022)	✓		ANN	41	Real		Train-Test	
Santos et al. (2023)		✓	GB, RF, XGBoost, AdaBoost	1	Real		k-fold	Grid Search
Liang et al. (2023)		✓	AdaBoost, Bagging, GB, k-NN,RF, SVM	1	Synthetic			

Regarding the target, only six studies developed models for simultaneous cost and duration forecasting, while the remaining studies focused exclusively on either one.

Regarding data, studies analyzing real project data involved fewer projects and tended to adopt less strict techniques for data scaling, CV, and hyperparameters tuning (e.g., no data scaling, Train-Test split instead of k-fold CV, and manual tuning). Alternatively, studies analyzing synthetic project data had the advantage of a larger sample size, allowing for capturing some project network properties. However, these studies could not account for the influence of other realistic internal and external factors on project performance.

About scaling, only ten studies pre-processed the data before providing them as input for model evaluation. Data scaling allows handling variations in metric values across different projects, reducing the complexity of relationships that ML methods need to infer to identify the major factors influencing project performance.

Concerning CV, only seven studies used a rigorous k-fold CV procedure, followed by nine studies that employed a less stringent train-test split approach. The remaining studies did not employ any CV technique. However, no study explicitly addressed the panel nature of the monitoring data. This implies that models were trained on specific project records and tested on the remaining ones, necessarily producing some degree of overfitting.

Regarding hyperparameter tuning, two studies implemented the exhaustive Grid Search CV procedure, one study used a standard Grid Search approach, two studies employed fmGA and DE CV, and eight studies manually adjusted the hyperparameters. The remaining studies did not explicitly mention any hyperparameter tuning. Hence, the limited use of automated hyperparameter tuning suggests potential improvements in model performance, especially when analyzing a large number of real project data.

In general, studies tend to overlook the "progress density" in the records. Suppose project reviews occur at fixed time intervals (e.g., weekly, monthly), and progress is nonlinear over time. In that case, records may miss the inherent features from project phases (e.g., early, middle, late stages). This increases the risk of overfitting the models to some stages (the training phases), making them inaccurate for upcoming (unseen) phases.

In summary, the literature reveals the absence of a standardized approach to model project performance assessment and forecasting. Most studies have not incorporated procedures to limit underfitting and overfitting either, and numerous studies analyzing real project data did not adopt strict procedures to ensure models' robustness. These findings underscore the importance of establishing a robust approach for developing ML regression models to provide more accurate forecasts than EVM and ES methodologies.

RESEARCH METHODOLOGY

This section outlines the proposed ML pipeline, which consists of five phases: Data Collection, Data Preprocessing, Feature Engineering, Model Selection, and Model Evaluation. Each phase includes multiple steps that tackle different aspects of model development and incorporate techniques to address underfitting and overfitting.

Data Collection

Data collection involves gathering monitoring data from completed projects and organizing them into a structured dataset. Project selection should prioritize projects that demonstrate similarities with those in which the ML models will be implemented. The similarities include projects within the same portfolio or industry, utilizing the same type of resources, or occurring in the same political or geographical territory (Müller et al. 2008).

The monitoring data consist of the following metrics: Tracking Period (t), Planned Duration (PD), Budget at Completion (BAC), Planned Value (PV), Earned Value (EV), Actual Cost (AC), and Actual Duration at Completion (AD).

Data Preprocessing

Data preprocessing involves transforming raw monitoring data into a suitable format for subsequent analysis and model development. This phase includes four steps: Handling Missing Values, Data Cleaning, Data Transformation, and Data Balancing and Augmentation.

Handling Missing Values

Handling Missing Values includes identifying dataset records containing one or more missing data and either removing or filling them in. Removing records may result in losing valuable information about the project performance, especially if many records have missing values or when the records are significant in explaining the relationships between project metrics. On the other hand, filling in the values can introduce bias and compromise the project performance assessment.

In the context under analysis, it is unlikely to have missing values in the monitoring data because of the simplicity of tracking them and their importance for stakeholders management. However, it is still recommended to manually enter any missing values if they occur to ensure the completeness of the dataset.

Data Cleaning

Data cleaning involves identifying and removing unsuitable records from the dataset. Records are deemed unsuitable in two cases: if their values contain errors or inconsistencies or relate to a project phase in which estimates are not carried out. The first case occurs when PV, EV, or AC are not increasing monotonically or PV and EV exceed the BAC. Instead, the second case refers to the projects' early and late stages, i.e., when data is insufficient for proper performance assessment and when conducting a bottom-up estimate of the remaining work is preferable, respectively. For the reasons above, our pipeline ignores each project's first record (i.e., when progress is null) and the last record (i.e., when the project is 100% complete).

Data Transformation

Data transformation involves normalizing, standardizing, or scaling the project metrics. This phase aims to reduce the amount of information ML methods must analyze to infer the relationships between the project metrics. It also allows for comparing projects under different metrics on the same scales.

Our pipeline implements data scaling by dividing cost metrics by the BAC (the project's planned cost) and time metrics by the PD (the project's planned duration). As a result, the scaled PV and EV range between 0 and 1 while the scaled t, AD, and AC range between 0 and inf.

Table 2 provides each project metric's original and scaled versions. The subscript "n" distinguishes the scaled versions of the project metrics, except for PV and EV, whose scaled versions are referred as Work Scheduled (WS) and Work Performed (WP), respectively, as per the EVM methodology.

Table 2. Monitoring metrics and respective scaled notation

Category	Metric	Scaled Metric
Progress	PV	$WS = PV/\text{BAC}$
	EV	$WP = EV/\text{BAC}$
Cost	BAC	$BAC_\text{n} = \text{BAC}/\text{BAC} = 1$
	AC	$AC_\text{n} = AC/\text{BAC}$
Schedule	t	$t_\text{n} = t/\text{PD}$
	PD	$\text{PD}_\text{n} = \text{PD}/\text{PD} = 1$
	AD	$\text{AD}_\text{n} = \text{AD}/\text{PD}$

Data Balancing and Augmentation

Data balancing and augmentation are related steps. Balancing involves making the dataset have the same number of records for each project. This prevents projects with more or fewer records from biasing model development. Balancing can be achieved by undersampling (i.e., reducing the number of records) or oversampling (i.e., increasing the number of records).

Both undersampling and oversampling techniques can involve real or synthetic records. Synthetic records are generated through data augmentation to describe the project status at specific time points. Incorporating these synthetic records increases the information available for the ML models to analyze, enhancing their understanding of project performance (Uddin et al. 2022).

Our pipeline implements balancing using synthetic records. Synthetic records are generated through interpolation to determine the values of the project metrics at 5% progress intervals, as per Eq. 8:

$$x(WP = z) = \{x(t_\text{n}) : WP(t_\text{n}) = z\} \qquad \forall\, z \in [0, .05, ..., .95, 1.00], \tag{8}$$

where x indicates the metric to interpolate and z the specific value of WP at t_n.

Interpolation can be linear or nonlinear. Linear interpolation assumes the progress rate to be constant between successive records, resulting in multiple straight lines connecting adjacent data points. Nonlinear interpolation assumes the progress rate to vary following a theoretical model with a non-uniform growth rate. This model preserves the monotonicity and cumulative nature of the project's t, PV, EV, and AC. Many studies, such as Putnam (1978), Carr

(1989), Cioffi (2005), Warburton (2011), Narbaev and De Marco (2014b), Warburton and Cioffi (2016), Narbaev and De Marco (2017), Simion and Marin (2018), Huynh et al. (2020), Le et al. (2021), and Warburton et al. (ress) have confirmed interpolation to be a feasible method to model the project metrics and predict future project behavior. For simplicity, this pipeline uses linear interpolation to generate synthetic records at predefined project monitoring intervals.

The proposed approach to balance the project dataset has multiple positive effects. Firstly, it increases the amount of information available for ML models, reducing underfitting. Secondly, it ensures models are trained on records that describe project performance over their whole makespan, thus reducing overfitting. However, determining the number of records must strike a balance between having enough observations for the ML models and avoiding introducing excessive noise or data distortion [e.g. if the difference between the values of the metrics of two successive records is too pronounced (Batselier and Vanhoucke 2017b)].

Table 3 compares the real and synthetic records generated through the proposed balancing procedure for an example project. The left side of the table displays the real records, which represent regular intervals of time elapsed (t_n = 0, .1111, .2222, ..., 1.3333). Instead, the right side of the table provides the corresponding values at 5% intervals of progress in WP, determined using Equation 8, assuming linear interpolation.

Table 3. Comparison of real and linearly-interpolated records for an example project

Real				Interpolated			
t_n	WS	WP	AC_n	t_n	WS	WP	AC_n
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.00	0.0000
0.1111	0.0500	0.0417	0.0521	0.1259	0.0600	0.05	0.0625
0.2222	0.1250	0.1042	0.1302	0.2148	0.1200	0.10	0.1250
0.3333	0.2250	0.1875	0.2344	0.2833	0.1800	0.15	0.1875
0.4444	0.4750	0.3958	0.4948	0.3400	0.2400	0.20	0.2500
0.5556	0.7250	0.6042	0.7552	0.3667	0.3000	0.25	0.3125
0.6667	0.8250	0.6875	0.8594	0.3933	0.3600	0.30	0.3750
0.7778	0.9000	0.7500	0.9375	0.4200	0.4200	0.35	0.4375
0.8889	0.9500	0.7917	0.9896	0.4467	0.4800	0.40	0.5000
1.0000	1.0000	0.8333	1.0417	0.4733	0.5400	0.45	0.5625
1.1111	1.0000	0.8889	1.1111	0.5000	0.6000	0.50	0.6250
1.2222	1.0000	0.9444	1.1806	0.5267	0.6600	0.55	0.6875
1.3333	1.0000	1.0000	1.2500	0.5533	0.7200	0.60	0.7500
				0.6167	0.7800	0.65	0.8125
				0.6889	0.8400	0.70	0.8750
				0.7778	0.9000	0.75	0.9375
				0.9111	0.9600	0.80	1.0000
				1.0333	1.0000	0.85	1.0625
				1.1333	1.0000	0.90	1.1250
				1.2333	1.0000	0.95	1.1875
				1.3333	1.0000	1.00	1.2500

Fig. 2 illustrates the proposed balancing procedure. The black squares indicate the real data points, while the gray rounds display the interpolated ones. The augmentation procedure involves connecting adjacent real data points with straight lines using linear interpolation. Then, the WP values at 0.05 intervals are projected horizontally onto the WP curve (Fig. 2a) and vertically from the WP curve onto the horizontal t_n axis. The latter determines the corresponding

t_n at which each specific WP value is expected to be achieved. This procedure is exemplified in Fig. 2 for a single point at $WP = .5$ to avoid cluttering the figure for all values between .05 and .95. Finally, projections from the interpolated WP points to the t_n axis determine the interpolated points for WS (Fig. 2b) and AC_n (Fig. 2c).

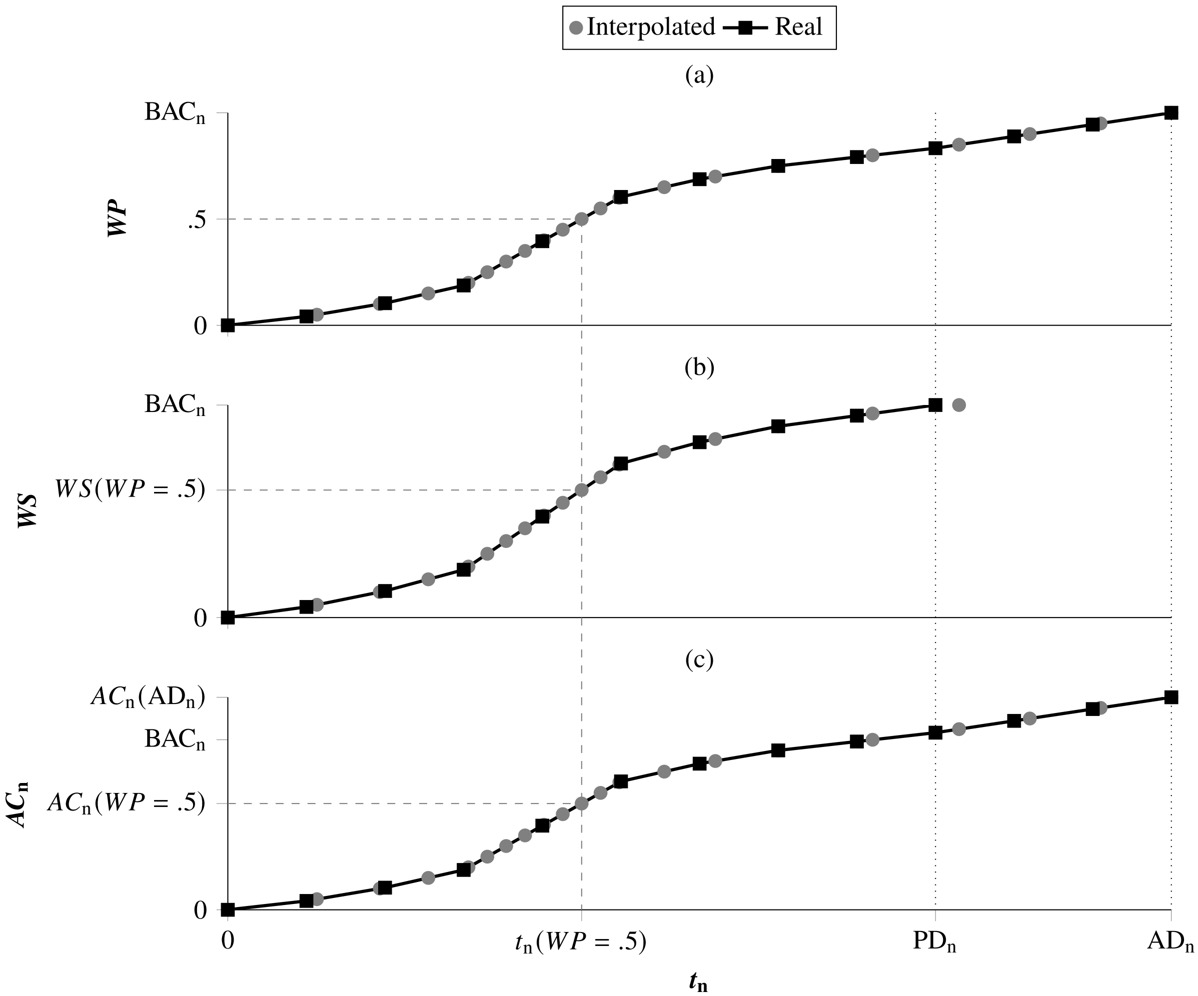

Fig. 2. Real and synthetic WP (a), WS (b), and AC_n (c) data points for example project

Feature Engineering

Feature engineering involves combining project metrics to evaluate additional features. They comprise input features, which serve as independent regression variables, and target features, representing the dependent variables to predict. This step improves the ML inferential process by guiding the analysis of significant features for assessing project performance while reducing both underfitting and overfitting.

Input Features

Input features comprise the project metrics and the monitoring indicators. The project metrics correspond to the synthetic values of t_n, WS, WP, and AC_n. The monitoring indicators consist of different combinations of project metrics.

The first set of indicators is based on the traditional EVM methodology. Concerning cost performance, the scaled Cost Variance (CV_n) is given by the difference between the Work Performed and the scaled Actual Cost, as per Eq. 9:

$$CV_\mathrm{n}(t_\mathrm{n}) = WP(t_\mathrm{n}) - AC_\mathrm{n}(t_\mathrm{n}), \tag{9}$$

The Cost Performance Index (CPI) is given by the ratio of the Work Performed to the scaled Actual Cost, as per Eq. 10:

$$CPI(t_\mathrm{n}) = WP(t_\mathrm{n})/AC_\mathrm{n}(t_\mathrm{n}). \tag{10}$$

The To Complete Cost Performance Index ($TCPI$) is given by the ratio of the difference between the project Budget at Completion and the Work Performed to the difference between the former and the Actual Cost, as per Eq. 11:

$$TCPI(t_\mathrm{n}) = [1 - WP(t_\mathrm{n})]/[1 - AC_\mathrm{n}(t_\mathrm{n})]. \tag{11}$$

Regarding schedule performance, the scaled Schedule Variance ($SV_\mathrm{n}^\mathrm{EVM}$) is given by the difference between the Work Performed and the Work Scheduled, as per Eq. 12:

$$SV_\mathrm{n}^\mathrm{EVM}(t_\mathrm{n}) = WP(t_\mathrm{n}) - WS(t_\mathrm{n}). \tag{12}$$

The Schedule Performance Index (SPI^EVM) is given by the ratio of the Work Performed to the Work Scheduled, as per Eq. 13:

$$SPI^\mathrm{EVM}(t_\mathrm{n}) = WP(t_\mathrm{n})/WS(t_\mathrm{n}). \tag{13}$$

The Critical Ratio (CR^EVM) measures the overall project cost performance relative to its planned schedule and is given by the product of the project Cost and Schedule Performance Indices, as per Eq. 14,

$$CR^\mathrm{EVM}(t_\mathrm{n}) = CPI(t_\mathrm{n}) \cdot SPI^\mathrm{EVM}(t_\mathrm{n}). \tag{14}$$

The second set of indicators is based on the ES methodology. The scaled Earned Schedule (ES_n) corresponds to the time z at which the Work Scheduled is equal to the current Work Performed, as per Eq. 15:

$$ES_\mathrm{n}(t_\mathrm{n}) = \{z : WS(z) = WP(t_\mathrm{n})\}. \tag{15}$$

Analogously, the scaled ES Variance ($SV_\mathrm{n}^\mathrm{ES}$) is given by the difference between the scaled Earned Schedule and the scaled Actual Time, as per Eq. 16:

$$SV_n^{ES}(t_n) = ES_n(t_n) - t_n. \tag{16}$$

The ES Performance Index, SPI^{ES}, is given by the ratio of the Earned Schedule to the Actual Time, as per Eq. 17:

$$SPI^{ES}(t_n) = ES_n(t_n)/t_n. \tag{17}$$

Additionally, as in the EVM methodology, the Critical Ratio can also be calculated in the ES theory as the product between the Cost Performance Index and the ES Performance Indices, as per Eq. 18:

$$CR^{ES}(t_n) = CPI(t_n) \cdot SPI^{ES}(t_n). \tag{18}$$

The To Complete Schedule Performance Index ($TSPI$) is given by the ratio of the difference between the project Planned Duration and scaled Earned Schedule to the difference between the former and the scaled Actual Time, as per Eq. 19,

$$TSPI(t_n) = [1 - ES_n(t_n)]/[1 - t_n]. \tag{19}$$

For the sake of clarity, Table 4 summarizes the input features, comprising both project metrics and monitoring indicators.

Table 4. Model development input features

Notation	Formula	Name
t_n	t/PD	Scaled Time
WS	PV/BAC	Work Scheduled
WP	EV/BAC	Work Performed
AC_n	$AC(t_n)$/BAC	Scaled Actual Cost
CV_n	$WP(t_n) - AC_n(t_n)$	Scaled Cost Variance
CPI	$WP(t_n)/AC_n(t_n)$	Cost Performance Index
$TCPI$	$[1 - WP(t_n)]/[1 - AC_n(t_n)]$	To Complete Performance Index
SV_n^{EVM}	$WP(t_n) - WS(t_n)$	Scaled Schedule Variance
SPI^{EVM}	$WP(t_n)/WS(t_n)$	Schedule Performance Index
CR^{EVM}	$CPI(t_n) \cdot SPI^{EVM}(t_n)$	Critical Ratio
ES_n	$z : WS(z) = WP(t_n)$	Scaled ES Metric
SV_n^{ES}	$ES_n(t_n) - t_n$	Scaled ES Variance
SPI_n^{ES}	$ES_n(t_n)/t_n$	ES Schedule Performance Index
CR^{ES}	$CPI(t_n) \cdot SPI^{ES}(t_n)$	ES Critical Ratio
$TSPI$	$[1 - ES_n(t_n)]/[1 - t_n]$	To Complete ES Performance Index

Target Variable

Target variable selection depends on both the target and the regression method used. The former is the project scaled cost or scaled duration at completion. The latter can either be direct or indirect regression.

Direct regression (DR) involves setting the target variable to the regression target. In this approach, records within the same project share identical values for the target variable. Consequently, the analysis of relationships between input features is limited, reducing the model's complexity. Hence, in DR, the method prioritizes reducing underfitting vs. overfitting.

Indirect regression (IR) involves setting the target variable to an intermediate variable, which is then used to calculate the regression target through a specific formula. In this approach, records from the same project have different values for the target variable. This allows for a more flexible analysis of relationships between input features. As a result, the method prioritizes reducing overfitting rather than overfitting.

This pipeline implements both regression methods for both regression targets (scaled cost and scaled duration). Let y denote the real value of the target variable, $\hat{y}$ denote the model forecast, and X represent the set of input features.

In DR, cost estimation is determined as per Eq. 20,

$$y(t_\mathrm{n}) = AC_\mathrm{n}(\mathrm{AD_n}) \rightarrow \hat{y}(t_\mathrm{n}) = f^{\mathrm{DR}}[X(t_\mathrm{n}); t_\mathrm{n}], \tag{20}$$

while the duration estimation is determined as per Eq. 21,

$$y(t_\mathrm{n}) = \mathrm{AD_n} \rightarrow \hat{y}(t_\mathrm{n}) = g^{\mathrm{DR}}[X(t_\mathrm{n}); t_\mathrm{n}], \tag{21}$$

where f^{DR} and g^{DR} represent the cost and duration regression models developed through DR, respectively.

In IR, the target variable is set to the performance factor that, when implemented within a predetermined formula, provides the project scaled cost or duration at completion. For cost estimation, the formula used is EVM's scaled EAC (as in Eq. 3), provided by Eq. 22:

$$cEAC(t_\mathrm{n}) = AC_\mathrm{n}(t_\mathrm{n}) + [1 - WP(t_\mathrm{n})]/cPF(t_\mathrm{n}), \tag{22}$$

where cPF denotes the Cost Performance Factor. Here, cPF is obtained based on Eq. 23, where f^{IR} represents the cPF regression model developed through IR.

$$\begin{aligned} y(t_\mathrm{n}) &= \{cPF(t_\mathrm{n}) : cEAC[cPF(t_\mathrm{n}); t_\mathrm{n}] = AC_\mathrm{n}(\mathrm{AD_n})\} \\ \rightarrow \hat{y}(t_\mathrm{n}) &= cEAC\left\{f^{\mathrm{IR}}[X(t_\mathrm{n})]; t_\mathrm{n}\right\} \end{aligned} \tag{23}$$

Regarding duration estimation, the set formula used is ES's scaled EAC, provided by Eq. 24:

$$tEAC^{\mathrm{ES}}(t_\mathrm{n}) = t_\mathrm{n} + [1 - ES_\mathrm{n}(t_\mathrm{n})]/sPF(t_\mathrm{n}), \tag{24}$$

where sPF denotes the Schedule Performance Factor. Here, sPF is obtained as per Eq. 25, where g^{IR} indicates the

sPF regression model developed through IR.

$$y(t_\mathrm{n}) = \left\{sPF(t_\mathrm{n}) : tEAC^{\mathrm{ES}}[sPF(t_\mathrm{n}); t_\mathrm{n}] = \mathrm{AD}_\mathrm{n}\right\}$$
$$\rightarrow \hat{y}(t_\mathrm{n}) = tEAC^{\mathrm{ES}}\left\{g^{\mathrm{IR}}[X(t_\mathrm{n})]; t_\mathrm{n}\right\} \tag{25}$$

Features Selection

Feature selection involves choosing a subset of input features. This limits the analysis of relationships between features to reduce underfitting and overfitting. It also improves the interpretability of the models by focusing on the most significant features. Lastly, it speeds up model training by eliminating irrelevant or redundant features.

Our pipeline employs a forward Sequential Feature Selection (fSFS) method with a tolerance-based stopping criterion. This algorithm starts with an empty set of features and iteratively evaluates the Root Mean Square Error ($RMSE$) by adding each feature to the model. The model includes the feature if the improvement exceeds a predefined threshold. This process is repeated until the improvement in $RMSE$ does not exceed the predetermined threshold or all available features have been annexed to the model.

The $RMSE$ is calculated as per Eq. 26:

$$RMSE = \left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right]^{1/2}. \tag{26}$$

The $RMSE$ prioritizes mitigating the impact of outliers rather than minimizing the overall magnitude of errors. This approach acknowledges that estimates, by nature, are just approximations and inherently imprecise. While a certain degree of estimation error is expected, an excessive error can result in incorrect assessments of project performance and potentially lead to irreversible control actions (Ballesteros-Pérez 2017).

Outliers in forecasts arise from project records in which risks have materialized that alter project costs. Following the EVM methodology, such forecasts would result in an out-of-control project cost-at-completion forecast. In this case, ML models' output considers the unlikelihood of such a situation coming true as the risk is mitigated or its effect is reduced over time.

Model Selection

Model selection consists of choosing the optimal model from a set of alternatives. Our pipeline evaluates 27 ML models and compares their performance for the accuracy and precision criteria. This comprehensive approach allows for assessing the pipeline's efficiency across all models and identifying differences in their performance.

Table 5 lists the 27 ML models used in this study, along with their respective categories (linear or nonlinear), subcategories, and sources.

Table 5. ML supervised models used in this study

Category	Subcategory	Model	Reference(s)
Linear	Linear	OLS	Goldberger et al. (2005)
		Ridge	Rifkin and Lippert (2007)
		Lasso	Kim et al. (2007)
		Elastic Net	Kim et al. (2007)
			Friedman et al. (2010)
		Lars	Efron et al. (2004)
		Lasso Lars	Efron et al. (2004)
		OMP	Rubinstein et al. (2008)
	Bayesian	Bayesian Ridge	Tipping (2001)
			Bishop (2006)
		ARD	Wipf and Nagarajan (2007)
	GLM	Tweedie (Normal)	Nelder and Wedderburn (1972)
	SGD	SGD	Bottou (1991)
	Robust Regression	RANSAC	Choi et al. (2009)
		Huber	Huber and Ronchetti (2009)
Nonlinear	Kernel Ridge	Kernel Ridge	Murphy (2012)
	SVM	SVR	Schölkopf et al. (2000)
		NuSVR	Schölkopf et al. (2000)
	k-NN	k-NN	Fix and Hodges (1989)
	Decision Trees	DT	Hastie et al. (2009)
			Breiman et al. (2017)
	Ensemble Methods	RF	Breiman (2001)
		ERT	Geurts et al. (2006)
		AdaBoost (DT)	Drucker (1997)
		GB	Friedman (2001)
			Friedman (2002)
			Hastie et al. (2009)
		Histogram-Based GB	Pedregosa et al. (2011)
		XGB	Chen et al. (2015)
		XGB RF	Chen et al. (2015)
		LightGBM	Ke et al. (2017)
	ANN	MLP	Hinton (1989)
			Glorot and Bengio (2010)
			He et al. (2015)
			Kingma and Ba (2015)

Model Evaluation

Model evaluation encompasses model training, hyperparameters optimization, and performance assessment. The initial step involves training the ML model using the selected features. Subsequently, the trained model is used to generate forecasts. These forecasts are then used to evaluate the model's performance by comparing the actual values of the target variable with the corresponding forecasts. Finally, the model performance is benchmarked against other ML models and the EVM and ES methodologies estimates.

Our pipeline implements the Group k-fold CV technique to reduce overfitting and hyperparameters tuning to reduce overfitting and underfitting.

Group K-Fold Cross Validation and Hyperparameters Tuning

The Group k-fold CV technique addresses the panel nature of project monitoring data. This technique involves dividing the dataset into k folds of equal size while ensuring that records from the same project are grouped within the same fold. During each iteration, the model is trained k times: each fold is used as the validation set once, while the remaining $k-1$ folds are used as the training set. This process allows for a comprehensive evaluation of the model's performance across different subsets of the data.

Our pipeline adopts $k = 5$. Within each fold, $\tilde{8}0\%$ of the projects are utilized for model training while the remaining $\tilde{2}0\%$ are used for validation. Table 6 presents an example of the Group 5-fold CV technique using a dataset of five projects, with two records per project.

Table 6. Group k-fold CV example (5 projects, $k = 5$, T = Train set, V = Validation set)

Project	t	i	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
			T	V	T	V	T	V	T	V	T	V
1	1	1.1	1.1		1.1		1.1		1.1			1.1
	2	1.2	1.2		1.2		1.2		1.2			1.2
2	1	2.1	2.1		2.1		2.1			2.1	2.1	
	2	2.2	2.2		2.2		2.2			2.2	2.2	
3	1	3.1	3.1		3.1			3.1	3.1		3.1	
	2	3.2	3.2		3.2			3.2	3.2		3.2	
4	1	4.1	4.1			4.1	4.1		4.1		4.1	
	2	4.2	4.2			4.2	4.2		4.2		4.2	
5	1	5.1		5.1	5.1		5.1		5.1		5.1	
	2	5.2		5.2	5.2		5.2		5.2		5.2	

Within each iteration, the model undergoes hyperparameter tuning using the Grid Search CV technique. The Grid Search CV technique explores all combinations of the values of the different hyperparameters to identify the optimal configuration that maximizes a set scoring metric (Table 10 presents the hyperparameters tuned for each model using the Grid Search CV procedure and their respective values). For consistency, the scoring metric is set to the $RMSE$ and the number of folds for CV to 5.

Performance Criteria

As noted earlier, models' performance is evaluated based on their accuracy and precision.

Accuracy refers to the extent to which a model's forecasts align with the respective true values. The criterion is assessed using two metrics: $RMSE$ and Mean Absolute Error (MAE). Both metrics are expressed in the same unit as the target variable. However, the $RMSE$ assesses the model's ability to handle outliers, while the MAE prioritizes its mean accuracy.

Let E_i indicate the prediction error for the ith record, given by the difference between the real value of the target variable ($Real_i$) and the forecasted value ($Forecast_i$). Then, $RMSE$ is calculated as per Eq. 27:

$$RMSE = \left(\frac{1}{n} \sum_{i=1}^{n} {E_i}^2 \right)^{1/2} = \left[\frac{1}{n} \sum_{i=1}^{n} (Real_i - Forecast_i)^2 \right]^{1/2}, \tag{27}$$

while MAE is determined as per Eq. 28:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |E_i| = \frac{1}{n} \sum_{i=1}^{n} |Real_i - Forecast_i|. \tag{28}$$

Precision refers to a model's ability to generate forecasts with a similar level of accuracy consistently. It is evaluated by examining the dispersion of residuals, which can be visualized using boxplots.

Fig. 3 provides a graphical representation of an example boxplot. In this representation, the box width represents the interquartile range (IQR), where the lower and upper edges indicate the first quartile (Q1) and third quartile (Q3), respectively. The line inside the box represents the residuals' median value (Q2). The left and right whiskers extend from the box to indicate the data range ($\pm 1.5 \cdot$ IQR), excluding outliers. The latter, if they exist, are represented as individual points beyond the whiskers as in the rightmost side of Fig. 3.

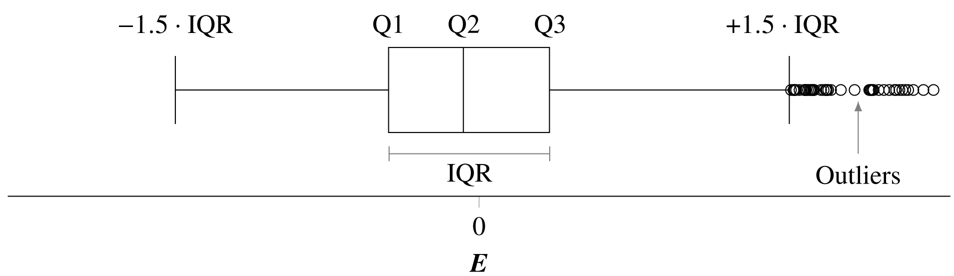

Fig. 3. Example boxplot

Summary

Algorithm 1 summarizes the proposed pipeline and how it was used to evaluate the performance criteria.

Algorithm 1 Research Methodology framework

```
Data Collection
Data Preprocessing
    Handling Missing Values
    Data Cleaning
    Data Transformation
    Data Balancing and Augmentation
Feature Engineering
for Target ∈ [Cost, Schedule] do
    for Method ∈ [DR, IR] do
        if Method = DR then
            if Target = Cost then
                y ← AC_n(AD_n)
            else if Target = Schedule then
                y ← AD_n
        else if Method = IR then
            if Target = Cost then
                y ← cPF : cEAC[cPF] = AC_n(AD_n)
            else if Target = Schedule then
                y ← sPF : tEAC^ES[sPF] = AD_n
        for Model ∈ Models do
            Group k-fold CV
                Feature Selection
                Hyperparameters Tuning
                Forecasts
            Performance Criteria Evaluation
    Results Comparison
```

Finally, for benchmarking purposes, the EVM project cost forecasts are evaluated using Eq. 22, and the duration forecasts are evaluated as per Eq. 29:

$$tEAC^{\mathrm{EVM}} = 1/SPI^{\mathrm{EVM}}(t). \tag{29}$$

Additionally, the ES project duration forecasts are evaluated using Eq. 24.

RESULTS

This section reports the results obtained after applying the ML pipeline to a dataset of real projects.

The pipeline was implemented as a Python 3.10.9 script using open-source libraries. NumPy (Harris et al. 2020) and Pandas (Reback et al. 2021) were used to manage the data structures. SciPy (Virtanen et al. 2020) was used to interpolate the project records in the Data Balancing and Augmentation step. Lastly, Scikit-learn (Pedregosa et al. 2011) was used for the sSFS method, Group k-fold CV technique, Grid Search CV technique, and ML models training and predictions. When specific hyperparameter values are not explicitly stated, the default values of the libraries used are applied.

The dataset comprises the monitoring data of 46 real projects selected from the OR&S database (Batselier and Vanhoucke 2015a; Vanhoucke et al. 2016). The database is available at projectmanagement.ugent.be/research/data/realdata and described in detail by Vanhoucke (2023). Table 7 presents the project year and code. The selection criteria ensured that the project belonged to the industrial engineering and construction industries and had planned and actual durations longer than six months. This prevented the augmentation procedure from returning unrealistic synthetic records when performing linear interpolation with 5% progress intervals.

Table 7. Projects codes according to OR&S database

Year	Code
2011	07, 12
2012	13
2013	01, 02, 03, 04, 05, 07, 08, 09, 10
2014	02, 04, 05, 06, 07, 08
2015	01, 02, 03, 04, 06, 07, 08, 09, 29, 30, 31
2016	01, 02, 03, 04, 05, 06, 07, 10
2018	10
2019	01, 02, 03, 04, 05, 06, 07, 08

Accuracy Results

Table 8 displays the ML models' $RMSE$ and MAE average values for the 46 projects.

Table 8. EVM, ES, and ML models $RMSE$ and MAE average values

Model	Cost				Schedule			
	$RMSE$		MAE		$RMSE$		MAE	
	DR	IR	DR	IR	DR	IR	DR	IR
EVM	0.1546		0.0640		0.5002		0.2592	
ES					0.4895		0.2370	
OLS	0.1488	0.1270	0.0909	0.0623	0.1466	0.1873	0.1100	0.1356
Ridge	0.1662	0.1252	0.0945	0.0621	0.1466	0.1789	0.1106	0.1304
Lasso	0.1263	0.1178	0.0915	0.0606	0.1596	0.1866	0.1246	0.1367
Elastic Net	0.1263	0.1179	0.0912	0.0606	0.1608	0.1684	0.1264	0.1253
Lars	0.1483	0.1270	0.0897	0.0623	0.1474	0.1873	0.1108	0.1356
Lasso Lars	0.1263	0.1178	0.0915	0.0606	0.1596	0.1866	0.1246	0.1367
OMP	0.1462	0.1270	0.0907	0.0623	0.1492	0.1863	0.1139	0.1330
Bayesian Ridge	0.1484	0.1270	0.0912	0.0625	0.1465	0.1857	0.1099	0.1344
ARD	0.1484	0.1269	0.0911	0.0624	0.1452	0.1849	0.1083	0.1334
Tweedie (Normal)	0.1268	0.1177	0.0913	0.0606	0.1624	0.1756	0.1254	0.1243
SGD	0.1632	0.1244	0.0932	0.0619	0.1468	0.1763	0.1113	0.1304
RANSAC	0.1436	0.1265	0.0633	0.0597	0.1614	0.1619	0.1174	0.1137
Huber	0.1536	0.1254	0.0862	0.0600	0.1517	0.1567	0.1113	0.1155
Kernel Ridge	0.1642	0.1681	0.0803	0.0862	0.1838	0.2144	0.1275	0.1543
SVR	0.1276	0.1267	0.0751	0.0683	0.1516	0.2055	0.1146	0.1177
NuSVR	0.1367	0.1280	0.0666	0.0598	0.1528	0.1541	0.1169	0.1149
k-NN	0.1130	0.1241	0.0642	0.0606	0.1533	0.1565	0.1169	0.1152
DT	0.1254	0.1165	0.0912	0.0611	0.1632	0.1630	0.1273	0.1171
RF	0.1298	0.1167	0.0804	0.0609	0.1650	0.1608	0.1249	0.1177
ERT	0.1254	0.1177	0.0912	0.0607	0.1632	0.1779	0.1273	0.1291
ADABoost	0.1542	0.1384	0.0923	0.0672	0.1621	0.1519	0.1266	0.1102
GB	0.1323	0.1177	0.0830	0.0606	0.1524	0.1627	0.1181	0.1138
Histogram-based GB	0.1357	0.1189	0.0818	0.0615	0.1554	0.1644	0.1183	0.1176
XGB	0.1229	0.1159	0.0659	0.0578	0.1577	0.2004	0.1195	0.1304
XGB RF	0.1302	0.1169	0.0832	0.0584	0.1657	0.1629	0.1210	0.1241
LightGBM	0.1377	0.1163	0.0755	0.0613	0.1592	0.1633	0.1252	0.1211
MLP	0.1145	0.1274	0.0739	0.0692	0.1725	0.1755	0.1325	0.1265

Regarding cost forecasting, all ML models, considering both DR and IR, exhibit lower $RMSE$ values than EVM, while the differences in MAE values are clearly less significant. This suggests two important considerations. Firstly, the EVM methodology provides a reasonable benchmark estimate. Secondly, when developed for cost forecasting, ML models can mitigate the impact of outliers in cost forecasts (lower $RMSE$) without significantly sacrificing their average accuracy (similar MAE). This prevents project managers from taking disruptive and irreversible responses proactively.

Concerning schedule forecasting, regardless of the method employed, all ML models demonstrate improvements in both $RMSE$ and MAE compared to EVM and ES methodologies. This indicates that the standard methodologies are susceptible to generating outlier forecasts when applied to schedule analysis. In contrast, ML models can mitigate outliers (lower $RMSE$) and identify the relationships between project schedule and network characteristics, thereby improving the accuracy of average predictions (lower MAE).

Table 9 summarizes the accuracy metrics for the best-performing models in terms of $RMSE$ for both cost and schedule targets (DR and IR approaches). For cost forecasting, both DR and IR outperform the EVM methodology regarding $RMSE$, with DR achieving 0.1130 and IR achieving 0.1159 compared to EVM's 0.1546. IR also performs best in MAE with a value of 0.0578 compared to EVM's 0.0640. Regarding schedule forecasting, the differences between the EVM and ES methodologies and the ML models are more noticeable. IR outperforms DR in both accuracy metrics, with an $RMSE$ of 0.1621 compared to DR's 0.1849. Similarly, IR has a lower MAE of 0.1266 compared to DR's 0.1334.

Table 9. EVM, ES, and best-performing ML models $RMSE$ and MAE values

Target	Method	Model	$RMSE$	MAE
Cost	EVM		0.1546	0.0640
	DR	k-NN	0.1130	0.0642
	IR	XGB	0.1159	0.0578
Schedule	EVM		0.5002	0.2592
	ES		0.4895	0.2370
	DR	ARD	0.1849	0.1334
	IR	ADABoost	0.1621	0.1266

Precision Results

Fig. 4 displays the boxplots of the residuals of the forecasting models from Table 9.

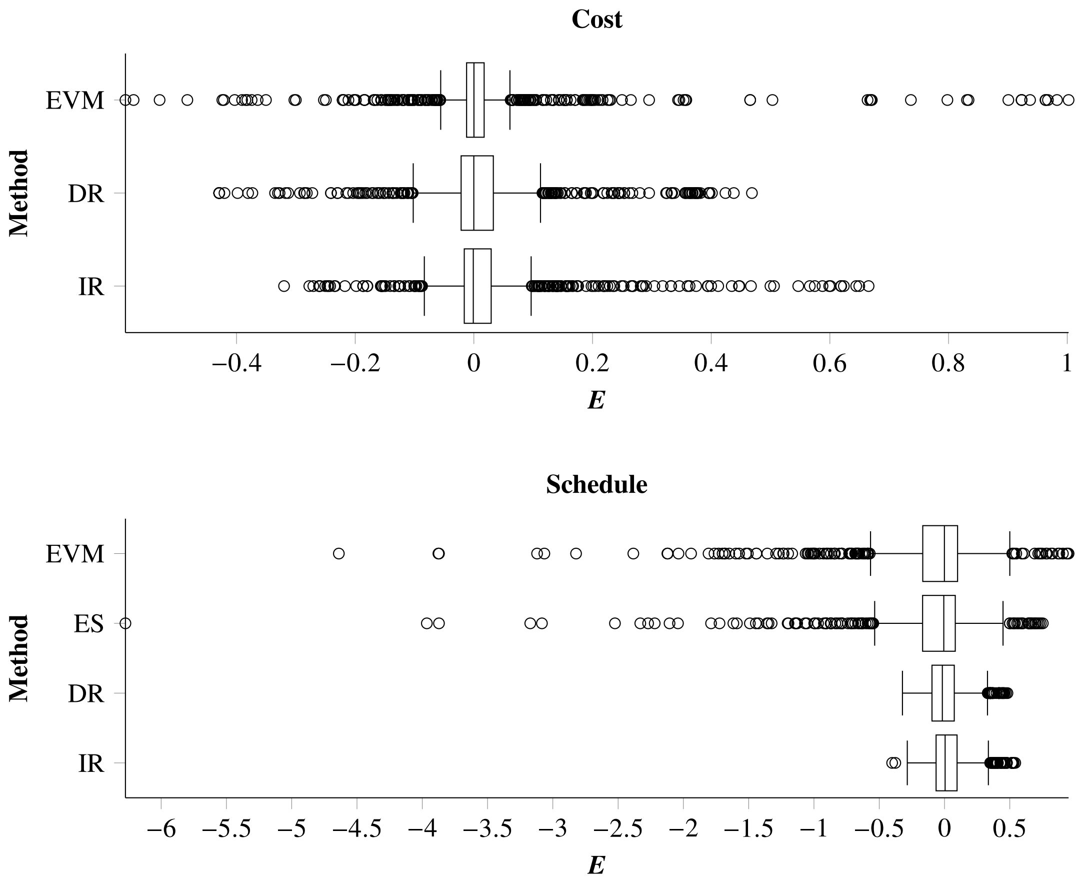

Fig. 4. Boxplots of residuals (E) of EVM, ES, and the best-performing ML models from Table 9

Regarding cost estimates, EVM exhibits a narrower IQR and range, indicating higher accuracy and precision in cost estimates. However, it is also evident that EVM is more prone to outliers. On the other hand, the boxplots for IR and DR show wider IQRs and higher ranges compared to EVM, suggesting lower accuracy and precision. However, when considering the residuals' absolute values, the differences in performance are negligible, with the minimum and maximum values not exceeding a ± 0.10 range approximately. This confirms that EVM provides a benchmark estimate with good accuracy and that ML models if developed appropriately, can mitigate outliers without significantly penalizing average forecasts.

Concerning schedule estimates, both EVM and ES exhibit larger IQRs and ranges and more outliers than ML models, confirming their improved performance according to accuracy and precision criteria.

DISCUSSIONS

This study aims to improve the project monitoring process by proposing an ML pipeline for developing enhanced project performance forecasting models. The pipeline comprises several steps, each addressing issues associated with monitoring data and model development. Specifically, these steps incorporate techniques reducing underfitting and overfitting to improve the models' effectiveness and robustness.

The study tested the pipeline on a dataset of 46 real projects by comparing the performance of 27 ML models against the EVM and ES methodologies. Concerning cost forecasts, the results prove that the EVM methodology provides accurate and precise forecasts but is more prone to outliers. In contrast, ML models can mitigate outliers while providing mean forecasts on par with the EVM methodology. Regarding duration forecasting, the results confirm that both EVM and ES methodologies provide accurate forecasts for specific records but are not that precise. ML models, on the other hand, can deliver more accurate and precise forecasts while mitigating outliers.

Comparing the DR and IR methods, their performance varies depending on the regression target and the ML model. On average, the two methods' performance is similar. However, focusing only on the best-performing models, IR proves slightly superior to DR (i.e., IR has lower $RMSE$ and MAE than DR). This suggests that ML models could benefit from shifting the focus from the relationships between the input features and the regression target to the intermediate variable.

This work has several implications. Using ML methods to develop project performance forecasting models requires addressing specific issues. Failing to do so compromises their implementation. In addition, implementing strict techniques to reduce underfitting and overfitting limits the differences in performance among the ML models. Another factor influencing the models' performance is the scoring metric chosen for the CV techniques, which determines whether the models prioritize mean forecasts vs. mitigating outliers (e.g., using $RMSE$ makes the ML models prioritize reducing outliers rather than mean deviations; instead, using MAE leads to the opposite). Nevertheless, our study proves the proposed pipeline can address all monitoring data and model development issues, ensuring the models' robustness while improving forecasting performance over the EVM and ES methodologies.

Limitations of this study mostly correspond to the dataset used, the parameters chosen for the CV techniques, and the values of the hyperparameters. Different datasets or parameter choices could yield different outcomes. However, the main 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 ES methodologies.

CONCLUSIONS

Proper assessment of project performance is fundamental to provide reliable estimates of the project cost and duration at completion, which, in turn, allow to take effective response actions when risks threatening project progress occur.

Thanks to their inferential capabilities, ML models represent an alternative to EVM and ES methodologies for project monitoring and control. These methods can assess the effects of external factors (e.g., stakeholders, risks) and internal factors (e.g., correlation between cost and time, project network properties) that affect the project, using them to improve project performance forecasts.

However, developing ML regression models must consider the monitoring data's characteristics and how these characteristics influence underfitting and overfitting. Ignoring these factors biases the performance of the models, compromising their implementation. In the case of underfitting, the models provide inaccurate predictions. In the case of overfitting, the models provide inaccurate predictions.

Previous studies tested several ML models for project performance forecasting. However, the studies developed the models and evaluated their performance through ML pipelines which did not consider all issues related to monitoring data. This implies that the studies prioritized optimizing forecasting accuracy, i.e., preventing underfitting, at the expense of model robustness and potentially incurring overfitting.

This study introduced an ML pipeline to develop future project performance forecasting models that are both effective and robust. The pipeline involves five phases, and each phase consists of several steps. The paper describes each step in detail, the issues that affect underfitting and overfitting, and the techniques used to reduce such phenomena.

The proposed pipeline differs from those found in the literature in several ways. First, it involves scaling the data by splitting cost metrics for BAC and time metrics for PD to reduce the complexity of the relationships determined by the different projects' economic orders of magnitude. Second, it involves project balancing through an augmentation procedure based on linear interpolation of project records to obtain the values of metrics at predefined progress intervals. This allows describing the project evolution homogeneously throughout its entire span. Third, we provide a group k-fold CV procedure for project performance evaluation, respecting the panel nature of project data, and avoiding that observations of the same projects are separated between training and validation sets. Fourth, we suggest both direct and indirect regression methods. In the latter, the target variable corresponds to the performance factor to be used within the EVM formulations for cost. In the ES formulation, the target variable for time is the project duration estimate at completion. This method also allows ML methods to analyze a larger sample of relationships.

The study tested the proposed pipeline's performance using 46 construction, industrial, and engineering projects. Results demonstrate that the models developed through the proposed pipeline exhibit superior accuracy and precision in predicting both project cost and duration at completion. Regarding cost estimation, although the improvement over the EVM methodology is modest, it significantly reduces the number and degree of outliers. On the scheduling side, though, the improvement is quite significant.

Our results have significant implications for project professionals. First, they stress the need for other ML pipelines for project performance forecasting that incorporate techniques that ensure effective and robust models. This will help reduce the disparities between predictions from the EVM and ES methodologies and those generated by ML models, particularly in duration forecasting. Once the best-performing model is identified, it can be trained on the whole dataset and applied to future projects. However, it is important to emphasize that ML model forecasts should complement, rather than replace, those obtained from EVM or ES methodologies, allowing a more comprehensive analysis.

There are certain limitations associated with ML models used in project performance forecasting. Firstly, these models often rely on external data sources, which can introduce uncertainties and potential biases. Additionally, ML models are considered black boxes, meaning their inner workings and decision-making processes may not be easily explainable or understandable. While ML model estimates can provide valuable insights, they should always be interpreted and enriched by expert judgment. It is important to carefully examine and scrutinize the forecasts generated by ML models to ensure their validity and relevance in the context of the specific project.

Future research in this field offers numerous avenues for further research and improvement. First, the study could be expanded to include a more comprehensive exploration of ML models, considering a wider range of algorithms and architectures. Furthermore, instead of just testing predefined models, future research could focus on optimizing the hyperparameters of these models to achieve better performance and accuracy. Additionally, the study's scope was limited to evaluating benchmark indicators from the EVM and ES methodologies. Future research could incorporate other indicators directly or indirectly influencing project cost or time performance. This broader set of indicators could provide a more comprehensive understanding of factors impacting project outcomes. In terms of feature selection, the study focused exclusively on the fSFS method. It would be beneficial to explore alternative feature selection techniques, such as backward elimination or regularized feature selection, to identify different subsets of relevant features and potentially improve model performance. Lastly, the IR method was developed based on the formulations of the EVM and ES methodologies for estimating revised costs and schedule forecasts. They also adopted the cost performance factor (cPF) and schedule performance factor (sPF) as target variables. Future research could replicate this procedure but with different formulations for the estimated cost at completion ($cEAC$) and schedule at completion ($tEAC$). By exploring alternative formulations, researchers may be able to assess their impact on the accuracy and precision of cost and schedule predictions more comprehensively.

DATA AVAILABILITY STATEMENT

All models or codes that support the findings of this study are available from the corresponding author upon request.

ACKNOWLEDGMENTS

This publication has been developed and reproduced with a grant from Project Management Institute, Inc.

NOTATION LIST

The following symbols are used in this paper:

AE	=	Absolute Error;
ANFIS	=	Adaptive Neuro-Fuzzy Inference System;
ANN	=	Artificial Neural Network;
AR	=	Association Rule;
ARD	=	Automatic Relevance Determination;
CV	=	Cross Validation;
DE	=	Differential Evolution;
DL	=	Deep Learning;
DT	=	Decision Tree;
ELFIS	=	Emotional Learning Fuzzy Interface System;
ERT	=	Extreme Randomized Tree;
EVM	=	Earned Value Management;
ES	=	Earned Schedule;
fmGA	=	Fast Messy Genetic Algorithm;
fSFS	=	Forward Sequential Feature Selection;
GAM	=	Generalized Additive Model;
GB	=	Gradient Boosting;
GLM	=	Generalized Linear Model;
k-NN	=	k-Nearest Neighbors;
ML	=	Machine Learning;
MLP	=	Multilayer Perceptron;
OLS	=	Ordinary Least Squares;
OMP	=	Orthogonal Matching Pursuit;
PMB	=	Performance Measurement Baseline;
RANSAC	=	Random Sample Consensus;
RF	=	Random Forest;
SE	=	Squared Error;
SGD	=	Stochastic Gradient Descent;
SVM	=	Support Vector Machine;
SVR	=	Support Vector Regression.

APPENDIX I. MODELS HYPERPARAMETERS VALUES

Table 10. ML models hyperparameters values for Grid Search CV procedure

Model	Hyperparameter	Values
Ridge	alpha	1e-4, 1e-3, 1e-2, 1e-1, 1e-0, 1e1, 1e2, 1e3, 1e4
	tol	1e-6, 1e-5, 1e-4, 1e-3, 1e-2
Lasso	alpha	1e-4, 1e-3, 1e-2, 1e-1, 1e-0, 1e1, 1e2, 1e3, 1e4
	tol	1e-6, 1e-5, 1e-4, 1e-3, 1e-2
Elastic Net	l1_ratio	0, .25, .5, .75, 1
	alpha	1e-4, 1e-3, 1e-2, 1e-1, 1e-0, 1e1, 1e2, 1e3, 1e4
Lasso Lars	alpha	1e-4, 1e-3, 1e-2, 1e-1, 1e-0, 1e1, 1e2, 1e3, 1e4
Passive Aggressive	C	1e-2, 1e-2, 1, 1e1, 1e2
	tol	1e-6, 1e-5, 1e-4, 1e-3, 1e-2
Bayesian Ridge	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
ARD	tol	1e-4, 1e-3, 1e-2
	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
Tweedie (Normal)	tol	1e-6, 1e-5, 1e-4, 1e-3, 1e-2
SGD	l1_ratio	0, .25, .5, .75, 1
	alpha	1e-6, 1e-5, 1e-4, 1e-3, 1e-2
Huber	epsilon	1.35, 2
	alpha	1e-6, 1e-5, 1e-4, 1e-3, 1e-2
	tol	1e-6, 1e-5, 1e-4, 1e-3, 1e-2
SVR	kernel	poly, rbf
	C	1e-2, 1e-1, 1
NuSVR	nu	.05, .25, .5, .75, 1
	C	1e-2, 1e-1, 1
k-NN	n_neighbors	1, 3, 5, 7, 10, 20, 50
	weights	uniform, distance
	p	1, 2
DT	max_depth	1, 3, 5
RF	max_depth	1, 3, 5
ERT	max_depth	1, 3, 5
AdaBoost (DT)	learning_rate	1e-2, 1e-1, 1
GB	learning_rate	1e-3, 1e-2, 1e-1
	max_depth	1, 3, 5
Histogram-Based GB	learning_rate	1e-2, 1e-1, 1
	max_depth	1, 3, 5
XGB	learning_rate	1e-3, 1e-2, 1e-1
	max_depth	1, 3, 5
XGB RF	learning_rate	1e-2, 1e-1, 1
	max_depth	1, 3, 5
MLP	activation	identity, logistic, tanh, relu

REFERENCES

Acebes, F., Pereda, M., Poza, D., Pajares, J., et al. (2015). "Stochastic earned value analysis using Monte Carlo simulation and statistical learning techniques." *International Journal of Project Management*, 33(7), 1597–1609.

Aidan, I. A., Al-Jeznawi, D., and Al-Zwainy, F. M. (2020). "Predicting earned value indexes in residential complexes' construction projects using artificial neural network model." *International Journal of Intelligent Engineering and Systems*, 13(4), 248–259.

Al Hares, E. F. T. and Budayan, C. (2019). "Estimation at completion simulation using the potential of soft computing models: Case study of construction engineering projects." *Symmetry*, 11(2).

Anbari, F. T. (2003). "Earned Value Project Management Method and Extensions." *Project Management Journal*, 34(4), 12–23.

Awada, M., Srour, F. J., and Srour, I. M. (2021). "Data-Driven Machine Learning Approach to Integrate Field Submittals in Project Scheduling." *Journal of Management in Engineering*, 37(1), 04020104.

Balali, A., Valipour, A., Antucheviciene, J., and Šaparauskas, J. (2020). "Improving the results of the earned value management technique using artificial neural networks in construction projects." *Symmetry*, 12(10), 1–17.

Ballesteros-Pérez, P. (2017). "Modelling the boundaries of project fast-tracking." *Automation in Construction*, 84(March), 231–241.

Ballesteros-Pérez, P., Sanz-Ablanedo, E., Mora-Melià, D., González-Cruz, M., et al. (2019). "Earned Schedule min-max: Two new EVM metrics for monitoring and controlling projects." *Automation in Construction*, 103, 279–290.

Barrientos-Orellana, A., Ballesteros-Pérez, P., Mora-Melià, D., Cerezo-Narváez, A., and Gutiérrez-Bahamondes, J. H. (2023). "Comparison of the Stability and Accuracy of Deterministic Project Cost Prediction Methods in Earned Value Management." *Buildings*, 13(5).

Barrientos-Orellana, A., Ballesteros-Pérez, P., Mora-Melia, D., González-Cruz, M. C., et al. (2021). "Stability and accuracy of deterministic project duration forecasting methods in earned value management." *Engineering, Construction and Architectural Management*, 29(3), 1449–1469.

Batselier, J. and Vanhoucke, M. (2015a). "Construction and evaluation framework for a real-life project database." *International Journal of Project Management*, 33(3), 697–710.

Batselier, J. and Vanhoucke, M. (2015b). "Empirical Evaluation of Earned Value Management Forecasting Accuracy for Time and Cost." *Journal of Construction Engineering and Management*, 141(11), 05015010.

Batselier, J. and Vanhoucke, M. (2015c). "Evaluation of deterministic state-of-the-art forecasting approaches for project duration based on earned value management." *International Journal of Project Management*, 33(7), 1588–1596.

Batselier, J. and Vanhoucke, M. (2017a). "Improving project forecast accuracy by integrating earned value management with exponential smoothing and reference class forecasting." *International Journal of Project Management*, 35(1), 28–43.

Batselier, J. and Vanhoucke, M. (2017b). "Project regularity: Development and evaluation of a new project characteristic." *Journal of Systems Science and Systems Engineering*, 26(1), 100–120.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer New York (dec).

Borges, W. F. and Mário, P. d. C. (2017). "Five Project-Duration Control Methods in Time Units: Case Study of a Linearly Distributed Planned Value." *Journal of Construction Engineering and Management*, 143(6), 05017002.

Bottou, L. (1991). "Stochastic Gradient Learning in Neural Networks." *Proceedings of Neuro-Nımes*, 91(8), 12.

Breiman, L. (2001). "Random forests." *Machine Learning*, 45(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification And Regression Trees*. Routledge (oct).

BS (2018). "BS ISO 21508 : 2018 BSI Standards Publication Earned value management in project and programme management.

Cândido, L. F., Heineck, L. F. M., and De Paula Barros Neto, J. (2014). "Critical analysis on earned value management (EVM) technique in building construction." *22nd Annual Conference of the International Group for Lean Construction: Understanding and Improving Project Based Production, IGLC 2014*, Oslo, 159–170.

Caron, F., Ruggeri, F., and Pierini, B. (2016). "A Bayesian approach to improving estimate to complete." *International Journal of Project Management*, 34(8), 1687–1702.

Carr, R. I. (1989). "Cost-Estimating Principles." *Journal of Construction Engineering and Management*, 115(4), 545–551.

Chang, H. K., Yu, W. D., and Cheng, T. M. (2020). "A Quantity-Based Method to Predict More Accurate Project Completion Time." *KSCE Journal of Civil Engineering*, 24(10), 2861–2875.

Chen, H., Ma, S., Li, T., and Zhuang, X. (2015). "Research on risk assessment and control decision-making of collaborative r&d between main manufacturer and suppliers for aviation complex products." *2015 IEEE International Conference on Grey Systems and Intelligent Services (GSIS)*, 409–414.

Chen, H. L., Chen, W. T., and Lin, Y. L. (2016). "Earned value project management: Improving the predictive power of planned value." *International Journal of Project Management*, 34(1), 22–29.

Chen, S. and Zhang, X. (2012). "An analytic review of earned value management studies in the construction industry." *Construction Research Congress 2012: Construction Challenges in a Flat World, Proceedings of the 2012 Construction Research Congress*, (May 2012), 236–246.

Cheng, M.-Y., Chang, Y.-H., and Korir, D. (2019). "Novel Approach to Estimating Schedule to Completion in Construction Projects Using Sequence and Nonsequence Learning." *Journal of Construction Engineering and Management*, 145(11), 04019072.

Cheng, M.-Y. and Hoang, N.-D. (2014). "Interval estimation of construction cost at completion using least squares support vector machine." *JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT*, 20(2), 223–236.

Cheng, M. Y., Hoang, N. D., Roy, A. F., and Wu, Y. W. (2012). "A novel time-depended evolutionary fuzzy SVM inference model for estimating construction project at completion." *Engineering Applications of Artificial Intelligence*, 25(4), 744–752.

Cheng, M.-Y., Peng, H.-S., Wu, Y.-W., and Chen, T.-L. (2010). "Estimate at Completion for construction projects using Evolutionary Support Vector Machine Inference Model." *Automation in Construction*, 19(5), 619–629.

Cheng, M.-Y. and Roy, A. F. (2010). "Evolutionary fuzzy decision model for construction management using support vector machine." *Expert Systems with Applications*, 37(8), 6061–6069.

Chin-Chi Huang and Min-Yuan Cheng (2011). "Estimate at completion for construction projects Using Evolutionary Gaussian Process Inference Model." *2011 International Conference on Multimedia Technology*, IEEE, 4414–4417 (jul).

Choi, S., Kim, T., and Yu, W. (2009). "Performance evaluation of RANSAC family." *British Machine Vision Conference, BMVC 2009 - Proceedings*.

Christensen, D. S. (1993). "The estimate at completion problem: A review of three studies." *Project Management Journal*, 24(1), 37–42.

Christensen, D. S., Antolini, R. C., and McKinney, J. W. (1995). "A Review of Estimate at Completion Research." *The*

Journal of Cost Analysis, 12(1), 41–62.

Cioffi, D. F. (2005). “A tool for managing projects: An analytic parameterization of the S-curve.” *International Journal of Project Management*, 23(3), 215–222.

Colin, J., Martens, A., Vanhoucke, M., and Wauters, M. (2015). “A multivariate approach for top-down project control using earned value management.” *Decision Support Systems*, 79, 65–76.

Dastgheib, S. R., Feylizadeh, M. R., Bagherpour, M., and Mahmoudi, A. (2022). “Improving estimate at completion (EAC) cost of construction projects using adaptive neuro-fuzzy inference system (ANFIS).” *Canadian Journal of Civil Engineering*, 49(2), 222–232.

De Koning, P. and Vanhoucke, M. (2016). “Stability of earned value management do project characteristics influence the stability moment of the cost and schedule performance index.” *Journal of Modern Project Management*, 4(1), 9–25.

De Marco, A., Narbaev, T., Rafele, C., and Cagliano, A. C. (2017). “Integrating risk in project cost forecasting.” *Proceedings of the Summer School Francesco Turco*, 2017-Septe(November 2019), 117–122.

Demachkieh, F. and Abdul-Malak, M.-A. (2018). “Degree of Criticality of Monitoring and Control to Project Success.” *Construction Research Congress 2018*, American Society of Civil Engineers (ASCE), 389–398 (mar).

Drucker, H. (1997). “Improving regressors using boosting techniques.” *14th International Conference on Machine Learning*, 107–115.

Du, J., Kim, B.-C., and Zhao, D. (2016). “Cost Performance as a Stochastic Process: EAC Projection by Markov Chain Simulation.” *Journal of Construction Engineering and Management*, 142(6), 04016009.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). “Least angle regression.” *Annals of Statistics*, 32(2), 407–499.

Elmousalami, H. H. (2020). “Artificial Intelligence and Parametric Construction Cost Estimate Modeling: State-of-the-Art Review.” *Journal of Construction Engineering and Management*, 146(1), 03119008.

Elshaer, R. (2013). “Impact of sensitivity information on the prediction of project’s duration using earned schedule method.” *International Journal of Project Management*, 31(4), 579–588.

Feylizadeh, M. R., Hendalianpour, A., and Bagherpour, M. (2012). “A fuzzy neural network to estimate at completion costs of construction projects.” *international journal of industrial engineering computations*, 3(3), 477–484.

Fix, E. and Hodges, J. L. (1989). "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties." *International Statistical Review / Revue Internationale de Statistique*, 57(3), 238.

Fleming, Q. W. and Koppelman, J. M. (1997). "Earned value project management." *Cost Engineering (Morgantown, West Virginia)*, 39(2), 13–15.

Fridgeirsson, T. V., Ingason, H. T., Jonasson, H. I., and Jonsdottir, H. (2021). "An authoritative study on the near future effect of artificial intelligence on project management knowledge areas." *Sustainability (Switzerland)*, 13(4), 1–20.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). "Regularization paths for generalized linear models via coordinate descent." *Journal of Statistical Software*, 33(1), 1–22.

Friedman, J. H. (2001). "Greedy function approximation: A gradient boosting machine." *Annals of Statistics*, 29(5), 1189–1232.

Friedman, J. H. (2002). "Stochastic gradient boosting." *Computational Statistics and Data Analysis*, 38(4), 367–378.

Gálvez, E. D., Ordieres-Meré, J. B., and Capuz-Rizo, S. F. (2015). "Analysis of project duration: Uncertainty using global sensitivity analysis." *Journal of Modern Project Management*, 2(3), 18–25.

Gálvez, E. D., Ordieres-Meré, J. B., and Capuz-Rizo, S. F. (2017). "On uncertainty and sensitivity analyses in project duration based on dependency information." *Journal of Modern Project Management*, 4(3), 98–109.

Gedi, J., Yunna, W., and Qing, C. (2018). "Research on Optimization of Earned Value Management Method in Engineering Budget Management." *The Open Civil Engineering Journal*, 9(1), 369–375.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). "Extremely randomized trees." *Machine Learning*, 63(1), 3–42.

Glorot, X. and Bengio, Y. (2010). "Understanding the difficulty of training deep feedforward neural networks." *Journal of Machine Learning Research*, Vol. 9, 249–256.

Goldberger, J., Roweis, S., Hinton, G., and Salakhutdinov, R. (2005). "Neighbourhood components analysis." *Advances in Neural Information Processing Systems*, 17.

Hajiali, M. T., Mosavi, M. R., and Shahanaghi, K. (2014). "Estimation of project completion time-based on a mixture of expert in an interactive space." *Modern Applied Science*, 8(6), 229–237.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., et al. (2020). "Array programming with NumPy." *Nature*, 585(7825), 357–362.

Hastie, T., Rosset, S., Zhu, J., and Zou, H. (2009). "Multi-class AdaBoost." *Statistics and Its Interface*, 2(3), 349–360.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.” *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 2015 Inter, IEEE, 1026–1034 (dec).

He, S., Du, J., and Huang, J. Z. (2017). “Singular-Value Decomposition Feature-Extraction Method for Cost-Performance Prediction.” *Journal of Computing in Civil Engineering*, 31(5), 04017043.

Hinton, G. E. (1989). “Connectionist learning procedures.” *Artificial Intelligence*, 40(1-3), 185–234.

Huang, Y., Shi, Q., Zuo, J., Pena-Mora, F., et al. (2021). “Research Status and Challenges of Data-Driven Construction Project Management in the Big Data Context.” *Advances in Civil Engineering*, 2021.

Huber, P. J. and Ronchetti, E. M. (2009). *Robust Statistics*. John Wiley & Sons, Inc., Hoboken, NJ, USA (jan).

Huynh, Q. T., Le, T. A., Nguyen, T. H., Nguyen, N. H., et al. (2020). “A Method for Improvement the Parameter Estimation of Non-linear Regression in Growth Model to Predict Project Cost at Completion.” *Proceedings - 2020 RIVF International Conference on Computing and Communication Technologies, RIVF 2020*, IEEE, 1–6 (oct).

Inan, T., Narbaev, T., and Hazir, Ö. (2022). “A Machine Learning Study to Enhance Project Cost Forecasting.” *IFAC-PapersOnLine*, 55(10), 3286–3291.

Iranmanesh, H., Mojir, N., and Kimiagari, S. (2007). “A new formula to "Estimate At Completion" of a Project’s time to improve "Earned Value Management System".” *IEEM 2007: 2007 IEEE International Conference on Industrial Engineering and Engineering Management*, 1014–1017.

Iranmanesh, S. and Zarezadeh, M. (2008). “Application of Artificial Neural Network to Forecast Actual Cost of a Project to Improve Earned Value Management System.” *International Journal of Educational and Pedagogical Sciences*, 2(6), 658 – 661.

Iranmanesh, S. H., Mirseraji, G. H., and Shahmiri, S. (2009). “An emotional learning based fuzzy inference system (ELFIS) for improvementof the completion time of projects estimation.” *2009 International Conference on Computers and Industrial Engineering, CIE 2009*, 4563(2006), 470–475.

Iranmanesh, S. H. and Mokhtari, Z. (2008). “Application of data mining tools to predicate completion time of a project.” *Proceeding of world academy of science, engineering and technology*, 32(August), 234–240.

Karaca, I., Gransberg, D. D., and Jeong, H. D. (2020). “Improving the Accuracy of Early Cost Estimates on Transportation Infrastructure Projects.” *Journal of Management in Engineering*, 36(5), 04020063.

Kareem Kamoona, K. R. and Budayan, C. (2019). "Implementation of Genetic Algorithm Integrated with the Deep Neural Network for Estimating at Completion Simulation." *Advances in Civil Engineering*, 2019.

Karimi, H., Taylor, T. R. B., Dadi, G. B., Goodrum, P. M., et al. (2018). "Impact of Skilled Labor Availability on Construction Project Cost Performance." *Journal of Construction Engineering and Management*, 144(7), 04018057.

Ke, G., Meng, Q., Finley, T., Wang, T., et al. (2017). "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." *Advances in Neural Information Processing Systems*, Vol. 30 of *NIPS'17*, Curran Associates, Inc., 3149–3157.

Khamooshi, H. and Golafshani, H. (2014). "EDM: Earned Duration Management, a new approach to schedule performance management and measurement." *International Journal of Project Management*, 32(6), 1019–1041.

Kim, B.-C. (2016). "Probabilistic Evaluation of Cost Performance Stability in Earned Value Management." *Journal of Management in Engineering*, 32(1), 04015025.

Kim, D. B., White, E. D., Ritschel, J. D., and Millette, C. A. (2019). "Revisiting reliability of estimates at completion for department of defense contracts." *Journal of Public Procurement*, 19(3), 186–200.

Kim, S. J., Koh, K., Lustig, M., Boyd, S., et al. (2007). "An interior-point method for large-scale L1-regularized least squares." *IEEE Journal on Selected Topics in Signal Processing*, 1(4), 606–617.

Kingma, D. P. and Ba, J. L. (2015). "Adam: A method for stochastic optimization." *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Kwak, Y. H. and Anbari, F. T. (2012). "History, Practices, and Future of Earned Value Management in Government: Perspectives from NASA." *Project Management Journal*, 43(1), 77–90.

Kwon, H. and Kang, C. W. (2019). "Improving Project Budget Estimation Accuracy and Precision by Analyzing Reserves for Both Identified and Unidentified Risks." *Project Management Journal*, 50(1), 86–100.

Le, T. A., Huynh, Q. T., and Nguyen, T. H. (2021). "A New Method to Improve Quality Predicting of Software Project Completion Level." *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, Vol. 379, Springer, 211–219.

Liang, A., Tao, L., and Lei, H. (2023). "Combined machine-learning and EDM to monitor and predict a complex project with a GERT-type network: A multi-point perspective." *Computers and Industrial Engineering*, 180(December 2022), 109256.

Lipke, W. (2003). "Schedule is Different." *The Measurable News*, 2, 31–34.

Mamghaderi, M., Khamooshi, H., and Kwak, Y. H. (2022). “Project Duration Forecasting: A Simulation-Based Comparative Assessment of Earned Schedule Method and Earned Duration Management.” *Journal of Modern Project Management*, 9(1), 7–19.

Marshall, R. A., Ruiz, P., and Bredillet, C. N. (2008). “Earned value management insights using inferential statistics.” *International Journal of Managing Projects in Business*, 1(2), 288–294.

Martens, A. and Vanhoucke, M. (2020). “Integrating Corrective Actions in Project Time Forecasting Using Exponential Smoothing.” *Journal of Management in Engineering*, 36(5), 04020044.

Maylor, H., Geraldi, J., Budzier, A., Turner, N., and Johnson, M. (2023). “Mind the gap: Towards performance measurement beyond a plan-execute logic.” *International Journal of Project Management*, 41(4), 102467.

Mayo-Alvarez, L., Alvarez-Risco, A., Del-Aguila-Arcentales, S., Sekar, M. C., et al. (2022). “A Systematic Review of Earned Value Management Methods for Monitoring and Control of Project Schedule Performance: An AHP Approach.” *Sustainability (Switzerland)*, 14(22).

Mohammed, S. J., Abdel-Khalek, H. A., and Hafez, S. M. (2021a). “Predicting performance measurement of residential buildings using an artificial neural network.” *Civil Engineering Journal (Iran)*, 7(3), 461–476.

Mohammed, S. J., Abdel-khalek, H. A., and Hafez, S. M. (2021b). “Predicting Performance Measurement of Residential Buildings Using Machine Intelligence Techniques (MLR, ANN and SVM).” *Iranian Journal of Science and Technology - Transactions of Civil Engineering*.

Müller, R., Martinsuo, M., and Blomquist, T. (2008). “Project Portfolio Control and Portfolio Management Performance in Different Contexts.” *Project Management Journal*, 39(3), 28–42.

Muriana, C. and Vizzini, G. (2017). “Project risk management: A deterministic quantitative technique for assessment and mitigation.” *International Journal of Project Management*, 35(3), 320–340.

Murphy, K. P. (2012). *Machine Learning A Probabilistic Perspective*. MIT press.

Narbaev, T. and De Marco, A. (2011). “Cost Estimate at Completion Methods in Construction Projects.” *2011 International Conference on Construction and Project Management*, 15(September), 32–36.

Narbaev, T. and De Marco, A. (2014a). “An Earned Schedule-based regression model to improve cost estimate at completion.” *International Journal of Project Management*, 32(6), 1007–1018.

Narbaev, T. and De Marco, A. (2014b). “Combination of Growth Model and Earned Schedule to Forecast Project Cost at Completion." *Journal of Construction Engineering and Management*, 140(1), 04013038.

Narbaev, T. and De Marco, A. (2017). "Earned value and cost contingency management: A framework model for risk adjusted cost forecasting." *Journal of Modern Project Management*, 4(3), 12–19.

Nelder, J. A. and Wedderburn, R. W. M. (1972). "Generalized Linear Models." *Journal of the Royal Statistical Society. Series A (General)*, 135(3), 370.

Netto, J. T., Oliveira, N. L. F. D., Freitas, A. P. A., and Santos, J. A. N. D. (2020). "Critical factors and benefits in the use of earned value management in construction." *Brazilian Journal of Operations & Production Management*, 17, 1–10.

Ngo, K. A., Lucko, G., and Ballesteros-Pérez, P. (2022). "Continuous earned value management with singularity functions for comprehensive project performance tracking and forecasting." *Automation in Construction*, 143(November).

Ottaviani, F. M. and De Marco, A. (2022). "Multiple Linear Regression Model for Improved Project Cost Forecasting." *Procedia Computer Science*, 196, 808–815.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., et al. (2011). "Scikit-learn: Machine learning in python." *Journal of Machine Learning Research*, 12, 2825–2830.

PMI (2005). *Practice Standard for Earned Value Management*. Project Management Institute, Newton Square, PA.

Putnam, L. (1978). "A General Empirical Solution to the Macro Software Sizing and Estimating Problem." *IEEE Transactions on Software Engineering*, SE-4(4), 345–361.

Reback, J., Jbrockmendel, McKinney, W., den Bossche, J. V., et al. (2021). "pandas-dev/pandas: Pandas 1.3.4." *Report no.* (oct).

Rezakhani, P. (2020). "Hybrid fuzzy-Bayesian decision support tool for dynamic project scheduling and control under uncertainty." *International Journal of Construction Management*, 22(15), 2864–2876.

Rifkin, R. M. and Lippert, R. A. (2007). "Notes on regularized least squares." *DSpace@MIT*, 10.

Rubinstein, R., Zibulevsky, M., and Elad, M. (2008). "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit." *CS Technion*, 40.

Santos, J. I., Pereda, M., Ahedo, V., and Galán, J. M. (2023). "Explainable machine learning for project management control." *Computers and Industrial Engineering*, 180(March).

Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L. (2000). "New Support Vector Algorithms." *Neural*

Computation, 12(5), 1207–1245.

Simion, C. P. and Marin, I. (2018). "Project cost estimate at completion: Earned value management versus earned schedule-based regression models. a comparative analysis of the models application in the construction projects in romania." *Economic Computation and Economic Cybernetics Studies and Research*, 52(3), 205–216.

Taboada, I., Daneshpajouh, A., Toledo, N., and de Vass, T. (2023). "Artificial Intelligence Enabled Project Management: A Systematic Literature Review." *Applied Sciences (Switzerland)*, 13(8).

Tariq, S., Ahmad, N., Usman Ashraf, M., Alghamdi, A. M., et al. (2020). "Measuring the Impact of Scope Changes on Project Plan Using EVM." *IEEE Access*, 8, 154589–154613.

Tipping, M. E. (2001). "Sparse Bayesian Learning and the Relevance Vector Machine." *Journal of Machine Learning Research*, 1(3), 211–244.

Uddin, S., Ong, S., and Lu, H. (2022). "Machine learning in project analytics: a data-driven framework and case study." *Scientific Reports*, 12(1), 15252.

Vanhoucke, M. (2010). "Using activity sensitivity and network topology information to monitor project time performance." *Omega*, 38(5), 359–370.

Vanhoucke, M. (2012). *Project Management with Dynamic Scheduling*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Vanhoucke, M. (2023). *The Illusion of Control: Project Data, Computer Algorithms and Human Intuition for Project Management and Control*. Springer, 1 edition.

Vanhoucke, M., Coelho, J., and Batselier, J. (2016). "An overview of project data for integrated project management and control." *Journal of Modern Project Management*, 3(3), 6–21.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., et al. (2020). "SciPy 1.0: fundamental algorithms for scientific computing in Python." *Nature Methods*, 17(3), 261–272.

Warburton, R. D. (2011). "A time-dependent earned value model for software projects." *International Journal of Project Management*, 29(8), 1082–1090.

Warburton, R. D. and Cioffi, D. F. (2016). "Estimating a project's earned and final duration." *International Journal of Project Management*, 34(8), 1493–1504.

Warburton, R. D., De Marco, A., and Sciuto, F. (2017). "Earned schedule formulation using nonlinear cost estimates at completion." *Journal of Modern Project Management*, 5(1), 75–81.

Warburton, R. D., Ottaviani, F. M., and De Marco, A. (in press). "Critical analysis of linear and nonlinear project duration forecasting methods." *Journal of Modern Project Management*.

Wauters, M. and Vanhoucke, M. (2014). "Support Vector Machine Regression for project control forecasting." *Automation in Construction*, 47, 92–106.

Wauters, M. and Vanhoucke, M. (2016). "A comparative study of Artificial Intelligence methods for project duration forecasting." *Expert Systems with Applications*, 46, 249–261.

Wauters, M. and Vanhoucke, M. (2017). "A Nearest Neighbour extension to project duration forecasting with Artificial Intelligence." *European Journal of Operational Research*, 259(3), 1097–1111.

Willems, L. L. and Vanhoucke, M. (2015). "Classification of articles and journals on project control and earned value management." *International Journal of Project Management*, 33(7), 1610–1634.

Wipf, D. and Nagarajan, S. (2007). "A new view of automatic relevance determination." *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis, eds., Vol. 20, Curran Associates, Inc.

Zhao, M. and Zi, X. (2021). "Using Earned Value Management with exponential smoothing technique to forecast project cost." *Journal of Physics: Conference Series*, 1955(1), 012101.