



Delft University of Technology

Document Version

Final published version

Licence

CC BY-NC-ND

Citation (APA)

Sari, M., Berawi, M. A., Imama, M. D. R., Saroji, G., & Miraj, P. (2026). Machine learning prediction of estimate at completion for infrastructure projects in Nusantara Indonesia. *Discover Applied Sciences*, 8(8), Article 921. <https://doi.org/10.1007/s42452-026-08965-8>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

In case the licence states "Dutch Copyright Act (Article 25fa)", this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

RESEARCH

Open Access



Machine learning prediction of estimate at completion for infrastructure projects in Nusantara Indonesia

Mustika Sari^{1,2}, Mohammed Ali Berawi^{1,2*}, Muhammad Daffa Reyhando Imama^{1,2}, Gunawan Saroji³ and Perdana Miraj⁴

*Correspondence:

Mohammed Ali Berawi
berawi@ui.ac.id

¹Department of Civil and Environmental Engineering, Faculty of Engineering, Universitas Indonesia, Depok 16424, Indonesia

²Center for Sustainable Infrastructure Development, Faculty of Engineering, Universitas Indonesia, Depok 16424, Indonesia

³Department of Civil Engineering Politeknik Negeri Bengkalis, Sungai Alam, Bengkalis 28734, Indonesia

⁴Design and Construction Management, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, Netherlands

Abstract

Accurate forecasting of Estimate at Completion (EAC) is essential for monitoring large infrastructure projects, particularly when project conditions change during execution. Conventional Earned Value Management (EVM) formulas estimate future cost through fixed relationships between cost and schedule indicators, which may produce unstable predictions when project performance fluctuates across reporting periods. This study evaluates the use of machine learning (ML) models to support EAC forecasting using project monitoring data. Three prediction models were examined, namely Linear Regression as a baseline statistical model, as well as Neural Network and Deep Learning as nonlinear models that capture complex relationships among monitoring indicators. The models were trained using financial and progress variables derived from e-monitoring records, the performance of which was evaluated using Mean Absolute Percentage Error (MAPE) that represents deviation between model outputs and the reference forecast values used in the monitoring system. The results show differences in deviation across the models, where the Neural Network produced the lowest deviation with a MAPE of 2.26%, while Linear Regression and Deep Learning produced larger deviations. The trained model was then applied to an active road project in Nusantara, Indonesia's new capital city, where it predicted an EAC of IDR 666 billion compared with the Budget at Completion of IDR 811 billion. These findings indicate that ML models reproduce forecast patterns embedded in monitoring data and generate responsive forecast signals that follow variations in monitoring indicators.

Keywords Machine learning, Estimate at completion, Construction cost forecasting, Earned value management, Infrastructure projects, Predictive analytics

1 Introduction

Smart cities operate through interconnected infrastructure systems that rely on real-time data, digital coordination platforms, and predictive analytics to maintain functionality and resilience. The development of these systems depends on precise and reliable delivery of public infrastructure to support seamless urban services, connectivity, and



© The Author(s) 2026. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

sustainability [1]. Infrastructure delivery operates as part of an intelligent urban control system and does not function as a standalone engineering activity. Smart city projects must support integrated planning systems that include automated monitoring and adaptive management mechanisms [2, 3]. Forecasting tools are planning aids that can influence procurement sequencing, contractor oversight, and government budget revisions across interdependent projects. In large-scale smart city programs, inaccurate cost forecasts affect multi-project budget coordination, institutional accountability, and public trust. When cost deviations are detected late, corrective action becomes difficult and financial pressure spreads across related infrastructure packages; hence, early and reliable forecasting is important in infrastructure governance.

Nusantara, Indonesia's new capital city, represents a national example of this transformation. Conceived as a technology-driven administrative and innovation hub, the city is planned with digitally connected infrastructure to support long-term national competitiveness [4]. Strategic investment sectors include transport corridors, energy, utilities, and civic assets that must meet cost, schedule, and planning constraints while feeding performance updates into citywide coordination platforms. Delivery in high-pressure environments such as Nusantara must remain accurate and financially responsible due to visibility at the national level [5, 6].

Budget overruns remain a persistent problem [7]. More than 50% of large infrastructure projects exceed planned costs due to fragmented coordination and slow detection of deviations during construction [8]. Nusantara's early-stage investments face this same exposure. In Indonesia and comparable economies, EAC forecasting errors in public infrastructure projects can exceed 20% when based solely on traditional methods [9]. Earned Value Management (EVM) is widely used to monitor project performance. It integrates scope, schedule, and cost through indicators such as Planned Value (PV), Earned Value (EV), Actual Cost (AC), and Budget at Completion (BAC). EVM produces Estimate at Completion (EAC) using fixed mathematical relationships between these indicators, which extend current performance trends into future projections [10]. However, they rely on stable assumptions and lagging indicators, which limits their responsiveness when project conditions change.

Machine Learning (ML) has been introduced as a data-driven alternative that can capture nonlinear relationships among project variables [11]. Prior studies report improved prediction capability using models such as Artificial Neural Networks (ANN) and Support Vector Machine (SVM) [12, 13]. Recent studies have also explored ensemble and gradient-boosting techniques. However, most studies rely on completed project datasets and retrospective evaluation that used full historical information that is not available during active project execution. As a result, their relevance for real-time monitoring conditions remains unclear.

A key limitation in current research lies in the absence of operational evaluation. Existing studies rarely examine how forecasting models function within routine monitoring workflows or how forecast outputs behave under evolving project conditions. In addition, comparisons between ML and EVM are typically conducted using historical datasets, without assessing their behaviour in active projects. This gap limits understanding of how predictive models can support decision-making during project execution.

In this study, the target variable used for model training is defined as a reference forecast value derived from the standard EVM formulation used in project monitoring

systems. This reference does not represent the realized final project cost, but it reflects the forecast values that are routinely used in operational monitoring environments. Therefore, the model learns the relationship between weekly monitoring indicators and this reference forecast signal. The objective is to examine whether ML can produce a stable and responsive forecasting signal within the same monitoring context, without reconstructing completion cost outcomes.

This study evaluates ML models using live weekly financial and physical progress data from an ongoing infrastructure project in Nusantara. The deployment case demonstrates how the model operates within an active project monitoring environment and produces forecast signals derived from ongoing project data.

The study positions ML as a supplementary forecasting tool for EAC within infrastructure project monitoring systems. It examines how ML models learn patterns from routine monitoring indicators used in EVM-based systems and describes how forecast signals differ across methods under live project conditions. It also demonstrates the feasibility of integrating ML-based forecasting into public infrastructure monitoring workflows. This perspective places EAC forecasting within an analytical layer of project monitoring systems that supports ongoing evaluation during project execution.

2 Literature study

2.1 Forecasting infrastructure in the context of smart city governance

Forecasting the performance of public infrastructure has become a critical part of smart city operations [14]. In digitally managed urban systems, cost prediction tools are no longer isolated project management instruments [15], they are part of the operational data layer that supports responsive decision-making, financial accountability, and service delivery. EAC, as a forward-looking financial indicator, supports coordination across procurement, scheduling, and risk mitigation systems in high-stakes public projects [16].

Smart cities integrate infrastructure planning into broader digital ecosystems, including real-time monitoring dashboards, geospatial data systems, and city management platforms. In this context, project forecasting requires tools that can interpret live data and produce outputs compatible with automated reporting or predictive controls. Manual and retrospective methods are no longer adequate, particularly in national-scale initiatives where project performance affects institutional credibility and operational continuity.

Although there is growing recognition of the value of predictive tools in smart city planning, many forecasting models remain disconnected from real-time operations. Prior studies rarely use operational monitoring data that originate from routine project reporting systems, and most evaluations rely on retrospective datasets instead of live project conditions. As a result, existing research provides limited insight into how forecasting models behave when applied to evolving monitoring data during project execution.

Moreover, the literature demonstrates advancement in smart infrastructure monitoring [17, 18], yet empirical validation of live EAC forecasting within operational smart city governance systems remains limited. There is also limited evidence on how different forecasting methods generate and interpret forecast signals when applied to the same monitoring data. Most studies focus on accuracy metrics and do not examine how

forecast outputs change across monitoring periods. Therefore, this study positions EAC forecasting within that broader system. Accurate and adaptable forecasting models contribute to smart infrastructure governance. They support decisions that extend beyond the project site. These decisions include adjustments to resource allocations, contract management, and timeline sequencing across a portfolio of interdependent public works.

2.2 From static to predictive models: EVM and ML

EVM remains the dominant method for project performance tracking in large-scale construction. Despite its widespread application, EVM provides lagging indicators and relies on static performance trends. These assumptions are unsuitable for projects with nonlinear workflows or variable field conditions. EVM also lacks compatibility with real-time data flows and digital oversight platforms used in smart city programs [11].

EAC refers to the projected total project cost at completion. BAC represents the total approved project budget. EV is the budgeted value of completed work at a specific time. CPI measures cost efficiency and is calculated as EV divided by AC. SPI measures schedule efficiency and is calculated as EV divided by PV. The most common EAC formulations are expressed as follows:

$$EAC_1 = BAC / CPI \quad (1)$$

$$EAC_2 = AC + (BAC - EV) \quad (2)$$

$$EAC_3 = AC + (BAC - EV) / (CPI \times SPI) \quad (3)$$

Where EAC_1 assumes that current cost efficiency continues for the remaining work, EAC_2 assumes that remaining work follows the original budget plan, and EAC_3 assumes that both cost and schedule efficiency continue consistently.

Few studies critically examine the structural limitations of EVM under dynamic project conditions. Most studies either apply EVM uncritically or use it as a benchmark without addressing its technical constraints in dynamic environments. As a result, EVM's actual contribution to predictive project control is overstated in many analyses, especially in studies that do not test its performance against live data.

ML offers a data-driven alternative that can capture nonlinear relationships among project variables [19]. In construction, ML has been applied across multiple domains, including safety, resilience, equipment management, and sustainability through ensemble models [20, 21], hybrid ML for climate resilience [22], ML-based fragility modelling [23–25], BIM-integrated XGBoost systems [26], and predictive analytics for occupational safety and productivity optimization [27, 28].

In terms of project cost prediction, Mahmoodzadeh et al. [12] applied a hybrid model using Gaussian Process Regression, Particle Swarm Optimization, and Grey Wolf Optimizer to predict tunnel project construction cost. Mohammed et al. [13] used an ANN trained on SPI and CPI indicators to estimate housing project outcomes [20]. Abed et al. [29] and Tijanić et al. [30] reported that ML models can capture complex relationships in construction data when sufficient and consistent datasets are available.

A key methodological issue across these studies lies in variable selection. Some models exclude cost baselines such as BAC, while others rely only on SPI and CPI without

considering absolute disbursements or physical progress indicators. Few studies provide a rationale for including or excluding specific variables, despite their operational importance in infrastructure monitoring systems. This weakens the reproducibility and practical utility of the models.

Despite the growing volume of ML-based forecasting research, most of the reviewed studies are not operationally useful in live environments. They are trained on historical or completed project datasets and do not incorporate real-time performance signals or interface with system-level infrastructure platforms. Few include variables directly linked to live physical progress, and almost none compare ML forecasts with EVM outputs in real-world conditions. This gap limits understanding of how ML-based forecasts behave within active monitoring systems.

2.3 Algorithm selection and research gap addressed

This study selected three algorithms that cover a wide range of characteristics relevant to infrastructure forecasting: Linear Regression (LR), Neural Network (NN), and Deep Learning (DL). LR serves as a baseline model, since it is simple and interpretable when relationships among inputs are linear. NN models capture nonlinear interactions that frequently arise in infrastructure settings, while DL extends NN capacity through multiple layers and can identify complex patterns across time-series data [31].

The selection criteria include interpretability, computational efficiency, scalability, and compatibility with digital monitoring platforms used in smart city governance. These criteria are relevant in operational settings where models must process periodic updates and produce outputs that align with public reporting systems.

Other algorithms such as SVM, Random Forest, and XGBoost were excluded due to operational barriers. Algorithms that require extensive feature engineering or high computational overhead may reduce feasibility in public monitoring systems where updates occur on a weekly basis. In addition, highly complex ensemble models can limit transparency for decision-makers who require understandable forecasting logic.

For this application, model interpretability and processing speed were prioritized over potential increases in predictive complexity. What distinguishes this study is its use of live project data from an active public works site. The study examines how forecasting outputs behave under operational monitoring conditions and compares ML-based forecasts with EVM-based estimates within the same data environment. The forecasting framework is designed for integration in smart city governance structures.

The model was developed to function within an operational setting that requires periodic data input, output compatibility with reporting platforms, and alignment with procurement and payment systems. This study evaluates the three algorithms using live project data from a strategic road corridor in Nusantara and examines how their forecast outputs compare with EVM-based estimates under the same monitoring conditions. This setup addresses the use of operational monitoring data, evaluation of forecast behavior under evolving conditions, and integration of predictive models into monitoring workflow. Prior studies on construction cost forecasting are summarized in Table 1.

The reviewed studies demonstrate progress in predictive modeling for construction management through ML and data driven forecasting techniques. Most studies rely on historical datasets and offline model validation within controlled research settings. Applications that use live financial and progress indicators within operational

Table 1 Summary of ML studies for construction cost forecasting

No.	Authors and year	Project type	Algorithms	Dataset	Key limitation
1	Mahmoodzadeh et al. [12]	Tunnel	GPR, PSO, GWO	Completed	No live data
2	Mohammed et al. [13]	Housing	ANN	Historical SPI/CPI	No benchmark
3	Abd et al. [31]	Road	ML (unspecified)	Progress + Contract	Static data only
4	Abed et al. [29]	Infrastructure	ML (various)	Multi-project	Needs data completeness
5	Tijanić et al. [30]	Public Works	ML (various)	Mixed	Lacks real-time inputs

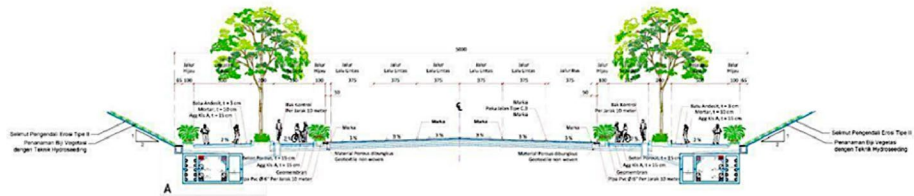


Fig. 1 Cross-sectional sketch of the national axis road

monitoring environments remain limited. This study addresses this gap by evaluating predictive models using operational data and examining how forecast outputs behave within an active project monitoring environment.

3 Methods

3.1 Case context and data sources

The case project used for model deployment is the Western Side National Axis Road Phase II in Nusantara. The project spans approximately 2.99 km with a cross-section width of 50 m and includes a Multi Utility Tunnel that accommodates electricity, telecommunications, water, and other underground infrastructure. The total contract value is IDR 811.68 billion. Figure 1 presents the cross-sectional configuration of the road corridor and the Multi Utility Tunnel.

Project monitoring data were obtained from the national e-monitoring system used by Indonesia’s Ministry of Public Works. The system records weekly construction data that include financial disbursement, physical progress, and key EVM indicators such as CPI and SPI. These records represent routine operational data used by public agencies to supervise infrastructure projects. The observations correspond to early and middle construction stages, which are critical for forward cost forecasting.

To construct the training dataset, historical monitoring records were compiled from three completed infrastructure projects in Jambi Province, including a road development project in Talang Duku, a canal construction project in Kedaton, and a road rehabilitation project near Sungai Gelam. These projects provide complete weekly monitoring records across full construction cycles, within 100 weeks of observations in total. Verified final cost outcomes are available for reference but are not used as supervised training labels.

The historical datasets were used for model training and validation, while the Nusantara project represents an ongoing infrastructure project and was reserved for model deployment. This separation prevents data leakage and supports evaluation under real monitoring conditions.

Each dataset entry represents a weekly observation from the project monitoring system, consisting of eight EVM-based input variables and a target variable defined as the projected EAC derived from the conventional EVM formulation, which serves as the reference forecast signal. The feature vector for each observation is defined as:

$$X = \{EV, PV, BAC, AC, \text{planned progress}, \text{actual progress}, CPI, SPI\} \quad (4)$$

where EV, PV, BAC, and AC denote standard EVM cost indicators, while planned and actual progress represent percentage completion, and CPI and SPI denote cost and schedule performance indices.

The target variable for each dataset entry corresponds to the projected EAC, calculated using the conventional EVM forecasting formulation (Eq. 2). This value corresponds to a reference forecast signal within the monitoring system and is separate from the final project cost outcome. Each observation therefore consists of a feature vector X and a target variable Y where:

$$Y = EAC \quad (5)$$

This formulation defines a learning structure in which the model captures relationships between monitoring indicators and the reference forecast signal across reporting periods.

Table 2 presents an example of the dataset structure. The observations represent a single project segment in which the target EAC value remains constant across rows because it corresponds to a project-level reference forecast endpoint. The input indicators vary across weeks as project conditions evolve. This structure reflects monitoring practice, where multiple time-based observations are associated with one forecast reference value.

The dataset provides weekly snapshots of financial and progress conditions. These observations capture variation in monitoring indicators across reporting periods and support analysis of how forecast signals relate to changes in project performance.

3.2 Feature selection and scenario design

3.2.1 Feature selection

The selected variables are derived from standard EVM-based monitoring systems and represent key aspects of project performance, including financial status, physical progress, and performance efficiency. These indicators are widely used in construction monitoring practice and have been applied in prior studies on EAC forecasting [32, 33].

The variables are grouped into two categories. Financial indicators, which include EV, PV, AC, and BAC, describe the monetary condition of the project. Progress and efficiency indicators, which include planned progress, actual progress, CPI, and SPI,

Table 2 Dataset structure

Week	EV (IDR bn)	PV (IDR bn)	AC (IDR bn)	BAC (IDR bn)	Planned Progress (%)	Actual Progress (%)	CPI	SPI	Target EAC (IDR bn)
12	4.075	3.999	3.120	5.700	70.16	71.49	1.306067	1.018957	4.400
13	5.073	5.015	3.860	5.700	87.99	89.00	1.314249	1.011479	4.400
14	5.622	5.618	4.320	5.700	98.56	98.64	1.301500	1.000811	4.400

Cost values are expressed in IDR billion

represent schedule status and cost performance. Together, these variables capture the monitoring conditions used in EVM-based cost forecasting models.

3.2.2 Data preprocessing and normalization

All numerical variables were standardized to maintain consistent measurement scales across the dataset. Cost variables were expressed in IDR billion to maintain numerical stability in the training process. Progress indicators were converted to percentage values representing the proportion of completed work.

Min-max normalization was applied to all numeric features to transform the variables into a comparable numerical range. This normalization reduces scale dominance among cost and progress variables and improves convergence during model training. Outliers were screened using interquartile range thresholds. Observations with CPI and SPI values outside the interval of 0.2 to 2.5 were reviewed against the original monitoring reports to verify whether the values represented real project conditions or recording errors.

3.2.3 Synthetic data augmentation

The historical monitoring dataset contained a limited number of observations because infrastructure monitoring data are recorded at weekly intervals. Synthetic observations were generated based on realistic combinations of cost and schedule performance indicators to expand variation in cost-performance patterns within the training dataset.

Synthetic samples were constructed to increase model exposure to variation in cost-performance patterns [34], using four combinations of CPI and SPI regimes that represent common project dynamics observed in construction monitoring systems, including:

1. $SPI > 1$, $CPI < 1$, representing accelerated progress with cost overrun.
2. $SPI < 1$, $CPI < 1$, representing delayed progress with cost overrun.
3. $SPI > 1$, $CPI > 1$, representing accelerated progress with cost savings.
4. $SPI < 1$, $CPI > 1$, representing delayed progress with cost savings.

The synthetic data samples follow the logical relationships defined by EVM variables and remain consistent with the structure of monitoring records. They do not introduce new target definitions and preserve the original monitoring logic.

The Nusantara project dataset was excluded from the augmentation process and from all model training and validation stages. These observations were reserved exclusively for deployment testing to prevent data leakage between model development and operational evaluation. Consequently, augmented samples were used only within the training dataset, while the testing dataset and the deployment case project consisted exclusively of original project monitoring records.

3.2.4 Correlation analysis

The correlation analysis is used as an exploratory step to examine relationships within the monitoring data. It provides an initial view of how variables move together across observations. The analysis does not establish causal relationships and is not used as the basis for model selection or predictive validity. The results of this analysis support the understanding of variable interactions within the dataset and provide context for interpreting model behaviour in later sections.

3.3 Model development and training protocol

Three ML algorithms were developed and compared in this study: Linear Regression (LR), Neural Network (NN), and Deep Learning (DL). LR was used as a baseline model to represent linear relationships among monitoring indicators and project cost outcomes. NN was introduced to capture nonlinear interactions between schedule and cost indicators that may influence EAC prediction. DL was further tested to examine whether deeper network architectures improve predictive performance when complex relationships exist within project monitoring data.

The modelling process used the same set of input variables across all algorithms. Each observation contains the corresponding eight input features, which become the number of the input dimension for all models. The cleaned dataset was partitioned into training used for model fitting and testing subsets to evaluate model performance against unseen observations with an 80% to 20% split.

Model validation was conducted using two-fold cross validation within the training dataset. The training data were divided into two subsets. Each subset was used once for training and once for validation. The average validation error across the folds was used to assess model stability. This procedure reduces overfitting risk and evaluates model stability when exposed to different subsets of the training data.

All models were implemented using RapidMiner Studio, a visual ML platform that provides modular operators for data preparation, model training, and performance evaluation. It was selected because it supports transparent model workflows.

The LR model was implemented using an ordinary least squares estimator that models the linear relationship between the selected monitoring indicators and the predicted EAC value. While the NN model was implemented using a feedforward neural network architecture with one hidden layer. The input layer contains eight neurons corresponding to the monitoring indicators. The hidden layer consists of ten neurons with sigmoid activation functions. The output layer contains one neuron that produces the predicted EAC value using a linear activation function.

Moreover, the DL model extends this structure by using three hidden layers with sizes of 16, 12, and 8 neurons, respectively. Each hidden layer uses a rectified linear unit activation function. The output layer uses a linear activation function to generate the predicted EAC value. Both NN and DL models were trained using backpropagation with a stochastic gradient descent optimizer available in RapidMiner. The loss function used for all models is mean squared error, which measures the difference between predicted values and the reference forecast label.

The training process was conducted with a maximum of 200 epochs. Early stopping was applied based on validation error. Training stopped when validation error did not decrease for ten consecutive iterations. The batch size was set to 16 observations. The random initialization of network weights followed the default settings of RapidMiner, and no fixed random seed was specified.

The RapidMiner operators used in the workflow include “Normalize” for feature scaling, “Split Data” for dataset partitioning, “Cross Validation” for model validation, “Linear Regression” for LR implementation, “Neural Net” for NN, and “Deep Learning” for DL. These operators define the full modelling pipeline from data preparation to prediction output. Figure 2 shows the operator process for the EAC prediction using NN, DL, and LR algorithm.

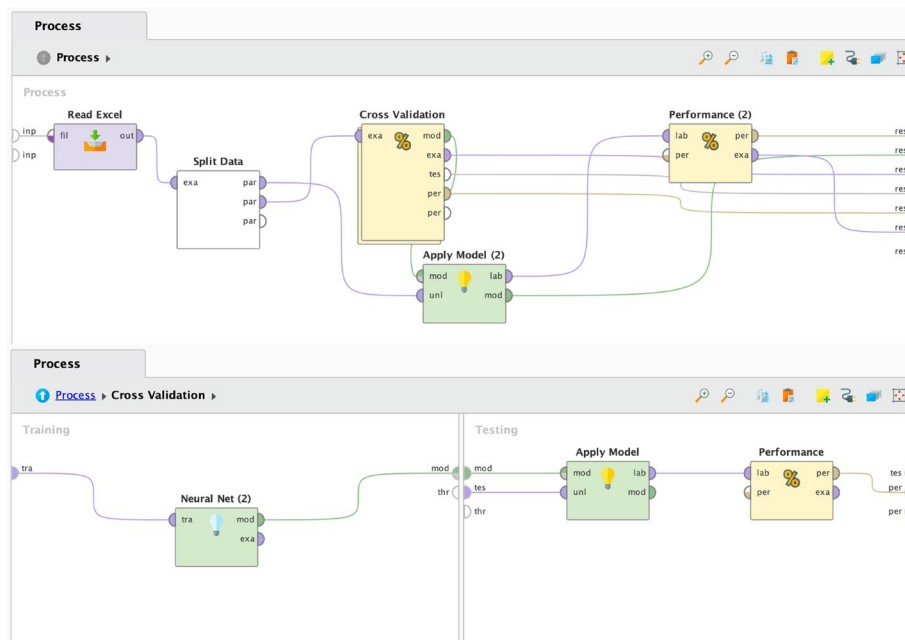


Fig. 2 ML model development workflow for EAC prediction

Model performance during training and validation was monitored using training error and validation error values across iterations. Overfitting was assessed by comparing training error and validation error trends. A large gap between these values indicates overfitting, while consistently high errors indicate underfitting. Training and validation loss values were recorded across epochs to assess model stability and detect potential overfitting or underfitting conditions. All models were trained using identical dataset partitions, input variables, and evaluation procedures.

3.4 Model deployment

The trained NN model was deployed using project monitoring data from the Nusantara road project. This dataset was not used in any stage of model training or validation and therefore represents an independent operational case for model application. Input variables were prepared using the same preprocessing procedures and feature structure applied during model training to maintain consistency in the prediction pipeline.

The deployment exercise demonstrates how the trained model can be applied to ongoing project monitoring data to generate early forecasts of the EAC. The purpose of this deployment is to examine how the model produces forecast outputs under real monitoring conditions and how these outputs evolve across reporting periods. The deployment is treated as an operational illustration of model use within a monitoring workflow, not as a validation of forecast superiority or accuracy.

The generated predictions provide a comparative forecast signal that can be examined alongside EVM-based estimates under the same monitoring inputs. This comparison is intended to support interpretation of forecast behavior during project execution.

3.5 Evaluation and final integration

Model performance was evaluated using Mean Absolute Percentage Error (MAPE), selected because it can express prediction error in percentage terms, which improves interpretability for project cost forecasting. The MAPE metric is defined as:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{EAC_i^{actual} - EAC_i^{predicted}}{EAC_i^{actual}} \right| \times 100 \quad (6)$$

where n represents the number of observations, EAC_i^{actual} denotes the reference forecast value derived from the standard EVM formulation, and $EAC_i^{predicted}$ represents the model output. The reference value used in this evaluation does not represent the realized final project cost. Table 3 lists the MAPE interpretation thresholds based on Lewis [35]. These categories are used as general indicators of prediction error magnitude within the dataset and are not used to infer real-world forecast accuracy.

The evaluation is conducted in two stages. The first stage uses training and testing datasets to assess how well the models reproduce the reference forecast values derived from monitoring data. The MAPE values in this stage indicate the degree of alignment between model outputs and the reference forecast signal.

The second stage corresponds to model deployment using the Nusantara project dataset. In this stage, ML-generated forecasts and EVM-based estimates are compared under the same monitoring inputs. This comparison is used to examine differences in forecast behavior across methods and does not indicate which method produces more accurate or reliable results, because the realized final project cost is not available for the case study. The comparison is limited to observing how forecast outputs evolve during project execution.

Model interpretability and deployment feasibility were considered in addition to MAPE during algorithm selection. These factors included training complexity, prediction stability, and compatibility with real-time input updates. The Neural Network model met these criteria and was integrated into the final workflow.

The selected model was deployed using a separate input file from the National Axis Road case project. All preprocessing steps followed the same structure used during model development. The prediction outputs were stored for later comparison with conventional EVM-based formulas in the results section. Figure 3 presents the research workflow from data preparation through model development, evaluation, and case study deployment.

3.6 Methodological limitations

This study is subject to several data-related limitations. The available dataset is relatively small due to limited access to structured monitoring records in public infrastructure projects. Synthetic data augmentation was applied to increase variation within the training dataset. The generated samples follow the logical structure of EVM variables, but

Table 3 Classification of performance based on MAPE

MAPE (%)	Interpretation
< 10	Strong alignment
10–20	Moderate alignment
20–50	Weak alignment
> 50	Large deviation from reference signal

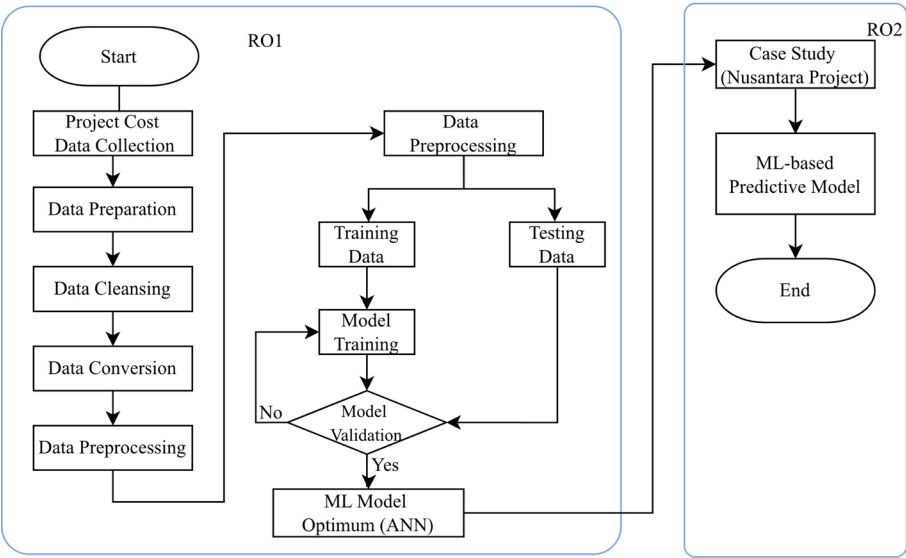


Fig. 3 Research workflow

Table 4 Performance comparison of ML models

No.	Algorithm	MAPE (%)
1	NN	2.26
2	LR	7.69
3	DL	15.46

they may not capture the full range of conditions observed in real project environments. In addition, the dataset structure follows reporting practices used in Indonesian government monitoring systems, which may differ from private sector or international project systems.

The supervised target variable represents a reference forecast value derived from standard EVM formulation used in monitoring systems. The model learns patterns associated with these forecast signals within the dataset and does not represent validation against final project cost outcomes. The outputs therefore correspond to the behavior of forecast values within the monitoring context.

4 Results and discussion

4.1 Model performance on training and testing data

The predictive models estimate the projected EAC using the defined monitoring indicators. Each weekly monitoring record represents one observation consisting of the input feature vector and its corresponding target value for EAC. The target value serves as the regression label during training. The models learn the relationship between project performance indicators observed at a specific monitoring period and the reference forecast value used in monitoring systems.

Table 4 summarizes the prediction deviation of the three ML algorithms evaluated in this study. Model performance was assessed using MAPE, where the NN, LR, and DL models achieved the lowest prediction error with a MAPE value of 2.26%, 7.69%, and 15.46%, respectively. Based on the classification thresholds, NN and LR fall within the “strong alignment” category, while DL falls within the “moderate alignment” category.

To provide a baseline comparison within the same dataset, the EVM-based estimation formula was applied to the testing dataset. The EVM forecast produced a MAPE value of 19.41%, which represents the deviation of the conventional formulation from the reference forecast pattern. This comparison is internal to the dataset. The ML models reproduce the reference forecast signal through learned relationships, while the EVM formulation represents a direct deterministic calculation. The comparison then shows how each method aligns with the same reference signal, without serving as an evaluation of forecasting performance. All ML models show lower deviation relative to the reference forecast signal compared to the EVM formulation within the dataset. This comparison reflects differences in alignment with the same reference signal across methods.

The differences in deviation values indicate how each model represents relationships among project monitoring indicators. LR estimates EAC using a linear function and cannot represent nonlinear interactions among cost and progress variables. DL introduces deeper network layers that increase model flexibility, yet this complexity can reduce training stability when the available dataset is limited. In contrast, the NN model provides sufficient nonlinear representation while maintaining stable training behaviour. Cross validation during training helped verify that the model did not exhibit significant overfitting or underfitting patterns across the dataset partitions. Furthermore, training and validation error values were monitored across epochs to assess model stability. The RapidMiner training process showed comparable training and validation error behaviour across the evaluated folds, which indicates stable model learning under the dataset conditions. Additional comparison between approximate training and testing MAPE values further supports this behaviour. The NN model produced approximate training and testing MAPE values of 2.0% and 2.26%, respectively, which indicates stable generalization across unseen observations. The LR model showed comparable training and testing deviations of 7.1% and 7.69%, while the DL model produced higher deviations of 12.8% and 15.46%. These results suggest that additional network complexity did not improve prediction stability for the available dataset size.

Based on these results, the NN model shows more stable alignment with the reference forecast signal across observations within the dataset and is selected for deployment in the operational case project described in the following subsection.

4.2 Feature correlation and model input analysis

Correlation analysis was conducted to examine relationships among the selected monitoring indicators and the target variable EAC within the dataset. Table 5 presents the Pearson correlation matrix among these variables and the projected EAC value.

The results indicate that EV, PV, AC, and planned progress exhibit extremely strong correlations with one another, with coefficients exceeding 0.99. This pattern occurs

Table 5 Correlation matrix of monitoring indicators and target variable

Variable	EV	PV	AC	CPI	SPI	Planned progress	EAC
EV	1.00	0.998	0.999	0.094	− 0.862	0.999	0.284
PV	0.998	1.00	0.997	0.112	− 0.839	0.998	0.282
AC	0.999	0.997	1.00	0.086	− 0.873	0.998	0.286
CPI	0.094	0.112	0.086	1.00	− 0.026	0.099	− 0.363
SPI	− 0.862	− 0.839	− 0.873	− 0.026	1.00	− 0.855	− 0.092
Planned progress	0.999	0.998	0.998	0.099	− 0.855	1.00	0.285
EAC	0.284	0.282	0.286	− 0.363	− 0.092	0.285	1.00

because these indicators represent cumulative quantities derived from the same project progress trajectory and baseline cost structure. As construction advances, earned value, planned value, and cumulative expenditure increase consistently with project progress, which produces the near-perfect correlations observed among these variables. Such relationships are expected in datasets derived from EVM monitoring records.

In contrast, CPI and SPI measure performance efficiency rather than cumulative quantities. CPI reflects cost efficiency through the relationship between EV and AC, whereas SPI reflects schedule efficiency through the relationship between EV and PV. Consequently, these variables exhibit weaker relationships with the cumulative indicators. CPI shows a moderate negative correlation with EAC (-0.363), which indicates that reduced cost efficiency is associated with higher projected completion cost. SPI exhibits relatively weak correlation with EAC (-0.092), which suggests that schedule efficiency alone does not strongly determine the final cost outcome in the observed dataset.

The correlations between the monitoring indicators and the projected EAC remain moderate, with EV, PV, AC, and planned progress exhibiting coefficients around 0.28. These values indicate that the projected EAC is associated with interactions among multiple monitoring variables, not a single indicator. The correlation results describe how variables move together within the dataset and provide context for interpreting model behavior in subsequent analysis.

4.3 Deployment results for the Nusantara case project

The trained models were applied to monitoring records from the Western Side National Axis Road Phase II project in Nusantara. The results present differences in forecast outputs generated under the same monitoring conditions during project execution.

The input records came from the national e-monitoring system used for government infrastructure supervision, with the same dataset used in the model development stage. Week 15 was selected as the demonstration point because construction activity was active during this period. The trained model then produced an EAC prediction using the monitoring indicators from this dataset.

The NN model predicted an estimated cost at completion of IDR 666.55 billion. The contractual BAC for the project is IDR 811.80 billion. This difference represents a variation in forecast outputs under current monitoring conditions. Table 6 presents the monitoring indicators used as model inputs at Week 15.

Three EVM forecasting formulas were applied using the same monitoring indicators. These formulas produced EAC values between IDR 912.05 billion and IDR 1,011.02 billion. All results exceed the prediction generated by the NN model. Table 7 summarizes the comparison between the ML prediction and the conventional EVM formulas. The

Table 6 Input indicators and forecast at week 15 of the National Axis Project

No.	Indicators	Results
1	EAC Prediction	666.55 IDR billion
2	EV (% BAC)	44.01%
3	PV (% BAC)	34.63%
4	BAC	811.80 IDR billion
5	%Constructed	54.28%
6	AC	549.19 IDR billion
7	CPI (EV/AC)	0.803
8	SPI (EV/PV)	1.273

Table 7 Comparison of EAC forecasts using ML model and EVM formulas

Method	EAC results
EAC (1)	IDR 1,011.02 billion
EAC (2)	IDR 912.054 billion
EAC (3)	IDR 919.962 billion
ML prediction	IDR 666.55 billion

Cost values are expressed in IDR billion

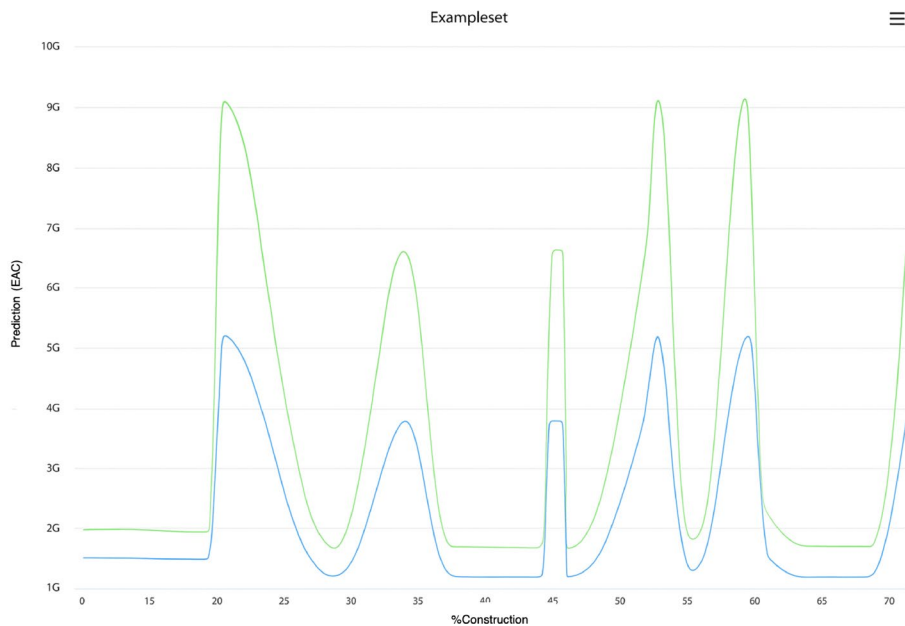


Fig. 4 EAC prediction trajectory using the Neural Network model

comparison illustrates how different forecasting methods produce distinct outputs when applied to the same monitoring inputs. The results show differences in forecast outputs across methods under the same monitoring inputs.

Differences in forecast outputs may be associated with variations in project conditions captured by the monitoring indicators. CPI and SPI values represent cost efficiency and schedule performance during the monitoring period, which influence how each method produces forecast values. Reporting lag between physical progress and financial disbursement may also affect the consistency of monitoring indicators. In addition, cumulative variables such as EV, PV, and AC introduce compounding effects across reporting periods that contribute to variation in forecast outputs. The predicted values are therefore interpreted as forecast signals within the monitoring context.

Figures 4, 5 and 6 present the prediction trajectories generated by the NN, DL, and LR models using the Nusantara monitoring dataset. The vertical axis uses RapidMiner’s scientific notation, where “G” denotes $\times 10^9$ (Giga). For example, the value “2G” on the vertical axis corresponds to IDR 2 billion and “9G” corresponds to IDR 9 billion. In contrast, labels such as “2G” or “5 C” on the horizontal axis represent identifiers for weekly monitoring records within the dataset. The green line represents the reference forecast values derived from monitoring records, while the blue line represents model predictions. Differences in trajectory patterns therefore indicate differences in prediction behavior under the same monitoring inputs. Sharp changes in EAC values occur when

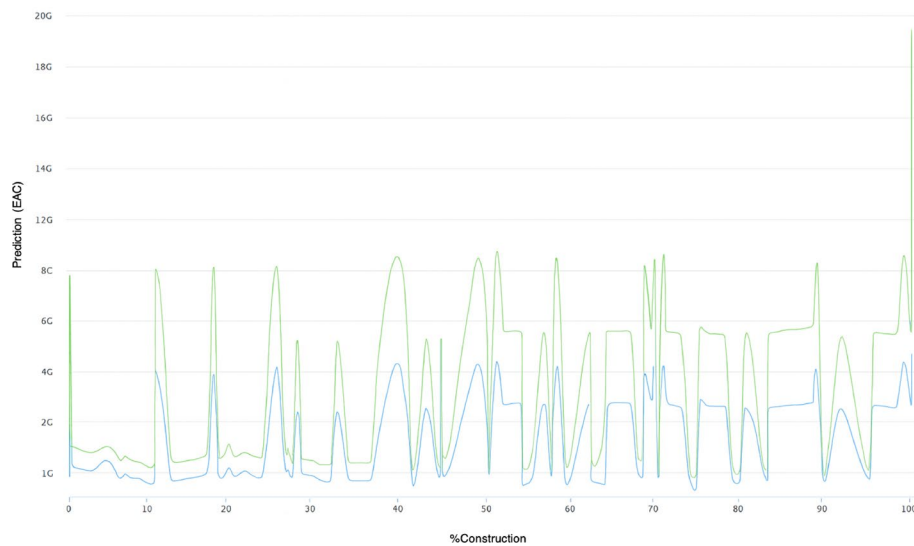


Fig. 5 EAC prediction trajectory using the Deep Learning model

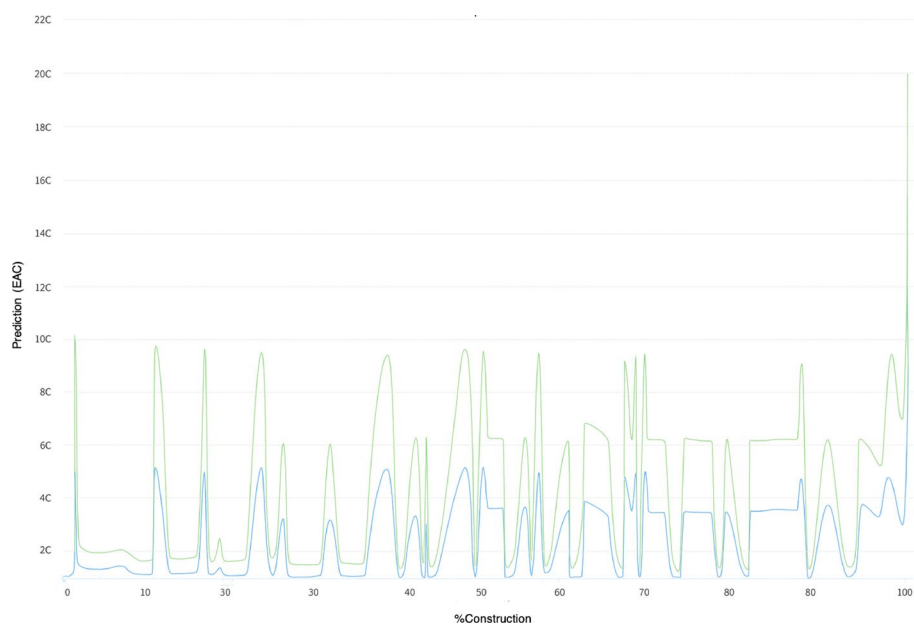


Fig. 6 EAC prediction trajectory using the Linear Regression model

changes in CPI and SPI coincide with cumulative cost updates across reporting periods. For example, a decline in CPI increases projected EAC in formulations such as $EAC_1 = BAC/CPI$, particularly during early-to-mid construction stages when the remaining budget component remains large.

The NN prediction line follows the monitoring pattern more consistently across the dataset, while DL and LR show larger deviations in several intervals. These differences become visible when CPI and SPI change across monitoring periods, which correspond to shifts in cost efficiency and schedule performance. LR exhibits systematic bias in several observations, while DL shows larger fluctuations at certain points. The variation in prediction curves is influenced by both performance indices and cumulative cost indicators, which shape the evolution of forecast outputs across monitoring periods.

4.4 Comparative discussion and research implications

4.4.1 Model behaviour and comparative performance

The comparative results indicate that the NN provides more stable forecast behavior within the dataset among the tested models. This difference suggests that project monitoring indicators interact in nonlinear ways, since construction cost development does not follow a simple proportional relationship with progress indicators. As a result, models that rely on linear assumptions struggle to represent the cost patterns that emerge during project execution.

The behaviour of the models becomes clearer when monitoring indicators change across the dataset. Cost and schedule indicators such as CPI and SPI fluctuate during project execution because physical progress, financial disbursement, and reporting timing do not always move at the same pace. Under these conditions, LR produces systematic bias because its structure assumes a fixed relationship between variables. DL did not produce stronger results in this study. The dataset size remained limited, which restricts the stability and training efficiency of deeper neural architectures.

The Nusantara case provides insight into how forecasting methods operate under operational monitoring conditions. ML models evaluate multiple monitoring indicators simultaneously and generate estimates based on interaction patterns among variables. In contrast, EVM formulas derive forecast values from fixed relationships based on CPI and SPI. These distinct formulations lead to different forecast outputs when applied to the same monitoring records.

The prediction trajectories illustrate these behavioural differences across the models. The NN maintains a consistent prediction path that follows the monitoring sequence with relatively small deviations. LR displays persistent bias in several observations, while DL shows larger fluctuations when monitoring indicators change. These patterns indicate differences in forecast stability and signal alignment under changing monitoring conditions.

4.4.2 Comparison with previous EAC prediction studies

Research on EAC forecasting has expanded beyond traditional EVM formulas toward data driven prediction methods. Earlier studies introduced statistical and ML models to estimate project cost outcomes using historical project data. One important observation from this body of research concerns the presence of nonlinear relationships among project performance indicators [36, 37]. Construction projects involve interactions among progress measurement, expenditure patterns, procurement timing, and reporting cycles. These interactions produce irregular cost patterns during project execution. Studies that apply ANN and similar ML models report the ability to capture nonlinear relationships among variables, which aligns with the behaviour observed in this study [38, 39]. The findings in this study show similar behaviour in terms of how models interpret relationships among monitoring indicators and generate forecast signals.

Another issue discussed in existing literature relates to the choice of input variables used for prediction. Some studies introduce additional project descriptors such as risk indices or organizational attributes to strengthen prediction models [40, 41]. These variables represent differences among project types and management structures. This study adopts a different perspective by focusing on monitoring indicators that already exist

within EVM-based reporting systems. This perspective supports the integration of predictive models within routine monitoring workflows.

These comparisons indicate that ML-based forecasting can complement traditional monitoring frameworks. EVM indicators remain essential for measuring project performance. Predictive models extend this framework by interpreting relationships among monitoring indicators and generating forward-looking forecast signals within the same monitoring context [37].

4.4.3 Theoretical implications for project cost forecasting

The results indicate a limitation in conventional project cost forecasting frameworks that rely on stable relationships between cost and schedule indicators. Project monitoring data exhibit dynamic interactions across reporting periods, which suggests that forecasting models must account for interdependencies among variables rather than rely on fixed relationships derived from individual indicators.

The findings also extend the analytical role of EVM indicators in project management research. Variables such as EV, PV, AC, CPI, and SPI have traditionally been used as descriptive performance measures. This study shows that these indicators can also be interpreted as a set of interrelated variables that generate forecast signals when analyzed together. Their analytical role therefore extends beyond performance tracking toward supporting structured interpretation of project conditions.

Another implication concerns the relationship between model structure and dataset characteristics. Model suitability depends on the scale and structure of monitoring data. Simpler models may fail to capture interactions among variables, while more complex architectures do not always produce stable results when data are limited. This indicates that model selection should consider both data availability and monitoring context.

The study also contributes to the theoretical understanding of forecasting within project monitoring systems. Many studies rely on completed project datasets, which do not represent active monitoring conditions. The deployment case examined here positions forecasting as an analytical layer within project monitoring systems, where models interpret ongoing monitoring data and generate forecast signals during project execution.

4.4.4 Practical interpretation and implications

The deployment results illustrate how predictive cost forecasting can support project monitoring during construction. Differences in forecast outputs provide signals that can assist project teams in identifying potential deviations in financial and progress conditions. These signals can prompt further examination of project data and support timely monitoring decisions during execution.

Forecast outputs can be used to support routine monitoring review activities. Project teams can examine cost records, verify reported progress, and assess expenditure patterns when forecast values differ from baseline expectations. For example, project authorities may conduct contract payment verification to confirm whether reported progress aligns with disbursement records. In cases where financial commitments progress faster than physical work, supervision teams may review construction schedules or adjust short-term budget allocation.

At the program level, early forecast signals support financial oversight in large infrastructure portfolios. They provide a basis for internal budget review before deviations

appear in final project accounts. When forecast outputs differ from planned budgets, project authorities can evaluate disbursement schedules, contract execution status, and progress distribution across work packages. This supports earlier identification of issues related to cost control and reporting consistency.

The results also show that predictive models can complement existing project control tools. EVM indicators remain the primary basis for performance measurement, while predictive models provide an additional layer of interpretation based on interactions among monitoring variables. The combined use of these tools supports a more structured understanding of cost conditions during project execution.

5 Conclusion

This study examined the use of ML for forecasting EAC within infrastructure project monitoring systems using operational monitoring data. The analysis focused on how models interpret relationships among cost and progress indicators within EVM-based reporting structures.

The results indicate that monitoring indicators contain interrelated patterns that can be used to generate forecast signals during project execution. Nonlinear models capture these interactions more consistently than linear formulations that rely on fixed relationships between variables. The deployment case further shows how predictive models can operate within an active monitoring environment and produce outputs based on ongoing project conditions.

This study contributes to project cost forecasting research by repositioning EVM indicators as a set of analytical variables that support signal-based interpretation within monitoring systems. It also demonstrates how ML models can function as a supplementary analytical layer within digital infrastructure monitoring environments.

Several limitations should be noted. The supervised target is defined as a reference forecast value derived from monitoring formulation, which frames the results as analysis of forecast signals within the dataset. The dataset is limited to a specific project context with a relatively small number of observations, and model behaviour may vary across different project types and monitoring systems. Future research should evaluate predictive models using broader multi-project datasets and examine the inclusion of additional variables such as schedule risk, resource variability, and external factors.

Acknowledgements

The authors would like to thank Universitas Indonesia for the support given to this research through PUTI Q2 (PKS-385/UN2.RST/HKP.05.00/2025).

Author contributions

M.S.: Writing original draft; Writing review & editing M.A.B.: Conceptualization & Supervision M.D.R.I.: Data Collection, Data Analysis, Software, Investigation, and writing G.S.: Methodology, Review P.M.: Writing review & editing.

Funding

The authors would like to thank Universitas Indonesia for the funding support given to this research through PUTI Q2 Grant (PKS-385/UN2.RST/HKP.05.00/2025).

Data availability

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable. This study did not involve any human participants or animals.

Consent for publication

Not applicable. This paper does not contain any individual person's data in any form.

Competing interests

The authors declare no competing interests.

Received: 28 November 2025 / Accepted: 2 June 2026

Published online: 08 June 2026

References

1. Das DK. Exploring the symbiotic relationship between digital transformation, infrastructure, service delivery, and governance for smart sustainable cities. *Smart Cities*. 2024;7:806–835. <https://doi.org/10.3390/SMARTCITIES7020034>.
2. Lopez LJR, Castro AIG. Sustainability and Resilience in Smart City Planning: A Review. *Sustainability*. 2020;2021(13):Page181–13181. <https://doi.org/10.3390/SU13010181>.
3. Waqar A, Barakat TAH, Almujiab HR, et al. Analytical approach to smart and sustainable city development with IoT. *Sci Rep*. 2025;15:23617. <https://doi.org/10.1038/S41598-025-08861-Y>.
4. Susantono B, Berawi MA, Sari M. Smart City Framework for Nusantara Capital City Development. In: Lytras MD, Alkhalidi A, Malik S, editors. *The Emerald Handbook of Smart Cities in the Gulf Region: Innovation, Development, Transformation, and Prosperity for Vision 2040*. Leeds: Emerald Publishing Limited; 2024. pp. 481–514.
5. Bibri SE. Data-driven smart sustainable cities of the future: urban computing and intelligence for strategic, short-term, and joined-up planning. *Comput Urban Sci*. 2021;1:8. <https://doi.org/10.1007/S43762-021-00008-9>.
6. Noori N, Hoppe T, de Jong M. Classifying pathways for smart city development: comparing design, governance and implementation in Amsterdam, Barcelona, Dubai, and Abu Dhabi. *Sustainability*. 2020;12:4030. <https://doi.org/10.3390/SU12104030>.
7. Berawi MA, Sari M, Amiri NHA, et al. Developing a Machine Learning Model to Improve the Accuracy of Owner Estimate Cost in the Capital Expenditure Procurement Process. *Int J Technol*. 2025;16:1179–89. <https://doi.org/10.14716/IJTECH.V16I4.7409>.
8. Adam A, Josephson PEB, Lindahl G. Aggregation of factors causing cost overruns and time delays in large public construction projects: trends and implications. *Eng Constr Arch Manage*. 2017;24: 393–406. <https://doi.org/10.1108/ECAM-09-2015-0135>.
9. Hamzah A, Tjendani HT, Witjaksana B. Analysis of Time Performance in Completing the Surabaya-Gempol Toll Road Reconstruction Project Using the Earned Value Method. *Asian J Eng Social Health*. 2025;4:478–87. <https://doi.org/10.46799/AJES.H.V4I3.553>.
10. Hasan R, Chowdhury S, Akter J. Construction project monitoring: the cost and schedule control by earned value method (EVM). *J Technol Manag Bus*. 2021;8:1-9 <https://doi.org/10.30880/jtmb.2021.08.01.001>.
11. Rahmaty M. Machine learning with big data to solve real-world problems. *J Data Analytics*. 2023. <https://doi.org/10.59615/jda.2.1.9>.
12. Mahmoodzadeh A, Nejati HR, Mohammadi M. Optimized machine learning modelling for predicting the construction cost and duration of tunnelling projects. *Autom Constr*. 2022;139:104305. <https://doi.org/10.1016/j.autcon.2022.104305>.
13. Mohammed SJ, Abdel-khalek HA, Hafez SM. Predicting Performance Measurement of Residential Buildings Using Machine Intelligence Techniques (MLR, ANN and SVM). *Iran J Sci Technol Trans Civil Eng*. 2022;46:3429–51. <https://doi.org/10.1007/s40996-021-00742-4>.
14. Cina E, Elbasi E, Elmazi G et al. The role of AI in predictive modelling for sustainable urban development: challenges and opportunities. *Sustainability*. 2025;17:5148. <https://doi.org/10.3390/SU17115148>.
15. Bibri SE, Huang J. Artificial intelligence of things for sustainable smart city brain and digital twin systems: Pioneering Environmental synergies between real-time management and predictive planning. *Environ Sci Ecotechnology*. 2025;26:100591. <https://doi.org/10.1016/J.ESE.2025.100591>.
16. Ottaviani FM, De Marco A, Narbaev T, Rebuglio M. Improving project estimates at completion through progress-based performance factors. *Buildings*. 2024;14:643. <https://doi.org/10.3390/BUILDINGS14030643>.
17. Rahimian F, Seyedzadeh S, Oliver S, et al. On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning. *Autom Constr*. 2020;110:103012. <https://doi.org/10.1016/j.autcon.2019.103012>.
18. Ekanayake B, Wong JKW, Fini AAF, et al. Deep learning-based computer vision in project management: automating indoor construction progress monitoring. *Project Leadersh Soc*. 2024;5:100149. <https://doi.org/10.1016/j.plas.2024.100149>.
19. Nishat MM, Ahsan A, Olsson NOE. Applying Machine Learning for Predictive Analysis in Project-Based Data: Insights into Variation Orders. *J Inform Technol Constr*. 2025;30:807–825. <https://doi.org/10.36680/jitcon.2025.033>.
20. Alshboul O, Shehadeh A, Mamlook RE, Al et al. Prediction liquidated damages via ensemble machine learning model: towards sustainable highway construction projects. *Sustainability*. 2022;14:9303. <https://doi.org/10.3390/su14159303>.
21. Park U, Kang Y, Lee H, Yun S. A Stacking Heterogeneous Ensemble Learning Method for the Prediction of Building Construction Project Costs. *Appl Sci*. 2022;12:12. <https://doi.org/10.3390/app12199729>.
22. Shehadeh A, Alshboul O, Tamimi MF. Hybrid machine learning solutions for mitigating climate-induced productivity losses in sustainable construction management. *J Clean Prod*. 2026;538:147285. <https://doi.org/10.1016/j.jclepro.2025.147285>.
23. Ding JY, Feng DC, Brunesi E, et al. Efficient seismic fragility analysis method utilizing ground motion clustering and probabilistic machine learning. *Eng Struct*. 2023;294:116739. <https://doi.org/10.1016/j.engstruct.2023.116739>.
24. Nguyen HD, Lee YJ, LaFave JM, Shin M. Seismic fragility analysis of steel moment frames using machine learning models. *Eng Appl Artif Intell*. 2023;126:106976. <https://doi.org/10.1016/j.engappai.2023.106976>.
25. Alshboul O, Shehadeh A, Almasabha G. Enhanced seismic resilience of cross-laminated timber buildings: A machine learning approach to developing robust fragility curves. *Eng Appl Artif Intell*. 2026;167:113848. <https://doi.org/10.1016/j.engappai.2026.113848>.
26. Alshboul O, Shehadeh A. Adaptive Integration of BIM and Navisworks for Real-Time Clash Detection Using the XGBoost Algorithm. *J Constr Eng Manag*. 2026;152:04025232. <https://doi.org/10.1061/jcemd4.coeng-16001>.
27. Karatas I, Budak A. Development and comparative of a new meta-ensemble machine learning model in predicting construction labor productivity. *Eng Constr Architectural Manage*. 2024;31:1123–44. <https://doi.org/10.1108/ECAM-08-2021-0692>.

28. Shehadeh A, Alshboul O. Enhancing occupational safety in Construction: Predictive analytics using advanced ensemble machine learning algorithms. *Eng Appl Artif Intell*. 2025;159:111761. <https://doi.org/10.1016/j.engappai.2025.111761>.
29. Abed YG, Hasan TM, Zehawi RN. Cost prediction for roads construction using machine learning models. *Int J Electr Comput Eng Syst*. 2022;13:927–936. <https://doi.org/10.32985/ijeces.13.10.8>
30. Tijanić K, Car-Pušić D, Šperac M. Cost estimation in road construction using artificial neural network. *Neural Comput Appl*. 2020;32:9343–9355. <https://doi.org/10.1007/s00521-019-04443-y>
31. Abd AM, Kareem YA, Zehawi RN. Prediction and Estimation of Highway Construction Cost using Machine Learning. *Eng Technol Appl Sci Res*. 2024;14:17222–31. <https://doi.org/10.48084/ETASR.8285>.
32. Cheng MY, Peng HS, Wu YW, Chen TL. Estimate at completion for construction projects using evolutionary support vector machine inference model. *Autom Constr*. 2010;19: 619–629. <https://doi.org/10.1016/j.autcon.2010.02.008>
33. PMBOK. A guide to the project management body of knowledge (PMBOK Guide). 2017.
34. Rolf E, Worledge T, Recht B, Jordan MI. Representation matters: assessing the importance of subgroup allocations in training data. In: *Proceedings of Machine Learning Research*. 2021; 139:9040–9051.
35. Lewis CD. *Industrial and Business Forecasting Methods: A Practical Guide to Exponential Smoothing and Curve Fitting*. London: Butterworth Scientific; 1982.
36. Kose T, Bakici T, Hazir O. Completing Projects on Time and Budget: A Study on the Analysis of Project Monitoring Practices Using Real Data. *IEEE Trans Eng Manag*. 2024;71:4051–62. <https://doi.org/10.1109/TEM.2022.3227428>.
37. Narbaev T, Hazir Ö, Khamitova B, Talgat S. A machine learning study to improve the reliability of project cost estimates. *Int J Prod Res*. 2024;62:4372–88. <https://doi.org/10.1080/00207543.2023.2262051>.
38. Ujong JA, Mbadike EM, Alaneme GU. Prediction of cost and duration of building construction using artificial neural network. *Asian J Civil Eng*. 2022;23:7. <https://doi.org/10.1007/s42107-022-00474-4>.
39. Shamim MMI, Hamid ABBA, Nyamasvisva TE, Rafi NSB. Advancement of artificial intelligence in cost estimation for project management success: a systematic review of machine learning. Deep learning, regression, and hybrid models. *Modelling*. 2025;6:35. <https://doi.org/10.3390/modelling6020035>
40. Fazil MW, Muhamad Tamyaz PF, Lee CK. Enhancing cost estimation performance through effective organizational control: the moderating role of project complexity in construction projects. *Int J Constr Manage*. 2024;24:1630–42. <https://doi.org/10.1080/15623599.2023.2286048>.
41. Babar S, Thaheem MJ, Ayub B. Estimated Cost at Completion: Integrating Risk into Earned Value Management. *J Constr Eng Manag*. 2016;143:04016104. [https://doi.org/10.1061/\(asce\)co.1943-7862.0001245](https://doi.org/10.1061/(asce)co.1943-7862.0001245).

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.