**INTERNATIONAL JOURNAL OF ADVANCED RESEARCH**
*International Peer-Reviewed Research Journal*
**ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972**
**Volume 3 • Issue 7 • November 2026**

# A HYBRID MACHINE LEARNING AND DEEP LEARNING APPROACH FOR PREDICTING COST OVERRUNS AND DELAYS IN IT PROJECTS

**Muhammad Sulaman[1], Usaid Iftikhar Ahmad[2], Muhammad Sajid Maqbool[*3], Muhammad Naman Chaudhary[4], Dr. Ghulam Irtaza[5], Abdul Basit[6], Muhammad Wajid Maqbool[7]**

[1] *Department of Information Technology, University of Southern Punjab, Multan, Pakistan*
[2,*3,4] *Department of Computer Science, NFC Institute of Engineering and Technology, Multan, Pakistan*
[5,7] *Department of Information Sciences, University of Education Multan Campus, Pakistan*
[6] *Department of Computer Science, MNS University of Engineering and Technology, Multan, Pakistan*

[1]sk7389884@gmail.com, [2]usaidch989@gmail.com, [*3]sajid.maqbool@nfciet.edu.pk, [4]naman@nfciet.edu.pk, [5]ghulam.irtaza@ue.edu.pk, [6]abdul.basit@mnsuet.edu.pk, [7]mwajidmaqboolb@gmail.com

ORCID: [*3]0000-0001-5583-3550

**DOI:** https://doi.org/10.5281/zenodo.24156314

**Keywords**
*IT project management, cost overrun prediction, schedule delay prediction, machine learning, deep learning, hybrid ensemble, Random Forest, gradient boosting, SVR, MLP, explainable AI.*

**Article History**
*Received: 05 August 2026*
*Accepted: 19 October 2026*
*Published: 12 November 2026*

**Copyright** @Author
**Corresponding Author: ***
**Muhammad Sajid Maqbool**

**Abstract**

*Cost overruns and schedule delays remain persistent challenges in information technology (IT) project management, often resulting in increased expenses, missed deadlines, and reduced project performance. These issues commonly stem from inaccurate cost and time estimates, changing project requirements, inadequate resource allocation, technical complexity, and ineffective stakeholder communication. Despite advances in project management methodologies and tools, these challenges are still difficult to manage because of the dynamic and unpredictable nature of IT projects. This paper implements and empirically evaluates a hybrid machine-learning and deep-learning framework that jointly predicts percentage cost overrun and percentage schedule delay from information available at project initiation. Using a dataset of 5,000 software projects, we train a Random Forest (RF), a gradient-boosted tree model (a scikit-learn implementation of the XGBoost approach), Support Vector Regression (SVR) with an RBF kernel, and a three-hidden-layer Multilayer Perceptron (MLP), and we combine them through a validation-weighted hybrid ensemble. The models are compared against a median baseline on a held-out test set (n = 750) using the mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination ($R^2$), with project-level bootstrap confidence intervals. The hybrid ensemble reduces the test MAE by roughly 35% (cost) and 36% (schedule) relative to the baseline, reaching $R^2$ values of 0.655 and 0.718, respectively. The hybrid ensemble is statistically indistinguishable from the best single model (Random Forest) for both targets. Permutation importance shows that the predictive signal comes almost entirely from the planned budget and planned duration, whereas project size, team size, requirement changes, developer experience,*

**INTERNATIONAL JOURNAL OF ADVANCED RESEARCH**
*International Peer-Reviewed Research Journal*
**ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972**
**Volume 3 • Issue 7 • November 2026**

*and complexity contribute essentially nothing. A data audit further shows that the planned and actual values in this dataset are uncorrelated ($|r| < 0.02$) and that the binary Delay and Budget Overrun labels are exact functions of the actual values. These findings indicate that the high $R^2$ is largely a mechanical consequence of the ratio-based target definition and that the dataset is likely synthetic.*

## 1. INTRODUCTION

Information technology projects support digital transformation across education, finance, healthcare, government, and commerce. Their success depends on delivering the required functionality within an acceptable budget and the planned schedule. Despite advances in software engineering practices and project management methodologies, forecasting the final cost and duration remains difficult. Initial estimates may become unreliable when the project scope changes, requirements are misunderstood, technical dependencies increase, or experienced staff become unavailable. Traditional techniques, such as expert judgment, analogy-based estimation, parametric models (e.g., COCOMO), and Earned Value Management (EVM), are useful for project planning and monitoring but may struggle to capture complex nonlinear relationships among project characteristics. Machine learning (ML) can learn such relationships from historical project records, while deep learning (DL) can learn complex representations of input features. However, deep learning does not automatically outperform tree-based methods on moderately sized tabular datasets [1], [2]. Hybrid models that combine multiple learning algorithms have also been investigated for cost and schedule prediction [3].

This paper implements and empirically evaluates a hybrid machine-learning and deep-learning framework for jointly predicting percentage cost overrun and percentage schedule delay using information available at project initiation. The study aims to: (i) identify project characteristics associated with cost and schedule deviations; (ii) develop machine-learning models for predicting cost overrun and schedule delay; (iii) implement a deep neural network for the same prediction tasks; (iv) integrate the predictions through a validation-weighted ensemble; and (v) compare the performance of the individual and hybrid models under a consistent evaluation protocol that includes uncertainty estimates. The evaluated models include Random Forest (RF), a gradient-boosted tree model based on the XGBoost approach, Support Vector Regression (SVR) with a radial basis function (RBF) kernel, and a three-hidden-layer Multilayer Perceptron (MLP). Their performance is compared with a median baseline on a held-out test set using the mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination ($R^2$).

The experimental results show that the hybrid ensemble reduces the test MAE by approximately 35% for cost overrun prediction and 36% for schedule delay prediction relative to the median baseline, achieving $R^2$ values of 0.655 and 0.718, respectively. These results indicate that the ensemble captures substantial variation in both targets compared with a simple baseline. However, the ensemble is statistically indistinguishable from the best-performing individual model, Random Forest, for both prediction tasks. This finding suggests that combining multiple algorithms does not necessarily provide a statistically significant improvement over a strong tree-based model on the dataset under investigation. Furthermore, permutation feature importance reveals that the planned budget and planned duration account for almost all of the predictive signal, whereas project size, team size, requirement changes, developer experience, and project complexity contribute negligibly.

The dataset audit also identifies important limitations: the planned and actual project values are nearly uncorrelated ($|r| < 0.02$), and the binary Delay and Budget Overrun labels are exact functions of the corresponding actual values. These findings raise concerns about the dataset's

realism and indicate that the observed predictive performance may largely reflect the mathematical construction of the ratio-based targets rather than meaningful relationships between project characteristics and project outcomes. Consequently, although the experimental results demonstrate the framework's predictive performance on the available data, they do not establish its generalizability to real-world IT projects. This study therefore emphasizes the importance of dataset validation, target-definition analysis, and independent real-world evaluation when developing machine-learning systems for project risk prediction.

The remainder of this paper is organized as follows. Section 2 reviews related studies on IT project cost and schedule prediction. Section 3 describes the dataset, preprocessing procedures, target definitions, and the proposed methodology. Section 4 presents the experimental results and comparative evaluation. Section 5 discusses the findings, limitations, and implications for practical project management. Finally, Section 6 concludes the study and outlines directions for future research.

## 2. LITERATURE REVIEW

### 2.1 Traditional Estimation and Monitoring Techniques

Expert judgment, analogy-based estimation, and parametric models such as COCOMO provide useful baselines when reliable historical data exist, but they are less adaptable when historical patterns change. EVM indicators such as the Cost Performance Index (CPI) and Schedule Performance Index (SPI) support baseline monitoring but do not by themselves guarantee accurate forecasts of the final cost or completion time.

### 2.2 Machine Learning and Deep Learning for Project Prediction

ML has been widely investigated for software effort estimation and project performance prediction [1], [2]. Random Forest handles nonlinear relationships and interactions among mixed predictors; gradient boosting (e.g., XGBoost) builds additive tree ensembles that iteratively correct residual error; and Support Vector Regression learns nonlinear mappings through kernel functions. A Multilayer Perceptron (MLP) can approximate complex nonlinear functions, but deep models usually require larger datasets to generalize and should not be assumed to outperform tree ensembles on tabular data.

A recent systematic review of 39 studies published between 2016 and 2024 examined ML, DL, regression, and hybrid models for cost estimation across several sectors [2]. It reports average accuracies of roughly 75–80% for ML models, 70–80% for regression models, and 85–90% for deep learning, and it concludes that deep-learning and hybrid models are the most accurate. These figures are aggregated over heterogeneous datasets, targets, and accuracy definitions, so they indicate a general trend rather than a controlled comparison of algorithms, and they do not settle whether deep models keep their advantage on moderately sized tabular datasets such as the one studied here.

Hybrid prediction combines complementary learners to reduce dependence on the weaknesses of any single model; related work has applied hybrid and optimized neural approaches to the joint prediction of project cost and time in civil-engineering projects [8].

### 2.3 Empirical Studies in Related Project Domains

Most empirical evidence comes from construction and infrastructure projects rather than software projects. Park et al. [15] analyzed 83 sewage pipeline projects in South Korea and compared statistical, machine-learning, and deep-learning regression models for duration prediction, using hyperparameter optimization, k-fold cross-validation, and $R^2$, RMSE, MAE, and MSE as metrics. Debero and Sinesilassie [18] used questionnaire data from public building projects in Hosanna, Ethiopia, to evaluate classical time–cost and time–floor-area models built with linear regression and proposed an alternative parametric model based on Random Forest, XGBoost,

decision tree, K-nearest neighbor, and polynomial regression.

Sohrabi and Noorzai [17] estimated cost overrun for 68 water-related projects using EVM-based time and cost factors together with risk and uncertainty variables. They found that accounting for uncertainty improved accuracy, that ensemble algorithms such as Random Forest and AdaBoost were the most accurate, and that SVR with a sigmoid kernel was the best single algorithm. Taha et al. [18] trained an MLP with 128, 64, and 32 ReLU units (the same layer sizes as the MLP used in this study) on 5,000 simulated scenarios built from 20 survey-derived risk factors and validated it on eight real projects, obtaining a test $R^2$ of 0.82 and average errors of 0.87 months and EGP 102,500. Table 1 summarizes these studies and contrasts them with the present work [19].

**Table 1. Comparison of selected related studies and the present work**

| Ref. | Domain and data | Methods | Main findings | Relevance and limitation |
|---|---|---|---|---|
| [2] | Systematic review of 39 studies (2016–2024) across several sectors; cost estimation | ML, DL, regression, and hybrid models | Average accuracy of about 75–80% (ML), 70–80% (regression), and 85–90% (DL); DL and hybrid models most accurate | Supports ML/DL/hybrid use; accuracies come from heterogeneous studies and are not directly comparable |
| [15] | 83 sewage pipeline projects, South Korea; duration | Statistical, ML, and DL regression; hyperparameter optimization; k-fold cross-validation | Compares model families using $R^2$, RMSE, MAE, and MSE | Duration only; small sample; evaluation protocol similar to this study |
| [18] | Public building projects, Hosanna, Ethiopia; questionnaire data; duration | Bromilow and Love et al. models (linear regression); RF, XGBoost, decision tree, KNN, polynomial regression | Proposes an ML-based alternative to classical time models for early duration estimation | Survey-based data; duration only; no cost target |
| [17] | 68 water-related projects; cost overrun | EVM factors plus risk variables; RF, AdaBoost, SVR; case-based reasoning; MAPE and NRMSE | Uncertainty variables improved accuracy; ensembles most accurate; SVR (sigmoid) best single model | Cost only; competitive ensembles are consistent with the RF and boosting results here |
| [22] | Irrigation canal-lining projects, Egypt; duration and cost | MLP (128-64-32); 20 risk factors from 93 via AHP–RII; leave-one-project-out validation | Test $R^2$ of 0.82; mean errors of 0.87 months and EGP 102,500 | Joint time and cost, but trained on simulated scenarios and validated on only eight real projects |
| [28] | Civil-engineering projects; cost and time | Hybrid ML with optimized neural networks | Joint prediction of project cost and time | Closest antecedent for joint hybrid prediction; not software projects |

| | | | | |
|---|---|---|---|---|
| This study | 5,000 software projects (origin undocumented); cost overrun and schedule delay | RF, gradient boosting, SVR, MLP, and validation-weighted hybrid ensemble | Hybrid cuts MAE by about 35–36% vs. baseline ($R^2$ = 0.655 and 0.718) but does not significantly beat RF; signal comes from planned values only | IT/software domain with joint targets, leakage control, and a dataset audit; dataset likely synthetic |

**2.4 Research Gap and Motivation**

Taken together, these studies show that tree ensembles, kernel methods, and neural networks can all be competitive, that no single model family is reported as best across all of them, and that results depend strongly on dataset size and feature quality. Three gaps remain. First, most evidence comes from construction and infrastructure projects with small samples (68 and 83 projects in [17] and [15], respectively), and comparable evidence for IT and software projects is scarce. Second, many studies model cost or time separately; Taha et al. note that most earlier construction research treated them in isolation [10]. Third, the studies in Table 1 emphasize predictive accuracy, whereas a systematic audit of target construction, label leakage, and dataset realism is not their focus, and the use of simulated training scenarios in [10], validated on only eight real projects, illustrates why such checks matter.

The literature therefore motivates (a) joint modeling of cost and schedule outcomes, (b) controlled comparison of ML and DL models using identical data partitions and metrics, (c) explicit leakage prevention, and (d) interpretable outputs that managers can act on. These four concerns shape the design described in Section 3.

## 3. PROPOSED METHODOLOGY

The proposed study adopts a quantitative, experimental research methodology to develop and evaluate a hybrid machine-learning and deep-learning framework for predicting software project cost overruns and schedule delays at the project initiation stage. A dataset of 5,000 completed software projects is used, containing 11 original variables: project size, team size, estimated and actual budget, estimated and actual duration, requirement changes, developer experience, project complexity, delay, and budget overrun. The dataset is first examined for missing values and duplicate records, and the targets are then constructed using percentage-based formulas for cost overrun and schedule delay. To prevent data leakage, the actual budget, actual duration, and existing binary outcome labels are excluded from the input features, ensuring that predictions rely only on information available at project initiation. Four additional features (Size_per_Member, Budget_per_Month, Req_Volatility, and Budget_per_Size) are engineered, and project complexity is one-hot encoded, resulting in 13 input features.

The dataset is randomly divided into training (70%), validation (15%), and test (15%) subsets. Four predictive models—Random Forest (RF), Gradient Boosting, Support Vector Regression (SVR), and a Multilayer Perceptron (MLP)—are developed and tuned using validation-based hyperparameter search. Their predictions are subsequently combined by a hybrid ensemble whose optimal weights are determined on the validation data. The overall workflow is shown in Fig. 1.

**INTERNATIONAL JOURNAL OF ADVANCED RESEARCH**
*International Peer-Reviewed Research Journal*
**ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972**
**Volume 3 • Issue 7 • November 2026**

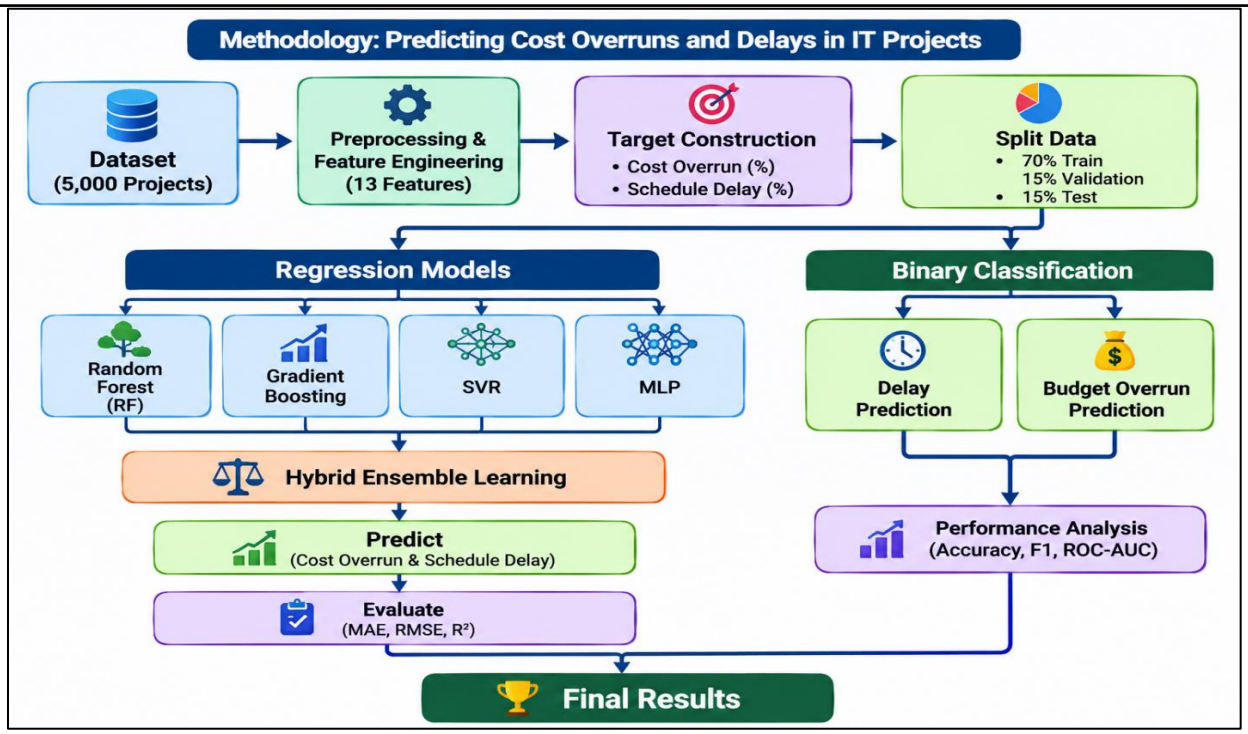



**Fig. 1. Proposed methodology flow.**

The framework predicts two continuous outcomes, cost overrun (%) and schedule delay (%), which are evaluated using MAE, RMSE, and $R^2$. In addition, bootstrap resampling and five-fold cross-validation are employed to assess model reliability, and the predicted outcomes are categorized into low, moderate, and high risk levels. A separate binary classification experiment is also conducted using Random Forest, Gradient Boosting, RBF-SVC, MLP, and a mean-probability hybrid model to predict project delays and budget overruns; its performance is assessed using accuracy, F1-score, and ROC-AUC. Finally, permutation feature importance is applied to examine the contribution of each input variable.

### 3.1 Research Data

The dataset (software_project_dataset_5000.xlsx) contains 5,000 projects and 11 variables: Project_Size, Team_Size, Estimated_Duration_Months, Actual_Duration_Months, Estimated_Budget, Actual_Budget, Requirement_Changes, Developer_Experience_Years, Project_Complexity (Low/Medium/High), and two binary labels, Delay and Budget_Overrun. There are no missing values and no duplicate rows. No project identifier, project type, or date is provided; therefore, chronological validation was not possible.

### 3.2 Target Construction

Following the research design, the two continuous targets are defined as:

$$\textit{Cost overrun}\ (\%) = (\textit{Actual_Budget}\ \textit{Estimated_Budget})\ \textit{Estimated_Budget}\ 100 \quad (1)$$

$$\textit{Schedule delay}\ (\%) = (\textit{Actual_Duration}\ \textit{Estimated_Duration})\ \textit{Estimated_Duration}\ 100 \quad (2)$$

All planned (estimated) budgets and durations are strictly positive, so the formulas are valid for every record. Positive values indicate overruns or delays, whereas negative values indicate completion under budget or ahead of schedule.

### 3.3 Leakage Control and Feature Engineering

Because the framework is intended as an early-warning tool, only information available at project initiation is used as input. Actual_Budget, Actual_Duration_Months, Delay, and Budget_Overrun are excluded from the predictors. A data audit confirmed why this matters: in 100% of records, Delay equals 1 exactly when the actual duration exceeds the estimated duration, and Budget_Overrun equals 1 exactly when the actual budget exceeds the estimated budget. A leakage test in which the actual values were added as inputs enabled a Random Forest to classify these labels with 98.8% (Delay) and 98.1% (Budget_Overrun) test accuracy, which does not represent a legitimate forecast.

Four features were engineered from initiation-time variables: Size_per_Member (project size ÷ team size), Budget_per_Month (estimated budget ÷ estimated duration), Req_Volatility (requirement changes ÷ estimated duration), and Budget_per_Size (estimated budget ÷ project size). Project_Complexity was one-hot encoded. The final design matrix has 13 features.

### 3.4 Data Partitioning and Preprocessing

Records were split at random (seed 42) into training (n = 3,500; 70%), validation (n = 750; 15%), and test (n = 750; 15%) sets. Each record corresponds to one project, so no project spans two partitions. Standardization (fitted on the training data only) was applied for the SVR and the MLP, whereas the tree models used raw features. Targets were also standardized for the SVR and MLP, and the predictions were transformed back to the percentage scale before evaluation. No outliers were removed: large overruns (maximum 5,217%) are genuine consequences of the target definition rather than data errors.

### 3.5 Models Used in Experiments

- **Median baseline.** Predicts the training-set median of each target.
- **Random Forest (RF).** 300 trees per target; the maximum depth, minimum leaf size, and feature fraction were selected by random search on the validation set (12 draws).
- **Gradient boosting (XGBoost-type).** The XGBoost library was not available in the execution environment; therefore, scikit-learn's GradientBoostingRegressor was used, which implements the same additive-tree boosting idea with shrinkage, subsampling, and column sampling. The learning rate, number of trees, depth, subsample ratio, and feature fraction were tuned by validation random search (12 draws). The results should therefore be read as representative of gradient boosting in general, not of the XGBoost library itself.
- **Support Vector Regression (SVR).** RBF kernel on standardized inputs and targets; C, γ, and ε were tuned by validation random search (10 draws).
- **Multilayer Perceptron (MLP).** A single multi-output network with hidden layers of 128, 64, and 32 ReLU units, the Adam optimizer (learning rate 0.001), a batch size of 32, and up to 200 epochs, with early stopping on a held-out 15% of the training data (patience 15). Training stopped after 43 epochs. Because scikit-learn's MLP does not support dropout, L2 regularization (α = 0.01) replaced the 0.20 dropout proposed in the design.

**Hybrid ensemble.** The hybrid ensemble combines the four component models. For each target k, the ensemble prediction is:

$$\hat{y}(H)k = \Sigma m = 14\, wm,k \cdot \hat{y}m,k \quad (3)$$

where $\hat{y}_{m,k}$ is the prediction of model m for target k, $w_{m,k} \geq 0$, and $\Sigma_m w_{m,k} = 1$. The weights were obtained by minimizing the mean squared error on the validation predictions using a constrained optimizer (SLSQP); the test set was never used for weighting.

### 3.6 Evaluation Protocol

All models share identical data partitions. Test-set MAE, RMSE, and $R^2$ are reported. Uncertainty is quantified with 1,000 bootstrap resamples of the test set, and the hybrid model is compared with the baseline and with the best single model using

the paired bootstrap difference in MAE. Five-fold cross-validation on the development data (training plus validation) is reported for the two tree models as a robustness check.

Risk categories follow the proposed thresholds (Low: <5%; Moderate: 5% to <10%; High: ≥10%) and are assigned separately for cost and schedule from the continuous predictions; they are therefore threshold-based categorizations rather than independently trained classifiers. Finally, because the dataset provides binary labels, classifiers (RF, gradient boosting, RBF-SVC, MLP, and a mean-probability hybrid) were also trained for Delay and Budget_Overrun using the same initiation-time features. Permutation importance (15 repeats, test set, $R^2$ scoring) was used in place of SHAP, which was unavailable.

## 4. RESULTS AND ANALYSIS

### 4.1 Dataset Description

Table 2 summarizes the dataset characteristics, and Table 3 reports the descriptive statistics of the raw variables.

**Table 2. Dataset characteristics**

| Characteristic | Value |
|---|---|
| Total number of projects | 5,000 |
| Data source | software_project_dataset_5000.xlsx (single sheet; origin not documented) |
| Collection period | Not provided |
| Input features (after engineering) | 13 |
| Cost overrun, median / mean | 17.7% / 145.5% |
| Schedule delay, median / mean | 23.1% / 141.2% |
| Projects with Delay = 1 / Budget_Overrun = 1 | 58.2% / 56.9% |
| Missing-value rate / duplicate rows | 0% / 0 |
| Complexity (Low / Medium / High) | 1,656 / 1,683 / 1,661 |
| Train / validation / test sizes | 3,500 / 750 / 750 |

International Journal of Advanced Research

**INTERNATIONAL JOURNAL OF ADVANCED RESEARCH**
*International Peer-Reviewed Research Journal*
**ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972**
**Volume 3 • Issue 7 • November 2026**

**Table 3. Descriptive statistics of the raw variables**

| Variable | Mean | Std | Min | Median | Max |
|---|---|---|---|---|---|
| Project size | 529.57 | 276.41 | 50.00 | 534.00 | 999.00 |
| Team size | 11.19 | 4.93 | 3.00 | 11.00 | 19.00 |
| Estimated duration (months) | 12.17 | 6.59 | 1.00 | 12.00 | 23.00 |
| Actual duration (months) | 15.09 | 8.32 | 1.00 | 15.00 | 29.00 |
| Estimated budget | 256,350 | 142,726 | 10,018 | 255,458 | 499,998 |
| Actual budget | 301,435 | 172,221 | 12,056 | 298,810 | 599,848 |
| Requirement changes | 24.65 | 14.34 | 0.00 | 25.00 | 49.00 |
| Developer experience (years) | 7.62 | 3.97 | 1.00 | 8.00 | 14.00 |
| Delay (0/1) | 0.58 | 0.49 | 0.00 | 1.00 | 1.00 |
| Budget overrun (0/1) | 0.57 | 0.50 | 0.00 | 1.00 | 1.00 |
| Cost overrun (%) | 145.49 | 447.37 | −97.42 | 17.74 | 5,217 |
| Schedule delay (%) | 141.24 | 383.39 | −95.65 | 23.08 | 2,800 |

Both targets are strongly right-skewed (skewness of 5.30 for cost and 4.03 for schedule; Fig. 2), with medians of 17.7% and 23.1% but means above 140%. The two targets are essentially uncorrelated with each other ($r = 0.009$), so cost and schedule deviations behave as independent outcomes in this dataset, which weakens the rationale for joint modeling on these data. Median outcomes differ little across complexity levels (cost: 17.6%, 19.4%, and 16.3% for Low, Medium, and High, respectively; Fig. 4).

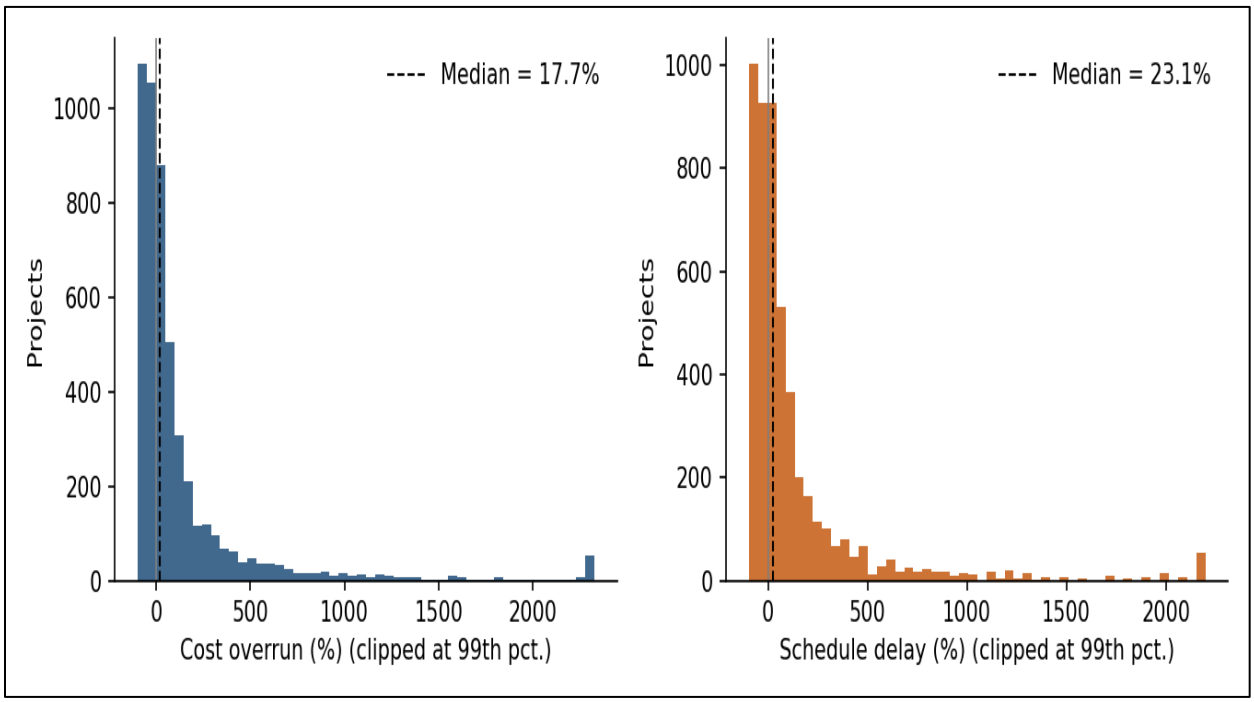



**Fig. 2. Distributions of cost overrun (%) and schedule delay (%).**

The correlation matrix (Fig. 3) is the most revealing exploratory result. Cost overrun correlates with the estimated budget ($r = -0.51$) and schedule delay with the estimated duration ($r = -0.56$), but all other predictors have $|r| < 0.03$ with either target. Moreover, the estimated and actual budgets are uncorrelated ($r = -0.009$), as are the estimated and actual durations ($r = -0.015$). Hence, actual outcomes behave as random draws independent of the plan, and a percentage target divided by the plan will automatically be lower for larger plans. This is a statistical artifact of the ratio, not evidence that large budgets "cause" smaller overruns.

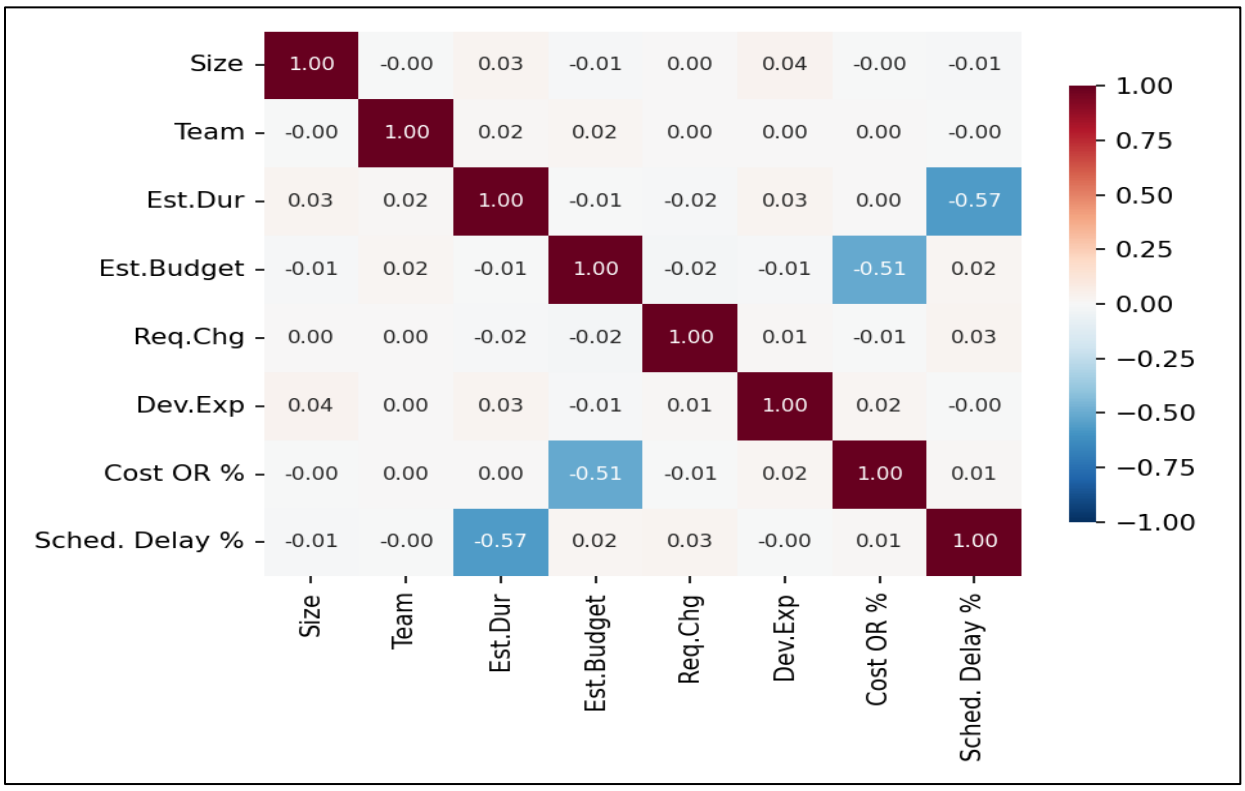



**Fig. 3. Pearson correlation matrix of predictors and targets.**

**INTERNATIONAL JOURNAL OF ADVANCED RESEARCH**
*International Peer-Reviewed Research Journal*
**ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972**
**Volume 3 • Issue 7 • November 2026**

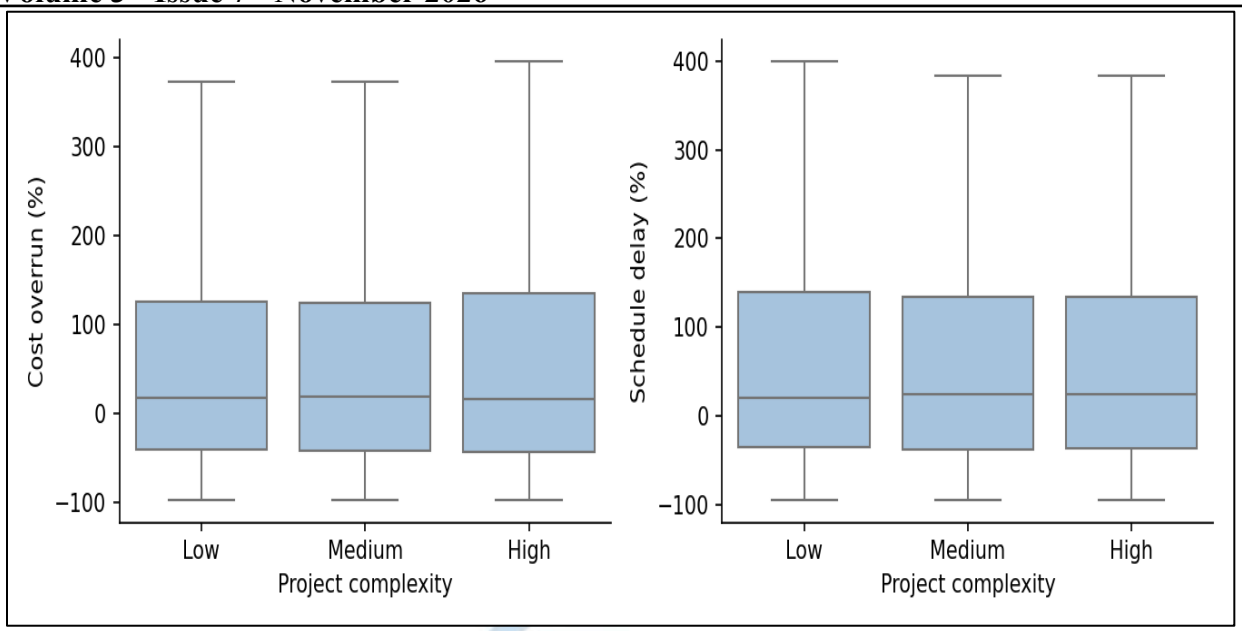



**Fig. 4. Cost overrun and schedule delay by project complexity.**

### 4.2 Cost Overrun Prediction

**Table 4. Test-set performance for cost overrun (percentage points)**

| Model | MAE | RMSE | $R^2$ |
|---|---|---|---|
| Median baseline | 178.3 | 405.2 | −0.090 |
| Random Forest | 115.1 | 232.1 | 0.642 |
| Gradient Boosting (XGBoost-type) | 115.5 | 228.9 | 0.652 |
| SVR (RBF) | 154.2 | 282.0 | 0.472 |
| MLP (128-64-32) | 130.6 | 250.4 | 0.584 |
| **Hybrid ensemble** | **115.2** | **228.1** | **0.655** |

All learned models clearly outperform the median baseline ($R^2$ = −0.090). Random Forest and gradient boosting are the strongest single models (MAE of 115.1 and 115.5, respectively); the MLP is somewhat weaker ($R^2$ = 0.584), and SVR is the weakest ($R^2$ = 0.472). The hybrid ensemble obtains the lowest RMSE (228.1) and the highest $R^2$ (0.655) but not the lowest MAE, where Random Forest is marginally ahead (115.1 vs. 115.2). The MAE of roughly 115 percentage points is large in absolute terms because the target ranges from −97% to over 5,000%.

**INTERNATIONAL JOURNAL OF ADVANCED RESEARCH**
*International Peer-Reviewed Research Journal*
**ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972**
**Volume 3 • Issue 7 • November 2026**

### 4.3 Schedule Delay Prediction

**Table 5. Test-set performance for schedule delay (percentage points)**

| Model | MAE | RMSE | $R^2$ |
|---|---|---|---|
| Median baseline | 171.0 | 392.1 | −0.094 |
| Random Forest | 109.7 | 198.8 | 0.719 |
| Gradient Boosting (XGBoost-type) | 110.2 | 197.6 | 0.722 |
| SVR (RBF) | 111.0 | 209.1 | 0.689 |
| MLP (128-64-32) | 112.5 | 200.1 | 0.715 |
| **Hybrid ensemble** | **109.1** | **199.1** | **0.718** |

The pattern is similar. The hybrid ensemble has the lowest MAE (109.1), while gradient boosting has the best RMSE (197.6) and $R^2$ (0.722). The four learners are much closer to each other than for cost: the MLP ($R^2$ = 0.715) is competitive with the tree models, and SVR ($R^2$ = 0.689) is only slightly behind. The ensemble does not improve materially on the best single model. Figs. 5 and 6 visualize the model comparison and the hybrid predictions, respectively.

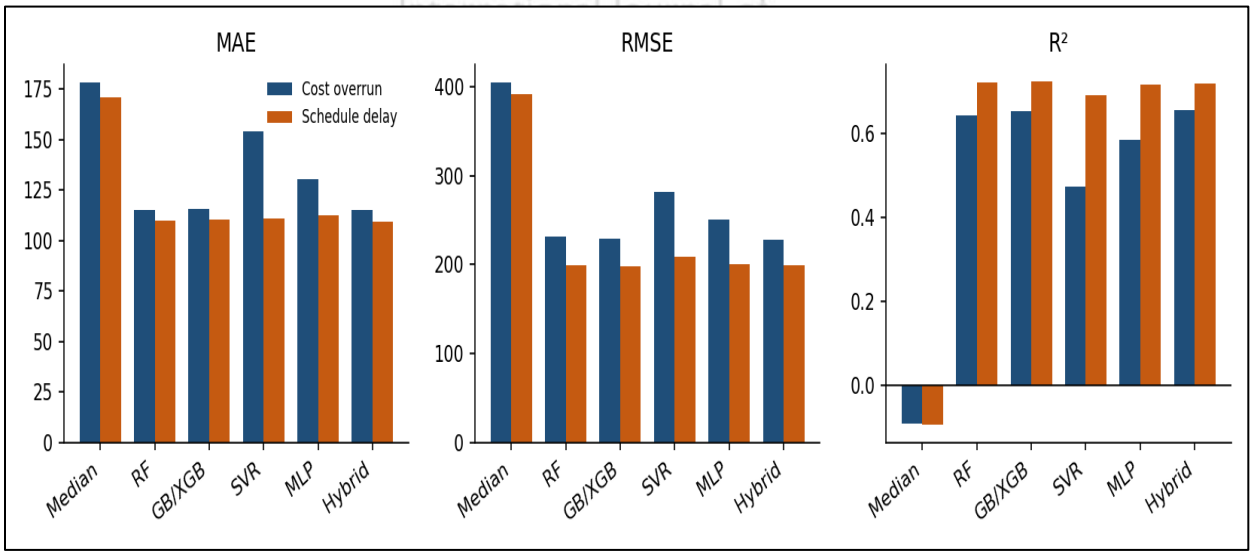



**Fig. 5. Test MAE, RMSE, and $R^2$ for all models and both targets.**

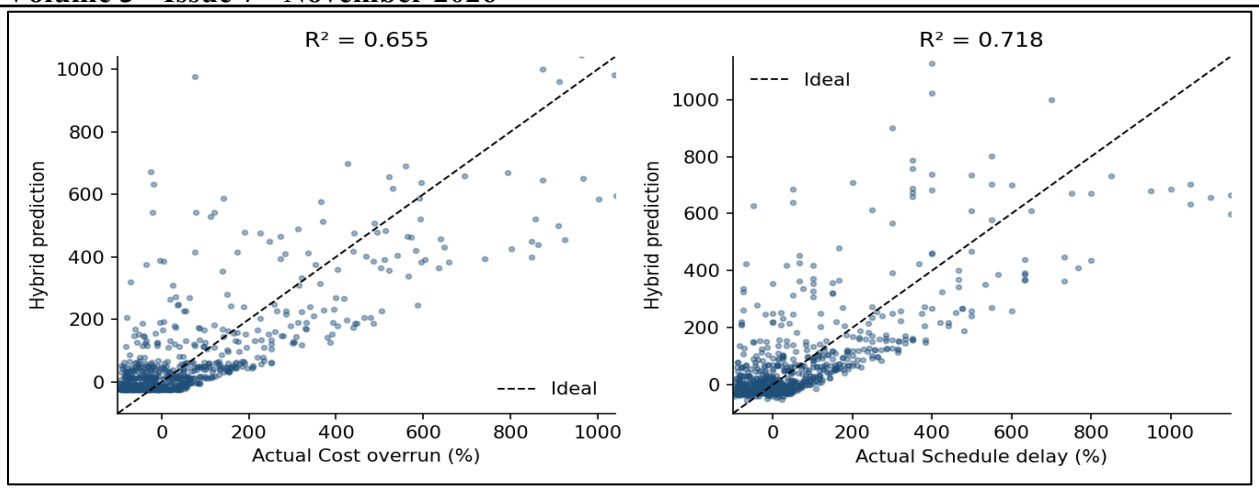



**Fig. 6. Hybrid-ensemble predictions versus actual values on the test set.**

### 4.4 Hybrid Ensemble Weights and Statistical Comparison

**Table 6. Validation-optimized ensemble weights**

| Target | RF | Gradient boosting | SVR | MLP |
|---|---|---|---|---|
| Cost overrun | 0.188 | 0.812 | 0.000 | 0.000 |
| Schedule delay | 0.548 | 0.000 | 0.452 | 0.000 |

The optimizer concentrates the weight on one or two models (RF and gradient boosting for cost; RF and SVR for schedule) and assigns zero weight to the MLP for both targets, which suggests that the neural network adds no complementary information beyond the tree and kernel models on these features.

**Table 7. Bootstrap uncertainty (1,000 resamples, test set)**

| Target | Comparison | Mean MAE gain (pp) | 95% CI |
|---|---|---|---|
| Cost | Hybrid vs. median baseline | 63.6 | [42.9, 85.5] |
| | Hybrid vs. best single (Random Forest) | −0.00 | [−3.55, 3.83] |
| Schedule | Hybrid vs. median baseline | 62.0 | [43.5, 81.6] |
| | Hybrid vs. best single (Random Forest) | 0.54 | [−2.04, 3.36] |

The hybrid's improvement over the baseline is large, and its confidence interval excludes zero for both targets. Its difference from Random Forest, the best single model, is tiny, and the interval spans zero (cost: [−3.55, 3.83]; schedule: [−2.04, 3.36]). We therefore cannot claim that the hybrid is better than the best individual model.

Five-fold cross-validation on the development data gave a mean $R^2$ (± SD) of 0.655 ± 0.082 (RF) and 0.633 ± 0.083 (gradient boosting) for cost, and 0.692 ± 0.017 and 0.651 ± 0.041 for schedule, which is consistent with the test results. The fold-to-fold spread for cost (SD ≈ 0.08) shows that $R^2$ for the heavy-tailed cost target is sensitive to a few extreme projects, which is also visible in the wide bootstrap interval for the hybrid's $R^2$ ([0.46, 0.77]).

**INTERNATIONAL JOURNAL OF ADVANCED RESEARCH**
*International Peer-Reviewed Research Journal*
**ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972**
**Volume 3 • Issue 7 • November 2026**

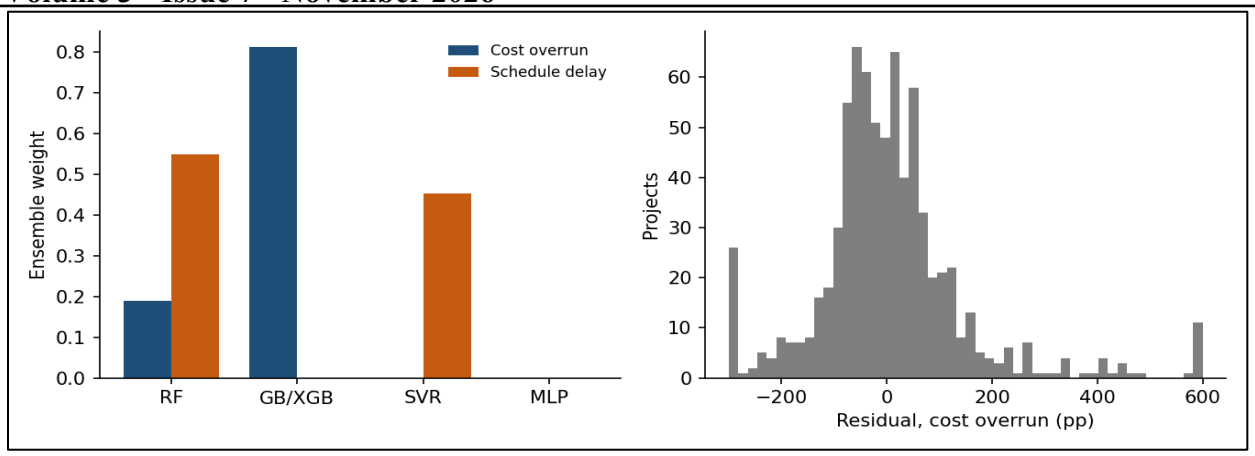



**Fig. 7. Ensemble weights (left) and residual distribution for cost overrun.**

### 4.5 Explainability: What Drives the Predictions

**Table 8. Top five features by mean permutation importance ($\Delta R^2$, averaged over RF and gradient boosting)**

| Rank | Cost overrun | $\Delta R^2$ | Schedule delay | $\Delta R^2$ |
|---|---|---|---|---|
| 1 | Estimated Budget | 1.439 | Estimated Duration Months | 1.062 |
| 2 | Budget per Month | 0.014 | Budget per Month | 0.040 |
| 3 | Budget per Size | 0.011 | Req Volatility | 0.009 |
| 4 | Estimated Duration Months | 0.002 | Team Size | 0.003 |
| 5 | Team Size | 0.001 | Estimated Budget | 0.001 |

The estimated budget dominates cost-overrun predictions ($\Delta R^2 \approx 1.44$), and the estimated duration dominates schedule-delay predictions ($\Delta R^2 \approx 1.06$); every other feature, including requirement changes, team size, developer experience, project complexity, and the engineered ratios, has an importance close to zero (Fig. 8). Because importance does not imply causation, and because the dominant effect is a ratio artifact (Section 4.1), this result should not be read as managerial guidance that "bigger budgets prevent overruns." Instead, it indicates that, in these data, none of the project-characteristic variables that the framework was designed to exploit carry predictive information.

**INTERNATIONAL JOURNAL OF ADVANCED RESEARCH**
*International Peer-Reviewed Research Journal*
**ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972**
**Volume 3 • Issue 7 • November 2026**

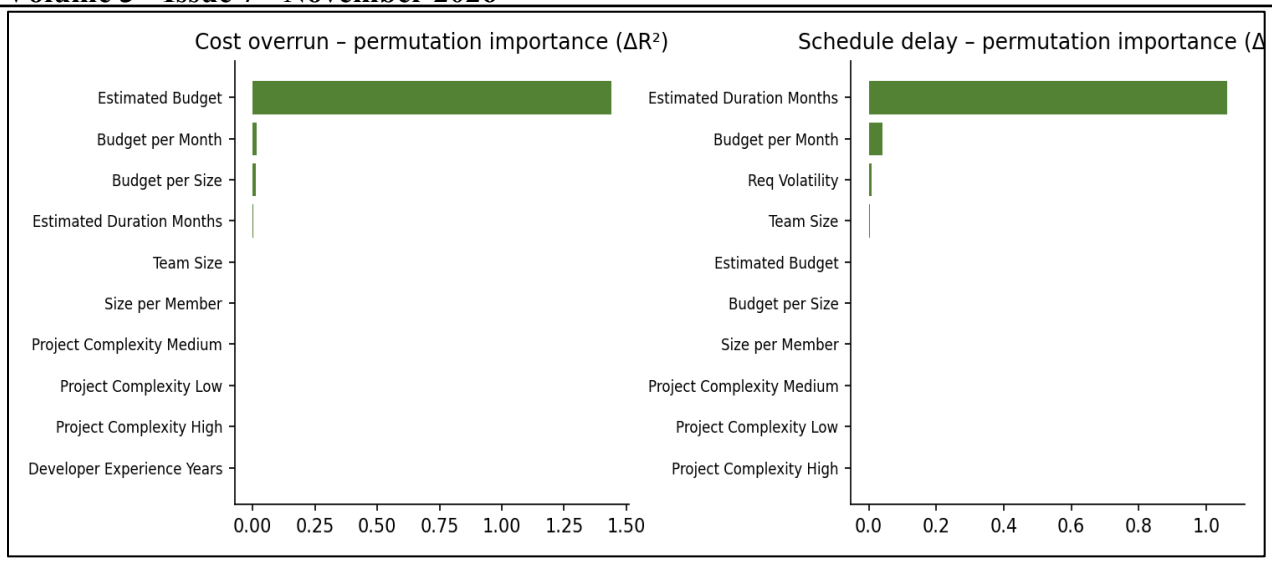



**Fig. 8. Top-10 permutation importances for cost overrun and schedule delay.**

### 4.6 Deep-Learning Training Behavior

The MLP trained for 43 epochs before early stopping, with a best internal validation $R^2$ of 0.668 on the standardized targets (Fig. 9). Early stopping on a validation split limited overfitting. As noted above, the network performed below the tree models for cost and on a par with them for schedule, which is consistent with the cautions in the literature that DL need not outperform tree ensembles on moderately sized tabular data.

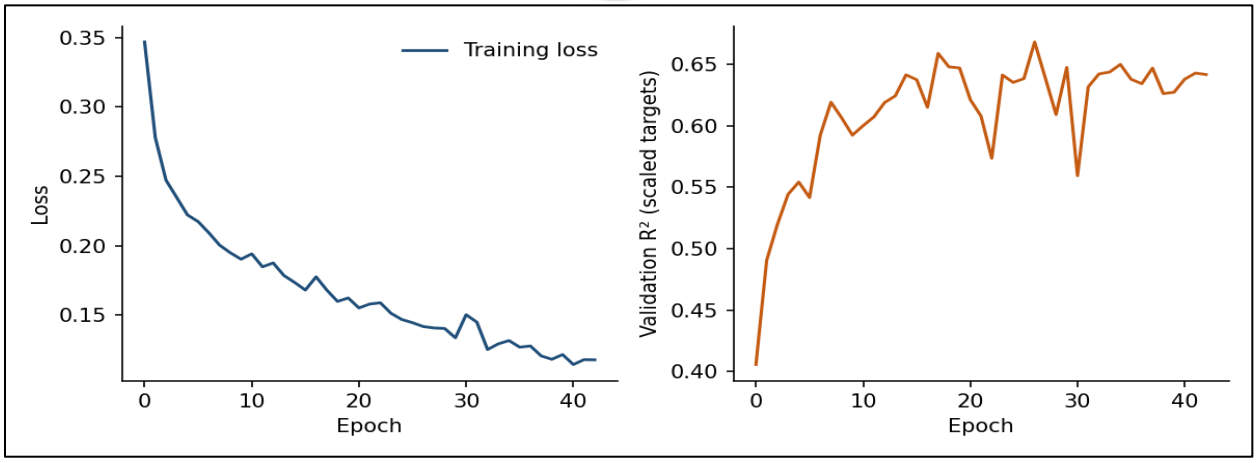



**Fig. 9. MLP training loss (left) and validation score (right).**

### 4.7 Threshold-Based Risk Classification

On the test set, 388 of 750 projects (51.7%) are High cost-risk, 347 are Low, and only 15 are Moderate; for schedule, 429 are High, 300 are Low, and 21 are Moderate. The thresholds of 5% and 10% are very tight relative to the spread of the outcomes, so the Moderate band is almost empty, and the macro metrics are therefore pulled down by a class that the models cannot learn.

**INTERNATIONAL JOURNAL OF ADVANCED RESEARCH**
*International Peer-Reviewed Research Journal*
**ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972**
**Volume 3 • Issue 7 • November 2026**

Table 9. Risk-category performance derived from continuous predictions

| Target | Model | Accuracy | Balanced accuracy | Macro-F1 |
|---|---|---|---|---|
| Cost | Median baseline | 51.7% | 33.3% | 0.227 |
| | Random Forest | 69.5% | 49.0% | 0.492 |
| | Gradient Boosting (XGBoost-type) | 69.6% | 49.1% | 0.492 |
| | SVR (RBF) | 63.6% | 42.9% | 0.429 |
| | MLP (128-64-32) | 69.1% | 47.1% | 0.470 |
| | **Hybrid ensemble** | **69.7%** | **49.2%** | **0.493** |
| Schedule | Median baseline | 57.2% | 33.3% | 0.243 |
| | Random Forest | 67.1% | 45.8% | 0.458 |
| | Gradient Boosting (XGBoost-type) | 66.8% | 45.4% | 0.454 |
| | SVR (RBF) | 67.9% | 45.6% | 0.456 |
| | MLP (128-64-32) | 65.1% | 43.7% | 0.437 |
| | **Hybrid ensemble** | **67.3%** | **45.4%** | **0.455** |

The hybrid reaches 69.7% accuracy for cost and 67.3% for schedule, against 51.7% and 57.2% for the always-High baseline. Macro-F1 is below 0.50 because the Moderate class is almost never predicted correctly (Fig. 10).

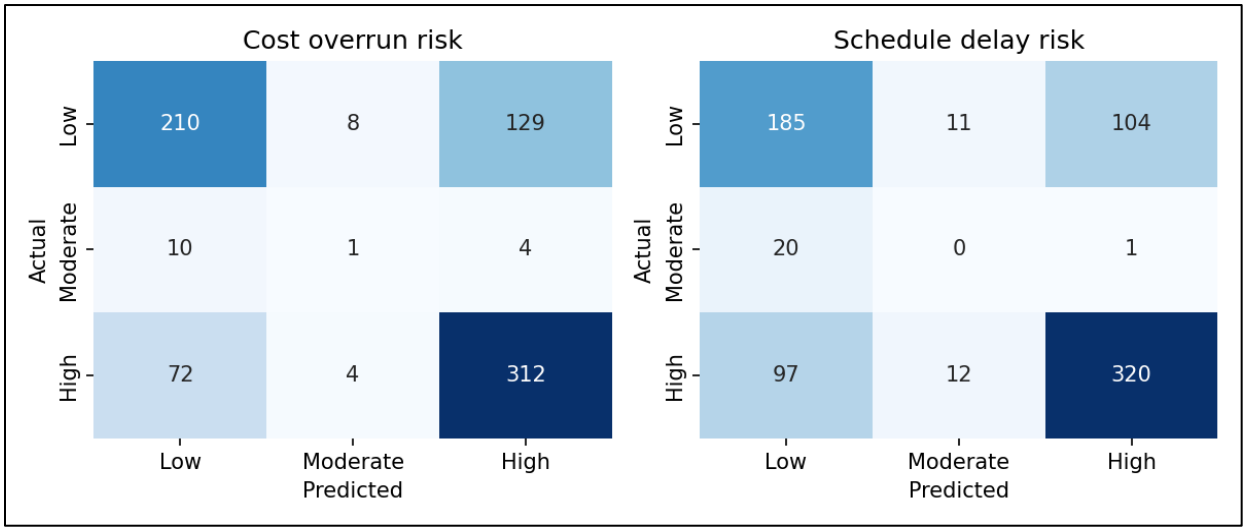



**Fig. 10. Confusion matrices of the hybrid ensemble's risk categories.**

**4.8 Binary Classification of the Dataset's Delay and Budget Overrun Labels**

**Table 10. Test-set classification results using initiation-time features only**

| Target | Model | Accuracy | F1-score | ROC-AUC |
|---|---|---|---|---|
| Delay | Majority baseline | 60.0% | 0.750 | 0.500 |
| | Random Forest | 68.8% | 0.732 | 0.767 |
| | Gradient Boosting (XGBoost-type) | 70.4% | 0.754 | 0.765 |
| | SVC (RBF) | 69.1% | 0.746 | 0.763 |
| | MLP | 70.1% | 0.757 | 0.769 |
| | Hybrid (mean prob.) | 69.5% | 0.750 | 0.770 |
| Budget Overrun | Majority baseline | 55.3% | 0.712 | 0.500 |
| | Random Forest | 71.6% | 0.755 | 0.801 |
| | Gradient Boosting (XGBoost-type) | 71.5% | 0.757 | 0.794 |
| | SVC (RBF) | 70.9% | 0.749 | 0.802 |
| | MLP | 70.7% | 0.751 | 0.793 |
| | Hybrid (mean prob.) | 71.1% | 0.753 | 0.801 |

International Journal of Advanced Research

Without leaked inputs, all models achieve an accuracy of roughly 69% to 72% and a ROC-AUC of about 0.76 to 0.80, a moderate improvement over the majority-class baseline (60% and 55%). This level is again explained by the planned duration and budget: a project with a very short plan is likely to exceed it, and one with a very long plan is likely to finish early, simply because the actual values are independent draws. The hybrid (mean probability) does not outperform the best single classifier.

## 5. DISCUSSION

Three conclusions follow. First, ML and DL models substantially outperform a median baseline on both regression targets, and the results are stable under cross-validation and bootstrap resampling. Second, the proposed hybrid ensemble is at least as good as the individual models but provides no statistically demonstrable gain over Random Forest or gradient boosting, and the validation-optimized weights discard the MLP entirely. Simple tree ensembles are thus the most economical choice here. Third, and most important, the predictive success is not evidence that project characteristics such as team size, requirement volatility, complexity, or developer experience forecast overruns. These characteristics have no measurable relationship with the targets, and the dominant signal comes from the denominator of the target formula.

These results are consistent with synthetic data in which the actual budget and duration are generated independently of the plan and of all other variables: the actual-versus-estimated correlations are near zero, requirement changes and complexity show no effect, and the two outcomes are independent of each other. If the dataset were real, the same evidence would imply that the planning process had no informational value, which is unlikely. We recommend verifying the data provenance before using these results for decision-making.

### 5.1 Practical Implications

- Predictions from this dataset should be treated as a demonstration of the pipeline, not as a validated forecasting tool for real IT projects.
- Percentage-based targets can create misleading $R^2$ values through the denominator; reporting

absolute-error targets (e.g., overrun in currency units or months) alongside percentages is advisable.

- Leakage audits (checking whether labels are deterministic functions of other columns) should precede any modeling.
- For real deployment, the framework should be rerun on organizational data containing richer predictors (project type, methodology, technology novelty, defect and change-request history, and time-stamped progress).

### 5.2 Limitations

- A single random split with one seed was used, and cross-validation was applied only to the two tree models.
- XGBoost and SHAP were unavailable, so scikit-learn gradient boosting and permutation importance were used; the MLP used L2 regularization instead of dropout.
- The hyperparameter search was limited to 10–12 random draws per model.
- No timestamps or project types were available, so chronological validation and cross-domain generalization could not be tested; an LSTM extension was therefore not applicable.
- Heavy-tailed targets make MAE and $R^2$ sensitive to a few extreme projects; log-transformed or quantile models could be explored.

## 6. CONCLUSION

This paper implemented the proposed hybrid ML and DL framework on 5,000 software projects and compared Random Forest, gradient boosting, SVR, an MLP, and a validation-weighted ensemble under a single leakage-controlled protocol. The best models reduced the MAE by about 35% relative to a median baseline, and the $R^2$ of the learned models ranged between 0.47 and 0.72. However, the hybrid did not significantly outperform the best single model, and almost all of the predictive power came from the planned budget and planned duration. The data audit indicates that this dataset cannot support conclusions about the managerial drivers of overruns and delays. Future work should validate the pipeline on authentic, time-stamped project records, include richer features, and test the LSTM and stacking extensions.

## REFERENCES

J. Galindo and S. Zavala, “A hybrid machine learning–operations research framework for adaptive resource allocation in complex projects,” 2026.

M. M. I. Shamim, A. B. B. A. Hamid, T. E. Nyamasvisva, and N. S. B. Rafi, “Advancement of artificial intelligence in cost estimation for project management success: A systematic review of machine learning, deep learning, regression, and hybrid models,” Modelling, vol. 6, no. 2, p. 35, 2025.

M. F. Khan, “Artificial intelligence in project scheduling management: A systematic literature review,” 2025.

S. Francios, “Optimizing project cost and duration forecasting through machine learning,” Doctoral dissertation, Politecnico di Torino, 2025.

S. Mustyala, P. Manchala, and M. Bisi, “Case studies: Machine learning approaches for software development effort estimation,” in Boosting Software Development Using Machine Learning. Cham, Switzerland: Springer Nature, 2025, pp. 95–126.

Ali et al., review of machine learning approaches for software effort prediction, covering datasets, estimation methods, and evaluation measures (as cited in the research design).

Review categorizing research on machine learning applications in software project management (as cited in the research design).

Study on hybrid machine-learning prediction of project cost and time with optimized neural networks, KSCE Journal of Civil Engineering (as cited in the research design).

J. Polaszek and W. Orzeszko, "Artificial intelligence in project KPIs prediction: A systematic scoping review of machine learning approach," Intelligent Data Analysis, 2026, Art. no. 1088467X261441796.

B. Taha, A. H. Ibrahim, and A. A. Soliman, "Risk-indexed artificial neural network for predicting duration and cost of irrigation canal-lining projects using survey-based calibration and Python validation," Scientific Reports, vol. 15, no. 1, Art. no. 40316, 2025.

T. Rahman, "Machine-learning-assisted project cost estimation for technology-intensive engineering programs," International Journal of Scientific Interdisciplinary Research, vol. 14, no. 12, pp. 1–54, 2024.

M. S. H. Hriday and A. Rehman, "Artificial intelligence and machine learning applications in construction project management: Enhancing scheduling, cost estimation, and risk mitigation," International Journal of Business and Economics Insights, vol. 5, no. 3, pp. 30–64, 2025.

A. S. Mali et al., "Application of artificial intelligence and machine learning in construction project management: A comparative study of predictive models," Asian Journal of Civil Engineering, vol. 26, no. 6, pp. 2671–2686, 2025.

M. Kussaiyn and R. M. Potluri, "Integrating ESG factors into cost forecasting for sustainable project management: Empirical evidence from Kazakhstan," World Development Sustainability, 2026, Art. no. 100279.

Naseem, S., Anwar, M., Farooq, U., Maqbool, M. S., Khan, M. M., Ahmad, F., & Raza, U. A. (2026). Advanced Detection of Pre-Ictal Stages in Epilepsy Using Convolutional Neural Networks and Automated Feature Extraction. Healthcare Technology Letters, 13(1), e70099.

S. J. Park, N. Nour, K. Y. Lee, and J. H. Kim, "Prediction of sewage pipeline construction duration by introducing machine learning and deep learning approaches," Journal of Civil Engineering and Management, vol. 31, no. 7, pp. 687–709, 2025.

A. Tiwari and A. Hussain, "Exploring artificial intelligence applications in construction using a black grey white box approach for predicting project schedule performance in India," Discover Computing, vol. 28, no. 1, Art. no. 295, 2025.

H. Sohrabi and E. Noorzai, "Risk-supported case-based reasoning approach for cost overrun estimation of water-related projects using machine learning," Engineering, Construction and Architectural Management, vol. 31, no. 2, pp. 544–570, 2024.

D. W. Debero and E. G. Sinesilassie, "Machine learning model for construction time prediction: A case of selected public building projects in Hosanna, Ethiopia," Journal of Engineering, vol. 2024, no. 1, Art. no. 5653690, 2024.

Aleem, A., Rafiq, M., Salahuddin, H., & Maqbool, M. S. (2026). BRAIN TUMOR DETECTION AND CLASSIFICATION USING DEEP LEARNING. International Journal of Advanced Research, 3(5), 4018-4036.

Maqbool, M. S., Zahra, S. R., Ismail, S., Nadeem, M., Fatima, N., & Ahmad, J. (2026). A CNN-BASED FRAMEWORK FOR EFFICIENT DETECTION OF EYE DISEASE IN FUNDUS IMAGES. Spectrum of Engineering Sciences, 4(4), 1157-1169.

Manzoor, I., Maqbool, M. S., Shahzad, F., Nadeem, M., Zulfiqar, A., & Naqvi, S. Q. (2026). A FRAMEWORK FOR HATE SPEECH IDENTIFICATION USING OPTIMIZED TEXT FEATURES AND NATURAL LANGUAGE PROCESSING ON TWITTER DATASET. Spectrum of Engineering Sciences, 4(6), 1389-1405.

Aslam, N., Maqbool, M. S., Nadeem, M., & Saleem, H. (2026). DEEPFAKESHIELD: ENHANCED VIDEO AUTHENTICITY DETECTION VIA CONVOLUTIONAL VISION TRANSFORMER. Spectrum of Engineering Sciences, 4(3), 1650-1665.

Tanveer, K., Aslam, N., Saleem, H., Maqbool, M. S., Akhter, M., & Rashid, M. H. (2026). INTEGRATING BLOCKCHAIN AND MACHINE LEARNING FOR ROBUST IOT DATA INTEGRITY IN CRITICAL INFRASTRUCTURES. Spectrum of Engineering Sciences, 4(3), 1875-1886.

Aslam, M., Maqbool, M. S., Aoun, M., Aslam, N., Razzaq, A. M., & Ali, S. (2026). HIGH-PERFORMANCE AND EFFICIENT BRAIN TUMOR SEGMENTATION FOR ENHANCED CLINICAL ANALYSIS. Spectrum of Engineering Sciences, 4(3), 195-210.

Israr, S., Maqbool, M. S., Hanif, I., Nadeem, M., Basit, A., & Batool, A. A. (2026). A ROBUST DEEP LEARNING FRAMEWORK FOR PREDICTING ACADEMIC SUCCESS USING ADVANCED NEURAL ARCHITECTURES. Spectrum of Engineering Sciences, 4(5), 1006-1022.

Nayab, S., Zahra, S., Mahjabeen, H., Majeed, B., & Maqbool, M. S. (2026). GLUCOTWIN: AN ARTIFICIAL INTELLIGENCE–DRIVEN INSULIN DOSAGE PREDICTION SYSTEM. Spectrum of Engineering Sciences, 4(6), 3265-3282.

Izhar, D. N. A., Maqbool, M. S., Nadeem, M., & Saleem, H. (2026). DEEPFAKESHIELD: ENHANCED VIDEO AUTHENTICITY DETECTION VIA CONVOLUTIONAL VISION TRANSFORMER. Spectrum of Engineering Sciences, 4 (3), 1650–1665.

Aslam, N., Meeran, M. T., Aslam, M., Maqbool, M. S., & Saeed, B. (2025). Understanding Urban Expansion Through Multi-Temporal Satellite Data Analysis. Kashf Journal of Multidisciplinary Research, 2(09), 252-273.

Maqbool, M. S., Fatima, N., Nazeer, R., Aslam, N., Abbas, F., Sumra, U., & Nadeem, M. (2025). A hybrid dataset-based ensemble strategy for efficient breast cancer detection. Kashf Journal of Multidisciplinary Research, 2(12), 39-57.

Hasnain, M. A., Ali, Z., Maqbool, M. S., & Aziz, M. (2024). X-ray image analysis for dental disease: A deep learning approach using efficientnets. VFAST Transactions on Software Engineering, 12(3), 147-165.

Hasnain, M. A., Ali, Z., Rehman, K. U., Ehtsham, M., & Maqbool, M. S. (2024). Idd-net: A deep learning approach for early detection of dental diseases using x-ray imaging. Journal of Computing & Biomedical Informatics, 7(02).

Abid, K., Aslam, N., Fuzail, M., Maqbool, M. S., & Sajid, K. (2023). An efficient deep learning approach for prediction of student performance using neural network. VFAST Transactions on Software Engineering, 11(4), 67-79.

Kanwal, F., Abid, M. K., Maqbool, M. S., Aslam, N., & Fuzail, M. (2023). Optimized classification of cardiovascular disease using machine learning paradigms. VFAST Transactions on Software Engineering, 11(2), 140-148.

Maqbool, M. S., Nazeer, R., Basit, A., & Zahra, K. (2023). Automated detection and localization of fungal infections on cotton leaves using YOLO-based object detection model. Machines and Algorithms, 2(2), 121-136.

Fazal, U., Khan, M., Maqbool, M. S., Bibi, H., & Nazeer, R. (2023). Sentiment analysis of omicron tweets by using machine learning models. VFAST Transactions on Software Engineering, 11(1), 67-75.

Maqbool, M. S., Hanif, I., Iqbal, S., Basit, A., & Shabbir, A. (2022). Optimized feature extraction and cross-lingual text reuse detection using ensemble machine learning models.

Basit, A., Hanif, I., Maqbool, M. S., Qayyum, W., Hasnain, M. A., & Nazeer, R. (2023). Cross-lingual information retrieval in a hybrid query model for optimality. Journal of Computing & Biomedical Informatics, 5(01), 130-141.

International Journal of Advanced Research