International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 11 | Nov 2026 www.irjet.net p-ISSN: 2395-0072

Predictive Scheduling in Construction: Bridging Machine Learning and Microsoft Project

Aryan Chawada[1], Gauri Chavake[2]

[1]*MBA-ACM Student, School of Construction, NICMAR University Pune, Pune*
[2]*B.Arch Student, SMMCA, Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur*

Abstract - *ML research in construction scheduling has progressed from project-level duration estimation to activity-level prediction, schedule-risk classification, dependency learning, and automated rescheduling. A comparative study evaluated KNN, SVR, gradient-boosting trees, and ANN for construction-duration forecasting [4]. Indian construction research has compared multiple linear regression, support vector regression, random forest, and gradient boosting, identifying planned duration, workforce size, equipment utilization, and material availability as important schedule-performance factors [3]. More recent work combines textual and numerical task features for improved construction-task duration prediction [1]. Other studies address direct schedule generation and revision: deep reinforcement learning optimizes schedules under precedence and resource constraints [5], while transformer-based approaches learn implicit activity dependencies and support lookahead-plan revisions [6]. Although these methods are technically advanced, they require substantial datasets, model development, and validation. Therefore, the proposed framework emphasizes incremental adoption, beginning with predictive analytics and progressing toward autonomous scheduling, while retaining human oversight throughout the scheduling process.*

Key Words: **Machine learning, Construction scheduling, Microsoft Project, Critical Path Method, Predictive analytics, Delay prediction, Human-in-the-loop, G+3 residential building**

1. INTRODUCTION

Construction project delivery is affected by uncertain productivity, resource availability, material delays, design changes, weather, rework and coordination problems. Schedule development therefore requires both formal network logic and engineering judgement. CPM-based software remains valuable because it translates activity durations and precedence relationships into a transparent project network. A 2025 study, for example, demonstrated the use of Microsoft Project and CPM to identify critical activities in a residential building schedule [13].

The limitation is not that CPM is incapable of scheduling. Rather, the quality of a deterministic schedule depends strongly on the quality of the durations and assumptions supplied by the planner. Historical project data contain information about how similar activities actually perform under particular crew, resource, site and project conditions. ML provides a means of learning patterns from those records and producing predictions that can be reviewed before a schedule is finalized.

Recent research demonstrates several directions: ML regression models have been compared for construction-duration prediction; Indian construction research has evaluated multiple ML models for schedule performance; and newer work has explored automated schedule revision, deep reinforcement learning and risk-aware scheduling.

This study therefore adopts a deliberately practical position: MS Project is retained as the scheduling system of record, while ML operates as a predictive decision-support layer. The planner remains responsible for accepting, rejecting or modifying AI-generated predictions. This human-in-the-loop structure is intended to be more realistic for ordinary construction organizations than a fully autonomous scheduling system.

1.1 Research Gap

Existing research has demonstrated ML-based duration prediction, schedule-performance classification and automated scheduling. However, the practical connection between these predictive outputs and the familiar CPM workflow used by planners remains an important implementation problem. Recent systematic work describes ML applications across planning, execution and monitoring and emphasizes data, processing, integration and output layers [14]. This paper narrows that problem to a practical MS Project-oriented workflow in which prediction outputs are explicitly reviewed by a planner before schedule changes are made.

1.2 Aim

To develop a practical conceptual framework for augmenting Microsoft Project-based construction scheduling with machine-learning predictions of activity duration and delay risk while retaining CPM logic and human planner control.

1.3 Objectives

- To identify the scheduling functions that can be complemented by ML without replacing CPM.

- To define the data inputs required for activity-duration and delay-risk prediction.
- To propose a seven-stage ML-MS Project integration workflow with a planner feedback loop.
- To demonstrate the framework using a representative G+3 residential building schedule with clearly labelled synthetic data.
- To define quantitative metrics and an experimental protocol for future empirical validation.

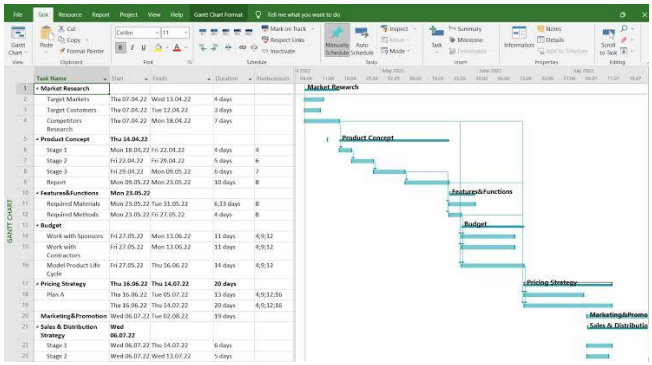

Figure 1: MS Project interface showing a Gantt chart view

2. LITERATURE REVIEW

ML research in construction scheduling has expanded from project-level duration estimation toward activity-level prediction, schedule-risk classification, dependency learning and automated rescheduling. A comparative study published in the Journal of Asian Architecture and Building Engineering evaluated KNN, SVR, gradient-boosting trees and ANN for forecasting construction duration [4].

Research specifically addressing Indian construction has compared multiple models including multiple linear regression, support vector regression, random forest and gradient boosting. The reported results indicate that model performance and interpretability can differ materially, with planned duration, workforce size, equipment utilization and material availability identified as important schedule-performance drivers [3].

More recent work has moved beyond simple duration prediction. A 2026 Advanced Engineering Informatics study proposed an ML model incorporating textual and numerical task features for construction-task duration prediction, illustrating the potential value of combining structured schedule records with richer task information [1].

Other studies investigate direct schedule generation or revision. Deep reinforcement learning has been used to optimize construction schedules under precedence and resource constraints [5], while transformer-based methods have been used to learn implicit activity dependencies and support lookahead-plan revisions [6]. These approaches are technically advanced, but they also require substantial datasets, model development and validation. The present framework therefore emphasizes incremental adoption: predictive analytics first, autonomous scheduling later.

3. MATERIALS AND METHODS

This study is a conceptual and applied framework study. It does not claim a field-validated ML experiment. The original G+3 case in the source manuscript is retained as a representative demonstration, but the numerical values used in the demonstration are treated as synthetic. This distinction is important because fabricated values must not be presented as measurements from a completed construction project.

3.1 Proposed Data Structure

A practical training dataset should contain one record per activity or work package, linked to project and progress information. Suggested features are listed below.

Table -1: Proposed activity-level data structure

Feature Group	Example Variables
Activity	Activity ID, activity type, work package, floor/zone
Schedule	Planned duration, predecessors, float, baseline dates
Resources	Crew size, labour hours, equipment utilization
Production	Quantity, productivity rate, work area
Site conditions	Weather exposure, access condition, work-front availability
Materials	Material availability, delivery delay, inspection status
Progress	Actual start, actual finish, percent complete, remaining duration
Risk	Previous delay, rework, design change, coordination issue
Target	Actual duration; delay flag/probability

3.2 Problem Formulation

For an activity i, the ML duration model can be represented as:

$$\hat{D}_i = f(X_i) \qquad (1)$$

where $\hat{D}_i$ is the predicted activity duration, X_i is the vector of activity, resource, site and progress features, and f is the trained regression model. A delay-risk model can be represented as:

$$P_i = P\ (Delay_i = 1 \mid X_i) \qquad (2)$$

where P_i is the predicted probability that the activity will exceed a predefined delay threshold. For example, a binary delay label may be defined as 1 when actual duration exceeds planned duration by more than 10%, and 0 otherwise. The threshold should be fixed before model evaluation and kept consistent across the dataset.

3.3 Candidate ML Models

Table -2: Candidate ML models for duration and risk prediction

Model	Primary Use	Advantage	Limitation
Linear Regression	Baseline duration model	Simple, interpretable	May miss nonlinear relationships
Random Forest	Duration/risk prediction	Handles nonlinear interactions; interpretable	Less efficient with very large feature spaces
Gradient Boosting / XGBoost	Duration/risk prediction	Strong predictive performance on tabular data	Requires tuning and careful validation
ANN / MLP	Complex nonlinear prediction	Can model complex relationships	Needs more data; less interpretable
NLP / Transformer	Delay-cause/dependency extraction	Uses text such as site diaries	Requires labelled text data

3.4 Proposed Hybrid Workflow

Table -3: Seven-stage ML-MS Project integration workflow

Sr.	Process	Output
1	Export historical MS Project task, resource and actual-progress records	Training dataset
2	Clean data, encode categories, handle missing values, define targets	Model-ready dataset
3	Train and validate regression/classification models	Duration model + risk model
4	Generate predicted duration/risk for a new schedule	Prediction table
5	Import predictions into MS Project custom fields	Planner-visible AI layer
6	Planner reviews and accepts/rejects changes	Approved schedule
7	Capture actual outcomes and retrain periodically	Continuous learning

The original five-stage concept is expanded here into seven operational steps to make data preparation and feedback explicit. The planner remains the final decision-maker. The model therefore provides evidence rather than an automatic instruction.

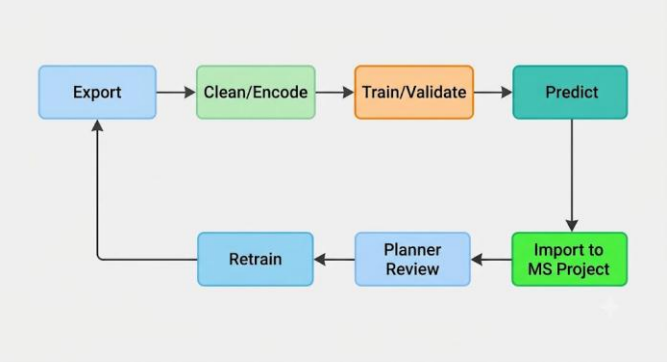

Figure 2: Seven-stage ML-MS Project hybrid workflow diagram

3.5 Evaluation Metrics

Regression performance should be evaluated using MAE, RMSE, MAPE and R^2. MAE is useful because it reports average absolute error in the original duration unit. RMSE penalizes larger errors more strongly. MAPE provides a percentage interpretation but should be used cautiously when actual durations approach zero.

$$MAE = (1/n)\ \Sigma\ |y_i - \hat{y}_i| \qquad (3)$$

$$RMSE = \sqrt{[(1/n)\ \Sigma\ (y_i - \hat{y}_i)^2]} \qquad (4)$$

$$MAPE = (100/n)\ \Sigma\ |(y_i - \hat{y}_i)/y_i| \qquad (5)$$

$$R^2 = 1 - [\Sigma(y_i - \hat{y}_i)^2 / \Sigma(y_i - \bar{y})^2] \qquad (6)$$

For delay-risk classification, accuracy alone should not be used because a dataset may contain many low-risk activities. Precision, recall, F1-score and ROC-AUC should be reported together with a confusion matrix.

3.6 Validation Protocol

To avoid optimistic results, project-level rather than random row-level splitting is recommended. Activities from the same project can be strongly correlated; placing records from one project in both training and test sets can cause information leakage. A stronger design is to train on earlier projects and test on later, unseen projects. Where sufficient projects exist, grouped cross-validation can be used. Hyperparameter tuning should be performed only within the training data.

The final comparison should include at least one conventional baseline, such as planner/MS Project planned duration, and one or more ML models. This makes it possible to determine whether the ML layer provides useful predictive improvement rather than merely producing another estimate.

4. ILLUSTRATIVE G+3 CASE APPLICATIONS

A representative G+3 residential building is used to demonstrate how predictions could appear beside a conventional MS Project schedule. The following values are synthetic demonstration values and must not be interpreted as field observations or model-validated results.

The demonstration shows the intended use of the AI layer: a planner may retain the original estimate, but the ML system highlights activities with wider uncertainty and higher predicted delay risk. In a real dataset, the range would be generated from model uncertainty or predictive intervals rather than manually assigned.

Table -4: Illustrative G+3 schedule with synthetic ML outputs

Activity	Plan.	ML Range	Risk	Illus. Actual
Site prep. & excavation	6 d	6-8 d	Low	7 d
PCC & foundation works	10 d	10-13 d	Med	12 d
Plinth & GF RCC	8 d	8-11 d	Med	10 d
Typical floor RCC cycle	5 d	5-7 d	Low	6 d
Brickwork & plaster	20 d	20-26 d	High	25 d
Plumbing/elec. 1st fix	12 d	12-15 d	Med	14 d
Finishing & handover	18 d	18-24 d	High	23 d

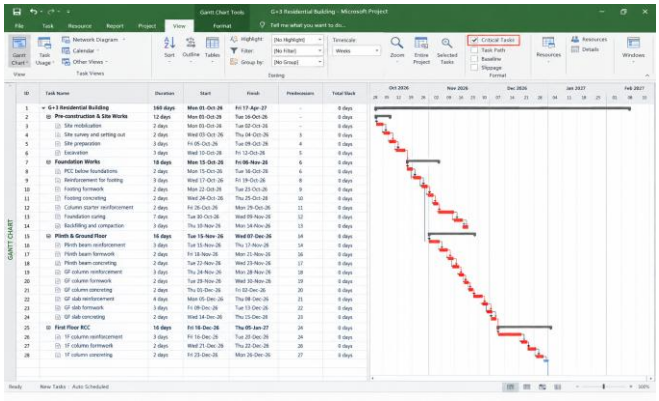

Figure 3: MS Project Gantt chart of the G+3 case study with critical path highlighted

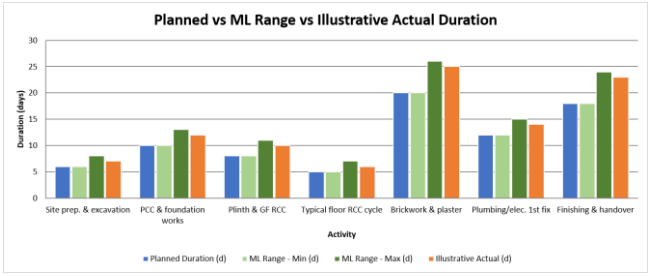

Chart -1: Planned vs. ML range vs. illustrative actual duration

4.1 Example MS Project Integration

The ML outputs can be represented through custom fields rather than replacing the normal Duration, Start, Finish and Predecessors fields. Suggested custom fields include AI Predicted Duration, AI Lower Bound, AI Upper Bound, AI Delay Probability, AI Risk Class and AI Recommendation. The standard CPM fields remain the authoritative schedule logic.

Table -5: Suggested MS Project custom fields for the AI layer

MS Project Field	Purpose
Duration	Planner-approved schedule duration
AI Predicted Duration	ML point prediction
AI Lower Bound	Lower predictive estimate
AI Upper Bound	Upper predictive estimate
AI Delay Probability	Predicted probability of threshold exceedance
AI Risk Class	Low / Medium / High
AI Model Version	Model used to generate prediction
Actual Duration	Observed outcome for feedback

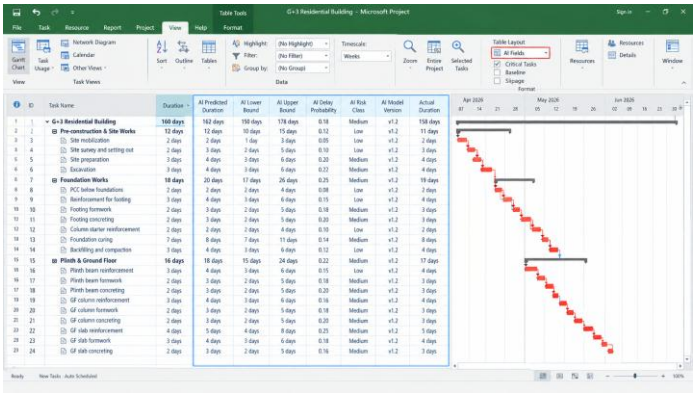

Figure 4: MS Project custom fields view with AI Predicted Duration, AI Risk Class, etc. added

4.2 Comparative Decision Logic

Table -6: Traditional vs. ML-augmented decision logic in MS Project

Parameter	Traditional MSP	ML-Augmented MSP
Duration basis	Planner single value	Planner value + ML prediction/range
Delay visibility	Critical path/float based	Explicit predicted risk
Historical learning	Manual	Retraining from completed projects
Resource insight	Planner/resource levelling	ML-informed prediction + planner review
Decision authority	Planner	Planner
Integration effort	Existing workflow	External data/ML pipeline required
Main risk	Subjective duration assumptions	Data quality/model error

5. DISCUSSION

The proposed framework changes the role of historical schedule data. In a conventional workflow, a completed project may remain primarily as a record. In the hybrid workflow, completed schedules become training examples. This creates the possibility of cumulative organizational learning: similar activities on future projects can be estimated using evidence from prior work rather than relying exclusively on individual judgement.

The framework also preserves the strengths of CPM. ML is not well suited to replacing explicit contractual logic, mandatory sequencing, engineering constraints or planner judgement in every situation. CPM remains useful for representing precedence and calculating the schedule network. ML instead addresses uncertainty around the values and risks supplied to that network.

Interpretability is particularly important. A high-risk prediction should be accompanied, where possible, by contributing factors such as low historical productivity, high crew variability, predecessor slippage or material availability. Recent Indian research has demonstrated the usefulness of feature-importance and sensitivity analysis for understanding schedule-performance models [3].

The main organizational challenge is data quality. A firm cannot obtain reliable predictions from poorly maintained schedules. Baselines, actual start and finish dates, quantities, resources and delay causes must be recorded consistently. This means that digital discipline is a prerequisite for AI adoption.

The second challenge is integration. A practical implementation may use Excel/CSV exports, Python or another analytics environment and custom fields in MS Project. Automated integration through APIs or scripts can reduce manual handling, but the integration layer itself must be tested for version compatibility and data consistency.

The third challenge is generalization. A model trained on low-rise residential projects may perform poorly on industrial, infrastructure or high-rise projects. Project-type segmentation, transfer learning or retraining may therefore be required. Recent research also indicates that different ML methods perform differently depending on the target and dataset, supporting the use of comparative model evaluation rather than assuming that one algorithm is universally superior.

6. PRACTICAL IMPLEMENTATION ROADMAP

Table -7: Staged implementation roadmap for construction firms

Phase	Action	Recommended Output
1.Data discipline	Save baselines, actual dates and delay causes	Clean historical schedule archive
2.Dataset creation	Export project/activity/resource records	Training dataset
3.Baseline model	Build simple regression/classification benchmark	Baseline MAE/RMSE/F1

Phase	Action	Recommended Output
4.ML comparison	Compare RF, boosting and ANN where justified	Model performance table
5. Integration	Write predictions into MS Project custom fields	AI-augmented schedule
6. Pilot	Run on one project without automatic schedule changes	Planner feedback
7. Scale	Retrain with completed-project outcomes	Organization-specific model

7. LIMITATIONS

- The G+3 numerical example is synthetic and illustrative; it is not evidence of ML accuracy.
- The paper proposes an evaluation methodology but does not claim a field-validated model.
- ML performance depends strongly on the quantity, quality and consistency of historical schedule data.
- Construction projects differ in technology, location, contractor capability and procurement strategy; therefore model transferability cannot be assumed.
- Delay-risk labels depend on a predefined threshold and may require domain-specific calibration.
- External integration between MS Project and an ML environment may require scripting, data transformation and maintenance.
- Fully autonomous schedule generation is outside the scope of this study

8. CONCLUSION

This paper proposes a practical hybrid approach in which machine learning complements rather than replaces Microsoft Project and CPM-based construction scheduling. The central idea is to use historical project data to predict activity duration and delay risk, expose those predictions through custom scheduling fields, and allow the planner to make the final decision. The approach preserves the transparency and established logic of CPM while adding a data-driven layer for uncertainty and risk.

The proposed workflow is particularly appropriate as a staged digital-transformation strategy. A firm can begin by improving schedule data capture, then develop a baseline prediction model, compare algorithms, integrate predictions into MS Project and finally introduce feedback-based retraining. The approach does not require an organization to abandon its existing scheduling software at the beginning of AI adoption.

The most important next step is empirical validation. A future study should assemble a multi-project dataset containing historical durations, resource conditions, quantities, predecessor performance, material availability and delay causes; train and compare several ML models; and evaluate them on unseen projects using MAE, RMSE, MAPE, R^2, precision, recall and F1-score. Such an experiment would establish whether the proposed hybrid workflow produces measurable improvement over conventional planner estimates.

Future extensions may connect the framework with BIM/4D planning, IoT-based progress capture, natural-language processing of site diaries, explainable AI and automated lookahead-plan revision. Recent research already demonstrates the feasibility of these individual directions, including ML/BIM schedule generation, dependency learning and deep reinforcement learning.

REFERENCES

[1] Y. Liu and H. Luo, "An efficient machine learning-based model for duration prediction of construction tasks with large-scale datasets," Advanced Engineering Informatics, vol. 69, 2026, Art. 103820. DOI: 10.1016/j.aei.2025.103820.

[2] C.-L. Fan, "Optimization and performance evaluation of machine learning classifiers for predicting construction quality and schedule," Automation in Construction, vol. 179, 2025, Art. 106470. DOI: 10.1016/j.autcon.2025.106470.

[3] 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, 2025, Art. 295. DOI: 10.1007/s10791-025-09769-x.

[4] "A comparative study of machine learning regression models for predicting construction duration," Journal of Asian Architecture and Building Engineering, 2023. DOI: 10.1080/13467581.2023.2278887.

[5] Y. Yao, V. W. Y. Tam, J. Wang, K. N. Le, and A. Butera, "Automated construction scheduling using deep reinforcement learning with valid action sampling," Automation in Construction, vol. 166, 2024, Art. 105622. DOI: 10.1016/j.autcon.2024.105622.

[6] F. Amer, Y. Jung, and M. Golparvar-Fard, "Construction schedule augmentation with implicit dependency constraints and automated generation of lookahead plan revisions," Automation in Construction, vol. 152, 2023, Art. 104896. DOI: 10.1016/j.autcon.2023.104896.

[7] "Generation of Construction Scheduling through Machine Learning and BIM: A Blueprint," Buildings, vol. 14, no. 4, 2024, Art. 934. DOI: 10.3390/buildings14040934.

[8] R. Shyam and S. Tiwari, "Predictive Assessment of Construction Delays on Schedule, Cost, and Quality Performance Using Machine Learning and Multi-Criteria Decision Analysis," Journal of The Institution of Engineers (India): Series A, 2026. DOI: 10.1007/s40030-026-00989-y.

[9] F. Fitzsimmons, Y. Hong, and I. Brilakis, "Improving Construction Project Schedules before Execution," Proceedings of the 37th ISARC, 2020, pp. 1144-1151. DOI: 10.22260/ISARC2020/0157.

[10] C. Le and H. D. Jeong, "Artificial Intelligence Framework for Developing a Critical Path Schedule Using Historical Daily Work Report Data," Construction Research Congress 2020, ASCE, 2020. DOI: 10.1061/9780784482889.059.

[11] J. Peng and C. Chen, "Deep learning and heuristic exploration for risk-aware scheduling in construction projects," Automation in Construction, vol. 187, 2026, Art. 106957. DOI: 10.1016/j.autcon.2026.106957.

[12] M. Abdelmottaleb Ali, "AI in Project Management: Machine Learning Application for Construction Scheduling," University of Salford, 2026.

[13] Suramli et al., "Critical path analysis scheduling project with critical path method (CPM) using application Microsoft Project on type 78 house construction project," Enrichment: Journal of Management, vol. 15, no. 2, 2025. DOI: 10.35335/enrichment.v15i2.2263.

[14] S. Sadeghi and C. Niu, "Machine learning-enabled construction project management: systematic review, comparative performance synthesis and implementation framework," Built Environment Project and Asset Management, vol. 16, no. 3, pp. 461-483, 2026. DOI: 10.1108/BEPAM-11-2024-0264.