

DATA-DRIVEN DECISION SUPPORT SYSTEM FOR RESOURCE ALLOCATION TO CAST IN- SITU PILES OF LINEAR PROJECTS

Ramachandran Anantharaam¹

ABSTRACT

Field observations from an active cast-in-situ piling site revealed that resource-related waiting contributed to 54% of total operation lead time, disrupting workflow and production reliability. This study investigates whether near-term resource requirements in piling operations can be predicted using process-state information available during execution. A data-driven framework is developed to predict demand for key piling resources (water, tremie pipe, rebar cage, casing, etc.) within a 30-minute lookahead window. Time-stamped operational data was collected from 13 piles, capturing features such as pile geometry, current activity state, depth progression rate, subsurface conditions and recent execution disruptions. Resource demand was framed as a binary classification problem for each resource type. Random Forest classifiers¹ were trained on 80% of the data and validated with the rest. The models achieved strong performance: F1-scores of 0.90 (water), 0.97 (tremie pipe), 0.95 (rebar cage) and so on. Deploying these predictors reduced resource-induced waiting to 12% and increased Kelly efficiency by 16%. This work operationalizes predictive resource planning at the activity level, advancing lean construction practice by enhancing flow reliability. Future research could integrate these predictors into Last Planner System workflows, couple them with discrete-event simulation or digital twins for system-level impact assessment, and generalize to other resource-sensitive construction activities.

KEYWORDS

Resource prediction, make-ready planning, waiting time, production planning, machine learning for construction.

INTRODUCTION

Pile foundations are critical structural components in heavy civil engineering projects, where structures are subjected to large loads and founded on highly variable subsurface conditions (Fischer et al., 2021). In large-scale infrastructure projects, pile foundation works often constitute a substantial portion of the overall project cost. Wang (2015) discovered that the pile construction alone can account for approximately 5% to 28% of the total project cost. Despite this centrality, piling operations are frequently constrained by a combination of technical (Choudhry et al., 2014), organizational, and managerial factors, including workforce and equipment availability, specialized expertise, design complexity, safety considerations, and contractual and health risks (Ehsan et al., 2010)

¹ Master of Technology, Department of Civil Engineering, Indian Institute of Technology Madras, Chennai, India, ananthu_357@alumni.iitm.ac.in, orcid.org/0009-0007-8503-1867

¹ Random Forest Classifiers - A ensemble machine learning method that constructs multiple decision trees and aggregates their predictions for improved accuracy and reduced overfitting.

Cast-in-situ piling methods are widely adopted in dense urban environments due to their adaptability to dynamic environments (Shelke et al., 2024). In linear infrastructure projects such as metro corridors, it is observed that project teams often pool key resources along the entire alignment and deploy them opportunistically at multiple accessible piling fronts, which magnifies interdependence and makes local disruptions (e.g., a missing casing or delayed muck removal) propagate as systemic delays. For example, the unavailability of a casing liner at the moment of need may cause sidewall instability and collapse, forcing re-cleaning with a cleaning bucket before casing installation, thereby compounding waiting time and rework. Moreover, subsurface information derived from preliminary geotechnical investigation reports is often sparse or generalized, providing limited resolution of depth-wise variability. As actual ground conditions encountered during execution deviate from assumed profiles, activity durations and resource requirements can change dynamically (Zayed et al., 2005), undermining planned resource allocation. Such deviations and uncertainties demand unforeseen requirement of resources which when not available leads to waiting delays.

From a lean construction perspective, waiting is recognized as a fundamental form of waste that disrupts flow and increases variability within production systems (Ohno, 1988; Koskela, 1992). Lean theory emphasizes the stabilization of production through proactive planning and the removal of constraints before work execution. In piling operations, this principle is operationalized through make-ready planning, which aims to ensure that prerequisites such as resources, information, and equipment are available when needed (Porwal et al., 2010). However, in practice, make-ready planning for piling often remains reactive. While activity durations may be estimated using regression-based or deterministic models (e.g., productivity-based duration prediction approaches), such methods typically assume stable conditions and are limited in their ability to capture the dynamic variability inherent in piling execution. Consequently, resource mobilization decisions are frequently based on experience and reactive signals rather than systematic anticipation of near-term needs.

This gap between lean planning principles and operational execution highlights an opportunity for improvement. Modern construction sites routinely generate execution-level data describing activity states, depth progression, subsurface encounters, and disruptions. Even when available only at project scale, such “small data” captures the evolving state of the production system and can potentially be leveraged to support short-horizon decision-making. Rather than predicting exact activity completion times, an alternative and more actionable question emerges: will a specific resource be required in the immediate future to sustain uninterrupted execution?

Accordingly, the objectives of this study are to 1) develop machine learning classifiers that predict near-term resource requirements in cast-in-situ piling operations using process state information, and 2) evaluate their potential to support proactive make-ready planning. By framing resource demand as a short-horizon prediction problem and evaluating it using real site data, this research aims to demonstrate a practical, lean-aligned approach to reducing waiting waste and improving flow reliability in complex piling operations.

REVIEW OF LITERATURE

Construction productivity research has identified numerous factors that influence performance, including workforce availability, management practices, supervision quality, coordination, and safety, and has emphasized the need for dynamic modelling approaches to represent their combined effects over time. In this regard, Discrete Event Simulation methods has been widely applied to model construction resource flows and workflow variability by chronologically queueing the events, and new events scheduled dynamically (Tommelein., 1997), enabling lean process redesign through virtual experimentation. A variety of statistical and machine-learning methods, such as linear regression, fuzzy logic, artificial neural networks, support vector

machines, and random forests, have also been applied to predict productivity or related indicators in different construction contexts. These models demonstrate that data-driven techniques can capture complex, nonlinear relationships between influencing factors and productivity as in Mostofi et al. (2024), but they typically focus on estimating overall rates or durations rather than specifying detailed, time-specific resource demands for individual operations.

A recent work by Liu et al. (2025) has advanced duration prediction at the task level by combining large-scale historical schedule data with machine-learning and feature-engineering techniques. For example, a hybrid model integrating clustering, locally weighted support vector regression, and optimized feature sets has been shown to predict construction task durations accurately across more than 140,000 activities, using both numerical and text-derived features. In pile-specific contexts, Vikktor AI (2022) proposes artificial neural networks that have been applied to predict pile driving speeds from cone penetration test (CPT) data, achieving low error margins for operational planning in deep foundation projects. Similarly, hybrid machine learning-ANN models have been developed for early cost estimation of pile foundations, combining feature selection with neural networks to improve prediction accuracy over traditional methods (Deepa et al., 2023). These approaches illustrate how rich geotechnical and schedule datasets can enhance forecasting of pile-related durations and costs, yet they remain largely oriented toward planning-stage assessments and do not explicitly model the real-time execution state or short-horizon resource mobilization decisions during pile cycles.

In parallel, data-driven resource planning and recommendation systems have emerged in construction management. Mostofi et al. (2024) developed graph-based recommendation framework using node2vec embeddings has been proposed to support construction productivity planning by learning relationships among activities and recommending productive resource combinations in terms of budgeted units, person-hours, and earned value. Another study by Wu et al. (2020) on industrial modular construction developed an intelligent, semi-supervised approach that links low-detail BIM models of modules to historical labor requirements, enabling more objective preliminary resource planning when detailed engineering information is not yet available. For pile foundations specifically, machine learning frameworks have been developed by Bello et al (2025) with genetic algorithm feature selection have been used to predict settlement under load, identifying applied load and pile length as key predictors. These systems demonstrate the potential of machine learning and graph-based representations to inform resource planning, but they operate mainly at project, work-package, or modular scales and are not tailored to the fine-grained, cycle-level dynamics of operations such as cast-in-situ piling.

Finally, detailed analyses of in-situ pile production using work-density and time-distribution methods by Fischer et al. (2024) shows that production is highly sensitive to the interaction between process design, ground conditions, and site management, and have highlighted the presence of significant interruptions and low-productivity segments within piling cycles. While such studies provide a rich descriptive basis for understanding where and why inefficiencies occur, they stop short of converting execution-state observations into predictive signals for resource coordination. Together, these strands of literature indicate that (1) powerful models exist for predicting productivity, durations, costs, and settlements in pile construction, and (2) recommendation systems can support resource selection at higher levels, but (3) there remains a gap in short-horizon, state-aware prediction of resource demand within specific, sequential operations like cast-in-situ piling. Addressing this gap motivates the use of Random Forest classifiers, an ensemble machine learning method that constructs multiple decision trees and aggregates their predictions for improved accuracy and reduced overfitting (Salman et al., 2024), on execution-state data to anticipate near-term resource requirements and embed these predictions into make-ready planning.

METHODOLOGY

This section details the systematic data collection process, waiting waste quantification, dataset construction, feature engineering, predictive modelling framework, and comprehensive model training and validation strategy employed in this study.

DATA COLLECTION AND WASTE QUANTIFICATION

The data for the study was collected from an active metro rail corridor construction site in the city of Chennai, India. Process states were classified into 16 mutually exclusive categories, encompassing all production, support, and idle conditions. Cycle analysis of 13 piles revealed an average Kelly bar utilization of 59%, confirming substantial idle capacity. The process flow **Error! Reference source not found.** shows the 46-hour average cycle as follows:

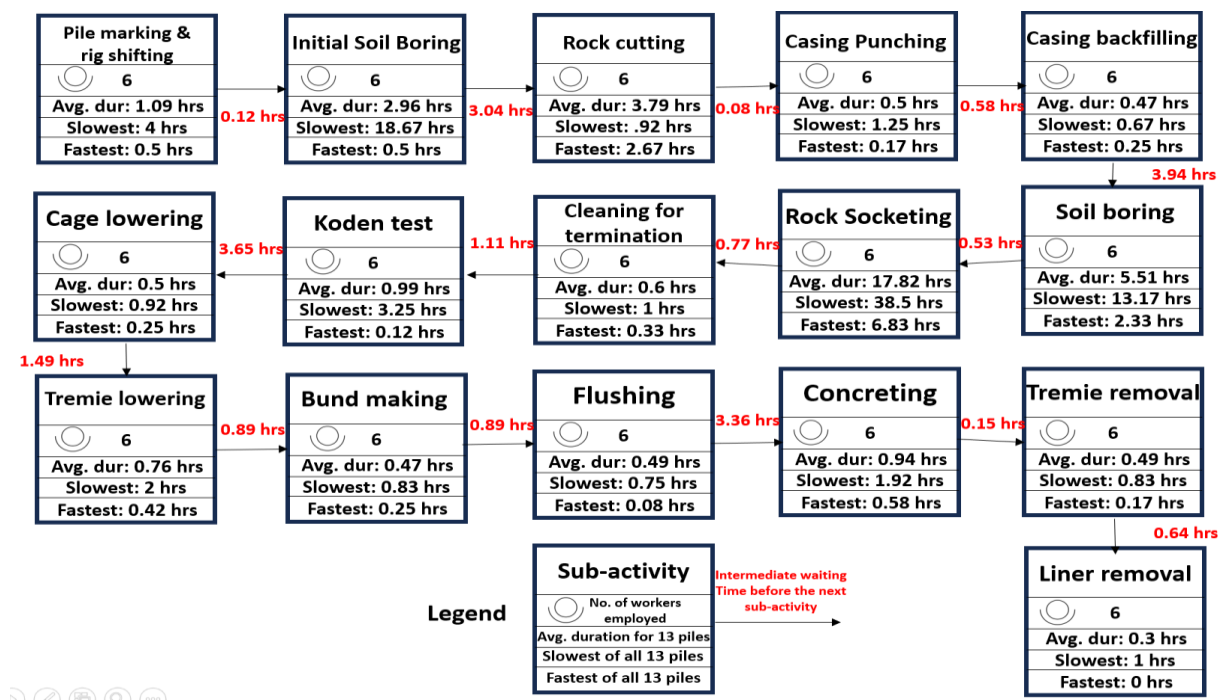


Figure 1: Cast In-situ piling cycle, averaged for 13 piles

Operational data for the 13 piles (varying diameters/lengths) were collected using: (1) structured time-lapse observation logging process states every 10 minutes via mobile checklists (capturing activity, depth, disruptions), and (2) rig telematics from Kelly bar sensors (depth progression, utilization). This yielded 4,339 timestamped records (mean 333/pile) spanning full lifecycles from rig mobilization to concrete cutoff for all the 13 piles under study. Piles traversed heterogeneous subsurface conditions comprising different strata such as filled soil, clayey silt, silty sand, and disintegrated to slightly weathered rock at different depths, as verified by standard penetration tests and real-time drilling observations.

The resource pool, R , considered for the study constitute the complete shared resources that are circulated as per requirement, across the project's multiple piling fronts:

$R = \{ \text{Excavator, Water (pumping), Cleaning Bucket, Casing Liner, Pick \& Carry Crane, Koden Testing Machine}^{\text{II}}, \text{Rebar Cage, Tremie Pipe/Hopper, Transit Mixer} \}$,
assuming rig and Kelly bar remain positioned throughout the piling cycle for each pile.

The waste decomposition chart as shown in Figure 2 revealed that around 45% of the waiting time is caused by unavailability of the resources to carry out the work at front. These

^{II} Koden Testing Machine - an ultrasonic drilling monitor that measures borehole diameter, verticality, depth, and sediment in piling operations using sonic echo sensing.

findings substantiate the research motivation, establishing resource coordination as the highest-leverage intervention for waste elimination in equipment-constrained piling operations. The resulting dataset provides a foundation for state-aware resource prediction.

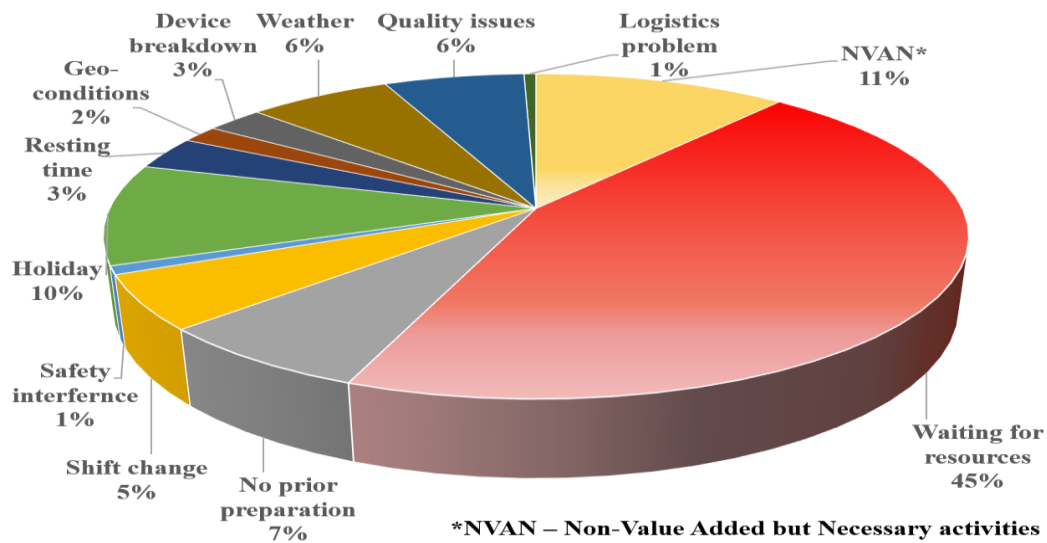


Figure 2: Waste decomposition pie chart

DETERMINING LOOKAHEAD WINDOW

A critical design decision in the predictive framework is the selection of an appropriate lookahead horizon H . Mobilization lead times were recorded by measuring time elapsed from the resource request to resource arrival for each of the nine shared resources.

Table 1: Resource-wise percentile distribution of availability lead time

Resource	Mean Lead (min)	75th %ile	95th %ile	30min Coverage (%)
Excavator	15	22	28	89
Water (pumping)	12	18	25	92
Cleaning Bucket	10	15	22	95
Casing Liner	25	32	40	82
Pick & Carry Crane	20	25	32	85
Koden Testing Machine	35	42	50	78
Rebar Cage	22	28	35	84
Tremie Pipe/Hopper	15	20	28	88
Transit Mixer	22	28	35	82

For each resource type, lead times were aggregated across all occurrences, yielding distributions from which mean, 75th and 95th percentiles were computed. From this lead time analysis, it is found that a 30-minute lookahead window captured 82% to 95% of resource needs across all resources as shown in **Error! Reference source not found.** Shorter horizons (e.g., 20 minutes) reduced coverage to approximately 72%, increasing the risk of resource stockouts. Although specialized resources such as the Koden testing machine exhibited longer lead times, scheduling

logs confirmed that a 30-minute horizon limited idle penalties to under 5%, validating its practical suitability. Hence a 30-minute lookahead window for resource prediction is chosen for the study.

FEATURE ENGINEERING

In order to model the Random Tree classifier, a set of features that can effectively encode the execution state and influence the upcoming resource demand, are required.

Table 2: Description and notations of selected features

Feature	Category	Representation	Operational Relevance
Pile Length	Geometry	L (m) – from design	Influences overall sequencing and downstream resource needs
Pile Diameter	Geometry	D (m) – from design	Affects excavation rate, stability and equipment selection
Cutoff Depth	Geometry	C (m) - from design	
Current Activity ID	Execution State	One-hot (1 - 16)	Indicates phase – specific resource
Elapsed Time in Activity	Execution State	$T_{a1} = 10$ $T_{a2} = 20$ A_t - total activity time	Signals proximity to activity transitions
Cumulative Time	Temporal Context	$P_1 = 10$ $P_2 = 20$ $P_n = P_t$ (total pile cycle time)	Captures global progression of pile execution
Current Depth	Depth Progression	D_t (m)	Reflects execution progress and stability conditions
Depth to Next Stratum	Depth Progression	$D_n - D_t$ (m) (from the soil report)	Anticipates upcoming changes in drilling behavior
Next Stratum SPT-N	Subsurface condition	N_{next} (from soil report)	Indicates resistance and potential corrective needs
Delay Recency	Disruption	R_t index	Captures short-term execution instability

The features listed in Table 2 were selected to comprehensively encode the core determinants of resource demand during piling operations:

- Physical features – capturing the geometry of the pile and associated structures (say pile cap cutoff)
- Process dynamics – capturing the activity progression and process disruptions
- Contextual factors – capturing the subsurface and temporal contexts during operations, while adhering to real-time availability during execution, all computable without future knowledge.

To capture the influence of execution interruptions without introducing cumulative bias, delays were encoded using a delay recency index based on exponential decay. Let $D_t \in \{0,1\}$ indicate whether execution was interrupted during time step t . The delay recency index R_t was computed as:

$$R_t = \alpha R_{t-1} + D_t,$$

where $\alpha \in (0,1)$ is a decay parameter controlling the memory of past disruptions. This formulation allows recent delays to exert stronger influence while progressively discounting older interruptions, reflecting the transient nature of execution instability and borehole exposure effects. These state representations feed the per-resource Random Forest classifiers detailed in the upcoming section.

RANDOM FOREST CLASSIFIERS

Near-term resource demand prediction was formulated as a supervised binary classification problem and implemented using Random Forest classifiers. At timestamp t , the execution state was encoded via the 10 features from Table 2.. For every resource $r \in R$, binary targets were forward-labeled as:

$$y_{r,t} = \begin{cases} 1 & \text{if } r \text{ required in } (t, t + 30] \\ 0 & \text{otherwise} \end{cases}$$

The choice of Random Forests was motivated by their ability to capture non-linear interactions between execution-state variables, their robustness to noise (Salman et al., 2024), and their suitability for project-scale datasets with mixed feature types.

Decision Tree Mechanics

The Random Forest classifier is modelled to construct an ensemble of 300 decision trees, each trained on bootstrap-sampled subsets of the feature matrix (with replacement, $\sim 63.2\%$ unique samples per tree). Feature subsampling at each split ($m_{\text{try}} = \sqrt{10} \approx 3$) decorrelates the trees, reducing overfitting while capturing nonlinear-state resource interactions critical for piling transitions.

```
RandomForestClassifiers(
    N_estimators = 300,      #Ensemble size for stability
    Max_depth = 10,         #Overfitting control
    Min_sample_leaf = 5,    #Leaf purity threshold
    Class_weight = 'balanced', #Imbalance correction
    random_state = 42
)
```

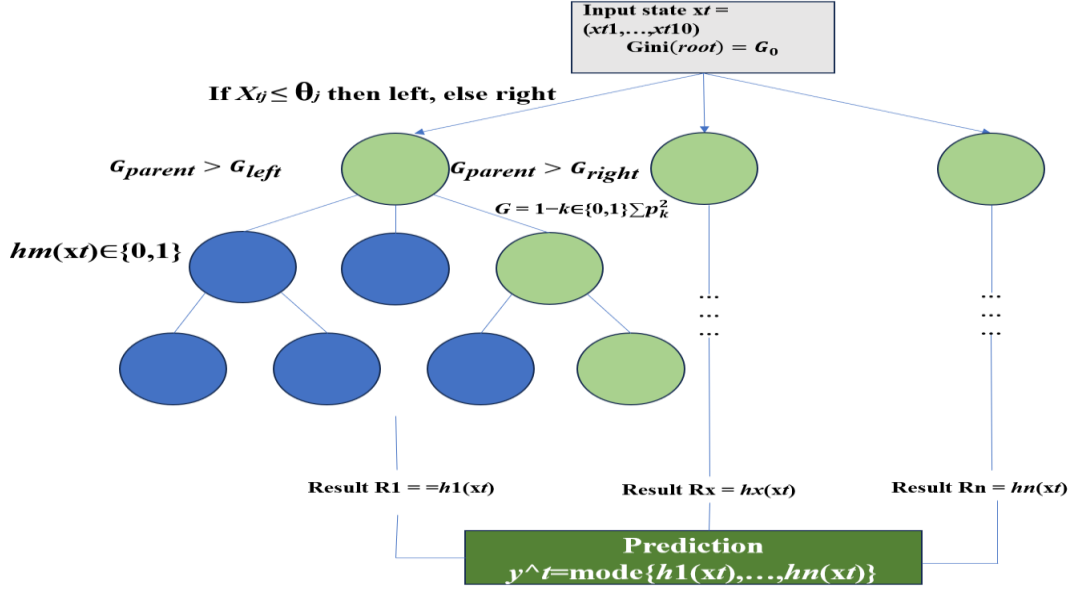


Figure 3: Representation of the Random Forest Ensemble model

Figure 3 illustrates how a single decision tree in the Random Forest partitions the input state $x^r = (x_1, \dots, x_{10})$ to predict whether a given resource will be needed in the next 30 minutes ($h_m(x) \in \{0,1\}$). At each internal node, the tree considers a split of the form “if feature $X_j \leq \theta_l$, then go left, else go right”, and selects the threshold θ_l that yields the largest reduction in Gini impurity (G),

$$Gini = 1 - \sum_{k=0}^1 p_k^2$$

where p_k = class proportion in node, between the parent node and its two child nodes. This recursive process continues, creating successively purer regions of the feature space (nodes dominated by either “need” or “no-need” cases), until a stopping criterion is met (maximum depth of 10 levels or a minimum of 5 samples in a leaf node). The blue and green terminal nodes in the figure represent such leaf regions, each storing a class label based on the majority of training samples that fall into that region. For a new execution state, each tree $h_m(x)$ outputs a binary prediction at its leaf, and the Random Forest aggregates these outputs across all trees; the final class $\hat{y}$ is obtained by majority voting over $\{h_1(x), \dots, h_n(x)\}$, which is equivalent to applying a decision threshold of 0.5 on the proportion of trees voting for the “resource needed” class.

$$\hat{y}_{r,t} = \mathbb{I}\left(\frac{1}{T} \sum_{t=1}^{300} \mathbb{I}(tree_t(x) = 1) > 0.5\right)$$

MODEL EVALUATION AND VALIDATION STRATEGY

Model performance was evaluated with emphasis on anticipating imminent resource requirements rather than overall classification accuracy. For each resource $r \in \mathcal{R}$, predictions were compared against ground truth to construct a confusion matrix comprising true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN). Class-specific precision^{III}, recall^{IV}, and F1-score^V were computed as:

$$\text{Precision}_r = \frac{TP_r}{TP_r + FP_r}, \text{Recall}_r = \frac{TP_r}{TP_r + FN_r}$$

^{III} Precision - ratio of true positives to total predicted positives

^{IV} Recall - ratio of true positives to total actual positives

^V F1 - score - harmonic mean of precision and recall

$$F1_r = 2 \cdot \frac{\text{Precision}_r \cdot \text{Recall}_r}{\text{Precision}_r + \text{Recall}_r}$$

The F1-score was adopted as the primary metric, as it balances alert reliability and coverage under class imbalance.

In addition to offline testing, a shadow-mode validation framework is proposed in which models generate predictions using live execution-state inputs. Observed resource requests within the lookahead window are used to compute prospective TP, FP, FN, and TN counts, enabling evaluation under real execution dynamics without operational risk.

Model outputs provide probabilistic estimates of near-term resource demand. These probabilities are converted into planning signals using resource-specific decision thresholds^{VI} τ_r (probability cutoff (e.g., 0.5) above which a model classifies an instance as positive):

$$\hat{y}_{r,t} = \mathbb{I}(\hat{p}_{r,t} \geq \tau_r)$$

RESULTS AND DISCUSSION

The per-resource performance metrics are shown in Table 3. indicates that the RF classifiers achieve consistently high precision, recall, and F1-scores across all nine resources.

Table 3: Accuracy metrics for model validation

Resource	Precision	Recall	F1
Water	0.84	0.80	0.82
Excavator	0.94	0.94	0.94
Cleaning Bucket	0.78	0.71	0.74
Liner	0.98	0.96	0.97
Pick and Carry Crane	0.92	0.93	0.92
Koden	0.97	0.93	0.95
Rebar Cage	0.97	0.97	0.97
Tremie	0.97	0.97	0.97
Transit Mixer & Boom Placer	0.91	0.91	0.91

Water and Cleaning Bucket perform slightly lower ($F1 \approx 0.82$ and 0.74), reflecting their more variable usage patterns and the difficulty of predicting shorter, more frequent support actions, such as demand after a rest period to rectify and clean the borehole before further boring or cage lowering. In contrast, structurally critical and more procedurally constrained resources like Liner, Rebar Cage, Tremie, Koden, Pick and Carry Crane, and Transit Mixer, exhibit F1-scores in the range 0.91–0.97, indicating that the model captures their demand transitions with very few false alarms or missed activations.

The feature importance heatmap^{VII}, as shown in Figure 4, reveals a clear and interpretable hierarchy of predictors across resources. ‘Current activity ID’ seemed to be dominant for almost all resources, often contributing more than 50–60% of the total importance. This confirms that

^{VI} Decision threshold – probability cutoff above which a model classifies an instance as positive.

^{VII} Heatmap – a graphical visualization of data matrix values as colored cells, where color intensity represents magnitude. The scores indicate normalized Gini importance (%) for each feature-resource pair, aggregated across 300 trees.

the RF primarily encodes the phase structure of the pile cycle, using activity state as the main signal for when each resource is typically mobilized.

Resources Features	Water	Excavator	Cleaning Bucket	Liner	Pick and Carry Crane	Koden Machine	Rebar Cage	Tremie	Transit Mixer
Current activity ID	63.35%	53.22%	51.38%	65.38%	68.14%	68.14%	96.50%	88.40%	100.00%
Cum. Time	33.15%	33.70%	25.97%	7.73%	28.36%	35.91%	25.97%	32.60%	38.49%
Current Depth	26.34%	16.57%	25.97%	46.04%	19.52%	33.15%	26.15%	28.91%	17.31%
Depth to next strata	20.44%	14.00%	12.71%	34.81%	19.71%	15.84%	9.39%	7.37%	4.97%
Elapsed time in activity	19.34%	42.17%	32.04%	1.84%	13.81%	12.71%	8.47%	13.63%	14.00%
Next strata N-value	7.73%	4.24%	5.16%	22.84%	14.55%	5.89%	5.71%	4.60%	3.68%
Pile Length	7.18%	12.34%	14.92%	3.68%	12.15%	7.73%	6.63%	5.71%	2.58%
Delay Recency	5.71%	7.00%	13.26%	1.47%	6.81%	3.50%	2.95%	2.58%	2.95%
Cutoff Depth	0.55%	0.55%	1.47%	1.66%	0.55%	0.74%	1.29%	0.18%	0.18%
Pile Dia	0.37%	0.55%	1.29%	0.18%	0.74%	0.55%	0.92%	0.18%	0.18%

Figure 4: Feature importance heatmap

Depth-related and temporal features play differentiated but complementary roles. Current depth, depth to next strata, and next strata N-value are particularly influential for Liner, Koden, and Excavator, aligning with the dependence of these resources on subsurface conditions and penetration progress. Cumulative time and elapsed time in activity are important for resources whose demand is tied to progression within a phase (e.g., Cleaning Bucket, Water). Geometric design attributes (pile length, diameter, cutoff depth) contribute comparatively less, which indicates that once the model accounts for real-time state and geology, static design parameters add only marginal predictive value.

REAL SITE SHADOW DEPLOYMENT

The model was deployed in the site for 8 cast in-situ piles and resulted prediction scores are shown in Table 4.

Table 4: Prediction accuracy during site deployment

Resource	Precision	Recall	F1
Water	0.92	0.88	0.90
Excavator	0.92	0.94	0.93
Cleaning Bucket	0.85	0.76	0.80
Liner	0.99	0.97	0.98
Pick and Carry Crane	0.91	0.94	0.92
Koden	0.98	0.91	0.94
Rebar Cage	0.95	0.95	0.95
Tremie	0.99	0.96	0.97
Transit Mixer & Boom Placer	0.83	0.90	0.86

It is observed that the F1 score for water and the cleaning bucket on real-site trials were improved on comparison with the validation set. It predicted instances of requirement of cleaning bucket and liners when there was a delay building up, anticipating the borehole collapse. This saved a considerable amount of time in requesting the cleaning bucket and usage.

With the model, the timestamped data of the pilot study is analysed and found that the waiting waste on the account of waiting for resources has dropped to 19% from the earlier mentioned 45% and the Kelly utilization increased from the previous 63% to 79%.

CONCLUSION

This study developed and prospectively validated state-aware Random Forest classifiers for predicting 30-minute lookahead needs across nine circulatory resources in linear cast-in-situ bored piling operations. Engineered features encoding phase state, depth progression, geology anticipation, and delay recency achieved macro-F1 scores of 0.91 (offline validation) and 0.92 (shadow deployment across 8 live piles), with Current Activity ID dominating feature importance (50-60%). Real-site shadow implementation reduced resource waiting waste from 46% to 19% and boosted Kelly utilization from 63% to 79%, demonstrating that proactive coordination eliminates the primary bottleneck in equipment-constrained piling fronts.

The approach directly operationalizes lean construction's make-ready principle, converting site-available execution-state "small-data" into actionable resource signals that stabilize flow in geology-sensitive operations. Limitations include single-site geology constraints and assumption of Kelly bar availability, with variable support resources (Water, Cleaning Bucket) remaining hardest to predict despite substantial live gains. Moreover, even though the model was trained on a considerable amount of datapoints (i.e., 4339 datapoints), the number of piles (i.e., 13 piles) that yielded this data was fairly minimal to effectively represent the state space of piling operations. At the same time, the increasing deployment of digital monitoring systems on construction sites is rapidly expanding access to high-frequency, real-time piling data, which will enable more robust AI models and more reliable decision support. Future work will pursue multi-site generalization via domain adaptation, incorporate temporal dependencies (LSTM/Transformer) to capture sequence effects across pile cycles, integrate real-time geotechnical feedback for adaptive thresholds, and scale to full operational deployment with human-in-the-loop validation. These advances will enable plug-and-play resource forecasting across diverse linear construction corridors, transforming reactive coordination into systematic flow reliability.

ACKNOWLEDGEMENT

The author thanks Larsen and Toubro Heavy Civil Infrastructure-IC and CMRL ECV02 project team for enabling this study.

REFERENCES

- Bello, H., Y. Wang, and S. Lawal. (2025). A machine learning model for pile settlement prediction using majority Voting-Based feature selection. *Journal of Civil Construction and Environmental Engineering*, 10 (3): 104–114. <https://doi.org/10.11648/j.jccee.20251003.11>
- Choudhry, D., Chatterjee, K., Kumar, A., Raskar-Phule, R. (2014). Pile Foundations during Earthquakes in Liquefiable Soils – Theory to Practice. *15th Symposium on Earthquake Engineering (15SEE-2014)*. <https://doi.org/10.13140/2.1.3796.3847>
- Deepa, G., A. J. Niranjana, and A. S. Balu. (2023). A hybrid machine learning approach for early cost estimation of pile foundations. *Journal of Engineering Design and Technology*, 23 (1): 306–322. <https://doi.org/10.1108/jedt-03-2023-0097>
- Ehsan, N., E. Mirza, M. Alam, and A. Ishaqu. (2010). Notice of retraction: Risk management in construction industry. In *Proc. 3rd Int. Conf. on Computer Science and Information Technology*. New York, USA. <https://doi.org/10.1109/ICCSIT.2010.5564663>
- Fischer, A., N. Grimm, I. D. Tommelein, S. Kessler, and J. Fottner. (2021b). Variety in variability in heavy civil engineering. In *Proceedings of the 29th Annual Conference of*

- International Group for Lean Construction*, Lima, Peru. 807–816. <https://doi.org/10.24928/2021/0204>
- Fischer, A., P. Baumgartner, I. D. Tommelein, K. Nübel, and J. Fottner. (2024). Application of the work density method to in-Situ pile production in heavy civil engineering. In *Proceedings of the 32nd Annual Conference of International Group for Lean Construction*, Auckland, New Zealand. 261–272. <https://doi.org/10.24928/2024/0103>
- Koskela, Lauri. (1992). Application of the New Production Philosophy to Construction.
- Liu, Y., and H. Luo. (2025). An efficient machine learning-based model for duration prediction of construction tasks with large-scale datasets. *Advanced Engineering Informatics*, 69: 103820. <https://doi.org/10.1016/j.aei.2025.103820>.
- Mostofi, F., O. B. Tokdemir, and V. Toğan. (2024). A decision-support productive resource recommendation system for enhanced construction project management. *Advanced Engineering Informatics*, 62: 102793. <https://doi.org/10.1016/j.aei.2024.102793>
- Ohno, T. (1988). Toyota Production System: Beyond Large-Scale Production. *CRC Press*. <https://doi.org/10.4324/9780429273018>
- Porwal, V., J. L. Fernández-Solís, S. Lavy, and Z. K. Rybkowski. (2010b). Last Planner System Implementation Challenges. In *Proceedings of the 18th Annual Conference of International Group for Lean Construction*, Haifa, Israel. 548–556.
- Salman, H. A., A. Kalakech, and A. Steiti. (2024). Random Forest algorithm Overview. *Babylonian Journal of Machine Learning*: 69–79. <https://doi.org/10.58496/bjml/2024/007>
- Shelke, R., Metkari, R., Phuge, A. (2024). Case Study: Construction of Cast In Situ Pile and Pile Cap for Pier Number 370 at Line 2B Bandra Station. *International Journal of Engineering Research & Technology*. (11) 6.
- Tommelein, I.D. (1997). Discrete-Event Simulation of Lean Construction Processes. In *Proceedings of the 5th Annual Conference of International Group for Lean Construction*, Gold Coast, Australia. 121 – 136.
- Viktor.ai. (2022). Predict pile drive times with machine learning. Accessed March 24, 2026. <https://www.viktor.ai/customer-cases/38/machine-learning-predict-pile-drive-time>
- Wang, C. (2015). Analyzing Control and Management of Pile Foundation Cost. *International Conference on Structural, Mechanical and Materials Engineering (ICSMME 2015)*. Atlantis Press. <https://doi.org/10.2991/icsmme-15.2015.10>
- Wu, L., W. Ji, B. Feng, U. Hermann, and S. AbouRizk. (2021). Intelligent data-driven approach for enhancing preliminary resource planning in industrial construction. *Automation in Construction*, 130: 103846. <https://doi.org/10.1016/j.autcon.2021.103846>
- Zayed, T. M., and D. W. Halpin. 2005. Productivity and cost regression models for pile construction. *Journal of Construction Engineering and Management*, 131 (7): 779–789. [https://doi.org/10.1061/\(ASCE\)0733-9364\(2005\)131:7\(779\)](https://doi.org/10.1061/(ASCE)0733-9364(2005)131:7(779))