Machine Learning-Based Cost Optimization and Investment Decision Support for Human-Robot Collaborative Construction Operations

Mahesa Widiastuti
Department of Computer Science, University of North Texas, Denton, TX, USA.
widiastutimahesa@unt.edu

Clark Hennelley
Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO, USA.
hennelley629@missouri.edu

Candido Vilar
School of Information Technology, University of Cincinnati, Cincinnati, OH, USA.
candido888@uc.edu

Jari Hemma van Allemanie
Department of Computer Science, University of Central Florida, Orlando, FL, USA.
contactjari@ucf.edu

Abstract

Human-robot collaborative construction operations can improve productivity and safety, but their economics are difficult to evaluate because task durations, labor substitution, safety clearances, and technology prices are uncertain and interdependent. Existing cost-estimation models rarely connect probabilistic machine-learning forecasts to the discrete choice of robot configurations and the timing of partially irreversible investment. This manuscript develops a decision-support framework that links three components: a hierarchical probabilistic learner that predicts task-level cost and duration distributions from project, crew, site, and robot features; a stochastic mixed-integer cost-optimization model that selects collaborative configurations and task allocations under joint cycle-time and safety chance constraints; and a real-option investment rule that values staged adoption and deferral. The framework treats forecasts as distributions rather than point estimates, propagates predictive uncertainty into the cost objective, and separates technical feasibility from economic acceptability. Analytically, the formulation shows why point-estimate cost comparisons can reverse configuration rankings when variance and safety constraints differ, and why waiting has value when robot prices, wage rates, or task-mix conditions are volatile. The contribution is a coherent, auditable decision architecture rather than a claim of field performance. It helps construction firms, robotics vendors, and project planners align prediction, scheduling, and capital allocation, and it identifies where calibration, distribution-shift monitoring, and human oversight assumptions matter most.

Keywords

human-robot collaboration; construction cost estimation; probabilistic machine learning; stochastic optimization; real options; investment decision support

1. Introduction

Construction operations remain labor-intensive, spatially constrained, and sensitive to task variability. Robotic systems have been developed for on-site and off-site work [1], and automated construction technologies have been promoted for productivity, quality, and safety benefits [2]. Human-robot collaboration differs from full automation because it allocates tasks between people and machines according to reach, payload, precision, cycle time, and hazard exposure. Collaborative arrangements can reduce the capital intensity of automation while retaining human judgment in unstructured or safety-critical work [3]. Their economic value, however, depends on interaction effects that are not captured by unit-rate costing: a robot may shorten one task but increase setup time, require additional clearance, or force changes in crew sequencing.

Construction cost-estimation research has a long record of using regression, neural networks, and case-based reasoning to improve early estimates [4], and machine-learning methods have been applied to delay-risk and cost-performance prediction [5]. These models are usually evaluated as predictors, not as components of an investment decision. Separately, capital-budgeting theory treats irreversible investment under uncertainty through real-option reasoning [6], [7]. The two literatures connect weakly. A prediction model may produce an accurate mean cost but say little about the probability that a collaborative cell will meet a project deadline or justify its capital expense. Conversely, an investment model may assume a known productivity distribution that construction planners cannot credibly supply. Design-for-site research further indicates that robotics economics must be assessed against site logistics, interfaces, and deployment requirements rather than factory assumptions alone [8].

The knowledge gap is therefore a missing decision architecture that connects task-level probabilistic forecasts, operational constraints specific to human-robot collaboration, and staged investment timing. This manuscript addresses that gap by specifying a machine-learning-based cost-optimization and investment decision-support framework. The contribution is methodological and analytical: it defines the prediction targets, the stochastic optimization formulation, the safety and cycle-time constraints, and the real-option rule that together determine whether and when a collaborative configuration should be adopted. The framework is intended for pre-project planning and staged technology adoption, where data are heterogeneous and investment is only partially reversible. The next section defines the decision problem and its domain requirements. Section 3 develops the framework. Section 4 assesses its validity, failure modes, and implementation conditions. Section 5 concludes.

2. Decision Problem and Domain Requirements

A collaborative construction operation can be represented as a task network in which each task i requires a duration, a set of resources, and a set of safety clearances. Human-robot collaboration changes the network through task allocation: some subtasks are performed by a robot under human supervision, some by a human with robotic assistance, and some jointly. Bock and Linner [9] emphasize that construction robots are not simply factory machines transferred to sites; their economics depend on modularity, interface design, and the ability to operate under variable site conditions. This observation has a direct modeling consequence. The cost of a collaborative configuration is not a fixed unit price multiplied by output. It includes capital depreciation, installation and commissioning, operator training, maintenance, safety equipment, supervision, and the rework that arises when human-robot coordination fails.

Uncertainty enters at several levels. At the project level, scope, schedule pressure, and labor availability vary. At the task level, duration and direct cost depend on site congestion, component geometry, weather exposure, crew skill, and robot reach. At the technology level, reliability and learning effects change over time. These dependencies imply that a point-estimate cost model is insufficient for comparing configurations. Stochastic programming provides a formal way to represent recourse and scenario-dependent cost [10], while robust optimization offers a conservative alternative when probability distributions are unreliable [11]. The decision problem considered here uses probabilistic forecasts where data support them and adopts a robust safeguard when safety or schedule consequences are severe.

The operational constraints are also domain-specific. A collaborative cell must satisfy cycle-time targets without violating clearance requirements or overloading human operators. Collision risk is not a generic penalty; it depends on shared workspace geometry, task sequence, sensor coverage, and human response time. A configuration that is cheap in expectation but has a non-negligible probability of blocked access or excessive supervision may be unacceptable on a congested site. The investment decision must therefore combine technical feasibility with economic value. This separation is important because a configuration can be technically feasible yet economically inferior, or economically attractive under average conditions yet unacceptable under safety constraints.

3. Methodological Framework

3.1 Problem formulation and lifecycle cost

Let k index candidate collaborative configurations, t index project periods, and r denote the discount rate. For configuration k, the lifecycle cost is the sum of capital, operation, installation, and safety-related costs. A deterministic representation is used as the accounting basis for the stochastic formulation:

$$C_k = C_cap,k + \sum_{t=1}^{T} C_op,k,\frac{t}{(1+r)^t} + C_int,k + C_safe,k$$

Here, C_cap,k is acquisition and depreciation cost, C_op,k,t is operating cost in period t, C_int,k is integration and training cost, and C_safe,k is the cost of safety equipment, supervision, and expected rework attributable to collaboration. All costs are expressed in present-value terms. The expression is an accounting identity, not a claim about observed project costs. Its purpose is to make explicit that a lower equipment price can be offset by higher integration or safety costs.

The decision vector x contains the selected configuration and the allocation of tasks to human-only, robot-only, or collaborative modes. A scenario ξ represents uncertain task durations, cost rates, and safety outcomes. The realized lifecycle cost under decision x and scenario ξ is denoted C(x,ξ). The planner's problem is not merely to minimize this quantity for a single scenario; it is to choose a configuration that performs acceptably across plausible scenarios. This is the role of the probabilistic learner and the stochastic program.

3.2 Probabilistic machine-learning forecasting

The forecasting component predicts a distribution of task cost and duration rather than a single value. For task i with feature vector x_i, the learner outputs distribution parameters through a function h_φ. If a lognormal cost model is appropriate for positive, right-skewed task costs, the expected cost is:

$$\mathrm{E}[y_i|x_i] = \exp\left(\mu(x_i) + \sigma^2\frac{(x_i)}{2}\right)$$

where y_i is task cost, μ(x_i) is the predicted log-mean, and σ(x_i) is the predicted log-standard deviation. The same structure can be applied to duration, with a separate output head. The lognormal assumption is not universal; it is a transparent default for cost and duration variables that cannot be negative and often exhibit long upper tails. More flexible density estimators can replace it when diagnostic checks reject the lognormal form.

The model is trained by minimizing a penalized negative log-likelihood:

$$L(\varphi) = -\sum_\{i = 1\}\^\{N\} log\ f_i(y_i \mid h_\varphi(x_i)) + \lambda||\varphi||\^2$$

where f_i is the predicted density, N is the number of training tasks, λ is a regularization coefficient, and φ denotes model parameters. This objective rewards both accurate location and calibrated spread. That distinction matters for investment decisions: an overconfident model can make a risky configuration appear safer than it is, while an underconfident model can suppress adoption even when expected savings are strong.

The feature set should reflect construction-domain mechanisms rather than generic project metadata. Useful covariates include task type, component weight and geometry, required tolerance, workspace congestion, travel distance, robot reach and payload, sensor coverage, crew size and skill level, shift timing, and the degree of human-robot shared space. Hierarchical or partial-pooling structures are appropriate because tasks are nested within projects and crews, and a model trained on one site may not transfer directly to another. Tree ensembles provide a strong tabular baseline [12], while deep ensembles and approximate Bayesian neural networks provide predictive uncertainty for larger and more complex feature sets [13], [14]. The broader statistical-learning literature cautions that flexible models require out-of-sample validation and regularization to avoid spurious fit [15]. In this framework, calibration is not an optional refinement; it is a condition for using forecast distributions inside chance constraints.

3.3 Stochastic cost optimization

Given probabilistic forecasts, the configuration and allocation problem is formulated as a mean-variance stochastic program. The planner chooses x to minimize expected lifecycle cost plus a variance penalty, subject to a joint chance constraint on cycle time and safety:

$$\Pr_\xi[T(x,\xi) \le T_max, R(x,\xi) \le R_max] \ge 1 - \alpha$$

In these expressions, T(x,ξ) is the realized cycle time, T_max is the project or work-zone deadline, R(x,ξ) is a collision or clearance-risk measure, R_max is the acceptable risk threshold, λ is a risk-aversion coefficient, and α controls the allowed violation probability. The joint constraint is deliberately stricter than separate average constraints. A configuration that meets the mean cycle time but frequently violates clearance requirements is not economically equivalent to one that meets both conditions with high probability. The risk measure R(x,ξ) can be defined from minimum separation distance, required stopping time, sensor blind spots, and human response delay; its precise form must be fixed by site safety rules.

The program is a mixed-integer stochastic optimization problem because configuration choice is discrete and task allocation is combinatorial. Scenario-based sample average approximation can approximate the expectation and chance constraint when a scenario generator or historical forecast sample is available [17]. When distributional information is

weak, a robust counterpart can replace the chance constraint with a deterministic safety margin [18]. The choice between stochastic and robust formulations is a trade-off between cost efficiency and conservatism. Stochastic programming can exploit diversification across tasks and scenarios, but it is sensitive to forecast bias. Robust optimization protects against worst-case realizations, but it may reject configurations that are acceptable under realistic variability. The framework therefore recommends reporting both the stochastic solution and a robust stress case rather than treating either as uniquely correct.

Investment timing enters this optimization as an outside option. A planner may defer adoption, pilot a configuration on a limited task set, or wait for prices or performance to change. The value of deferral is not captured by a single-period cost comparison, and real-option logic provides the appropriate valuation language [16]. This is why the cost program is paired with the dynamic investment rule in the next subsection rather than solved as a static capital-budgeting problem.

3.4 Investment decision support and timing

Let S_t denote the state at decision time t, including robot price, labor wage, expected task mix, forecast quality, and technical reliability. Let π_t(S_t) be the immediate operating payoff or cost saving from adopting the configuration, and let I_t be the investment outlay. If the planner can either invest now or wait and reconsider, the value function satisfies:

$$V_t(S_t) \;=\; max\{\pi_t(S_t) \;-\; I_t, (1/(1+r))\; E[V_\{t+1\}(S_\{t+1\}) \mid S_t]\}$$

The first term is the net value of immediate adoption. The second term is the discounted continuation value of waiting. The maximizing choice yields a simple decision rule:

$$Invest\ if\ \pi_t(S_t) \;-\; I_t \;\geq\; (1/(1+r))\; E[V_\{t+1\}(S_\{t+1\}) \mid S_t]$$

with indifference at equality. The state transition can be represented as a probabilistic graphical model that encodes dependencies among wages, technology prices, forecast revision, and task demand [19]. This representation is useful because the variables are not independent: a fall in robot price may coincide with improved reliability, and a rise in wages may increase both the benefit of adoption and the cost of the human supervision that remains. The graphical structure makes those dependencies explicit and prevents the investment rule from treating them as separate certainty-equivalent inputs.

The decision rule also clarifies why a positive expected net present value is not sufficient for immediate investment. When uncertainty is high and the investment is partially irreversible, waiting preserves the option to learn. The threshold for adoption is therefore higher than the zero-net-present-value threshold. This result is not an empirical claim about a particular project; it is a structural consequence of the option value embedded in the Bellman equation. In practice, the rule should be evaluated with scenario-consistent forecasts from the probabilistic learner, so that the same uncertainty representation drives both the cost program and the investment model.

4. Analytical Assessment and Technical Discussion

The framework's internal consistency depends on three conditions. First, the forecast distribution used in the chance constraint must be calibrated for the population and site conditions to which it is applied. A model that is well calibrated on average but miscalibrated in congested work zones can produce unsafe adoption recommendations. Second, the risk measure R(x,ξ) must be defined before optimization, not after, so that the optimizer cannot trade safety against cost through an unspecified penalty. Third, the investment state must

include the information available at the decision time; using future price or wage realizations would overstate the value of waiting and distort the adoption threshold.

A central analytical result follows from the separation of mean and variance in the cost objective. Two configurations can have identical expected cost but different variance and different safety-violation probabilities. If the planner is risk averse or if the joint chance constraint binds for only one configuration, the ranking can reverse relative to a point-estimate comparison. This is not a statistical artifact; it reflects the fact that construction operations are schedule-constrained and safety-constrained. A low-cost configuration that occasionally blocks a critical path or requires unplanned supervision can be more expensive in realized project terms than a higher-cost configuration with stable cycle times. The framework makes that trade-off visible instead of hiding it inside a contingency percentage.

The machine-learning component also introduces failure modes that are specific to construction. Distribution shift is likely when a model trained on one building type, crew, or robot model is applied to another. Sparse safety events are another concern: collision and near-miss data are rare, so a model may underestimate tail risk even if it predicts average duration well. The use of hierarchical features, ensemble uncertainty, and robust stress cases mitigates but does not eliminate these risks. Forecast revision should therefore be built into the decision process. A staged pilot can generate task-level observations that update the predictive distribution before the full investment is committed. This is consistent with the option-value logic: the pilot is valuable not only for its direct output but also for the information it produces.

Sensitivity in the investment rule is driven by the discount rate, the volatility of robot prices and wages, the reversibility of the investment, and the credibility of the forecast. Higher volatility increases the value of waiting, while a shorter project horizon reduces the benefit of deferral. Higher labor wages increase the immediate payoff from adoption, but they may also increase the cost of human supervision in collaborative tasks. The net effect is configuration-specific and cannot be reduced to a single automation index. The framework therefore recommends reporting a threshold surface rather than a single hurdle rate: for each configuration, the planner can identify the combinations of wage growth, robot price decline, and productivity improvement at which immediate adoption becomes preferable to waiting.

Cost-estimation practice provides a further caution. Surveys of intelligent cost-estimation techniques show that data quality, feature selection, and validation design strongly affect predictive performance [20]. In the present framework, the cost model is not an end in itself. Its outputs are used in a constrained optimization and an investment rule, so errors in the upper tail of the forecast distribution matter more than errors near the mean. This argues for evaluating the learner with quantile or interval-based metrics in addition to mean error, and for reporting calibration by task family and site condition. The framework also requires an auditable link between forecast inputs and decision outputs. If a configuration is rejected, the planner should be able to identify whether rejection was caused by expected cost, variance, cycle-time risk, safety risk, or the value of waiting.

Implementation should proceed in stages. A first stage defines the task taxonomy, cost accounts, and safety-risk measure. A second stage trains and calibrates the probabilistic learner on available project and pilot data. A third stage solves the stochastic program for candidate configurations and evaluates the robust stress case. A fourth stage applies the real-option rule with scenario-consistent state transitions. Human oversight is required at the boundaries: safety thresholds should not be learned from cost data, and allocation decisions

that affect worker workload or trust should be reviewed by site managers. These are design conditions, not after-the-fact qualifications. They determine whether the decision support system remains a planning aid rather than an automated authority.

5. Conclusion

This manuscript has developed a machine-learning-based decision-support framework for cost optimization and investment timing in human-robot collaborative construction operations. The framework connects probabilistic task-level forecasting, a stochastic mixed-integer cost program with joint cycle-time and safety chance constraints, and a real-option investment rule. Its main analytical insight is that collaborative configurations cannot be ranked by expected cost alone. Variance, tail risk, safety feasibility, and the option to defer adoption can change the preferred choice, especially when forecasts are uncertain and investment is partially irreversible. The framework also clarifies the conditions under which machine-learning forecasts are decision-relevant: they must be calibrated, evaluated in the tail, and propagated into the optimization and investment models rather than reduced to point estimates. For construction planners, the practical implication is to treat robot adoption as a staged, information-generating decision and to separate technical feasibility from economic acceptability. For robotics vendors, the implication is that integration, supervision, and safety costs are part of the value proposition, not externalities. Future work can instantiate the framework on specific task families and test whether the predicted ranking reversals occur in practice, but the present contribution is the specification of a coherent and auditable decision architecture.

References

1. Bock, T. (2007). Construction robotics. Autonomous Robots, 22(3), 201–210.
2. Davila Delgado, J. M., Oyedele, L., Ajayi, A., Akanbi, L., Akinade, O., Bilal, M., & Owolabi, H. (2019). Robotics and automated systems in construction: Understanding industry-specific challenges for adoption. Journal of Building Engineering, 26, 100868.
3. You, S., Kim, J. H., Lee, S., Kamat, V., & Robert, L. P. (2018). Enhancing perceived safety in human–robot collaborative construction using immersive virtual environments. Automation in Construction, 96, 161–170.
4. Kim, G. H., An, S. H., & Kang, K. I. (2004). Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning. Building and Environment, 39(10), 1235–1242.
5. Gondia, A., Siam, A., El-Dakhakhni, W., & Nassar, A. H. (2020). Machine learning algorithms for construction projects delay risk prediction. Journal of Construction Engineering and Management, 146(1), 04019085.
6. Dixit, A. K., & Pindyck, R. S. (1994). Investment under uncertainty. Princeton University Press.
7. Trigeorgis, L. (1996). Real options: Managerial flexibility and strategy in resource allocation. MIT Press.
8. Luo, D., & Yu, L. (2021). From factory to site—designing for industrial robots used in on-site construction. In Automating cities: design, construction, operation and future impact (pp. 87-109). Singapore: Springer Singapore.

9. Bock, T., & Linner, T. (2016). Construction robots: Elementary technologies for offsite and onsite automation. Cambridge University Press.

10. Birge, J. R., & Louveaux, F. (2011). Introduction to stochastic programming. Springer.

11. Bertsimas, D., & Sim, M. (2004). The price of robustness. Operations Research, 52(1), 35–53.

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

13. Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in Neural Information Processing Systems (Vol. 30, pp. 6402–6413).

14. Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the 33rd International Conference on Machine Learning (pp. 1050–1059).

15. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

16. McDonald, R., & Siegel, D. (1986). The value of waiting to invest. Quarterly Journal of Economics, 101(4), 707–727.

17. Shapiro, A., Dentcheva, D., & Ruszczyński, A. (2009). Lectures on stochastic programming: Modeling and theory. SIAM.

18. Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). Robust optimization. Princeton University Press.

19. Koller, D., & Friedman, N. (2009). Probabilistic graphical models: Principles and techniques. MIT Press.

20. Elfaki, A. O., Alatawi, S., & Abushandi, E. (2014). Using intelligent techniques in construction project cost estimation: 10-year survey. Advances in Civil Engineering, 2014, 107926.