

Leveraging Historical Schedule Performance Data to Predict Construction Delays in Hyperscale Data Center Projects

Jiongwen Zhou¹, Elizabeth Onyekachi Umekwe-Edevwie², Chinyere Famakinwa³, Promise Oduh⁴, Peace Oduh⁵

¹ Independent Researcher, USA

jiongwen@amazon.com

² Independent Researcher, Nigeria

lizzyumekwe@gmail.com

³ Independent Researcher, Nigeria

chybius@yahoo.com

⁴ North Carolina State University

⁵ Independent researchers, Nigeria

DOI: 10.56201/wjimt.v9.no8.2025.pg99.126

Abstract

Historical construction schedules contain repeated evidence about how design release, procurement, field production, quality, utility coordination, and commissioning shape delivery outcomes. Their predictive value is often lost because organizations overwrite schedule updates, mix incompatible definitions, split activity rows from the same project across training and test data, or use information recorded after the forecast date. This methods paper specifies a retrospective multi-project cohort design for predicting delay in hyperscale data-center construction from time-stamped schedule-performance histories. It defines three prediction tasks: probability of missing a contractual or governance milestone by more than 14 days, expected days of delay, and a prediction interval for service readiness. The design combines schedule snapshots with project attributes, procurement events, design and submittal records, installed quantities, quality outcomes, commissioning readiness, utility evidence, and portfolio dependencies. It compares governance baselines, regularized regression, survival models, random forests, gradient boosting, temporal models, and graph-aware models. Projects, rather than activity rows, form the unit of data partitioning. Rolling-origin and leave-project-out validation preserve temporal order and estimate transfer to unseen campuses. Calibration, mean absolute error, interval coverage, precision-recall, decision value, lead time, subgroup performance, and drift form the evaluation set. Explainability uses global and local attribution, stability checks, and process review, while causal claims remain outside the predictive design. The paper contributes a leakage-resistant data model, feature dictionary, validation protocol, model-governance standard, and deployment workflow. It reports no invented performance result. The protocol is designed for prospective execution when an authorized historical project dataset is available.

Keywords

Construction delay; hyperscale data centers; schedule analytics; historical schedules; machine learning; temporal validation; explainable artificial intelligence; commissioning; model governance

1 Introduction

Hyperscale data-center programs convert capital into usable computing capacity only after tightly connected electrical, mechanical, controls, life-safety, network, and commissioning systems reach acceptance. Construction progress alone does not define success. A campus may appear nearly complete while utility energization, switchgear start-up, controls integration, defect closure, or integrated systems testing still controls service readiness. A useful predictive system must therefore estimate operational milestones, preserve uncertainty, and provide enough lead time for an accountable response. Historical schedules offer a direct record of planned sequence, actual progress, remaining-duration judgment, forecast movement, and realized outcomes. When organizations retain each update, they gain a longitudinal view of how projects evolve. This differs from a final as-built schedule, which compresses history into one retrospective record. Prediction requires the information available at a defined forecast date. Without snapshots, researchers cannot distinguish early warning from hindsight. Construction-delay studies increasingly apply machine-learning algorithms to project attributes and risk factors. Gondia et al. (2020), Yaseen et al. (2020), Egwim et al. (2021), and Santos et al. (2023) show the promise of nonlinear models and explainable analytics. Yet reported scores often depend on sampling design, label construction, and data availability. Random activity-level splitting leaks project identity. Final project variables leak outcomes. Small, homogeneous samples limit transfer. Accuracy alone conceals calibration and operational value.

Data-center construction raises additional challenges. Campuses use repeated halls and systems, but common vendors and designs create dependence. Long-lead equipment moves through engineered-to-order processes whose promised dates change. Utility work may sit outside the main contractor's detailed schedule. Commissioning generates defects, retests, queues, and witness constraints. Portfolio concurrency creates competition for specialized labor and equipment. A model trained on generic building outcomes may miss these mechanisms.

These mechanisms distinguish hyperscale delivery from the general commercial and residential stock that supplies most published delay datasets. A retail or office project rarely depends on a single switchgear lineup whose factory-test outcome determines whether an entire building can energize, and few building types stage acceptance through integrated systems testing that spans electrical, mechanical, controls, and network scopes simultaneously. A model built on general building data can therefore learn associations that do not hold once equipment concentration, staged commissioning, and utility-controlled energization dominate the outcome. This paper's protocol treats that mismatch as a design constraint: every feature family in Table 1 is chosen because it has a stated mechanism specific to hyperscale delivery, not because it appears in a general construction-delay dataset. This paper answers four questions. Which historical records should define a prediction-ready project history? How should targets and features be frozen at each forecast date? Which model and validation comparisons establish credible performance? How should predictions enter project decisions without being mistaken for causal evidence or contractual fact? The contribution is a full empirical protocol, not a claim of completed validation. The intended users are owner project-controls teams, contractors, commissioning leaders, data engineers, and researchers. The unit of value is a decision checkpoint. Examples include releasing contingency, escalating a vendor, adding controls capacity, resequencing turnover, or revising an executive commitment. A model earns deployment only when it improves a defined decision against an existing baseline at an acceptable error and alert rate.

A concrete checkpoint illustrates the intended use. Six weeks before a scheduled integrated-systems-test window, a project-controls team reviewing this protocol's output sees rising

probability of a missed test date, driven by open severe defects and falling witness-capacity evidence rather than by any change in installed quantity. The available responses, adding a second witness shift, triaging defect closure by system criticality, or accepting a later test date and notifying the commissioning authority, are each materially different in cost and lead time. The model's contribution is not to choose among these responses but to surface the driver early enough, and with enough specificity about which system is at risk, that the choice can be made deliberately rather than discovered at the test itself.

The protocol design draws on 75 peer-reviewed studies of schedule-risk modeling, earned-value and earned-schedule forecasting, explainable machine learning, and clinical prediction-model methodology (Riley et al., 2019; Steyerberg, 2019), selected for explicit treatment of validation design, leakage control, or calibration rather than for reported accuracy alone.

2 Conceptual model and hypotheses

The conceptual model treats delay as an emergent property of a changing delivery network. Baseline characteristics establish prior difficulty. Time-varying evidence then updates the risk of a milestone miss. Five pathways connect evidence to outcome: work readiness, production flow, external gates, quality and rework, and system integration. Portfolio dependencies influence all five. The model separates what teams know at sanction, what they observe during execution, and what becomes known only after completion. Each pathway maps to a specific feature family in Table 1. Work readiness covers design release, submittal cycling, and material availability, the conditions that determine whether a work package can start as planned. Production flow covers installed quantity, plan reliability, and crew stability, the conditions that determine whether started work finishes at the expected rate. External gates cover utility evidence, permitting, and weather, conditions the delivery organization can escalate but not directly control. Quality and rework cover inspection outcomes, nonconformance, and defect closure, which convert apparent physical progress into durable, acceptance-ready progress or fail to. System integration covers commissioning readiness and controls verification, the pathway specific to why a data center's completion date diverges from its construction-complete date. A milestone miss can originate in any one pathway or in an interaction between two, which is why the evaluation framework in Section 6 requires ablation by feature family rather than a single pooled importance ranking.

H1 predicts that time-varying schedule-health and performance features will improve forecasts beyond static project attributes. Static features describe scale and delivery context, while snapshots reveal current trajectory. H2 predicts that procurement-state evidence will outperform the latest vendor promised date for equipment-driven milestones. H3 predicts that commissioning-readiness and quality features will add the largest incremental value near service readiness.

H4 predicts that models validated with random activity rows will show materially better apparent performance than models tested on unseen projects, indicating leakage or project memorization. H5 predicts that hierarchical models using campus and project effects will outperform complete pooling when repeated halls share design but retain local variation. H6 predicts that probability recalibration will improve decision value when delivery conditions shift across project generations.

H1's static-versus-dynamic distinction has a direct operational reading. Two campuses can share identical power capacity, topology, and delivery model at sanction and still diverge sharply once execution begins, one holding open-end rate and near-critical mass steady while the other's schedule health deteriorates update over update. A model that uses only baseline attributes

cannot separate these campuses; only time-varying schedule, design, procurement, and commissioning evidence can. H4's leakage hypothesis follows from a known failure mode in this literature: a project reports hundreds of activities per update, and random row splitting almost guarantees that some activities from every test project already appear, under near-identical conditions, in the training set. A model exploiting this can report strong apparent accuracy while having learned nothing that would generalize to a genuinely new campus.

H6's recalibration hypothesis and H8's structured-adjustment hypothesis both address how the same model behaves as conditions change rather than how it performs at a single point in time. Delivery conditions shift across project generations in observable ways, a design that used bespoke switchgear for early campuses may standardize on a catalog product for later ones, changing the base rate of factory-test failure without changing the underlying feature definitions. H6 predicts that recalibrating the probability mapping, rather than discarding and retraining the ranking model, should restore performance at lower cost and with less instability. H8 predicts the opposite failure mode has the opposite fix: unrestricted override, where a reviewer can silently overwrite any forecast without documenting why, should reintroduce the same optimism bias the model was built to correct, while structured adjustment, which requires the reviewer to cite specific new evidence, should not.

H7 predicts that graph-aware representations will add value when the target depends on path structure and shared resources. A flat table loses precedence and shared-vendor relationships. H8 predicts that a structured human adjustment process will improve combined forecasts when reviewers add documented information absent from the model, but unrestricted overrides will reintroduce optimism bias. These hypotheses require prospective testing after retrospective development. These hypotheses are grounded in concrete delivery mechanics rather than abstract model comparison. H2, for example, follows directly from how engineered-to-order equipment moves: a transformer or switchgear lineup passes through engineering approval, material release, manufacturing, factory acceptance testing, and logistics before it reaches site, and a vendor's promised ship date can lag its actual manufacturing progress by weeks without any visible schedule variance. H3 follows from the observation that a data hall can show substantial installed quantity while commissioning prerequisites, controls-point verification, and defect closure remain the true constraint on service readiness; a model trained only on installed-quantity features will therefore systematically underestimate risk in the final quarter of delivery. H5 follows from the repeated-hall design pattern itself: pooling all halls as identical ignores local crew and access effects, while treating each hall as unrelated ignores shared design and vendor exposure, so a hierarchical model that partially pools across halls should dominate both extremes.

H7's graph-aware hypothesis reflects a specific structural feature of hyperscale portfolios: multiple campuses frequently draw on the same switchgear, generator, or chiller manufacturer, the same commissioning specialists, and sometimes the same utility interconnection queue. A flat feature table can encode a shared-vendor exposure count, but it cannot represent how a delay at one node propagates along precedence and shared-resource edges to affect milestone risk at a different, seemingly unrelated project. Testing H7 therefore requires an explicit ablation, described in Section 5, that compares a graph-aware model against the same features flattened into centrality and shared-vendor summary statistics, so that any apparent graph advantage cannot be attributed simply to having more parameters. This structural view is consistent with evidence that project schedules exhibit scaling regularities in their network structure that a flat feature table cannot represent (Vazquez, 2023).

2.1 Expanded evidence synthesis

For feature construction, the probabilistic-uncertainty literature carries a specific warning. Ökmen and Öztaş (2008) and Song and Vanhoucke (2025) both show that correlated risk sources inflate completion variance beyond what independent three-point estimates predict; Agarwal and colleagues (2017), Andrade et al. (2019), and Avlijaš and Radunović (2019) add that elicited ranges degrade quickly once a schedule is out of date. Translated into this protocol's pipeline, that means schedule-health features (open ends, constraint density, logic churn) must be computed at the same snapshot as any duration or risk feature they accompany, and correlation structure among vendor, design, and labor risks should be an explicit model input rather than an assumption resolved by data-cleaning.

The critical-path-sensitivity stream informs which activities deserve dedicated features rather than aggregate ones. Nasir, McCabe, and Hartono (2003) and Shu-Shun and Shih (2009) identify resource chains that dominate completion risk; Hendradewa (2019) and Mabeba et al. (2018) confirm the pattern on separate project types. None of these studies, however, addresses a network with several simultaneously near-critical trains, which is the normal condition once power, cooling, and controls compete for criticality on a data hall. This protocol therefore computes near-critical mass and critical-path turnover (Table 1) at every snapshot rather than tracking only the single reported critical path, consistent with Virle and Mhaske's (2013) finding that earned-value signals lag when tracked against one path alone.

Monte Carlo studies establish the target this protocol's probabilistic baselines must beat. Luu et al. (2009) and Okmen (2013) simulate correlated or repetitive-unit networks; Moselhi and Roofigari-Esfahan (2013) targets compression rather than forecasting; Fitzsimmons et al. (2022) hybridizes simulation with learned models to sharpen contingency, the closest prior design to the ensemble strategy proposed in Section 5. Dubrovskaya et al. (2021) is included for its treatment of network-construction software rather than risk method. Because none of these studies reports project-level holdout performance, this protocol treats a disciplined Monte Carlo forecast, not a black-box learner, as the benchmark that any machine-learning candidate must outperform before adoption. Resource-constraint research shapes how procurement and labor features should be scaled. Carr and Tah (2001) and Curtis (1993) frame risk in qualitative or fuzzy terms suited to early design, while Sami Ur Rehman et al. (2020) ties resource risk to building-information-model data and Andreas, Tinumbia, and Anggraini (2023) compares earned-value, earned-schedule, and earned-duration indices for resource-loaded control. Carpenter (2025) reports integrated cost-and-schedule risk analysis in capital-intensive petroleum construction, whose long equipment lead times resemble data-center procurement. The shared implication is that resource and procurement features should be expressed as rates and exposure measures (Table 1), not raw counts, so that a campus with twice the equipment does not appear twice as risky by construction. Delay-causation surveys are useful for feature selection but not for feature weighting, because their rankings are context-bound. Rahman, Lee, and Ha (2014) and Tabish, Pandian, and Jha (2011) rank causes for Bangladeshi and Indian construction; Karimi, Taylor, and Goodrum (2017) isolates craft-labor availability in North America; Sullivan et al. (2017) compares delivery methods rather than causes; Xu et al. (2018) proposes a hybrid dynamic model instead of a ranked list. This protocol treats such studies as a source of candidate feature families, not as a fixed prior on feature importance, and requires project-level holdout validation (Section 6) precisely because cause rankings from other sectors and geographies should not be assumed to transfer to hyperscale delivery without local re-estimation.

Earned-schedule research supports this protocol's choice of system-weighted, not only aggregate, progress features. Choudhry et al. (2014) and Mahamid (2017) test time-based forecasting on bridges and roads; Takahashi and Hiraki (2006) on dam construction; Chaterine and Simanjuntak (2020) on interior fit-out. Ballesteros-Pérez et al. (2019) and Batselier and Vanhoucke (2015) compare forecasting accuracy and sensitivity metrics across project types and finds that some indicators remain informative regardless of sector while others do not, a result this protocol operationalizes by requiring every candidate earned-schedule feature to be benchmarked against the current forecast before being retained. Civil-heavy aggregate progress is an especially poor proxy for a data center, where commercial value depends on power, cooling, and controls acceptance rather than gross quantity installed.

Bayesian and fuzzy risk models justify this protocol's inclusion of hierarchical and mixed-effects candidates alongside tree-based learners. Salanke and El-Gafy (2013) and Koulinas et al. (2020) build Bayesian belief networks and simulation-based expert systems for delay risk; Yang (2021) and Kiruthiga Lakshmi Narayanaswamy with Sathyabama University (2016) survey computer-assisted scheduling more broadly; Ma and Wu (2019) links a big-data failure-mode framework to schedule risk in apartment construction. These designs represent dependence explicitly but rarely validate on unseen projects. This protocol keeps their structural insight, dependence between design, procurement, and commissioning risk, while requiring the calibration and holdout evidence (Section 6) that the cited studies do not themselves report.

Building-information-model integration research mainly demonstrates feasibility, which limits its use for feature-timing decisions. Tahrir, Ramadhani, and Baco (2023) and Bansal and Pal (2009) show model-linked schedule visualization for residential and general construction; Hossen, Kang, and Kim (2015) applies a combined AHP-RII method to a nuclear power-plant program, whose coordination density is the closest analogue in this literature to a data center; Kanade (2026) and Franck, Neumann, and Schwindt (2001) address digital-twin-linked probabilistic analysis and resource-constrained scheduling algorithms respectively. This protocol borrows their emphasis on stable object identifiers (Table 1, design and schedule families) while requiring that any model-derived feature be timestamped to its source system, not to the visualization date. Digital-twin and monitoring studies inform how this protocol timestamps live-evidence features rather than whether such features exist. Zhao, Meng, and Pang (2020) and Arvikar et al. (2025) demonstrate live-state capture on EPC and augmented-reality-linked programs without reporting forecast accuracy against held-out outcomes; Sun, Man, and Wang (2015) and Sun, Zhou, and Wang (2013) connect model data to cost-and-schedule early warning and power-generation analysis; Bunni (1993) supplies the risk vocabulary the newer studies still use. Because none of these establishes a leakage-free evaluation, this protocol requires every twin-derived feature to carry a source timestamp and synchronization-latency measure, so that apparent predictive power cannot originate from information not yet available at the forecast date. Shittu, Shittu, Adeleke, and Adedokun (2023) apply the same digital-twin monitoring logic to real-time fault detection in smart electrical substations, a nearby domain that timestamps live-state evidence under comparable synchronization constraints and supports treating utility-side telemetry as a legitimate snapshot feature rather than an aspirational one.

Cost-and-schedule interaction studies motivate treating procurement and schedule risk as one feature family rather than two. Bogucki and Polonski (2019) study gas-pipeline construction; Isah and Kim (2021) integrate schedule risk with multi-skilled resource scheduling; Yang and Lai (2023) compare earned-value, earned-schedule, and earned-duration measures; Liberatore (2002) applies fuzzy logic to schedule uncertainty; Siregar, Jayady, and Pribadi (2026) evaluate

procurement-risk mitigation across cost, quality, and schedule outcomes in industrial construction, the closest analogue to equipment-driven data-center risk. Accordingly, Table 1 places procurement promise-volatility and shared-vendor exposure beside schedule forecast-movement features rather than in a separate cost table.

Forecast-validation studies expose the specific design flaw this protocol is built to avoid. Yin's three 2023 papers apply a similar building-information-model-based method across different infrastructure classes without a consistent held-out test, illustrating how a method can appear repeatedly confirmed while being re-demonstrated on similar data rather than genuinely validated. Kog et al. (1999) and Paul and Basu (2016) instead identify determinants of schedule performance and combine scenario planning with failure-mode analysis, offering candidate predictors rather than tested forecasts. This contrast is the direct justification for Section 6's requirement that every candidate model compete against current-forecast and historical-median baselines under project-level, not row-level, holdouts.

Governance and decision-threshold studies converge on a point central to this protocol's deployment design: identifying a risk driver is not the same as acting on it. Mahamid (2013) ranks delay contributors from an owner's perspective; Dziadosz and Rejment (2015) and Dai (2014) compare risk-analysis methods on large-scale projects; Hegazy, Said, and Kassab (2011) incorporates rework explicitly into schedule analysis, linking quality failure to duration growth in a way this protocol mirrors through its nonconformance and rework-hours features; Nadia Diandra (2025) tracks schedule deviation through S-curve analysis on a residential project. None specifies a probability threshold, response owner, or escalation path, which is why Section 7 requires each deployed alert to carry all three before it can trigger action.

3 Study design

The proposed design is a retrospective multi-project cohort with prospective silent deployment. Eligible projects are hyperscale or large colocation data-center builds with at least one governed baseline, monthly or more frequent schedule snapshots, traceable milestone definitions, and verified actual dates. Projects lacking service-ready outcomes may contribute to interim milestones but should not enter the final-service target. Inclusion and exclusion rules must be approved before outcome analysis. The primary observation is a project-snapshot pair. One project contributes multiple observations at scheduled forecast dates, such as month-end or thirteen-, twenty-six-, and fifty-two-week horizons before the target. Observations remain clustered within projects. Statistical analysis and resampling must preserve this structure. Activity-level records supply features but are not independent samples. Eligibility should distinguish facility types that share the label hyperscale but differ in delivery mechanics. A greenfield campus built to a single hyperscaler's specification differs from a multi-tenant colocation facility built to accommodate several future tenants' unknown equipment, and a retrofit or capacity-expansion project inside an operating campus differs from both in that live-facility constraints, limited outage windows, and phased energization govern its schedule in ways a greenfield build does not. This protocol admits all three where governed baselines and snapshot history exist, but requires facility type to enter as a stratification variable rather than being pooled silently, since a model trained predominantly on greenfield campuses should not be assumed, without evidence, to transfer to a live-facility retrofit. The primary binary outcome equals one when actual milestone completion exceeds the contemporaneous committed date by more than fourteen calendar days. Secondary thresholds of seven, thirty, and sixty days support sensitivity analysis. The regression outcome is signed calendar-day variance. A time-to-event

outcome supports survival analysis when projects remain incomplete. The service-ready milestone requires written acceptance criteria covering power, cooling, controls, life safety, network prerequisites, commissioning, and turnover evidence.

The index date is the timestamp at which a forecast is issued. Every feature must be reproducible using records created on or before that timestamp. Later corrections stay unavailable to the historical simulation unless they were known then. The data pipeline should implement an as-of join, retain source-system transaction times, and reject variables lacking time lineage. This rule is the primary defense against target leakage. Milestone definitions should be fixed before any snapshot is labeled, because hyperscale programs use overlapping terms loosely in daily practice. Utility power available, substantial completion, mechanical completion, and service readiness are not interchangeable: a campus can reach mechanical completion while integrated systems testing remains open, and utility power available can precede or follow controls integration depending on delivery sequence. This protocol requires each project to map its internal terminology onto the acceptance-criteria structure defined in Section 3 before any label is assigned, and requires that mapping to be version-controlled, since owners occasionally redefine milestone criteria mid-program in response to commercial pressure. A model trained across mappings that silently changed mid-series would confuse definitional drift with genuine schedule improvement. A data-use agreement should define ownership, access, security, retention, publication, and de-identification. Project and vendor names should receive stable pseudonyms. Personally identifying workforce fields should be excluded unless a specific approved need exists. Analysis should focus on work systems and resource conditions rather than individual worker scoring. An illustrative eligible cohort clarifies the scale this design assumes. A owner with twelve hyperscale campuses delivered over six years, each containing four to eight data halls, would yield on the order of sixty to ninety project-level records but several thousand project-snapshot observations once monthly schedule updates, procurement events, and commissioning milestones are counted individually. This asymmetry, few independent projects but many correlated observations per project, is the reason Section 6 mandates project-level rather than row-level partitioning: a model that appears to explain snapshot-to-snapshot variation may simply be recognizing which of the sixty to ninety projects it has already seen.

The retrospective phase develops and validates candidate models. The silent prospective phase runs the chosen model without influencing decisions for at least two reporting cycles or an agreed number of milestone observations. Teams compare live forecasts with later outcomes, data availability, latency, and existing governance. Decision use begins only after the model passes predefined acceptance criteria.

4 Data engineering and feature construction

Source integration begins with a canonical hierarchy: portfolio, project, campus, building, hall, system, subsystem, control account, work package, activity, asset, document, issue, test, and milestone. Crosswalks need version dates because codes change. Raw source tables should remain immutable. Curated analytical tables should record transformation code, units, missingness, and quality checks.

Schedule snapshots should preserve activity identifiers, names, hierarchy, calendars, logic, lags, constraints, baseline dates, current dates, actuals, remaining durations, float, percent complete, resources, and status date. Derived measures include forecast movement, baseline change, criticality, near-critical mass, logic churn, out-of-sequence progress, open ends, constraint

density, and remaining-work concentration. Project Management Institute (2019) and Hulett (2009) provide foundations for schedule-quality interpretation.

Feature engineering must respect both horizon and scale. A total-project value may hide a failing system. The pipeline should calculate features at project, milestone network, system, and critical-work-package levels. Rolling windows of one, three, and six reporting periods distinguish current state from trend. Rates should use exposure denominators, such as defects per accepted test or RFIs per active work package. Missingness is information but needs careful treatment. An absent factory-test date may indicate an immature procurement package, a late data feed, or a source-system failure. The pipeline should retain missing indicators, distinguish structural from unexpected missingness, and measure source latency. Imputation must occur inside each training fold to prevent information transfer from the test set.

Text data require entity linkage and timestamp discipline. Meeting minutes, RFIs, issue narratives, and commissioning comments can yield topic, age, recurrence, dependency, and escalation features. Language models may extract structured fields, but sampled records need human validation. The raw text should remain protected, while derived features should contain the minimum detail required for prediction.

Source-system heterogeneity is the rule rather than the exception in practice. A single campus commonly runs a scheduling tool for the integrated master schedule, a separate commissioning-management platform for scripts and defects, a controls-integration platform for points verification, a document-control system for submittals and RFIs, and one or more vendor portals for procurement status, often with no shared project identifier across them. Building the canonical hierarchy in this section therefore requires an entity-resolution layer, mapping vendor equipment tags, drawing numbers, and commissioning script identifiers to a single work-package key, before any feature dictionary entry in Table 1 can be populated reliably. Teams that skip this step typically discover it only when two data-center halls report contradictory prerequisite completeness because their commissioning defects were logged under different naming conventions. Graph construction represents activity precedence, system interfaces, shared vendors, and shared resources. Nodes hold time-varying attributes. Edges need typed meanings and validity dates. Graph features can include path distance to milestone, centrality, number of unresolved predecessors, shared-vendor exposure, and propagation potential. Advanced graph neural networks should compete with simple engineered graph measures before adoption. Table 1 lists the complete feature dictionary that results from this construction process, organized by domain, source, timing, and predictive rationale. Edge validity dates matter as much as edge existence. A precedence link between two activities, or a shared-vendor edge between two work packages, can be added, removed, or redefined as the schedule and procurement plan evolve, and a graph feature computed without respecting when an edge was actually active would leak later network structure into an earlier forecast date. The pipeline should therefore version the graph itself, not only the node-level features drawn from it, so that any graph-derived predictor can be reconstructed exactly as it existed at each historical index date.

Table 1. Feature dictionary for reproducible model development

Domain	Feature	Definition	Source	Timing	Predictive rationale
Baseline	Power capacity	MW at approved scope	Project charter	Sanction	Reference-class scale and system complexity

Domain	Feature	Definition	Source	Timing	Predictive rationale
Baseline	Facility topology	Buildings halls power trains cooling trains	Design basis	Sanction	Separates repeated structure and redundancy
Baseline	Delivery model	Design-bid-build design-build EPCM or hybrid	Contract records	Sanction	Captures governance and interface allocation
Baseline	Design maturity	Percent packages approved at baseline	Document control	Baseline	Measures readiness behind planned start
Baseline	Utility scope	Owner utility or shared responsibility	Interconnection plan	Sanction	Separates external gate exposure
Schedule	Logic density	Relationships per active activity	Schedule snapshot	Each update	Flags underlinked networks
Schedule	Open-end rate	Open-ended activities per 100 activities	Schedule snapshot	Each update	Signals incomplete dependency logic
Schedule	Constraint density	Hard constraints per 100 activities	Schedule snapshot	Each update	Detects date forcing
Schedule	Near-critical mass	Remaining activities under float threshold	Schedule snapshot	Each update	Represents path convergence
Schedule	Forecast movement	Days moved since prior snapshot	Schedule archive	Each update	Measures deterioration or recovery
Schedule	Logic churn	Added or deleted links per period	Schedule archive	Each update	Captures plan instability
Schedule	Remaining-duration growth	Net increase among in-progress tasks	Schedule archive	Each update	Detects optimistic status correction
Schedule	Out-of-sequence	Activities	Schedule	Each	Measures

Domain	Feature	Definition	Source	Timing	Predictive rationale
	rate	progressing against logic	snapshot	update	execution-plan mismatch
Schedule	Critical path turnover	Change in controlling path family	Schedule archive	Each update	Signals instability
Design	Release reliability	Packages released on or before plan	Document control	Each update	Measures downstream readiness
Design	Revision intensity	Revisions per active package	Document control	Each update	Captures design volatility
Design	RFI aging	Median and tail age of open RFIs	RFI system	Weekly	Represents unresolved field information
Design	Submittal cycle time	Days from submission to approved state	Submittal log	Weekly	Measures approval queue
Design	First-pass approval	Share approved without resubmission	Submittal log	Weekly	Measures package quality
Design	Interface issue age	Age of cross-system unresolved issues	Issue log	Weekly	Signals integration risk
Procurement	PO release variance	Actual versus planned award date	Procurement system	Event	Measures start of lead-time chain
Procurement	Engineering approval	Evidence state for vendor drawings	Procurement documents	Weekly	Distinguishes promise from readiness
Procurement	Manufacturing stage	Verified stage percentage or state	Vendor report	Weekly	Tracks physical progress
Procurement	Factory-test readiness	Prerequisite completeness	Quality system	Weekly	Predicts test date credibility
Procurement	Factory-test	Pass fail and	Quality system	Event	Updates

Domain	Feature	Definition	Source	Timing	Predictive rationale
	outcome	open findings			shipment risk
Procurement	Logistics status	Booked departed customs and delivered	Logistics system	Event	Tracks remaining transitions
Procurement	Promise volatility	Count and magnitude of vendor date changes	Procurement archive	Each update	Measures commitment instability
Procurement	Shared-vendor exposure	Concurrent packages using supplier	Portfolio data	Monthly	Captures common capacity risk
Production	Installed quantity rate	Accepted quantity per crew-hour	Field system	Weekly	Measures realized productivity
Production	Plan reliability	Completed commitments divided by planned	Look-ahead plan	Weekly	Measures workflow control
Production	Constraint removal	Constraints cleared before work date	Constraint log	Weekly	Measures readiness discipline
Production	Crew stability	Turnover and reassignment by work package	Labor system	Weekly	Captures learning disruption
Production	Congestion	Active crews or permits per zone	Field records	Daily	Represents workspace interference
Production	Material readiness	Work packages with full kit available	Material system	Weekly	Separates labor from supply constraint
Quality	Inspection pass rate	Accepted inspections divided by attempts	Quality system	Weekly	Measures hidden completion
Quality	Nonconformance rate	Open NCRs per active work	Quality system	Weekly	Predicts rework demand

Domain	Feature	Definition	Source	Timing	Predictive rationale
		package			
Quality	Rework hours	Hours charged to corrective work	Labor system	Weekly	Measures productivity loss
Quality	Closure age	Median and 90th percentile defect age	Quality system	Weekly	Captures unresolved backlog
Quality	Repeat defect rate	Recurring defect families per system	Quality system	Weekly	Signals systemic failure
Commissioning	Prerequisite completeness	Required documents and checks complete	Cx platform	Weekly	Measures test readiness
Commissioning	Script approval	Approved scripts divided by due scripts	Cx platform	Weekly	Captures preparation maturity
Commissioning	Controls point verification	Verified points divided by planned	Controls platform	Weekly	Measures integration readiness
Commissioning	First-test pass rate	Tests passed on first execution	Cx platform	Weekly	Predicts retest demand
Commissioning	Severe defects open	High-severity defects by system	Cx platform	Daily	Directly threatens acceptance
Commissioning	Retest time cycle	Days from failure to passed retest	Cx platform	Weekly	Measures defect recovery
Commissioning	Witness capacity	Available witness hours versus demand	Resource plan	Weekly	Captures test queue constraint
Commissioning	Turnover completeness	Accepted records divided by required	Document system	Weekly	Links physical work to acceptance
External	Utility evidence	Completed	Utility tracker	Weekly	Measures

Domain	Feature	Definition	Source	Timing	Predictive rationale
	state	prerequisites for energization			external milestone maturity
External	Permit comment age	Age of unresolved authority comments	Permit tracker	Weekly	Predicts approval slippage
External	Weather exposure	Observed and forecast lost-work conditions	Weather source	Daily	Represents exogenous productivity risk
External	Inspection availability	Booked authority slots versus need	Inspection plan	Weekly	Captures external queue
Portfolio	Shared specialist load	Demand versus capacity by specialist role	Portfolio resource plan	Weekly	Measures cross-project competition
Portfolio	Shared equipment exposure	Projects dependent on common equipment family	Portfolio procurement	Monthly	Captures common-mode supply risk
Portfolio	Utility concentration	Capacity milestones under same provider	Portfolio utility tracker	Monthly	Captures correlated external risk
Outcome	Committed milestone date	Date active at index time	Governance register	Each index	Defines contemporaneous commitment
Outcome	Actual milestone date	Verified accepted completion	Acceptance record	Outcome	Defines realized event
Outcome	Service readiness	Operational acceptance state	Owner acceptance	Outcome	Primary business-relevant completion
Governance	Data freshness	Age of newest source record	Pipeline metadata	Each run	Flags stale predictions
Governance	Override	Human	Forecast	Each	Supports

Domain	Feature	Definition	Source	Timing	Predictive rationale
	magnitude	adjustment to model forecast	ledger	decision	human-model evaluation
Governance	Intervention completion	Action executed after alert	Action tracker	Each alert	Measures response pathway

5 Modeling strategy

The model comparison begins with governance baselines: the current schedule forecast, no-change from prior forecast, historical median variance for a reference class, and earned-schedule projection. These baselines establish whether machine learning adds value. A complex model that fails to outperform the current forecast at the same horizon should not enter production.

Statistical baselines include regularized logistic and linear regression. Their coefficients, shrinkage, and probability outputs support interpretation and calibration. Survival models estimate time to milestone while retaining incomplete projects. Mixed-effects models account for repeated snapshots and campus clustering. Generalized additive models allow nonlinear effects while retaining graphical interpretability. General guidance on building and tuning such baselines without overfitting follows standard applied predictive-modeling practice (Kuhn and Johnson, 2013). Interpretability requirements are not uniform across use cases and should not be treated as a single design choice. A forecast that informs internal prioritization can tolerate a less transparent model provided its calibration is monitored, but a forecast that influences vendor escalation, contractual notice, or executive commitment communication needs a model whose drivers can be explained to a party outside the data-science team without appeal to feature-importance scores alone. Regularized regression and mixed-effects models satisfy the second requirement directly through their coefficients; tree-based and temporal candidates satisfy it only through post hoc explanation methods that Section 7 requires to be validated against process-expert judgment before their output is treated as a driver rather than a correlate.

Tree-based candidates include random forest and gradient boosting. They handle nonlinear interactions, mixed scales, and missingness patterns common in project data. Hyperparameters must be tuned within nested training folds. Class weighting or focal objectives should be evaluated for rare delay thresholds, but probability calibration must be checked after any imbalance treatment. Repeated-hall structure has a specific consequence for hyperparameter tuning that generic tabular-modeling guidance does not address. A gradient-boosted model tuned by ordinary cross-validation can inadvertently reward splits that key on hall identity within a single campus, since repeated halls sharing one design and one vendor set produce highly similar feature vectors with correlated outcomes. Nested tuning folds in this protocol must therefore be constructed at the project level, not the hall or snapshot level, so that a hyperparameter setting cannot be selected because it fits the idiosyncrasies of halls the final test set will later reuse under a different label. Temporal candidates use sequences of snapshot features. Recurrent networks, temporal convolution, or transformer-based models might capture acceleration and deterioration patterns. Their data demand is high. Sequence padding, irregular intervals, and changing project scope require explicit treatment. Simpler lagged features may perform similarly with lower governance cost. Graph-aware models use schedule and dependency structure. Message-passing models can represent propagation along precedence links or shared resources. Their advantage should be tested through ablation against flat models with critical-path and centrality features.

Graph changes over time create versioning complexity, and explanations must identify which nodes and edges influence the forecast. An ensemble may combine models with distinct error patterns. Stacking weights should be learned only from validation predictions, not the test set. The final system should favor parsimony when performance differences are small. Model selection should consider calibration, lead time, stability, inference latency, maintainability, and actionability alongside discrimination. Table 2 summarizes the candidate models compared under this strategy and the controls each requires. A worked comparison illustrates how the candidate models would be expected to diverge. Consider a campus where a generator vendor's engineering-approval evidence stalls two weeks behind its promised date while the vendor's stated ship date remains unchanged. The current schedule forecast, anchored to the vendor's promise, would show no movement. The historical reference class would flag elevated risk only once enough comparable campuses existed to estimate a base rate for this equipment family. A regularized regression using procurement evidence features (Table 1) should detect the engineering-approval lag immediately, since it is an explicit predictor, while a gradient-boosted model with access to the full feature set should additionally weigh the lag against concurrent schedule-health and near-critical-mass signals to judge whether downstream float can absorb the delay. This scenario is the type of case the ablation and ensemble analyses in this section are designed to surface and score, not a claim about which model performs best in general.

Table 2. Candidate model comparison

Model	Purpose	Key controls
Current schedule forecast	Operational benchmark	Same index date and milestone definition
Historical reference class	Outside-view benchmark	Class frozen before testing
Regularized regression	Interpretable nonlinear baseline through transforms	Nested tuning and calibrated probabilities
Mixed-effects model	Repeated snapshots and project clustering	Random effects estimated without test leakage
Survival model	Incomplete projects and time to milestone	Time-varying covariates frozen at index
Random forest	Nonlinear tabular benchmark	Project-level folds and calibration
Gradient boosting	High-performance tabular candidate	Early stopping inside training folds
Temporal sequence model	Trajectory learning	Rolling-origin validation
Graph-aware model	Precedence and shared-resource propagation	Time-versioned nodes and edges
Stacked ensemble	Combine distinct errors	Weights learned from validation predictions only

6 Validation and statistical analysis

The primary split holds out entire later projects. Training uses earlier projects, validation supports tuning and threshold selection, and a locked test set estimates final performance. A second leave-one-project-out analysis measures sensitivity to individual campuses. A geographic or delivery-partner holdout should test transfer where sample size permits. Random row splits may appear only as a demonstration of optimistic bias.

Rolling-origin evaluation recreates the live forecasting process. At each origin, the pipeline trains on information previously available and predicts the next eligible projects or snapshots. Feature definitions, imputers, encoders, selection, and calibration must be fitted inside the origin. This prevents subtle leakage from future distributions.

Regression evaluation includes mean absolute error, median absolute error, root mean squared error, signed bias, and error quantiles. Interval forecasts require empirical coverage, width, and conditional coverage across horizons. Classification evaluation includes precision-recall area, receiver-operating area, sensitivity, specificity, positive predictive value, Brier score, log loss, and calibration slope and intercept. Interval width should be interpreted against the decision it supports, not treated as a single universal target. A fourteen-day prediction interval around a design-freeze date eighteen months from service readiness may be entirely adequate for capital planning, while the same fourteen-day interval around an integrated-systems-testing date four weeks out would leave a response team with almost no usable lead time. This protocol therefore requires interval coverage and width to be reported by horizon, consistent with Table 3, so that a narrowing interval as service readiness approaches can be distinguished from a model that is simply miscalibrated at every horizon alike. Performance should be reported by forecast horizon, project phase, facility type, region, delivery model, and target prevalence. Bootstrap confidence intervals must resample projects rather than rows. Pairwise model differences should use the same held-out predictions. Statistical significance does not replace operational relevance; a predeclared minimum improvement should govern selection. Decision evaluation assigns costs to missed alerts, false alerts, and interventions. Net-benefit curves or an explicit cost matrix compare thresholds. Lead time measures days between the first sustained alert and the milestone. Alert burden measures alerts per project-month. A useful model should improve response opportunity without flooding teams. Cost assignment should be elicited from the same project-controls teams who would act on the alerts, not set analytically by the modeling team alone. The cost of a false alert on an energization-risk indicator, an unnecessary vendor executive call, a wasted contingency-planning cycle, is small relative to the cost of a missed alert that allows a campus to reach a scheduled integrated-systems-test window without adequate witness capacity or defect closure. Where such elicitation is not feasible before initial deployment, this protocol requires the evaluation to report net-benefit curves across a plausible range of cost ratios rather than committing to a single assumed ratio, so that a project-controls team can locate its own judgment on the curve after the fact. Ablation analysis removes each feature family to estimate incremental predictive contribution. Stability analysis repeats feature attribution across folds, horizons, and seeds. Error analysis reviews false negatives, false positives, large residuals, and unsupported cases. Reviewers should classify whether errors arise from missing data, definition failure, unprecedented events, model weakness, or human override. Table 3 summarizes the evaluation framework, mapping each dimension to an acceptance question a project-controls team can apply directly. Subgroup reporting should reflect the specific ways hyperscale delivery varies rather than generic demographic categories. Relevant strata include redundancy topology (N, N+1, 2N), cooling technology (air-cooled, direct liquid, hybrid), delivery model (design-build, design-bid-build, EPCM), utility ownership of the interconnection, and project generation (first-of-a-design versus repeated hall of an established design). A model that performs well on average but poorly for first-of-a-design halls or for projects where the owner rather than a utility controls interconnection would misinform the teams facing exactly the conditions where forecasting is hardest and most valuable.

Table 3. Evaluation framework

Dimension	Measures	Acceptance question
Accuracy	MAE median AE RMSE signed bias	Does the model reduce date error against baselines?
Uncertainty	Interval coverage width and conditional coverage	Do stated ranges contain outcomes at the promised rate?
Discrimination	Precision-recall ROC sensitivity specificity PPV	Does the model separate milestone misses at useful thresholds?
Calibration	Brier score slope intercept reliability plot	Do reported probabilities match observed frequencies?
Decision value	Net benefit intervention cost avoided loss	Does use improve the decision under stated costs?
Timeliness	First alert lead time and sustained alert lead time	Does warning arrive before response options close?
Stability	Fold horizon seed and attribution stability	Does the signal persist across reasonable choices?
Equity and transfer	Performance by region partner type and project generation	Where does the model fail to transfer?
Operations	Latency missingness alert burden override rate	Does the service work inside reporting cadence?

7 Explainability governance and deployment

Global explanation uses coefficient paths, permutation importance, grouped feature importance, partial dependence, and SHAP summaries as appropriate. Correlated features require grouped interpretation because individual rankings become unstable. Local explanations show the evidence behind a specific alert. Neither form proves causation. Process experts should assess whether the pattern is plausible and actionable. Model-agnostic explanation methods and their proper use follow established treatments of explanatory model analysis and interpretable machine learning (Biecek and Burzykowski, 2021; Molnar, 2022).

A model card should document target, horizon, intended use, excluded use, training period, project population, algorithms, validation, calibration, subgroup results, limitations, and monitoring. A data sheet should document sources, definitions, timestamps, lineage, missingness, and access. Each prediction should link model version, feature snapshot, output, explanation, threshold, user response, and outcome.

Deployment should start in silent mode, then advisory mode. In advisory mode, the model supplies probability, interval, driver groups, data freshness, and supported actions. It does not alter the contractual schedule automatically. Users who override the output should record new evidence, direction, magnitude, and rationale. Later analysis compares model-only, human-only, and combined forecasts. A concrete override case illustrates why structured adjustment, not silent human correction, is required. Suppose the model raises the probability of an energization miss because switchgear factory-test evidence has stalled, while a project executive knows, from a call the vendor has not yet logged in any system, that the test has actually passed and documentation is simply pending. Recording only a revised forecast would erase the evidence the model correctly used; recording the original forecast, the adjusted forecast, the stated reason, and the date the vendor's system evidence caught up allows later analysis to confirm whether the

executive's early information was reliable in general or accurate only in this instance. Over enough cases, this record is what converts anecdote about human judgment into an evaluable claim. Monitoring covers data completeness, latency, feature drift, probability calibration, error, subgroup performance, alert rate, override patterns, and intervention outcomes. A stop rule should suppress predictions when required sources fail, the case lies outside training support, or calibration breaches a threshold. Recalibration may precede full retraining when ranking remains stable. Security and privacy controls should apply least privilege, encrypted transfer and storage, approved retention, and auditable access. Commercially sensitive vendor performance should have defined uses. Workforce data should remain aggregated when individual identity is unnecessary. Governance should prevent the system from becoming a punitive score that encourages data manipulation. Vendor-performance sensitivity deserves specific mention because it differs from typical construction-analytics privacy concerns, which focus mainly on workforce data. A manufacturing-stage or factory-test-outcome feature effectively scores a specific equipment supplier's delivery reliability. Sharing that score outside the immediate project-controls function, particularly with parties negotiating future contracts with the same vendor, raises commercial and legal considerations distinct from workforce privacy. Governance should therefore define, separately from workforce-data rules, which roles may see vendor-identified procurement features and which may see only de-identified, aggregated versions.

8 Expected contribution limitations and research agenda

The protocol advances the field by treating prediction as a longitudinal, leakage-controlled decision service. Its contribution lies in the design, not unobserved results. Once executed, the study will show whether historical schedule evidence transfers across hyperscale projects, which feature families add value at each horizon, and whether calibrated alerts improve decisions beyond current controls. That framing has a practical consequence for how the protocol should be read by a reviewer or an adopting organization. It does not promise that any specific algorithm, gradient boosting, a graph neural network, or a survival model, will prove superior; it promises that whichever algorithm is chosen will have been evaluated under conditions that make the result trustworthy, on unseen projects, against honest baselines, with calibration and lead time reported alongside accuracy. A protocol that guaranteed a particular model's superiority in advance of execution would itself be an example of the presenting-conjecture-as-measured-performance problem this paper is designed to prevent.

The main limitation is data availability. Hyperscale project histories are commercially sensitive and may use inconsistent coding. Small project counts constrain deep learning even when activity rows number in the millions. Organizational changes create concept drift. Actual service dates may be influenced by commercial decisions beyond construction. These limits require careful target definition and restrained generalization. A related limitation concerns the boundary between construction and commissioning records, which is drawn differently across organizations. Some owners treat commissioning as a subcontracted scope with its own schedule and data systems, entirely separate from the general contractor's integrated master schedule, while others embed commissioning activities directly inside the same schedule file used for construction sequencing. The feature dictionary in Table 1 assumes commissioning evidence is available at the same snapshot cadence as construction evidence; where it is not, an owner adopting this protocol should expect an early phase of aligning commissioning data governance with construction data governance before commissioning-family features can be populated reliably, and should treat any interim model that lacks them as provisional rather than final.

Future work should examine causal interventions after predictive validation. Questions include whether earlier vendor escalation reduces delay, whether added commissioning capacity improves service readiness, and whether acceleration increases defect-driven variance. Predictive importance should nominate hypotheses, while causal designs evaluate actions.

A further limitation deserves explicit statement: this protocol assumes that organizations already retain schedule snapshots, procurement events, and commissioning records in a form that supports as-of reconstruction. Many owners and contractors currently overwrite schedule updates or retain only the latest procurement status, which means the first deliverable of any execution of this protocol may be a data-retention change rather than a model. Organizations beginning from this position should treat Level 1 of the maturity progression, establishing controlled snapshots and objective progress measurement, as a prerequisite phase with its own timeline, not as a step that can be compressed to reach model development sooner.

Federated analysis may support cross-owner learning without pooling raw data. Shared feature definitions, secure aggregation, and external validation would test transfer while protecting confidentiality. A public benchmark using de-identified or synthetic schedule networks would improve reproducibility. Any synthetic set must preserve temporal and dependency properties without being represented as actual performance. A federated design is attractive in this sector specifically because no single owner, however large, delivers enough campuses to satisfy the project-level holdout standard this protocol requires with high confidence. An owner with twenty completed campuses can estimate a reference class, but cannot reliably estimate rare, high-consequence events, a first-of-kind cooling technology failing integrated testing, for example, from twenty observations alone. Secure aggregation across owners, contractors, and commissioning firms would pool exactly these rare events without requiring any party to disclose commercially sensitive vendor performance or contract terms, provided the shared feature definitions in Table 1 are adopted consistently enough to make pooled records comparable.

9 Conclusion

Historical schedule performance data support credible delay prediction only when records preserve what was known at each forecast date. The proposed protocol defines project-snapshot observations, operational outcomes, as-of feature construction, project-level temporal validation, strong baselines, uncertainty reporting, explainability, decision thresholds, and deployment monitoring. The decisive controls are methodological. Projects, not activity rows, must remain separated across training and test sets. Later information must not enter earlier forecasts. Performance must include calibration, intervals, horizon, subgroup behavior, and decision value. Advanced models must outperform current schedules and simple historical baselines.

None of these controls is unique to machine learning. A purely statistical baseline, an earned-schedule projection, or a human forecaster is equally subject to project-level holdout logic, temporal integrity, and calibration reporting once the comparison is made explicit. The protocol's contribution is to make that comparison unavoidable: any candidate, however produced, enters the same evaluation pipeline and is judged against the same current-forecast and historical-median baselines under the same project-level split. A hyperscale owner adopting this protocol should therefore expect its first concrete output to be a disciplined benchmark of existing practice, not necessarily a new algorithm.

The paper reports no synthetic performance claim. Its next step is execution on an authorized multi-project dataset, followed by silent prospective validation. Only measured results should support claims of accuracy, generalizability, or schedule improvement.

This protocol is the second paper in a planned three-part program. The first paper in this series established why deterministic schedule assurance reaches its limit on hyperscale AI data-center programs and proposed a layered predictive architecture as a conceptual response; this paper operationalizes the data, feature, and validation requirements that architecture presupposes but did not itself specify in executable form. A subsequent paper is expected to develop and validate the resulting framework against the propositions and acceptance criteria defined here, closing the sequence from conceptual architecture through empirical protocol to tested framework.

Appendix A Operational specification of candidate predictors

This appendix specifies, by feature family, how the predictors listed in Table 1 should be operationalized so that independent teams can reconstruct the same variable from the same source systems. Every family shares four non-negotiable rules: a feature must be computable using only records with a system timestamp on or before the forecast date; its source system, unit, and permissible range must be documented; missingness must be flagged rather than silently imputed at the pipeline stage; and any transformation or imputation must be fit inside each training fold, never on the full dataset. Association between a feature and delay should prompt investigation of the underlying process, not a claim of causation.

Baseline features. Power capacity, facility topology, delivery model, design maturity, and utility scope are fixed or slowly changing at sanction and describe the scale and governance context of the project rather than its trajectory. They come from the project charter, design basis, contract records, document control, and interconnection plan. Their predictive role is chiefly to define the reference class a project belongs to: a 2N, liquid-cooled, design-build campus with owner-controlled interconnection is a different prediction problem from an N+1, air-cooled, design-bid-build campus on a utility-controlled substation, and baseline features let the model, and the reference class described in Section 2, condition on that difference before any time-varying evidence arrives. **Schedule features.** Logic density, open-end rate, constraint density, near-critical mass, forecast movement, logic churn, remaining-duration growth, out-of-sequence rate, and critical-path turnover are computed from each schedule snapshot and its predecessor. They measure the health of the network itself rather than physical progress, and they exist because a milestone date calculated from a degraded network, one with excessive open ends, forced constraints, or a critical path that changes identity every update, is not a reliable input to any downstream model regardless of algorithm. These features should be reviewed before, not after, any duration or risk model is trained, consistent with the schedule-quality gating principle carried over from the companion review paper in this series. **Design features.** Release reliability, revision intensity, request-for-information aging, submittal cycle time, first-pass approval rate, and interface-issue age come from document control, the RFI system, the submittal log, and the issue log. They lead procurement and field risk because a package released late, or approved only after repeated resubmission, propagates directly into fabrication and installation dates; interface-issue age in particular captures the specific data-center risk that electrical, mechanical, and controls packages must reconcile with one another before any single trade can proceed with confidence. **Procurement features.** PO release variance, engineering-approval evidence, manufacturing stage, factory-test readiness and outcome, logistics status, promise volatility, and shared-vendor exposure come from the procurement system, vendor reports, the quality system,

and the logistics system. This family exists because a vendor's promised ship date is a target, not an observation: engineering-approval evidence and verified manufacturing stage are the closest available proxies for physical progress, and promise volatility, counting how often and by how much a date has moved, distinguishes a vendor executing to plan from one whose commitments should not be taken at face value. Production and quality features. Installed quantity rate, plan reliability, constraint removal, crew stability, congestion, and material readiness describe field execution, while inspection pass rate, nonconformance rate, rework hours, closure age, and repeat defect rate describe whether that execution will hold up under acceptance testing. The two families are grouped together operationally because a rising installed-quantity rate accompanied by a falling inspection pass rate is a specific, actionable pattern, work is proceeding quickly but not durably, that neither family alone would flag with the same confidence. Plan-reliability measurement in particular follows Last Planner System metrics validated against schedule performance in prior construction research (Lagos et al., 2024).

Commissioning features. Prerequisite completeness, script approval, controls-point verification, first-test pass rate, severe open defects, retest cycle time, witness capacity, and turnover completeness come from the commissioning platform, the controls platform, and the document system. This family carries the largest incremental weight near service readiness because commissioning is where construction status and operational acceptance either converge or diverge: a hall can be physically complete while failing integrated systems testing, and only commissioning-specific features observe that gap before it appears as a missed milestone.

External and portfolio features. Utility evidence state, permit comment age, weather exposure, and inspection availability represent conditions outside the delivery organization's direct control, while shared specialist load, shared equipment exposure, and utility concentration represent risk shared across projects in the same portfolio. Both groups require governance different from endogenous features: an alert driven by utility evidence state should trigger escalation and contingency planning, not a request that the field team simply work faster, and a portfolio feature indicating three campuses drawing on one commissioning specialist pool should trigger resequencing analysis at the portfolio level, not at any single project.

Outcome and governance features. Committed milestone date, actual milestone date, and service readiness define what the model predicts, while data freshness, override magnitude, and intervention completion describe the operating condition of the forecasting system itself. Governance features exist because a deployed model is itself part of the delivery system: stale inputs should suppress confident output, and recording the magnitude and rationale of every human override is what allows later analysis to compare model-only, human-only, and combined forecasts, the comparison Section 7 identifies as necessary before any model is trusted with unreviewed authority.

Declarations

Ethics approval and consent to participate

The retrospective study requires review under the policies governing the participating organizations before project data are accessed. No project or personal data were analyzed for this methods paper.

Consent for publication

Not applicable.

Availability of data and materials

No project-level dataset was analyzed for this methods paper. The manuscript specifies the data model, feature dictionary, and validation protocol required for later execution on an authorized, owner- or contractor-approved historical project dataset.

Competing interests

The authors declare no competing interests.

Funding

The authors received no specific funding for this work.

Authors contributions

Jiongwen Zhou, Elizabeth Onyekachi Umekwe-Edevwie, Chinyere Famakinwa, Promise Oduh, and Peace Oduh jointly contributed to the study concept, protocol design, analytical framework, critical review, and approval of the manuscript; no individual author's contribution was limited to a single role. Final submission should retain only contributions verified by each author.

Acknowledgements

None.

References

- Batselier, J., and Vanhoucke, M. (2015). Empirical evaluation of earned value management forecasting accuracy for time and cost. *Journal of Construction Engineering and Management*, 141(11), 05015010.
- Biecek, P., and Burzykowski, T. (2021). *Explanatory Model Analysis*. CRC Press.
- Egwim, C. N., Alaka, H., Toriola-Coker, L. O., Balogun, H., and Sunmola, F. (2021). Applied artificial intelligence for predicting construction projects delay. *Machine Learning with Applications*, 6, 100166. <https://doi.org/10.1016/j.mlwa.2021.100166>
- Gondia, A., Siam, A., El-Dakhakhni, W., and Nassar, A. H. (2020). Machine learning algorithms for construction projects delay risk prediction. *Journal of Construction Engineering and Management*, 146(1), 04019085. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001736](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001736)
- Hulett, D. T. (2009). *Practical Schedule Risk Analysis*. Gower.
- Kuhn, M., and Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
- Lagos, C. I., Herrera, R. F., Alarcon, L. F., and Salazar, L. A. (2024). Predicting construction schedule performance with Last Planner System metrics. *Automation in Construction*, 2024.
- Molnar, C. (2022). *Interpretable Machine Learning*. Second edition.
- Project Management Institute. (2019). *Practice Standard for Scheduling*. Third edition.
- Riley, R. D., van der Windt, D., Croft, P., and Moons, K. G. M. (2019). *Prognosis Research in Healthcare: Concepts, Methods, and Impact*. Oxford University Press.
- Santos, J. I., Pereda, M., Ahedo, V., and Galan, J. M. (2023). Explainable machine learning for project management control. *Computers and Industrial Engineering*, 180, 109261.
- Shittu, M. A., Shittu, H. A., Adeleke, O. J., & Adedokun, O. J. (2023). Digital twin modeling for real time monitoring and fault detection in smart substations. *International Journal of Industrial Engineering Research and Development*, 14(2), 25-44. https://doi.org/10.34218/IJIERD_14_02_003
- Steyerberg, E. W. (2019). *Clinical Prediction Models*. Second edition. Springer.
- Vazquez, A. (2023). Emergence of scaling in project schedules. arXiv:2312.12906.
- Yaseen, Z. M., Ali, Z. H., Salih, S. Q., and Al-Ansari, N. (2020). Prediction of risk delay in construction projects using a hybrid artificial intelligence model. *Sustainability*, 12(4), 1514. <https://doi.org/10.3390/su12041514>
- Ökmen, Ö., & Öztaş, A. (2008). Construction Project Network Evaluation with Correlated Schedule Risk Analysis Model. *Journal of Construction Engineering and Management*, 134(1), 49-63. [https://doi.org/10.1061/\(asce\)0733-9364\(2008\)134:1\(49\)](https://doi.org/10.1061/(asce)0733-9364(2008)134:1(49))
- Song, Y., & Vanhoucke, M. (2025). Schedule risk analysis for project control with risk interactions. *Annals of Operations Research*. <https://doi.org/10.1007/s10479-025-06668-8>
- National Institute of Construction Management and Research Pune India., Agarwal, A. L., Mahajan, D. A., & National Institute of Construction Management and Research Pune India, (2017). A PROBABILITY ANALYSIS OF CONSTRUCTION PROJECT SCHEDULE USING RISK MANAGEMENT TOOL. *MATTER: International Journal of Science and Technology*, 3(1), 104-109. <https://doi.org/10.20319/mijst.2017.31.104109>
- Andrade, P. A. D., Martens, A., & Vanhoucke, M. (2019). Using real project schedule data to compare earned schedule and earned duration management project time forecasting capabilities. *Automation in Construction*, 99, 68-78. <https://doi.org/10.1016/j.autcon.2018.11.030>

- Avlijaš, G., & Radunović, M. (2019). Application of Event Chain Methodology in Schedule Risk Analysis. *European Project Management Journal*, 9(2), 26-34. <https://doi.org/10.18485/epmj.2019.9.2.3>
- Virle, R., & Mhaske, S. (2013). Application of Earned Value and Earned Schedule to Construction Project. *International Journal of Scientific Engineering and Research*, 1(1), 97-101. <https://doi.org/10.70729/5130926>
- Nasir, D., McCabe, B., & Hartono, L. (2003). Evaluating Risk in Construction-Schedule Model (ERIC-S): Construction Schedule Risk Model. *Journal of Construction Engineering and Management*, 129(5), 518-527. [https://doi.org/10.1061/\(asce\)0733-9364\(2003\)129:5\(518\)](https://doi.org/10.1061/(asce)0733-9364(2003)129:5(518))
- Hendradewa, A. P. (2019). Schedule Risk Analysis by Different Phases of Construction Project Using CPM-PERT and Monte-Carlo Simulation. *IOP Conference Series: Materials Science and Engineering*, 528(1), 012035. <https://doi.org/10.1088/1757-899x/528/1/012035>
- Shu-Shun, L., & Shih, K. C. (2009). A framework of critical resource chain for project schedule analysis. *Construction Management and Economics*, 27(9), 857-869. <https://doi.org/10.1080/01446190903171196>
- Mabeba, M., Pretorius, J. H. C., & Pretorius, L. (2018). SCHEDULE RISK ANALYSIS OF RAILWAY PROJECTS USING MONTE CARLO SIMULATION FOR IMPROVED PROJECT MANAGEMENT. *Proceedings of International Structural Engineering and Construction*, 5(2). <https://doi.org/10.14455/isec.res.2018.124>
- Luu, V. T., Kim, S. Y., Tuan, N. V., & Ogunlana, S. O. (2009). Quantifying schedule risk in construction projects using Bayesian belief networks. *International Journal of Project Management*, 27(1), 39-50. <https://doi.org/10.1016/j.ijproman.2008.03.003>
- Moselhi, O., & Rofigari-Esfahan, N. (2013). Project schedule compression: a multi-objective methodology. *Construction Innovation*, 13(4), 374-393. <https://doi.org/10.1108/ci-03-2011-0010>
- Dubrovskaya, T. A., Kravchenya, I. N., & Strizhak, A. I. (2021). Features of using Microsoft Project software when calculating the construction schedule. *Modern Technologies. System Analysis. Modeling*, (4), 212-224. [https://doi.org/10.26731/1813-9108.2021.4\(72\).212-224](https://doi.org/10.26731/1813-9108.2021.4(72).212-224)
- Okmen, O. (2013). Evaluating Schedule Uncertainty in Unit-Based Repetitive Building Projects. *Journal of Construction Engineering and Project Management*, 3(2), 21-34. <https://doi.org/10.6106/jcepm.2013.3.2.021>
- Fitzsimmons, J. P., Lu, R., Hong, Y., & Brilakis, I. (2022). Construction schedule risk analysis - a hybrid machine learning approach. *Journal of Information Technology in Construction*, 27, 70-93. <https://doi.org/10.36680/j.itcon.2022.004>
- Carr, V., & Tah, J. H. M. (2001). A fuzzy approach to construction project risk assessment and analysis: construction project risk management system. *Advances in Engineering Software*, 32(10-11), 847-857. [https://doi.org/10.1016/s0965-9978\(01\)00036-9](https://doi.org/10.1016/s0965-9978(01)00036-9)
- Curtis, B. (1993). Engineering construction risks - A guide to project risk analysis and risk management. *International Journal of Project Management*, 11(3), 187-188. [https://doi.org/10.1016/0263-7863\(93\)90053-p](https://doi.org/10.1016/0263-7863(93)90053-p)
- Sami Ur Rehman, M., Thaheem, M. J., Nasir, A. R., & Khan, K. I. A. (2020). Project schedule risk management through building information modelling. *International Journal of Construction Management*, 22(8), 1489-1499. <https://doi.org/10.1080/15623599.2020.1728606>

- Carpenter, C. (2025). Integrated Cost and Schedule Risk Analysis Eases Consequences of Project Delays. *Journal of Petroleum Technology*, 77(12), 1-4. <https://doi.org/10.2118/1225-0009-jpt>
- Andreas, A., Tinumbia, N., & Anggraini, E. (2023). CONSTRUCTION PROJECT CONTROL ANALYSIS USING EARNED VALUE MANAGEMENT, EARNED SCHEDULE MANAGEMENT, EARNED DURATION MANAGEMENT. *Jurnal Infrastruktur*, 9(2), 87-98. <https://doi.org/10.35814/infrastruktur.v9i2.4943>
- Xu, X., Wang, J., Li, C. Z., Huang, W., & Xia, N. (2018). Schedule risk analysis of infrastructure projects: A hybrid dynamic approach. *Automation in Construction*, 95, 20-34. <https://doi.org/10.1016/j.autcon.2018.07.026>
- Rahman, M. M., Lee, Y. D., & Ha, D. K. (2014). Investigating Main Causes for Schedule Delay in Construction Projects in Bangladesh. *Journal of Construction Engineering and Project Management*, 4(3), 33-46. <https://doi.org/10.6106/jcepm.2014.4.3.033>
- Sullivan, J., Asmar, M. E., Chalhoub, J., & Obeid, H. (2017). Two Decades of Performance Comparisons for Design-Build, Construction Manager at Risk, and Design-Bid-Build: Quantitative Analysis of the State of Knowledge on Project Cost, Schedule, and Quality. *Journal of Construction Engineering and Management*, 143(6). [https://doi.org/10.1061/\(asce\)co.1943-7862.0001282](https://doi.org/10.1061/(asce)co.1943-7862.0001282)
- Tabish, S., Pandian, S. A. H., & Jha, K. N. (2011). Analysis of Contract Clauses Affecting The Schedule Performance of A Construction Project. *Journal of Real Estate, Construction & Management*, 26(1), 34-51. <https://doi.org/10.1177/2977657020110103>
- Karimi, H., Taylor, T. R. B., & Goodrum, P. M. (2017). Analysis of the impact of craft labour availability on North American construction project productivity and schedule performance. *Construction Management and Economics*, 35(6), 368-380. <https://doi.org/10.1080/01446193.2017.1294257>
- Choudhry, R. M., Aslam, M. A., Hinze, J. W., & Arain, F. M. (2014). Cost and Schedule Risk Analysis of Bridge Construction in Pakistan: Establishing Risk Guidelines. *Journal of Construction Engineering and Management*, 140(7). [https://doi.org/10.1061/\(asce\)co.1943-7862.0000857](https://doi.org/10.1061/(asce)co.1943-7862.0000857)
- TAKAHASHI, H., & HIRAKI, K. (2006). Integrated evaluation on the period/schedule of dam construction project. *Journal of Construction Management, JSCE*, 13, 63-75. <https://doi.org/10.2208/procm.13.63>
- Chaterine,, & A. Simanjuntak, M. R. (2020). Analysis of Schedule Project Management's Indicators and Cost Project Management's Indicators in Interior Construction. *IOP Conference Series: Materials Science and Engineering*, 1007(1), 012083. <https://doi.org/10.1088/1757-899x/1007/1/012083>
- Mahamid, I. (2017). Analysis of schedule deviations in road construction projects and the effects of project physical characteristics. *Journal of Financial Management of Property and Construction*, 22(2), 192-210. <https://doi.org/10.1108/jfmpc-07-2016-0031>
- Ballesteros-Pérez, P., Cerezo-Narváez, A., Otero-Mateo, M., Pastor-Fernández, A., & Vanhoucke, M. (2019). Performance comparison of activity sensitivity metrics in schedule risk analysis. *Automation in Construction*, 106, 102906. <https://doi.org/10.1016/j.autcon.2019.102906>
- Yang, J. (2021). Application Analysis of Computer Technology in Construction Project Schedule Control in Information Age. *Journal of Physics: Conference Series*, 1881(3), 032090. <https://doi.org/10.1088/1742-6596/1881/3/032090>

- Koulinas, G., Xanthopoulos, A., Tsilipiras, T., & Koulouriotis, D. (2020). Schedule Delay Risk Analysis in Construction Projects with a Simulation-Based Expert System. *Buildings*, 10(8), 134. <https://doi.org/10.3390/buildings10080134>
- Ma, G., & Wu, M. (2019). A Big Data and FMEA-based construction quality risk evaluation model considering project schedule for Shanghai apartment projects. *International Journal of Quality & Reliability Management*, 37(1), 18-33. <https://doi.org/10.1108/ijqrm-11-2018-0318>
- Kiruthiga Lakshmi Narayanaswamy,, & Sathyabama University, (2016). Construction Technologies for Effective Project Schedule in High Rise Construction - A Review. *International Journal of Engineering Research and*, V5(11). <https://doi.org/10.17577/ijertv5is110311>
- Salanke, P. V., & El-Gafy, M. (2013). Using Risk Analysis To Model Construction Schedule Delays: A Bayesian Belief Networks Approach. *Journal of Real Estate, Construction & Management*, 28(1), 5-15. <https://doi.org/10.1177/2977657020130101>
- Tahrir, M., Ramadhani, F., & Baco, D. A. (2023). Critical Path Analysis Using Microsoft Project 2016 on the Implementation Schedule of Residential House Construction Projects. *TEPIAN*, 4(4), 196-206. <https://doi.org/10.51967/tepiant.v4i4.2969>
- Hossen, M. M., Kang, S., & Kim, J. (2015). Construction schedule delay risk assessment by using combined AHP-RII methodology for an international NPP project. *Nuclear Engineering and Technology*, 47(3), 362-379. <https://doi.org/10.1016/j.net.2014.12.019>
- Kanade, U. (2026). Digital Twin Implementation for Probabilistic Construction Schedule Risk Analysis. *International Journal of Creative and Open Research in Engineering and Management*, 02(08), 1-9. <https://doi.org/10.55041/ijcope.v2i8.041>
- Bansal, V. K., & Pal, M. (2009). Construction schedule review in GIS with a navigable 3D animation of project activities. *International Journal of Project Management*, 27(5), 532-542. <https://doi.org/10.1016/j.ijproman.2008.07.004>
- Franck, B., Neumann, K., & Schwindt, C. (2001). Truncated branch-and-bound, schedule-construction, and schedule-improvement procedures for resource-constrained project scheduling. *OR-Spektrum*, 23(3), 297-324. <https://doi.org/10.1007/pl00013356>
- Zhao, Z., Meng, Q., & Pang, N. (2020). Schedule Risk Analysis of EPC Project for Power Transmission and Transformation Project Led by Design Enterprise. *Journal of Physics: Conference Series*, 1646(1), 012158. <https://doi.org/10.1088/1742-6596/1646/1/012158>
- Arvikar, S., Aung, P. P. W., Cha, G., & Park, S. (2025). Augmented Reality-based construction site management using optimized BIM and project schedule integration. *Automation in Construction*, 175, 106204. <https://doi.org/10.1016/j.autcon.2025.106204>
- Sun, C., Man, Q., & Wang, Y. (2015). Study on BIM-based construction project cost and schedule risk early warning. *Journal of Intelligent & Fuzzy Systems*, 29(2), 469-477. <https://doi.org/10.3233/ifs-141178>
- Sun, H. F., Zhou, L. P., & Wang, Y. Y. (2013). Joint Analysis of the Power Generation Construction Project Schedule, Cost and Quality Based on DEA. *Applied Mechanics and Materials*, 467, 282-286. <https://doi.org/10.4028/www.scientific.net/amm.467.282>
- Bunni, N. G. (1993). Engineering construction risks: a guide to project risk analysis and risk management. *Engineering Structures*, 15(6), 466-467. [https://doi.org/10.1016/0141-0296\(93\)90065-c](https://doi.org/10.1016/0141-0296(93)90065-c)

- Bogucki, M., & Polonski, M. (2019). Risk Analysis for High Pressure Gas Pipeline Construction Schedule. IOP Conference Series: Materials Science and Engineering, 471, 112042. <https://doi.org/10.1088/1757-899x/471/11/112042>
- Isah, M. A., & Kim, B. S. (2021). Integrating Schedule Risk Analysis with Multi-Skilled Resource Scheduling to Improve Resource-Constrained Project Scheduling Problems. Applied Sciences, 11(2), 650. <https://doi.org/10.3390/app11020650>
- Yang, J. B., & Lai, T. H. (2023). Selecting EVM, ESM and EDM(t) for managing construction project schedule. Engineering, Construction and Architectural Management, 31(12), 4988-5006. <https://doi.org/10.1108/ecam-02-2023-0115>
- Liberatore, M. J. (2002). Project Schedule Uncertainty Analysis using Fuzzy Logic. Project Management Journal, 33(4), 15-22. <https://doi.org/10.1177/875697280203300403>
- Siregar, S., Jayady, A., & S. Pribadi, K. (2026). Assessing the Effectiveness of Procurement Risk Mitigation on Cost, Quality, And Schedule Performance in Industrial Construction Projects. Engineering and Technology Journal, 11(06). <https://doi.org/10.47191/etj/v11i06.50>
- Yin, S. (2023a). A construction schedule management method of large-scale construction project based on BIM model. International Journal of Critical Infrastructures, 19(1), 1. <https://doi.org/10.1504/ijcis.2023.10046163>
- Yin, S. (2023b). A construction schedule management method of large-scale construction project based on BIM model. International Journal of Critical Infrastructures, 19(2), 140. <https://doi.org/10.1504/ijcis.2023.130456>
- Yin, S. (2023c). A construction schedule management method of large-scale construction project based on BIM model. International Journal of Critical Infrastructures, 19(3), 1. <https://doi.org/10.1504/ijcis.2023.10046738>
- Kog, Y. C., Chua, D. K. H., Loh, P. K., & Jaselskis, E. J. (1999). Key determinants for construction schedule performance. International Journal of Project Management, 17(6), 351-359. [https://doi.org/10.1016/s0263-7863\(98\)00058-1](https://doi.org/10.1016/s0263-7863(98)00058-1)
- Paul, V. K., & Basu, C. (2016). Scenario Planning and Risk Failure Mode Effect and Analysis (RFMEA) based Management. Journal of Construction Engineering and Project Management, 6(2), 24-29. <https://doi.org/10.6106/jcepm.2016.6.2.024>
- Mahamid, I. (2013). CONTRIBUTORS TO SCHEDULE DELAYS IN PUBLIC CONSTRUCTION PROJECTS IN SAUDI ARABIA: OWNERS' PERSPECTIVE. Journal of Construction Project Management and Innovation, 3(2), 608-619. <https://doi.org/10.36615/jcpmi.v3i2.46>
- Dziadosz, A., & Rejment, M. (2015). Risk Analysis in Construction Project - Chosen Methods. Procedia Engineering, 122, 258-265. <https://doi.org/10.1016/j.proeng.2015.10.034>
- Dai, W. J. (2014). Risk Analysis of a Large-Scale Construction Project. Advanced Materials Research, 989-994, 5070-5074. <https://doi.org/10.4028/www.scientific.net/amr.989-994.5070>
- Hegazy, T., Said, M., & Kassab, M. (2011). Incorporating rework into construction schedule analysis. Automation in Construction, 20(8), 1051-1059. <https://doi.org/10.1016/j.autcon.2011.04.006>
- Nadia Diandra, (2025). Analyzing Construction Schedule Deviation Using S-Curve: A Residential Project Case. Jurnal Teknologi dan Desain, 1(01), 58-68. <https://doi.org/10.51170/jtd.v1i01.115>