



Data-Driven Machine Learning Approach to Integrate Field Submittals in Project Scheduling

Mohamad Awada¹; F. Jordan Srour, Ph.D.²; and Issam M. Srour, Ph.D., A.M.ASCE³

Abstract: Construction projects are data-rich environments. However, those data are usually captured for site-specific reasons, e.g., the filing and approval of inspection requests, with little regard to how they can be leveraged for improved project management. Typically, scheduling techniques rely on general probability estimates, which do not capture the details of the site processes causing schedule deviations. This paper illustrates how machine learning techniques can mine project data to forecast delay in the midst of the project. The proposed method uses concrete pouring requests as an example of a site data stream and implements a random forest predictive model to forecast the likelihood of acceptance for these requests. Embedded in the proposed approach is an analysis that allows for the addition of probabilistic time delays associated with the forecast of rejected requests. The methodology was tested on a real-world case study, allowing for the comparison between a project duration estimate based on critical path method (CPM) with static buffers and a project duration obtained using the proposed method. The results show a difference of 10% between the two durations. The paper shows how using data streams from a construction site with machine learning techniques can enhance project duration estimates in execution. DOI: 10.1061/(ASCE)ME.1943-5479.0000873. © 2020 American Society of Civil Engineers.

Author keywords: Field submittals; Data analytics; Machine learning; Scheduling.

Introduction

Modern construction sites have an abundant amount of data including engineering data from the planning and design stages, accounting and job progress data, and field data collected by wearables, mobile devices, and sensors on equipment/materials (Bilal et al. 2016). However, it is postulated that 96% of the data collected during construction goes unused (Snyder et al. 2018). Furthermore, despite the large effort invested in construction scheduling to reduce delays, construction projects are still experiencing an average schedule overrun of 42.7% (Ansar et al. 2016), which necessitates the search for novel tools to strengthen construction managers' decision making and improve the accuracy of schedule estimates. Data analytics has the potential to change that reality (Wu et al. 2014).

Various techniques have been developed to help project managers schedule the required project work according to the prescribed timeline. In the construction field, the critical path method (CPM) has been widely accepted as the main project planning technique. Typically, the CPM relies on network techniques to ascertain the project duration on the basis of task early and late start dates

derived as the result of logical constraints, task durations, and various other activity characteristics that dictate the project's critical path (Lu and Li 2003). Despite numerous advantages for using the CPM, one of its main pitfalls is the assumption of preset task durations, which are fixed based on engineering judgments and previous experience (Alves and Tommelein 2004), thus neglecting the variability found on constructions sites.

Some project planning tools merge probabilistic scheduling techniques with the CPM to account for the variability associated with construction work (Wang 2005). The program evaluation and review technique (PERT) strategy works by using a distribution of task completion times rather than a point estimate. However, using PERT can lead to an optimistic estimation of the project duration if all noncritical paths are ignored and the critical path is the focus of analysis (Lee and Arditi 2006). Another probabilistic strategy is that of Monte Carlo simulation, which recognizes the duration of each activity as a probability distribution from which potential task durations are drawn during extensive simulation, yielding a cumulative probabilistic curve of the total project duration (Khedr 2006; Dawood 1998). Fuzzy models of task duration offer another method to incorporate variability in task duration estimations (Balta et al 2018; Marín Ruiz et al 2018). A major drawback of these probabilistic techniques is the effort required to estimate the necessary distribution parameters, which require calibration through previous experience and archived data. This process generally yields domain- or context-specific estimates with limited generalizability to other projects (McCabe 2003).

Closer to the domain of machine learning, the case-based reasoning (CBR) technique is another scheduling technique that works by archiving data on past projects. The distance between a new project and a collection of past projects is then calculated. The closest or most similar past cases then serve to inform the cost or schedule duration estimate for the novel case (Li et al. 2017). Users of this method usually rely on the subjective assessment of experts to assign weights for the features describing the cases (Zhao et al. 2019). Because these methods focus on the case level, they are better suited for estimating the cost or schedule of a project a priori.

¹Research Assistant, Dept. of Civil and Environmental Engineering, American Univ. of Beirut, P.O. Box 1107-2020, Beirut 11-0236, Lebanon. Email: mohamadawada14@gmail.com

²Associate Professor, Dept. of IT and Operations Management, Lebanese American Univ., P.O. Box 13-5053, Chouran, Beirut 1102 2801, Lebanon. ORCID: <https://orcid.org/0000-0001-7623-723X>. Email: jordan.srou@lau.edu.lb

³Associate Professor, Dept. of Civil and Environmental Engineering, American Univ. of Beirut, P.O. Box 1107-2020, Beirut 11-0236, Lebanon (corresponding author). ORCID: <https://orcid.org/0000-0002-3498-6790>. Email: is04@aub.edu.lb

Note. This manuscript was submitted on February 17, 2020; approved on August 26, 2020; published online on November 9, 2020. Discussion period open until April 9, 2021; separate discussions must be submitted for individual papers. This paper is part of the *Journal of Management in Engineering*, © ASCE, ISSN 0742-597X.

Once a project is underway, the project features evolve, yielding a novel partial case with novel causes of delay, thus hindering the application of this method for in-execution updates.

Multiple authors have sought to develop reliable and generalizable estimates of task duration variation by studying the causes of delay. Wambeke et al. (2011) organized the causes of task duration variation into eight categories: (1) prerequisite work; (2) detailed design/working method; (3) labor force; (4) tools and equipment; (5) material and components; (6) work/jobsite conditions; (7) management/supervision/information flow; and (8) weather or external conditions. Subsequent to the aforementioned study, other researchers focused on specific root causes such as interruptions in the supply of materials or equipment, design errors irregular financing, and equipment downtime (Rachid et al. 2019; Zidane and Andersen 2018; Durdyyev et al. 2017). In general, for contractors to accommodate all of these uncertainties in the project schedule, a time reserve or buffer is added (Russell et al. 2013). In order to make the buffer as accurate as possible, a recent study by Gondia et al. (2020) related nine root causes (owner, consultant, contractor, design, labor, material, equipment, project, and external aspects) to the severity of task overrun via two machine learning techniques—decision trees and naive Bayesian classification algorithms. Given a general lack of electronic data in the location of their research (Egypt), they measured the root causes in terms of risk scores, which were solicited from project managers based on lower-order risk factors yielding a data set with 51 records. In contrast, this study examines actual site data taken from 1,001 concrete pour requests (CPRs) of a large residential project.

A CPR is an inspection request for various structural elements prior to pouring concrete to ensure that all work has been properly completed. It is important to have the supervising engineer, representing the project owner, check for any defects (e.g., steel reinforcement diameter/spacing and coating, formwork and staging, and spacing between other imbedded items) before pouring. Once concrete has been poured, there is no way to know what issues exist in the structure (Jenkins and Lew 2003). Typically, contractors file the CPR and propose a date for pouring; however, they cannot proceed with the pouring activity until they receive the approval from the site personnel representing the owner. In this way, the acceptance or rejection of a CPR represents a tangible outcome of many of the aforementioned root causes identified, while at the same time providing a measure of delay through the need for rework following a rejection decision.

Several studies have investigated the effect of inspection request submittal rejection on construction schedules. In a questionnaire delivered to 52 companies asking about the main causes of delays on construction sites, 74% of the respondents considered that late approval and rejection of submittals was a major factor of delay caused by the consultant onsite (Ren et al. 2008). Ko and Li (2014) showed, using conventional review/inspection systems, that a 50% approval rate of submittals on construction sites can result in a 6% delay in construction schedules.

Given that the contractor cannot anticipate if an inspection request will be rejected or accepted, there is an embedded uncertainty associated with these inspections, and thus the flow of successor activities depends heavily on the acceptance of every inspection. This uncertainty can be eliminated by using data analytics to extract insights from construction data to increase the reliability of the project schedule (Han and Pan 2011). Contractors and project managers are flooded with data and information but do not generally have the necessary analytical tools to explain and analyze the acquired data in ways that enhance the decision-making process during construction. Data analytics is a solution for the problem of accurate project duration estimation (Cho et al. 2013).

This paper proposes a data analytics-based methodology that leverages site-level data in predicting the remaining project duration during the construction phase. The proposed methodology focuses on the use of inspection requests as a data artefact from a construction project in Beirut, Lebanon. Based on archived CPRs, a machine learning (ML) model predicts whether a CPR filed by the contractor will be rejected or accepted by the supervising engineer. The model, equipped with robust data analysis, will help the contractor plan for the lost time associated with the requests for which rejection has been forecasted, and as such obtain an accurate project duration estimation.

Literature Review

Data analytics is broadly defined as a process by which important questions are framed and answered through the discovery, interpretation, and communication of meaningful patterns in data. These patterns in the data can simply be described as in descriptive analytics, can be used for forecasting as in predictive analytics, and ultimately to inform decision making as in prescriptive analytics. In this work, the focus is on predictive analytics because the strategy introduced predicts the acceptance/rejection of a CPR and then predicts the impact of the outcome on the project schedule.

The implementation of data analytics methods such as simulations, optimization, statistical analysis, and so on, gives the contractor the chance to produce more accurate schedules. For example, Ökmen and Öztaş (2008) presented a simulation tool to evaluate construction activity networks under uncertainty. The authors speculated that the larger and more accurate the data introduced into the simulation are, the greater the reduction in project schedule risks is through better decision making. Focusing on decision making to achieve project duration targets, Jun and El-Rayes (2009) presented a multiobjective optimization model to simultaneously optimize resource allocation and leveling. The proposed model uses a genetic algorithm framework to minimize the project duration and maximize resource utilization efficiency. König et al. (2012) integrated building information modeling and discrete event simulation to automatically define valid interdependencies among construction activities and as such create a robust estimation of the construction project schedule.

Esmaili and Matthew (2013) integrated safety risk data with project schedules. The proposed framework can give contractors an improved understanding of all high-risk construction activities, thus avoiding future safety risks and their implied delays to projects. More recently, De Andrade et al. (2019) used documented data from 125 construction projects to improve the accuracy of project duration estimation through the application of a forecasting framework. Similarly, Gondia et al. (2020) used two machine learning techniques to predict task overruns based on stated risk scores across multiple areas on 51 construction projects. The strategy recommended in this paper also depends on machine learning as a predictive analytics tool.

Machine Learning and Random Forest Algorithm

Machine learning refers to a set of algorithmic structures designed to provide computer systems with the ability to learn and improve their performance via the discovery of patterns in data. In this way, ML techniques can perform decision-making tasks with minimal human interference (Alpaydin 2009). At the broadest level, ML methods are designated as supervised, unsupervised, or reinforcement (Burkov 2019). Supervised machine learning algorithms are designated as such because the user specifies the target for the algorithm. Thus, the algorithm focuses on modeling the relationship

between a set of input variables (predictors) and the specified target. Supervised learning is useful for problems including regression and classification. In contrast, unsupervised learning algorithms are not given a target variable, but are trained with unlabeled data and use mathematical techniques to mine for rules and group instances, thereby revealing hidden patterns. In this paper, supervised learning techniques are invoked to predict CPR acceptance or rejection on the basis of a set of predictor variables—both the target and predictor variables are fed to the model from past data.

Within supervised learning, methods include (1) classification, (2) regression, (3) clustering, and (4) association (Witten et al. 2016). In general, both classification and regression analysis discover predictive patterns; however, classification predicts categorical or nominal features whereas regression predicts real number features. Cluster analysis is the process of segmenting a data set into subsets or clusters so that the data points within a cluster are like each other more than those in other clusters. Association rule learning is a rule-based technique that reveals hidden relations between variables in a large data set.

A major supervised algorithm falling within the classification and regression category of learning algorithms is the random forest (RF) algorithm. The RF algorithm is a flexible and easy to use ensembling ML algorithm that yields highly accurate results for classification problems. Ensembling is a supervised ML method that merges multiple base models to produce a single optimal predictive model (Zhang and Ma 2012). A RF algorithm creates multiple classification and regression trees (CARTs), also known as decision trees, and then combines their individual output to derive the outcome.

Decision tree algorithms are defined as classification models, which behave on the idea of information gain at all nodes (Zhang and Ma 2012). In this procedure, all the variables (or features) are considered and various split points are tested using a pre-defined cost function. The split scoring the lowest cost is selected as the best model. Decision trees are very simple, straightforward, and easy to understand; however, they have relatively poor predictive power and are considered weak learners (Appel et al. 2013).

A RF represents a collection of decision trees, but it does not select all the data points and features in every single tree. The algorithm samples random data points and variables to shape multiple decision trees and then assembles their predicted outcome much like a committee voting—the outcome with the majority vote is put forth as the predicted outcome. A RF was selected over other ML algorithms because of its ability to reduce overfitting, overcome bias issues and achieve good predictive power. Also, a RF algorithm helps the user identify the most important features in a data set affecting the target variable through the application of robust feature selection (Zhang and Ma 2012).

Machine Learning in Construction Scheduling

Statistical inference techniques have been widely used in the literature to understand the drivers for construction schedule duration (Abu Hammad et al. 2010). ML applications have proven to outperform existing techniques, methods, and human decision making on construction sites (Hammad et al. 2014). Table 1 presents different studies that implement ML approaches in different construction management areas.

Among the studies included in Table 1, several investigated machine learning applications to reduce the uncertainty associated with task durations. At the macro level of project management, Hammad et al. (2014) used the expectation maximization clustering algorithm on data extracted from a real-world construction project to improve current resource management practices and as such reduce project delays. Furthermore, Soibelman and Kim (2002) employed different machine learning algorithms (decision tree and neural networks) to understand the major causes of delays in construction activities. Similarly, Asadi et al. (2015) presented a machine learning classification approach to predict whether a certain project will be delayed given a set of construction conditions (weather, poor communication, and equipment unavailability, among others) In a related vein, Lee et al. (2017) used the *k*-means clustering technique under a larger framework to retrieve similar projects and identify the schedule performance ratios associated

Table 1. Machine learning applications in construction management

References	Machine learning type	Algorithm	Construction management area	Application
Sakhakarmi et al. (2019)	Classification	Support vector machine	Safety	Scaffolding safety
Golparvar-Fard et al. (2013)	Classification	Support vector machine	Control and monitoring	Action recognition of earthmoving equipment
Tixier et al. (2016)	Classification	Random forest	Safety	Construction injury prediction
Williams and Gong (2014)	Classification	Neural networks	Cost	Predicting construction cost overruns
Ryu et al. (2019)	Classification	Multilayer perceptron	Control and monitoring	Worker action recognition
Iyer et al. (2012)	Classification	Neural networks	Inspection	Detection of defects in concrete pipes
Mahfouz et al. (2010)	Classification	Support vector machine	Organization	Document classification
Soibelman and Kim (2002)	Classification	Decision tree, neural networks	Scheduling	Schedule delays
Asadi et al. (2015)	Classification	Decision tree, Naïve Bayes	Scheduling	Schedule delays
Gondia et al. (2020)	Classification	Decision tree, Naïve Bayes	Scheduling	Schedule delays
Desai and Joshi (2010)	Regression	Decision trees	Control and monitoring	Labor productivity
Yip et al. (2014)	Regression	Neural networks	Cost	Maintenance cost of construction equipment
Dursun and Stoy (2016)	Regression	Neural networks	Cost	Estimation of construction costs
Ma and Wu (2019)	Regression	Neural networks	Scheduling control	Risk analysis
Cheng et al. (2019)	Regression	Neural networks	Scheduling	Estimate schedule to completion
Al Qady and Kandil (2014)	Clustering	Single pass	Organization	Document classification
Brilakis et al. (2005)	Clustering	<i>k</i> -means	Control and monitoring	Material recognition
Hammad et al. (2014)	Clustering	Expectation maximization	Scheduling	Optimal labor allocation
Lee et al. (2017)	Clustering	<i>k</i> -means	Scheduling control	Risk analysis
Wang et al. (2018)	Association	Equivalence class transformation	Safety	Workplace hazard identification
Cheng et al. (2015)	Association	A priori	Inspection	Defect analysis

with their case studies. More recently, Gondia et al (2020) used a set of project risk scores to predict the extent of overruns on a set of construction projects.

Turning to uses of machine learning for schedule updates in execution, Cheng et al. (2019) proposed a recurrent neural network based model to forecast reliable scheduling estimates for the project completion date at any time during construction. In another study, Ma and Wu (2019) evaluated construction-scheduling control by applying artificial neural network techniques. In both of these studies, however, the predictors fed into the neural network structure included project features (such as size and budget) as well as external factors (such as weather and labor market), none of which are controlled by the project manager. Thus, although these models provide insight, they cannot support decision making onsite.

In summary, most of the studies using machine learning approaches in construction scheduling predict the overall duration of the project schedule based on historical data from previous projects. The primary predictors are macroscopic-level project features or external factors beyond the control of the project manager. Activities onsite are generally overlooked despite playing a major role in determining the overall project duration. In this study, the focus is on field data generated on a continuous basis during the project's execution in order to predict and subsequently account for the uncertainty associated with quality assurance/quality control (QA/QC) activities (e.g., CPR inspection).

This paper offers a unique contribution because it presents a novel approach for contractors to leverage field data (e.g., acceptance/rejection decisions of submittals) to estimate the delays associated with the inherent uncertainty in the inspection decision outcome. A classification random forest algorithm is used to predict the outcome of an inspection request filed by the contractor. The algorithm is tested on data collected from an ongoing construction project. Statistical analysis is then applied to the data in order to fit probabilistic delay distributions to determine the extent of delay whenever the RF-algorithm-generated model predicts a rejected request. Finally, the obtained delay distributions are merged into the schedule to acquire an accurate updated schedule for the remaining part of the project.

Methodology

The methodology depicted in Fig. 1 includes four steps: (1) collect data regarding CPRs for an ongoing construction project, (2) implement a random forest algorithm on the collected data to develop a model to predict whether a CPR filed by the contractor will be rejected or accepted by the supervising engineer, (3) perform statistical analysis on the collected data to determine the time related impact of a rejected CPR, and finally (4) conduct a schedule simulation to accurately estimate the timeframe for the remaining portion of the project.

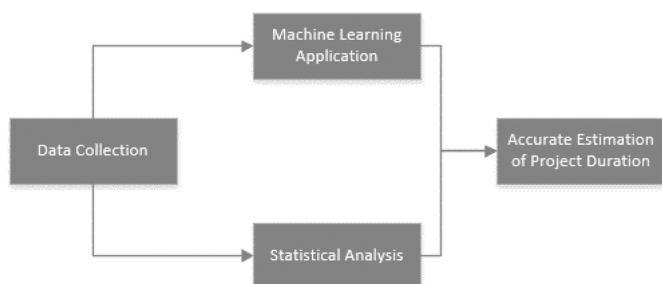


Fig. 1. Overall proposed methodology.

Data Collection

To illustrate the proposed methodology, CPR data were collected from an ongoing mixed-use multitower construction project in Beirut, Lebanon (Blocks A–F). The construction project follows a standard organization scheme with a main contractor, subcontractors, a contract administrating firm referred to as the engineer, and an owner representative. To keep track of the work progression, the contractor submits daily site reports to the engineer; these submittals include the CPRs.

The collected requests span over a duration of approximately 1.5 years and were acquired in Portable Document Format (PDF) files. The data set had more than 1,000 records, each of which includes the following information:

- Volume: volume of concrete to be poured (m^3).
- Duration: the pouring process expected duration (hours).
- Structure: the structural element to be cast [column, core wall, shear wall, slab, nonstructural wall (NSW), stairs, and others].
- Block: location of the pouring activity.
- Compressive strength of the concrete: 40 or 55 MPa.
- Date on which the contractor files a CPR to the engineer (request date).
- Proposed date by the contractor to start concrete pouring.
- Date of the engineer's response to the CPR.
- Engineer's response (accepted or rejected).

The data from the CPR were manually entered into a Microsoft Excel sheet, with the number of rows equal to the number of instances (1,001 rows), and a column for each of the aforementioned variables (nine columns). The overall number of CPRs recorded for the project was 2,148, but at the time the data were obtained (i.e. midway the project), the number of completed CPRs was 1,001. Additionally, columns were added to capture the time between two variables: (1) the proposed date of pouring and the request date (P-R duration), and (2) the checking date by the engineer and the proposed date of pouring (C-P duration).

Machine Learning Application

Fig. 2 presents a summary of the ML modeling framework adopted for this study. The framework consists of four major steps: (1) data cleaning, (2) feature selection, (3) data augmentation, and (4) algorithm tuning. The original data set includes 11 columns (categorical and continuous) and a binary target variable describing the outcome of a submitted CPR (rejected/accepted).

The first step is to collect, process, and clean the field data by removing outliers and missing data instances. Feature selection follows, with the aim of identifying the variables with the most prominent effect on the outcome of a submitted CPR. Then, the cleaned data set is divided into a training set and a testing set. Should the training data set be unbalanced, which is the case in this study, data augmentation is used to produce a balanced training data set. The resulting data set, along with the testing data set, are then used to tune a RF classification algorithm to reach the optimal classifier.

Data Cleaning

ML algorithms are typically very sensitive to the distribution of data points in the input data set. Outliers in the training and test data sets can result in less robust and less accurate ML algorithms. This can mislead or skew the results of the training procedure. In general, an outlier is defined as an observation that lies at an irregular distance from other points in a random sample from a population (Witten et al. 2016).

Outliers can be detected through graphical means such as boxplots or scatterplots. For continuous variables, outliers are the points that lie outside the $1.5 \times IQR$ margin, where IQR is defined

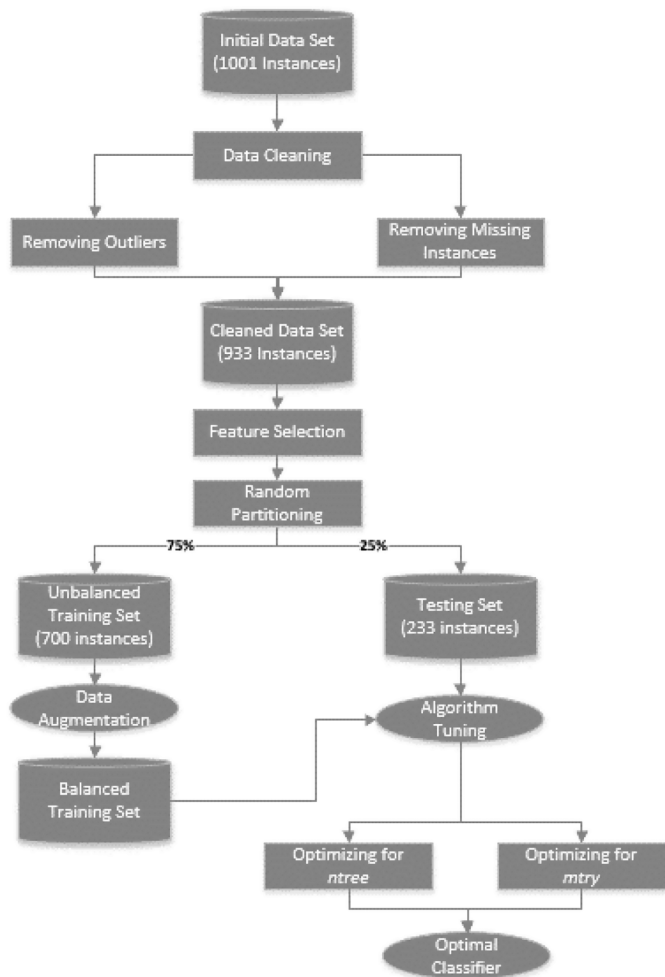


Fig. 2. Machine learning modeling framework.

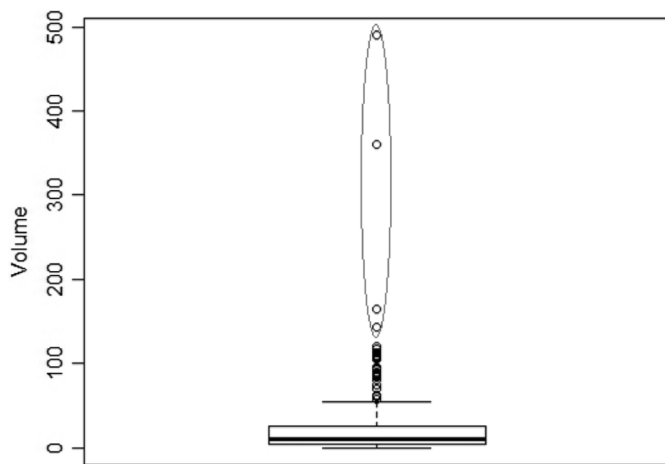


Fig. 3. Boxplot of the volume feature.

as the interquartile range, which is the difference between the 25th and 75th quartiles. The boxplot for the volume category is shown in Fig. 3, where the outliers are circled for emphasis.

Fig. 3 shows several outliers for the considered case study; however, due to the relatively small size of the data, values may be outlying not because these values are wrong but rather because the rest of the data points are compacted.

An exploratory data analysis was conducted revealing that 88% of the volume values were less than 50 m^3 , which explains the low values of the mean, 25th, and 75th quartiles. As for the remaining 12% of the data with volume values larger than 50 m^3 , the analysis showed that 93% of these values were associated with slabs. Slabs are major structural elements that cannot be overlooked and as such were not removed from the analysis even though they were considered as outliers based on the boxplot. The extreme circled points shown in the plot were removed because they are the only four instances representing a foundation and no conclusion can be drawn from four instances alone.

In addition, 64 data points were removed from the data set because of major missing entries. The final count of instances is equal to 933. Thus, the final data set was composed of 72 rejected CPRs and 861 accepted CPRs.

Feature Selection

The second step of the analytics process is the identification and selection of features that are most significant in terms of their influence on the acceptance/rejection of CPRs. The process of feature selection is vital for supervised ML problems. It is the process by which a data analyst directs ML systems toward a determined target. Feature selection can dramatically reduce costs and data volume (curse of dimensionality) by eliminating the least relevant features in a data set. Furthermore, building a model based on the most precise feature set can help reduce the problem of overfitting (Domingos 2012).

A common technique for feature selection presented by the caret R version 3.5.1 package is called recursive feature elimination (Kuhn 2008). A recursive feature elimination algorithm serves to evaluate the model and to explore the available feature subsets. The pool of features submitted to the recursive feature elimination algorithm includes volume, duration, structure, block, proposed pouring time, concrete compressive strength, the difference between the contractor's proposed date for pouring and the request submission date, and finally the difference between the engineer's response date and the contractor's proposed date for pouring. The feature selection process revealed that the most relevant features are the difference between the engineer's response date and the contractor's proposed date for pouring (C-P duration), volume of concrete to be poured, type of the structure to be poured, block or location of pouring, and concrete compressive strength.

The volume, type, location, and compressive strength of the concrete to be poured make sense as predictors because a larger volume will require a greater area to be inspected, thereby raising the likelihood of defects. The type will influence the underlying structures that must be inspected and thus alter the likelihood of defects. The location will influence the ease of access and underlying flaws as a result of differential site conditions. Finally, the compressive strength will impact the complexity of the pour, thus influencing the inspection process of the site elements in the location of the pour.

At first glance, the inclusion of the C-P duration may be surprising. However, this item is at the heart of how important it is to capture microlevel processes in schedule forecasting. On many sites, it is the practice of the site manager to request an inspection prior to the completion of the work with the anticipation that by the time the inspection comes, the preceding work will be complete. The effect of this practice is captured in this variable whereby inspections that occur longer in advance of the requested pour date tend to meet rejection. Thus, this variable reflects situations where the site manager has requested an inspection expecting that it will occur closer to the requested pour date, but is then caught unprepared when the inspection occurs earlier than anticipated. This variable

Instance	Block
1	Block A
2	Block F
3	Block C
4	Block A
5	Others

Instance	Block A	Block C	Block F	Others
1	1	0	0	0
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0
5	0	0	0	1

Fig. 4. One-hot encoding example.

also represents a point of control for the site manager when filing the request.

Therefore, the ML classification algorithm was built based on these five features. The first two features are continuous variables, whereas the last three features are categorical and require processing before developing the model. The Type of Structure variable has seven categories (NSW, core wall, shear wall, stairs, slab, column, and others), and the block feature contains four categories (Block A, Block C, Block F, and other blocks). The category Others was introduced into these variables because of the unequal distribution of the data. Blocks B, D, and E did not include enough instances to form a general hypothesis about them; therefore, these categories were merged into a combined category termed Others.

In general, ML models do not deal directly with categorical data. It is thus necessary to convert the categorical variables into numerical data types. One-hot encoding is considered the standard approach to complete this task and it works well especially if the categorical values are not that many. The one-hot encoding method generates new binary columns, signifying the presence of each possible value from the original data values (Bruce and Bruce 2017). By implementing this method, the number of features increased from 5 to 14 (two continuous variables, seven variables to represent the structural elements, four variables to represent the pouring location, and a single binary variable to represent the compressive strength of concrete). Fig. 4 illustrates an example of the one-hot encoding applied over the categorical variable Block.

The authors do not claim a causal relationship between the proposed feature space and the acceptance/rejection of a CPR. However, because CPRs represent a tangible data artefact from the project, they represent an important source of project information. The proposed ML framework seeks to learn the hidden target function governing the relation between the stated CPR input and the output (rejection/acceptance).

Data Augmentation

The RF algorithm is used to predict whether an inspection request filed by the contractor will be accepted or rejected by the engineer. The target variable is the final response of the engineer indicating whether a given CPR will be accepted or rejected. The data set was split randomly into a training set (75%) and a testing set (25%).

Upon examination, the training set was unbalanced. Unbalanced data refers to a problem with classification ML models, where the values of the target variable are not represented equally. For example, a two-class (A and B) binary classification with 100 data points may be susceptible to the problem of unbalanced data if a total of 85 points are labeled A and the remaining points are labeled B. As a result, a model built on this training set will always favor the larger class, neglect the minority, and achieve an 85% accuracy rate.

The training set included 646 (92%) accepted instances and 54 (8%) rejected instances. After training the model using RF and testing it over the test set, the results emerging from the model revealed a high accuracy of 93.3%, a perfect precision of 100% but a very low recall of 11.1%. Table 2 presents the confusion matrix corresponding to the trained classifier. A confusion matrix is used

Table 2. Initial confusion matrix

	Actually rejected	Actually accepted
Predicted as rejected	TP = 2	FP = 0
Predicted as accepted	FN = 16	TN = 215

to describe the performance of a classification algorithm on a data set for which the true values are known. True positives (TP) and true negatives (TN) are observations correctly predicted, whereas false negatives (FN) and false positives (FP) are observations predicted incorrectly. Naturally, a robust classifier minimizes the FN and FP but maximizes the TP and TN.

Accuracy is defined as the number of all correct predictions (TP and TN) divided by the total number of observations in the data set. It is not a reliable measure for the actual performance of an algorithm because it is misleading when the data set is unbalanced and the model predicts the dominant class ignoring the minority, resulting in a paradoxical high accuracy. On the other hand, the precision and recall metrics are very useful measures of the success of classification algorithms when dealing with unbalanced classes. Precision is defined as the fraction of correct positive predictions (TP) to the total positive predicted observations (TP and FP). Recall is calculated as the ratio of correct positive predictions (TP) over the TP and FN observations (Yang and Liu 1999).

To overcome the problem of unbalanced classes, an oversampling technique was applied to generate new synthetic samples in the minority classes. Several systematic algorithms can be used to create these synthetic data points. One of the most common algorithms is the synthetic minority oversampling technique (SMOTE) algorithm. The SMOTE algorithm generates pseudoinstances in the neighborhoods of the minority group. It takes each data point in the minority class and finds its corresponding k -nearest minority samples. Then, the algorithm randomly selects i of these nearest neighbors and creates synthetic data points over the lines connecting the minority instance and its i neighbors (Chawla et al. 2002).

In general, the SMOTE algorithm's synthetic data points are a linear combination of two existing instances from the minority class (X and Y) (Chawla et al. 2002). Mathematically, a synthetic sample (Z) is represented as follows:

$$Z = X + u \times (Y - X) \quad (1)$$

where u = real number between 0 and 1; X is selected randomly; and Y is chosen randomly from the k - nearest neighbors minority instances of X .

After applying the SMOTE algorithm over the training set, a balanced data set was created. The augmented training set contained 50% accepted data points, 45% synthetic rejected instances, and 5% real rejected samples.

Algorithm Tuning

In the field of ML, hyperparameter tuning is the process of selecting the optimal hyperparameters for a given learning algorithm that maximizes its performance, where performance is defined as the success of the resulting model in predicting the correct results (Bergstra and Bengio 2012). A hyperparameter is a factor characterizing an algorithm and whose value is defined before the learning process starts. There are several hyperparameters corresponding to the RF learning algorithm, but in this study only two are taken into consideration because they have the largest effect on the results (Brownlee 2016):

- *n*tree: the number of decision trees to grow within the RF algorithm. The higher the number of trees, the more computationally

Table 3. Baseline confusion matrix

	Actually rejected	Actually accepted
Predicted as rejected	TP = 5	FP = 20
Predicted as accepted	FN = 13	TN = 195

expensive it is to build the RF classifier. The default value of *n*tree in the RF implementation within R is 500.

- *m*try: the number of variables randomly selected at a node split in a certain decision tree. The default value of *m*try in R is the square root of the total number of features under study. In this case, $mtry = \sqrt{14} \approx 4$.

When tuning a learning algorithm, it is vital to build a baseline for comparison and evaluation by using the default values for each parameter. Table 3 presents the baseline confusion matrix after building the RF model over the augmented training data set with the default parameters' values.

The following results suggest that the RF classifier was able to overcome the problem of class unbalance after the synthetic creation of instances in the minority class:

$$\begin{aligned} \text{Baseline accuracy} &= \frac{TP + TN}{TP + TN + FN + FP} \\ &= \frac{5 + 195}{5 + 13 + 195 + 20} \times 100 = 85.8\% \quad (2) \end{aligned}$$

$$\text{Baseline precision} = \frac{TP}{TP + FP} = \frac{5}{5 + 20} \times 100 = 25.0\% \quad (3)$$

$$\text{Baseline recall} = \frac{TP}{TP + FN} = \frac{5}{5 + 13} \times 100 = 27.8\% \quad (4)$$

To determine the optimal values of the hyperparameters, a grid search technique was implemented. Grid search or parameter sweep is a common methodology to shape and evaluate a classifier for every combination of algorithm parameter indicated in a manually predefined grid (Bergstra and Bengio 2012).

The grid search analysis yields the values of accuracy, recall and precision for every combination of *n*tree and *m*try. The *n*tree grid contains values between 1 and 500 inclusive, whereas the *m*try grid contains values between 4 and 14 inclusive. The optimal combination proposed by the grid search was to train the RF classifier including 11 trees and 14 features. The accuracy was around 91%, the recall value was equal to 61%, and the precision was approximately 44%. Table 4 presents the confusion matrix of this classifier.

The optimal classifier showed a drastic improvement in performance compared with the baseline classifier. The accuracy increased by 5.2%, the precision metric improved by 19.0%, and the recall witnessed the highest enhancement by 33.3%. Some classifiers witnessed higher accuracies whereas others had higher precision; however, the model with highest recall was chosen for two main reasons:

- Recall is defined as the ability of a trained learning algorithm to detect all relevant cases within a given data set. In this context, rejections are the most relevant cases. Contractors onsite are interested in predicting rejected CPRs, which would help them

Table 4. Optimal classifier's confusion matrix

	Actually rejected	Actually accepted
Predicted as rejected	TP = 11	FP = 14
Predicted as accepted	FN = 7	TN = 201

Table 5. Comparative results between the optimal and baseline classifiers

	<i>n</i> tree	<i>m</i> try	Accuracy (%)	Precision (%)	Recall (%)	<i>F</i> ₁ score (%)
Baseline classifier	500	4	85.8	25	27.8	26.3
Optimal classifier	11	14	91.1	44	61.1	51

work proactively to prevent the likelihood of rejection or accommodate it in the schedule. Thus, the higher the recall value, the better the model is from the contractor's point of view.

- The *F*₁ score is another metric used by data analysts to capture the trade-off between precision and recall when training a learning algorithm (Yang and Liu 1999). The *F*₁ score is the harmonic mean of precision and recall. The following equation presents the mathematical computation of the *F*₁ score:

$$F_1 = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \quad (5)$$

The *F*₁ score combines both metrics into one measure and gives equal weight to both precision and recall. To create a balanced classifier, a data analyst should aim to maximize the *F*₁ score of the built model. In the considered case study, the trained RF classifier reached a maximum *F*₁ score of 0.51 when *n*tree and *m*try were equal to 11 and 14, respectively. An *m*try value of 14 may seem high; however, in light of the fact that a feature selection algorithm was previously applied, it is expected that all variables will have consistently significant predictive power.

To summarize, Table 5 presents the overall comparative results between the optimal and the baseline classifiers in terms of *n*tree, *m*try, accuracy, precision, recall, and *F*₁ score.

Statistical Analysis

In parallel to building the prediction model via machine learning, a statistical analysis was conducted on the collected field data to determine the timeline associated with CPRs on this site. Whenever a CPR is submitted, the time spent is equal to the time elapsed between the submission of the request and the engineer's check date. More specifically, this includes two time spans: the duration between the contractor's pouring request submission date and the proposed date for pouring (P-R duration), and the duration between the proposed date for pouring and the engineer's check (C-P duration).

Figs. 5(a-f) present the probability distribution for both P-R and C-P durations for slab, column, and NSW CPR submittals in one of the blocks of the considered case study. These probability distributions change with different block locations. The durations of P-R and C-P for the remainder of the project follow the built probabilistic distributions.

If a CPR is rejected, the process of submitting the CPR is repeated as in Fig. 6. However, the timing of the request, inspection and pour events are altered.

In a typical scheduling scenario, contractors build a buffer to account for the chance of delay due to rejection of submittals. They typically allocate a deterministic buffer time span based on personal experience (Park and Peña-Mora 2004). The methodology proposed in this paper attempts to investigate these delays based on data acquired during construction and replaces the traditional deterministic buffers with probabilistic distributions. In total, there were 72 rejected CPRs, but only 30 revised requests (resubmitted after rejection) were used to obtain the probability distributions (the remaining 42 were missing in the data set). Given that the revised data points are few (only 30), the probability distributions built

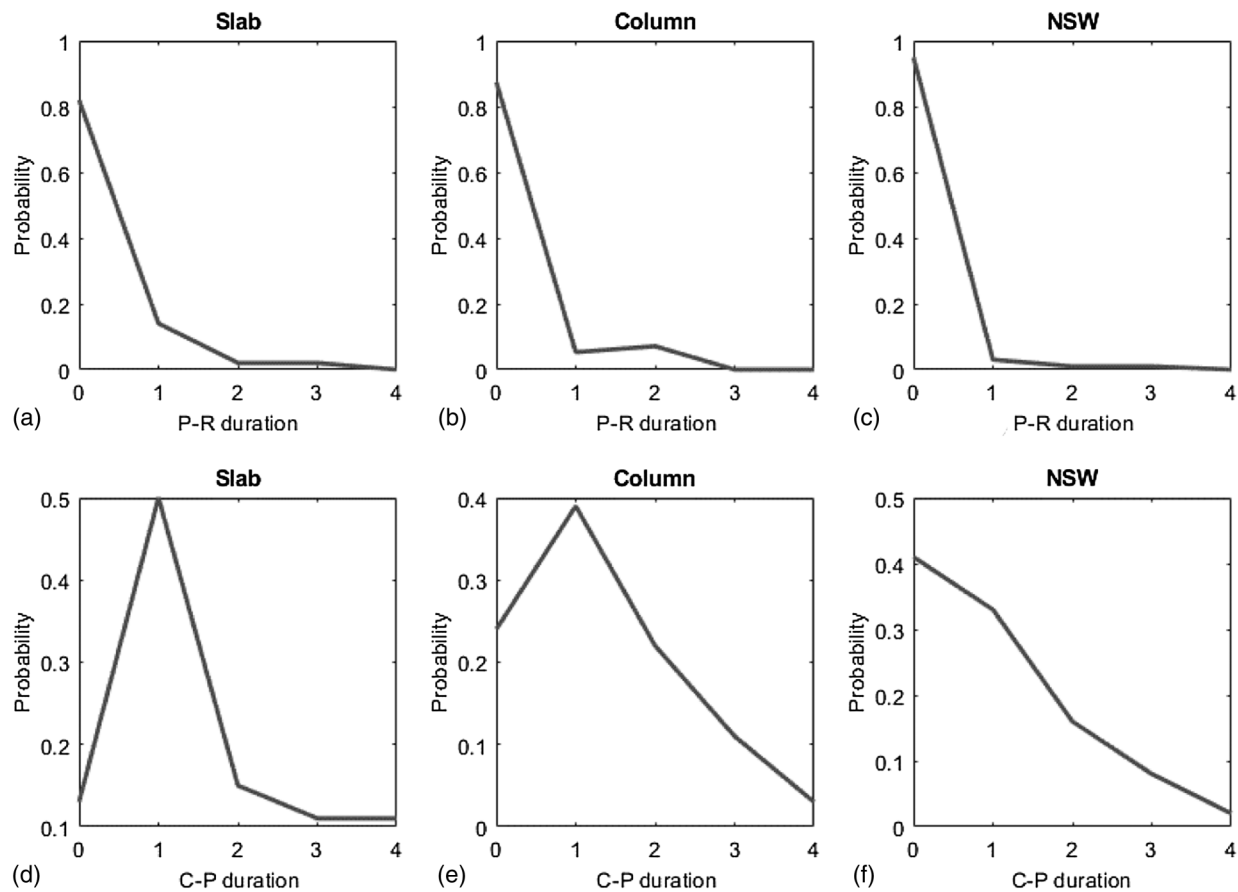


Fig. 5. Probability distribution for (a–c) R-P of the slab, column, and NSW, respectively; and (d–f) C-P of the slab, column, and NSW, respectively.

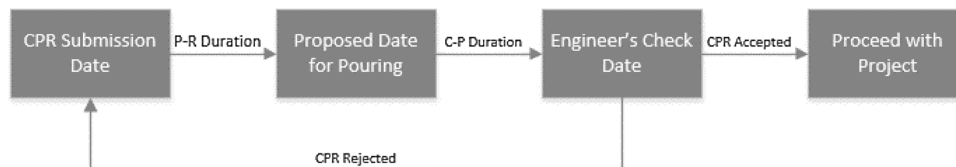


Fig. 6. Schedule flow under different CPR decisions.

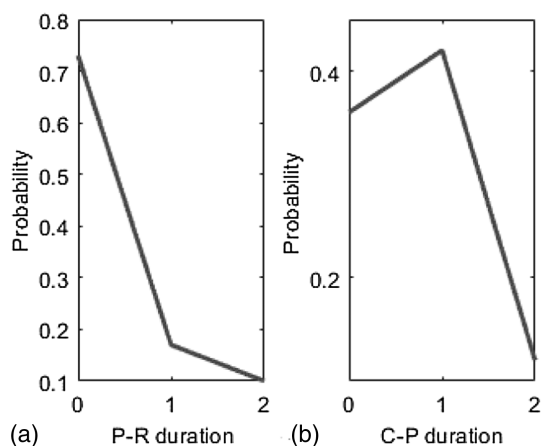


Fig. 7. (a) P-R; and (b) C-P durations of revised CPRs.

in Fig. 7 were not categorized by the type of structure or the location of concrete pouring. Based on these data, the chance of getting another rejection after submitting a revised CPR is only 3%; thus, the simulation assumes that all revised CPRs are directly accepted.

Method Validation

The last step of the proposed methodology is to validate its use by applying the results of the built RF model and probability distributions to forecast the duration of a segment of work on the case-study site. The proposed methodology is illustrated in this section for the case of an electrical room located in Block A of the considered case study. The room, which is used for storing electrical supplies, has a footprint area of 10 × 10 m and a ceiling height of 3 m. The room contains two columns with radii equal to 0.2 m each. The slab thickness for the floor and the ceiling of the room is equal

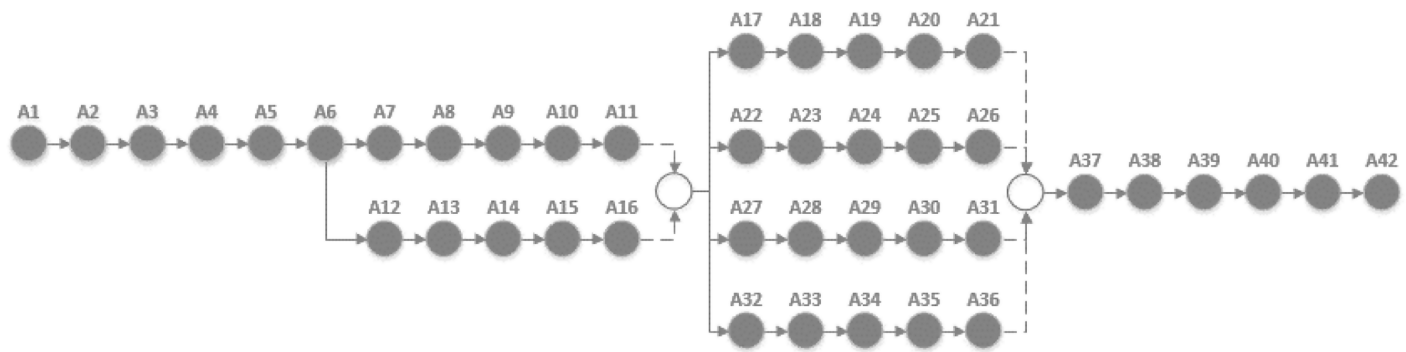


Fig. 8. Construction schedule using CPM.

to 0.1 m. Two schedules were built for the construction of the electrical room: one using the traditional CPM method and one using the proposed methodology.

CPM Schedule with Built-In Deterministic Buffer

Fig. 8 shows the construction schedule using CPM. Table 6 presents each activity's task ID along with its allocated duration. The reported values came from the site engineers.

As indicated in Table 6, concrete pouring will be performed on top and bottom slabs, two circular columns, and four NSWs. For each concrete pouring activity, the contractor files a CPR for the engineer's approval. As illustrated in the table, the contractor assigns a duration of 2 days for the approval of slab related CPRs, and 1 day for the NSW- and-column related CPRs. These durations are considered static buffers added by the contractor to accommodate for unexpected rejections of these CPRs. The buffer added by the contractor varies with the type of structure: a slab is given 2 buffer days because it has more design specifications, but a column or a NSW is given only 1 buffer day. Thus, if a slab related CPR is rejected, more time is required to resolve the problem and file a new request than if a column- or NSW-related CPR is rejected. By applying the CPM, the project duration was estimated to be 34.3 days.

Schedule Using Proposed ML-Based Probabilistic Method

The proposed method was adopted to derive a schedule for the scope of the same electrical room using the RF model and probability distributions and fit in the "Methodology" section.

To apply the proposed method, two adjustments to the activity task listing were necessary:

- As opposed to having a single node for CPR approval, two nodes are placed instead: the first one is the P-R duration, and the second one is the C-P duration, as shown in Fig. 9. For example, consider Activity A5 in the initial schedule (Fig. 8). In the revised schedule, A5 is divided into activities A5' and A5'' (Fig. 9). Activity A5' corresponds to the P-R duration, whereas Activity A5'' represents the C-P duration.
- As shown in Fig. 8, if a CPR is rejected, the process of submitting a new CPR is required; this is indicated by the backflowing arrows labeled Rejection.

A Monte Carlo simulation was conducted on the built schedule. A simulation was selected to apply the RF model and duration estimates because applying the model in practice was not possible. However, if one were to use the proposed strategy onsite, the RF model would be applied at each moment of request submission—in this way the site manager filing the request would know how likely

it is that the request would be rejected. In practice, the manager could then alter the timing of the request to influence the likelihood of acceptance. In this simulation, it was assumed that the manager does not intervene and the project proceeds as modeled incurring the inspection related delays, if any. Furthermore, by running the

Table 6. Activity task ID and duration

Task ID	Activity	Duration (days)
A1	Bottom slab steel	2
A2	Electrical and mechanical	2
A3	Bottom slab formwork	1
A4	Order concrete for bottom slab	2
A5	Approve CPR-bottom slab	2
A6	Pour concrete-bottom slab	0.5
A7	Round column1 steel	2
A8	Round column1 formwork	1
A9	Order concrete for column1	2
A10	Approve CPR-column1	1
A11	Pour concrete-column1	0.1
A12	Round column2 steel	2
A13	Round column2 formwork	1
A14	Order concrete for column2	2
A15	Approve CPR-column2	1
A16	Pour concrete-column2	0.1
A17	NSW1 block work	3
A18	NSW1 embedded installation	2
A19	Order concrete for NSW1	2
A20	Approve CPR-NSW1	1
A21	Pour concrete-NSW1	0.1
A22	NSW2 block work	3
A23	NSW2 embedded installation	2
A24	Order concrete for NSW2	2
A25	Approve CPR-NSW2	1
A26	Pour concrete-NSW2	0.2
A27	NSW3 block work	3
A28	NSW3 embedded installation	2
A29	Order concrete for NSW3	2
A30	Approve CPR-NSW3	1
A31	Pour concrete-NSW3	0.1
A32	NSW4 block work	3
A33	NSW4 embedded installation	2
A34	Order concrete NSW4	2
A35	Approve CPR NSW	1
A36	Pour concrete NSW4	0.2
A37	Top slab steel	2
A38	Electrical and mechanical	2
A39	Top slab formwork	2
A40	Order concrete for top slab	2
A41	Approve CPR top slab	2
A42	Pour concrete top slab	0.5

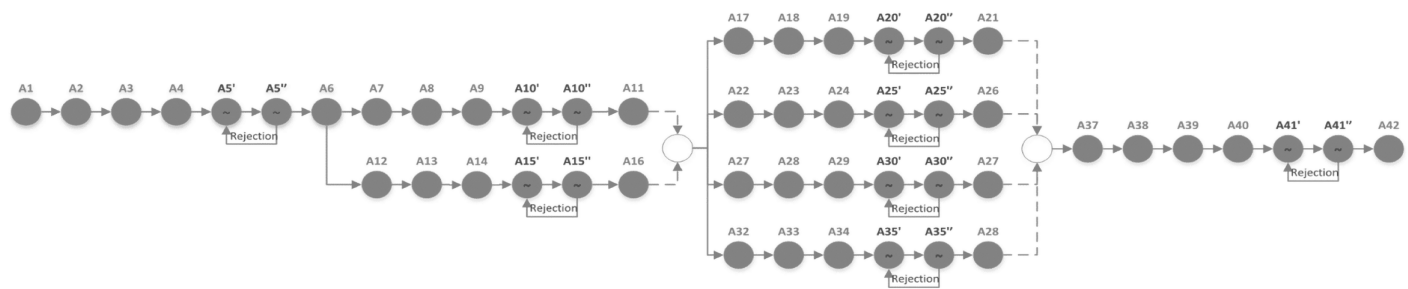


Fig. 9. Construction schedule using the proposed methodology.

Table 7. Prediction made by the ML algorithm based on input data

Block	Structure	Input			Output
		Volume (m ³)	Compressive strength (MPa)	Duration between proposed pouring date and engineer's check (days)	
A	Slab	10	55	0	Accepted
A	Slab	10	55	1	Accepted
A	Slab	10	55	2	Accepted
A	Slab	10	55	3	Rejected
A	Slab	10	55	4	Rejected
A	Column	1	55	0	Accepted
A	Column	1	55	1	Accepted
A	Column	1	55	2	Accepted
A	Column	1	55	3	Rejected
A	Column	1	55	4	Rejected
A	NSW	1	40	0	Rejected
A	NSW	1	40	1	Accepted
A	NSW	1	40	2	Accepted
A	NSW	1	40	3	Accepted
A	NSW	1	40	4	Accepted
A	NSW	2	40	0	Accepted
A	NSW	2	40	1	Accepted
A	NSW	2	40	2	Accepted
A	NSW	2	40	3	Accepted
A	NSW	2	40	4	Accepted

simulation with no assumed intervention, the results of the CPM and the proposed method can be more fairly compared.

The schedule starts with Activity A1 until it reaches Activity A5', which is the CPR for a slab, at which point the simulation determines the duration based on Fig. 7(a) (P-R). Then, the simulation proceeds to Activity A5'', where it similarly determines the duration based on Fig. 7(b) (C-P). At that point, the ML algorithm predicts whether the CPR is accepted based on Table 7, which presents the prediction made by the RF algorithm based on the input data. For example, for the slab pouring requests in Block A (10 m³ in volume and 55 MPa compressive strength), when the duration between the proposed pouring date and the engineer's check is between 0 and 2 days, the request is accepted. However, when this duration exceeds 2 days, the request is rejected.

Once the acceptance/rejection prediction is made as per Table 7, two possible scenarios arise:

- The request is rejected, leading to a resubmission of the CPR. The simulation goes back to Activity A5', and then the whole process is repeated based on new probability distributions of P-R and C-P, as shown in Fig. 7.
- The request is accepted, and the simulation proceeds with the proposed schedule.

After obtaining acceptance of a CPR, the simulation proceeds in a similar manner for all CPRs in the schedule until it reaches the last activity. Finally, the overall project duration is computed for a single simulation.

The Monte Carlo simulation was conducted 5,000 times, and a lognormal distribution was fitted as shown in Fig. 10. The Monte Carlo simulation resulted in an average project duration of 37.2 days with a standard deviation of 3.2 days. This duration is approximately 10% higher than the duration obtained using the CPM-based schedule with a deterministic buffer. This result indicates that the deterministic buffer was optimistic or insufficient to account for the uncertainty inherent in the outcome of CPR inspections. This is not a surprising result. Most construction projects end up overrunning their target or baseline schedules, even when float is built into the baseline schedule. Based on this finding, the methodology outlined here shows significant promise for the use of ML techniques on construction sites. Specifically, by relating microlevel site data to schedule effects, the project manager can gain control over the schedule by (1) consciously selecting the concrete pour date—and understanding the impact of that selection when filing requests, and (2) keep all parties informed about potential schedule overruns during execution.

Conclusions

One of the most important implications of this study is the significance of machine learning to synthesize unique site features and procedures through their data artefacts into a forecast of inspection request outcomes and their impact on project schedule.

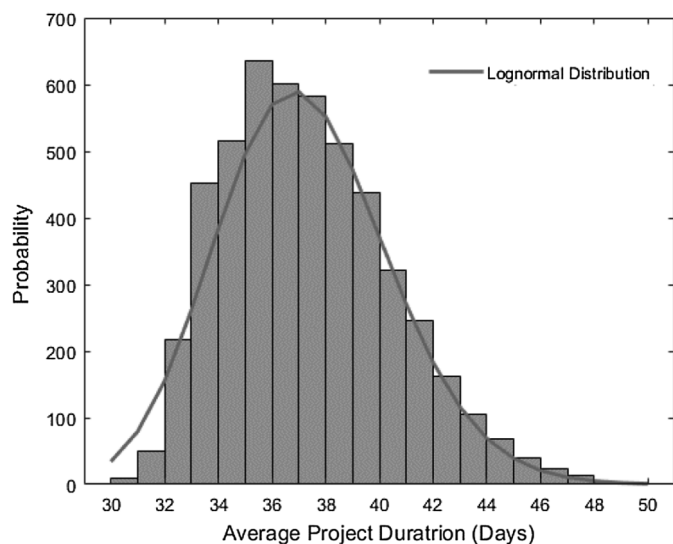


Fig. 10. Lognormal distribution for the Monte Carlo simulation 25.

Specifically, this paper illustrates how a RF learning algorithm can be employed to produce a method for forecasting CPR decisions and enhance the accuracy in determining project durations.

The proposed methodology was applied on a real-world case study. The applied RF algorithm presented a 91% accuracy in predicting the CPR decisions, along with a recall and precision of 61% and 44%, respectively. The project schedule obtained using the proposed methodology is probabilistic and has an average duration of 37.2 days as opposed to 34.3 days obtained from a deterministic CPM schedule.

Managerial Implications

With regards to this particular application of machine learning to microlevel site data, two managerial implications emerge. The first implication is that by collecting and modeling the relationship among the time of request filing, the time of inspection, the requested pour date, and the success of the request, the project manager can gain insight into how site practices of gaming the inspection process influence rejection rates and the project schedule. This insight is sorely needed as mentioned by Pestana et al (2014), who stated that “that durations used to manage the submittal process are unreliable and often do not match what is indicated in schedules and contractual requirements.”

To elaborate on this site phenomenon, Table 8 indicates that the contractor may decide to submit a CPR upon the completion of work (e.g., after finishing the steel reinforcement diameter/spacing and coating), thereby reducing the chances that the CPR will be rejected by the supervising engineer. However, this outcome comes at the expense of schedule performance—there will be at least 1 day between the filing of the request and the execution of the inspection. To speed up the process, contractors often decide to submit their CPRs several days prior to the completion of field work with

the aim of having the remaining work completed by the time the supervising engineer visits the site. The trade-off in this case is an increase in the chance of rejection, resulting in delays due to rework and submittal of a revised CPR with corrective work. Researchers have observed that the late approval and rejection of submittals is a major factor of delay (Ren et al. 2008; Ko and Li 2014). By predicting the acceptance/rejection of submittals and associating any rejection with the appropriate schedule delay, the proposed ML model can be used by site engineers and managers to inform their decisions on whether, and when, to submit their CPRs with respect to the expected time of work completion.

However, the proposed ML model is built on past site data and thus will inherently include the past behavior of contractors relative to request filing. For this reason, it is important to refresh the model throughout the project because the use of the model will invariably impact request filing behavior. This, in turn, points to the second managerial implication of this work. The contractor’s project manager, with the help of project control personnel, can use the proposed ML model to update the project’s schedule in real-time.

With the submission of every new submittal at any stage in construction, the model can be retrained, and predictions for future submissions are updated along with the projected delays. Delays and discrepancies between actual and planned project schedules are among the most prevalent problems on construction sites (Bagaya and Song 2016). Although the proposed methodology does not cover all the problems that can affect the project schedule, it sets a roadmap for construction professionals to invest in data generated from their construction projects and employ it to improve the prediction of construction delays.

Limitations and Future Research

The proposed methodology was validated theoretically; therefore, the generalizability of the findings to other construction sites should be validated through use in the field. In addition, ML models should never be used in isolation of experts’ judgment; the RF model did not present perfect accuracy, precision, and recall, meaning that some CPR decisions may be predicted wrongly. Another limitation of this research work is that the data-entry process (conversion to Excel from PDF files) was manual. In order to fully leverage construction site data streams, the use of automation with optical character recognition or the use of tablets onsite is highly recommended.

This research work could also be expanded to cover the causes behind the rejection of inspection requests on a construction project. When rejecting a request, the engineer writes down the cause behind the decision and delivers it to the contractor to fix the problem when submitting the revised CPR. If enough instances are collected, a statistical hypothesis could be built regarding the major causes of rejections and their probability of occurrence. The contractor can use this analysis proactively to reduce the likelihood of rejection by making sure that the major causes of rejections are dealt with before the engineer’s check date.

The method of mining site data and building predictive models for microlevel site processes can also be generalized to other critical

Table 8. Outcomes and schedule impacts associated with inspection request submittals

Time of inspection request submittal	Potential outcomes	Impact on schedule
At the completion of work	Rejection (low probability)	Major delay due to rework
	Acceptance (high probability)	Delay due to time lost between the filing of request and arrival of inspector
Prior to the completion of work	Rejection (high probability)	Delay due to rework or need to complete work
	Acceptance (low probability)	No delay

areas of project management. For example, by leveraging site data on material deliveries and equipment reliability, estimates of project schedule duration could be further enhanced. Similarly, the same methodology proposed in this paper could be applied to predict budget overruns based on field data. Forecasts of risk could be made on the basis of data collected from wearable devices.

Data Availability Statement

Some data, models, or code generated or used during the study are available from the corresponding author by request. These include the aggregate level anonymized summaries of the CPR data set along with the R code used in machine learning.

Acknowledgments

The authors would like to thank the American University of Beirut's University Research Board for funding this study (Project No. 24709 and Award No. 103604). The authors gratefully acknowledge the data entry work of Ahmad Mikdach and Ali Badra under the support of the Lebanese American University's Adnan Kassar School of Business Graduate Assistantship Program. The authors also thank four anonymous reviewers whose effort served to improve this paper.

References

Abu Hammad, A. A., S. M. A. Ali, G. J. Sweis, and R. J. Sweis. 2010. "Statistical analysis on the cost and duration of public building projects." *J. Manage. Eng.* 26 (2): 105–112. [https://doi.org/10.1061/\(ASCE\)0742-597X\(2010\)26:2\(105\)](https://doi.org/10.1061/(ASCE)0742-597X(2010)26:2(105)).

Alpaydin, E. 2009. *Introduction to machine learning*. Cambridge, MA: MIT Press.

Al Qady, M., and A. Kandil. 2014. "Automatic clustering of construction project documents based on textual similarity." *Autom. Constr.* 42 (Jun): 36–49. <https://doi.org/10.1016/j.autcon.2014.02.006>.

Alves, T., and I. D. Tommelein. 2004. "Simulation of buffering and batching practices in the interface detailing-fabrication-installation of HVAC ductwork." In *Proc., of IGLC-12*. Elsinore, Denmark: International Group for Lean Construction.

Ansar, A., B. Flyvbjerg, A. Budzier, and D. Lunn. 2016. "Does infrastructure investment lead to economic growth or economic fragility? Evidence from China." *Oxford Rev. Econ. Policy* 32 (3): 360–390. <https://doi.org/10.1093/oxrep/grw022>.

Appel, R., T. Fuchs, P. Dollár, and P. Perona. 2013. "Quickly boosting decision trees—pruning underachieving features early." In *Proc., Int. Conf. on Machine Learning*, 594–602. Brookline, MA: Journal of Machine Learning Research.

Asadi, A., M. Alsubaey, and C. Makatsoris. 2015. "A machine learning approach for predicting delays in construction logistics." *Int. J. Adv. Logist.* 4 (2): 115–130. <https://doi.org/10.1080/2287108X.2015.1059920>.

Bagaya, O., and J. Song. 2016. "Empirical study of factors influencing schedule delays of public construction projects in Burkina Faso." *J. Manage. Eng.* 32 (5): 05016014. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000443](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000443).

Balta, S., M. T. Birgonul, and I. Dikmen. 2018. "Buffer sizing model incorporating fuzzy risk assessment: Case study on concrete gravity dam and hydroelectric power plant projects." *ASCE-ASME J. Risk Uncertainty Eng. Syst., Part A: Civ. Eng.* 4 (1): 04017039. <https://doi.org/10.1061/AJRUA6.0000948>.

Bergstra, J., and Y. Bengio. 2012. "Random search for hyper-parameter optimization." *J. Mach. Learn. Res.* 13 (1): 281–305.

Bilal, M., L. O. Oyedele, J. Qadir, K. Munir, S. O. Ajayi, O. O. Akinade, and M. Pasha. 2016. "Big data in the construction industry: A review of

present status, opportunities, and future trends." *Adv. Eng. Inf.* 30 (3): 500–521. <https://doi.org/10.1016/j.aei.2016.07.001>.

Brilakis, I., L. Soibelman, and Y. Shinagawa. 2005. "Material-based construction site image retrieval." *J. Comput. Civ. Eng.* 19 (4): 341–355. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2005\)19:4\(341\)](https://doi.org/10.1061/(ASCE)0887-3801(2005)19:4(341)).

Brownlee, J. 2016. "R machine learning." Accessed October 1, 2020. <https://machinelearningmastery.com/tune-machine-learning-algorithms-in-r/>.

Bruce, P., and A. Bruce. 2017. *Practical statistics for data scientists: 50 essential concepts*. Newton, MA: O'Reilly Media.

Burkov, A. 2019. Vol. 1 of *The hundred-page machine learning book*. Quebec: Andriy Burkov.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. "SMOTE: Synthetic minority over-sampling technique." *J. Artif. Intell. Res.* 16: 321–357. <https://doi.org/10.1613/jair.953>.

Cheng, M. Y., Y. H. Chang, and D. Korir. 2019. "Novel approach to estimating schedule to completion in construction projects using sequence and nonsequence learning." *J. Constr. Eng. Manage.* 145 (11): 04019072. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001697](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001697).

Cheng, Y., W.-D. Yu, and Q. Li. 2015. "GA-based multi-level association rule mining approach for defect analysis in the construction industry." *Autom. Constr.* 51 (Mar): 78–91. <https://doi.org/10.1016/j.autcon.2014.12.016>.

Cho, D., J. S. Russell, and J. Choi. 2013. "Database framework for cost, schedule, and performance data integration." *J. Comput. Civ. Eng.* 27 (6): 719–731. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000241](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000241).

Dawood, N. 1998. "Estimating project and activity duration: A risk management approach using network analysis." *Constr. Manage. Econ.* 16 (1): 41–48. <https://doi.org/10.1080/014461998372574>.

De Andrade, P. A., M. Annelies, and V. Mario. 2019. "Using real project schedule data to compare earned schedule and earned duration management project time forecasting capabilities." *Autom. Constr.* 99 (Mar): 68–78. <https://doi.org/10.1016/j.autcon.2018.11.030>.

Desai, V. S., and S. Joshi. 2010. "Application of decision tree technique to analyze construction project data." In *Proc., Int. Conf. on Information Systems, Technology and Management*, 304–313. Berlin: Springer.

Domingos, P. 2012. "A few useful things to know about machine learning." *Commun. ACM* 55 (10): 78–87. <https://doi.org/10.1145/2347736.2347755>.

Durdyev, S., M. Omarov, and S. Ismail. 2017. "Causes of delay in residential construction projects in Cambodia." *Cogent Eng.* 4 (1): 1291117. <https://doi.org/10.1080/23311916.2017.1291117>.

Dursun, O., and C. Stoy. 2016. "Conceptual estimation of construction costs using the multistep ahead approach." *J. Constr. Eng. Manage.* 142 (9): 04016038. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001150](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001150).

Esmaili, B., and H. Matthew. 2013. "Integration of safety risk data with highway construction schedules." *Constr. Manage. Econ.* 31 (6): 528–541. <https://doi.org/10.1080/01446193.2012.739288>.

Golparvar-Fard, M., A. Heydarian, and J. C. Niebles. 2013. "Vision-based action recognition of earthmoving equipment using spatio-temporal features and support vector machine classifiers." *Adv. Eng. Inf.* 27 (4): 652–663. <https://doi.org/10.1016/j.aei.2013.09.001>.

Gondia, A., A. Siam, W. El-Dakhkhni, and A. H. Nassar. 2020. "Machine learning algorithms for construction projects delay risk prediction." *J. Constr. Eng. Manage.* 146 (1): 04019085. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001736](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001736).

Hammad, A., S. AbouRizk, and Y. Mohamed. 2014. "Application of KDD techniques to extract useful knowledge from labor resources data in industrial construction projects." *J. Manage. Eng.* 30 (6): 05014011. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000280](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000280).

Han, S., and H. Pan. 2011. "Lessons learned from schedule estimation using real-time data in a concreting operation." In *Proc., 28th Int. Symp. on Automation and Robotics in Construction*. London: International Association for Automation and Robotics in Construction.

Iyer, S., S. K. Sinha, B. R. Tittmann, and M. K. Pedrick. 2012. "Ultrasonic signal processing methods for detection of defects in concrete pipes." *Autom. Constr.* 22 (Mar): 135–148. <https://doi.org/10.1016/j.autcon.2011.06.012>.

- Jenkins, J., and J. Lew. 2003. *Creating a plan for quality control on the construction jobsite*. West Lafayette, IN: Dept. of Building Construction Management, Purdue Univ.
- Jun, D. H., and K. El-Rayes. 2009. "Multi-objective optimization of resource scheduling in construction projects." In *Proc., Construction Research Congress 2009: Building a Sustainable Future*, 806–815. Reston, VA: ASCE.
- Khedr, M. K. 2006. "Project risk management using Monte Carlo simulation." In *Proc., AACE Int. Transactions, RI21*. Fairmont, WV: AACE International Transactions.
- Ko, C. H., and S. C. Li. 2014. "Enhancing submittal review and construction inspection in public projects." *Autom. Constr.* 44 (Aug): 33–46. <https://doi.org/10.1016/j.autcon.2014.03.027>.
- König, M., C. Koch, I. Habenicht, and S. Spieckermann. 2012. "Intelligent BIM-based construction scheduling using discrete event simulation." In *Proc., 2012 Winter Simulation Conf. (WSC)*, 1–2. New York: IEEE.
- Kuhn, M. 2008. "Building predictive models in R using the caret package." *J. Stat. Software* 28 (5): 1–26. <https://doi.org/10.18637/jss.v028.i05>.
- Lee, D.-E., and D. Arditi. 2006. "Automated statistical analysis in stochastic project scheduling simulation." *J. Constr. Eng. Manage.* 132 (3): 268–277. [https://doi.org/10.1061/\(ASCE\)0733-9364\(2006\)132:3\(268\)](https://doi.org/10.1061/(ASCE)0733-9364(2006)132:3(268)).
- Lee, K. P., H. S. Lee, M. Park, D. Y. Kim, and M. Jung. 2017. "Management-reserve estimation for international construction projects based on risk-informed k-NN." *J. Manage. Eng.* 33 (4): 04017002. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000510](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000510).
- Li, Y., K. Lu, and Y. Lu. 2017. "Project schedule forecasting for skyscrapers." *J. Manage. Eng.* 33 (3): 05016023. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000498](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000498).
- Lu, M., and H. Li. 2003. "Resource-activity critical-path method for construction planning." *J. Constr. Eng. Manage.* 129 (4): 412–420. [https://doi.org/10.1061/\(ASCE\)0733-9364\(2003\)129:4\(412\)](https://doi.org/10.1061/(ASCE)0733-9364(2003)129:4(412)).
- Ma, G., and M. Wu. 2019. "A big data and FMEA-based construction quality risk evaluation model considering project schedule for Shanghai apartment projects." *Int. J. Qual. Reliab. Manage.* 37 (1): 18–33. <https://doi.org/10.1108/IJQRM-11-2018-0318>.
- Mahfouz, T., J. Jones, and A. Khalda. 2010. "A machine learning approach for automated document classification: A comparison between SVM and LSA performances." *Int. J. Eng. Res. Innovation* 2: 53–62.
- Marín Ruiz, N., M. Martínez-Rojas, C. Molina Fernández, J. Soto-Hidalgo, J. Rubio-Romero, and M. Vila Miranda. 2018. "Flexible management of essential construction tasks using fuzzy OLAP cubes." In *Fuzzy hybrid computing in construction engineering and management*, edited by A. Fayek, 357–388. Bingley, UK: Emerald.
- McCabe, B. 2003. "Construction engineering and project management III: Monte Carlo simulation for schedule risks." In *Proc., 35th Conf. on Winter Simulation: Driving Innovation*, 1561–1565. New Orleans, LA: Winter Simulation Conference.
- Ökmen, Ö., and A. Öztaş. 2008. "Construction project network evaluation with correlated schedule risk analysis model." *J. Constr. Eng. Manage.* 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)).
- Park, M., and F. Peña-Mora. 2004. "Reliability buffering for construction projects." *J. Constr. Eng. Manage.* 130 (5): 626–637. [https://doi.org/10.1061/\(ASCE\)0733-9364\(2004\)130:5\(626\)](https://doi.org/10.1061/(ASCE)0733-9364(2004)130:5(626)).
- Pestana, A. C. V., T. D. C. Alves, and A. R. Barbosa. 2014. "Application of lean construction concepts to manage the submittal process in AEC projects." *J. Manage. Eng.* 30 (4): 05014006. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000215](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000215).
- Rachid, Z., B. Toufik, and B. Mohammed. 2019. "Causes of schedule delays in construction projects in Algeria." *Int. J. Constr. Manage.* 19 (5): 371–381. <https://doi.org/10.1080/15623599.2018.1435234>.
- Ren, Z., M. Atout, and J. Jones. 2008. "Root causes of construction project delays in Dubai." In *Proc., 24th Annual ARCOM Conf.*, 1–3. Cardiff, UK: Association of Researchers in Construction Management.
- Russell, M. M., G. Howell, S. M. Hsiang, and M. Liu. 2013. "Application of time buffers to construction project task durations." *J. Constr. Eng. Manage.* 139 (10): 04013008. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000735](https://doi.org/10.1061/(ASCE)CO.1943-7862.0000735).
- Ryu, J., J. Seo, H. Jebelli, and S. Lee. 2019. "Automated action recognition using an accelerometer-embedded wristband-type activity tracker." *J. Constr. Eng. Manage.* 145 (1): 04018114. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001579](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001579).
- Sakhakarmi, S., J. Park, and C. Cho. 2019. "Enhanced machine learning classification accuracy for scaffolding safety using increased features." *J. Constr. Eng. Manage.* 145 (2): 04018133. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001601](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001601).
- Snyder, J., A. Menard, and N. Spare. 2018. *Big data: Big questions for the engineering and construction industry*. Raleigh, NC: Fails Management Institute.
- Soibelman, L., and H. Kim. 2002. "Data preparation process for construction knowledge generation through knowledge discovery in databases." *J. Comput. Civ. Eng.* 16 (1): 39–48. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2002\)16:1\(39\)](https://doi.org/10.1061/(ASCE)0887-3801(2002)16:1(39)).
- Tixier, A. J.-P., M. R. Hallowell, B. Rajagopalan, and D. Bowman. 2016. "Application of machine learning to construction injury prediction." *Autom. Constr.* 69 (Sep): 102–114. <https://doi.org/10.1016/j.autcon.2016.05.016>.
- Wambeke, B. W., S. M. Hsiang, and M. Liu. 2011. "Causes of variation in construction project task starting times and duration." *J. Constr. Eng. Manage.* 137 (9): 663–677. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000342](https://doi.org/10.1061/(ASCE)CO.1943-7862.0000342).
- Wang, W.-C. 2005. "Impact of soft logic on the probabilistic duration of construction projects." *Int. J. Project Manage.* 23 (8): 600–610. <https://doi.org/10.1016/j.ijproman.2005.05.008>.
- Wang, X., X. Huang, Y. Luo, J. Pei, and M. Xu. 2018. "Improving workplace hazard identification performance using data mining." *J. Constr. Eng. Manage.* 144 (8): 04018068. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001505](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001505).
- Williams, T. P., and J. Gong. 2014. "Predicting construction cost overruns using text mining, numerical data and ensemble classifiers." *Autom. Constr.* 43 (Jul): 23–29. <https://doi.org/10.1016/j.autcon.2014.02.014>.
- Witten, I., E. Frank, M. Hall, and C. Pal. 2016. *Data mining: Practical machine learning tools and techniques*. Amsterdam, Netherlands: Morgan Kaufmann.
- Wu, X., X. Zhu, G. Wu, and W. Ding. 2014. "Data mining with big data." *IEEE Trans. Knowl. Data Eng.* 26 (1): 97–107. <https://doi.org/10.1109/TKDE.2013.109>.
- Yang, Y., and X. Liu. 1999. "A re-examination of text categorization methods." In *Proc., 22nd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 42–49. New York: Association for Computing Machinery.
- Yip, H.-L., H. Fan, and Y.-H. Chiang. 2014. "Predicting the maintenance cost of construction equipment: Comparison between general regression neural network and Box–Jenkins time series models." *Autom. Constr.* 38 (Mar): 30–38. <https://doi.org/10.1016/j.autcon.2013.10.024>.
- Zhang, C., and Y. Ma. 2012. *Ensemble machine learning: Methods and applications*. New York: Springer.
- Zhao, X., Y. Tan, L. Shen, G. Zhang, and J. Wang. 2019. "Case-based reasoning approach for supporting building green retrofit decisions." *Build. Environ.* 160 (Aug): 106210. <https://doi.org/10.1016/j.buildenv.2019.106210>.
- Zidane, Y. J. T., and B. Andersen. 2018. "The top 10 universal delay factors in construction projects." *Int. J. Managing Projects Bus.* 11 (3): 650–672. <https://doi.org/10.1108/IJMPB-05-2017-0052>.