

Article

Generation of Construction Scheduling through Machine Learning and BIM: A Blueprint

Mazen A. Al-Sinan ^{1,*}, Abdulaziz A. Bubshait ²  and Zainab Aljaroudi ¹

¹ Procurement & Supply Chain Management, Saudi Aramco, Dhahran 31311, Saudi Arabia; zainab.jaroudi@aramco.com

² Department of Architectural Engineering and Construction Management, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia; bushait@kfupm.edu.sa

* Correspondence: mazen.sinan@aramco.com

Abstract: Recent advancements in machine learning (ML) applications have set the stage for the development of autonomous construction project scheduling systems. This study presents a blueprint to demonstrate how construction project schedules can be generated automatically by employing machine learning (ML) and building information modeling (BIM). The proposed solution should utilize building information modeling (BIM) international foundation class (IFC) 3D files of previous projects to train the ML model. The training schedules (the dependent variable) are intended to be prepared by an experienced scheduler, and the 3D BIM files should be used as the source of the scheduled activities. Using the ML model can enhance the generalization of model application to different construction projects. Furthermore, the cost and required resources for each activity could be generated. Accordingly, unlike other solutions, the proposed solution could sequence activities based on an ML model instead of manually developed constraint matrices. The proposed solution is intended to generate the duration, cost, and required resources for each activity.

Keywords: autonomous systems; autonomous project management system; BIM; ChatGPT; construction; GPT; LLMs; machine learning; project scheduling



Citation: Al-Sinan, M.A.; Bubshait, A.A.; Aljaroudi, Z. Generation of Construction Scheduling through Machine Learning and BIM: A Blueprint. *Buildings* **2024**, *14*, 934. <https://doi.org/10.3390/buildings14040934>

Academic Editor: Ahmed Senouci

Received: 23 January 2024

Revised: 21 March 2024

Accepted: 25 March 2024

Published: 28 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Developing a construction project schedule is a complex process that demands considerable time and effort. In addition, the traditional manual approach to creating a schedule can be susceptible to human error [1]. Activities sequencing poses an added challenge, as schedulers must identify resource needs and realistic completion times. Although software solutions are available, such as MSP2019 and Primavera P6, they still require labor-intensive manual input [2,3]. Poor-quality schedules might lead to contract issues, unsatisfied stakeholders, and disputes [4]. Thus, a smart computer-based solution is needed to enhance construction schedule generation.

The development of automated construction schedules has been widely studied, including the utilization of BIM in generating schedules. These studies, however, have fallen short of offering widely adopted solutions [5,6]. The previous methods were manually formed and based on a specific project, which makes it difficult to generalize them to different construction projects [4].

Although a vast amount of research has been conducted on employing AI in various aspects of project management, research on practical construction applications is scarce [7]. This could be due to the nature of the construction industry, which has always resisted integrating the latest technologies and changing the status quo in terms of its practices [8,9]. Its reluctance toward change is further intensified by other deterrents, including costs, the uncertainty of benefits and gains, lack of investment in research and development, lack of standardization of data protection and cybersecurity, legal and contractual issues,

difficulty in adopting new technologies, and the fragmented and project-based nature of the industry [8,10,11].

This paper presents a blueprint to develop an autonomous system that can generate a schedule for building projects directly from BIM files by leveraging ML. BIM data, such as building dimensions, materials, and relationships, can be used to create construction schedules automatically. The blueprint provides a solution that could be capable of sequencing activities and determining the resources, duration, and cost of each activity. Although this blueprint has been developed for building projects, it could be expanded to other types of construction projects.

This research introduces a theoretical foundation and the potential implications of the proposed autonomous scheduling system, anticipating future research work to include full implementation of the system, using data from different types of construction projects and comprehensive evaluations. This paper is organized as follows: Section 2 discusses the methodology. Section 3 reviews the literature and provides relevant background information, focusing on the concepts of AI and BIM. Section 4 describes the proposed blueprint for creating an autonomous construction project scheduling system. Section 5 provides a brief discussion. Finally, Section 6 concludes the study.

2. Methodology

The research strategy was based on action research methodology. Action research is also known as an approach to problem-solving. In action research, the investigator examines the current situation, identifies the issue, makes adjustments to improve the situation, and then assesses the results of those adjustments [12,13].

Examining the traditional process undertaken by schedulers to develop an independent schedule, a scheduler typically gathers project information from a variety of sources, including work breakdown structures (WBSs), statements of work, contract documents, specifications, drawings, and conversations with project engineers. The process is time-consuming and demanding, and a need for creating a general automatic schedule was identified. Although several models using building information modeling (3D BIM) and AI have been reported in the literature, this study proposes a blueprint that could adjust and improve the process. This study did not include an assessment component.

3. Literature Review

The following subjects are presented in this section: previous studies, AI, and Web scraping.

3.1. Previous Studies

Several studies have used AI in coordination with 3D and 4D BIM to generate construction schedules [2,13–18]. Kim et al. [2] proposed a framework for automating schedule generation from BIM, where the sequencing rules are predefined. Hong et al. [14] proposed graph-based automated scheduling (GAS) to automatically store the best practices of past projects and reuse them to generate a new project schedule. Faghihi et al. [15] proposed a method for retrieving information from BIM to develop construction sequencing using a genetic algorithm (GA). Faghihi [16] developed a matrix of constructability constraints (MoCC) for the relationships and dependencies of project elements based on the experience gained from similar works as a prerequisite for his proposed solution to create construction schedules using a genetic algorithm based on BIM. To develop the MoCC, a set of predefined structural stability rules was considered. These rules are related to constructability and stability common knowledge—e.g., upper-level columns should be installed after lower-level columns [16].

Wu et al. [17] automatically generated a construction schedule by extracting data from the BIM model and combining an ontology constraint rule and a genetic algorithm (GA). They introduced the concept of ontology to build constraint rules for building components. El-Menshawey et al. [18] proposed a model for automated schedule generation from BIM, in which the output was exported to Primavera. Then, the user can modify the generated

schedule. In addition, they proposed an optimization model that provided a near-optimum solution resulting from various scenarios. However, all of these methods entail extensive manual preparation and are based on a specific project, and thus encounter challenges to be generalized to other projects.

Classical computational tools, such as genetic algorithms (GAs), have been extensively employed to navigate complex challenges such as optimizing times and costs. Nevertheless, the intricate nature of architectural, engineering, and construction (AEC) projects is often characterized by considerable variables and constraints. Determining optimal solutions can be a protracted process. However, recent advancements in quantum computing have significant potential for tackling such optimization issues, offering approaches that could revolutionize the way complex, large-scale optimization challenges are addressed in the field [19].

3.2. Artificial Intelligence

In the context of autonomy, AI is defined as “a system’s ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation” [20]. In addition to building information modeling (BIM), various algorithms and artificial intelligence (AI) tools, such as case-based reasoning (CBR), knowledge-based systems, GAs, expert systems, and neural networks, have been considered for automating schedules [16–19].

Machine Learning

ML, an algorithm that processes large amounts of data to generate patterns (e.g., classification and categorization), has expanded the potential applications of AI. ML algorithms are fed data from which they learn by performing statistical and probabilistic analyses. The algorithm then extrapolates possible outcomes based on the results of the analyses [21].

Three main ML techniques are distinguished by the learning method [22,23]:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

Some researchers occasionally refer to a fourth ML technique called deep learning, a semi-supervised method in which a large mix of labeled and unlabeled data is used in training and analysis [19]. In this technique, a special neural network imitates the human learning process by using examples. One of the most popular deep learning neural networks is the convolutional neural network (CNN). Deep learning models can directly classify images, sounds, or texts with high accuracy, outperforming humans [20]. The most popular ML algorithms for prediction include artificial neural networks (ANNs), k-nearest neighbors (k-NN), eXtreme gradient boosting (XGBoost), logistic regression (LR), support vector machine (SVR), random forest (RF), and decision tree [24].

Natural Language Processing (NLP) refers to a model’s ability to process human speech and text and simultaneously understand their associated meaning [25]. Computational linguistics, rule-based modeling of human language, ML, and deep learning techniques have been used in NLP development. NLP applications include spam detection, text mining, information retrieval, language translation, Q&A systems, and dialogue systems (chatbots) [26]. Text-based automation is another approach for automatically generating a project schedule, where the focus is on analyzing the textual data of the schedule. For example, CBR involves a text base in which an old, similar case is recalled in the present case and used to suggest a solution to the new problem. Researchers have recently explored the use of NLP to analyze descriptions of construction schedule activities [27].

Amer et al. [28] proposed an ML transformer language model to learn the implicit dependency constraints and the flexibility of construction schedule relationships. Amer et al. [29] proposed an ML-based method using LSTM-RNNs, where construction sequences extracted from previous project schedules were used as training data. The model learns potential successor activities based on a sequence of previously scheduled activities. Amer’s

method depends on NLP principles. Building construction vocabulary vectors were developed using limited words to retrieve activities from old schedules. The method has limitations since the dependencies of the activities had to be developed manually. This method cannot be transformed into a reliable automated solution to generate a project schedule. It merely demonstrates that AI can predict successor activities after training the NLP model. In general, scheduling solutions based on NLP models have inherent challenges, because each project has its size, activity duration, and resource usage, meaning that previous project records cannot be used as a basis to develop a project schedule.

Large Language Models (LLMs) are NLP models that have a substantially higher number of parameters than other NLPs. Using deep learning techniques, LLMs are equipped with transformers to generate more accurate results [30]. The process of predicting the next word is known as language modeling. Usually, the model is fine-tuned to perform a specific task, such as answering a question or translating a language [31]. LLMs are classified based on their parameters and the amount of sampled training data. As with any ML model, the more training data fed and the more parameters used, the more accurate the response generated by the model.

Generative Pre-Trained Transformers (GPTs) are a group of LLMs that produce natural language texts based on input prompts, similar to other LLMs, but they have transformed beyond that. Despite the use of an astronomical number of parameters and training data, GPT models have not yet perfected human thinking. Bias from training data, inability to reason like humans, lack of common sense, and problems with deterministic answers are some of ChatGPT's biggest weaknesses, as the algorithm of the model is probabilistic in nature [25,32].

Only one study reported in the literature discussed the use of ChatGPT to schedule construction projects [33]. That study assessed the capability of ChatGPT to generate resource-loaded project schedules, revealing that ChatGPT is promising but needs to be explicitly trained to generate acceptable results [33]. The authors believed that ChatGPT could be essential for building an autonomous construction project scheduling solution but not act as the main backbone for the solution. Additionally, ChatGPT could play a more effective role when an autonomous project management system is built based on extracting recommendations and requirements from big data to address ad hoc situations.

3.3. Web Scraping

Web scraping refers to the process by which a program extracts data from the Web [34]. AI-powered Web scrapers have become a common business tool used to collect and analyze data, draw conclusions, and make informed decisions. In the construction industry, they have been used for industry growth analysis, project identification and assessment, construction market data analysis for supply chain optimization, and talent acquisition and recruitment. Due to their ability to deliver accurate, timely, and comprehensive data, businesses are now more equipped to make strategic decisions that optimize operations across various business domains [35].

4. Autonomous Construction Project Scheduling System Development

Project scheduling is one of the most significant activities that affect a project's success and requires high levels of experience and intelligence. Accordingly, autonomous project planning may foster the development of a comprehensive autonomous project management solution for all project phases.

The proposed system should provide a solution that performs the following:

1. Develop a schedule autonomously from BIM files
2. Assign the duration and resources required (staff, equipment, and materials) for each activity
3. Determine the cost of each activity
4. Optimize project costs and schedules by analyzing the assigned resources (staff, equipment, and materials)

The proposed solution should be cloud-based, allowing multiple users to access it. Having multiple users is important in the retrieval of real-life data for training data; this ensures that the model is regularly updated, thus improving its performance. The system should be integrated with an enterprise resource planning (ERP) system, which might entail reconfiguring some existing databases to make them compatible with the solution. User data, particularly those related to human resources and equipment, are essential for developing a resource-loaded project schedule. Moreover, real-time data on these resources aid in identifying the activities that must be subcontracted.

4.1. Autonomous Project Scheduling System Structure

A project schedule that indicates the tasks, due dates, expenses, and available resources is a resource-loaded schedule. Generating such a schedule requires identifying and sequencing project activities, assigning resources, and determining the project size, cost, and duration. A BIM-based framework for creating schedules based on a set of sequencing principles can be implemented using a five-phase method [2]: (1) construction of the BIM, (2) parsing of the BIM data, (3) transforming the parsed data into activity data, (4) generating a schedule, and (5) refinement of the schedule. The proposed solution would employ BIM data files to determine scheduled activities. Unlike other solutions, the proposed solution would sequence activities based on an ML model rather than constraint matrices [16,17]. Accordingly, separate iterations of training should be conducted for each type of project (e.g., multi-story buildings or roads) to improve the performance of the system. Ultimately, the solution should be able to generate a resource-loaded schedule for any construction project type after training.

For demonstration, the proposed solution assumed that the project is a multi-story building. The proposed solution implicitly develops the constraints within the model during the training. This could provide more flexibility in accommodating unique designs. A diagram of the proposed autonomous project scheduling solution is shown in Figure 1.

The proposed solution consists of the following components:

1. Project analyzer.
2. Data warehouse/Smart warehouse.
3. Activity identifier and sequencing.
4. Activity magnitude and direct cost calculator.
5. Activity duration calculator.
6. Schedule analyzer (project actual performance).

The components mentioned can interact through different methods; for instance, the project analyzer can communicate with the scheduling model via remote procedure calls (RPCs) to determine the project type. After the project type is identified, the applicable ML model will be selected. The BIM files can be transmitted to the system through application programming interfaces (APIs) or as extract, transform, and load (ETL) processes. Subsequently, the ML model will retrieve data from the BIM, activity dictionary, labor folder, and equipment folder, using Shared Memory to generate the schedule. Figure 2 illustrates the process of generating the resource-loaded schedule.

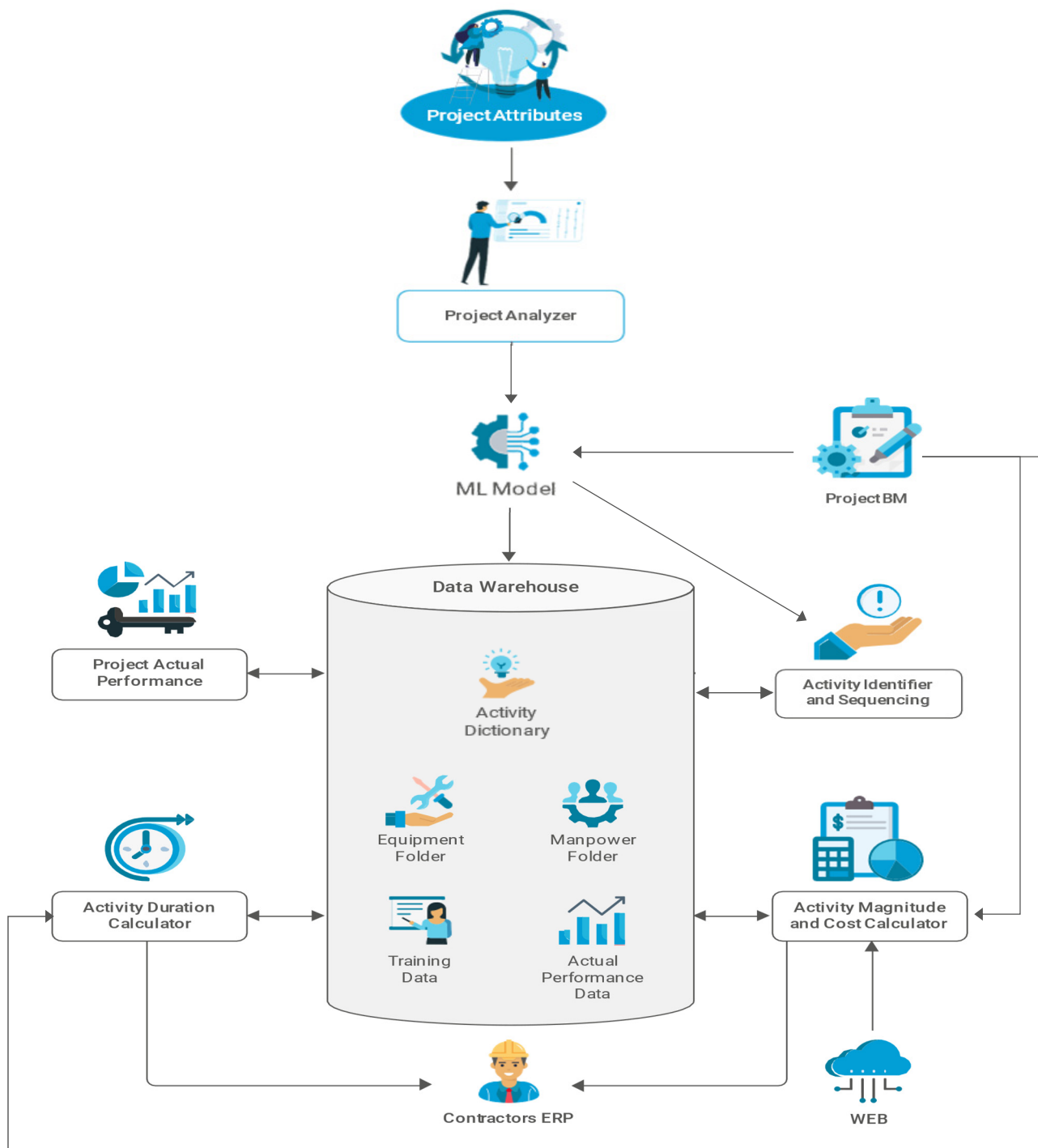


Figure 1. Autonomous Construction Project Scheduling System.

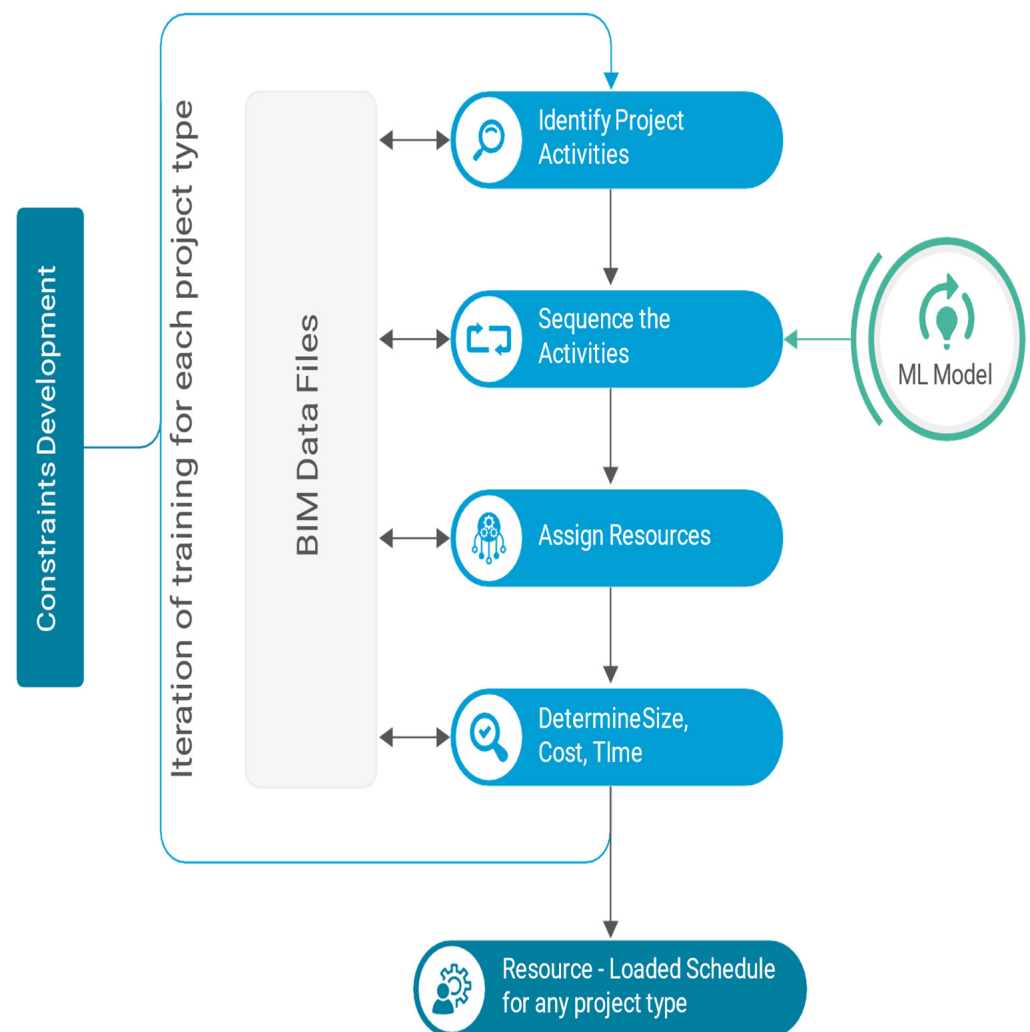


Figure 2. Resource-loaded schedule generation process.

4.1.1. Project Analyzer

Each construction project is unique with its own site characteristics, weather conditions, workforce, and equipment. The project type (e.g., commercial, industrial, or residential), size, and location (offshore or onshore) affect the development of the entire schedule. Initially, users should enter the general attributes of the project into the project analyzer. The project attributes could include the project's budget, duration, applicable standards, quality and safety requirements, and the maximum subcontracting percentage.

A user-selectable dropdown menu containing various project types could be implemented, which would enable the user to specify the project type, allowing the solution to employ the corresponding machine-learning model that has been specifically trained for that construction project type.

4.1.2. ETL and Data Warehouse

Extracting data from multiple sources and formats; and transforming them into a unified data structure that can be utilized for data warehousing and integration is known as extract, transform, and load (ETL). Three data categories are typically found in BIM databases: unstructured, semi-structured, and structured. All categories must go through the ETL procedure. On the other hand, information is exchanged between the BIM and the proposed model through the extract, load, and transform (ELT) process, which provides workflow management for cleaning and loading. The data warehouse consists of the following:

- a. Activity Master Data Folder (Activity Dictionary).
- b. Equipment Directory Folder.
- c. Manpower Directory Folder.
- d. Training Data Repository.
- e. Actual Performance Data.

Each folder in the data warehouse serves as a repository for specific datasets. The data contained in each folder are intended to be related and interconnected, allowing for seamless querying and aggregation using data-querying languages, such as “Not only Structured Query Language” (NoSQL).

Activity Master Data Folder (Activity Dictionary)

To develop the solution, it is essential to standardize project activities. Universal project activity master data (UPAMD) is an advanced, expanded construction elements dictionary. Since the most comprehensive source for identifying activities is 3D drawings, these elements would be based on BIM IFC models, parsing of the BIM data, and transforming the parsed data into activity data, with each element regarded as an activity. The project type and contractor’s resources should also be considered and addressed during the development of the UPAMD. Additionally, an identifier should be assigned to each element based on its location. For example, in a multi-story building, there would be a unique code for each element depending on the BIM level of development (LOD).

Each activity (construction element) should have a file with an assigned code that provides a description of the activity and its characteristics (including multiple sub-activities to accomplish the structure). There could be multiple versions of the same activity, which would require multiple files with different codes. The activity file could contain a certain number of fields, as listed in Table 1.

Table 1. Dictionary Activity File Contents.

No.	Field	Description
1	Activity Code	Each activity type is assigned a specific code
2	Activity Name	Shows the activity title, such as “Painting”
3	Activity Type Description	Provides a general description to define the activity
4	Required Equipment	Lists the equipment and tools required to perform the activity
5	Required Worker Skills	Lists the work skills required to perform the activity
6	Required Materials	Lists the materials required to perform the activity
7	Measurement Unit	Identifies the measurement unit based on the nature of the activity (e.g., square meter or cubic meter)
8	Hard and soft logic	If hard and soft logic are both required, an activity code is shown here.
9	Workers’ productivity	Indicates the number of hours of each worker skill required to complete one measurement unit
10	Productivity of Equipment	Displays the number of equipment hours to complete one measurement unit
11	Required Material Quantity Per Measurement Unit	For example, the number of gallons of paint per square meter
12	Safety Requirements	Applicable safety requirements
13	Quality Test	Applicable quality test
14	Special Requirements	Reports on special requirements

Developing an activity master data folder (dictionary) should be a prerequisite for conducting system training. Developing an activity dictionary could be a demanding task in which GPT can assist. For a cloud-based solution, the time and cost to build an activity dictionary should justify such efforts.

Equipment Directory Folder

For each piece of equipment available on the market, a file with several fields should be created where the equipment information could be retained. Table 2 shows an example of a piece of equipment file.

Table 2. Equipment File Contents.

No.	Field	Description
1	Equipment Code	Each piece of equipment is assigned a specific code
2	Equipment Name	Name(s) of the piece of equipment
	Equipment Description	Brief description of the equipment
	Equipment Manufacturer	The name and address of the manufacturer
3	Equipment Uses	All activities in which the equipment could be used are listed as activity codes. This field should feed the required equipment field in the activity master data dictionary.
4	Equipment Lease Cost	Indicated daily or hourly lease cost; this could be either scraped from the web (in real-time) or from a vendor site. The rate in this field should feed project schedule activity equipment costs.
5	Required Operators	The job code and number of required operators
6	Energy Type	Diesel, gasoline, or electrical energy sources indicated
7	Energy Consumption	The average consumption of energy per hour; this field could be used to adjust the activity cost during construction when the cost of energy changes.
8	Equipment Capacity	The standard capacity measurement of the equipment (e.g., 1 horsepower for a compressor)
9	Equipment Model	Specifies the model available; this field could be used for analysis.
10	Equipment Operation Cost	This cost is extracted either from the web or as input from the manufacturer.
11	Equipment Weight/Dimensions	Equipment weight or dimensions shown

The original equipment manufacturer (OEM) could be granted temporary access to create a file for any piece of equipment. In the case of contractor-owned equipment, there should be an equipment database using a file structure similar to that used in the equipment directory folder. Additional fields could be added to capture information related to the location, amount of available equipment, maintenance costs, preventive maintenance schedule, and other relevant information, as applicable. The proposed solution should prioritize utilizing owned equipment to optimize scheduling rather than leasing additional equipment.

Workforce Directory Folder

A universal folder with the job titles of all those working in the AEC industry should be created with a file for each job title. Table 3 shows an example of a job file, along with possible fields.

Contractors should have a digital human resources system with files similar to those found in the workforce directory folder. Each job title could have additional fields to capture further information, such as the available headcount. A subfile could also be created for each worker to capture historic productivity from previous projects, availability schedules, and other qualifications. The contractor might have limited available human resources, with a need to either hire temporary workers or subcontract certain activities.

Table 3. Job File Contents.

No.	Field	Description
1	Job Code	Each job is assigned a specific code
2	Job Title	Each job is assigned a specific title
	Job Description	A brief description of the job
3	Job Requirements	The minimum qualifications and years of experience required for the job
4	Activities to be Assigned	All activities to which this job can be assigned; this field should feed the worker requirements field in the activity database dictionary.
5	Hourly Rate	This could be extracted from the web or ERP and fed to the activity cost in the project activity cost.

Training Data Repository, ML Model, and Actual Performance Data Folders

All training data (drawings and files from BIM), with their developed schedules, should be uploaded to the training data repository. In addition, schedules developed by the system and the actual project's progress after project execution should be retained in the training data repository and used as training data to improve the future system's performance. For each completed project, a record of the estimated resources, costs, and duration of each activity compared to the actual duration and cost after completion should be stored under the actual performance data for analysis and further model training.

4.1.3. Training Data Preparation

To develop the proposed solution effectively, the training data should encompass all elements expected within a generated schedule. The following processes are essential to creating high-quality training data: data preparation, feature extraction, activity classification, recognition of activities and their relationships, and sequencing. Data preparation is particularly critical due to the lack of standardized terminology and manually developed schedules. This phase involves renaming activities using an activity dictionary, refining schedules, and rectifying errors. Subsequent steps will leverage AI capabilities for enhanced efficiency.

Each activity in the activity dictionary has a code and a corresponding activity title. Thus, training for activity sequencing logic should depend on the activity code, which can be translated into a specific activity in the schedule via the activity title (see Fields 1 and 2 in Table 1). Some activities can be interchangeable or concurrent, and since manually developed schedules will be conducted by different schedulers, will cater to various acceptable logics. Manually developed schedules should be used as training data to develop the activity sequence logic. Finally, the model should be able to determine the logical sequence of activities based on the relationships among the BIM components. With its exhaustive initial efforts, this approach should provide a comprehensive approach that could handle any project design.

4.1.4. Model Training

After identifying construction activity dependencies during data preparation, the model can finally be trained. The refined schedules and BIM files should be used as inputs to generate an ensemble model that should be fully capable of autonomous scheduling. Ensemble modeling is a more effective method than using predefined sets of rules or structural stability rules for identifying project activity dependencies. Ensemble methods integrate multiple learning algorithms to improve performance metrics such as accuracy, precision, and robustness by mitigating the issues of bias and variance. A prominent technique within ensemble learning is "Disjoint Sample Aggregation," or Dagging, which involves dividing the original dataset into several exclusive subsets for training. These subsets serve as distinct training sets for the base learners. These learners' predictions are

then synthesized by a meta-learner to produce a final prediction that typically has higher reliability and performance than any single learner's output [24].

The training process entails developing a schedule for each BIM model as the dependent variable, whereas the project elements extracted from the IFC files should be considered as independent variables. Any changes or delays in the project will automatically trigger the creation of a new revised schedule. Each type of project (e.g., building or bridge) would require separate training and a separate model.

4.1.5. Activity Identifier and Sequencing

Lee et al. [36] used BIM 3D to generate activity units. Lee et al. [36] used 3D BIM to generate activity units. They applied PointNet++ to classify the 3D model into activity units with BIM and used a genetic algorithm to optimize the scheduled activity sequence and to generate schedule information. The IFC files store the geometry information of elements based on the *buildingSMART* standards. Accordingly, the spatial and geometric data of an element can be obtained from an IFC reader or even a text reader since IFC files are text-based [16]. For the proposed blueprint, several software tools can be used to extract structural elements from IFC files, including Autodesk Revit and Bentley Systems. Alternatively, software such as *IfcOpenShell*, which is programmable, can be used to access and navigate the data from IFC files to extract details about structural elements. *IfcOpenShell* provides python bindings that can extract specific information about structural elements, such as walls, column beams, and slabs. A third approach is to employ natural language processing (NLP) to extract the project elements from the IFC since it is text-based. Key attributes, such as object geometry, dimensions, materials, and relationships, can be extracted from IFC files. By leveraging these attributes, the artificial intelligence (AI) model can enable object detection and classification using deep-learning techniques. This should entail creating a neural network capable of identifying and categorizing BIM components (e.g., walls, doors, and windows) within the 3D model using BIM. Once categorized, the AI model needs to recognize how these components relate to specific construction activities. Following this, NLP can be employed to generate descriptions of each BIM component based on its attributes, relationships, and contexts. These descriptions align with the language used in the predefined UPAMD to link the components with the required activities.

After feeding the solution with the project data type, the system should select the applicable model. Subsequently, the system will generate activities with their dependencies (Figure 3). The project elements should be extracted from the IFC files. Accordingly, the applicable ML will generate the sequence of the activities. Each activity in the activity sequence should depend on the preceding activity. For each identified activity, an activity file similar to the activity dictionary file will be created. Table 4 presents the project schedule activity file.

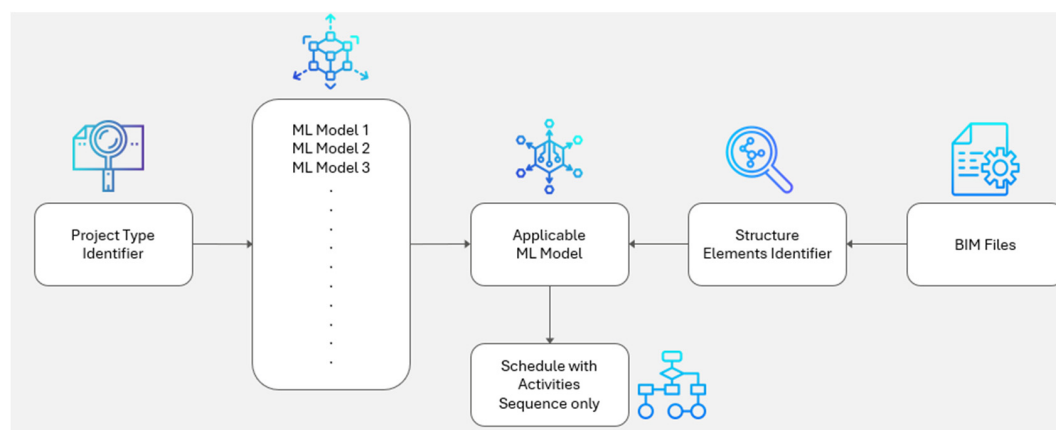


Figure 3. Activity sequencing.

Table 4. Project Schedule Activity File.

No.	Field	Description
1	Activity Code	Activity dictionary code number
2	Activity Title	Activity title, such as “Excavation”
3	Activity Sequence	This is generated from the ML model.
4	Activity Predecessors	This field should be used to determine the project’s critical path and the total project duration.
5	Activity Type Description	A general description of the activity
6	Activity Magnitude	To be fed from the activity magnitude calculator (see Section 4.1.6)
7	Required Equipment	Used to calculate the required hours based on the magnitude of the activity (see Field 6) and the standard equipment hour per activity unit of measurement
8	Required Worker Skills	Used to calculate the required manhours by multiplying the magnitude from Field 6 by the standard required manhours per activity unit of measurement
9	Required Materials	Used to calculate the required material by multiplying the magnitude of the activity from Field 6 by the required material per unit of measurement
10	Activity Measurement Unit	Identifies the measurement unit (e.g., square meter or cubic meter) based on the nature of the activity
11	Mandatory Pre-Activity	The prior activity or activities identified by the ML model
12	Mandatory Post-Activity (Predecessors)	Predecessor activity or activities identified by the ML model

4.1.6. Activity Magnitude and Direct Cost Calculator

Banihashemi et al. [37] proposed a framework to integrate a cost estimation standard with BIM. Since each structural element is treated as an activity, the dimensions of the elements are retained in the BIM database (UPAMD). Accordingly, the magnitude of the work of each activity can be determined. This should be fed to the activity magnitude field in the project activity file (Field 6 in Table 4). From the element size, the material quantity, staff requirements, and equipment can be determined. Since worker productivity for each unit of measurement is one of the variables in the activity file (Field 9 in Table 1), the required hours for each activity and equipment hours (Filed 10 in Table 1) could be estimated. Likewise, the necessary quantity of material could be determined by dividing the magnitude of the required material per unit (Field 11 in Table 1). The total activity cost could then be estimated by multiplying the resources with the updated actual cost of materials, staff, and equipment. The material costs could be retrieved directly from the Web using an AI-powered search engine, while the costs of equipment and workers could be sourced from the contractor’s ERP system. The estimated total direct project cost could be determined by aggregating the direct costs of all activities. Initially, for the sake of calculating a project’s direct cost, resource constraints could be ignored. The indirect cost is a function of the project duration, which could be determined once the project duration is set. Figure 4 shows a schematic for the activity cost calculator.

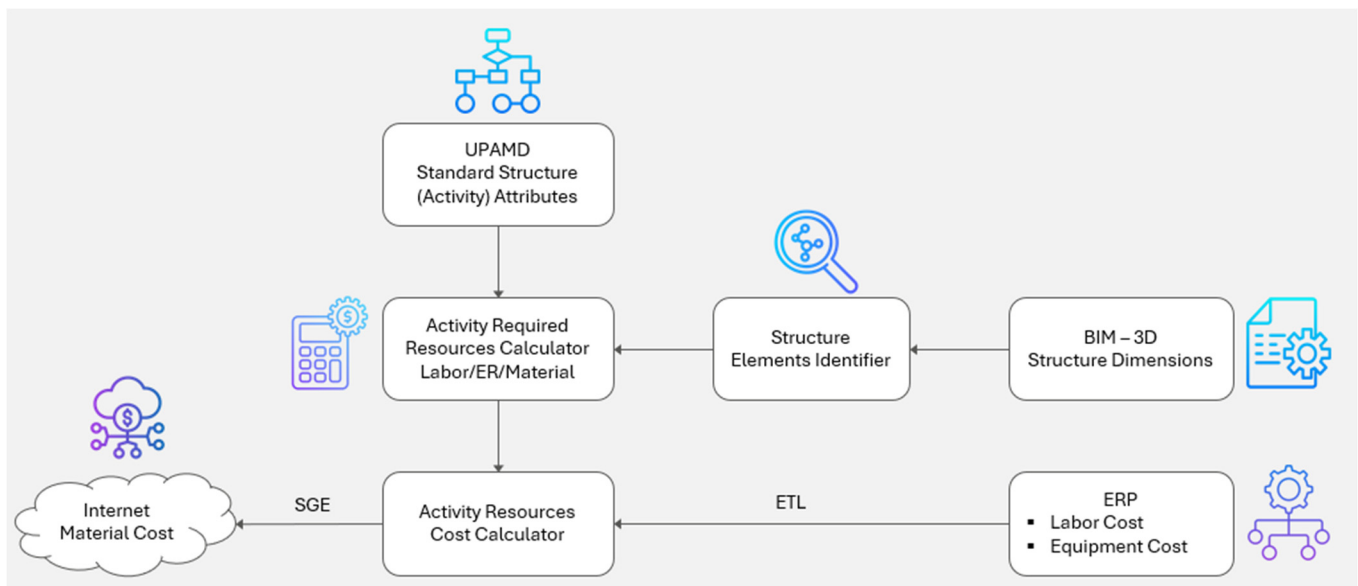


Figure 4. Activity cost calculator.

4.1.1.7. Activity Duration Calculator

The duration of each activity varies depending on the resources assigned, including both labor and equipment, making it essential to create a resource allocation matrix to distribute these resources effectively for each task. Further, constraints on resources may stem from operational challenges, such as limited workspace or a lack of available resources. Furthermore, it is critical that the resources committed to each task and any concurrent tasks do not surpass the total resources available, as indicated by the contractor's Enterprise Resource Planning (ERP) system. To determine the time required to complete each activity, the duration could be computed by dividing the work volume of the task by the productivity rate of the allocated resources (refer to Fields 9 and 10 in Table 1 for workers and equipment, respectively). Figure 5 shows a schematic for the activity duration calculator.

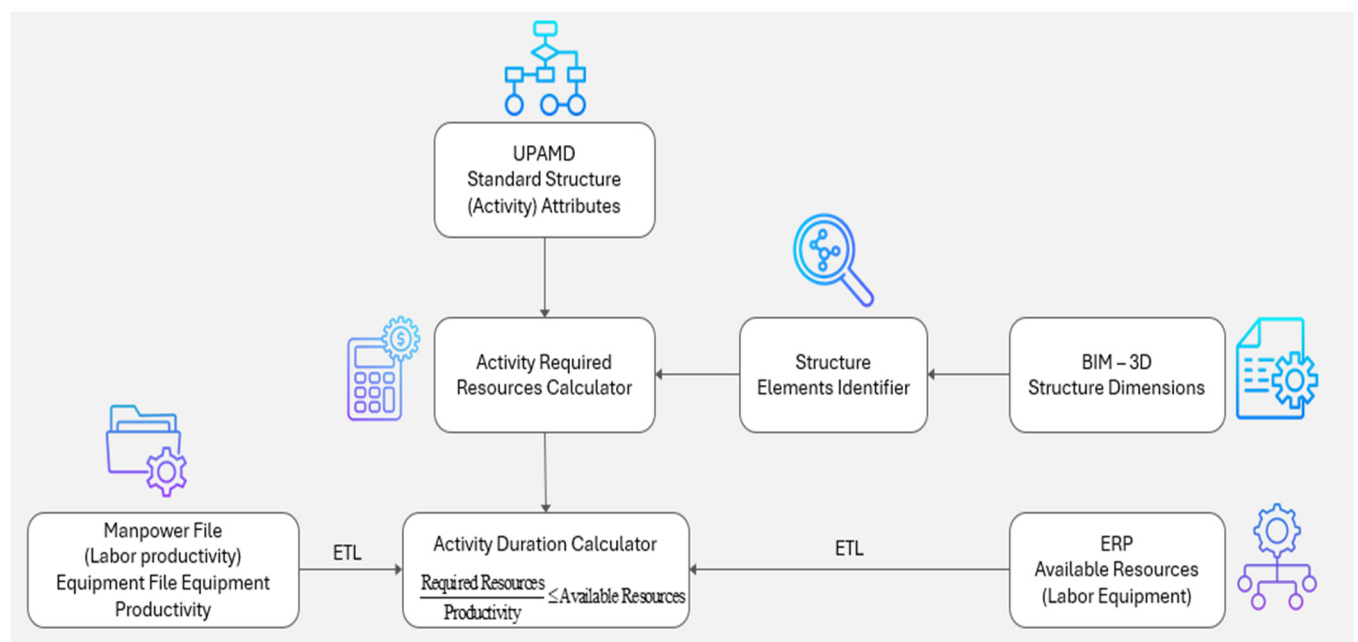


Figure 5. Activity duration calculator.

4.1.8. Schedule Analyzer

This system could calculate the direct costs of the project. The indirect costs should be determined by the contractor. The indirect cost could be a predefined amount per day or could be a function of the project's active resources. Determining the indirect cost is important for determining the optimal project cost and duration schedule, as there is a trade-off between project cost and duration.

Multi-objective linear programming could be employed to determine the optimal project cost and duration [38–40]. The financial analysis could also be conducted based on the progress payment percentage terms and the user's financial position. The actual duration and cost during the execution of the project should be monitored and used as training data to improve future performance and adjust the schedule to meet its targets.

5. Discussion

Using NLP to generate the construction schedule based on textual input from previous schedules as training data might not result in generating a reliable schedule. Each project is unique, and there is always room for innovation in design. In addition, schedulers are inconsistent when they develop project schedules, which makes the format of the schedule vary from one company to another. Another fundamental challenge is that the duration of the activities, costs, and resources cannot be retrieved from old schedules unless the projects are repetitive, as the magnitude of each activity varies between projects. Thus, the NLP could only predicate the successor activity and will always encounter challenges with concurrent activities. The proposed blueprint could generate a schedule that surmounts all shortcomings of NLP-generated schedules.

One of the major challenges to building the proposed solution is to improve the current IFC standard for cost and resources to be compatible with the proposed UPAMD. This entails who will take the initiative to build the proposed solution to work with the organizations (e.g., ISO, *BuildingSmart*) responsible for establishing and maintaining standards for BIM implementation.

Similar to the solutions proposed by Lee et al. [36] and Faghihi [16], the proposed blueprint will generate a schedule in which each structural element is considered an activity. This could be impractical for a schedule that is used by a project manager since the number of activities is too high.

The solution could be designed to generate the five levels of the construction schedule. Level 1 is the executive summary, commonly known as the project master schedule (PMS). It is a major milestone schedule that lists the main project activities, milestones, and deliverables. Level 5 is the detailed schedule used to map out the detailed tasks needed to coordinate day-to-day work in a specific area. Alternatively, this challenge can be resolved by developing a solution that aggregates the activities based on the geometric data—e.g., first-floor activity can be aggregated into one activity. Further, solutions to this challenge could be one of the future research projects after the first version of the proposed blueprint is successfully developed.

The system could be enhanced by integrating it with another solution that could monitor the progress of its execution autonomously and feed the proposed solution with the actual execution duration for each activity. This could provide precise future training data and replace site staff in monitoring and reporting on project progress.

Feeding the actual project execution cost and duration at the activity level could help enhance future system performance, a process that embodies a lesson-learned process.

Building information modeling (BIM) could be employed to identify safety risks arising from overlapping activities [41,42]. Thus, the proposed solution could be enhanced to cater to safety risks arising from overlapping activities, where mitigating these requires adjusting the project schedule. Further, identifying safety risks stemming from overlapping activities could be automated via integration into the autonomous construction project scheduling solution.

The solution should be cloud-based, where multiple users can access it. Having multiple users is important to retrieve real-life data for training data; this ensures the model is regularly updated, thus improving its performance. Integrating blockchain and BIM technologies could ensure data availability and reliability among all stakeholders [43].

Notably, the solution could be programmed to generate not only project durations but also individual activity costs, as well as a perpetual calculation of project performance indices and productivity during project execution. These details can aid project managers in identifying time and cost overruns early and in taking required corrective actions. By employing multi-objective linear programming, the project manager can make informed decisions to determine the optimal project cost and duration. Financial analyses could also be conducted based on progress payment percentages and users' financial positions.

6. Conclusions

This study proposed a concept for an autonomous construction project schedule solution that harnesses the power of artificial intelligence (AI) technology and leverages building information modeling (BIM) files to outline the sequence of activities. Although this paper proposed a synthesis of existing solutions, many details have been suggested as well as technical features of possible use in some existing solutions. Accordingly, the proposed blueprint could serve as a foundation for future development. The construction industry needs a technology provider who will take the lead and build a cloud-based solution similar to the proposed blueprint, where the construction industry could benefit from emerging technologies such as AI.

It is anticipated that the schedule resulting from this solution will yield an extensive array of activities, providing project managers with a tool to detect any delays swiftly. In addition, the proposed solution could have the ability to analyze real-time data (e.g., costs, productivity, and progress) and make adjustments to optimize schedules as needed. Furthermore, the model could be programmed to improve itself perpetually by utilizing completed projects as training data.

The proposed solution has the potential to improve project performance through proactive planning and effective risk management. Issues such as last-minute staffing needs, idle times, material shortages, and missing dependencies could be relegated to the past, whereas the integration of autonomous project scheduling solutions with other components of autonomous project management systems could overcome numerous challenges faced by project managers. Ultimately, this will lead to increased cost efficiency and fewer delays. Autonomous scheduling could transform the trajectory of the AEC industry and profoundly reshape the construction industry.

Author Contributions: Conceptualization, M.A.A.-S.; validation, A.A.B. and Z.A.; investigation, M.A.A.-S. and A.A.B.; writing—original draft, M.A.A.-S.; writing—review and editing, A.A.B. and Z.A.; Supervision, A.A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: Authors Mazen A. Al-Sinan and Zainab Aljaroudi were employed by the company Saudi Aramco. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

AEC	Architecture, Engineering, Construction
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
BIM	Building Information Modeling

CBR	Case-Based Reasoning
CNN	Convolutional Neural Network
ERP	Enterprise Resource Planning
GA	Genetic Algorithm
GAS	Graph-Based Automated Scheduling
GPT	Generative Pretrained Transformer
IFC	International Foundation Class
K-NN	K-Nearest Neighbors
LLM	Large Language Model
LR	Logistic Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
NOSQL	Not Only Structured Query Language
OEM	Original Equipment Manufacturer
Q&A	Question and Answer
RF	Random Forest
RNN	Recurrent Neural Network
RPC	Remote Procedure Call
SGE	Search Generative Experience
SME	Subject Matter Expert
SVR	Support Vector Regression
UPAMD	Universal Project Activity Master Data

References

1. Mikulakova, E.; König, E.; Tauscher, E.; Beucke, K. Knowledge-Based Schedule Generation and Evaluation. *Adv. Eng. Inform.* **2010**, *24*, 389–403. [\[CrossRef\]](#)
2. Kim, H.; Anderson, K.; Lee Sand Hildreth, J. Generating Construction Schedules Through Automatic Data Extraction Using Open BIM (Building Information Modeling) Technology. *Autom. Constr.* **2013**, *35*, 285–295. [\[CrossRef\]](#)
3. Saini, H.; Malik, U. Project management in construction using Primavera. *Int. J. Civ. Eng. Technol.* **2017**, *8*, 538–549. Available online: <http://www.iaeme.com/ijciet/issues> (accessed on 15 December 2023).
4. Singh, A.; Pal, A.; Kumar, P.; Lin, J.; Hsieh, S. Prospects of Integrating BIM and NLP for Automatic Construction Schedule Management. In Proceedings of 40th International Symposium on Automation and Robotics in Construction (ISARC 2023), Chennai, India, 3–9 July 2023. [\[CrossRef\]](#)
5. Faghihi, V.; Nejat, A.; Reinschmidt, K.F.; Kang, J.H. Automation in Construction Scheduling: A Review of the Literature. *Int. J. Adv. Manuf. Technol.* **2015**, *81*, 1845–1856. [\[CrossRef\]](#)
6. Amer, F.; Koh, H.; Golparvar-Fard, M. Automated Methods and Systems for Construction Planning and Scheduling: Critical Review of Three Decades of Research. *J. Constr. Eng. Manag.* **2021**, *147*, 03121002. [\[CrossRef\]](#)
7. Patel, D. Revolutionizing Project Management with Generative AI. *Int. Sci. J. Eng. Manag.* **2023**, *2*, 1–8. Available online: <https://isjem.com/download/revolutionizing-project-management-with-generative-ai> (accessed on 5 June 2023).
8. Al-Sinan, M.; Bubshait, A.; Al-Dossary, S. A Framework for Embracing Construction 4.0. In Proceedings of the 1st Int'l Conference on Challenges in Engineering, Medical, Economics and Education: Research & Solutions (CEMEERS-23), Lisbon, Portugal, 21–22 June 2023. [\[CrossRef\]](#)
9. Rajabi, M.S.; Rezaeiashtiani, M.; Radzi, A.R.; Famili, A.; Rezaeiashtiani, A.; Rahman, R.A. Underlying Factors and Strategies for Organizational BIM Capabilities: The Case of Iran. *Appl. Syst. Innov.* **2022**, *5*, 109. [\[CrossRef\]](#)
10. Demirkesen, S.; Tezel, A. Investigating major challenges for industry 4.0 adoption among construction companies. *Eng. Constr. Archit. Manag.* **2022**, *29*, 1470–1503. [\[CrossRef\]](#)
11. Rajabi, M.S.; Radzi, A.R.; Rezaeiashtiani, M.; Famili, A.; Rashidi, M.E.; Rahman, R.A. Key Assessment Criteria for Organizational BIM Capabilities: A Cross-Regional Study. *Buildings* **2022**, *12*, 1013. [\[CrossRef\]](#)
12. Azhar, S.; Ahmad, I.; Sein, M.K. Action research as a proactive research method for construction engineering and management. *J. Constr. Eng. Manag.* **2010**, *136*, 87–98. [\[CrossRef\]](#)
13. Khan, S.; Tzortzopoulos, P. Using design science research and action research to bridge the gap between theory and practice in lean construction research. In Proceedings of the 26th Annual Conference of the International Group for Lean Construction, Chennai, India, 18–22 July 2018; pp. 209–219.
14. Hong, Y.; Xie, H.; Agapaki, E.; Brilakis, I. Graph-Based Automated Construction Scheduling without the Use of BIM. *J. Constr. Eng. Manag.* **2023**, *149*, 05022020. [\[CrossRef\]](#)

15. Faghihi, V.; Reinschmidt, K.F.; Kang, J.H. Construction Scheduling Using Genetic Algorithm Based on Building Information Model. *Expert Syst. Appl.* **2014**, *41*, 7565–7578. [\[CrossRef\]](#)
16. Faghihi, V. Automated and Optimized Scheduling Using BIM. Ph.D. Thesis, Texas A&M University, College Station, TX, USA, 2014.
17. Wu, Z.; Ma, G. Automatic generation of BIM based construction schedule: Combining an ontology constraint rule and a genetic algorithm. *Eng. Constr. Archit. Manag.* **2023**, *30*, 5253–5279. [\[CrossRef\]](#)
18. ElMenshawhy, M.; Marzouk, M. Automated BIM Schedule Generation Approach for Solving Time–Cost Trade-off Problems. *Eng. Constr. Archit. Manag.* **2021**, *28*, 3346–3367. [\[CrossRef\]](#)
19. Naderi, H.; Shojaei, A.; Rokooei, S. Interface of Quantum Computation and Architecture, Engineering, and Construction (AEC) Industry. *Proc. Int. Struct. Eng. Constr.* **2023**, *10*, 1. [\[CrossRef\]](#)
20. Kaplan, A.; Haenlein, M. Siri, Siri, In My Hand: Who’s the Fairest in the Land? On the Interpretations, Illustrations, and Implications of Artificial Intelligence. *Bus. Horiz.* **2019**, *62*, 15–25. Available online: https://www.researchgate.net/publication/328761767_Siri_Siri_in_my_hand_Who%E2%80%99s_the_fairest_in_the_land_On_the_interpretations_illustrations_and_implications_of_artificial_intelligence (accessed on 4 November 2023). [\[CrossRef\]](#)
21. Allen, G.C. *Understanding Artificial Intelligence Technology*; DoD Joint AI Center: Washington, DC, USA, 2020.
22. Annoni, A.; Benczur, P.; Bertoldi, P.; Delipetrev, B.; De Prato, G.; Feijoo, C.; Fernandez, M.E.; Gomez, E.; Iglesias, M.; Junklewitz, H.; et al. *Artificial Intelligence A European Perspective*; Craglia, M., Ed.; EUR 29425 EN; Publications Office: Luxembourg, 2018.
23. Shah, C. *A Hands-on Introduction to Data Science*; Cambridge University Press: Cambridge, UK, 2020.
24. Morteza, A.; Sadipour, M.; Fard, R.S.; Taheri, S.; Ahmadi, A. A dagging-based deep learning framework for transmission line flexibility assessment. *IET Renew. Power Gener.* **2023**, *17*, 1092–1105. [\[CrossRef\]](#)
25. Sarkar, D. *Text Analytics with Python: A Practitioner’s Guide to Natural Language Processing*, 2nd ed.; Apress: Bangalore, India, 2019. [\[CrossRef\]](#)
26. Zhao, Y. The State-of-Applications of NLP: Evidence from ChatGPT. *Highlights Sci. Eng. Technol.* **2023**, *49*, 237–243. [\[CrossRef\]](#)
27. Hong, Y.; Xie, H.; Bhumbra, G.; Brilakis, I. Comparing Natural Language Processing Methods to Cluster Construction Schedules. *J. Constr. Eng. Manag.* **2021**, *147*, 04021136. [\[CrossRef\]](#)
28. Amer, F.; Golparvar-Fard, M. Formalizing Construction Sequencing Knowledge and Mining Company-Specific Best Practices from Past Project Schedules. In Proceedings of the ASCE International Conference on Computing in Civil Engineering, Atlanta, GA, USA, 17–19 June 2019; pp. 215–223. [\[CrossRef\]](#)
29. Amer, F.; Jung, Y.-H.; Golparvar-Fard, M. Construction Schedule Augmentation with Implicit Dependency Constraints and Automated Generation of Lookahead Plan Revisions. *Autom. Constr.* **2023**, *152*, 104896. [\[CrossRef\]](#)
30. Council of the European Union, General Secretariat. ChatGPT in the Public Sector—Overhyped or Overlooked? Analysis, and Research Team, April 2023. Available online: https://www.consilium.europa.eu/media/63818/art-paper-chatgpt-in-the-public-sector-overhyped-or-overlooked-24-april-2023_ext.pdf (accessed on 15 August 2023).
31. Jagdishbhai, N.; Thakkar, K.Y. Exploring the Capabilities and Limitations of GPT and ChatGPT in Natural Language Processing. *J. Manag. Res. Anal.* **2023**, *10*, 18–20. [\[CrossRef\]](#)
32. Rahaman, M.S.; Ahsan, M.M.T.; Anjum, N.; Terano, H.J.R. From ChatGPT-3 to GPT-4: A Significant Advancement in AI-Driven NLP tools. *J. Eng. Emerg. Technol.* **2023**, *1*, 50–60. [\[CrossRef\]](#)
33. Prieto, S.A.; Mengiste, E.T.; García de Soto, B. Investigating the Use of ChatGPT for the Scheduling of Construction Projects. *Buildings* **2023**, *13*, 857. [\[CrossRef\]](#)
34. Diouf, R.; Sarr, E.N.; Sall, O.; Birregah, B.; Bousso, M.; Mbaye, S.N. Web Scraping: State-of-the-Art and Areas of Application. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 6040–6042. [\[CrossRef\]](#)
35. Kerner, S.M. Google Search Generative Experience (SGE). Tech Target. Available online: <https://www.techtarget.com/whatis/definition/Google-Search-Generative-Experience-SGE> (accessed on 23 June 2023).
36. Lee, J.; Kang, L.-S. Generation of Construction Schedule Information from Activity Unit Classification of 3d Models in the Design Phase. Available online: <https://ssrn.com/abstract=4691646> (accessed on 23 September 2023).
37. Banihashemi, S.; Khalili, S.; Sheikhhoshkar, M.; Fazeli, A. Machine learning-integrated 5D BIM informatics: Building materials costs data classification and prototype development. *Innov. Infrastruct. Solut.* **2022**, *7*, 215. [\[CrossRef\]](#)
38. Al-Subhi Al-Harbi, K.; Selim, S.; Al-Sinan, M. A Multiobjective Linear Program for Scheduling Repetitive Projects. *Cost Eng.* **1996**, *38*, 41–44.
39. Singh, P.; Smarandache, F.; Chauhan, D.; Bhaghel, A. A Unit Based Crashing Pert Network for Optimization of Software Project Cost. *Int. J. Contemp. Math. Sci.* **2015**, *10*, 29.
40. Lotfi, R.; Kargar, B.; Gharehbaghi, A.; Afshar, M.; Rajabi, M.S.; Mardani, N. A data-driven robust optimization for multi-objective renewable energy location by considering risk. *Environ. Dev. Sustain.* **2022**. [\[CrossRef\]](#)
41. Rashidi Nasab, A.; Malekitabar, H.; Elzarka, H.; Nekouvaght Tak, A.; Ghorab, K. Managing Safety Risks from Overlapping Construction Activities: A BIM Approach. *Buildings* **2023**, *13*, 2647. [\[CrossRef\]](#)

42. Naderi, H.; Heydari, M.H.; Parchami Jalal, M. Risk Analysis in Implementing Building Energy Performance Projects: Hybrid DANP-VIKOR Model Analysis—A Case Study in Iran. *Buildings* **2023**, *13*, 2066. [[CrossRef](#)]
43. Shojaei, A.; Naderi, H. Blockchain Technology for a Circular Built Environment. In *A Circular Built Environment in the Digital Age*; De Wolf, C., Çetin, S., Bocken, N.M.P., Eds.; Circular Economy and Sustainability; Springer: Cham, Switzerland, 2024. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.