

## AI-ENHANCED DATA WORKFLOWS FOR ROAD INFRASTRUCTURE BIM: A PYTHON-VPL FRAMEWORK ENABLING 4D–7D INTEGRATION

Pietro SERRA<sup>1</sup>, Mattia INTIGNANO<sup>1</sup>, Salvatore Antonio BIANCARDO<sup>1\*</sup>,  
Francesco ABBONDATI<sup>2</sup>, Gianluca DELLACQUA<sup>1</sup>

<sup>1</sup>*Department of Civil, Construction and Environmental Engineering,  
Federico II University of Napoli, Napoli, Italy*

<sup>2</sup>*Department of Engineering, Pegaso Telematics University, Napoli, Italy*

Received 16 January 2026; revised 20 February 2026; accepted 22 February 2026

**Abstract.** The transition of road projects from a purely geometric, two-dimensional or three-dimensional configuration to one rich in information for sustainability-oriented management requires a multi-dimensional approach, i.e., exploiting the full potential of Building Information Modelling methodology. This work proposes an AI-backed BIM workflow which, by adopting the solutions offered by an AI agent, integrates Python scripts to automate analyses and calculations related to time-, cost-, and sustainability-related assessments. The methodology was validated on a case study of a suburban infrastructure, involving a road intersection and a hydraulic underpass. The IT tools used are those specified in the tender documentation, in particular Autodesk Civil 3D, Revit, and Navisworks software. In this environment, Dynamo was used to insert Python scripts into an environment that supports visual programming language. The result is models exported in open IFC 4.3 format. The approach adopted suggests that AI can be successfully exploited to increase automation levels in BIM processes. This leads to a reduction in human error, shorter timescales, and therefore lower costs, thanks to the acceleration of iterative processes and the reduction of repetitive tasks. Its application to computations and analyses supporting cost calculation, scheduling construction and work phases, and calculation of carbon footprint as an index of sustainability opens up to increasingly relevant uses in terms of environmental sustainability and project quality. The results suggest that the integration of AI, visual and textual programming methods, and established BIM-Authoring software can lead to innovative sustainability techniques and digital twin approaches in infrastructure management.

**Keywords:** I-BIM, AI Agent, Dynamo, Python, IFC.

### 1. Introduction

Building Information Modeling (BIM) enables digital processes optimization across the entire civil project life cycle (buildingSMART, n.d.). BIM has become increasingly popular as a tool in the construction industry (Xu et al., 2014) and integrates multi-disciplinary data, such as architectural, structural, mechanical context (Kunz & Fischer, 2020 for design support, management, and maintenance (Bryde et al., 2013; Zou et al., 2017; Love et al., 2013).

Moreover, BIM methodology is developing in various applications, as into linear infrastructures, where the terminology is modified in Infrastructure-BIM (I-BIM) (Cheng et al., 2016; McGraw-Hill Construction, 2012; Abbondati et al., 2020; Intignano et al., 2025). BIM is supported by open data formats, as the Industry

Foundation Classes (IFC), developed by buildingSMART International (bSI) (Ribeiro Antunes et al., 2024). The last IFC version, IFC 4.3, is dedicated to linear infrastructures and enhances interoperability, one of the key issues for exchanging data, by improving descriptive capabilities of I-BIM models (Yu et al., 2023).

The building information model of an asset serves as its graphical representation and information database, thus constituting a multidimensional model covering geometry (3D), time or scheduling (4D), costs estimation (5D), sustainability (6D), maintenance and governance (7D).

Therefore, this paper deals with optimized infrastructure modeling, structuring a dynamic methodology, including BIM dimensions and data interoperability. The methodology presented, can be integrated developing a semi-automated workflow enhanced by Artificial

\* Corresponding author. E-mail: [salvatoreantonio.biancardo@unina.it](mailto:salvatoreantonio.biancardo@unina.it)

Intelligence (AI), with AI agent support into Visual Programming Language (VPL). AI is a technology that is developing a pivotal role in contemporary research and technical development, by boosting automation and data-driven decision-making processes. By using a specific branch of AI, the large language models (LLMs), AI agents can be leveraged to support in building design, planning, and modeling (Yu et al., 2023; Zhou et al., 2024).

Thus, this works provides AI agent implementation, for producing generative AI through Dynamo scripts into BIM digital processes. For an efficient agent's implementation, LLMs require a specific and well-structured data training process. In this case, AI agent is configured with rules and instructions that enable it to be an expert in generating Dynamo scripts by Python language.

In the most recent research, BIM is integrated with Visual Programming Language (VPL) in Dynamo platform through AI systems for automating digital processes (Elghaish et al., 2022), 3D parametric modeling, and structural design optimization (Cascone & Sangiorgio, 2025).

Moreover, AI and BIM through the results application prove real benefits in several themes such as model detailing (Saad et al., 2024), time and cost analysis, sustainability and design coordination phases (Liu et al., 2023), and lifecycle integration (Elmousalami et al., 2025; Rane et al., 2023; Ahmadpanah et al., 2023).

Therefore, AI applied through AI agents develops Python code with VPL enables a dynamic and powerful workflow, as well as a sustainable one, reducing project completion times and streamlining the process with optimized calculation scripts.

## 2. Methodology and case study overview

This case study focuses on hydraulic and road infrastructure design using Autodesk platforms, i.e. Civil 3D, Dynamo, and Navisworks, and LLM, to boost automation and helping into digital processes. At first it is created a 3D infrastructure model linked to a Digital Terrain Model (DTM), designing horizontal and vertical alignments following engineering standards. The result is a combined infrastructure with roads, pipelines and canals. By the 3D

model the checking phase for clash detection is conducted.

Moreover, with this research a more powerful computational method is achieved by AI agents connected to Dynamo into Civil 3D 2024. AI chatbot is realized for programming Python codes for Dynamo.

This agent is set up using LangChain for prompts, Gradio for the interface and Ollama for running a local model which realizes specific codes. LangChain, in particular, is designed to build intelligent AI agents rather than simple chatbots. It enables the connection of LLMs to external tools such as files, databases, APIs, the web, and computational utilities. It also allows the creation of agents and the structuring of the reasoning process to be executed. In practice, with LangChain the model is able to reason and act. Ollama makes it possible to assess models in an offline environment, without relying on OpenAI or external APIs.

While Gradio provides a fast and efficient way to create web-based interfaces for interacting with the agent via a browser. It allows users to upload files and to visualize outputs in a clear and interactive manner. The user submits a Dynamo script through the agent's chat interface. After executing the script, the agent provides a response in natural language, and a dedicated Python script is generated to process and manipulate the Dynamo file accordingly.

The main goal is training an AI agent with Dynamo scripts examples to ensure a correct Python code useful for .dyn files realization. This reduces time for learning in depth how to use Visual Programming Language (VPL) that automates digital process. Agent's skills give, through a final report, model-based analysis, earthwork volume calculation, length of road alignment, road pavement layers cost analysis and emissions evaluation.

This approach makes the process more sustainable, so during digital design, BIM software users can modify geometric and parametric 3D infrastructure in function of chatbot global analysis model.

The result is an infrastructure 3D model, clash detection test, script computation, and AI chatbot which enhances workflow, avoiding issues, and more support for decision-making process.

Figure 1 shows the workflow adopted in this study.

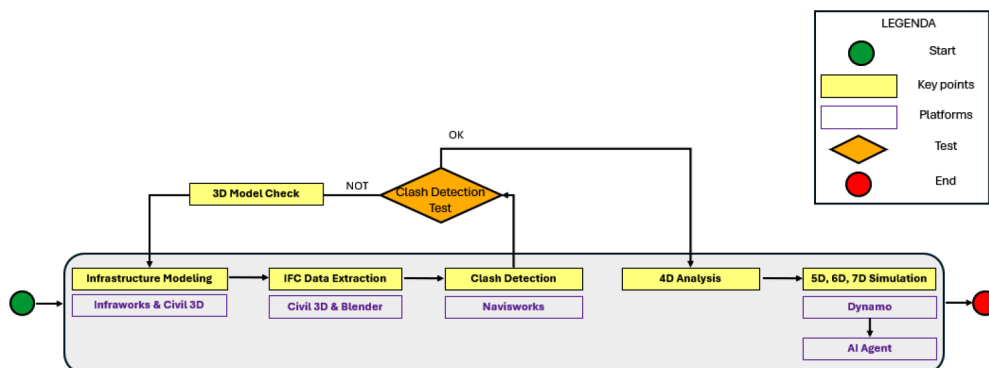


Figure 1. Methodology phases

## 2.1. Infrastructure modeling

The case study was conducted in the Tuscan countryside, near the city of Florence. The project involved two intersected suburban roads, along with an underground drainage system. For starting geometric design, the DTM was generated using satellite imagery in Autodesk InfraWorks through Model Builder function, employing the georeferenced coordinate system for Italy “UTM 32N with Europe 1950 datum, W of 12 deg E” (see Figure 2).

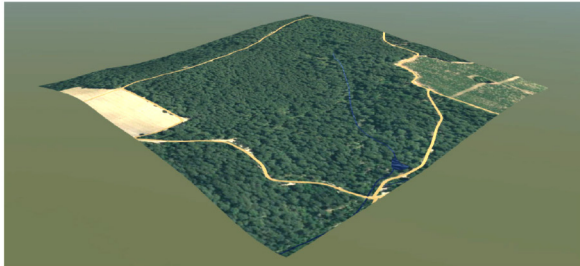


Figure 2. Digital Terrain model

The road geometry is developed using Autodesk Civil 3D following Italian road design standards. The project includes primary and secondary alignments defined according to coherent design parameters. After the horizontal and vertical profile, the corridor is created including pavement layers, with wearing course, binder, base and subbase, and shoulders, lateral slopes, and drainage elements (see Figure 3).



Figure 3. Geometric model

## 2.2. IFC data extraction

After geometric design in Civil 3D, the project is converted into the Industry Foundation Classes (IFC) open format 4.3 to ensure data interoperability. The IFC schema facilitates secure transfer and data exchange across different platforms. In this case study, the infrastructure model is exported by Civil 3D to open IFC file into Autodesk Navisworks platform.

## 2.3. Clash detection, 3D model check and 4D analysis

Clash Detection analysis was conducted on Autodesk Navisworks platform to verify interferences among geometric elements. Initially there was a significant conflict between the top of the box girder and the pavement layers. Consequently, the design was revised to resolve the conflict with the roadway for reducing earthworks.

The solution limits excavation volumes and ensures road structural continuity (see Figure 4).

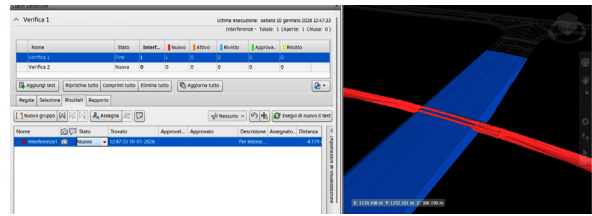


Figure 4. Geometric model

Subsequently the 4D BIM simulation is conducted in Navisworks, where building activity was connected to schedule and plan of each element. This provides a visualization of construction progress along the alignment and detects potential spatial conflicts among teams and equipment (see Figure 5).

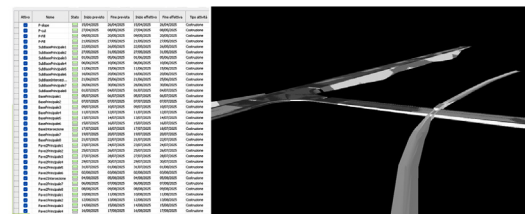


Figure 5. 4D simulation

The infrastructure works scheduling required first to organize modeled elements into categories in order to build sub-phases. Therefore, the main and the secondary alignments were segmented into 100-meter sections. The planning consisted of establishing a direct correspondence between individual construction activities and the 3D solids generated in Civil 3D.

## 2.4. 5D, 6D and 7D simulations

For each pavement layer, the corresponding material volumes can be automatically computed through dedicated Dynamo scripts (see Figure 6) and subsequently associated with unit costs (see Figure 7).

The workflow starts by referencing the active Civil 3D document and the related corridor model, which is generated from defined baselines and cross-sections. During the design phase, Dynamo scripts were employed to extract the solids representing the pavement layers within the road corridor. For each extracted solid, material volumes were calculated and pavement costs were estimated by linking quantities to official unit prices, expressed per cubic meter ( $m^3$ ), according to the relevant price list for each layer.

This approach provides a dynamic and responsive cost-estimation solution, as it enables automatic updating of quantity take-offs and cost results in response to any geometric modification of the corridor. Furthermore, it offers a straightforward method for isolating and evaluating specific portions of the infrastructure

(e.g., left-side layers only), thereby improving control over the final cost assessment.

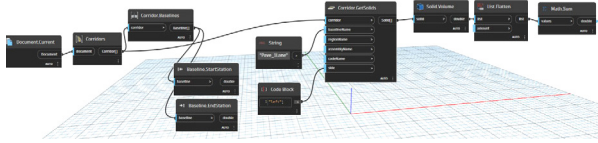


Figure 6. Graph on material volumes calculation

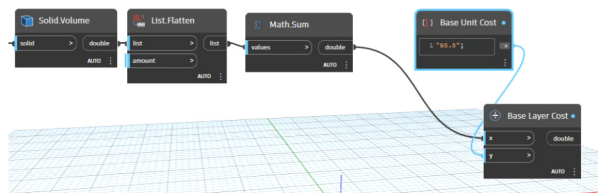


Figure 7. Graph on unit costs

The 7D BIM simulation is implemented through an automated calculation framework aimed at supporting the selection of optimal paving materials and extending the life cycle of the pavement superstructure, thereby mitigating the aging effects affecting pavement layers over time.

This approach is also closely related to the 6D BIM dimension, where sustainability-driven criteria enable the evaluation of alternative solutions with the objective of reducing environmental impacts while considering both material properties and associated costs.

### 2.5. AI Agent

The AI agent represents a key enabling technology for automating the workflow proposed in this study, supporting the decision-making process, as well as volume estimation, cost analysis, and sustainability assessment. The agent is implemented as an AI chatbot based on a Large Language Model (LLM). This constitutes a machine learning application in which the LLM is employed as a conversational agent capable of learning from input information and providing explanations to the user in natural language through a text-based interface.

In this research, the AI agent was developed using LangChain for reasoning and orchestration, Gradio to provide a user-friendly chatbot interface, and Ollama to run the language model locally. These tools form the software stack required to implement an expert agent in the Python programming language, enabling the system to generate correct and functional algorithms when needed.

Since Python is a core requirement within the Dynamo environment, the AI agent was trained using a dataset of .dyn files, enabling it to generate custom scripts. The workflow produces outputs that support the user throughout the project by providing pre-configured, ready-to-use Dynamo scripts. These scripts require only

the connection of nodes to the corresponding geometric elements within the project file.

The proposed agent is based on open-source technologies and can therefore be distributed and published. For the initial configuration, the chatbot is launched via a Windows PowerShell command.

To set up the container hosting all components required for correct agent operation, a main project folder must be created and organized into dedicated subfolders. One subfolder stores the input files used to train the model, another contains the Python script (.py) defining the agent logic and instructions, and a third subfolder is reserved for the output files, namely the generated Dynamo scripts (.dyn) (see Figure 8).

```

agent_training.py
File Modifica Visualizza

import os
import json
from datetime import datetime
import gradio as gr

from langchain.llm import Ollama
from langchain.prompts import PromptTemplate
from langchain.chains import LLMChain

# ===== PERCORSI =====
BASE_DIR = r"C:\agentesat_c3d"
OUTPUT_FOLDER = os.path.join(BASE_DIR, "dynamo_scripts", "output")
os.makedirs(OUTPUT_FOLDER, exist_ok=True)

# ===== PROMPT LLM =====
template = """
Sei un agente AI esperto in Dynamo per Civil 3D.
Ricevi una richiesta dall'utente e genera il contenuto JSON di un file .dyn funzionante che risponde a quella richiesta.
Rispondi solo con il contenuto del file .dyn (JSON), senza spiegazioni aggiuntive.

Richiesta utente: {input}
...

prompt = PromptTemplate(
    input_variables=["input"],
    template=template,
)

llm = Ollama(model="codellama:7b-instruct") # Sostituisci con il modello Ollama che preferisci
chain = LLMChain(llm=llm, prompt=prompt)

# ===== CORE =====
def generate_dyn_with_llm(user_input):
    try:
        dyn_content = chain.run(input=user_input)
        # Verifica che il contenuto sia JSON valido
        dyn_json = json.loads(dyn_content)
        timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
        output_name = f"AI_{timestamp}.dyn"
        output_path = os.path.join(OUTPUT_FOLDER, output_name)
        with open(output_path, "w", encoding="utf-8") as f:
            json.dump(dyn_json, f, indent=2)
        return f"File Dynamo generato correttamente:\n{output_path}"
    except Exception as e:
        return f"ERRORE nella generazione del file Dynamo:\n{str(e)}"

# ===== UI =====
def chatbot_interface(user_input):
    return generate_dyn_with_llm(user_input)

if __name__ == "__main__":
    print("Avvio agente AI Dynamo (generazione autonoma)")
    iface = gr.Interface(
        fn=chatbot_interface,
        inputs="text",
        outputs="text",
        title="Agente AI Dynamo - Civil 3D (Generazione autonoma)",
        description="L'agente genera file Dynamo .dyn su richiesta, sfruttando un modello LLM locale tramite Ollama."
    )
    iface.launch(server_name="127.0.0.1", server_port=None, debug=True)
    
```

Figure 8. .py file coding for training agent

It is important to indicate the correct denomination of folders and files that will be recalled by the agent when it works while user accesses commands prompt. After some procedural steps on Windows Power Shell an access URL to the agent web interface is generated (see Figure 9).

```

* Running on local URL: http://127.0.0.1:7860
* To create a public link, set 'share=True' in 'launch()'.
    
```

Figure 9. URL access to AI agent through Windows Power Shell

Upon successful access, specific requests may be submitted in the chat interface. In this case it is required to create Dynamo scripts for extracting volumes from geometric elements, unit cost and total volume of each material pavement.

Therefore, the process for developing the LLM is illustrated below (see Figure 10).

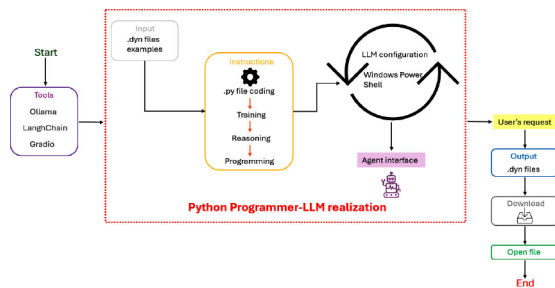


Figure 10. AI Agent realization framework

Finally, the project can be exported through the IFC 4.3 Extension for Autodesk Civil 3D application to complete the BIM workflow.

### 3. Conclusions

This paper describes a semi-automated workflow developed in accordance with the I-BIM methodology. The proposed approach aims to implement a streamlined digital process by combining BIM platforms with AI integration. The workflow can be considered standardized, as it follows the main phases of the infrastructure digital process, and is further enhanced through the application of LLMs, specifically via a Python programmer agent.

The digital process begins with (DTM) extraction, followed by the geometric design phase, which is applied to the modelling of an infrastructure project located in the Tuscany region. Interoperability is ensured by exporting the project into the IFC open standard format. Subsequently, a clash detection analysis is performed and can be successfully completed after the validation of the 3D model.

In addition, a 4D simulation is carried out using Autodesk Navisworks, enabling the scheduling of construction activities for each geometric element. 5D, 6D, and 7D simulations are instead managed through algorithms developed in Dynamo, supported by the proposed AI agent.

Furthermore, through the LLM-based Python programmer agent, the user is assisted during the 5D, 6D, and 7D stages, resulting in a more efficient and enhanced workflow. The developed technology allows users to directly access and use .dyn scripts for infrastructure design tasks, such as volume calculation, cost estimation of road pavement materials, and sustainability assessment.

Overall, the results demonstrate a standardized workflow based on Autodesk platforms, while also providing an innovative enhancement through AI integration within Dynamo. The agent is trained to generate .dyn scripts, and, based on the provided input files, it can read, simplify, modify, and produce ready-to-use Dynamo workflows. This approach particularly benefits users with limited experience in Dynamo, reducing manual effort and significantly saving time.

However, the proposed framework currently represents only a semi-automated implementation of the digital process, and further improvements may enable a higher level of automation. At present, no solution has been developed to allow the agent to operate fully autonomously inside the Dynamo environment, independently generating custom scripts without user guidance. Finally, it is important to highlight that developing such an agent is not a trivial task, as it requires advanced programming expertise and strong organizational capabilities to properly structure and train the model. Nevertheless, once implemented, the agent enables a substantial acceleration in the production and compilation of Dynamo scripts.

### References

- Ahmadpanah, H., Haidar, A., & Latifi, S. M. (2023). BIM and machine learning (ML) integration in design coordination: Using ML to automate object classification for clash detection. In *Proceedings of the International Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe)* (Vol. 2, pp. 619–628). CumInCAD. <https://doi.org/10.52842/conf.ecaade.2023.2.619>
- Abbondati, F., Biancardo, S. A., Sicignano, G., Guerra de Oliveira, S., Tibaut, A., & Dell'Acqua, G. (2020). BIM parametric modelling of a railway underpass. *Ingegneria Ferroviaria*, 75(6), 443–459.
- buildingSMART. (n.d.). <https://www.buildingsmart.org>
- Bryde, D., Broquetas, M., & Volm, J. M. (2013). The project benefits of building information modelling (BIM). *International Journal of Project Management*, 31(7), 971–980. <https://doi.org/10.1016/j.ijproman.2012.12.001>
- Cheng, J. C., Lu, Q., & Deng, Y. (2016). Analytical review and evaluation of civil information modeling. *Automation in Construction*, 67, 31–47. <https://doi.org/10.1016/j.autcon.2016.02.006>
- Cascone, S., & Sangiorgio, V. (2025). Sustainable architecture: Computational modeling of green roofs through BIM and Dynamo VPL integration. In R. Corrao, T. Campisi, S. Colajanni, M. Saeli, & C. Vinci (Eds), *Proceedings of the 11<sup>th</sup> International Conference of Ar.Tec. Colloqui.AT.e 2024. Lecture Notes in Civil Engineering* (Vol. 612). Springer. [https://doi.org/10.1007/978-3-031-71867-0\\_13](https://doi.org/10.1007/978-3-031-71867-0_13)
- Elghaish, F., Chauhan, J. K., Matarneh, S., Pour Rahimian, F., & Hosseini, M. R. (2022). Artificial intelligence-based voice assistant for BIM data management. *Automation in Construction*, 140, Article 104320. <https://doi.org/10.1016/j.autcon.2022.104320>
- Elmousalami, H., Maxy, M., Hui, F. K. P., & Aye, L. (2025). AI in automated sustainable construction engineering management. *Automation in Construction*, 175, Article 106202. <https://doi.org/10.1016/j.autcon.2025.106202>
- Intignano, M., Biancardo, S. A., Guerra de Oliveira, S., Tibaut, A., & Dell'Acqua, G. (2025). Semantic-driven automation of BIM for stone-paved roads: An ontology-VPL integrated approach. *Journal of Computational and Cognitive Engineering*, 5(1). <https://doi.org/10.47852/bonviewJCCCE52025931>
- Kunz, J., & Fischer, M. (2020). Virtual design and construction. *Construction Management and Economics*, 38(4), 355–363. <https://doi.org/10.1080/01446193.2020.1714068>

- Love, P. E. D., Simpson, I., Hill, A., & Standing, C. (2013). From justification to evaluation: Building information modeling for asset owners. *Automation in Construction*, 35, 208–216. <https://doi.org/10.1016/j.autcon.2013.05.008>
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9), Article 195. <https://doi.org/10.1145/3560815>
- McGraw-Hill Construction. (2012). *The business value of BIM for infrastructure*. [https://images.autodesk.com/adsk/files/business\\_value\\_of\\_bim\\_for\\_infrastructure\\_smartmarket\\_report\\_2012.pdf](https://images.autodesk.com/adsk/files/business_value_of_bim_for_infrastructure_smartmarket_report_2012.pdf)
- Ribeiro Antunes, M. L., Lenz Cèsar Junior, K. M., Lopes Ribeiro, J. C., Silva de Oliveira, D., Franco de Carvalho, J. M., & (2024). Analysis of IFC interoperability data schema for project representation. *Automation in Construction*, 166, Article 105650. <https://doi.org/10.1016/j.autcon.2024.105650>
- Rane, N., Choudhary, S., & Rane, J. (2023). *Integrating building information modelling (BIM) with ChatGPT, Bard, and similar generative artificial intelligence in the architecture, engineering, and construction industry: Applications, a novel framework, challenges, and future scope*. SSRN. <https://doi.org/10.2139/ssrn.4645601>
- Saad, S., Haris, M., Ammad, S., & Rasheed, K. (2024). AI-assisted building design. In *AI in material science* (pp. 143–168). CRC Press. <https://doi.org/10.1201/9781003438489-7>
- Xu, X., Ma, L., & Ding, L. (2014). A framework for BIM-enabled life-cycle information management of construction project. *International Journal of Advanced Robotic Systems*, 11(8). <https://doi.org/10.5772/58445>
- Yu, Y., Kim, S., Jeon, H., & Koo, B. (2023). A systematic review of the trends and advances in IFC schema extensions for BIM interoperability. *Applied Sciences*, 13(23), Article 12560. <https://doi.org/10.3390/app132312560>
- Zou, Y., Kiviniemi, A., & Jones, S. W. (2017). A review of risk management through BIM and BIM-related technologies. *Safety Science*, 97, 88–98. <https://doi.org/10.1016/j.ssci.2015.12.027>
- Zhou, L., Huang, Y., Yan, T., Li, D., Cai, H., Pan, L., & Shi, J. (2024). Research on the automatic BIM modeling method of substation based on LLM agent. In *Proceedings of the 6th International Conference on Artificial Intelligence and Advanced Manufacturing* (pp. 686–692). The Institution of Engineering and Technology. <https://doi.org/10.1049/icp.2024.4301>