



---

# 06.

# CHAPTER 06

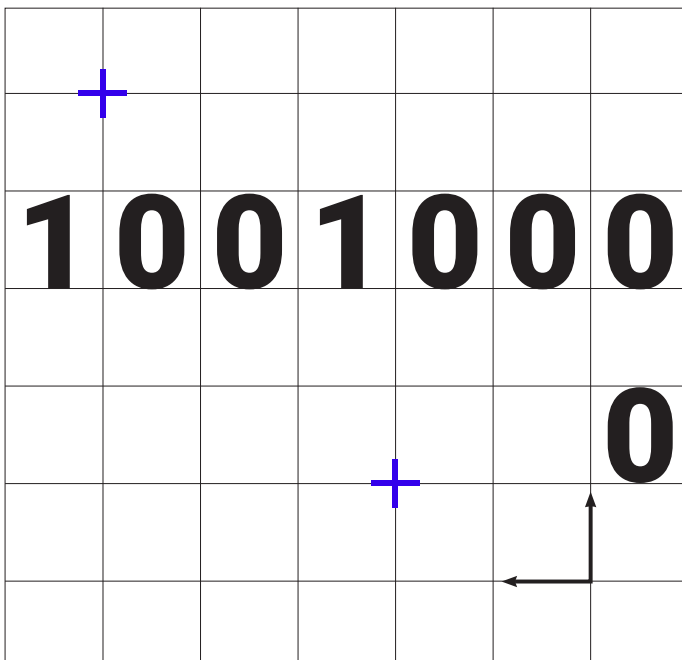
---

## Integration of AI and LLMs for Enhanced BIM Analysis: A Study on .IFC Model Interpretation

---

Groenmeyer Leon, Grunwald Gregor

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
|   |   |   |   |   |   |   |
|   | + |   |   |   |   |   |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   | 0 |
|   |   |   | + |   |   |   |
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |



# Introduction

The digital era has ushered in a transformative wave across various industries, with the construction sector witnessing a paradigm shift through the integration of Building Information Modeling (BIM). The construction industry is experiencing a significant push for digital transformation, driven by the potential for increased efficiency and cost savings recognized by both governments and private enterprises (**Grunwald, Kawasaki, Hanke, & Eidam, 2023**). This push is predominantly characterized by the enhanced precision and efficiency in project management and execution. BIM, a digitized approach to architectural and engineering design, facilitates detailed 3D modeling and data management throughout the construction lifecycle. A pivotal aspect of BIM is the Industry Foundation Classes (IFC) file format, which serves as a neutral data format to describe, exchange, and share information typically required for construction and facility management. While various data formats exist in the construction industry, the Industry Foundation Class (IFC) is particularly notable. It is widely recognized for its ability to capture building semantics and object properties along with the full 3D geometry (**Zabin et al., 2022**). Despite its comprehensive nature, the IFC format presents a significant challenge in terms of accessibility and interpretability, particularly for those lacking specialized software or technical expertise. This gap hinders the broader application and potential impact of BIM, confining its utility to a limited professional community. The entire AUFLADEN project (**Grunwald, 2023**) aims to fill this gap, with this study as a partial outcome of the project focusing on the aspect of using Chat GPT 4.0 focused on interpreting and analysing IFC files.

The research question at the core of this investigation is how AI, particularly LLMs, can be utilized to decode and interact with the complex data encoded in .IFC files. The goal is to explore the extent to which AI can interpret IFC files, enabling intuitive, conversational inquiries about BIM models, such as querying specific details like the number of columns in a project, directly from the IFC data. BIM models are essentially highly structured representations of a well-defined data schema (**Elghaish et al., 2022**). This approach could revolutionize the way stakeholders interact with BIM models, democratizing access to critical data and insights. The increasing synergy between BIM and optimization techniques underscores the potential for enhanced functionalities, such as automatic generation, visualization, and simulation of optimized solutions and layout schemes (**Pan & Zhang, 2021**).



---

# Integration of AI and LLMs for Enhanced BIM Analysis: A Study on .IFC Model Interpretation

---

Understanding and interpreting BIM data requires expertise and familiarity with specialized software, a barrier that limits its accessibility and practicality for broader audiences. According to Hanke, Kawasaki, and Grunwald (2023), “a high level of technical understanding and basic design knowledge is required to be able to transfer the described process [...]”. The incorporation of LLMs into this realm presents an opportunity to transcend these limitations, offering a more user-friendly and accessible interface. “Representation of all the information needed to describe buildings throughout the whole design, construction and management process has long been an objective for those applying information technology in building” (Howard & Björk, 2008). By harnessing the advanced capabilities of AI in natural language processing, this research aims to explore how AI can currently interact with IFC files, allowing complex BIM data to be queried and understood through simple, everyday language. This approach not only simplifies the interaction with BIM models but also opens doors for interdisciplinary collaboration, enabling architects, engineers, construction managers, and even clients to engage with BIM data more effectively. While this study is a small step, it contributes to a broader effort towards making BIM data more inclusive and collaborative, and thus complements previous studies by the authors (Grunwald, 2024). Integrating BIM with generative AI is set to transform design approaches in the AEC industry. With the advancement of AI algorithms, architects and designers will be able to explore unique and innovative design solutions, expanding the limits of creativity (Rane, Choudhary, & Rane, 2023).

**You**

Column.ifc  
File

Analyze the .ifc file's code concerning the columns. Interpret the code concerning columns and tell me what information you can retrieve from it.

**BIM Insight**

To analyze and interpret the code concerning columns in the provided .ifc file, I will extract and examine the relevant sections of the file. This will involve looking for instances of IFC entities related to columns, such as `‘IfcColumn’`, `‘IfcColumnType’`, and any related geometrical or material data. After this analysis, I'll provide insights into the properties, dimensions, materials, and other relevant characteristics of the columns as defined in the .ifc file. Let's begin the analysis.

Error analyzing ▾

It appears that the `‘ifcopenshell’` module is not available in my current environment, which prevents me from directly analyzing the .ifc file's code. However, I can guide you through the process of analyzing an .ifc file for information about columns.

In an .ifc file, columns are typically represented by the `‘IfcColumn’` entity. Here are the steps you would follow to analyze column data:

- Identify `‘IfcColumn’` Entities:** Search for instances of `‘IfcColumn’` in the .ifc file. Each instance represents a column in the building model.
- Extract Column Properties:** For each `‘IfcColumn’` entity, extract its properties such as:
  - `‘GlobalId’`: A unique identifier for the column.
  - `‘Name’`: The name or description of the column.
  - `‘ObjectType’`: The type of column (e.g., structural, architectural).
- Geometric Representation:** Columns have geometric representations (`‘Representation’`) detailing their shape and size. This could include:
  - `‘IfcExtrudedAreaSolid’`: Common for straight columns, representing the extrusion of a 2D shape (like a circle or rectangle) along a path.
  - `‘IfcSweptDiskSolid’`: Used for columns with circular cross-sections.
- Material Information:** The `‘IfcRelAssociatesMaterial’` relationship can provide material information for columns, such as concrete, steel, or timber.
- Positioning:** The `‘IfcLocalPlacement’` entity defines the column's location and orientation within the building.
- Related Elements:** Columns might be connected to other elements like beams, slabs, or foundations. These relationships can be found through

Figure 1: ChatGPT Interface for Analyzing .ifc File (Developed by Author).

## Methodological Approach

This study exclusively utilized ChatGPT 4.0 to investigate the potential of AI in interpreting IFC files, focusing on iterative refinement to improve the model's capability in understanding and processing BIM data. The methodology involved several steps, each aimed at enhancing the interaction between the AI model and the BIM data encoded in IFC files.

### *Initial Experimentation with ChatGPT 4.0*

The process began by providing ChatGPT with an IFC 4 file containing a single column. This initial phase was designed to test the model's ability to understand and extract information from a basic IFC file. The IFC (Industry Foundation Classes) format is a standardized data model used to describe building and construction industry data. Given its complexity and detailed structure, interpreting IFC files requires specialized tools and software. Upon receiving the IFC file, ChatGPT attempted to utilize the **'ifcopenshell'** module to parse and interpret the data. **'ifcopenshell'** is an open-source Python library specifically designed for reading and writing IFC files. It provides a comprehensive suite of tools for manipulating the data structures within these files, making it a popular choice among BIM professionals for programmatically handling IFC data. Typically, 'ifcopenshell' is installed using the 'pip' package manager in Python 3 environments, allowing for seamless integration with Python-based workflows.

However, ChatGPT operates within a constrained environment that does not support the direct execution of external libraries like **'ifcopenshell'**. This limitation prevented the model from fully leveraging the capabilities of **'ifcopenshell'** to parse the IFC file. Instead of executing code, ChatGPT relies on its pretrained knowledge and language processing abilities to understand and generate text-based responses.

This step highlighted a significant limitation: while ChatGPT can provide theoretical and syntactical guidance on using libraries like **'ifcopenshell'**, it cannot directly execute or test code that depends on these external libraries. This restriction necessitated a different approach to enable the AI to interact meaningfully with IFC data. Understanding this constraint was crucial for shaping the subsequent methodological steps and finding alternative ways to provide the necessary data to ChatGPT.

### **Manual Provision of IFC Data Structures**

To address this limitation, a manual approach was undertaken to equip ChatGPT with the necessary information to understand IFC files. BuildingSMART, the organization responsible for the development and maintenance of the IFC standard, provides extensive documentation detailing the structure and semantics of IFC data. This documentation includes definitions, data schemas, and examples for various IFC entities, such as columns, beams, and walls. “Before examining IFC as a data schema specified for BIM data representation and exchange, it is important to understand the contents that need to be transferred between applications while maintaining meaning” (**Huang, Ninić, & Zhang, 2021**).

All necessary data structures and specific information about columns were extracted from the BuildingSMART documentation. This included detailed descriptions of the **'IFCCOLUMN'** entity, its attributes, relationships, and typical usage within an IFC file. Additionally, information on related entities and how they interact with columns in a BIM model was compiled. This comprehensive set of documents was then converted into a single, consolidated PDF.

The rationale behind creating this PDF was to circumvent the AI's inability to execute code by providing it with detailed, structured information that it could process. By uploading this PDF to ChatGPT, the goal was to leverage the model's natural language processing capabilities to interpret the information and apply it to the analysis of IFC files. Upon uploading the document, ChatGPT was prompted with queries related to the IFC data structures and the specific attributes of columns.

This approach aimed to enable the AI to utilize its extensive pretrained knowledge in conjunction with the provided documentation to offer insights and analyses on the IFC file's contents. The creation and utilization of this PDF represented a critical step in the methodology, allowing for a workaround that provided ChatGPT with the contextual understanding required to interpret the complex data encoded in IFC files. This method also highlighted the flexibility and potential of AI in processing and understanding structured technical documentation.

## **Code-Based Interaction**

Next, the approach involved copying and pasting the code of an IFC file directly into ChatGPT and querying the model to determine the number of columns. This method was chosen to bypass the limitations encountered with the *'ifcopenshell'* module and to utilize ChatGPT's ability to process and analyze text-based data.

The IFC file format is structured in a way that each element within the BIM model is represented by specific entities. For columns, this is typically the *'IFCCOLUMN'* entity. By examining the raw text of the IFC file, it was possible to look for instances of the *'IFCCOLUMN'* entity to determine the number of columns present. ChatGPT was provided with the code of the IFC file in plain text format. The query was framed to instruct the model to identify and count all occurrences of the *'IFCCOLUMN'* entity within the provided code. This approach leveraged ChatGPT's text analysis capabilities to parse through the file and locate the relevant data points.

GPT-4 has demonstrated its ability to formulate design spaces, set objectives, and define constraints. It can also select suitable search algorithms for given problems, highlighting its utility as a foundational component in creating inverse design systems (*Makatura et al., 2023*).

## **Utilizing Visual Studio Code and GPT Plugin**

To advance the investigation, Visual Studio Code was employed along with user-created GPTs from the GPT Store, particularly the Python GPT by Nicholas Barker. These tools facilitated the use of *'ifcopenshell'* in a Python 3 environment, providing a robust framework for developing and executing Python scripts to manipulate IFC files. The setup involved configuring Visual Studio Code, a versatile code editor chosen for its extensive support for Python development, including features like IntelliSense, linting, and debugging.

---

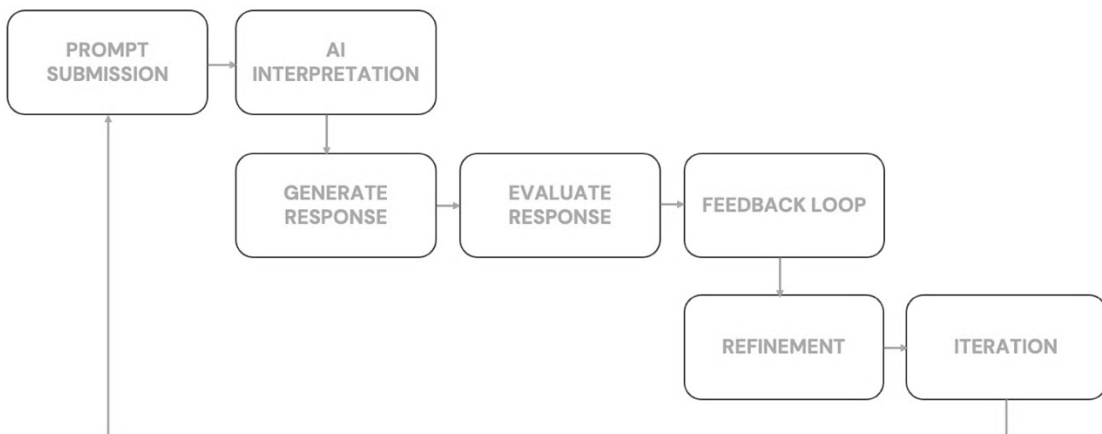
## Integration of AI and LLMs for Enhanced BIM Analysis: A Study on .IFC Model Interpretation

---

The Python extension was installed to further enhance these capabilities. Integrating the Python GPT by Nicholas Barker allowed for enhanced code suggestions and debugging tips, providing valuable assistance throughout the development process.

The primary task was to develop a Python script that could create a column in an IFC file with specified dimensions and material properties. This development began by importing *'ifcopenshell'* and setting up the script to create an IFC model. The script defined the column's dimensions, such as a base of 30 cm x 30 cm and a height of 3 meters, and specified it as being constructed from reinforced concrete.

The script's development process was iterative, involving multiple revisions based on feedback and errors encountered during execution. Each iteration involved running the script in the Visual Studio Code terminal, identifying issues through error messages and debugging tools, and consulting ChatGPT for suggestions on how to fix these issues. This feedback was then incorporated to refine the script further.



**Figure 2:** Iterative Process of AI-Assisted BIM Data Analysis *(Developed by Author)*.

The iterative refinement ensured that each version of the script addressed the previously encountered issues, aiming to improve functionality and accuracy. This process continued until the script executed successfully, creating an IFC file with the defined column.

This methodical approach, combining the capabilities of Visual Studio Code and AI assistance, ensured thorough testing and refinement of the script for creating IFC files. By leveraging these advanced tools and iterative development, the investigation was able to systematically address the complexities involved in handling BIM data programmatically.

## **Methodological Insights and Conclusion**

The methodological approach adopted in this study provided valuable insights into the potential and limitations of using ChatGPT 4.0 for interpreting IFC files. Each phase of the methodology contributed to understanding how AI can be integrated with BIM data analysis, revealing both challenges and incremental progress. The initial experimentation phase highlighted the constraints of ChatGPT's environment, particularly its inability to execute external libraries like 'ifcopenshell'. This limitation necessitated alternative strategies for enabling the AI to process and interpret complex IFC data. The manual provision of data structures via a comprehensive PDF underscored the importance of providing contextual information to enhance the AI's understanding of BIM models.

Copying and pasting IFC file code directly into ChatGPT demonstrated the model's capacity for text-based analysis, though within certain limitations. This approach emphasized the need for precise query framing to effectively utilize ChatGPT's natural language processing abilities for identifying and counting specific entities within the IFC data. Integrating Visual Studio Code and user-created GPTs facilitated the development and iterative refinement of a Python script to create a column in an IFC file. This setup, combined with the real-time assistance from the Python GPT, enabled a systematic approach to troubleshooting and improving the script. Each iteration provided valuable feedback, contributing to the refinement of the script and leading to the successful creation of an IFC file with the specified column.

This study's methodological approach highlights the potential of combining AI with traditional programming tools for BIM data analysis. The iterative steps taken in this investigation demonstrate a practical framework for addressing the challenges associated with AI-driven BIM data interpretation.

## Findings

The findings from this investigation reveal several critical insights into the capabilities and limitations of using ChatGPT 4.0 for interpreting IFC files, as well as the effectiveness of various methodologies employed to enhance these capabilities.

### *Initial Experimentation with ChatGPT 4.0*

The initial attempts to use ChatGPT for direct analysis of IFC files revealed several key insights into the model's capabilities and limitations. Firstly, the constrained environment in which ChatGPT operates significantly limited its ability to handle complex BIM data. The inability to execute external libraries such as 'ifcopenshell' was a major obstacle, highlighting the need for alternative methods to facilitate meaningful interaction with IFC data. Despite these challenges, the experiment demonstrated that ChatGPT could still provide valuable theoretical and syntactical guidance on using libraries like 'ifcopenshell'. This insight emphasized the model's strength in offering conceptual advice, even if it could not execute the code directly. The process of enriching BIM models semantically involves using algorithms that leverage expert knowledge to infer necessary information for specific applications that are not explicitly present in the model's data (Sacks et al., 2020).

Moreover, the experimentation phase underscored the importance of developing strategies to work around the constraints of ChatGPT's environment. The need for indirect methods of data provision, such as manual data provisioning or converting data into a format interpretable by the model's natural language processing capabilities, became evident. This phase was instrumental in identifying the boundaries within which ChatGPT operates and highlighted the necessity for integrating additional tools and manual interventions to enable effective BIM data analysis. Understanding these constraints early on allowed for a more targeted approach in subsequent phases, ensuring that the AI's capabilities were utilized to their fullest potential within the given limitations.

Overall, the findings from this phase provided a deeper understanding of the intricacies involved in BIM data parsing and the specific requirements for effective analysis. By recognizing the limitations of ChatGPT's environment, the study was able to pivot towards alternative methods that combined AI capabilities with traditional programming tools. This foundational understanding was essential for the iterative development and refinement processes that followed, ultimately contributing to the study's overall aim of enhancing BIM data accessibility and usability.

## **Manual Provision of IFC Data Structures**

To address the identified constraints, an extensive PDF was created, consolidating detailed information about IFC data structures from BuildingSMART documentation. This PDF included specific details about the **'IFCCOLUMN'** entity, covering all its relations and subcategories. It provided comprehensive descriptions of attributes, relationships, and typical usage within an IFC file, along with extensive information on related entities and their interactions with columns in a BIM model. The process of creating this PDF involved meticulously extracting and organizing data directly from BuildingSMART, the organization responsible for the development and maintenance of the IFC standard. BuildingSMART's documentation is highly detailed, comprising hundreds of pages that outline the structure, hierarchy, and relationships of various IFC entities. The PDF compiled from this documentation served as an exhaustive reference, ensuring that ChatGPT had access to the full context necessary for accurate interpretation of IFC data.

When this detailed document was uploaded to ChatGPT, the AI model was prompted with specific queries related to the IFC data structures and the attributes of columns. The goal was to leverage ChatGPT's natural language processing capabilities to interpret the provided information and apply it to the analysis of IFC files. Despite acknowledging the detailed data, ChatGPT struggled to generate actionable analysis or meaningful insights from the document. This outcome indicated that while providing structured information can enhance contextual understanding, it does not fully overcome the limitations of text-based analysis for complex data structures like IFC files.

This finding highlighted the model's limitations in handling complex data, even when detailed contextual information was provided. It became clear that while ChatGPT could process and understand the text, it lacked the capability to perform the sophisticated data parsing and analytical tasks required for effective BIM analysis. This limitation necessitated the exploration of alternative strategies to enable more meaningful interaction with IFC data. Furthermore, this phase of the study underscored the importance of not just providing detailed information but also ensuring that the format and presentation of this information are conducive to the AI's processing abilities. The comprehensive PDF served as a critical experiment in understanding how to best present complex data to ChatGPT, revealing that traditional text-based formats might not be sufficient for advanced data interpretation tasks.

---

## Integration of AI and LLMs for Enhanced BIM Analysis: A Study on .IFC Model Interpretation

---

The insights gained from this phase were instrumental in shaping subsequent methodological steps. Recognizing the AI's limitations with detailed but static documentation prompted a shift towards more dynamic and interactive methods of data provision. This phase underscored the need for integrating AI with supplementary tools that could handle the more technical aspects of data parsing and analysis, thereby allowing ChatGPT to focus on providing higher-level insights and guidance.

Overall, this phase of the investigation provided a deeper understanding of the challenges involved in using AI for BIM data analysis and highlighted the necessity for a multifaceted approach that combines AI capabilities with traditional data processing tools. The creation and utilization of the comprehensive PDF represented an important step in exploring how to effectively equip ChatGPT with the contextual knowledge needed for interpreting complex data encoded in IFC files.

```
1 # The task is to write a Python script to count the number of columns (IFCCOLUMN) in the given IFC file content.
2 # The IFC file content is structured, with each entity starting with a # sign and the entity type followed by an "=".
3 # We will scan through the file content line by line and count the occurrences of "IFCCOLUMN".
4
5 ifc_file_content = """
6 ISO-10303-21;
7 ...
8 #68=IFCCOLUMN('Ziw0q4oLeKG890stZAbfvP',#2,'St\\X\\FCtze-001',$,$,#112,#40,'ACE98D04-C95A-1440-8258-DB78CA969E59',.COLUMN.);
9 ...
10 #78=IFCCOLUMN('1KD5B$Q1xbJPHSFyefu0bI',#2,'St\\X\\FCtze-002',$,$,#113,#43,'543452FF-681E-E54D-945C-3FCA29E00952',.COLUMN.);
11 ...
12 #100=IFCCOLUMN('3GRQas9pT0IBe07z06Nvdo',#2,'St\\X\\FCtze-003',$,$,#114,#50,'D06DA936-2737-5848-BA00-1FD0065F99F2',.COLUMN.);
13 ...
14 END-ISO-10303-21;
15 """
16
17 def count_ifc_columns(ifc_content):
18     # Split the file content into lines
19     lines = ifc_content.split('\n')
20
21     # Counter for columns
22     column_count = 0
23
24     # Iterate over each line and count "IFCCOLUMN"
25     for line in lines:
26         if 'IFCCOLUMN' in line:
27             column_count += 1
28
29     return column_count
30
31 # Commenting out the function call to adhere to the instructions
32 # print(count_ifc_columns(ifc_file_content))
33
34
```

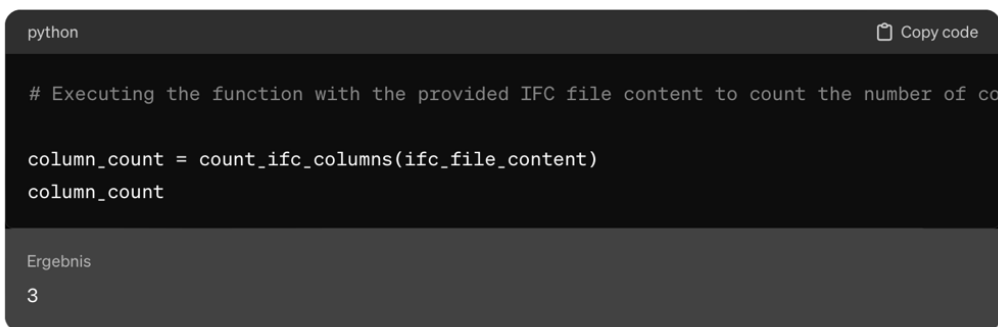
**Figure 3:** Automated Column Counting Script Generated by ChatGPT  
(Developed by Author).

## Code-Based Interaction

The approach of copying and pasting the raw IFC file code into ChatGPT demonstrated the model's capability to process and analyze text-based data. By providing the textual representation of the IFC file and querying ChatGPT to identify instances of the **'IFCCOLUMN'** entity, it was possible to leverage its natural language processing strengths to parse and interpret the structured data effectively. ChatGPT's ability to recognize and count the occurrences of specific entities within the raw text highlighted its proficiency in text analysis. The model successfully followed instructions to locate and identify the **'IFCCOLUMN'** tags, demonstrating its potential for extracting relevant information from complex data formats like IFC files.

This ability to analyze and interpret structured data from text alone provided a useful approach for handling IFC data without needing direct execution capabilities for specialized libraries.

In addition to manually identifying **'IFCCOLUMN'** entities, ChatGPT generated and executed a Python script to automate the process. The script was designed to read through each line of the IFC file and increment a counter each time it encountered an **'IFCCOLUMN'** entity. This automated approach not only streamlined the task but also ensured greater accuracy and efficiency in counting the columns within the file.



```
python Copy code  
  
# Executing the function with the provided IFC file content to count the number of co  
  
column_count = count_ifc_columns(ifc_file_content)  
column_count  
  
Ergebnis  
3
```

The script has processed the provided IFC file content and found that there are 3 columns in the file. [>\_]

**Figure 4:** Execution Result of Python Script for Counting Columns in IFC File *(Developed by Author)*.

The successful execution of the script underscored ChatGPT's capability to bridge theoretical guidance and practical application. By generating a functional Python script, ChatGPT was able to directly engage with the data and perform a specific task that required both an understanding of the file structure and coding proficiency. This result demonstrated that ChatGPT could assist in automating repetitive and data-intensive tasks within BIM workflows.

Moreover, the process of refining the queries and script iterations provided insights into the importance of precise and well-structured instructions. The accuracy and utility of the output were highly dependent on the clarity and specificity of the queries given to the AI. Initial queries often required adjustments based on the AI's responses, highlighting the necessity of iterative refinement to enhance the model's performance in analyzing IFC data. Overall, this phase of the investigation showcased ChatGPT's strengths in text-based data analysis and its ability to execute scripts to achieve practical outcomes. The findings from this approach provided valuable lessons on the importance of precise query formulation and demonstrated the potential for ChatGPT to contribute effectively to BIM data analysis tasks.

### ***Utilizing Visual Studio Code and GPT Plugin***

The integration of Visual Studio Code with user-created GPTs, particularly the Python GPT, proved instrumental in refining Python scripts for manipulating IFC files. The primary focus was to generate a column within an IFC file, and this process yielded several key findings. Initial attempts to define a column with specified dimensions and material properties encountered challenges, as the first few script executions resulted in incomplete or empty IFC files when loaded into ArchiCAD 24. This highlighted the complexity involved in accurately defining object relationships and geometric representations within the IFC schema.

The debugging process revealed that merely defining the column's properties was insufficient; it was crucial to establish proper relationships and placements within the IFC structure. This included defining **'IFCLOCALPLACEMENT'** for positioning the column accurately within the model and creating geometric representations to ensure the column's dimensions and shape were correctly captured. Additionally, the script needed to establish containment relationships among project elements, linking the column to the building storey and the overall building structure.

Repeated script enhancements and iterative refinement led to significant progress. Adjustments to the placement and geometric definitions gradually resulted in more accurate representations within the IFC file. These refinements underscored the intricacies of programmatic BIM data manipulation and the need for detailed attention to both entity definitions and their spatial relationships.

The iterative process, combined with the real-time suggestions from the Python GPT, facilitated efficient identification and resolution of issues. This setup allowed for a more structured and systematic approach to script development, enhancing control over the coding and debugging process. The findings underscored the value of using an integrated development environment like Visual Studio Code, supplemented by AI-driven assistance. While initial outputs highlighted the complexities of accurately modeling BIM data, continued refinements and eventual success demonstrated the potential for effectively combining traditional programming tools with AI to handle complex BIM workflows. The successful execution of the refined script, which accurately generated an IFC file with the defined column, underscored the potential of integrating AI tools with conventional development environments to enhance BIM data manipulation and analysis.

---

# Integration of AI and LLMs for Enhanced BIM Analysis: A Study on .IFC Model Interpretation

---

```
1 from ifcopenshell import create_entity
2 import ifcopenshell.api
3 import ifcopenshell.util.element
4 import ifcopenshell.util.placement
5
6 # Create a new IFC file
7 filename = "/mnt/data/ReinforcedConcreteColumn.ifc"
8 ifc_file = ifcopenshell.file()
9
10 # Define the dimensions of the column
11 width = 0.30 # 30 cm
12 depth = 0.30 # 30 cm
13 height = 3.00 # 3 m
14
15 # Create a basic column entity
16 column = create_entity(ifc_file, "IfcColumn")
17 column.Name = "Reinforced Concrete Column"
18
19 # Create a material
20 material = create_entity(ifc_file, "IfcMaterial")
21 material.Name = "Reinforced Concrete"
22
23 # Create a material layer
24 material_layer = create_entity(ifc_file, "IfcMaterialLayer")
25 material_layer.Material = material
26 material_layer.LayerThickness = width
27
28 # Create a material layer set
29 material_layer_set = create_entity(ifc_file, "IfcMaterialLayerSet")
30 material_layer_set.MaterialLayers = [material_layer]
31 material_layer_set.LayerSetName = "Column Layer Set"
32
33 # Apply the material to the column
34 ifcopenshell.api.run("material.assign_material", ifc_file, product=column, material=material_layer_set)
35
36 # Create a geometric representation
37 extruded_area_solid = create_entity(ifc_file, "IfcExtrudedAreaSolid")
38 extruded_area_solid.Depth = height
39
40 # Define a rectangle profile
41 rectangle_profile_def = create_entity(ifc_file, "IfcRectangleProfileDef")
42 rectangle_profile_def.XDim = width
43 rectangle_profile_def.YDim = depth
44 rectangle_profile_def.ProfileType = "AREA"
45
46 # Assign the profile to the extruded area solid
47 extruded_area_solid.SweptArea = rectangle_profile_def
48
49 # Define the placement of the column
50 placement = ifcopenshell.util.placement.create_ifclocalplacement(ifc_file)
51
52 # Assign the placement and geometric representation to the column
53 column.ObjectPlacement = placement
54 column.Representation = create_entity(ifc_file, "IfcProductDefinitionShape", Representations=[extruded_area_solid])
55
56 # Write the IFC file
57 ifc_file.write(filename)
58
```

**Figure 5:** Python Script Generated by ChatGPT for Creating a Reinforced Concrete Column in an IFC File *(Developed by Author)*.

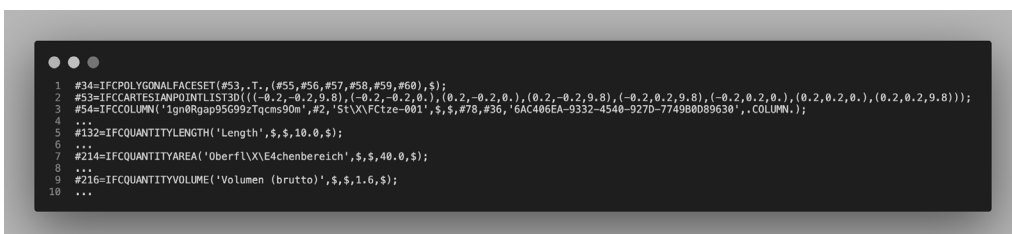
## Manual Code Editing and Validation

Manual editing of an existing IFC file, guided by ChatGPT, successfully modified the column height, which was verified through ArchiCAD. This process began by selecting an IFC file with an existing column and copying its code into Visual Studio Code for detailed inspection and modification. ChatGPT provided step-by-step guidance on identifying the relevant sections of the IFC code that defined the column's geometry and attributes. The specific task involved changing the height of a column.

ChatGPT guided the process by instructing how to locate the **'IFCCOLUMN'** entity and its associated **'IFCPRODUCTDEFINITIONSHAPE'** and **'IFCLOCALPLACEMENT'** entities. The detailed guidance included modifying the parameters that defined the column's height within the geometric representation. Additionally, ChatGPT highlighted the importance of updating related data, such as the column's volume, to ensure consistency within the IFC file.

While ChatGPT could generate the necessary lines of code to adjust the column height and related attributes, it could not directly implement these changes into the existing code due to the overall length and complexity of the file. Instead, I manually inserted the provided code lines into the appropriate sections of the IFC file. This manual intervention ensured that all necessary relationships and dependencies were correctly updated. After manually editing the code to adjust the column height and related attributes, such as volume, the updated IFC file was saved and loaded into ArchiCAD for verification.

The successful modification was confirmed visually and dimensionally within the BIM software, demonstrating that the changes had been accurately implemented.



```

1 #34=IFCPOLYGONALFACESET(#53, .T., (#55, #56, #57, #58, #59, #60), $);
2 #53=IFCCARTESIANPOINTLIST3D([(-0.2, -0.2, 9.8), (-0.2, -0.2, 0.), (0.2, -0.2, 0.), (0.2, -0.2, 9.8), (-0.2, 0.2, 9.8), (-0.2, 0.2, 0.), (0.2, 0.2, 0.), (0.2, 0.2, 9.8)]);
3 #54=IFCCOLUMN('Ign0Rgap05G99zTqcms90m', #2, 'S:\Fctze-001', $, $, #78, #36, '6AC406EA-9332-4540-927D-774980D89630', .COLUMN.);
4 ...
5 #132=IFCQUANTITYLENGTH('Length', $, $, 10.0, $);
6 ...
7 #214=IFCQUANTITYAREA('Oberfl\X\E4chenbereich', $, $, 40.0, $);
8 ...
9 #216=IFCQUANTITYVOLUME('Volumen (brutto)', $, $, 1.6, $);
10 ...

```

Figure 5: ChatGPT-Generated IFC Code Modification (Developed by Author).

This process showcased the potential of combining AI-driven guidance with manual interventions for specific data manipulations. ChatGPT's detailed instructions and code generation capabilities were crucial in identifying and making the necessary adjustments, while manual intervention ensured the correct integration of these changes into the existing code.

## Results

The findings from this investigation provide a nuanced understanding of the potential and limitations of using ChatGPT 4.0 for interpreting IFC files and performing BIM data analysis. This study's multi-phase approach allowed for a comprehensive evaluation of how AI can be integrated with traditional data processing tools to enhance BIM workflows, revealing several critical insights. The initial experimentation phase starkly highlighted the constraints of ChatGPT's current operational environment. The inability to execute external libraries such as 'ifcopenshell' prevented the model from directly parsing complex BIM data. This limitation was a pivotal moment in the study, emphasizing the need for alternative methods to facilitate meaningful interaction with IFC files. Despite these constraints, this phase was instrumental in identifying the operational boundaries of ChatGPT and underscored the importance of developing strategies to mitigate these limitations.

Providing structured information through an extensive PDF was one such strategy. This document, derived from BuildingSMART's detailed documentation, was intended to equip ChatGPT with the necessary context to interpret IFC data. However, the AI's struggle to generate actionable insights from the provided information indicated that simply supplying detailed contextual information is insufficient for effective analysis of complex data structures like IFC files. This finding suggests that AI models require not only structured data but also dynamic interaction capabilities to process and analyze such information effectively. The approach of copying and pasting raw IFC file code into ChatGPT showcased the model's proficiency in text-based data analysis.

By querying the model to identify instances of the '**IFCCOLUMN**' entity, the study leveraged ChatGPT's natural language processing strengths to parse and interpret the structured data. The successful generation and execution of a Python script to automate this process were significant achievements, demonstrating that ChatGPT can bridge theoretical guidance with practical application. However, the necessity for precise query formulation and iterative refinement became evident.

The model's ability to produce accurate and useful results was highly dependent on the clarity and specificity of the queries. This phase underscored that while AI can significantly aid in automating data-intensive tasks, human oversight remains crucial to ensure accuracy and relevance.

Integrating Visual Studio Code with user-created GPTs, particularly the Python GPT by Nicholas Barker, marked a substantial advancement in the investigation. This integration provided a robust framework for developing and refining Python scripts to manipulate IFC files.

The iterative development process, characterized by continuous adjustments based on feedback from error messages and debugging tools, revealed the complexity of accurately defining object relationships and geometric representations within the IFC schema. Initial script executions, which resulted in incomplete or empty IFC files, highlighted the intricacies of programmatic BIM data manipulation. However, the continuous refinement and eventual success of these scripts demonstrated the significant potential of combining traditional programming environments with AI-driven assistance. This integration facilitated more efficient debugging and refinement processes, underscoring the synergetic benefits of such a hybrid approach.

The manual editing of IFC files, guided by ChatGPT, provided further insights into the practical applications of AI in BIM workflows. This phase involved modifying specific attributes, such as column height and volume, based on detailed instructions from ChatGPT. While the AI could generate the necessary lines of code for these modifications, it could not directly implement these changes due to the complexity and length of the IFC files.

Manual intervention was required to ensure that all necessary relationships and dependencies were correctly updated. The successful verification of these changes in ArchiCAD confirmed the accuracy and effectiveness of this hybrid approach, demonstrating that AI-driven guidance combined with human execution can achieve precise and reliable results.

This phase highlighted several key insights. First, it emphasized the importance of a collaborative approach where AI provides theoretical guidance, and humans handle the practical implementation and validation.

---

## Integration of AI and LLMs for Enhanced BIM Analysis: A Study on .IFC Model Interpretation

---

This collaboration leverages the strengths of both AI and human expertise, enhancing the overall efficiency and accuracy of BIM data manipulation. Second, the process underscored the ongoing need for robust external tools to supplement AI capabilities. While ChatGPT offered valuable guidance, the execution and validation of changes relied heavily on tools like Visual Studio Code and BIM software such as ArchiCAD. The investigation's findings suggest several directions for future research and development.

Enhancing AI's ability to handle complex data structures and improving its integration with specialized libraries like `ifcopenshell` could significantly broaden its applicability in BIM workflows. Additionally, developing more intuitive and dynamic methods for data provision and interaction with AI models could further enhance their utility and accessibility.

For instance, integrating AI with real-time data processing tools and interactive interfaces could provide a more seamless and efficient workflow for BIM professionals. Moreover, the study highlights the potential for AI to assist in automating repetitive and data-intensive tasks within BIM workflows, freeing up human experts to focus on more complex and creative aspects of their work. This shift could lead to significant productivity gains and improvements in the quality and accuracy of BIM models. However, realizing this potential will require ongoing efforts to refine and enhance AI capabilities, ensuring they are robust, reliable, and easy to integrate into existing workflows.

Overall, this investigation provides a foundational understanding of the practical applications of AI in BIM data analysis, revealing both the opportunities and challenges that lie ahead. The study demonstrates that while AI holds significant promise for enhancing BIM workflows, its full potential can only be realized through a multifaceted approach that combines AI capabilities with traditional data processing tools and human expertise. By leveraging the strengths of each, it is possible to create more efficient, accurate, and user-friendly BIM workflows, paving the way for future advancements in the field.

## Conclusion

This study explored the integration of AI, specifically ChatGPT 4.0, with BIM data analysis, focusing on the interpretation of IFC files. Through a series of methodological steps, the investigation highlighted both the potential and limitations of using AI for complex data tasks in the Architecture, Engineering, and Construction (AEC) industry. The study illustrated that while AI has significant potential to enhance BIM data analysis, its current capabilities are best utilized in conjunction with robust external tools and human expertise.

“Leveraging a fusion of various technologies can significantly improve productivity and profitability, which the construction industry seeks the most” **(Su, Li, & An, 2021)**. The iterative methodological approach provided a practical framework for navigating the complexities of BIM data, highlighting the need for a multifaceted strategy to fully leverage AI’s potential.

The insights gained from this investigation point to several future directions for research and development. Enhancing AI’s ability to handle complex data structures, improving execution environments, and developing more intuitive interaction methods are essential steps toward broader AI integration in BIM workflows. Moreover, fostering collaboration between AI developers and BIM professionals will be crucial for driving innovation and ensuring that AI tools meet the practical needs of the AEC industry.

BIM models are valuable throughout the lifecycle of a project and require continuous maintenance. Integrating digital data systems with physical systems provides owners and operators with an efficient way to quickly capture information from a virtual facility model. However, as the BIM data repository grows, it necessitates more advanced analytical tools.

“Building Information Modelling (BIM) within the industry is hindering the extensive utilization of the technology” **(Bhat, Grunwald, & Hanke, 2023)**. These data science tools, particularly AI, help reduce uncertainties and discover new solutions, thereby enhancing the accuracy of the decision-making process **(Pedral Sampaio et al., 2022)**.

In conclusion, this study serves as a foundational step towards understanding the practical applications of AI in BIM data analysis. By addressing the current limitations and exploring new avenues for development, AI can significantly enhance BIM workflows, driving innovation and improving outcomes across the AEC industry. The combination of AI capabilities with traditional data processing tools and human expertise offers a promising path forward, paving the way for future advancements in the field.

The integration of BIM and AI holds promise, yet its implementation faces numerous challenges that require careful consideration **(Rane, 2023)**.

## Future Prospects

The findings and discussions from this study illuminate several promising avenues for future research and development in the integration of AI with BIM data analysis. “Numerous research on PC combined with BIM and other computing technologies have been investigated in recent years due to the multiple benefits that have been demonstrated” (**Veerakumar Rangasamy & Yang, 2024**).

As AI technologies continue to evolve, there are numerous opportunities to enhance their capabilities and address the limitations identified in this investigation. Integrating BIM with AI plays a crucial role in the digital transformation of the AEC/FM industry through automated applications such as big data analytics. Recent studies have combined BIM and AI techniques to tackle complex AEC/FM problems, yet a comprehensive review of these integrated applications, particularly concerning future research and practice, remains lacking (**Zhang et al., 2022**).

One of the primary areas for future development is improving AI’s ability to handle complex data structures like IFC files. Enhancing AI models to better understand and process the intricate relationships and dependencies within BIM data is crucial. This could involve integrating more advanced natural language processing techniques and machine learning algorithms capable of parsing and interpreting detailed data schemas. By refining these capabilities, AI could provide more accurate and actionable insights, further automating and optimizing BIM workflows.

Improving the execution environment for AI models is another key prospect. The inability of ChatGPT to directly execute external libraries like *‘ifcopenshell’* was a significant limitation in this study. Future iterations of AI platforms should aim to support the integration and execution of specialized libraries, enabling more seamless and comprehensive data analysis. This improvement would allow AI to handle more sophisticated tasks independently, reducing the need for manual interventions and external tools.

The necessity to address collaborative IT challenges has significantly contributed to the development and expansion of online construction collaboration technology. Developing more intuitive and dynamic methods for data provision and interaction with AI models is also essential. This could involve creating interactive interfaces that allow users to input and modify data in real-time, with AI providing immediate feedback and suggestions.

Such interfaces would make AI tools more accessible and user-friendly, particularly for professionals who may not have extensive technical expertise. Enhancing the user experience in this way could significantly broaden the adoption of AI in BIM workflows.

The integration of AI with real-time data processing tools and collaborative platforms represents another exciting prospect. By enabling AI to process and analyze data in real-time, professionals could benefit from immediate insights and recommendations, improving decision-making processes and project outcomes. Collaborative platforms that integrate AI capabilities could facilitate better communication and coordination among project stakeholders, enhancing the overall efficiency and effectiveness of BIM projects.

Further research should also explore the potential of AI to assist in automating more complex and creative aspects of BIM workflows. “The complex task of construction planning requires methodical knowledge, knowledge of general processes and structures [...]” (**Heins, Grunwald, & Helmus, 2021**).

While this study focused on relatively straightforward tasks, there is significant potential for AI to support more advanced functions, such as design optimization, predictive analysis, and automated compliance checking. By leveraging AI for these tasks, professionals could achieve higher levels of productivity and innovation, pushing the boundaries of what is possible in BIM.

The development of AI-driven tools tailored specifically to the needs of the AEC (Architecture, Engineering, and Construction) industry is another important direction. These tools should be designed to integrate seamlessly with existing BIM software and workflows, providing targeted solutions that address the unique challenges faced by professionals in this field.

Customizing AI tools in this way would maximize their relevance and impact, ensuring they deliver tangible benefits to users. Moreover, fostering collaboration between AI developers and BIM professionals is crucial for driving innovation and ensuring that AI tools meet the practical needs of the industry. Collaborative research initiatives, industry partnerships, and cross-disciplinary projects could facilitate the exchange of knowledge and ideas, leading to the development of more effective and impactful AI solutions.

Finally, addressing ethical considerations and ensuring the responsible use of AI in BIM is paramount. As AI becomes more integrated into professional workflows, it is essential to consider issues such as data privacy, security, and the potential biases inherent in AI algorithms.

Developing guidelines and best practices for the ethical use of AI in BIM will help ensure that these technologies are deployed in ways that are fair, transparent, and beneficial to all stakeholders. In conclusion, the future prospects for integrating AI with BIM data analysis are vast and promising.

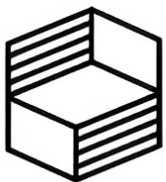
By addressing the current limitations and exploring new avenues for development, AI has the potential to significantly enhance BIM workflows, driving innovation and improving outcomes across the AEC industry. The insights gained from this study provide a strong foundation for future research and development, highlighting the exciting possibilities that lie ahead in the evolving landscape of AI and BIM.

## Acknowledgement

This research was supported by the AUFLADEN project at Jade Hochschule, Oldenburg which is funded by Stiftung Innovation in der Hochschullehre.

## Conflict of Interests

The author declares no conflict of interest.



**Stiftung  
Innovation in der  
Hochschullehre**

*Figure 07.* Logo „Stiftung Innovation in der Hochschullehre“

## References

Bhat, V., Grunwald, G., & Hanke, T. (2023). Understanding BIM through a simulation game: Case study of Indian students subjected to this course. In Proceedings of the 40th International Symposium on Automation and Robotics in Construction, ISARC 2023, July 2023. <https://doi.org/10.22260/ISARC2023/0044>

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, 104320. <https://doi.org/10.1016/j.autcon.2022.104320>

Grunwald, G., Heins, C. (2024). BIM Game: A testing ground for specifying, modeling, evaluating, and visualizing information in IFC formats. In Kang, T. (Ed.), Proceedings of the 5th International Conference on Civil Engineering and Architecture, ICCEA 2022. Lecture Notes in Civil Engineering (Vol. 369). Springer, Singapore. [https://doi.org/10.1007/978-981-99-4049-3\\_52](https://doi.org/10.1007/978-981-99-4049-3_52)

Grunwald, G., Kawasaki, J., Hanke, T., & Eidam, J. (2023). Robotic process automation to increase teaching efficiency in higher education. In Proceedings of the 2nd Asia Pacific Conference on Educational Research, Social Science and Humanities, APCERSSH 2023, September 2023, Bangkok. <https://doi.org/10.52783/jes.2481>

Hanke, T., Kawasaki, J., & Grunwald, G. (2023). Manufacturing processes of complex shapes and structures using 3D printing and augmented reality. In Nia, H. A., & Rahbarianyazd, R. (Eds.), Full paper proceedings of the 6th International Conference of Contemporary Affairs in Architecture and Urbanism, ICCAUA2023, 14-16 June 2023. E-ISBN: 978-605-71006-7-2.

Heins, C., & Grunwald, G. (2024). BIM and IPA: Excerpt of an automated assessment system for an autodidactic teaching concept. In Proceedings of the 41st International Symposium on Automation and Robotics in Construction, ISARC 2024.

Heins, C., Grunwald, G., & Helmus, M. (2021). Gamification and BIM: The didactic guidance of decentralized interactions of a real-life BIM business game for higher education. In Proceedings of the 38th International Symposium on Automation and Robotics in Construction, ISARC 2021, November 2021. <https://doi.org/10.22260/ISARC2021/0126>

Howard, R., & Björk, B.-C. (2008). Building information modeling – Experts' views on standardization and industry deployment. *Advanced Engineering Informatics*, 22(2), 271-280. <https://doi.org/10.1016/j.aei.2007.03.001>

Huang, M. Q., Ninić, J., & Zhang, Q. B. (2021). BIM, machine learning, and computer vision techniques in underground construction: Current status and future perspectives. *Tunnelling and Underground Space Technology*, 108, 103677. <https://doi.org/10.1016/j.tust.2020.103677>

Kawasaki, J., & Grunwald, G. (2023). Enhanced learning and teaching through AR: Case studies on design-build projects. In Proceedings of the 71st JSEE Annual Conference, September 2023. ISSN 2424-1466. [https://doi.org/10.20549/jseen.2023.0\\_18](https://doi.org/10.20549/jseen.2023.0_18)

Makatura, L., Foshey, M., Wang, B., Hähnlein, F., Ma, P., Deng, B., Tjandrasuwita, M., Spielberg, A., Owens, C. E., Chen, P. Y., Zhao, A., Zhu, A., Norton, W. J., Gu, E., Jacob, J., Li, Y., Schulz, A., & Matusik, W. (2023). How can large language models help humans in design and manufacturing? *arXiv*. <https://doi.org/10.48550/arXiv.2307.14377>

Pan, Y., & Zhang, L. (2021). Roles of artificial intelligence in construction engineering and management: A critical review and future trends. *Automation in Construction*, 122, 103517. <https://doi.org/10.1016/j.autcon.2020.103517>

Pedral Sampaio, R., Aguiar Costa, A., & Flores-Colen, I. (2022). A systematic review of artificial intelligence applied to facility management in the building information modeling context and future research directions. *Buildings*, 12, 1939. <https://doi.org/10.3390/buildings12111939>

Rane, N. (2023). Integrating Building Information Modeling (BIM) and artificial intelligence (AI) for smart construction schedule, cost, quality, and safety management: Challenges and opportunities. SSRN. <http://dx.doi.org/10.2139/ssrn.4616055>

Rane, N., Choudhary, S., & Rane, J. (2023). Integrating Building Information Modeling (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. <http://dx.doi.org/10.2139/ssrn.4645601>

Sacks, R., Girolami, M., & Brilakis, I. (2020). Building information modeling, artificial intelligence, and construction tech. *Developments in the Built Environment*, 4, 100011. <https://doi.org/10.1016/j.dibe.2020.100011>

Su, T., Li, H., & An, Y. (2021). A BIM and machine learning integration framework for automated property valuation. *Journal of Building Engineering*, 44, 102636. <https://doi.org/10.1016/j.jobe.2021.102636>

Veerakumar Rangasamy, & Yang, J.-B. (2024). The convergence of BIM, AI, and IoT: Reshaping the future of prefabricated construction. *Journal of Building Engineering*, 84, 108606. <https://doi.org/10.1016/j.jobe.2024.108606>

Zabin, A., González, V. A., Zou, Y., & Amor, R. (2022). Applications of machine learning to BIM: A systematic literature review. *Advanced Engineering Informatics*, 51, 101474. <https://doi.org/10.1016/j.aei.2021.101474>

Zhang, F., Chan, A. P. C., Darko, A., Chen, Z., & Li, D. (2022). Integrated applications of building information modeling and artificial intelligence techniques in the AEC/FM industry. *Automation in Construction*, 139, 104289. <https://doi.org/10.1016/j.autcon.2022.104289>