EPiC Series in Built Environment

Volume 4, 2023, Pages 381–389

Proceedings of 59th Annual Associated Schools of Construction International Conference

Extracting BIM data to support a machine learning model for automated clash resolution

Ashit Harode, and Walid Thabet, Ph.D, CM-BIM
Virginia Tech
Blacksburg, Virginia

Clash resolution is considered a critical step to resolve issues among the different disciplines for a construction design to be realized as expected. This step, however, continues to remain slow and manual which can significantly delay a project and drive-up costs. A combined machine learning model was proposed by Harode and Thabet (2021) to automate the clash resolution process. A large amount of labeled dataset is required to train and test the proposed model. The dataset is planned to be extracted from various industry-provided federated construction BIMs. Federated construction models are created from multiple subcontractor component models authored using different software. As a result, data is stored in various formats using different data structures making the extraction process difficult. In this paper, the authors demonstrate the use of commercially available software tools including iConstruct, Dynamo, and Talend to overcome this limitation and extract the necessary data. The paper first defines the required data structure followed by a data extraction process to capture required data from clashing elements in the federated BIMs. The paper also discusses a novel method of extracting end point coordinates and moveable area for clashing elements using bounding boxes. The paper concludes with future research directions.

Key Words: Design Coordination, Clash Resolution, Machine Learning, Data Extraction, IFC

Introduction

On any construction project, the coordination of mechanical, electrical, and plumbing (MEP) systems accounts for 6% of the total MEP cost (Hu et al., 2020). MEP coordination can be divided into two steps. In the first step, clash detection is performed and is focused on identifying MEP elements that occupy and compete for the same physical space. Clashes can either be detected manually using model walkthroughs, or automatically using Building Information Modeling (BIM) software tools such as Clash Detective from Navsiworks. In the second step, clashes are analyzed, filtered, classified, and discussed in coordination meetings to identify potential resolution strategies (Hu & Castro-Lacouture, 2019). Clash resolution remains a manual and slow process that relies heavily on the BIM coordinators' experience. One way to improve clash resolution is through the use of machine learning. Using machine learning to automate clash resolution has many benefits but requires a large amount of data to ensure its effectiveness and accuracy (Xu et al., 2021). Limited availability of data when

training a machine learning model can limit its effectiveness due to model overfitting. Overfitting is a modeling error that introduces bias to the model making it too closely or exactly related to a particular set of data, and irrelevant to other data sets. The model may fail to fit additional data or predict future observations reliably. Therefore, effective development of the machine learning model requires a large amount of labeled dataset as input that relies heavily on capturing sufficient expert knowledge.

Harode and Thabet (2021), and Harode, Thabet and Gao (2022) proposed a machine learning model to effectively automate clash resolution with limited data. The proposed model uses a combined supervised and reinforcement machine learning algorithm to automate clash resolution. The supervised learning model will be trained using limited clash resolution data. This model is later improved upon by acting as pre-training data for the reinforcement learning model. The proposed algorithm is hypothesized to be faster, more efficient, and require less data input to resolve clashes compared to using supervised learning or reinforcement learning individually.

Selecting the appropriate features for training a machine learning model is an important step that needs to be conducted to ensure that the model generates better data relationships and is explainable and implementable (Harode, Thabet, & Leite, 2022). Using literature review and industry interviews with experts from several general contractors and mechanical contractors, Harode, Thabet and Leite (2022) identified 13 factors (or features) of a clash that are utilized by clash coordination experts to resolve clashes. These factors are summarized in table 1. To implement an effective machine learning model, the proposed model will require a reasonable size dataset containing factors and their values for different clashing elements. Using several case study BIMs, the authors extracted data for the 13 factors from clashing elements in these models and stored the data as shown in figure1. Factors 1 through 11 are specific to each clashing element, whereas, factors 12 and 13 are common between the clashing elements. A detailed description of these factors is provided by Harode et. al., 2022. Example values for each factor are provided in figure 1 to illustrate the types of data that were captured from the case study models.

Table 1
List of factors (features) identified

S. No.	Features	S. No.	Features
1.	Start and End Point (X, Y, Z)	8.	Clashing Element Rigidity
2.	Element Dimensions	9.	Number of Clashes
3.	Clashing Element Type	10.	Moveable Area
4.	Clashing Element System Type	11.	Number of Connections
5.	Clashing Element Constrained Slope	12.	Critical Element in the Clash
6.	Insulation Size	13.	Location of the Clash
7.	Clashing Element Material		

With reference to figure 1, area 1 shows the list of 13 factors identified by Harode et al. (2022). In this paper, the authors explore, and test data extraction methodologies and tools used to capture data from the case study BIMs to populate area 2. This data will later be used to train the proposed machine learning model. Area 3 is the label heading for the machine learning model and area 4 represents potential options for resolving each clash acting as labels for the training dataset. Data for area 3 and 4 are not part of the data extraction process detailed in this paper and will be populated by the authors for each clash based on authors experience, discussion with the industry experts, and comparison with the final coordinated as-build models.

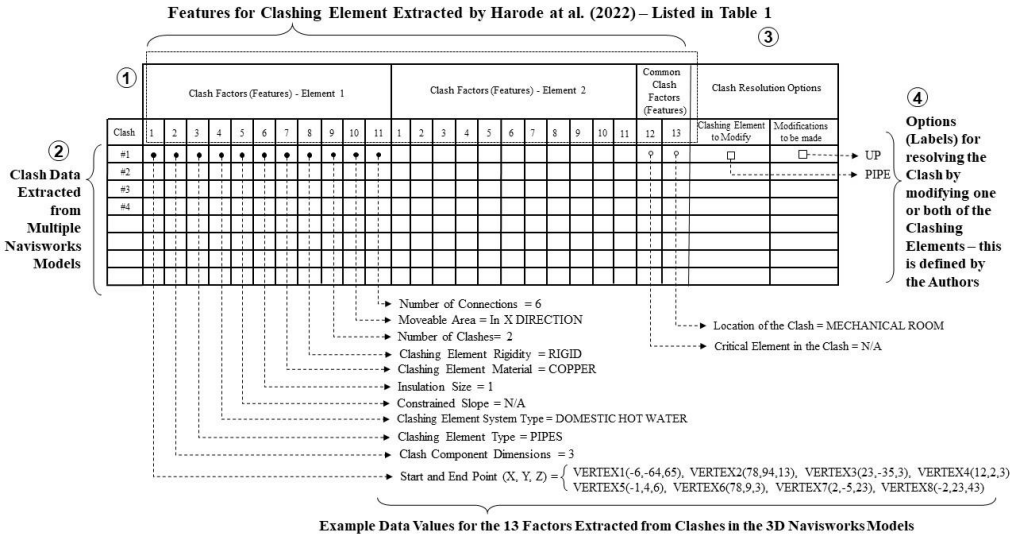

Figure 1. Organization of Factors and their values, and labels and their values for training the machine learning model proposed by Harode and Thabet (2021).

Xu et al. (2021) concluded the need to make construction data more available to support the development of Machine Learning in the construction industry. Given that Machine Learning is a highly data-driven process, they highlighted that data availability plays a key role in its implementation. They also concluded that difficulty in construction data acquisition and manual annotation is one of the limiting factors of Machine Learning adoption in construction. Bilal et al. (2016) while talking about the pitfalls of big data in the construction industry addressed fragmented data management practices as a cause for slower adoption of trends like big data. As a move forward towards effective data extraction from BIMs, Ignatova et al. (2018) explored and compared the extraction of embedded BIM data in the Revit model using standard Revit tools, SharpDevelop code editor, and Microsoft Visual Studio Plug-in. The data was extracted to support the future development of smart cities. Kim et al. (2013) also developed a framework that extracted data stored in BIM models to automate the generation of construction schedules. This framework was tested using a BIM model of a basic building. To improve the data extraction process for BIM models, Guo et al. (2020) developed a method to automatically generate SPARQL (Standard Query Language and Protocol for Linked Open Data and RDF database) queries based on users' data requirement. This method was validated using multiple case studies in which effective and accurate SPARQL queries were generated using the proposed method to facilitate data extraction. Kadcha et al. (2022) proposed an integrated solution using Dynamo and Power BI to facilitate data extraction and visualization. The data was extracted in the AEC domain of cost extraction, clash detection, change detection and plan extraction. Dynamo was used to extract data from the BIM models while Power BI was used to visualize the extracted data.

On a majority of construction projects, discipline component models (architecture, structure, HVAC, mechanical pipe, electrical, plumbing, fire protection, etc.) are first created using different model authoring tools, then integrated into a federated model. This creates challenges in extracting data from the federated model due to different data structures, formats and naming conventions used by the different source discipline models. Therefore, the research question that this paper is focused on answering is how can clash data from federated model created using models authored by multiple

software be effectively extracted to support creation of machine learning models for clash resolution? To answer this question, the authors explored and tested multiple software tools that can be utilized to facilitate extracting data for the 13 factors identified from federated Navisworks models. Tools tested included, iConstruct (https://iconstruct.com/), Dynamo (https://dynamobim.org/), and Talend Studio (https://www.talend.com/). iConstruct is an add-on to Navisworks and provides users with a large suite of tools to manage design and construction information. Users can export custom-built clash reports using the Clash Report tool or export the Navisworks model as IFC with desired properties using IFC Exporter. Dynamo allows users to interact with Revit API using Visual Programing to process data, generate geometry, and extract information. Finally, Talend Studio was used to create a data pipeline that can combine data in several Excels into a single Excel spreadsheet using a common data value. Talend can be used to preprocess data from any type and any number of data sources and export the data to a user-defined format.

The following section discusses the detailed research steps adopted by the authors to extract values for the 13 factors from several case study BIMs. The research focused the analysis on clashes between ducts and pipes only, using several Navisworks models of various case studies provided by industry partners. The paper concludes with a discussion and conclusion drawn from the tested data extraction process.

Research Steps

Figure 2 summarizes five main steps involved in the data extraction process.Using Clash Report function of iConstruct (1), clash test data including clash test name, clash group, clash name, and clashing element GUIDs were exported to an Excel file (ClashData.xlsx). Using the IFC export function of iCosntruct (2), the Navisworks model is exported and saved as an IFC file along with element properties like element GUIDs, element system type, element material type, area of the element, and element type. It should be noted that these element properties should be pre-defined in the Navisworks models used so that they can be exported with the model and used in the anlysis. The IFC model file is imported into Revit (3) and Dynamo (4) was used to extract the 13 clash factors to a second Excel file (ElementData.xlsx). Using the GUID parameter common between the two Excel files, data from both files (1-3 and a-e) are combined using the tMap function of Talend Studio (5). To test the proposed data extraction tools described in figure 2, the Navisworks model for a medical facility case study is selected. The federated model is generatred from combining several component discipline models with varying file formats including *.nwc, .dwg, and .dwfx*.. Parameter data defined in figure 2 forclashes between duct and pipe elements are identified and extracted using the tools and steps described. The following subsections provide a more detailed description of how each tool was used to extract the required data.

Using iConstruct to export Clash Data and IFC model

The Navisworks model for the medical facility case study was pre-loaded with clash tests conducted by the general contractor during the design coordination phase. iConstruct's clash report tools allow users to export clash data in PDF or Excel format. These clash reports are completely customizable and can include any data embedded into the clash elements.. In the proposed methodology, the clash report tool was used to custom-create an Excel template to export different clash information including the clash name, clash test group, and clashing elements GUIDs. The exported clash report (ClashData.xlsx) will be utilized to identify a clashing element using their GUIDs in the IFC model.

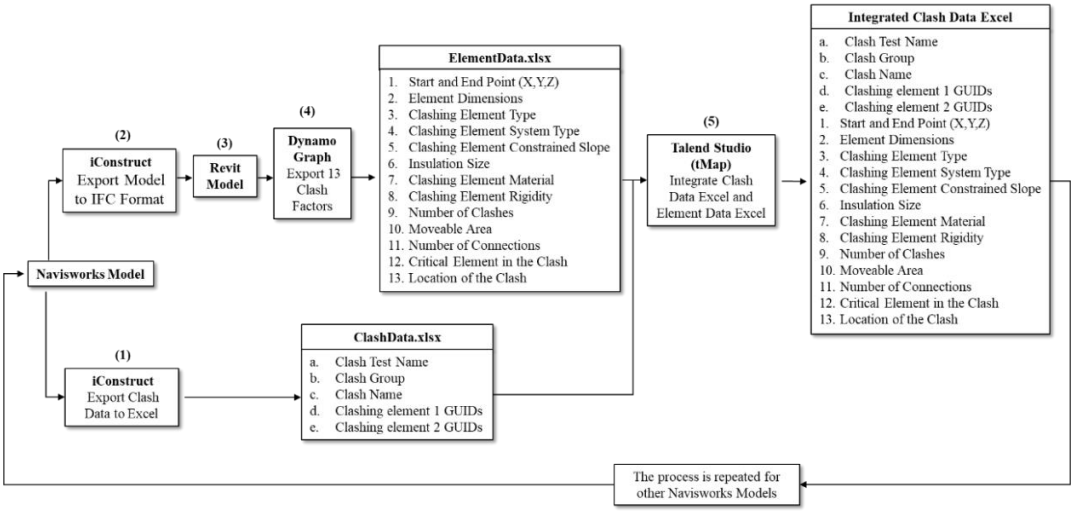

Figure 2. Proposed Methodology to extract clashing element data.

While retrieving data from BIMs, the lack of use of BIM authoring tools with standard file formats creates an interoperability issue and results in significant challenges to extract the data using common extraction rules. Different sub-contractors utilize different model authoring tools leading to the use of different data structures and property naming conventions. To overcome these challenges, the Smart IFC Export tool of iConstruct was used to export the Navisworks model to an IFC file along with the desired properties. In this proposed methodology the mechanical and plumbing models of the medical facility are exported as IFC files along with additional properties such as GUID, Material, Type Area, Layer, Source File, and Description in Navisworks.

Using Dynamo to extract and export factor data from the IFC model

Using the Revit model, Dynamo Graphs is used to extract values for the 13 clash factors from clashing elements identified by their GUIDs. Various spatial properties of clashing elements such as their start and end points and the moveable area around the clashing elements need to be extracted. Since duct and pipe elements in the converted IFC to Revit model did not originate from a Revit component discipline model, they are missing geometric parameters like element line segments and end point coordinates, and could not be extracted directly. The authors used the concept of bounding boxes to extract the missing spatial properties. Using Dynamo, bounding boxes were first created around all clashing elements and then converted into cuboid geometry. As shown in figure 3a, the eight vertices of these cuboids were extracted as the endpoint coordinates of the clashing elements. To identify available free space around each clashing element, six similar bounding boxes are created and placed around the boundaries of the element. These bounding boxes were then checked for possible clashes with other surrounding elements. If a bounding box returned a true value this indicated that another element interfered with the bounding box. All six sides around the clashing element were checked for other elements within its vicinity to determine feasible options to resolve the clash. Figure 3b shows the 6 bounding boxes surrounding the clashing element. The clashing element cannot be moved in the direction of bounding box “6” because it is clashing with the air terminal restricting the movement of the clashing element in that direction.

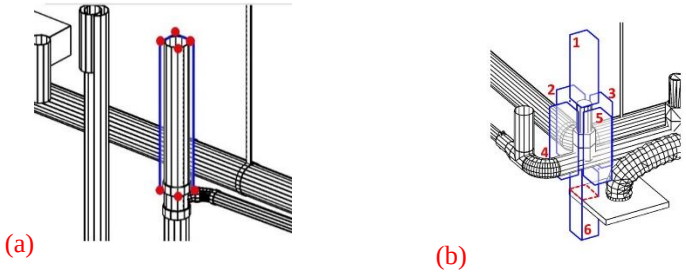

Figure 3. (a) Bounding box around an IFC element (blue) along with eight vertices (red). (b) Six bounding boxes surround the clashing element.

Because original component models were authored using different software platforms, embedded data in each model has a different structure hierarchy with different naming convention adopted resulting in inconsistent names for the same parameters. This did not allow for a straightforward approach to extract required parameter data of clashing elements from the federated model. This limitation is partially overcome by converting the Navisworks model into IFC resulting into a standardized data hierarchy as per IFC schema. Limitation of different naming convention of properties name could not be overcome required the authors to manually identify property names used by each component model and manually input the names into the Dynamo Graph script to be able to extract their values. The extracted 13 clash factors using Dynamo are exported to an Excel file (ElementData.xlsx) along with a unique GUID value corresponding to each of the clashing element. As discussd in the next step, this GUID is used to combine the ClashData.xlxs with ElementData.xlsx.

Using Talend to combine the two Excel data

Talend Studio is a software tool that can be used to build data pipelines focused on data integration, data cleaning, data pre-processing, and data management. Talend can handle large datasets from multiple data sources and create replicable pipelines that can speed up the data processing time. In this paper, Talend is used to combine data from the ClashData.xlxs with data from the ElementData.xlsxinto a single Excel file. This step can also be performed using the "VLookUp" function in Excel or writing a custom code in python. The use of Talend in this paper has been prompted by the need to perform data integration and data cleaning on large datasets when thinking about the practical implementation of machine learning. The use of Excel is limited to smaller datasets and file format (.xlsx) and python requires users to have experience with the coding language. To overcome these limitations, the authors in this paper have tested the Talend Studio software. Talend provides a graphical user interface platform that can be used to create pipelines for data integration, transformation, and pre-processing. It is also capable of handling big data more effectively making it a more powerful and more efficient replacement for Excel and python when handling a large amount of data commonly generated while performing design coordination on larger projects. In this paper, a relatively small dataset is used to test the proposed methodology. But in a practical scenario, while training a machine learning algorithm, thousands of rows of data are required. Therefore, the authors decided to explore the tools that can be utilized to work with big data for practical industry implementation.

Talend's *tMap* component can transform and route data from multiple sources to a single destination making it an ideal candidate for data integration using the two Excel files. tMap's mapping capabilities allow for defining the data routing and transformation of the final data. Two or more data sources can be combined using a common column as a relationship link. Users can export combined files in the format of their choice. Two comma-separated file for each of the two Excel files

(ClashData.xlsx and ElementData.xlsx) are generated and imported into Talend. Using the tMap component, the GUID 1 value for the first row of data in the ClashData.csv is read. This GUID value is then searched in the ElementData.csv and the row associated with this GUID value is copied and merged to the end of the first row of the Clash Data Excel spreadsheet. Similarly, the row corresponding to the GUID 2 value in the ElementData.csv is searched and merged to the end of the first row in the ClashData.csv adding the clashing elements BIM and spatial data to the ClashData.csv. This process is repeated for all rows in the ClashData.csv. The new Excel is saved with the name IntegratedClashData.xlsx as shown in figure 6b completing the data integration process.

Results

Table 2 shows an example of data collected for two clashes using the proposed workflow. The clash factor number in the table corresponds to the factors shown in figure 1. To understand how machine learning will be utilized to automate clash resolution, the table is divided horizontally into two sections. The Clash Factors representing required input or *Features*, and Clash Resolution Options representing required output or *Labels*. Clash *Features* contain the information for each clash that clash coordination experts will look for before making the decision on how to resolve the clash as discussed by Harode et al. (2022). This information includes information regarding individual clashing elements (Features 1 through 11) and information regarding the clash as a whole (Feature 12 and 13). Clash *Labels* comprise information on how the clash will be resolved. Examples of *Labels* may include element priority, direction of movement, and angle of sloping the element. This data for the *Labels* is decided by the authors based on experience, discussion with the authors, and extracting results from coordinated final as-built models.

Table 2
Sample clash data

	Clash Factors (Features)			
Clash Factor	**Clash #1**		**Clash #2**	
	Element 1	**Element 2**	**Element 1**	**Element 2**
1	-165.849, -94.920, 29.241, -165.849, -92.253, 29.241, -170.536, -92.253, 29.241, -170.536, -94.920, 29.241, -165.849, -92.253, 28.074, -165.849, -94.920, 28.074, -170.536, -94.920, 28.074, -170.536, -92.253, 28.074	-168.897, -95.764, 29.276, -168.897, -93.457, 29.276, -169.074, -93.457, 29.276, -169.074, -95.764, 29.276, -168.897, -93.457, 29.099, -168.897, -95.764, 29.099, -169.074, -95.764, 29.099, -169.074, -93.457, 29.099	-234.549, -125.611, 28.986, -234.549, -121.923, 28.986, -237.216, -121.923, 28.986, -237.216, -125.611, 28.986, -234.549, -121.923, 27.653, -234.549, -125.611, 27.653, -237.216, -125.611, 27.653, -237.216, -121.923, 27.653	-232.937, -122.874, 29.089, -232.937, -122.697, 29.089, -235.468, -122.697, 29.089, -235.468, -122.874, 29.089, -232.937, -122.697, 28.911, -232.937, -122.874, 28.911, -235.468, -122.874, 28.911, -235.468, -122.697, 28.911
2	3	1	3	2
3	Ducts	Pipe	Ducts	Pipe
4	Exhaust Air	Domestic Hot Water	Exhaust Air	Gravity Pipe
5	No	No	No	Yes
6	0	1	0	0
7	Sheet Metal	Copper	Sheet Metal	Cast Iron

8	Rigid	Rigid	Rigid	Rigid
9	2	1	2	1
10	In X Direction	In -Z Direction	In X Direction	In Y Direction
11	2	2	2	2
12	N/A		N/A	
13	Mechanical Room		Mechanical Room	
Clash Resolution Options (Labels)				
Element to Modify	Element 2		Element 1	
Modification to be made	Up		Left	

The objective of any machine learning algorithm is to identify the relationship between the two sections of table 2. During the training phase of the machine learning model, the algorithm will be provided with two inputs, Clash Factors (*Features*) for all the clashes identified and their Clash Resolution Options (*Labels*). The algorithm will then attempt to identify an equation that maps the values of Clash Factors (*Features*) to their corresponding Clash Resolution Options (*Labels*) by assigning weights to these features. The weight of each factor is calculated by minimizing the loss function (e.g. squared mean error) between the predicted and actual labels for the clashes in the training data set. For testing the accuracy of the trained machine learning model, the algorithm is provided a different set of clash factors and the predicted labels are compared with the actual labels to assess accuracy.

Discussion and Conclusion

Extracting, integrating, transforming, and cleaning large amounts of data is an essential step in machine learning. As we continue the research to explore how machine learning can be used to automate clash resolution and improve the process, similar efforts need to be made to explore tools that can facilitate extraction of required data. The use of machine learning for construction applications is currently challenged by the limited availability of sufficient relevant data that can be defined as training data for machine learning implementations. When available in models, data is often present in a fragmented inconsistent format due to the use of multiple authoring software tools resulting in data embedded using different data structures and different naming conventions. To overcome these limitations and to support the implementation of the machine learning algorithm proposed in Harode and Thabet (2021), the authors have explored and tested a proposed workflow that utilized several commonly available off-the-shelf technology tools, including iConstruct, Dynamo Graph, and Talend, to extract data from Navisworks models. The proposed workflow is focused on extracting the required data to create the necessary dataset of features and labels required for the implementation of a proposed supervised-reinforcement machine learning model for automation of clash resolution. The required data being extracted is based on 13 clash factors considered by industry experts while resolving clashes identified using literature reviews and industry interviews conducted by Harode, Thabet and Leite (2022). The proposed workflow for data extraction was tested on a single Navisworks model with a small-size dataset. The workflow can also be scaled to extract a larger dataset using multiple models, a scenario more suitable for the development of an effective machine learning model. The authors utilized a novel method to extract spatial properties namely end point coordinates of the clashing elements and moveable area around the clashing elements using the concept of Bounding Boxes. This novel method can be utilized to extract spatial information for the clashing elements that are not authored using Revit and/or do not have a line element associated with them.

The following are several future steps that are being investigated and implemented to support the machine learning model proposed by Harode and Thabet (2021): (1) extract a larger parameter data set from multiple Navisworks models, (2) each row in the figure 1 will be populated with a potential clash resolution options (*Labels*) completing area 3 and 4, (3) the data will be preprocessed (scrubbed, dimensionally reduced, encoded, standardized, etc.) to make it more suitable for implementation of the Machine Learning algorithms, (4) a Supervised Learning algorithm will be trained using this data to predict potential clash resolution options, (5) the model developed using the Supervised Learning algorithm will be improved upon by using Reinforcement Learning to improve the effectiveness of the automation model for clash resolution, and (6) the improved automation model will be tested for effectiveness and efficiency.

Acknowledgement

The authors would like to sincerely thank the following individuals for their support and for providing an academic license of their software that helped make this research possible: Robert Gadbaw, iConstruct (iConstruct), and Lisa Neu and Danielle Sacks, Talend (Talend). The views and findings expressed in this paper are those of the authors and do not reflect those of iConstruct or Talend.

References

Bilal, M., Oyedele, L. O., Qadir, J., Munir, K., Ajayi, S. O., Akinade, O. O., Owolabi, H. A., Alaka, H. A., & Pasha, M. (2016). Big Data in the construction industry: A review of present status, opportunities, and future trends. *Advanced engineering informatics*, *30*(3), 500-521.

Guo, D., Onstein, E., & Rosa, A. D. L. (2020). An approach of automatic SPARQL generation for BIM data extraction. *Applied Sciences*, *10*(24), 8794.

Harode, A., & Thabet, W. (2021). Investigation of Machine Learning for Clash Resolution Automation. *EPiC Series in Built Environment*, *2*, 228-236.

Harode, A., Thabet, W., & Gao, X. (2022). An Integrated Supervised Reinforcement Machine Learning Approach for Automated Clash Resolution. Construction Research Congress 2022,

Harode, A., Thabet, W., & Leite, F. (2022). Feature Engineering for development of a Machine Learning Model for Clash Resolution. *EPiC Series in Built Environment*, *3*, 398-406.

Hu, Y., & Castro-Lacouture, D. (2019). Clash relevance prediction based on machine learning. *Journal of computing in civil engineering*, *33*(2), 04018060.

Hu, Y., Castro-Lacouture, D., Eastman, C. M., & Navathe, S. B. (2020). Automatic clash correction sequence optimization using a clash dependency network. *Automation in Construction*, *115*, 103205.

Ignatova, E., Zotkin, S., & Zotkina, I. (2018). The extraction and processing of BIM data. IOP Conference Series: Materials Science and Engineering,

Kadcha, Y., Legmouz, D., & Hajji, R. (2022). AN INTEGRATED BIM-POWER BI APPROACH FOR DATA EXTRACTION AND VISUALIZATION. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.

Kim, H., Anderson, K., Lee, S., & Hildreth, J. (2013). Generating construction schedules through automatic data extraction using open BIM (building information modeling) technology. *Automation in Construction*, *35*, 285-295. https://doi.org/https://doi.org/10.1016/j.autcon.2013.05.020

Xu, Y., Zhou, Y., Sekula, P., & Ding, L. (2021). Machine learning in construction: From shallow to deep learning. *Developments in the built environment*, *6*, 100045.