Automatic generation of Structural models from BIM models using Semantics and Machine Learning

Lini Xiang[1], Gang Li[1], Haijiang Li[2]
[1]Dalian University of Technology, China, [2]Cardiff University, UK
xln01214@mail.dlut.edu.cn, ligang@dlut.edu.cn, lih@cardiff.ac.uk

Abstract. Simulation of a BIM model under different disasters can provide reliable data for designing safer structures, and precondition is to generate the matched finite element models from the BIM model correctly. However, traditionally, obtaining the structural model mainly relies on expert experience and manual programming, and once the scenario changes, the code needs to be developed again. This paper introduces an automatic approach of mapping from BIM model context to finite element parameters utilizing semantic embeddings and artificial neural networks. It can generate a structural model immediately usable from BIM model via pre-trained models and background knowledge, without manual adjustment. A reinforced concrete frame is used to verify the validity of the proposed method. The results show that new tools of artificial intelligence are able to solve the challenges of applying BIM data in expert models, but it needs more evidence in improving efficiency, coding workload, automation, and generalization capability.

1. Introduction

The finite element (FE)-based uncertainty quantification is widely used in many computationally intensive tasks, such as structure design in civil, mechanical, and aerospace fields (Melchers et al., 2017). However, the application of BIM models for uncertainty analysis and structural reliability is hindered due to the difficulty of generating structural models from BIM models. Compared with carbon neutrality and structural health monitoring, less attention and research have been devoted to this area.

There have been many attempts to convert BIM models to finite element models. The technical challenges they faced were how to extract information from Industry Foundation Class (IFC) files to compose simulation-oriented models - from a uniform information exchange format across the domain to a proprietary format for a single domain; from serialized information (Step), including component types, geometric 3D data, and physical properties, to an analysis language (MATLAB, python); from descriptions based on real objects to descriptions based on mathematical abstractions. The spread of BIM models around the world has increasingly revealed the lack of means for structural engineers to make efficient use of BIM models.

Machine learning (ML) is a subfield of artificial intelligence, where algorithms are developed to learn from data, generalize and make predictions (Mitchell, 1997). Under the right development, ML models are able to bypass the difficulty of directly converting IFC instances into structural model elements, as we use statistical learning instead of descriptive semantic mapping to construct geometry, sectional parameters, and material parameters. This process does not require human intervention or compatibility with different structure interpreters. In addition, semantic web, ontologies and knowledge graphs are validated in capturing and reusing domain knowledge from BIM models (Chen et al., 2022, Johansen et al., 2022, Shahinmoghadam et al., 2022). Therefore, the BIM model is the natural data source for ML training, providing semantic texts on building components as well as parameters.

This paper combines the advantages of both approaches and proposes a method to perform structural element mapping according to the context of BIM models, enabling the generation of customized structural models in shorter time and with less hard-coding. In Section II, introduce how we prepare BIM training data, build artificial neural networks (ANNs), perform mapping of components to FE elements and assemble them into OpenSees models. In Section III, a reinforced concrete (RC) frame is used to validate the proposed method. Sections IV and V summarize the key findings of this study and implications for similar work.

2. Related Work

There are three different ideas to solve the challenge of how to convert serialized building information in IFC files to FE models. First, shape inference. The main work is to create complex geometry and mesh it by surface reconstruction algorithms from 3D polylines and 3D pointcloud data (Xu et al., 2019, Park et al., 2020, Rasoulzadeh et al., 2022). The reconstructed geometries with volume mesh are then paired with boundary conditions, load and material information to compose a structural analysis model. This field focuses on 3D modeling, efficient algorithms for point-to-surface, and meshing, so it is suitable for dealing with large structures with complex surfaces.

Second, rule-based reasoning. In the past time, many studies used semantic information from BIM models as middleware to translate structural modeling intents (Ramaji et al., 2018, Sibenik et al., 2020, Sibenik et al., 2021, Jia et al., 2022). Because serialized building information (Step file) can provide a shared network of concepts and relationships, often presented in a graphical form, this is the Semantic Web. Computers are able to reason about the Semantic Web to obtain logical facts using the Web Ontology Language (OWL), which is called rule-based reasoning process. An approach is to create IFC-to-structural mapping rules directly. Or use algorithms to solve constraint graphs (a graph in which the parameters, shapes, and functions of the IFC components form the graph elements while the constraint relationships form the network) (Kirchner, 2022). By solving the (n+1)B-consistency problem, constraint graphs can be transformed into a set of equations and variables of the structure. But these methods rely too much on formalization knowledge and symbol reasoning, therefore it is the least efficient and too abstract to understand.

Third, API interface development. The most direct and least compatible method is to use hard coding to rewrite process of defining BIM model information to SAM files (Alirezaei et al., 2016, Zhang et al., 2017), such as from Revit to CATIA. This approach has high accuracy, but low efficiency and poor generalization capability to reuse between different structural analysis software (e.g. ETABS, MIDAS, SAP2000).

Considering the drawback of the above methods, we would like to improve in terms of efficiency, coding workload, automation, generalization capability, and ease of understanding. There is a good prior study that is a set of machine learning-based BIM-to-OpenSees code developed by the NHERI Center in the United States (Wang et al., 2019). However, they only focused on machine learning and did not play the role of building semantics. This paper is aimed to propose a new solution that takes advantage of both sides to achieve a better conversion from BIM model to FE model.

3. Methodology

This section will introduce the types of structural analysis, the establishment of ML models, and the construction of FE models. Figure 1 illustrates the complete flowchart.

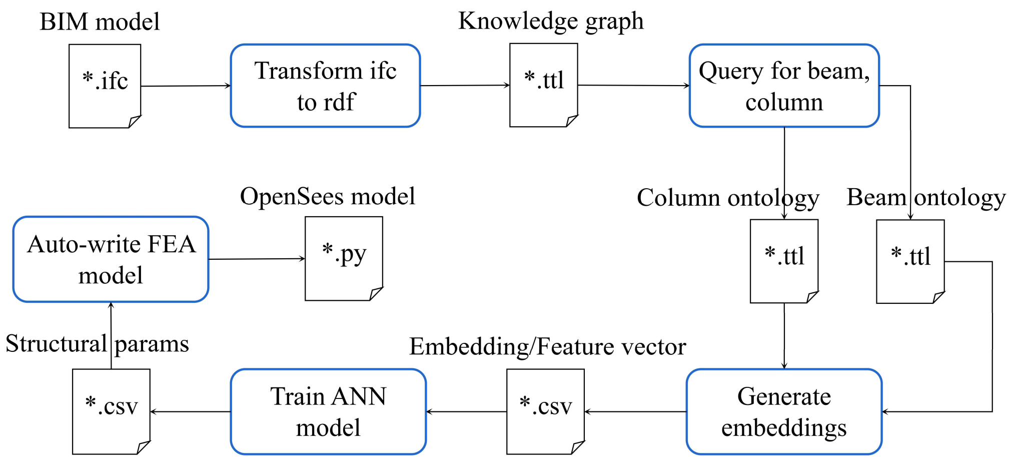

Figure 1: The flowchart of generating FE models automatically from BIM models

3.1 Structural Analysis Concept

Structural Dynamics Equations. The following equation is a general form for calculating the structural dynamic response (Vurtur Badarinath et al., 2021):

$$\mathbf{M}\ddot{\boldsymbol{u}}(\boldsymbol{t})+\mathbf{C}\dot{\boldsymbol{u}}(\boldsymbol{t})+\mathbf{K}\boldsymbol{u}(\boldsymbol{t})=\boldsymbol{f} \tag{1}$$

where $\boldsymbol{u}(\boldsymbol{t})$ is the displacement vector containing the nodal degrees of freedom depending on time $\boldsymbol{t}$, and $\dot{\boldsymbol{u}}(\boldsymbol{t})$ and $\ddot{\boldsymbol{u}}(\boldsymbol{t})$ are its first and second derivatives of time, respectively representing the velocity and acceleration. $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix, $\mathbf{K}$ is the stiffness matrix, and $\boldsymbol{f}$ is the vector of discretized loads applied to each degree of freedom. $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are determined subsequently after the geometric form and material properties are determined. Given the loading vector $\boldsymbol{f}$, time history analysis is conducted to obtain the discrete response data of each degree of freedom at each time point:

$$\boldsymbol{f}=\left[f_1,\ldots f_n\right]\rightarrow \boldsymbol{u}=\left[u_1,\ldots,u_n\right],\dot{\boldsymbol{u}}=\left[\dot{u}_1,\ldots,\dot{u}_n\right],\ddot{\boldsymbol{u}}=\left[\ddot{u}_1,\ldots,\ddot{u}_n\right] \tag{2}$$

OpenSees FEA Model. In this paper, the RC building is simplified as a 2D shear model, considering only the response along the lateral direction. This multi-story, multi-bay frame with gravity system comprise nodes, elastic beam elements and nonlinear column elements. FE analysis is conducted using the OpenSees fiber elements that are discretized into the longitudinal reinforcing bars and the concrete fibers wrapped outside. Rigid beam-column connections are adopted. The degrees of freedom of the bottom nodes are restricted to zero. OpenSees variables are considered include:

- Geometrical parameters: the cross-sectional area of reinforcing bar (As), column height (height), beam span (width), and thickness of concrete protective layer (cover).

- Material parameters: yield strength of the steel (fy), elastic modulus (E), strain hardening coefficient (b) and compressive yield strength of concrete (fc).

3.2 Machine Learning Training

Data Preparation. The training process is to map the beam and column entities of the BIM model directly to the necessary structural parameters. Considering that all entities and their properties are usually exported in RDF or OWL format for querying and reasoning in a natural language manner, there exists a gap making BIM entities not applicable to ML processes. Fortunately, the BIM community has developed a method to generate embeddings of semantic texts inside knowledge graphs in RDF or OWL format (Shahinmoghadam et al., 2022). The embedding is a feature vector representation of semantic information to be used for digital tasks.

In order to convert beam and column instances in BIM model to ML training data, embedding learning is conducted:

1. SPARQL query is used to extract all beam and column entities, as well as related mechanical properties (listed in the above section) entities, in turtle serialization.
2. The pyRDF2Vec library (Vandewiele et al., 2022) is used to generate feature vectors through *.ttl file, with 100-dimension by default for each entity.
3. t-SNE (t-distributed Stochastic Neighbor Embedding) algorithm is employed to reduce the dimension of the generated embeddings. Since large number of irrelevant dimensions usually cause overfitting in training, 2-dimension embeddings are used in our regression tasks.

ANNs Model. ANNs are complex models that emulates the structure and function of neurons in the human brain to solve real-world complex problems. It uses neurons and weights to represent the connections of data, and uses activation functions to tune the output of neurons (Kelleher et al., 2020). We employ a supervised learning on a specific dataset to generate the expected output for a set of paired inputs, to build regression models based on the embeddings generated from structural components and their mechanical properties in the BIM model, to predict parameters of new beam/column instances. The learning process is accomplished by adjusting the network weights and various hyperparameters (learning rate, hidden layer, batch size) to minimize the observed errors (Kelleher et al., 2020). In this paper, it was trained with 5000 epochs using Adam optimizer, with Rectified Linear Unit (ReLu) activation function, learning rate of 0.0001, and batch size of 32. The details of the ANNs are shown in Table 2. All code was developed by Keras library in Python.

Table 1: Details for ANNs structure in this study.

Layer	Description	Parameter
Input Layer	Factors	Beam embedding / Column embedding
	Number of Units	1
	Number of Hidden Layers	4
Hidden Layer	Number of Units	512, 64, 64, 64
	Activation Function	ReLu
Output Layer	Dependent Variables	width,fy,E,b,As / height,fc,cover
	Number of Units	5 / 3
	Error Function	Mean Squared Error

For regression tasks in this study, the quality of predictions is quantified by mean absolute error (MAE, $\mathrm{MAE} = \frac{1}{\boldsymbol{n}} \sum_{\boldsymbol{i}=0}^{\boldsymbol{n}-1} \left| \boldsymbol{y}_i - \hat{\boldsymbol{y}}_i \right|$), mean squared error (MSE, $\mathrm{MSE} = \frac{1}{\boldsymbol{n}} \sum_{\boldsymbol{i}=0}^{\boldsymbol{n}-1} \left(\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i \right)^2$), and R-squared ($R^2$, $\boldsymbol{R}^2 = 1 - \frac{\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \left(\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i \right)^2}{\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \left(\boldsymbol{y}_i - \overline{\boldsymbol{y}} \right)^2}$). ANNs with higher R^2 closer to 1 and lower MAE and MSE closer to 0, are better fitted.

3.3 OpenSees Model Generation

Given the embeddings, and after the prediction via pre-trained ANNs, output features in one-dimension are obtained. The ultimate goal is to use these features to auto-write a reality-compliant FE model. A simple regression (ordinary least squares, OLS) is conducted to map the features to the real-world parameters. The component parameters will be fed into the template of the 2D shear frame model in the OpenSeesPY library. In addition to our method, Tcl builder written in C++ language (Wang et al., 2019) is able to write FE model in Tcl language.

4. Results

In this section for validation, a BIM model of a 10-story, 14-bay RC frame structure is used (see Figure 2). The BIM model was serialized by IFCtoRDF tool (Pauwels et al., 2016, Pauwels, 2020) containing the basic building component types, geometric and spatial information, physical properties, and various instances.

Two different datasets are created, corresponding to beam elements, beam parameters and column elements, column parameters, respectively. Inside the datasets, the labels are embeddings of extracted properties, while the data are the embeddings of the entities. Each dataset is split into training samples (80%) and test samples (20%). The summary of the building ontology and the two datasets are shown in Table 2.

Figure 2: BIM model of multi-story RC frame in Shenyang, China

Table 2: Summary statistics of the building ontology and two datasets.

	Description of the Knowledge Graph			**Description of the Dataset to be trained**		
	Triples	**Maximum LoD**		**Train/Test samples**	**Data**	**Label**
Beam ontology	1360	7	Beam dataset	35 9	44 (entity)	[44×5] (embedding width, fy, E, b, As)
Column ontology	64948	-	Column dataset	312 78	390 (entity)	[390×3] (embedding height, fc, cover)

In the first step of the experiment, embeddings were calculated. Figure 3 shows the 2D visualization of embeddings of beam/column entities and embeddings of width/height parameters by t-SNE algorithm. It can be observed that there are spatial pattern similarities between the beam and beam parameters. The scatter of column and column parameters, however, shows a tendency to be uncorrelated. The embeddings of the other parameters are not shown here.

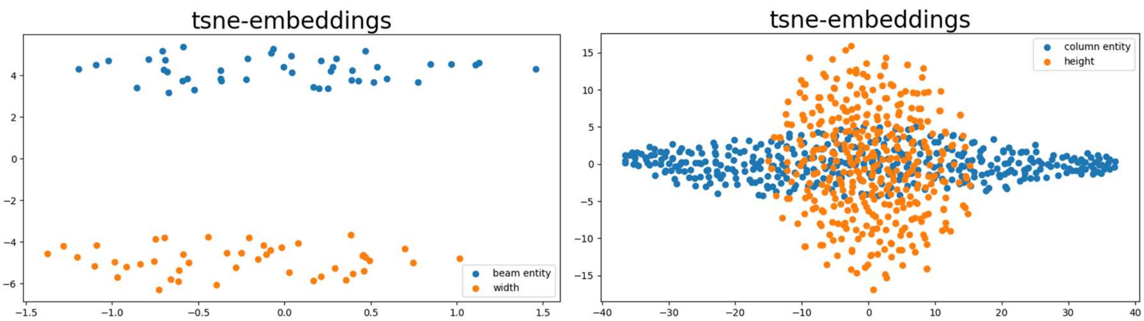

Figure 3: 2D visualization of embeddings of (a) beam and its width (b) column and its height

The ANN training process is shown in Figure 4 and Figure 5. The left graph shows the gradual convergence of MSE as the epoch increases, while the right graph shows the error between predictions and true values of test samples. More performance metrics on ANNs are summarized in Table 3. Regarding MAE and MSE, the values for beam-ANN model are 0.3288 and 0.1764, respectively, and for column-ANN model are 1.0188 and 1.4909. All the above indicators are close to 0, indicating that hyperparameters are correctly chosen and there is no overfitting or underfitting. However, the R^2 for the beam-ANN model is 0.3389, indicating that only 33% of the variation in variables were predicted. The R^2 of the column-ANN model is 0.0047, indicating that the prediction accuracy of this ANN model is similar to that of taking the sample mean directly. Considering both samples are small and dimensionality reduction and the ANNs are the same, the difference in training results between the two datasets can only be interpreted as the fundamental difference in the semantic text, i.e., whether there is an explicit linking path (or implicit one-to-one relation) in the target ontology that can connect the entities to the physical properties.

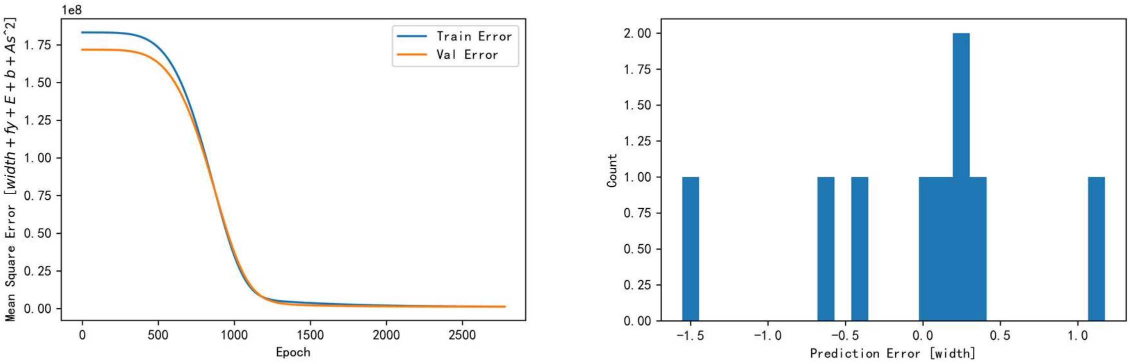

Figure 4: The training loss curves and the prediction of Beam-width

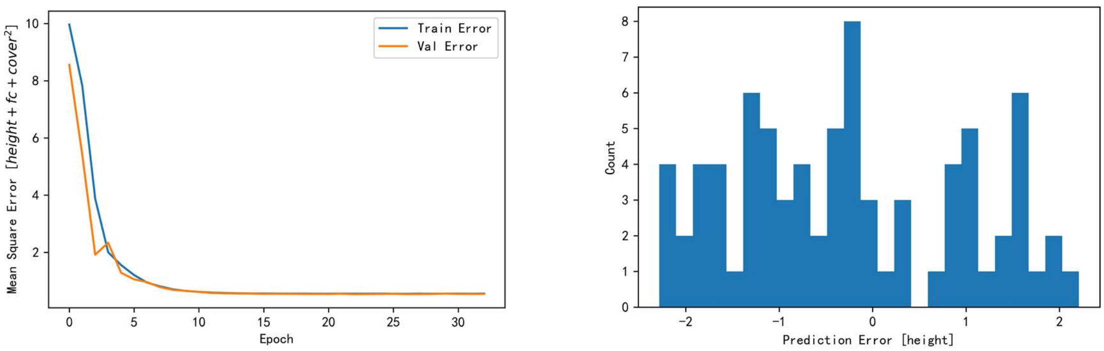

Figure 5: The training loss curves and the prediction of Column-height

Table 3: Performance of the two ANNs on the testing set.

Metrics	Beam-ANN model	Column-ANN model
Mean Absolute Error	0.3822	1.0188
Mean Squared Error	0.1764	1.4909
R-squared	0.3389	0.0047

Inst:IfcBeam_449008 and *inst:IfcColumn_141474* as entities were introduced into the pre-trained ANNs, to verify how effective the semantic text is in communicating modelling intents. Table 4 summarized input variables and output prediction. Comparing the predictions of both sides, it is obvious that beam semantics-communicated parameters is learned better.

Table 4: Randomly input variables and the corresponding ANNs prediction.

Entity	**Input variable/Embedding**	**Output variable**					
			Width (inch)	**Fy (kips)**	**E (kips)**	**b**	**As (in ^2)**
IfcBeam_4490	[1.18580759,	**Actual data**	149.61	60	30000	0.01	0.6

08	3.09675908]	**Prediction**	146.80	57.96	29537.48	0.03	0.64
			Height (inch)	**Fc (kips)**	**Cover (inch)**		
IfcColumn_141	[-2.41622810,	**Actual data**	165.35	5	0.98		
474	1.19574952]	**Prediction**	159.98	6.72	1.14		

The obtained parameters were filled into OpenSeesPY code to perform reliability analysis under ground motion. Given that time history analysis is time-consuming, the computational cost of large-scale uncertainty simulation is unacceptable if performed with Monte Carlo simulation. Here we chose 50 Latin hypercube sampling (LHS). For each LHS, all predicted structural parameters were set as Gaussian variables, to perform time history analysis under the "elCentro" acceleration record, to calculate the maximum displacement of the top floor. After 50 loops, see if any displacement record exceeded 1% of the total height. Figure 10 shows the uncertainty propagation results. The simulation results show that all displacements are less than 16 inches (1% of the total height), but computing failure probability needs more simulations. We only verified the success BIM-FEA conversion process, not accurate failure probability.

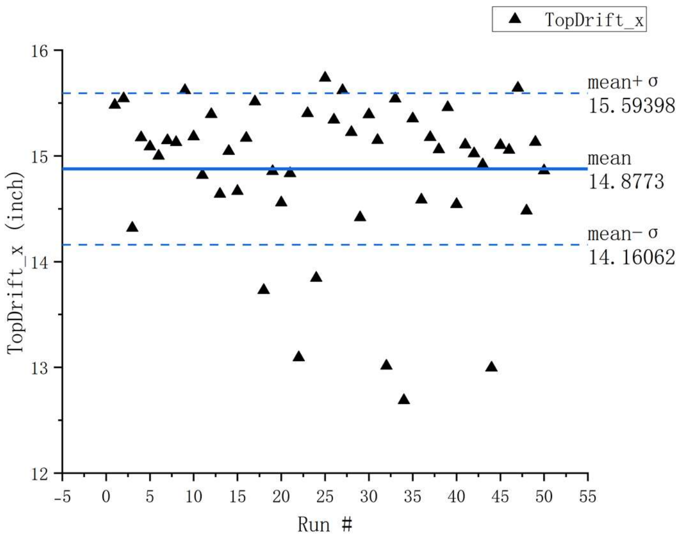

Figure 6: Earthquake anlysis for the generated FE model (Uncertainty propagation results for the maximum displacement of the top layer)

5. Discussion

Section 4 showed the flowchart results step by step. The experiment demonstrated that, it is feasible to achieve prediction of FE parameters and to automatically construct FE models consistent with BIM models, by statistical learning based on the embeddings of building ontology. The validity of the new method was verified. However, it should also be seen that the prediction of the machine learning model in this paper should be further improved. Table 4 shows that the proposed ANN model has satisfactory accuracy in predicting the beam parameters, although R^2 is not close to 1 as desired (R^2=0.3389). But the effect on column entities (R^2=0.0047) is only slightly better than directly using the average value to replace the prediction process. Increasing the learning sample size may be necessary.

In addition, further comparisons of other methods mentioned in related work are needed to verify the degree of improvement in terms of efficiency, coding workload, automation and generalization capability. In particular, compared with the rule-based inference approaches, how much work does our method decrease in getting specific properties of the specified building entity. Besides, compared with pure machine learning, how better does learning and prediction results be by using context data sources or building semantics. Finally, regarding generalization capabilities, the proposed solution is expected to be adaptable when engineering scenarios change, so more work is needed to demonstrate that it is not limited to being applied to frame structures.

6. Conclusion

In this paper, we propose an automatic transformation approach from BIM models to structural analysis models, using BIM ontology and artificial neural networks. The results demonstrate that building semantics are able to convey implicit modeling knowledge and simulation intent by machine learning process, and finally to construct the responding FE model.

The value of this study is that it provides a better solution to achieve the conversion of BIM models to finite element models - using new tools of artificial intelligence to solve the challenges of applying BIM data in expert models. And, compared with similar studies, there is more space for improvement in our approach, as it relies more on machine intelligence rather than human manual work. However, the drawback of this study is that only the effectiveness of the new workflow has been validated, and more perceived potential advantages are not yet supported by sufficient evidence. Therefore, future work includes completing comparisons with similar studies and different types of conversion methods.

Acknowledgements

This research was funded by the National Natural Science Foundation of China (Grant No.: 11872142). The BIM model was provided by Associate Professor Haiyan Lu of Shenyang University of Technology.

References

Alirezaei, M., Noori, M., Tatari, O., Mackie, K.R. and Elgamal, A. (2016). BIM-based Damage Estimation of Buildings under Earthquake Loading Condition. Procedia Eng., 145, pp. 1051-1058.

Chen, X., Saluz, U., Staudt, J., Margesin, M., Lang, W., and Geyer, P. (2022). Integrated data-driven and knowledge-based performance evaluation for machine assistance in building design decision support. In: 29th International Workshop on Intelligent Computing in Engineering, EG-ICE 2022, Aarhus, Denmark.

Jia, J., Gao, J., Wang, W., Ma, L., Li, J., and Zhang, Z. (2022). An Automatic Generation Method of Finite Element Model Based on BIM and Ontology. Buildings, 12(11), 1949.

Johansen, K., Schultz, C. and Teizer, J. (2022). BIM-based Fall Hazard Ontology and Benchmark Model for Comparison of Automated Prevention through Design Approaches in Construction Safety. In: 29th International Workshop on Intelligent Computing in Engineering, EG-ICE 2022, Aarhus, Denmark. 408-417.

Kelleher, J.D., Namee, B.M. and D'arcy, A. (2020). Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies (2nd Edition), Cambridge, MA, The MIT Press.

Kirchner, J. (2022). Computation of Ranges in Interval-based Constraint-Geometry of Building Models. In: 29th International Workshop on Intelligent Computing in Engineering, EG-ICE 2022, Aarhus, Denmark. 206-215.

Melchers, R.E. and Beck, A.T. (2017). Measures of Structural Reliability, Hoboken, NJ, John Wiley & Sons.

Mitchell, T.M. (1997). Machine Learning, New York, McGraw-Hill.

Park, S.I., Lee, S.H., Almasi, A. and Song, J.H. (2020). Extended IFC-based strong form meshfree collocation analysis of a bridge structure. Autom. Constr., 119, 103364.

Pauwels, P. (2020). IFCtoRDF (Version 0.4). https://github.com/pipauwel/IFCtoRDF, accessed April 2023.

Pauwels, P. and Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Autom. Constr., 63, pp. 100-133.

Ramaji, I. J. and Memari, A. M. (2018). Interpretation of structural analytical models from the coordination view in building information models. Autom. Constr., 90, pp. 117-133.

Rasoulzadeh S., Senk V., Kovacic I., Reisinger J., Füssl J., Hensel M. (2022). Linking Early Design Stages with Physical Simulations using Machine Learning Structural Analysis Feedback of Architectural Design Sketches. In: 29th International Workshop on Intelligent Computing in Engineering, EG-ICE 2022, Aarhus, Denmark. 216-226.

Shahinmoghadam, M., Motamedi, A. and Soltani, M. (2022). Leveraging Textual Information for Knowledge Graph-oriented Machine Learning: A Case Study in the Construction Industry. In: 29th International Workshop on Intelligent Computing in Engineering, EG-ICE 2022, Aarhus, Denmark. 259-268.

Sibenik, G. and Kovacic, I. (2020). Assessment of model-based data exchange between architectural design and structural analysis. J. Build. Eng., 32, pp. 101589.

Sibenik, G., Kovacic, I., Petrinas, V. and Sprenger, W. (2021). Implementation of Open Data Exchange between Architectural Design and Structural Analysis Models. Buildings, 11, pp. 605.

Vandewiele, G., Steenwinckel, B., Agozzino, T. and Ongenae, F. (2022). pyRDF2Vec: A Python Implementation and Extension of RDF2Vec. arXiv. https://arxiv.org/abs/2205.02283, accessed April 2023.

Vurtur Badarinath, P., Chierichetti, M. and Davoudi Kakhki, F. (2021). A Machine Learning Approach as a Surrogate for a Finite Element Analysis: Status of Research and Application to One Dimensional Systems. Sensors, 21, pp. 1654.

Wang, C., Jiang, C.G., Yu, S.X., Mckenna, F. and Law, K.H. (2019). NHERI-SimCenter/BIM2SAM.AI: Release v1.0 (Version 1.0). Zenodo. http://doi.org/10.5281/zenodo.3509957, accessed April 2023.

Xu, Z., Rao, Z., Gan, V.J.L., Ding, Y., Wan, C., and Liu, X. (2019). Developing an Extended IFC Data Schema and Mesh Generation Framework for Finite Element Modeling. Adv. Civ. Eng., 2019, 1434093.

Zhang, X.Y. and Hu, Z.Z. (2017). Research on model conversion approach towards structural finite element analysis. Gongcheng Lixue/Eng. Mech., 34, pp. 120-127.