Room Type Classification for Semantic Enrichment of Building Information Modeling Using Graph Neural Networks

Zijian Wang, zijian.wang@campus.technion.ac.il
Seskin Virtual Construction Lab, Faculty of Civil and Environmental Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel

Timson Yeung, timsonyeung@campus.technion.ac.il
Seskin Virtual Construction Lab, Faculty of Civil and Environmental Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel

Rafael Sacks*, cvsacks@technion.ac.il
*Seskin Virtual Construction Lab, Faculty of Civil and Environmental Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel (*corresponding author)*

Zhiya Su, zhiyasu@gmail.com
Seskin Virtual Construction Lab, Faculty of Civil and Environmental Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel

Abstract
Although Building Information Modeling (BIM) is widely adopted in the Architecture, Engineering, and Construction (AEC) industry, difficulties in model interoperability remain, hindering domain-specific applications. Semantic enrichment supplements a model's semantic information automatically as an alternative approach to solve the problem. In this research, a part of the EU funded 'Cloud-BIM' project, we applied a novel deep learning technique, graph neural networks (GNNs), to enrich a BIM model, with the goal of testing the applicability of graph representation and the use of GNNs for semantic enrichment. The application scenario concerned classification of room types in apartments. An apartment layout graph dataset containing seven room types and room relationships was compiled from 224 apartment layouts. A high-performance GNN model, GraphSAGE, was selected to train and predict room classes. Despite the small dataset and the sparse relationship features used in the experiment described here, GraphSAGE succeeded in classifying room types with an accuracy of 73%. This was 12% higher accuracy than two non-graph machine learning algorithms with which it was compared, and the prediction ability was better balanced. The findings provide clear directions for significantly improving the degree of accuracy by expanding the feature set of both node and edge properties, and this is the subject of ongoing work to refine the use of GNNs for semantic enrichment of BIM models.

Keywords: Building Information Modelling (BIM), Graph Neural Networks (GNNs), Semantic Enrichment (SE), Deep Learning, Machine Learning, Room Classification

1 Introduction

Interoperability in BIM is the ability to exchange model data across applications and domains. It improves the collaborative workflows among project participants by enabling smooth data exchange within the BIM environment throughout the project lifecycle (Sacks *et al.*, 2018). Interoperability is an essential feature of "Open BIM" systems and of future "Cloud BIM" systems, where project participants collaborate concurrently using a shared, federated project model held in a cloud repository (Afsari *et al.*, 2016). Current challenges in interoperability lie mainly in the loss of information during data exchange. Different applications utilize diverse and numerous information requirement specifications, and there is a lack of robust semi-automated or automated means to transfer domain knowledge.

Semantic enrichment refers to the automatic or semi-automatic addition of meaningful information to a building model through the extraction and interpretation of the explicit and implicit information in the model (Belsky *et al.*, 2016). SeeBIM 1.0, an early prototype software for semantic enrichment, was tested through application to precast concrete modeling (Belsky *et al.*, 2016) and design detailing for cost estimation (Aram, 2014). SeeBIM 1.0 demonstrated the feasibility of the semantic enrichment approach but faced limitations in the geometry approximation of complex shapes and the robustness of the IF-THEN rule sets (Sacks *et al.*, 2017). SeeBIM 2.0 removed these limitations by utilizing a BIM query language for spacial and topological operators and devising a robust rule compilation procedure that combined topological relationships and expert knowledge to generate unique identification strings for each object class (Ma et al., 2018).

All the previously discussed semantic enrichment approaches used rule-based inferencing, with explicit IF-THEN format rules collected from expert knowledge and experience. However, some problems, such as room function classification, do not lend themselves to rule inferencing (Bloch and Sacks, 2018). An alternative general approach is to enrich a model using machine learning (ML) algorithms, training models on labeled datasets (Neal, 2007). The selection of input features and the model architecture are integral to the performance of the models.

In this work, we explored a graph-based approach to semantic enrichment. In simple terms, a graph consists of nodes connected by edges, where nodes indicate objects and edges describe the relationships between nodes. As a non-Euclidean structured data format, graphs do not have a solid form. However, due to their ability to capture the underlying relationships among data, they have proven to be an effective medium for data storage, query, and analysis in academia and commerce (Scarselli *et al.*, 2009; Vicknair *et al.*, 2010).

Graph representation of building information is not a new concept. In the BIM field, researchers have explored the use of graph data models to perform data queries based on the Industrial Foundation Classes (IFC) schema (Ismail et al., 2017; Khalili & Chua, 2015; Tauscher et al., 2017) and performed pathfinding analysis using graph theory (Skandhakumar *et al.*, 2016). Graph theory has also been used to generate architectural floor plans (Wang *et al.*, 2018) and shape retrieval (Wessel *et al.*, 2008). Data mining was performed on the graph representation of building spaces using an unsupervised machine learning algorithm to extract spatial design knowledge from BIM data (Jin *et al.*, 2018). However, the full potential of graphs for BIM has yet to be appreciated yet, especially for the interoperability problem and semantic enrichment. Furthermore, progress in the development of graph neural networks (GNNs) and success in their application to previously challenging machine learning problems (e.g., natural language processing, chemical property prediction, online shopping recommendation tasks) suggest that there may be value in pursuing the application of GNNs to semantic enrichment of BIM model graphs.

2 Aims and methods

This paper introduces GNNs as a data processing method to the field of BIM. We aim to explore GNNs application to the semantic enrichment process by assessing its general performance in the room type classification problem and by comparing it against other machine learning methods.

2.1 Experiment design

Our experiment used a self-labeled dataset of apartment room layouts based on floor plans obtained from Internet sources as described in Section 3.1. One GNN model and two other machine learning approaches were trained and compared using the same dataset. Each model went through a learning process of train-validate-test using the labeled dataset as input. All models' performance was measured and compared using their predictions on an evaluation dataset.

Our experiment has several limitations. First, the construction of the labeled dataset is a manual process that is prone to human errors of inconsistency or mislabeling as it relies on the judgment of the labeler. Our ongoing research aims to resolve this concern by automating the dataset construction process. Second, since GNNs are still regarded as a new field of research in the deep learning field, and needless to say also in the BIM field, this experiment's contribution to GNNs in BIM should be regarded as exploratory. Furthermore, the lack of a recognized open dataset for BIM ML research means that our models' performance cannot be directly compared with other existing approaches to ML-based semantic enrichment.

2.2 Graph neural networks

The application of deep learning has revolutionized many tasks, such as natural language processing (Nadkarni *et al.*, 2011) and object detection (Bochkovskiy *et al.*, 2020). All these algorithms were designed for data represented in a Euclidean space, i.e., a data table or an image. However, many datasets have an underlying structure that is a non-Euclidean space (Ritter, 1999). A graph is a typical non-Euclidean structured data format, consisting of nodes and edges without a fixed form (Zhou *et al.*, 2018). Recently, using deep learning approaches to process graphs has attracted the attention of researchers, and a new technique, graph neural networks, has appeared (Scarselli *et al.*, 2009).

GNNs contain two main categories distinguished by their embedding learning mechanics. The first is spatial-based GNNs, generating new embeddings by considering nodes' spatial relationships. For example, GraphSAGE (Hamilton *et al.*, 2017) calculates the target node embedding by its previous neighborhood nodes' features. Another GNN category applies spectral graph theory from the perspective of graph signal processing (Ma *et al.*, 2019). These are called spectral-based GNNs, like Graph Convolutional Networks (GCN) (Kipf and Welling, 2017).

The common factor of all GNN models is that they take the relationships among data into account during learning, while these are usually ignored or hardly represented by other data formats. Because of this, GNNs are better at processing relationship-related tasks. Indeed, there are three main tasks in the GNNs domain, including node classification, edge prediction, and graph classification. Each of them has been applied in practical scenarios with satisfying performance. For example, social media companies utilize GNNs to recommend potential friends (Guo and Wang, 2021), where people are nodes and their relationships are edges. Similarly, researchers predicted the properties of chemical compounds (Kipf and Welling, 2017), a graph classification task. Due to the successful application of GNNs, we expect that GNNs can solve many BIM-field problems because relationships between entities are a structural feature of all BIM models.

2.3 Room type classifier

The proposed framework of room type classification is presented in Figure 1. The input is apartment layout graphs, whose nodes represent the rooms and whose edges indicate the relationships between rooms. A GraphSAGE model is constructed in this experiment, containing four hidden SAGEConv layers with Relu activation function. The output is node prediction of

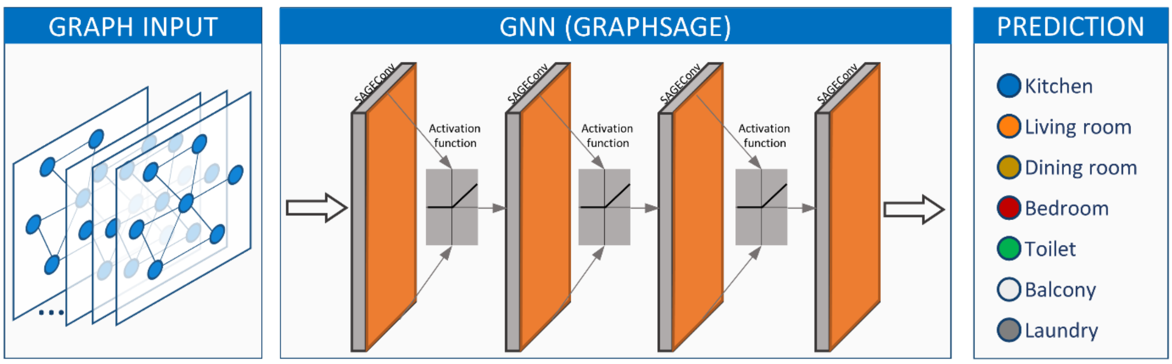


Figure 1. The proposed framework of room type classification based on a four layer GraphSAGE model.

seven room types, including a kitchen, living room, dining room, bedroom, toilet, balcony, and laundry. The classes of each node are hidden before inputting to the model, and all nodes participate in the training process; therefore, the experiment is a supervised node classification task.

GraphSAGE is a general inductive algorithm that leverages node features information (e.g., text attributes) to efficiently generate node embeddings for previously unseen data (Hamilton *et al.*, 2017). Instead of training individual embeddings for each node, the basic principle in GraphSAGE for generating node embeddings is to sample and aggregate features from itself and local neighborhoods. Therefore, GraphSAGE mainly relies on relationships among neighborhood nodes. It is similar to the process in which a person would identify a room type from relationship information alone. For example, we would likely consider a room as a living room if it connects directly with all other rooms in an apartment.

The GraphSAGE embedding update procedure contains three steps. In detail, the target node samples the neighboring nodes that have a direct connection with the target node. Then, the neighbors are fed into an aggregator function to generate neighbor features. There are different ways to design an aggregator function (e.g., mean, pooling, LSTM, etc.). Where graphs are large (thousands of nodes or more), to improve the calculation efficiency, only a subset of neighboring nodes is used by the aggregator function. However, in the room type classification task, the number of neighbors is small, and therefore the aggregator function utilizes all neighboring nodes. Lastly, the features obtained from the neighboring nodes and the feature of the target node from the previous layer are input to an update function to generate a new embedding.

3 Experiment

3.1 Dataset

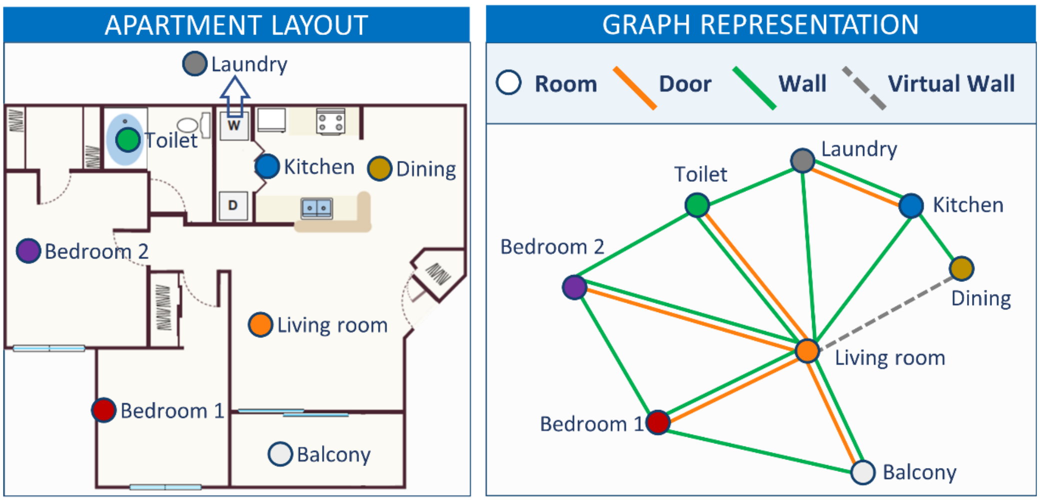


Figure 2. Schematic diagram of graph representation for an apartment layout. The lefthand image is an apartment layout, and the righhand image is its graph. Each room is regarded as a node and the relationships between rooms are edges. Three different relationships – door, wall and virtual wal connection – are used for graph construction.

At the time of writing, no open repository of BIM apartment models was available. The datasets from earlier research applying machine learning to BIM models (such as on room type classification (Bloch and Sacks, 2018) and object classification (Koo *et al.*, 2019)) were not published on open platforms. A convenient approach to compile BIM graph datasets is to extract information from IFC files and store the data in graphs directly. Ismail et al. (2017) and Pauwels et al. (2016) designed and implemented parsers to transfer data from IFC files to construct graphs, although the graphs were not directly amenable to machine learning. Therefore, there are two significant obstacles when constructing an apartment layout graph dataset: a) no repository of apartment models and b) no suitable parser are available. Given these limitations, we simplified the data construction process, constructing and labeling graphs manually from layout images of apartments. Research assistants identified the room types as nodes and the relationships between rooms as edges based on their professional experience and knowledge, and constructed graphs as shown in Figure 2.

Over 200 graphs were constructed based on the apartment layout images from three different countries (UK, US and China) from the Internet. Seven main room classes were adopted. And three relationships between rooms inside an apartment were utilized to describe 1) whether there is a door to access from room A to room B (door connection), 2) whether there is a shared wall (wall connection), and 3) where there are two independent spaces with no physical separation (virtual wall connection). Note that only relationships were adopted in this experiment; all other topological and geometrical features were ignored as the contribution of the relationships was the focus of the analysis. One reason is that the relationships between data are hard to be presented and calculated by other data formats and machine learning approaches, while the nature of GNNs is to consider the relationships between nodes. Relationships can theoretically be retrieved from or inferred from IFC files directly. It means that the procedure can be applied to BIM models after compiling a parser that converts IFC files to the desired graph format.

The statistics of the dataset are illustrated in Figure 3. The dataset includes 224 graphs, where 100 come from China and the other 124 are grouped as the UK and the US because apartments in these two countries have similar layouts. The seven main room types were selected as target prediction classes. The bedrooms had the most instances, 577 in total, while the laundry had the smallest number of occurrences, 64 and 81 respectively, in each of the sub-datasets.

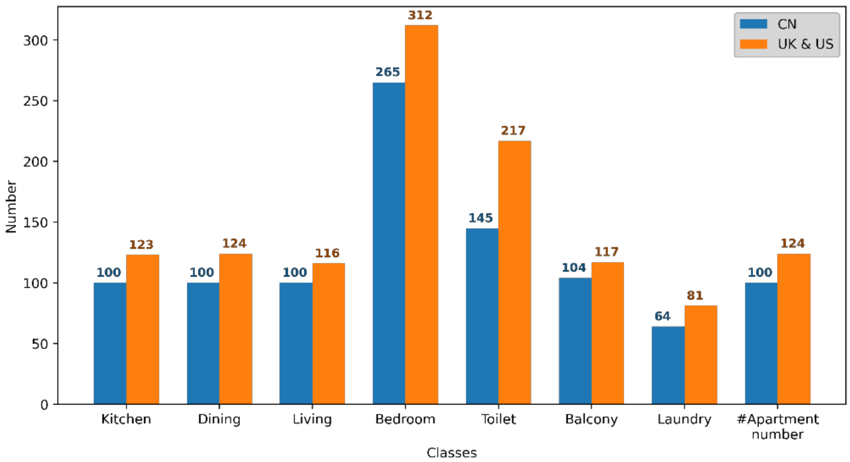


Figure 3. Distribution of room types in the dataset.

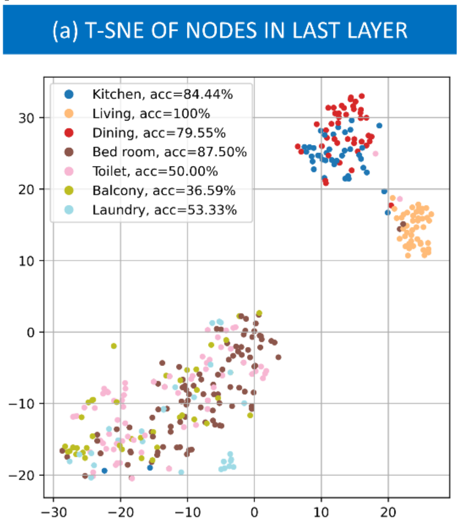
3.2 Training and testing

The dataset was split into training, testing and validation subsets. The training and testing were done on a personal computer with an Intel 6-core i7 CPU (2.70 GHz) and 32.0 GB RAM. We adopted Adam optimizer and cross-entropy to calculate the loss, and set the learning rate at 0.005 and the training epoch at 200 (Shchur *et al.*, 2018). We adopted accuracy to measure the correctness performance during the testing phase, as shown in equation (1), and the confusion matrix to present each class's predictive ability in the model.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Number\ of\ predictions} \quad (1)$$

4 Results and discussion

4.1 GraphSAGE results



(b) CONFUSION MATRIX OF GRAPHSAGE

Accuracy = 72.87%

Actual \ Predicted	KIV	LIV	DIN	BED	TOI	BAL	LAU
KIV	84%	0%	11%	0%	0%	4%	0%
LIV	0%	100%	0%	0%	0%	0%	0%
DIN	16%	5%	80%	0%	0%	0%	0%
BED	0%	0%	0%	88%	7%	4%	1%
TOI	0%	0%	0%	40%	50%	9%	1%
BAL	0%	0%	0%	27%	29%	37%	7%
LAU	0%	0%	0%	23%	10%	13%	53%

Figure 4. GraphSAGE results. (a) The t-SNE visualization of node embeddings in the last layer of the GraphSAGE model. (b) Confusion matrix and accuracy.

The feature embedding learned by the model can also be investigated quantitively, and therefore we provide a visualization of the t-SNE (van der Maaten and Hinton, 2008) used to degrade high-dimensional embeddings in the last layer of GraphSAGE into a two-dimensional map, as shown in Figure 4 (a). Each color presents a room class. The top right three classes, kitchen, living, and dining, are almost clearly differentiated, demonstrating that the model already has the ability to distinguish among them. The bedroom class, shown by brown points scattered in the lower left quadrant of the chart, has an accuracy of 87.5%. In contrast, toilet, balcony, and laundry classes were misclassified to the bedroom class and distributed around it.

The performance is also reflected in the confusion matrix shown in Figure 4 (b). Although the first four classes have an accuracy of over 80%, the last three classes cannot be distinguished. In detail, 40%, 27%, and 23% of the toilet, balcony, and laundry class instances were classified incorrectly as bedrooms. Furthermore, these three classes have a high probability of being misclassified as one of the other two classes. For example, 29% and 7% of balcony instances were classified as a toilet or a laundry, respectively. One reason may be that the three classes connect to the living room by door and wall connections with similar feature patterns, and they have fewer edges than living and bedroom classes.

Generally speaking, GraphSAGE accurately classified kitchen, living, dining, and bedroom classes, while its performance with the other classes was relatively poor. We emphasize that in this experiment, only relationships were used as the features for learning, and these were a limited set of the characteristics of entities in the model. The levels of accuracy achieved are considered reasonable as it is challenging even for a person to classify a room type based solely on the three relationships.

4.2 Comparison with machine learning algorithms

We also selected two machine learning algorithms to compare the performance, including a decision tree (Smola and Schölkopf, 2004) and a support vector machine (Song and Lu, 2015). The two models were trained based on the same dataset with the same features. We fine-tuned their hyperparameters (e.g., learning rate, epochs) to achieve their best performance under this dataset. The test results are listed in Table 1.

The two models achieved accuracies of 55.9% and 57.9% respectively. These results are around 15% lower than those achieved using GraphSAGE. For the decision tree model, although the accuracy of the kitchen class was over 80%, many other class instances were misclassified as kitchens. For example, a majority of the living room instances (58%) were wrongly predicted as a kitchen. Even worse, the accuracy of toilet, balcony, and laundry classes was under 50%, where most instances were output as a bedroom. A similar situation also happens to the SVM model. In contrast, only the balcony's accuracy was under 50% in the GraphSAGE model, and most classes achieved an accuracy of over 80%. After analysis, it is clear that GraphSAGE has higher accuracy and a more balanced ability to classify similar classes when compared with the two machine learning approaches.

Table 1. Performance of the state-of-the-art ML algorithms for room classification.

Model	**Decision Tree**							**Support Vector Machine**						
Accuracy	**55.84%**							**57.87%**						
Confusion Matrix	**Predicted**							**Predicted**						
Actual	KIV	LIV	DIN	BED	TOI	BAL	LAU	KIV	LIV	DIN	BED	TOI	BAL	LAU
KIV	83%	7%	3%	7%	0	0	0	86%	7%	0	7%	0	0	0
LIV	58%	39%	3%	0	0	0	0	52%	47%	2%	0	0	0	0
DIN	0	0	100%	0	0	0	0	0	0	100%	0	0	0	0
BED	1%	0	0	68%	21%	9%	1%	0	0	0	64%	25%	11%	0
TOI	0	0	0	47%	34%	12%	7%	0	0	0	41%	47%	12%	0
BAL	0	0	0	50%	9%	41%	0	0	0	0	25%	34%	41%	0
LAU	0	0	0	33%	58%	0	8%	0	0	0	13%	88%	0	0

5 Conclusion

The main contributions of this paper are the following:

- Application of a new deep learning technique, graph neural networks, to BIM models expressed as graphs. GNNs consider the relationships between data, which are usually ignored by other data formats. Due to this natural mechanism, GNNs have the potential to solve challenging problems in BIM where relationships among entities play an essential role.
- Compilation of an apartment room function and relationship dataset and its use to illustrate how to apply GNNs in a BIM context. The experiment results showed that GraphSAGE achieved 73% accuracy using the dataset and only using sparse node features.
- Comparison of the GNN model with two machine learning algorithms under the same dataset and same features. GraphSAGE performed better than the other two machine learning approaches, with 15% higher accuracy.

The presented research is only a first step in applying GNNs to BIM models. In future work, we think it is worthwhile to explore the following three aspects:

- Although four of the classes' performance in the GraphSAGE model is acceptable, the other three class predictions still have much room for improvement. One direction is to improve the model's prediction ability on all classes and have a more balanced performance.
- We propose to do this by adapting the algorithm to consider edge features as well as node features. Current GNN models can only process node features because node-classification open datasets do not have edge features. Yet edge features are necessary when building BIM graph datasets. Constructing an improved GNN model that can take both node and edge features into account can fill in the gap and improve the performance, in this and, presumably, in other similar problems.
- Compilation and publication in the public domain of an open BIM graph dataset. Publishing an extensive BIM graph dataset will allow others to seek better GNN type solutions.

Acknowledgments

This work is part of the Cloud-based Building Information Modelling (CBIM) project, a European Training Network. The CBIM project receives funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant with agreement No 860555.

References

Afsari, K., Eastman, C. and Shelden, D. (2016), "Cloud-Based BIM Data Transmission: Current Status and Challenges", *ISARC 2016 - 33rd International Symposium on Automation and Robotics in Construction*, pp. 1073–1080.

Aram, S. (2014), *A Knowledge-Based System Framework For Semantic Enrichment And Automated Detailed Design In The Aec Projects*, PhD Thesis, Georgia Institute of Technology, Atlanta, available at: https://smartech.gatech.edu/handle/1853/53496.

Belsky, M., Sacks, R. and Brilakis, I. (2016), "Semantic Enrichment for Building Information Modeling", *Computer-Aided Civil and Infrastructure Engineering*, Vol. 31 No. 4, pp. 261–274.

Bloch, T. and Sacks, R. (2018), "Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models", *Automation in Construction*, Elsevier, Vol. 91 No. July 2017, pp. 256–272.

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y.M. (2020), "YOLOv4: Optimal Speed and Accuracy of Object Detection", *ArXiv*, available at: http://arxiv.org/abs/2004.10934 (accessed 13 May 2020).

Guo, Z. and Wang, H. (2021), "A Deep Graph Neural Network-Based Mechanism for Social Recommendations", *IEEE Transactions on Industrial Informatics*, IEEE Computer Society, Vol. 17 No. 4, pp. 2776–2783.

Hamilton, W.L., Ying, R. and Leskovec, J. (2017), "Inductive representation learning on large graphs", *Advances in Neural Information Processing Systems*, Vol. 2017-Decem, pp. 1025–1035.

Ismail, A., Nahar, A. and Scherer, R. (2017), *Application of Graph Databases and Graph Theory Concepts for Advanced Analysing of BIM Models Based on IFC Standard*, *Digital Proceedings of the 24th EG-ICE International Workshop on Intelligent Computing in Engineering 2017*, available at: (accessed 4 April 2021).

Jin, C., Xu, M., Lin, L. and Zhou, X. (2018), "Exploring BIM Data by Graph-based Unsupervised Learning", *ICPRAM 2018 - Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods*, Vol. 2018-Janua No. Icpram, pp. 582–589.

Khalili, A. and Chua, D.K.H. (2015), "IFC-based graph data model for topological queries on building elements", *Journal of Computing in Civil Engineering*, Vol. 29 No. 3, pp. 1–14.

Khalili, A. and Chua, D.K.H. (2016), *Generic BIM Queries Based on the IFC Object Model Using Graph Theory*, *Journal of Computing in Civil Engineering*, Vol. 29, available at: http://ascelibrary.org/doi/10.1061/%28ASCE%29CP.1943-5487.0000331 (accessed 14 February 2021).

Kipf, T.N. and Welling, M. (2017), "Semi-supervised classification with graph convolutional networks", *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, available at: http://arxiv.org/abs/1609.02907 (accessed 26 January 2021).

Koo, B., La, S., Cho, N.W. and Yu, Y. (2019), "Using support vector machines to classify building elements for checking the semantic integrity of building information models", *Automation in Construction*, Elsevier B.V., Vol. 98, pp. 183–194.

Ma, L., Sacks, R., Kattel, U. and Bloch, T. (2018), "3D Object Classification Using Geometric Features and Pairwise Relationships", *Computer-Aided Civil and Infrastructure Engineering*, Blackwell Publishing Inc., Vol. 33 No. 2, pp. 152–164.

Ma, Y., Hao, J., Yang, Y., Li, H., Jin, J. and Chen, G. (2019), "Spectral-based graph convolutional network for directed graphs", *ArXiv*, 21 July, available at: http://arxiv.org/abs/1907.08990 (accessed 16 February 2021).

van der Maaten, L. and Hinton, G. (2008), "Visualizing data using t-SNE", *Journal of Machine Learning Research*, Vol. 9, pp. 2579–2625.

Nadkarni, P.M., Ohno-Machado, L. and Chapman, W.W. (2011), "Natural language processing: an introduction", *Journal of the American Medical Informatics Association*, Vol. 18 No. 5, pp. 544–551.

Neal, R.M. (2007), "Pattern Recognition and Machine Learning", *Technometrics*, Springer New York, NY, USA, Vol. 49 No. 3, pp. 366–366.

Pauwels, P. and Terkaj, W. (2016), “EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology”, *Automation in Construction*, Elsevier, Vol. 63, pp. 100–133.

Ritter, H. (1999), “Self-Organizing Maps on non-euclidean Spaces”, *Kohonen Maps*, Elsevier, pp. 97–109.

Sacks, R., Eastman, C., Lee, G. and Teicholz, P. (2018), *BIM Handbook*, *John Wiley & Sons*, John Wiley & Sons, Inc., Hoboken, New Jersey, available at:https://doi.org/10.1002/9781119287568.

Sacks, R., Ma, L., Yosef, R., Borrmann, A., Daum, S. and Kattel, U. (2017), “Semantic Enrichment for Building Information Modeling: Procedure for Compiling Inference Rules and Operators for Complex Geometry”, *Journal of Computing in Civil Engineering*, Vol. 31 No. 6, p. 04017062.

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M. and Monfardini, G. (2009), “The graph neural network model”, *IEEE Transactions on Neural Networks*, Institute of Electrical and Electronics Engineers Inc., Vol. 20 No. 1, pp. 61–80.

Shchur, O., Mumme, M., Bojchevski, A. and Günnemann, S. (2018), “Pitfalls of Graph Neural Network Evaluation”, *ArXiv*, available at: https://www.kdd.in.tum.de/gnn-benchmark (accessed 26 January 2021).

Skandhakumar, N., Salim, F., Reid, J., Drogemuller, R. and Dawson, E. (2016), “Graph theory based representation of building information models for access control applications”, *Automation in Construction*, Elsevier B.V., Vol. 68, pp. 44–51.

Smola, A.J. and Schölkopf, B. (2004), “A tutorial on support vector regression”, *Statistics and Computing*, Vol. 14 No. 3, pp. 199–222.

Song, Y.Y. and Lu, Y. (2015), “Decision tree methods: applications for classification and prediction”, *Shanghai Archives of Psychiatry*, Editorial Department of theShanghai Archives of Psychiatry, Vol. 27 No. 2, pp. 130–135.

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y. and Wilkins, D. (2010), “A comparison of a graph database and a relational database”, *Proceedings of the 48th Annual Southeast Regional Conference on - ACM SE ’10*, ACM Press, New York, New York, USA, p. 1.

Wang, X.Y., Yang, Y. and Zhang, K. (2018), “Customization and generation of floor plans based on graph transformations”, *Automation in Construction*, Elsevier, Vol. 94 No. August, pp. 405–416.

Wessel, R., Blümel, I. and Klein, R. (2008), “The room connectivity graph: Shape retrieval in the architectural domain”, *16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG’2008 - In Co-Operation with EUROGRAPHICS, Full Papers*, pp. 73–80.

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., *et al.* (2018), “Graph Neural Networks: A Review of Methods and Applications”, *ArXiv*, 20 December, available at: http://arxiv.org/abs/1812.08434 (accessed 8 February 2021).