

Room-based Energy Demand Classification of BIM Data using Graph Supervised Learning

H. Kiavarz¹, M. Jadidi¹, A. Rajabifard², G. Sohn¹

¹ Geomatics Engineering, Department of Earth & Space Science & Engineering, York University, Toronto, Canada - (hkiavarz, mjadidi, gsohn)@yorku.ca

² Department of Infrastructure Engineering, Melbourne School of Engineering, University of Melbourne, Melbourne, Australia – abbas.r@unimelb.edu.au

KEYWORDS: BIM, Knowledge Graph, Machine Learning, GraphSAGE, Building Energy Efficiency

ABSTRACT

Nowadays, cities and buildings are increasingly interconnected with new modern data models like the 3D city model and Building Information Modelling (BIM) for urban management. In the past decades, BIM appears to have been primarily used for visualization. However, BIM has been recently used for a wide range of applications, especially in Building Energy Consumption Estimation (BECE). Despite extensive research, BIM is less used in BECE data-driven approaches due to its complexity in the data model and incompatibility with machine learning algorithms. Therefore, this paper highlights the potential opportunity to apply graph-based learning algorithms (e.g., GraphSAGE) using the enriched semantic, geometry, and room topology information extracted from BIM data. The preliminary results are demonstrated a promising avenue for BECE analysis in both pre-construction step (design) and post-construction step like retrofitting processes.

1. INTRODUCTION

The Building Information modeling (BIM) describes the physical characteristics of building elements utilizing their three-dimensional (3D) geometry, semantic, and topology data. Geometric and semantic data represent the individual properties (e.g., location, dimension, and material) of building elements. However, topological data denotes spatial relationships among the building elements, including connection, adjacency, containment, separation, and intersection (Ohori *et al.*, 2017). The well-known BIM open standard model is Industry Foundation Classes (IFC). IFC provides the interoperability of BIM data across construction, engineering, and architecture domains that share the building information to serve at different applications, e.g., building security management, facility management, emergency pathfinding, and energy efficiency (Chong, Lee and Wang, 2017). Recently, the use of IFC as a 3D data source for Building Energy Consumption Estimation (BECE) has gained momentum (Andriamamonjy, Saelens and Klein, 2018; Pezeshki and Ivani, 2018). Current states of the arts demonstrate the role of detailed geometrical and semantic information in predicting building energy consumption in both The literature review also illustrates the value of 3D information in providing detailed indoor building information useful for data-driven approaches (Fumo, 2014; Bourdeau *et al.*, 2019). In this application, the indoor space concept (known as the *IfcSpace* class) can be used as a sub-unit of buildings. The spatial links between *IfcSpace* objects are considered topology information, leading to the development of a knowledge graph for analyzing the energy transfer from one room to another. In the knowledge graph, the node can represent spaces in which the semantic information of each space (node) is defined as vector information assigned to the node. The edges in the knowledge graph connect a pair of spaces and capture the spatial relationship if there is any energy transfer. Also, knowledge graphs have started to play a central role in machine learning to incorporate real-world objects with a relationship for knowledge extraction and phenomena prediction. There are many physical and data-driven modeling

techniques available that can be used to model the energy consumption of buildings (Bourdeau *et al.*, 2019).

Data-driven models emerge as the most suitable option for the BECE analysis rather than classical physics-based modeling (Li *et al.*, 2010). Moreover, the recent researchers employed detailed information from the IFC model for BECE analysis. However, most of them use traditional machine learning models for BECE analysis and ignore the topological information (e.g., adjacent room/*IfcSpace* information) in their learning process, which causes inaccurate results in BECE data-driven models. Because the energy is transferred between the adjacent rooms if two rooms have an adjacency relationship using a shared wall, window, roof, or floor. For example, if a room has a shared wall with a cold room (with low energy efficiency), the heating loss rate increases significantly from the warm area to the cold area (Fan, Xiao and Zhao, 2017). Therefore, the spatial relationship between the spaces is vital in BECE-based analysis, which recent studies ignore. Because of the complexity of topological information in the IFC model and unmaturing graph-based data-driven methods in BECE-based analysis. This paper adopted the GraphSAGE algorithm as a Graph Neural Network (GNN) machine learning model for room-based BECE analysis to apply space properties and topological information in the learning process. Finally, the algorithm classifies the rooms in a multi-level building into two energy-efficient and inefficient classes with determining the probability of each class. The proposed algorithm improves the accuracy of the BECE analysis because of utilizing the room's properties and their relationship in the learning process, which are extracted from the IFC model.

The rest of the paper is organized as follows: The proposed methodology is presented in Section 2. Section 3 describes the experimental results, and Section 4 concludes the paper and future study.

2. METHODOLOGY

This section discusses the proposed methodology to tackle the challenge of learning algorithms in data-driven models. There are

two main steps in the below sections. First, a framework is proposed to generate a space-based knowledge graph from the IFC model. Then, in the second step, a graph-based classification algorithm is implemented to show how we can apply the proposed graph's learning method by involving the room's geometrical, semantical, and neighborhood information.

2.1 A Framework to Generate Space-based Knowledge Graph

The knowledge graph (KG) represents a collection of interlinked real-world entities and their properties (Wu *et al.*, 2020). Knowledge graph includes nodes as entities and the links which represent the relationship of entities. Indeed, from year to year, Machine Learning and Knowledge Learning on Graphs are advancing expeditiously, both in scale and depth, but in different directions. On the one hand, Machine Learning techniques are getting better at performing various tasks (e.g., Classification and Prediction.) on different datasets with great precision. Moreover, the Knowledge Graph provides an infrastructure for data to be organized into connected graph structures, and thus multi-source and heterogeneous data can be interlinked and integrated (Abbad and Bouchaib, 2017). This study adopts this concept to convert 3D data models (IFC) to a Knowledge Graph to provide an intermediate interconnected data structure linked to different datasets and compatible with machine learning algorithms. The proposed graph contains room (space) object information as a node, their properties as an assigned vector, and their relationships with neighbor rooms as an edge.

This paper's primary intent is to classify the indoor rooms of building into two efficient and non-efficient classes from energy consumption. Then, the classification method will be applied to the proposed knowledge graph, including each room's geometrical, semantical, and relational information. Therefore, the proposed framework generates the knowledge graph using the IFC file for the final classification task. The framework encompasses two main modules: Feature Extraction and Knowledge Graph Construction, as presented in Figure 1, implemented by Python using the IfcOpenShell and NetworkX libraries (Newman, 2003).

Feature Extraction is the first module to extract all geometrical and semantical information from the IFC file of building rooms (space). This module includes four main functions. The first function (FG) extracts each room's accurate geometrical (Volume, Area, and Perimeter) information from the IFC file. The second function (FS) extracts and calculates each room's semantical information (Thermal Resistance Index – R-Value). The third function (FNB) finds adjacent rooms for the target room by considering the shared wall, roof, and floor. Moreover, the fourth function (Feature Generator (FFG)) creates a vector list of feature values for each room, along with a GUID as a unique Id.

KnowledgeGraph_Construction is the second module of the framework. This module constructs the adjacency matrix (Bapat, 2010) and the knowledge graph based on the first module's output. It includes three main functions. The process starts with the Normalization function (FN) to normalize the feature values. Since each parameter has a different scale, the normalization must create common scale feature values for machine learning algorithms (Becerik-Gerber *et al.*, 2014). Therefore, we applied the min-max normalization method using F_N (Patro and sahu, 2015). The second function (F_A) generates the adjacency matrix using the given neighbour list from module one. Indeed, the adjacency matrix represents the knowledge graph as a square matrix with the size of $n * n$ (n : number of nodes). Finally, the F_{KG} function is designed to loop through the adjacency matrix to find the neighbour nodes. In

this step, we have used the Networkx library in Python to construct the knowledge graph using the adjacency matrix and embed six feature values as each node's attributes. The generated graph is heterogeneous graph because nodes have different edges as the room's neighborhood. This knowledge graph will be used for the next step for graph based classification.

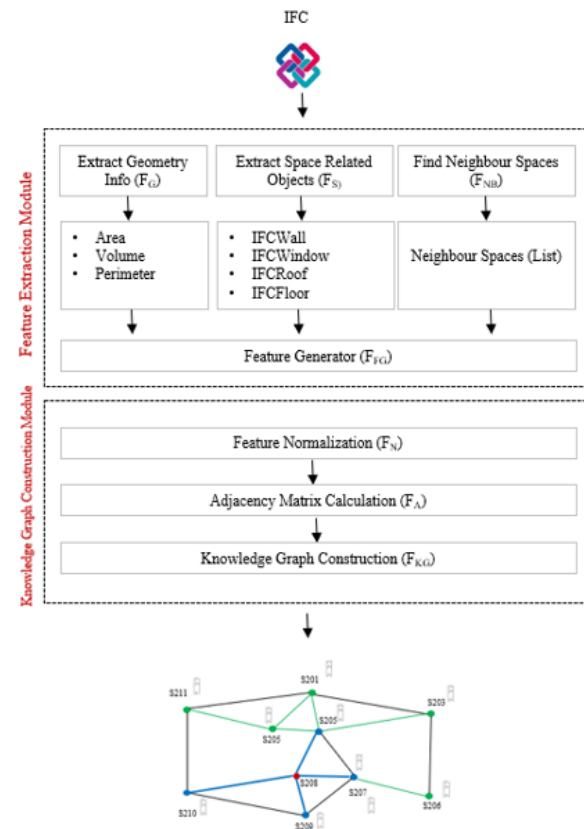


Figure 1. Knowledge Graph Framework

2.2 Room Classification based on GraphSAGE Learning Algorithm using BECE features

This section introduces a solution to adopt the GraphSAGE learning algorithm (Xu *et al.*, 2018) based on the generated knowledge graph from the IFC model. The primary intent is to show how a learning algorithm can be employed on an IFC-based knowledge graph include building room's properties such as Total-Wall-Area, External-Wall-Area, Window-Area, External-Wall-R-Value, Internal-Wall-R-Value, Window-R-Value, and topology information between rooms. Instead of using traditional machine learning classification tasks, we consider using a graph neural network (GNN) to perform node classification problems to classify the rooms (nodes) of the knowledge graph into two classes of efficient and inefficient by calculating the probability values for the classes for each room. By providing an explicit link between the rooms, the classification method is no longer classified the rooms independently, such as traditional BECE-based learning algorithms but leveraging graph structures such as the degree of rooms and neighborhood information. The usefulness of graph properties assumes that individual rooms are correlated with other rooms. The GraphSAGE method as a supervised classification method is trained based on training nodes (80% of nodes in the graph), then the trained model predicts the efficiency class of the other nodes (rooms). Eventually, the model accuracy is measured by comparing the predicted efficiency class and the actual efficiency class of testing nodes.

GraphSAGE is an inductive learning algorithm capable of predicting a new node's attributes without requiring a re-training procedure. To do so, GraphSAGE learns aggregator functions that can induce a new node's attributes, given its features and neighborhood. This is called inductive learning, suitable for both supervised and unsupervised node classification (Xu *et al.*, 2018). Therefore, we have chosen GraphSAGE as our graph-based classification algorithm. We need to classify our rooms into two classes as a supervised learning process in our application by determining the probability of each class for each room. Thus, GraphSAGE can be a suitable GNN method for our use case. We applied the GraphSAGE classification task to our BECE-based knowledge graph into three main parts as **context construction**, **information aggregation**, and **Learning process by loss function** described below:

2.2.1 Context Construction: The algorithm has a parameter K that controls the neighborhood depth. If K is 1, only the adjacent room are involved in the learning process. If K is 2, the rooms at walk depth two are considered. Remark that having $k = 2$ means rooms at neighborhood depth two can affect each other through the room in the middle. The value of K is determined experimentally using multiple neighborhoods. Figure 5a shows an example of information sharing in GraphSAGE with a neighborhood depth of two for node 3 of the whole knowledge graph

2.2.2 Information Aggregation: Having defined the neighborhood, now we need an information-sharing procedure between neighbors. Therefore, in the first step, we generate a computational graph (Dondi, Mauri and Zoppis, 2018) for each room in the graph to calculate new embedding (feature) values for the target room. Next, aggregation functions or aggregators accept the neighborhood rooms as input and aggregates the neighbor's attributes (features) with weights to create a neighborhood embedding for the target node. Aggregator weights are either learned or fixed depending on the function. To learn embeddings with aggregators, we first initialize all room features' embeddings to node features as node attributes. In turn, for each neighborhood depth until K , we create a node embedding with the aggregator function for each node. Different aggregation functions are LSTM aggregator, Pooling aggregator, and Mean aggregator (Hamilton, Ying, and Leskovec, 2017). We have chosen the Mean aggregator for our calculation because of its simplicity in the implementation. Equation 1 demonstrates the Mean aggregation function in which h^{k-1} shows the feature values of the neighbor rooms and $|N(v)|$ is the number of the neighborhood of room v (Xu *et al.*, 2018).

$$AGG_{u \in N(v)} = \frac{h_u^{k-1}}{|N(v)|} \quad (1)$$

It means, each room has a feature vector with a size of $6 * 1$ in our research, and after aggregation, it generates an embedding feature of node 3 with the size of $6 * 1$. Also, node 3 has its feature values with the size of $(6 * 1)$ in Layer-1. After concatenation, a feature vector with the size of $(12 * 1)$ is generated, which is the neural network's input layer (gray-box B in Figure 5b). When each node is processed, we normalize the embeddings to have a unit norm. Equation 2 represents the aggregation and concatenation of the target node using the neighborhood's features and the target node's features. In Equation 2, h_v^k denotes, as an embedding features node v in walk depth K and σ represents the activation function.

$$h_v^k \leftarrow \sigma(w \cdot MEAN(\{h_v^{k-1}\} \cup \{h_v^{k-1}, \forall u \in N(v)\})), \quad (2)$$

We apply the activation function to add nonlinearity to our model. In this research, we apply the Sigmoid activation function (Shrikumar, Greenside and Kundaje, 2017) because its output is between 0 and 1, which is suitable to calculate the probability of output classes (inefficient and efficient rooms). The algorithm is implemented by python language with PyTorch, NumPy, DGL, Panda, and sklearn libraries in the google colab environment. Then, the concatenated vector passed to the neural network layer (Figure 5b-gray box B) to update the node embedding. As a result, the neural network in box B is designed with 12 neurons (features), two hidden layers, and two neurons (efficient and inefficient) for the output layer.

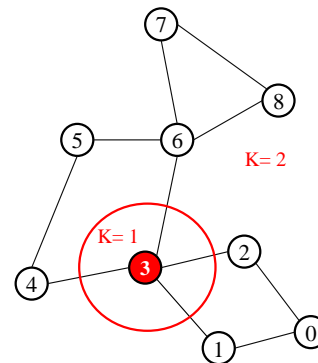


Figure 2. Two neighborhood depth of target room 3

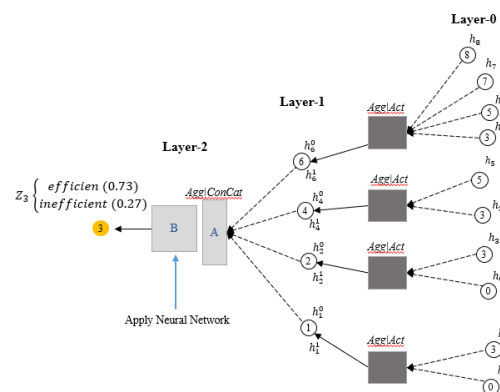


Figure 3. Computation graph of Node 3 (neighbor depth - $k=2$)

2.2.3. Learning process by the loss function: We have applied the aggregation steps to generate node embeddings and the learning process. Nevertheless, to learn the neural network weights, we need a differentiable loss function to calculate the distance between the actual value of node class and the predicted values. We have applied the Squared Error Loss (SE) function (Choromanska *et al.*, 2015) for each room classification (130 nodes in our dataset). Then, we split the nodes into a training set (100 nodes) and a testing set (30 nodes). The predicted process takes input features from each computation graph and calculates the probability for each room. The distance of real and predicted output value is measured by the SE function for 100 nodes called loss value. The mean of loss values (Mean Squared Error – MSE (loss function)) for 100 nodes is calculated for each iteration (epoch). At the end of each epoch, the neural network's metrics weight is adjusted by the backpropagation process. The learning process is continued iteratively to catch the best accuracy on training nodes.

3. EXPERIMENTAL RESULTS AND DISCUSSION

Autodesk office building (Trapelo), which is located in Massachusetts in the United States, is chosen as a case study dataset. The BIM model of this building is a commercial three-story building with 130 rooms which is downloaded in IFC format from the Open IFC Model Repository. This dataset's completeness of indoor information (LOD4) and entity-relationship (topology) motivated us to choose this IFC file as a case study. The knowledge graph is generated for this dataset with 130 nodes and a 6-dimension vector assigned. Then train GraphSAGE classification based on training data and measure the accuracy of the result for test data. We calculate the classification accuracy from 1000 epoch to 5000 and get the best accuracy of 86.6% in epoch 3500. Therefore, we pick up the weight matrices in epoch 3500. It means 26 rooms of 30 in the test dataset are assigned to the correct class, and only 4 rooms are wrongly classified.

4. CONCLUSION AND FUTURE WORK

This research adopted GraphSAGE as an inductive learning algorithm with a BECE-based classification task. The promising result demonstrated that the proposed solution helps the decision-maker evaluate each room's energy efficiency in buildings with a large area. This algorithm can evaluate the energy consumption of the rooms in the building, which has not been built yet (design level), retrofitting tasks, and helping to redesign with adding new rooms in a building or aggregated with the other rooms and evaluating the room efficacy. Since the model is trained and tested by a single building, we need to investigate the result of the algorithm by applying an enriched dataset. Also, this research considers the similar weight value for rooms relationships in the learning process. However, in the future, the weighted knowledge graph can be employed for classification tasks to consider accurate energy transformation between the rooms.

REFERENCES

Abbad, H. and Bouchaib, R., 2017. Towards a Big Data Analytics Framework for Smart Cities. *Proceedings of the Mediterranean Symposium on Smart City Application*, pp. 17:1--17:5. doi: 10.1145/3175628.3175647.

Andriamamonjy, A., Saelens, D. and Klein, R., 2018. An automated IFC-based workflow for building energy performance simulation with Modelica. *Automation in Construction*. doi: 10.1016/j.autcon.2018.03.019.

Bapat, R. B., 2010. Adjacency Matrix. *Graphs and Matrices*. doi: 10.1007/978-1-84882-981-7_3.

Becerik-Gerber, B. *et al.*, 2014. Civil engineering grand challenges: Opportunities for data sensing, information analysis, and knowledge discovery. *Journal of Computing in Civil Engineering*. doi: 10.1061/(ASCE)CP.1943-5487.0000290.

Bourdeau, M. *et al.*, 2019. Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustainable Cities and Society*. doi: 10.1016/j.scs.2019.101533.

Chong, H. Y., Lee, C. Y. and Wang, X., 2017. A mixed review of the adoption of Building Information Modelling (BIM) for sustainability. *Journal of Cleaner Production*. doi: 10.1016/j.jclepro.2016.09.222.

Choromanska, A. *et al.*, 2015. The loss surfaces of multilayer

networks. *Journal of Machine Learning Research*.

Dondi, R., Mauri, G. and Zoppis, I., 2018. Graph Algorithms, in *Encyclopedia of Bioinformatics and Computational Biology. ABC of Bioinformatics*. doi: 10.1016/B978-0-12-809633-8.20424-X.

Fan, C., Xiao, F. and Zhao, Y., 2017. A short-term building cooling load prediction method using deep learning algorithms. *Applied Energy*. doi: 10.1016/j.apenergy.2017.03.064.

Fumo, N., 2014. A review on the basics of building energy estimation. *Renewable and Sustainable Energy Reviews*. doi: 10.1016/j.rser.2013.11.040.

Hamilton, W. L., Ying, R. and Leskovec, J., 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*.

Li, X. *et al.*, 2010. A novel hybrid approach of KPCA and SVM for building cooling load prediction. *3rd International Conference on Knowledge Discovery and Data Mining, WKDD 2010*. doi: 10.1109/WKDD.2010.137.

Newman, M. E. J., 2003. The Structure and Function of Complex Networks. *SIAM Review*, 45(2), pp. 167–256. doi: 10.1137/S003614450342480.

Ohori, K. A. *et al.*, 2017. TOWARDS AN INTEGRATION of GIS and BIM DATA: WHAT ARE the GEOMETRIC and TOPOLOGICAL ISSUES?. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(4W5), pp. 1–8. doi: 10.5194/isprs-annals-IV-4-W5-1-2017.

Patro, S. G. K. and sahu, K. K., 2015. Normalization: A Preprocessing Stage. *IARJSET*. doi: 10.17148/iarjset.2015.2305.

Pezeshki, Z. and Ivvari, S. A. S., 2018. Applications of BIM: A Brief Review and Future Outline. *Archives of Computational Methods in Engineering*. Springer Netherlands, 25(2), pp. 273–312. doi: 10.1007/s11831-016-9204-1.

Shrikumar, A., Greenside, P. and Kundaje, A., 2017. Learning important features through propagating activation differences. *34th International Conference on Machine Learning, ICML 2017*.

Wu, Z. *et al.*, 2020. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*. doi: 10.1109/tnnls.2020.2978386.

Xu, K. *et al.*, 2018. Representation learning on graphs with jumping knowledge networks. *35th International Conference on Machine Learning, ICML 2018*.