Transforming Construction with Reality Capture Technologies:
The Digital Reality of Tomorrow

August 23-25, 2022, Fredericton, New Brunswick, Canada

# IFC-BIM-BASED CRITICAL ENERGY CONSUMPTION ZONES IDENTIFICATION IN BUILDINGS

Kiavarz, H[1*] and Jadidi, M[1*]

[1] Geomatics Engineering, Department of Earth & Space Science & Engineering, York University, Toronto, Canada

[*] hkiavarz@yorku.ca

**Abstract:** A recent wave of interest in building energy consumption and estimation has inbred a considerable amount of energy data and building information, which boosts the data-driven algorithms for broad application throughout the building industry. Also, cities and buildings are increasingly interconnected with modern data models like the 3D city model and Building Information Modelling (BIM) for urban management these days. 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. Unfortunately, despite extensive research, BIM is less used in 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 to find the critical zones in the perspective of energy consumption in different spaces of the building. The preliminary results demonstrated a promising finding of critical zones in buildings that improve pre-construction and post-construction steps.

**Keywords: GIS, BIM, IFC, Energy Efficiency, Machine Learning**

## 1 INTRODUCTION

The energy consumption rate has increased considerably over the last decade worldwide (Cao, Dai, and Liu 2016). Specifically, the building sector alone consumes more than 40% of the global energy. Therefore, efficiently designed and operated buildings can realize essential energy savings. Heating load (HL) and cooling load (CL) are measures of energy that must be added or removed from space by a heating ventilation and air conditioning (HVAC) system to provide the desired level of thermal comfort within space. Therefore, early predictions of building CL and HL can help engineers design energy-efficient buildings that need accurate building entity information to predict critical zones in the buildings (Chou and Bui 2014). In order to tackle this issue, recent research and building development projects have been focused on using new data models such as Building Information Models (BIM). As a result, the applied data-driven solutions emerge as the most suitable option for the Building Energy Consumption Estimation (BECE) analysis rather than employing classical models to improve the estimation and prediction of building energy consumption in different stages of design development and retrofitting. However, despite extensive research, the current data-driven models suffer from data model incompatibility with new modern models like BIM. Therefore, they ignore using BIM in their analysis and lose the detailed information between building entities and the 3D topology information. Therefore, this research investigates the advantage of BIM and its digital representation IFC (Industry Foundation Classes) in finding critical zones in terms of energy consumption at the level of room entity. To tackle the data model compatibility issue, a novel object-oriented framework is proposed as an intermediate data model to solve the incompatibility issue. Then, the object-oriented model is converted to generate a graph-based data model compatible with machine learning (ML) and deep learning (DL) algorithms. As a result, a graph-based classification algorithm is designed to classify the rooms into two classes of critical and non-critical zones by considering the room's properties and the topology between them.

On the other hand, most researchers who employ data-driven models use traditional machine learning algorithms for building energy estimation and ignore the topological information (e.g., adjacent room) in their learning process, which causes inaccuracy. Energy is transferred between the adjacent rooms if two rooms have an adjacency relationship using a shared wall, window, floor, or roof; 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 to cold areas (Fan, Xiao, and Zhao 2017). Therefore, the indoor spatial relationship between the rooms is essential for this analysis, which has been ignored in recent studies. One of the reasons for neglecting such information is the complexity of topological information in the IFC model and unmatured graph-based data-driven methods. In building energy estimation analysis, the indoor space (room) concept (known as the IfcSpace class) can be used as a sub-unit of buildings. The spatial links between rooms are considered topology information, leading to the development of a room-based graph for analyzing the energy transfer from one room to another. In the 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 graph connect a pair of rooms and capture the spatial relationship if there is any energy transfer. Also, 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 (Yuan et al. 2020).To consider room information and its adjacency in the learning process, this paper adopted the GraphSAGE algorithm as an inductive Graph Neural Network (GNN) machine learning model for room-based classification. Finally, the algorithm classifies the rooms in a multi-level building into two critical and non-critical classes by determining the probability of each class. The proposed algorithm improves the accuracy of the room-based classification because it utilizes the room's properties and their relationship in the learning process. A comparative analysis is performed with well-known non-graph-based classification methods. The rest of the paper is organized, and 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 proposed methodology is discussed in this section. The graph-based data model and proposed learning algorithms for room-based classification tackle the mentioned challenges in finding the critical zones in the building. There are two main steps in the below sections. First, a framework is proposed to generate a room-based graph from the IFC model. Then, in the second step, a graph-based classification algorithm is adopted with the generated graph from the IFC model to show how we can apply the proposed graph's learning method by involving the room's geometrical, semantical, and neighborhood information.

### 2.1 An Object-Oriented Framework to Generate Room-based Graph

This paper employed the graph concept (Hamilton, Ying, and Leskovec 2017) to convert 3D data models (IFC) to a room-based graph to provide a data model to be linked to different datasets and compatible with machine learning algorithms. As a case study, we use an IFC file as a building 3D data model to construct

a graph. The proposed graph contains room object information as a node, their relationships with neighbor rooms as edge, and the other data sources like entity material information with different data models. Our primary intent is to classify the indoor rooms of building into two critical and non-critical classes from the energy consumption perspective. The classification method is applied to the proposed graph, including each room's geometry, semantic, and relational information. Therefore, the proposed framework generates the graph using the IFC file for future classification. The framework encompasses two main modules: Feature Extraction and Graph Construction, as presented in Figure 1, implemented by Python using the IfcOpenShelland NetworkX libraries (Andriamamonjy, Saelens, and Klein 2018).
**Feature Extraction** is the first module to extract all geometrical and semantical information of building rooms (space) from the IFC file. This module includes four main functions. The first function extracts each room's geometry (FG) (Volume, Area, and Perimeter) information from the IFC file. The second function ($F_S$) extracts and calculates semantical information (Thermal Resistance Index) for each room. The third function ($F_{NB}$) finds adjacent rooms for the target room by considering the shared wall. Moreover, the fourth function (Feature Generator ($F_{FG}$)) creates a list of feature values (important parameters) for each room, along with a GUID as a unique Id (Figure 1).

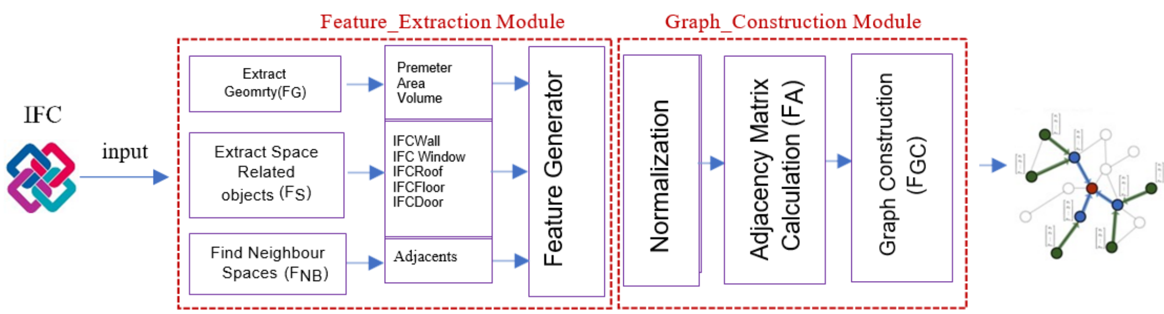


Figure 1: The workflow of the proposed framework

**Graph_Construction** is the second module of the framework. This module constructs the adjacency matrix (Bapat 2010) and the room-based graph from the first module's output. It includes three main functions. The process starts with the Normalization function ($F_N$) 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 FN's min-max normalization method (Patro and Sahu 2015). The second function ($F_A$) generates the adjacency matrix using the neighboring objects (room) list in module one. Indeed, the adjacency matrix represents the graph as a square matrix with the size of n * n (n: number of nodes). Finally, the $F_{GC}$ function is designed to loop through the adjacency matrix to find the neighbor nodes. In this step, we have used the Networkx library in Python to construct the graph using the adjacency matrix and embed six feature values as each node's attributes. The generated graph is homogeneous because all nodes have equivalent types (room) and similar edges to the room's neighborhood. This graph will be used in the next step in applying machine learning classificationalgorithm.

## 2.2 Finding Critical Zone using GraphSAGE Algorithm

This paper aims to apply the room classification task regarding energy consumption to find the critical zones in the building using a graph-based learning algorithm that includes room information such as Total-Wall-Area, External-Wall-Area, Window-Area, External-Wall-R-Value, Internal-Wall-R-Value, Window-R-Value the topology information between the rooms. Graph Neural Network (GNN) node classification is the room classification task in building energy analysis. For room classification, we applied an inductive GNN method based on GraphSAGE (Xu et al. 2018). The reason for selecting an inductive method is that inductive learning builds a generic model where any new node (room) would be predicted based on an observed set of training data points. However, transductive learning, such as GCN (Graph Convolutional Network), builds a model that fits the training and testing data points it has already observed, which is not proper for our research application (Yan et al. 2019). For example, in retrofitting process, the structure of the building is changed, a lot of new rooms are added, and some rooms are demolished. Therefore, we need to design and train a model applied to new data and rooms. Instead of 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 graph into two classes by calculating the probability values for each room. By providing an explicit link between the rooms, the classification method no longer classifies the rooms independently, such as traditional building energy estimation 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. We applied the GraphSAGE classification task to our room-based 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 is 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.

### 2.2.2 Information Aggregation

The information-sharing procedure between neighbors is needed in this step. 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 aggregate the neighbor's attributes (features) with weights to create a neighborhood embedding for the target node. To learn embeddings with aggregators, we 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). The mean aggregator for our calculation is 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).

$$AGG_{u \epsilon N(v)} = \frac{{h_u}^{k-1}}{|N(v)|} \qquad \text{(Eq. 1)}$$

Each room has a feature vector with a size of 6 * 1 for this test case, and after aggregation, it generates an embedding feature of node 3 with the size of 6 * 1. We normalize the embeddings when each node is processed to have a unit norm. Equation 2 represents updating node embedding calculation using the neighborhood's and target node's features. In Equation 2, $h_v^k$ denotes, as an embedding features node $v$ in walk depth k and $\sigma$ represents the activation function

$$h_v^k \leftarrow \sigma(w.MEAN\left(\{h_v^{k-1}\} \cup \{h_v^{k-1}, \forall u \in N(v)\}\right) \qquad \text{(Eq. 2)}$$

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 for calculating the probability of output classes.

### 2.2.3 Traning the GraphSAGE by the loss function

To train the neural network weights in GraphSAGE, we need a differentiable loss function to calculate the distance between the actual value of the node class and the predicted values. Therefore, we have applied the Squared Error Loss (SE) function (Wu et al. 2018) 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 actual 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 CASE STUDY AND RESULT

To test the proposed methodology, the Autodesk office building (Trapelo), which is located in Massachuset in the United States, is chosen as a case study dataset (Figure 2). The BIM model of this building is a commercial three-story building with 130 rooms downloaded in IFC format from the Open IFC Model Repository. The graph is generated for this dataset with 130 nodes and a 6-dimension vector. Then train GraphSAGE classification based on training data and measure the accuracy. Accuracy refers to the fraction of the number of correct predictions over the number of all samples regardless of classes of the result for test data (Table 1). 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 four rooms are wrongly classified. Finally, the accuracy is calculated for GraphSAGE with two and three neural network layers and two none-graph-based (ANN and SVM) methods to compare the result of the proposed method with other classification methods. The GraphSAGE with three hidden layers performs better from a perspective of accuracy.

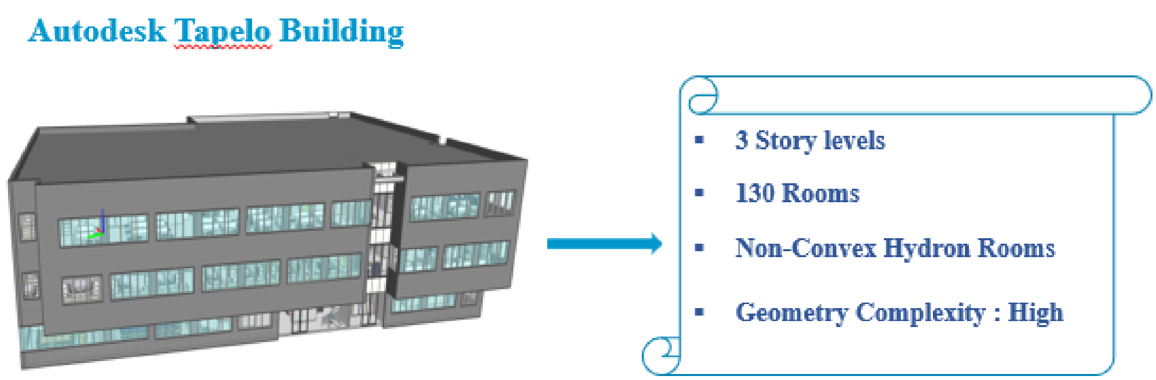


Figure 2: Case study BIM Model

Table 1: Classification Models Accuracy

| Model | **Accuracy** |
|---|---|
| GraphSAGE (2 layers) | 83.2 |
| GraphSAGE (3 layers) | 86.6 |
| ANN | 72.6 |
| SVM | 71.8 |

## 4 CONCLUSION AND FUTURE WORK

This paper proposed an object-oriented framework to convert the IFC data model to a room-based one. Also, it adopted GraphSAGE as an inductive learning algorithm for a room 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), retrofit tasks, and help to redesign by adding new rooms in a building or aggregating with the other rooms to evaluate 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. Although we get high accuracy results from the graph-based classification method, designers must understand and interpret how and why our models make their predictions. For example, to understand the reason behind the critical zone, we need to design an explainable model to interpret the essential parameters of each critical zone which is considered for our future work in this study.

## REFERENCES


Andriamamonjy, Ando, Dirk Saelens, and Ralf Klein. 2018. "An Automated IFC-Based Workflow for Building Energy Performance Simulation with Modelica." *Automation in Construction*. https://doi.org/10.1016/j.autcon.2018.03.019.

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

Becerik-Gerber, Burcin, Mohsin K. Siddiqui, Ioannis Brilakis, Omar El-Anwar, Nora El-Gohary, Tarek Mahfouz, Gauri M. Jog, Shuai Li, and Amr A. Kandil. 2014. "Civil Engineering Grand Challenges: Opportunities for Data Sensing, Information Analysis, and Knowledge Discovery." *Journal of Computing in Civil Engineering*. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000290.

Cao, Xiaodong, Xilei Dai, and Junjie Liu. 2016. "Building Energy-Consumption Status Worldwide and the State-of-the-Art Technologies for Zero-Energy Buildings during the Past Decade." *Energy and Buildings* 128 (September): 198–213. https://doi.org/10.1016/j.enbuild.2016.06.089.

Chou, Jui-Sheng, and Dac-Khuong Bui. 2014. "Modeling Heating and Cooling Loads by Artificial Intelligence for Energy-Efficient Building Design." *Energy and Buildings* 82 (October): 437–46. https://doi.org/10.1016/j.enbuild.2014.07.036.

Dondi, Riccardo, Giancarlo Mauri, and Italo Zoppis. 2018. "Graph Algorithms." In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*. https://doi.org/10.1016/B978-0-12-809633-8.20424-X.

Fan, Cheng, Fu Xiao, and Yang Zhao. 2017. "A Short-Term Building Cooling Load Prediction Method Using Deep Learning Algorithms." *Applied Energy*. https://doi.org/10.1016/j.apenergy.2017.03.064.

Hamilton, William L., Rex Ying, and Jure Leskovec. 2017. "Inductive Representation Learning on Large Graphs." In *Advances in Neural Information Processing Systems*.

Patro, S.Gopal Krishna, and Kishore Kumar sahu. 2015. "Normalization: A Preprocessing Stage." *IARJSET*. https://doi.org/10.17148/iarjset.2015.2305.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje. 2017. "Learning Important Features through Propagating Activation Differences." In *34th International Conference on Machine Learning, ICML 2017*.

Wu, Lijun, Fei Tian, Yingce Xia, Yang Fan, Tao Qin, Jianhuang Lai, and Tie Yan Liu. 2018. "Learning to Teach with Dynamic Loss Functions." In *Advances in Neural Information Processing Systems*.

Xu, Keyulu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken Ichi Kawarabayashi, and Stefanie Jegelka. 2018. "Representation Learning on Graphs with Jumping Knowledge Networks." In *35th International Conference on Machine Learning, ICML 2018*.

Yan, Xiongfeng, Tinghua Ai, Min Yang, and Hongmei Yin. 2019. "A Graph Convolutional Neural Network for Classification of Building Patterns Using Spatial Vector Data." *ISPRS Journal of Photogrammetry and Remote Sensing*. https://doi.org/10.1016/j.isprsjprs.2019.02.010.

Yuan, Hao, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2020. "Explainability in Graph Neural Networks: A Taxonomic Survey," 1–15. http://arxiv.org/abs/2012.15445.