

## Article

# An Automated Space-Based Graph Generation Framework for Building Energy Consumption Estimation

Hamid Kiavarz <sup>1,\*</sup>, Mojgan Jadidi <sup>1,\*</sup> , Abbas Rajabifard <sup>2</sup>  and Gunho Sohn <sup>1</sup><sup>1</sup> Geomatics Engineering, Lassonde School of Engineering, York University, Toronto, ON M3J 1P3, Canada<sup>2</sup> Department of Infrastructure Engineering, Melbourne School of Engineering, University of Melbourne, Melbourne, VIC 3010, Australia

\* Correspondence: hkiavarz@yorku.ca (H.K.); mjadidi@yorku.ca (M.J.)

**Abstract:** The 3D information in Building Information Modeling (BIM) has received significant interest for smart city applications. Recently, employing Industry Foundation Classes (IFC) for BIM in data-driven methods for Building Energy Consumption Estimation (BECE) has gained momentum because of the enriched geometric and semantic information. However, despite extensive studies on applying the IFC data in BECE analysis, employing the full potential of the BIM remains poor due to its complex data model and incompatibility with data-driven algorithms. This paper proposes a framework to extract accurate semantic, geometry, and topology information from the room-level (space) IFC schema by introducing new geo-computation algorithms to deal with these challenges. Additionally, we define a new topological weighted relationship between spaces in different stories by combining common geometry area with energy resistance value. Eventually, the proposed weighted space-based graph will be constructed to decrease the original complexity of the IFC model, and it is compatible with graph-based machine learning algorithms. The results are promising, with more than 90% accuracy in extracting the geometry information for the convex and non-convex polyhedron rooms and 100% accuracy in detecting vertical and horizontal adjacent rooms. This study confirms the proposed approach's efficiency, accuracy, and feasibility for space-based BECE analysis.

**Keywords:** BIM; machine learning; graph theory; building energy consumption

**Citation:** Kiavarz, H.; Jadidi, M.; Rajabifard, A.; Sohn, G. An Automated Space-Based Graph Generation Framework for Building Energy Consumption Estimation. *Buildings* **2023**, *13*, 350. <https://doi.org/10.3390/buildings13020350>

Academic Editor: Junxiang Zhu

Received: 25 December 2022

Revised: 17 January 2023

Accepted: 24 January 2023

Published: 27 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

3D information has dominated the world of building modeling using Building Information Modeling (BIM) for more than a decade [1]. BIM includes the 3D representation of the building's semantic and geometry information and the relationships of the elements [2]. A series of standards are introduced to represent the BIM data model; the most well-known digital standard of BIM is the Industry Foundation Classes (IFC) released by the International Alliance for Interoperability (IAI), known as BuildingSMART Alliance (BSA) [3]. The IFC model is an object-oriented data model incorporating 3D geometry, spatial relationships (topology), quantities, and building element properties (semantic). IFC facilitates the interoperability problems in the different disciplines and shares almost all the building components in the construction and building management domain [4,5]. Different disciplines, such as facility management and energy efficiency studies, need accurate geometry, semantics, and topology as preliminary information to lead to reliable and accurate results. Furthermore, the 3D information in IFC models has received significant interest in today's application domain in smart cities and smart buildings [6]. Recently, IFC as a 3D data source for Building Energy Consumption Estimation (BECE) has gained momentum, according to a study by [7].

Indeed, BECE vitally depends on the geometry, topology, and material information of the building's elements to recognize the spaces in a building with high energy demand [8,9], which leads to evaluating different building design scenarios and operation plans. The

physical modeling and the data-driven approach are widely used for BECE [9]. Physical models (also known as white-box models) rely on thermodynamic rules for detailed energy modeling and analysis. These models calculate building energy consumption based on detailed building and environmental parameters such as building physical details, construction schedules, and HVAC design [8]. On the other hand, data-driven methods estimate building energy consumption based on learning from historical data and building properties [9,10]. However, two shortcomings limit the BECE physical models. First, the high number of required input physical parameters that need to be measured by deploying sensors, and second, they are not designed to use geometry information, one of the influential parameters to estimate energy consumption [11,12].

On the other hand, data-driven methods can accept BECE-based analysis hyperparameters, making them flexible in using various data sources in their process [5]. Despite extensive studies on data-driven methods, employing the IFC model as a 3D model remains poor due to its complexity and incompatibility with data-driven methods such as machine learning algorithms. This paper intends to answer the research question: How can the IFC data model be compatible with BECE data-driven model? Indeed the IFC model includes the indoor environment's primary semantic, geometry, and topology information which are of considerable interest for data-driven methods. Especially by describing the indoor space concept (known as the *IfcSpace* class) in the IFC standard model and defining spatial links between them open the potential for BECE analysis in different parts of the building. Indeed, it is possible to create a space-based graph directly to analyze the energy transfer between them. In this graph, the nodes denote spaces in which the property information of each space (node) is defined as vector information assigned to the node. An edge in the graph connects a pair of spaces and captures the spatial relationship of spaces if there is any energy transfer [13]. Graphs have also begun to play a crucial role in machine learning to incorporate real-world objects with relationship information for knowledge extraction and phenomenon prediction [14]. BECE analysis based on machine learning algorithms in space scale depends on indoor geometry, semantic, and relationship parameters such as relative compactness, shading coefficient (SC), the overall heat transfer coefficient of building walls (R-Value), roof heat transfer coefficient (R-Value), the absorption coefficient for solar radiation of exterior walls, window-wall ratio (WWR), height of space, orientation, and glazing area distribution [9,15]. Unfortunately, the spaces (*IfcSpace*) in most existing IFC models have inaccurate geometric information and space-based topological descriptions for BECE-based analysis. Therefore, the mentioned feature information is not explicitly available in the IFC model and needs to be computed from a combination of semantic and geometry information. Moreover, possible inconsistency between geometric and semantic information causes inaccurate results when using the IFC dataset in machine learning algorithms. Therefore, a study is emergent to bridge this gap, consider existing models' limitations, and propose a solution for the IFC data model's compatibility with machine learning algorithms.

This paper proposes a framework to generate an object-oriented data model as an intermediate data model to translate the IFC model to the BECE space-based graph, including BECE-related information and spatial relationship between IFC spaces. In this research, we employed IFC specification 4.0.2.1 under ISO 16739-1:2018 because the IFC models are available under this version for the case study [16]. The proposed framework extracts and calculates implicit BECE-related information by introducing new objects, such as a wall's inner and outer face objects, roof, window, floor, and related geometry information. At the same time, the framework introduces accurate topological information, such as the spatial relationship between spaces horizontally and vertically. Furthermore, in contrast to other studies considering just the binary relationship between the IFC elements, the framework calculates the weighted relationship using energy transfer between the spaces. Undeniably, the advantage of using the proposed intermediate object-oriented data model is the possibility of extracting implicit BECE-related information from IFC by preserving the spatial relationship (topology information) between the spaces. Thus, the framework

extracts the complex IFC data model, including the buildings' meaningful semantics for BECE, geometry, and spatial relationship, into a BECE-based graph. The main benefits of the proposed methodology are: (1) to enable building space objects in IFC to be automatically translated into a BECE-based data model; (2) to enhance the hidden values in the original IFC to make it usable in building energy estimation without human interference; and (3) to develop a more reliable and accurate BECE space-based graph from IFC models to be compatible with machine learning algorithms.

The paper is structured as follows. Section 2 provides state-of-the-art, while Section 3 lists the challenges and shortfalls. Section 4 discusses the suggested methodology, which enables us to consider the indoor 3D space and its topology data. Section 5 contains the data sources, case studies, and outcomes. Finally, Section 6 summarises the lessons learned, concluding remarks, and path for further research.

## 2. Related Work

Data-driven methods benefit from developments in machine learning, such as providing flexibility and reliability in modeling and forecasting building energy consumption [15,16]. Therefore, many studies have been conducted on the semantics and geometry data exchange between BIM and BECE using standard data schemas (IFC). Briefly, three types of solutions can be distinguished [17,18].

- Type I: Application Programming Interface (API) into BIM-based software or employing an integration solution for integrating a BECE simulation engine into the BIM tool, such as Revit.
- Type II: Use BECE simulation software to import the pertinent data from the BIM into a file utilizing the gbXML format [19,20].
- Type III: To import the IFC format into BECE tools, export the full BIM, such as Revit format, to the IFC format [18,21].

Jeong et al. (2014) in [22] applied type one in their simulation using REVIT API and the Modelica Building Library. The most significant disadvantage of this strategy is the mismatch between the two environments and the reliance on a particular version of proprietary software. Cemesova, Hopfe, and McLeod (2015) in [23] employ the second type in their analysis and use commercial software to convert the BIM format to the gbXML data model. This model has restricted geometry-defining capabilities even though it keeps the topology information of the BIM data (e.g., preserving only rectangular shapes based on the element centerline). Therefore, it can support convex polyhedron shapes and ignore non-convex polyhedrons, leading to a non-accurate estimation of the building energy consumption [24]. Most parts of the third type of solution focus on geometry representation despite enriched information from the IFC schema, such as the face geometry properties and space topological information. All of them are considered implicit in the Type III solution.

Consequently, the IFC model's complexity and incompatibility with data-driven methods restrict being used in BECE data-driven methods. Furthermore, most published approaches only concentrate on one particular component of BIM data (for example, geometry or HVAC) and employ conventional physical simulation engines as a result of the difficulties noted with using the IFC model in BECE data-driven methodologies (e.g., EnergyPlus) [25]. Therefore, utilizing an efficient data model to have the following three characteristics is necessary: (1) the capability of the data model to be connected to different data sources; (2) the capability of the data model to preserve geometric, semantic, and relationship information; and (3) compatibility of the data model with the machine learning algorithm.

## 3. Challenges in Extracting the BECE-Based Geometry, Semantic and Topology Information from the IFC Model

The current state of the art in estimating building energy consumption recognized the beneficial use of IFC models for extracting the building semantics, geometry information, and 3D spatial relationship between the spaces [25,26]. Indeed, efficient 3D analysis of

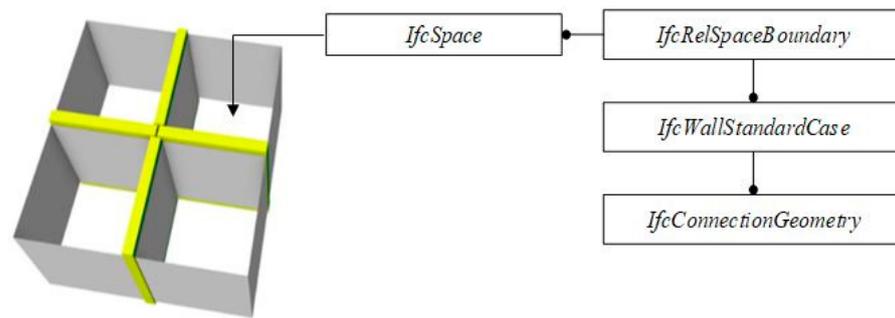
building energy consumption relies on the indoor environment's semantic information, the geometric description of the IFC entities, and their spatial relationships. The geometry and material information of the entities such as walls, windows, roof, and floor are the main entities of each room in IFC, affecting the room's heat gain and loss. Therefore, taking the precise geometry and room's properties, such as material information of these entities for each room, is necessary. Most IFC-based building energy modeling relies on the information stored in the model. However, incomplete or erroneous implementation of the IFC and implicit topological information, typically found in many IFC export modules, cause inaccurate geometry information, such as each space's wall and window area. Additionally, recognizing accurate topological information, such as the spatial relationship between the spaces, is hard to extract from the IFC models. These errors generate an invalid graph in the perspective of extracting BECE-based information and wrong space spatial relationships. In this situation, sophisticated geometric processing is required to produce the desired graphs from only geometric data. In detail, the challenges mentioned above regarding the spaces' geometry and topology information in the IFC model are categorized as two main problems: invalid geometry information in the *IfcSpace* object and inaccurate space adjacency (topology) information described in the following sections.

### 3.1. Illustrating Invalid Geometry Information in the *IfcSpace* Object

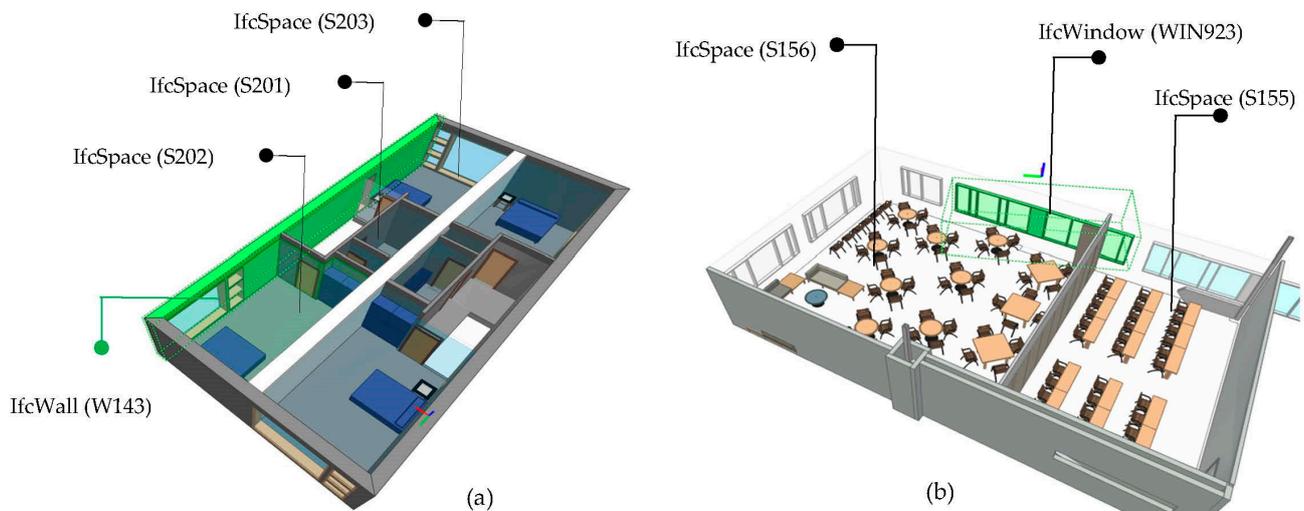
To evaluate the energy consumption of the different spaces of the buildings using data-driven approaches and the IFC model, we use the concept of space representing the concept of room [25]. Space splits the buildings into different parts and represents an area or volume bounded. Therefore, the geometry information of each space and its related entities, such as windows, walls, doors, roofs, and floors which significantly influence the energy load, should be extracted accurately to improve a room's energy efficiency analysis [27]. The primary geometry information of these entities is listed as the Window–Wall Ratio (WWR), Exterior–Wall Ratio (EWR), Interior–Wall Ratio (IWR), Floor–Room Ratio (FRR), and Roof–Room Ratio (RRR), the total area of the room, and the total height of the room.

At the high-quality IFC model, it is sufficient to rely on the query of *IfcSpace* objects described in an IFC model to retrieve 3D indoor geometry information to extract each room's wall, window, roof, and floor geometry properties. Figure 1 illustrates an example of high-quality semantic and geometrical modeling of *IfcSpaces* and the schema of *IfcSpace* in which the walls are split geometrically for each room (yellow lines). In this model, a semantic-based query using the *IfcOpenShell* library extracts accurate geometry information, such as the wall area for each room, because the walls are split geometrically based on the boundary of each space. However, it is common for *IfcSpace* objects to contain the erroneous entity's geometry despite the standard definitions and implementation instructions established to point toward appropriate IFC models. Most IFC models do not have the second boundary definition of the objects (split geometry) for spaces and inaccurate geometry calculations [28]. Figure 2a represents an IFC model with the non-topological definition of walls and windows for the corresponding spaces. This figure shows a shared wall (W143) between three spaces (S202, S201, and S203). In this case, the query returns an equal value of 3.84 m<sup>2</sup> of the wall-143 for each room, equal to the total area of the wall-143 because it is not split based on space's boundaries. If we rely on this type of erroneous geometry value and apply them in the BECE process, it impacts the analysis of each room's energy consumption.

Additionally, the window is a critical entity in a building with a central role in energy gain or loss. In this regard, windows are revealed to impact room energy consumption and contribute to more than 10% of the building's energy load. If a room includes an exterior low-efficient window, the room's energy loss is increased drastically. Therefore, the energy consumption of a room directly depends on the window size, location, and resistance rate [29]. For example, Figure 2b represents a shared window (WIN923) between S155 and S156. Performing a query from the existing IFC model returns an equal area value of window (WIN923) for both rooms.



**Figure 1.** Valid IFC definition of the wall for each space.

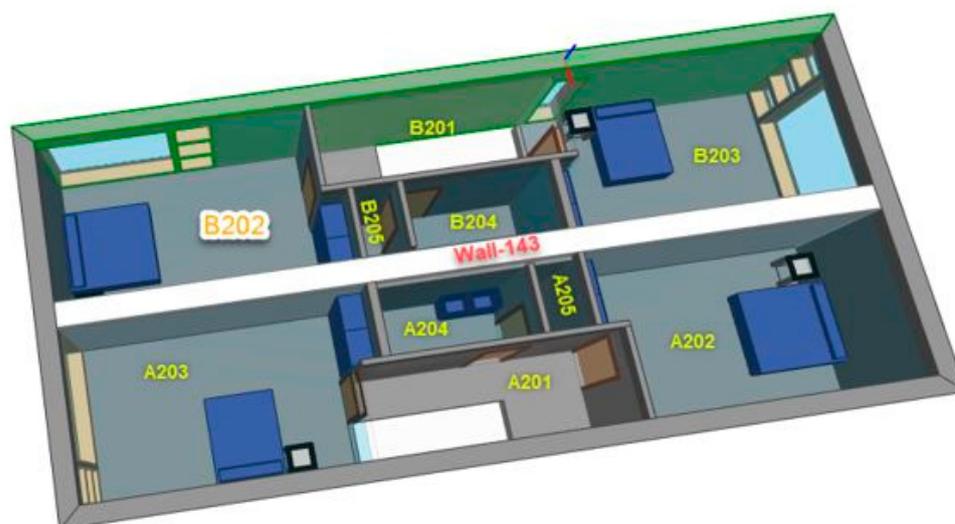


**Figure 2.** Invalid geometry of the wall and window for each space in the IFC models: (a) Illustration of the shared wall (green wall) between multiple IfcSpace objects; (b) Illustration of shared windows (green windows) between multiple IfcSpace objects.

In contrast, the window size in S156 is significantly larger than in S155. Therefore, the impact of window-923 on the heating loss and gain in room-156 is more than its impact on room-155, which is not recognized if we rely on the current IFC query result. This leads to the necessity of geometric computation to extract accurate geometry information for space geometry information before using them in data-driven methods.

### 3.2. Identifying Inaccurate Space Adjacency (Topology) Information in the IFC Model

Two spaces have adjacency in a 3D model when at least one shared wall, window, roof, or floor between them exists. In the domain of BECE, the energy transferred between the adjacent rooms affects the energy demand of each room. For instance, the heating loss rate from the warm area to the cold area increases noticeably if a room shares a wall or floor (such as the basement) with a cold room (with low energy efficiency). Therefore, extracting accurate adjacent information is essential. Nevertheless, the IFC semantic-based query cannot return the valid adjacent information because of each space's invalid (non-split geometry) wall, roof, window, and floor geometry. Figure 3 illustrates the instance of this error which results in the wrong adjacency list.



**Figure 3.** The IFC model (LOD04) includes space and wall objects.

Table 1 describes a semantic-based query using the current information in the IFC model. It employs the *BoundedBy* property of space objects to retrieve the *IfcWall* and *IfcCurtainWall* objects bounded to each space. By iterating through all space around the target space (B202 in Figure 3), the neighbor space objects are added to the adjacent list and considered adjacent. The result shows the wrong adjacent spaces for the target room because there is a continued wall-143 between the target room, rooms A202 and B203. Indeed, no geometrical relationship is available between them, which causes no energy transfer; however, the IFC semantic query considers their adjacent spaces. Therefore, retrieving the topological information based on the semantic query from the IFC model is inaccurate and causes erroneous results for generating the BECE space-based graph.

**Table 1.** IFC semantic-based query to extract space adjacency with the concept of the shared wall.

Pseudo Code	Result
<code>Adjacent_list = [ ]</code>	
<code>spaceB202_walls = [ ]</code>	
<code>foreach wall in space_B202.BoundedBy:</code>	Space B201
<code>// include ifcwall and ifcCurtainWall</code>	Space B205
<code>spaceB202_walls.append(wall)</code>	Space A203
<code>foreach floor_space in spaces_in_floor:</code>	Space B203
<code>foreach wall in floor_space, BoundedBy:</code>	Space A202
<code>if wall in spaceB202_walls:</code>	
<code>Adjacent_list.append(floor_space)</code>	

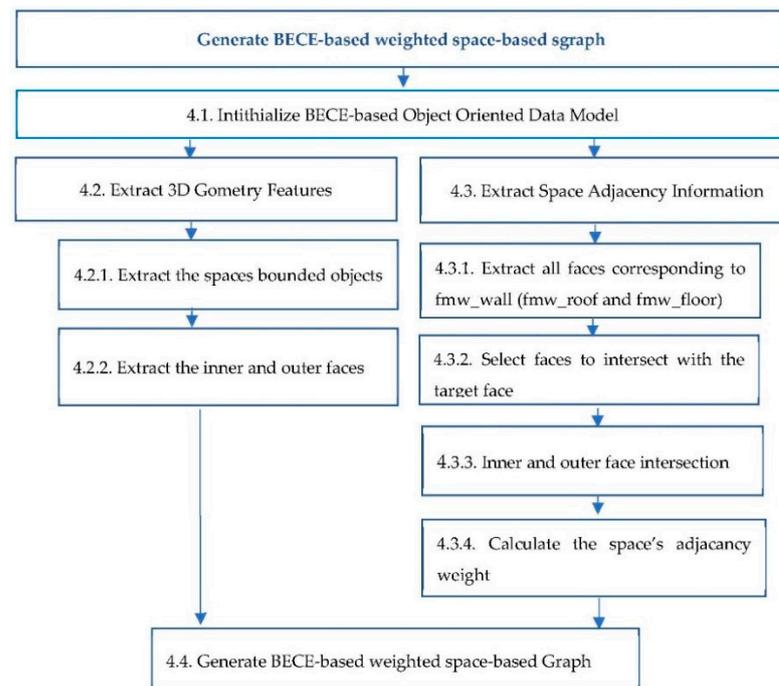
On the other hand, calculating the quantified value of space adjacency (weighted adjacency) is another parameter ignored in recent studies. Most researchers rely on binary space adjacency to find if two spaces are navigable by recognizing any opening object, such as a window or door between two spaces, as indoor navigation applications. However, in BECE analysis, the weight of this relationship is important because the amount of transferred energy from one room to another is directly related to the surface area of the shared wall, window, roof, floor, and material information. For instance, space A201 in Figure 3 is a cold room on this floor and significantly impacts the heating transfer to its adjacent spaces. In principle, the heating transformation from room A204 to room A201 (from a warm area to a cold area) is more than the heating transformation from A205 to A201 because of the differential in the shared wall area. Unfortunately, this information is not explicitly available in the IFC model. They have to be derived from the available entities using advanced geometric processing and material isolation information to obtain the desired weighted graphs. Hence, developing a framework to translate the IFC model into

a weighted space-based graph with accurate feature values compatible with BECE-based data-driven methods.

#### 4. Methodology

This research proposed a framework to generate a BECE-based Weighted space-based Graph (BECE-WG) to solve the compatibility issue between the IFC data model and graph-based machine learning algorithms. The graph represents a collection of interlinked IFC entities and their properties.

BECE-WG arranged the IFC dataset into connected graph structures, allowing for the integration and linking of data from several sources. In order to create a data structure connected to many datasets and appropriate for machine learning methods, this study uses the idea of converting 3D building data models (IFC) to a graph. The proposed graph contains space (room) object information as a node and their relationships with neighbor rooms as the edge. The BECE-based information of each space is extracted and calculated geometrically and semantically and assigned as a feature vector to each node in the graph. Different machine learning tasks such as supervised and unsupervised classification, node clustering, graph clustering, and node label prediction can employ the generated BECE-WG for building energy analysis. The framework has four main components: (1) BECE-based Object-Oriented data model, (2) Extract 3D Geometry Features, (3) Extract Space Adjacency Information, and (4) Generate BECE Weighted Space-based Graph. Each component includes some detailed steps, which are represented in Figure 4.



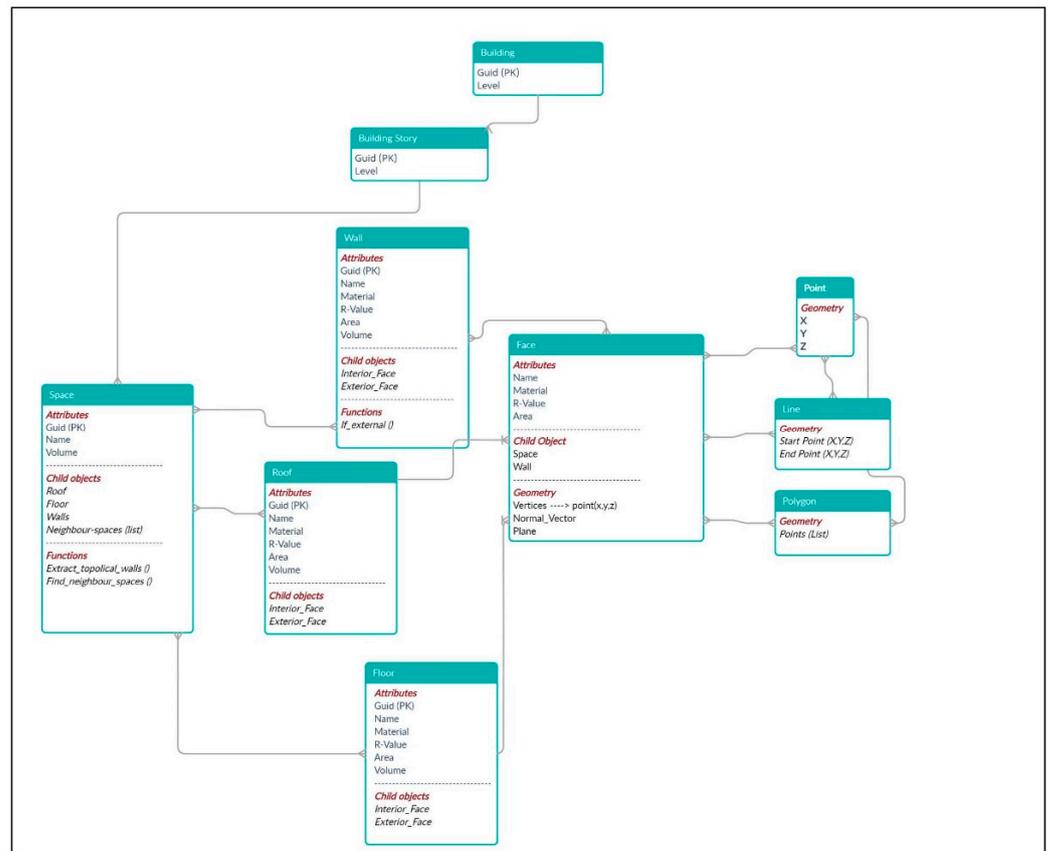
**Figure 4.** The workflow of generating a BECE-weighted space-based graph.

##### 4.1. BECE-Based Object-Oriented Data Model

An object-oriented data model is proposed by introducing BECE-based building objects with the superclass of building and story classes. Story class includes space objects and child objects such as the wall, window, floor, roof, and face objects representing three-dimensional objects. In this research, fmw is used as a prefix for the proposed framework object (fmw-object), and IFC (IFC-object) is employed as a prefix for standard IFC objects.

Figure 5 represents the proposed BECE-based data model, including the objects, properties, functionalities, and relationships. The proposed objects in the BECE framework include three main parts of properties, child objects, and functions. The property section encompasses the unique id, energy transfer properties such as material information, and

thermal Resistance Value (R-Value). It also includes accurate geometry information such as area and volume for wall, window, floor, and roof objects for each space. The Section 2 contains the child objects, which define the relationship between the framework objects. This section includes a single object or a list of objects. The relationship between the building entities can be defined by assigning the child object to the parent object, improving the query process, and extracting information from the proposed data model. For example, we proposed an *fmw-face* object as a child object of the *fmw-wall* to extract accurate geometry information on both sides (face of the wall) for each space. By adding the *fmw-space* object as a child object in the *fmw-face* object, we can find the corresponding space (grand-parent) of the specific face with a simple query process. However, this process leads to a complicated query in the IFC model to access the higher hierarchical objects (e.g., parent of parent objects).



**Figure 5.** Proposed object-oriented BECE framework from IFC schema.

The Section 6 of the proposed data model for objects is their functionality. The functions associated with each object calculate geometry properties (e.g., area, volume, etc.), semantic property values (e.g., the type of materials) called *Extract\_3DGeometry\_Features*, and identify the topological relationships between spaces, called *Find\_Adjacent\_spaces*. However, as discussed in Section 3.1, the geometry features cannot be extracted precisely by the IFC semantic-based query. Therefore, we proposed the *Extract\_3DGeometry\_Features* (Section 4.2) function for space objects to apply the face partitioning concept to extract accurate face geometry information of wall, window, roof, and floor objects in each space, as explained in Section 4.2 in detail.

The proposed data model tackles the space-adjacent challenges in the IFC model (Section 3.2) and improves the space-based relationship information between building objects in the IFC model. The relationships are defined based on the collection of child objects associated with each *fmw-object*. For example, an *fmw-space* object includes a collection of *fmw-wall*, and each *fmw-wall* object includes a collection of *fmw-face* as child objects.

Therefore, by initializing the *fmw-space* object and calling the *Find\_Adjacent\_Spaces* function, adjacent spaces for the target space are calculated and populate the *Neighbor-space* list of the *fmw-space* object. Therefore, we can apply a standard object-oriented query to access all wall, face, and adjacent spaces objects corresponding to each space.

The pseudocode code in Table 2 describes *fmw-space* object initialization and calculates the space properties and adjacent list by calling both *Extract\_3DGeometry\_Features* and *Find\_Adjacent\_spaces* functions. Section 4.3 discusses *Find\_Adjacent\_spaces* in detail.

**Table 2.** A pseudocode description of initializing the *fmw-space* object.

Pseudo Code
<b>Ifc_file = ifcopenshell.open('input IFC file')</b>
<b>Ifc-space = ifc_file['Guid'] — &gt; a single ifcSpace from IFC file</b>
<b>fmw-space = new fmw-Space(ifc-space) — &gt; create fmw-space object from proposed framework</b>
<b>fmw-space. Extract_3DGeometry_Features()</b>

#### 4.2. Extract 3D Geometry Features

The main goal of this section is to extract accurate geometry information such as Exterior Window–Wall area Ratio (EWWR), Exterior–Wall area ratio (EWR), Interior–Wall area ratio (INWR), Floor–Room area Ratio (FRR), and Roof–Room area Ratio (RRR) for building entities for each space. A closed volume characterizes a space in the IFC model of bounded walls, roof, and floor (space-bounding objects). Each bounded object includes a collection of faces in which a face is composed of a wire. The wire consists of edges and vertices. We used this granular information in IFC to create the *fmw-face* object for each space's walls, windows, roof, and floor to solve invalid space-based geometry issues, as mentioned in Section 3.1. The geometry granularity description of faces allows us to generate an *fmw-face* for each space, considered an inner face for the target space. The steps of creating face geometry consist of: (1) Extract the space's bounded objects (walls, windows, roof, and floor); (2) Extract the inner and outer faces of the bounded objects for each space; and (3) Define a new wall (*fmw-wall*) object and face (*fmw-face*) object for each space in the building. These steps extract the accurate geometry information for convex and non-convex spaces (non-convex hydron). Many previous studies focused on extracting the geometry information from the IFC. However, their solutions were limited to extracting the geometry information for convex spaces as a boundary box (bbox). In reality, the IFC model includes convex and non-convex polyhedrons. Therefore, we dealt with this challenge by proposing a solution to extract the accurate geometry of bounded objects for convex and non-convex polyhedron spaces.

##### 4.2.1. Extract the Space's Bounded Objects

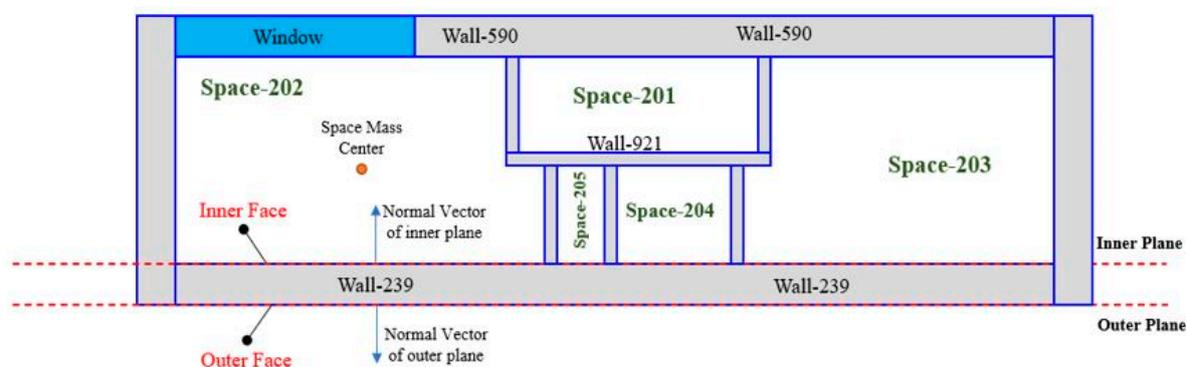
In this method, we expect the IFC model to accurately include the wall-required information, such as the classes *IfcSpace*, *IfcRelSpaceBoundary*, *IfcConnectionGeometry*, and *IfcWallStandardCase*. Each space's bounding walls can be listed using the *BoundedBy* function, which returns *ifcRelSpaceBoundary* elements, including the walls and *ifcCurtainWalls*. The *ifc\_wall\_list* includes all walls and curtain walls corresponding to *ifcSpace*. For instance, if we consider space-B202 as the target space, the query results in Table 3 are listed as two exterior walls: wall-534, wall-590, and four interior walls: wall-960, wall-921, wall-301, and wall-239.

**Table 3.** Extracting the space's bounded walls using the semantic query.

Pseudo Code
<pre> for rel_spaceboundary in ifcSpace.BoundedBy:     related_element = rel_spaceboundary.RelatedBuildingElement     if related_element.is_a('IfcWall') or is_a('IfcCurtainWall'):         ifc_wall_list.append(related_element) </pre>

#### 4.2.2. Extract the Inner and Outer Faces of the Bounded Objects

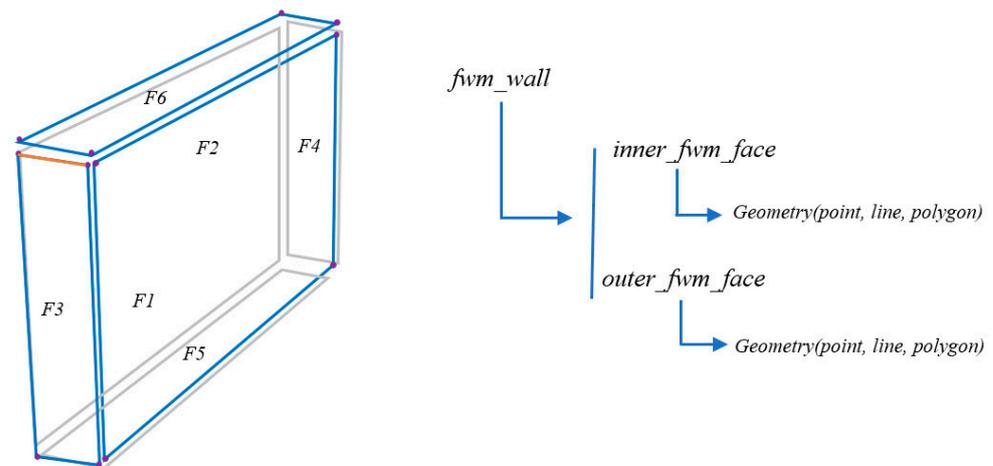
The reason for extracting the walls' inner face is to calculate the accurate geometry information (wall area for space). A shared wall between two spaces can have different areas based on the face geometry on each side. Figure 6 represents the 2D perspective of space-202 and the two inner and outer faces, planes, and normal vectors of each face for wall-239. The inner and outer face extraction is performed in two main steps.

**Figure 6.** 2D perspective of space-202 and inner and outer of Wall-239.

Step I: In this step, we need to determine which face of the wall towards the inner of the space becomes part of the space's bounding volume. The IFC model has several shapes: Solids, Shells, Faces, Wires, Edges, Vertices, and Compounds. We can iterate over the faces that bound the solid shape. Then, we can get a reference to the face's underlying surface. We can determine the plane for each face by knowing a point that belongs to the face ( $p_0 = (x_0, y_0, z_0)$ ) because the IFC model is composed of planar geometry. Therefore, we know that the face's surface conforms to a plane and a vector perpendicular ( $a, b, c$ ) to plane  $n$ . Equation (1) illustrates the plane question for all faces of the wall for space:

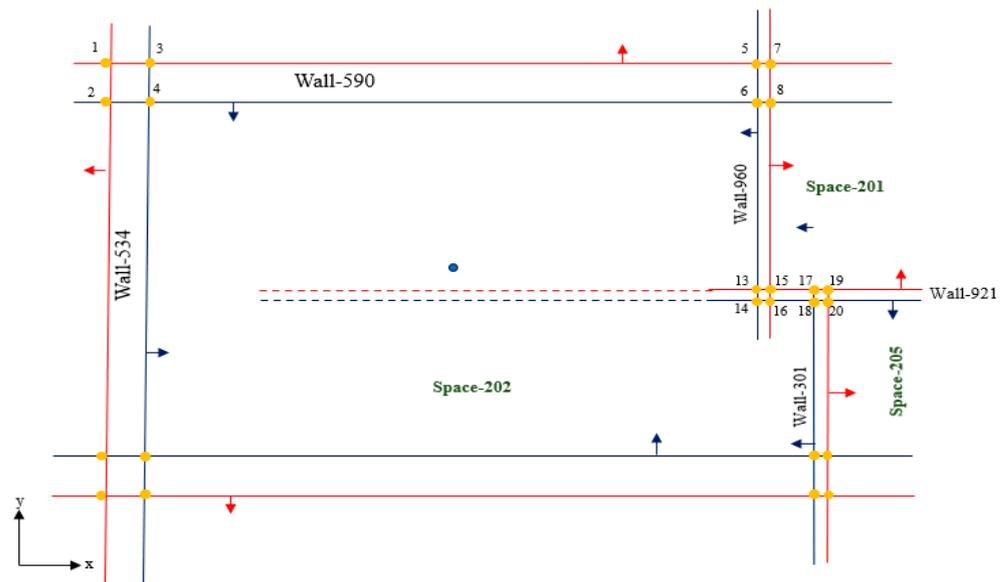
$$a(x - x_0) + b(y - y_0) + c(z - yz_0) = 0 \quad (1)$$

By iterating through `ifc_wall_list` (as presented in Section 4.2.1), all faces for the walls for a specific space are listed, and their planes are calculated by selecting a node on the surface and its normal vector (Equation (1)) and storing them as a child object of `frm_wall` object (`frm_face_list`). Figure 7 shows the semantic query to extract the bounding walls of `ifcSpace` and store them as a list of walls for a specific space (room). Figure 7a illustrates the face representation, and Figure 7b explains the framework's `fwm_wall` object's hierarchy. For example, the sample wall in Figure 7a illustrates six faces. To extract two prominent faces (inner and outer faces), the method iterates through the faces and calculates the area of each one.



**Figure 7.** (a). ifcWall representation with faces, faces’ lines, and points; (b). the proposed fwm\_wall and fwm\_face object-oriented data model.

The faces are sorted in ascending order based on their area value, and the two first faces are chosen as the main face objects. To find out the inner and outer face objects between these two faces, we applied the distance from plane method to calculate the distance of space’s center point (Figure 6, space-202) from the plane of these two faces. The face with the minimum distance from the space’s center point is considered the inner face, and the second is assigned as the outer face object. This method has a high accuracy rate in distinguishing between the inner and outer faces of convex polyhedron spaces. However, this method has incorrect results returned for some walls in non-convex polyhedron spaces. Figure 8 represents an example of this non-convex polyhedron space. The outer face of wall-921 is closer to the space’s center point than the inner face. Therefore, the algorithm assigns the outer face (red dashed line) as the inner face object, which is incorrect.



**Figure 8.** The 2D perspective of the face intersection of space-202. The arrows display the direction of the normal vector for each wall.

Consequently, we need additional conditions to increase the result accuracy and correct the wrong faces assignment. The next step describes this process. Additionally, after finding the first step of the inner and outer faces, we can explore which portion (polygon surface) of the plane belongs to the target space. To do that, all extracted planes of inner faces are intersected (Equations (2) and (3)).

Therefore, there is no intersection for those planes whose normal vector's dot product is zero, and the algorithm considers them parallel faces. By looping through the `frm_face_list`, all pairwise planes corresponding to each face are considered the vector equation of the intersection line. (Equation (2)) is calculated by the cross-product of the normal vectors of the given planes. To do that, a point  $(x_0, y_0, z_0)$  on the line of intersection is necessary. Therefore, the equations of the given planes as a system of linear equations (Equation (3)) are employed and set the  $z = \text{floor}$  as the  $z$ -value in this equation. Then, calculate the intersected line equation and store the intersected line in an array list (`line_arraylist`).

$$v = |\text{plane1} \times \text{plane 2}| = \begin{vmatrix} i & j & k \\ a_{p1} & b_{p1} & c_{p1} \\ a_{p2} & b_{p2} & c_{p2} \end{vmatrix} = (a, b, c) \quad (2)$$

$$L = \frac{x - x_0}{a} = \frac{y - y_0}{b} = \frac{z - z_0}{c} = t \quad (3)$$

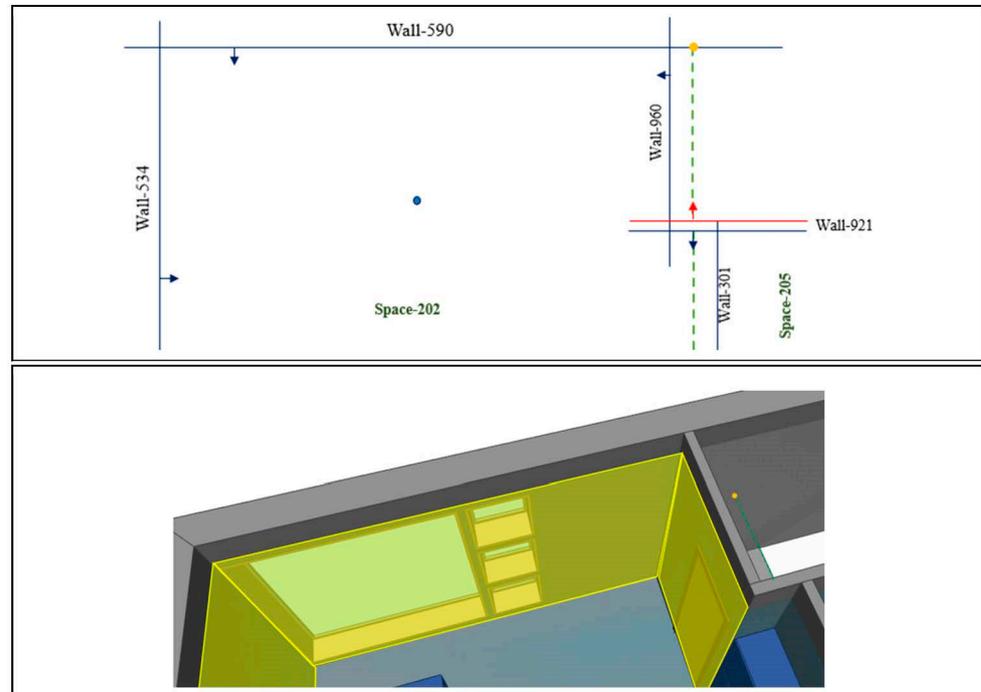
By the intersection of walls, windows, roof, and floor with the target face, four lines are extracted to delineate the polygon surface of each inner face. Each inner space's polygon geometry is determined using this result, and geometry information such as area, perimeter, and space boundary (wire) for each wall can be calculated. The intersection method helps solve the issue regarding inaccurate geometry information when there is a continuous multi-space shared wall in the IFC model, as previously mentioned in Section 3.2. The face intersection method can reach the accurate geometry of faces and obtain the accurate geometry of each wall, roof, and floor for the target space.

Step II: At this step, post-processing of inner and outer face extraction is applied to rectify the possible error in recognizing walls' inner and outer faces. Therefore, we use the advantage of wall partitioning (`fmw_face` object) results to investigate the potential error for some walls in a non-convex polyhedron room. We apply the ray-tracing intersection method for all recognized inner faces in the first step to solve mentioned issue. By iterating through to all recognized inner faces from step 1, we can make a ray-crossing line for each face using the face center point and its plane's normal vector. Then, intersect the created ray-cross (line) with all inner faces and find the intersection point  $(x, y, z)$ . If the intersected point is at least inside one of the inner faces' polygon, the target face is an inner face. It is considered an outer face if it does not meet this condition. Figure 9 illustrates a sample of applying the ray-crossing algorithm to rectify the extracted inner and outer face for wall-921.

#### 4.3. Extract Space Adjacency Information

The adjacent space analysis is a critical spatial analysis in BECE to measure the effect of the neighbor spaces on each room's energy transfer. Therefore, in the BECE application, it is necessary to investigate the adjacent spaces affected by different adjacent degrees when a room has high energy consumption. However, recent research does not explore the algorithm to extract the accurate space adjacent and the quantified weight of adjacency from the IFC model [29,30]. Section 3.2 discusses the problem of the semantic query from the original IFC model in extracting space-adjacent information. It implies that the current methods of extracting space adjacents are inaccurate because they assume that if two walls corresponding to two spaces have a shared edge in the floor plan, they are adjacent. The `IfcSpace` can obtain information about the nearby walls, but since numerous `IfcSpace` objects can share a wall, it is not realistic to assume that the `IfcSpace`'s adjacency relations can be obtained by using the same wall. In this research, we develop our space's adjacent extraction method on the inter-relationships among `IfcSpaces` to extract the quantified space adjacent extraction and use this method as a node relationship information (edge) to generate the final graph. As we discussed in Section 4.1, the object in the proposed framework includes some functions to extract the information not available in the IFC model, such as finding adjacent spaces for each space. The proposed method encompasses

three steps. First, we extract all faces related to each wall in a space using a query from framework wall objects, not the IFC objects. Second, select the faces whose normal vector of planes has the opposite direction from the target face. Finally, the polygon intersection method detects if two faces are intersected.



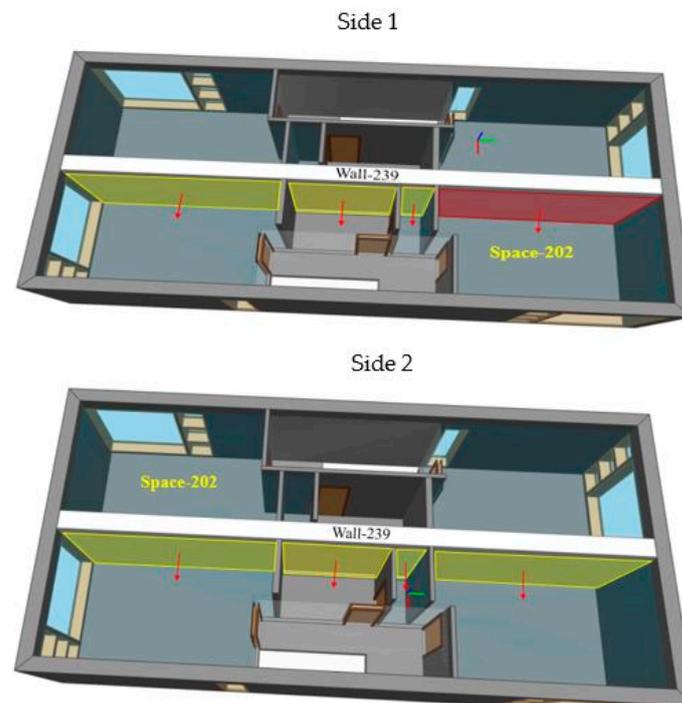
**Figure 9.** 2D and 3D perspectives of ray-crossing lines to find inner and outer faces. The arrows display the direction of the normal vector for each wall. The intersection of the red-face ray-crossing of wall-921 with the inner face of wall-590 generates a 3D point outside of all highlighted inner faces of space-202. Therefore the red face, considered the inner face in step one, is assigned as the outer face because it does not meet the condition. Moreover, the blue face of wall-921 is assigned to the inner face object for space-202. Finally, applying these three steps to all IFC model spaces, WWR, EWR, INW, FRR, and RRR are calculated accurately using the rich geometry computation in the proposed framework and storing these calculation results as the space objects' attributes.

#### 4.3.1. Extract All Faces Corresponding to fwm\_wall

The proposed framework wall object (`fwm_wall`) lists `fwm_face` child objects. First, the face objects in this list are calculated and populated based on the proposed method in Section 4.2. Next, we define and create the face object by extracting the `fwm_face` object and assigning it as a child object for each wall in the IFC model. This object is not defined explicitly in the IFC model, but the proposed framework makes it available for BECE analysis. Figure 7a describes the query syntax of extracting all faces (both sides) for wall-239. Line 1 to 3 of the syntax code shows how to apply a simple query to access eight `fwm_face` objects for wall-239.

#### 4.3.2. Select the Faces to Intersect with the Target Face

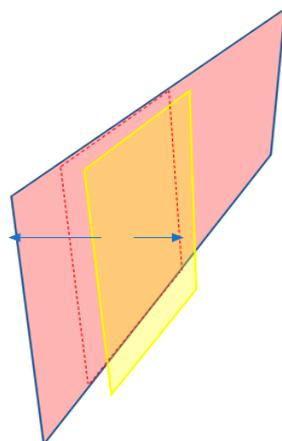
By accessing all faces and their polygon geometry, the algorithm filters the faces to detect any intersection between the `target_face` (red face in Figure 10—Side 1). Therefore, the dot product of the plane's normal vector of the target face and the other faces is calculated, and those with a dot product value of more than zero are selected for the intersection test (yellow faces in Figure 10—Side 2).



**Figure 10.** Find the adjacent spaces for space-202.

#### 4.4. Inner and Outer Face Intersection

This step applies a pairwise intersection between the outer faces extracted from the previous section and the target face. The research assumes all faces are a convex 2D polygon; accordingly, we apply the 2D polygon intersection method to detect if two faces are intersecting. However, we need two polygons on the same plane to apply the intersection method. Therefore, we project the outer face polygon on the target face's plane, then apply an intersection between the two polygons. Figure 11 represents the projected polygon schema (dashed red line) of the outer face of the target plane. The projected polygon intersects with the target face's polygon to find the area of intersection (Table 4, lines 8, 9, and 10). Table 5 explains the `calculate_polygon_intersection_area` method, which uses the `shapely` open-source library to calculate two polygons' intersected areas. Consequently, just one face from wall-239 intersects with the target face, and the parent object of that face (`fmw_space`) is recognized as one of the adjacent spaces. Using the proposed method in Section 4.3, we can extract both the 3D adjacent spaces for each space and the area value of adjacency. Finally, this method is applied on the floor and roof to find the adjacent spaces on different floor levels with calculated the shared area value.



**Figure 11.** 3D schema of projection of face's polygon onto the target plane.

**Table 4.** Pseudo code for select the faces to intersect with the target face.

Pseudo Code	
1-	<code>fmw_wall_236 = framework.WALL('wall-239')</code>
2-	<code>targer_face = framework.SPACE('space-202')</code>
3-	<code>for fmw_face in fmw_wall_236.fmw_face_list:</code>
4-	<code>    if fmw_face != targer_face:</code>
5-	<code>        If dot(target_face.plane.normal_v fmw_face.plane.normal_v) &lt;&gt; 0:</code>
6-	<code>            projected_polygon =</code>
7-	<code>                framework.project_polygon(fmw_face.polygon, targer_face.plane)</code>
8-	<code>                int_area = framework.calculate_polygon_intersection_area</code>
	<code>                (projected_polygon, targer_face.polygon)</code>
9-	<code>            If int_area &gt; 0:</code>
10-	<code>                target_space.adjacent_fmw_spaces.append</code>

**Table 5.** Extracting the space's bounded walls using semantic query.

Pseudo Code	
	<code>for rel_spaceboundary in ifcSpace.BoundedBy:</code>
	<code>from shapely.geometry import</code>
	<code>p = Polygon([(x<sub>p1</sub>, y<sub>p1</sub>), (x<sub>p2</sub>, y<sub>p2</sub>), (x<sub>p3</sub>, y<sub>p3</sub>), (x<sub>p4</sub>, y<sub>p4</sub>)])</code>
	<code>q = Polygon([(x<sub>q1</sub>, y<sub>q1</sub>), (x<sub>q2</sub>, y<sub>q2</sub>), (x<sub>q3</sub>, y<sub>q3</sub>), (x<sub>q4</sub>, y<sub>q4</sub>)])</code>
	<code>if p.intersects(q):</code>
	<code>    int_area =p.intersects(q).area</code>

#### Calculate the Spaces' Adjacency Weight

The edges in our proposed graph are representative of the adjacent space relationship. The BECE-based graph is a weighted graph in which weights reflect the relative importance of the rooms' relationships. In other words, some links between nodes have lower weights (low importance), and others have higher weights (high importance). For example, suppose two rooms have more energy transfer in our application. In that case, the relationship between those rooms is more important than two other rooms with a low rate of energy transfer. The energy is transferred between two rooms through the walls, roofs, floors, and windows. Each of these building elements comprises several layers with different materials. The energy transfer of each building element is determined by the thermal resistance value (R-Value). In fact, the R-value assesses how well a two-dimensional barrier, such as an insulation layer, a window, a wall, or a ceiling, resists heat conduction [31]. Therefore, we introduced a wight for the room relationship by calculating the barrier's R-Value and the shared area value. Equation (4) describes the proposed weight value of edges in a BECB space-based graph. The  $Weight_{n1,n2}$  shows the importance value of the two room's relationship. Share area is the common area value of rooms  $n1$  and  $n2$ . The  $\sum_{l=1}^m R - Value_l$  is the sum of the R-Value for  $n$  number of barriers between two rooms.

$$Weight_{n1,n2} = \frac{Shared\ area}{\sum_{l=1}^n R - Value_l} \quad (4)$$

#### 4.5. Generate BECE-Based Weighted Space-Based Graph

After creating `fmw_face` objects and extracting the BECE-based attributes and relationship information, a new method is developed to transfer the `fmw-face` objects to a weighted BECE space-based graph. This method constructs the adjacency matrix following and the weighted space-based graph based on the `fmw-face` object list from Sections 4.2 and 4.3. It includes three main steps. First, the Normalization Function (NF) is used to normalize the feature values. The normalizing must produce common scale feature values for machine learning algorithms since each parameter has a different scale. Therefore, we applied the min-max normalization method to normalize the input feature values. The second step

generates the adjacency matrix using the list of adjacent objects (*fmw\_space*). Indeed, the adjacency matrix represents the graph as a square matrix with the size of  $n * n$  ( $n$ : number of nodes). In the last step of this process, the function is designed to iterate through the spaces to find the space's adjacency and assigned weight value. In this step, the space-based graph was created utilizing the adjacency matrix and six feature values embedded as each node's properties using the Python NetworkX module [32]. The generated graph is homogeneous because all nodes have equivalent types (room) and similar edges to the room's neighborhood. Therefore, the BECE space-based graph is a compatible data model for future space-based energy consumption analysis using a graph-based machine learning algorithm (Graph-ML).

#### 4.6. Measuring Accuracy for Validation

The validity of the proposed algorithm in extracting geometry information is verified using the Root Mean Error (RMSE), a standard way to measure a model's error (or level of accuracy) to evaluate the model's estimated values for quantitative data [33,34]. Indeed, the geometry (area) of walls, windows, roofs, and floors for all spaces in the two test cases are extracted by BIMVision software which is considered a ground truth value. Then, the RMSE value is calculated using Equation (5) [35] for the extracted geometry information of building elements for estimated values from both the IFC model and framework objects.  $RMSE_e$  shows the accuracy of the estimated area value for the building elements (Wall, Window, Roof, and Floor). The  $a_i$  is the calculated area of elements from models, and  $a_g$  represents the area value based on the ground truth measurement.

$$RMSE = \text{SQRT} \left( \sum_{i=1}^N \frac{(a_i - a_g)^2}{N} \right) \quad (5)$$

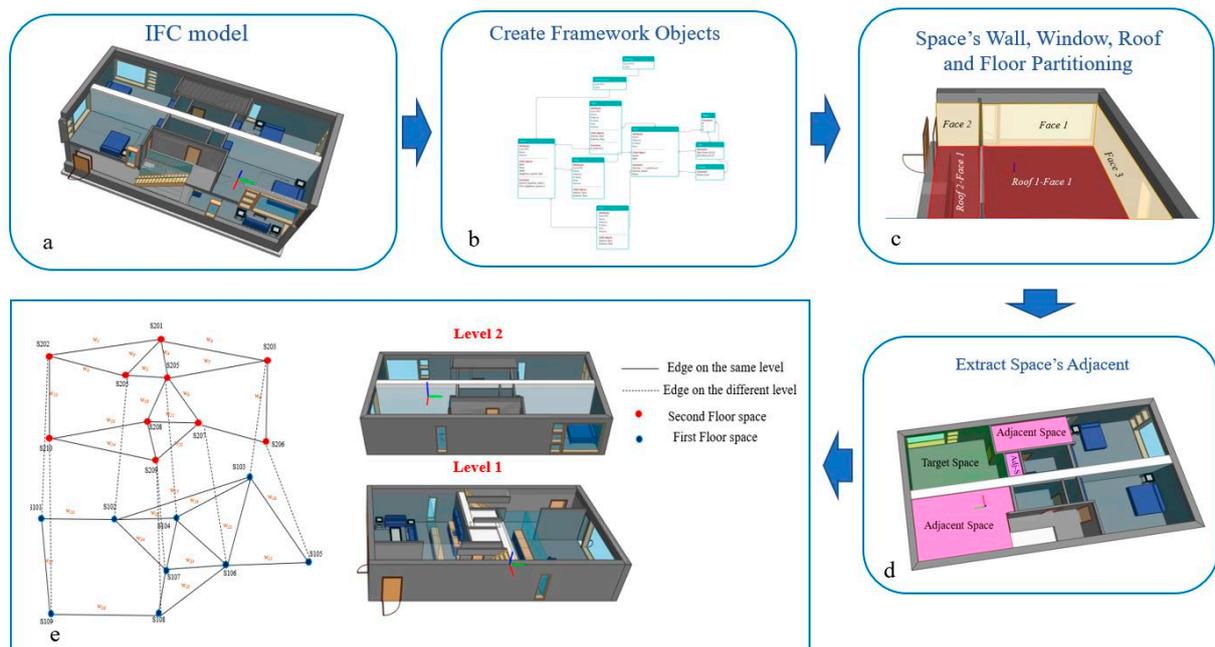
## 5. Experiments and Results

In order to test and validate our approach, we apply the proposed methodology to two test case studies and two multi-level IFC models of ERDC "<http://openifcmodel.cs.auckland.ac.nz>" (accessed on 14 June 2020): the Duplex Apartment Model and the Autodesk Trapleo building. These test cases are chosen because multi-level building information is available in the models. Additionally, the *ifcSpace* objects are defined in these two IFC models. We hypothesized that the framework's extracted semantic and space relationship information are identical due to the manual measurement of geometry information and visual space relationship extraction.

The entire BECE framework discussed in the previous sections has been implemented by the Python language, which several open-source libraries support, such as *IfcOpenShell* and *NetworkX* [31]. Finally, for visual representation and ground truth measurements, we notably used *Revit* and *BIMVision* software while supporting at the same time the geometry, the topology, and the semantic information of the original IFC models.

### 5.1. Case Study I: ERDC-Duplex Model

The first IFC model as a test case is a two-story building with eighteen rooms on two levels. This model includes the convex and non-convex spaces, and all the walls, roofs, and floors are perpendicular. Additionally, eight continuous exterior and two continuous interior walls are available in this model, which describes the first mentioned issue in this research. The multi-room shared wall issue in this model opens this opportunity to test our methodology from the perspective of accurate geometry extraction and recognizing the correct space adjacent. The proposed BECE framework generates an accurate graph from the IFC model for test case 1. Figure 12 represents the steps of this process for the IFC model.



**Figure 12.** The steps of generating a BECE space-based graph from the IFC model in test case 1.

Figure 12a represents the IFC model of this test case. Using the proposed method, in the first step (Figure 12b), the framework objects (parent objects) such as `fmw_wall`, `fwm_roof`, `fmw_floor`, and `fmw_window` are instantiated for each space. Next, the face extraction method is applied to extract both faces of the wall, window, roof, and floor objects and store them as child objects for each space parent object (Figure 12c). Finally, the Extract Space Adjacency method is employed to find the adjacents and calculate the weight value for all spaces (room) in two building levels (Figure 12d,e).

Figure 13 illustrates geometry and adjacent information for space-202 extracted from the IFC model (a and b) and proposed framework (c and d). The accuracy of extracted information is evaluated by the ground truth values measured using BIMVision software. Comparing these two tables' values declares the wrong area value for continued walls of 239, 534, and 590 when geometry information is extracted from the IFC model. However, the framework has accurate extracted information.

### 5.2. Case Study II: Trapelo Building

As a test case, the second IFC model is a commercial three-story building with 130 rooms (Figure 14a). This model includes the convex and non-convex spaces, but several walls have a non-perpendicular intersection (Figure 14b), unlike the case test. More than 40 continuous exterior walls and 25 shared windows are available in this model. This test case is a more complex model than the first one because of the number of levels, geometry, and space relationships. Therefore, it could be a suitable model to evaluate the proposed methodology to extract accurate geometry and semantics. The four steps of generating a framework data model, extracting BECE-based geometry and semantic information, extracting space's adjacents information, and generating the BECE-based weighted space-based graph are applied to this test case. The proposed methods for these two test cases are evaluated by validation parameters introduced in the following section.

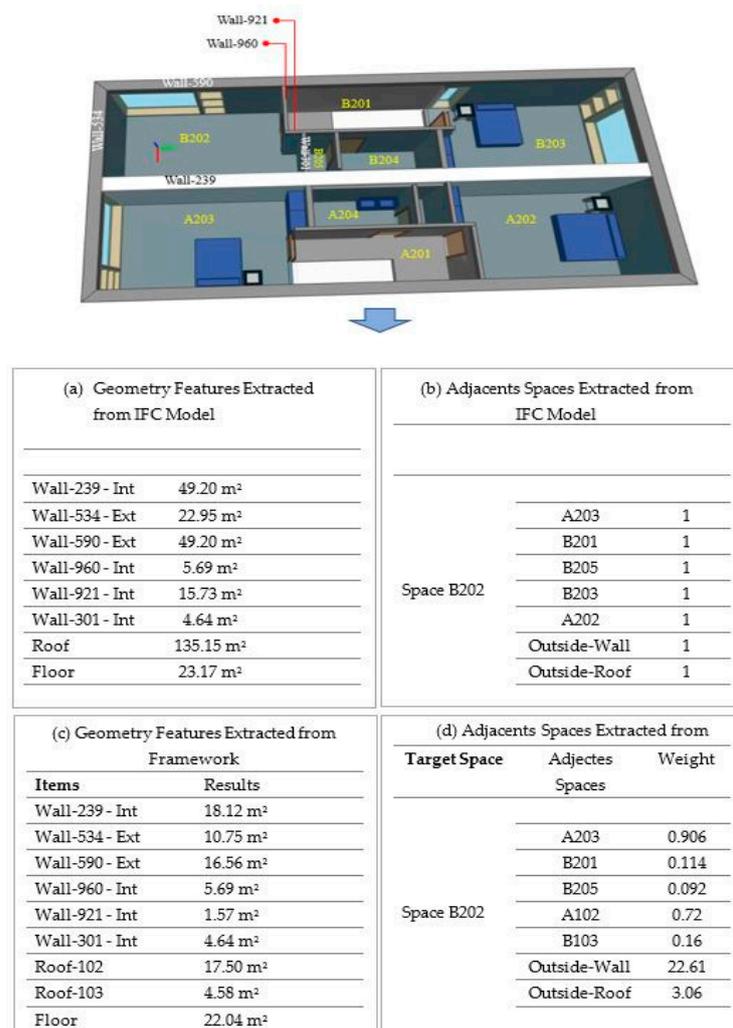


Figure 13. The extracted information from the IFC model and framework.

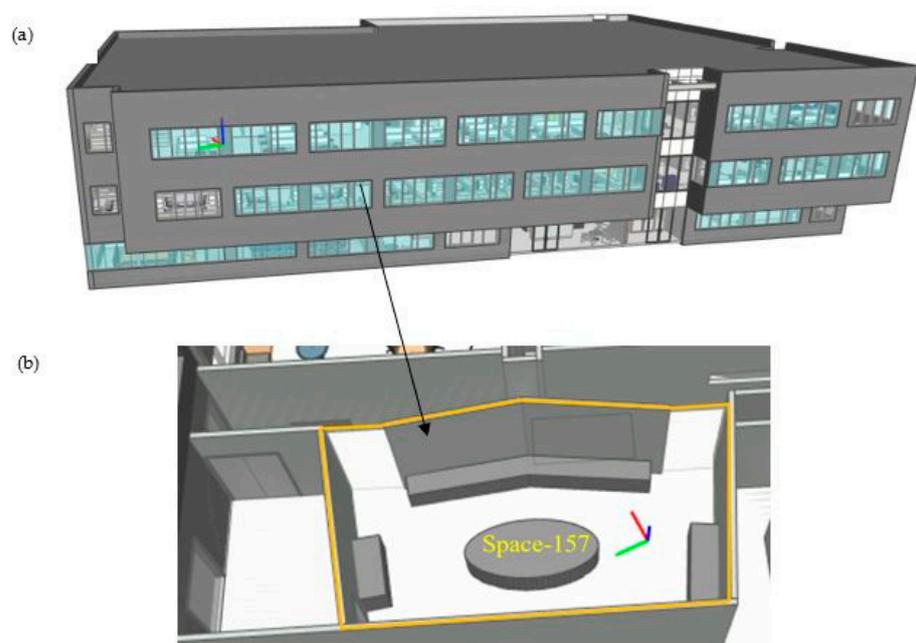


Figure 14. (a) The 3D IFC Model of Test Case; (b) the TheNon-Perpendicular Room (Space-157).

### 5.3. Experimental Result and Discussion

Tables 6 and 7 represent the root mean square error of geometry information extracted from the IFC model and calculated by the Test Case I and II frameworks. A comparative analysis of the errors between the geometry estimated values from the IFC model and the BECE-framework model for two test cases. Table 4—test case I and Table 5—test case II) demonstrate the high accuracy in calculating space's geometry information using the proposed methodology. Additionally, the error in the area value extracted from the IFC model is incremented by increasing the number of rooms. For example, the error of extracted wall geometry information from the IFC model is increased between case study I and case study II because of the higher number of non-perpendicular rooms in case study II. Additionally, in test case II, the number of continuous walls (shared between multiple rooms) and windows is higher than in test case I, which causes a decrease in the accuracy of geometry information. Thus, it demonstrates that BECE analysis cannot account for space's geometry information in the IFC model because the inaccurate extracted information is causing the wrong BECE data-driven analysis.

**Table 6.** The error of geometry information extracted from the IFC model of test case I with 18 rooms.

Validation Parameters	RMSE of IFC Model (m <sup>2</sup> )	RMSE of BECE-Framework (m <sup>2</sup> )
Wall area estimation	2.1	0.10
Roof area estimation	1.9	0.09
Window area estimation	1.78	0.058
Floor area estimation	1.89	0.026

**Table 7.** The error of geometry information extracted from the IFC model of test case 2 with 130 rooms.

Validation Parameters	RMSE of IFC Model (m <sup>2</sup> )	RMSE of BECE-Framework (m <sup>2</sup> )
Wall area estimation	3.9	0.11
Roof area estimation	1.4	0.12
Window area estimation	3.1	0.08
Floor area estimation	3.2	0.03

Another aspect being considered is the high possibility of the failure of the space relationship (adjacent information). This is primarily due to non-topological information about the IFC model's wall, window, roof, and floor elements. Indeed, the IFC model's incorrect information of space adjacent leads to a wrong BECE space-based graph. Table 8 describes the percentage of the total numbers of the correct detected adjacent rooms to the total number of rooms for each floor. The low percentage of adjacent information from the IFC model illustrates the importance of recognizing and extracting the valid adjacent information in the IFC model. The proposed methodology for extracting adjacent information shows high accuracy for vertical and horizontal adjacent information. However, although the framework has a high performance in finding the adjacent spaces, it could not extract 100% adjacent spaces for case study 1 because there is more than one wall (two walls—IfcWallStandardCase and PartyWall) between two spaces. Therefore, this is a limitation of the proposed methodology. Another limitation to be considered is that the primary assumption of this research is that the ifcSpace object is defined in the original IFC model. However, the methodology failed for those IFC models without the ifcSpace object. Additionally, the proposed framework does not support spaces with complex geometry, such as star shape geometry or complex architectural roofing spaces. Therefore, this part of the solution remains an unsolved challenge for future attempts.

**Table 8.** The percentage of correct extracted adjacent spaces for two test cases.

Name	Query from IFC(%)	BECE-Framework (%)
Test Case 1	70	98
Test Case 2 (First Level)	65	100
Test Case 2 (Second Level)	60	100
Test Case 2 (Third Level)	60	100
Test Case 2 (All Levels)	70	100

## 6. Conclusions and Future Work

This paper introduced a BECE-based framework to fill the gap of using data-driven models, such as machine learning algorithms, for building energy consumption estimation (BECE) using the IFC model. There are three main contributions in this paper which are the advantage of the proposed methodology compared to existing methods. First, this research proposed a framework to generate intermediate object-oriented data models to facilitate the object-oriented-based semantic and geometry query to reduce the complexity of the query from the IFC model. Second, a BECE-based weighted space-based graph from the IFC model is developed to solve the compatibility issue between the IFC model and data-driven algorithms for BECE analysis. The third one is regarding employing topology information in BECE analysis. The proposed methodology accurately extracts adjacent information (3D topology) for vertical and horizontal weighted space-based adjacent information to increase the accuracy of BECE data-driven models. In contrast, the existing approaches rely on the semantic and geometry information in the IFC model, which causes several erroneous analysis results. Additionally, topology information is ignored in the current research in BECE analysis. The proposed methodology defined classes and objects in the new data model comprising the explicit semantic, geometry, and space adjacents information that is not available in the native IFC models. In addition, some functionalities are also designed and implemented to apply geometry computation to extract accurate semantics, geometry, and weighted relationship for *ifcSpaces* in the IFC mode.

Moreover, the proposed methodology is not limited to extracting the information for spaces with convex polyhedron geometry; it is designed to apply geometry computation for convex and non-convex polyhedron spaces. To this end, an accurate BECE-based weighted graph is generated. However, in the current research, the spatial queries can be applied to convex-polyhedron spaces, and there are erroneous results for non-convex polyhedron spaces. Thus, the proposed graph encompasses accurate semantic, geometry, and space-adjacent information. Additionally, the generated space-based graph does not have IFC model complexity and is compatible with data-driven algorithms.

Nevertheless, the success of the process depends on how well the input IFC model's space definitions are defined. If the input spaces are not adequately defined in the model, stability and topology validity cannot be guaranteed. Another aspect that needs to be considered in future work is the adjacent information between the spaces, which can be calculated more accurately with more than one wall, roof, or floor. Our proposed methodology assumes a single object between the spaces and calculates the weight of a space's relationship based on area and material. However, additional attempts must be considered and investigated to find the objects' number and material information between the two spaces. Consequently, the proposed data model has more accurate geometry and semantic information for spaces, more realistic objects close to real-world entities, and a more readable structure for different applications.

**Author Contributions:** Conceptualization, H.K., M.J., A.R. and G.S.; Methodology, H.K., M.J., A.R. and G.S.; Validation, H.K., M.J., A.R. and G.S.; Formal analysis, H.K. and M.J.; Resources, M.J.; Data curation, H.K.; Writing—original draft, H.K. and M.J.; Writing—review & editing, M.J., A.R. and G.S.; Visualization, H.K.; Supervision, M.J., A.R. and G.S.; Funding acquisition, M.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC DG 485794) and York University.

**Data Availability Statement:** “<http://openifcmodel.cs.auckland.ac.nz>” (accessed on 14 June 2020).

**Acknowledgments:** The authors would like to thank Autodesk for providing open-source BIM data for this study. We would also like to thank the three reviewers for their feedback, which has helped improve the quality of the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Gao, H.; Koch, C.; Wu, Y. Building information modelling based building energy modelling: A review. *Appl. Energy* **2019**, *238*, 320–343. [CrossRef]
2. Borrmann, A.; Schraufstetter, S.; Rank, E. Implementing Metric Operators of a Spatial Query Language for 3D Building Models: Octree and B-Rep Approaches. *J. Comput. Civ. Eng.* **2009**, *23*, 34–46. [CrossRef]
3. BuildingSMART, I. BuildingSMART International. 2015. Available online: <https://www.buildingsmart.org/> (accessed on 19 October 2022).
4. Liu, Y.; van Nederveen, S.; Hertogh, M. Understanding effects of BIM on collaborative design and construction. An empirical study in China. *Int. J. Proj. Manag.* **2017**, *35*, 686–698. [CrossRef]
5. Chong, H.Y.; Lee, C.Y.; Wang, X. A mixed review of the adoption of Building Information Modelling (BIM) for sustainability. *J. Clean. Prod.* **2017**, *142*, 4114–4126. [CrossRef]
6. Diakit , A.A.; Zlatanova, S. Valid Space Description in BIM for 3D Indoor Navigation. *Int. J. 3-D Inf. Model.* **2016**, *5*, 1–17. [CrossRef]
7. Andriamamonjy, A.; Saelens, D.; Klein, R. An automated IFC-based workflow for building energy performance simulation with Modelica. *Autom. Constr.* **2018**, *91*, 166–181. [CrossRef]
8. P rez-Lombard, L.; Ortiz, J.; Pout, C. A review on buildings energy consumption information. *Energy Build.* **2008**, *40*, 394–398. [CrossRef]
9. Wei, Y.; Zhang, X.; Shi, Y.; Xia, L.; Pan, S.; Wu, J.; Han, M.; Zhao, X. A review of data-driven approaches for prediction and classification of building energy consumption. *Renew. Sustain. Energy Rev.* **2018**, *82*, 1027–1047. [CrossRef]
10. Fumo, N. A review on the basics of building energy estimation. *Renew. Sustain. Energy Rev.* **2014**, *31*, 53–60. [CrossRef]
11. Nouvel, R.; Zirak, M.; Dagtageri, H.; Coors, V.; Eicker, U. Urban energy analysis based on 3d city model for national scale applications. In Proceedings of the 5th German-Austrian IBPSA Conference, Aachen, Germany, 22–24 September 2014.
12. Sun, Y.; Haghighat, F.; Fung, B.C. A review of the-state-of-the-art in data-driven approaches for building energy prediction. *Energy Build.* **2020**, *221*, 110022. [CrossRef]
13. Yumusak, S.; Dogdu, E.; Kodaz, H.; Kamilaris, A.; Vandenbussche, P.-Y. SpEnD: Linked Data SPARQL Endpoints Discovery Using Search Engines. *IEICE Trans. Inf. Syst.* **2017**, *E100.D*, 758–767. [CrossRef]
14. Jin, C.; Xu, M.; Lin, L.; Zhou, X. Exploring BIM Data by Graph-based Unsupervised Learning. In Proceedings of the ICPRAM 2018—The 7th International Conference on Pattern Recognition Applications and Methods, Funchal, Portugal, 16–18 January 2018.
15. Bourdeau, M.; Zhai, X.Q.; Nefzaoui, E.; Guo, X.; Chatellier, P. Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustain. Cities Soc.* **2019**, *48*, 101533. [CrossRef]
16. Liebich, T. IFC4—The new buildingSMART Standard. In *BuildingSMART*; bSI Publications: Helsinki, Finland, 2013.
17. Ma, J.; Cheng, J.C. Estimation of the building energy use intensity in the urban scale by integrating GIS and big data technology. *Appl. Energy* **2016**, *183*, 182–192. [CrossRef]
18. Hitchcock, R.J.; Wong, J. Transforming IFC architectural view BIMs for energy simulation: 2011. In Proceedings of the Building Simulation 2011: 12th Conference of International Building Performance Simulation Association, Sydney, Australia, 14–16 November 2011.
19. Remmen, P.; Cao, J.; Sebastian, E.; Frisch, J.; Lauster, M.; Maile, T.; O’Donnell, J.; Sergio, P.; R dler, J.; Streblow, R.; et al. An Open Framework for Integrated Bim-Based Building Performance Simulation using Modelica. In Proceedings of the 14th International Conference of IBPSA—Building Simulation 2015, BS 2015, Conference Proceedings, Hyderabad, India, 7–9 December 2015.
20. Dong, B.; Lam, K.P.; Huang, Y.C.; Dobbs, G.M. A comparative study of the IFC and gbXML informational infrastructures for data exchange in computational design support environments. In Proceedings of the IBPSA 2007—International Building Performance Simulation Association 2007, Beijing, China, 3–6 September 2007.
21. Arayici, Y.; Fernando, T.; Munoz, V.; Bassanino, M. Interoperability specification development for integrated BIM use in performance based design. *Autom. Constr.* **2018**, *85*, 167–181. [CrossRef]
22. Jeong, W.; Kim, J.B.; Clayton, M.J.; Haberl, J.S.; Yan, W. Translating Building Information Modeling to Building Energy Modeling Using Model View Definition. *Sci. World J.* **2014**, *2014*, 1–21. [CrossRef]
23. Cemesova, A.; Hopfe, C.J.; Mcleod, R.S. PassivBIM: Enhancing interoperability between BIM and low energy design software. *Autom. Constr.* **2015**, *57*, 17–32. [CrossRef]

24. Bazjanac, V.; Maile, T.; Nytsch-Geusen, C. Generation of building geometry for energy performance simulation using Modelica. In Proceedings of the CESBP/BauSim 2016, Dresden, Germany, 14–16 September 2016.
25. Bazjanac, V.; Berkeley, L. Space boundary requirements for modeling of building geometry for energy and other performance simulation. In Proceedings of the CIB W78 2010: 27th International Conference, Cairo, Egypt, 16–19 November 2010.
26. Gunduz, M.; Tang, S.; Shelden, D.R.; Eastman, E.M.; Pishdad-Bozorgi, P.; Gao, X. A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. *Autom. Constr.* **2019**, *95*, 127–139.
27. Fernández, I.; Borges, C.E.; Penya, Y.K. Efficient building load forecasting. In Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Toulouse, France, 5–9 September 2011.
28. Lilis, G.N.; Giannakis, G.I.; Rovas, D.V. Automatic generation of second-level space boundary topology from IFC geometry inputs. *Autom. Constr.* **2017**, *76*, 108–124. [\[CrossRef\]](#)
29. Kim, S.; Zadeh, P.A.; Staub-French, S.; Froese, T.; Cavka, B.T. Assessment of the Impact of Window Size, Position and Orientation on Building Energy Load Using BIM. *Procedia Eng.* **2016**, *145*, 1424–1431. [\[CrossRef\]](#)
30. Lee, J.; Kwan, M.-P. A combinatorial data model for representing topological relations among 3D geographical features in micro-spatial environments. *Int. J. Geogr. Inf. Sci.* **2005**, *19*, 1039–1056. [\[CrossRef\]](#)
31. Yu, Z.; Fung, B.C.M.; Haghghat, F. Extracting knowledge from building-related data—A data mining framework. *Build. Simul.* **2013**, *6*, 207–222. [\[CrossRef\]](#)
32. Boeing, G. OSMnx: A Python package to work with graph-theoretic OpenStreetMap street networks. *J. Open Source Softw.* **2017**, *2*, 215. [\[CrossRef\]](#)
33. Tashakkori, H.; Rajabifard, A.; Kalantari, M. A new 3D indoor/outdoor spatial model for indoor emergency response facilitation. *Build. Environ.* **2015**, *89*, 170–182. [\[CrossRef\]](#)
34. Roy, S.; Bhunia, G.S.; Shit, P.K. Spatial prediction of COVID-19 epidemic using ARIMA techniques in India. *Model. Earth Syst. Environ.* **2020**, *2019*, 0123456789. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Scellato. *NetworkX: Network Analysis with Python Today's Outline*, no. February; University of Cambridge: Cambridge, UK, 2013; pp. 1–51.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.