

Соловей Ольга Леонідівна

Кандидатка технічних наук, доцентка, докторантка кафедри інформаційних технологій,

<https://orcid.org/0000-0001-8774-7243>

Київський національний університет будівництва і архітектури, Київ

Гончаренко Тетяна Андріївна

Докторка технічних наук, завідувачка кафедри інформаційних технологій,

<https://orcid.org/0000-0003-2577-6916>

Київський національний університет будівництва і архітектури, Київ

МЕТОД СТВОРЕННЯ ГРАФІВ НА ОСНОВІ BIM-МОДЕЛЕЙ ДЛЯ ПРЕДИКТИВНОГО МОНІТОРИНГУ БУДІВЕЛЬ

Анотація. Розроблено та верифіковано методологію автоматизованого перетворення BIM-моделей у фізично обґрунтовані графові представлення, придатні для подальшого застосування в задачах прогнозування довговічності будівель. Вирішено такі завдання: розробка методу перетворення BIM-даних у графову структуру, де вершини представляють фізичні елементи, а ребра – їхні конструктивні та просторові взаємозв'язки; створення системи критеріїв для оцінки коректності роботи методу, що включає кількісну та якісну перевірку; експериментальна апробація розробленого методу на наборі еталонних BIM-моделей різного масштабу для підтвердження його надійності та точності. Розроблено метод BuildIFCGraph, який поєднує спеціалізований геометричний аналіз типових конструктивних вузлів, загальний аналіз перетинів елементів та обробку семантичної інформації з IFC-схеми. Верифікацію методу проведено на трьох еталонних моделях із бібліотеки KIT: «AC20-FZK-Haus» (40 вершин), «AC20-Institute-Var-2» (448 вершин) та «AC20-Smiley-West-10-Bldg» (831 вершина). Кількісна оцінка показала повну відповідність за критеріями ідентифікації вершин та конструктивних ребер для всіх моделей. Якісна візуальна перевірка на прикладі моделі «AC20-FZK-Haus» підтвердила, що метод коректно розрізняє несучі взаємодії та зв'язки просторової суміжності. Розроблений метод є інструментарієм для автоматизованого створення графів, які слугують основою для побудови графових нейронних мереж (GNN), здатних прогнозувати поширення деградаційних процесів. Це відкриває перспективи для впровадження систем проактивного управління безпекою та життєвим циклом будівельних активів. Напрями подальших досліджень включають розробку математичних моделей для визначення фізичного стану елементів залежно від часу та умов експлуатації, а також створення архітектури GNN для предиктивного аналізу.

Ключові слова: графова нейронна мережа; фізично обґрунтований граф; трансформація даних; BIM-модель; IFC-файл

Вступ

Розвиток підходів до управління будівельними активами в умовах впровадження інформаційного моделювання будівель (Building Information Modeling, BIM) створює передумови для підвищення оперативності та ефективності управління безпекою. Водночас BIM-модель за своєю природою є статичним представленням стану об'єкта і не враховує часовозалежні процеси деградації, зокрема корозію, втому матеріалів, проникнення вологи чи осідання конструкцій. Крім того, такі моделі не відображають системні ефекти, за яких погіршення стану окремого елемента може призводити до негативних наслідків для суміжних компонентів будівельної системи [1; 2].

З метою подолання зазначених обмежень в архітектурно-будівельній галузі набувають поширення графові підходи, що надають можливість явно формалізувати топологічні та функціональні взаємозв'язки між елементами будівлі, які залишаються поза увагою традиційних об'єктно-орієнтованих методів аналізу BIM-даних. Представлення об'єкта будівництва у вигляді графа відкриває можливості застосування графових нейронних мереж (англ. Graph Neural Networks, GNN), здатних навчатися одночасно на атрибутах елементів (вершин) та на їхніх взаємозв'язках (ребрах), що, своєю чергою, забезпечує моделювання поширення локальних деградаційних процесів у межах всієї системи [3].

Мета та задачі дослідження

Метою дослідження є розроблення методу автоматизованого перетворення BIM-моделей з формату IFC у фізично обґрунтовані графові структури з метою створення основи для застосування графових нейронних мереж (GNN) у задачах прогнозування технічного стану та довговічності будівельних активів.

Для досягнення поставленої мети визначено такі завдання:

1. Провести аналіз сучасних досліджень, методів і моделей для оцінювання технічного стану та прогнозування довготривалої деградації інженерних конструкцій з метою обґрунтування доцільності застосування фізично обґрунтованих графових нейронних мереж.

2. Розробити метод перетворення BIM-даних з формату IFC у графову структуру, що відображає як конструктивні, так і просторові взаємозв'язки між елементами будівлі.

3. Розробити систему критеріїв для кількісної та якісної оцінки коректності запропонованого методу та провести його валідацію за цими критеріями.

Аналіз останніх досліджень і публікацій

Аналіз наукових робіт виявив декілька ключових напрямів досліджень, кожен з яких має власні переваги та обмеження, що в сукупності підкреслює необхідність розроблення нових гібридних методів.

Перша група робіт зосереджена на створенні фізично коректних моделей. У роботі [4] представлено ймовірнісний аналіз залізобетонних рам, у якому враховано взаємодію «грунт–конструкція» (soil–structure interaction, SSI). Запропонована модель поєднує нелінійний аналіз методом скінчених елементів (МСЕ) зі стохастичними моделями корозії та осідання ґрунту, тоді як надійність системи оцінюється із застосуванням методу Монте-Карло. Ключовим висновком авторів є те, що ігнорування SSI призводить до суттєвого завищення показників надійності конструкції. Водночас основним обмеженням такого підходу є його висока обчислювальна вартість. Подібний підхід продемонстровано у дослідженні [5], де проаналізовано вплив просторової мінливості корозії на надійність балкових мостів. Розроблене стохастичне поле дозволяє детально дослідити, як локалізовані процеси деградації впливають на глобальну поведінку системи. Однак істотним обмеженням цієї роботи є зосередженість лише на одному механізмі деградації – корозії, тоді як реальні

конструкції зазнають комбінованого впливу багатьох чинників, зокрема осідання, повзучості та втоми матеріалів.

Друга група робіт визнає проблему високої обчислювальної складності та пропонує її вирішення шляхом використання сурогатних моделей і класичних нейронних мереж. У дослідженні [6] запропоновано застосування методу крігінгу для побудови моделі, що апроксимує довготривалі деформації бетонних конструкцій на основі даних МСЕ. Хоча такий підхід істотно прискорює обчислення, його обмеженням є «чорна скринька» цих моделей, які не мають внутрішнього розуміння фізичної топології конструкції, що може знижувати їхню точність для складних систем. У роботі [7] фізично інформовані нейронні мережі (Physics-Informed Neural Networks, PINNs) інтегровано з BIM для моніторингу мостів. Такий метод дозволяє створити точний «цифровий двійник» для оцінювання поточного стану, однак сам підхід не містить інструментів для прогнозування довготривалої деградації.

Зазначені обмеження стимулюють пошук нових архітектур, здатних поєднати обчислювальну ефективність із фізичною обґрунтованістю. Таким інструментом постають графові нейронні мережі (Graph Neural Networks, GNN), які здатні враховувати топологію конструкції – спосіб з'єднання елементів і передачі між ними зусиль, що є ключовим аспектом для моделювання системних ефектів.

Третя група робіт демонструє успішні приклади застосування GNN у задачах механіки. Важливим прецедентом використання графових нейронних мереж у оцінюванні технічного стану є робота [8], у якій розроблено графоорієнтований цифровий двійник для швидкої оцінки сейсмічних пошкоджень портфелів будівель. Дослідження ґрунтується на гіпотезі, що стан пошкодження окремої будівлі не є незалежним від стану сусідніх об'єктів; цей зв'язок моделюється шляхом формування ребер графа на основі просторової близькості. Кожна будівля представлена у вигляді вершини з такими характеристиками, як вік, висота та конструктивний тип. Автори переконливо демонструють, що модель на основі GNN істотно перевершує традиційні методи машинного навчання, які ігнорують топологічну структуру графа.

Водночас запропонована модель орієнтована на «nowcasting» наслідків миттєвої події – землетрусу. Вона призначена для прогнозування пошкоджень, що вже відбулися, і не придатна для моделювання повільних довготривалих процесів деградації протягом місяців і років, таких як корозія, втома матеріалів або осідання. Таким чином, хоча ця робота підтверджує ефективність GNN для врахування просторових взаємозалежностей у забудованому

середовищі, вона оперує на макрорівні (рівні будівель) і розглядає статичний, післяподієвий сценарій, що обмежує можливості застосування GNN до часово залежних процесів.

У роботі [9] запропоновано просторово-часову модель відмов, яка поєднує GNN з моделлю у просторі станів для прогнозування залишкового ресурсу конструкцій. Методологія базується на двоетапному підході: на першому етапі GNN використовується для моделювання просторових взаємозв'язків між елементами системи (наприклад, між датчиками або конструктивними компонентами); на другому етапі результати GNN інтегруються у стохастичну модель у просторі станів, яка описує часову еволюцію характеристик деградації. Таке поєднання дозволяє одночасно враховувати просторову структуру та часову динаміку відмов, забезпечуючи точне прогнозування залишкового терміну експлуатації навіть у складних нелінійних сценаріях.

Проведений огляд визначив, що з одного боку, існують фізично точні, але обчислювально надмірно складні моделі; з іншого – швидкі методи машинного навчання, які або ігнорують системну топологію конструкцій, або ще не адаптовані до задач довготривалого прогнозування деградації.

У цьому контексті розроблення фізично обґрунтованої графової нейронної мережі (Physically-Grounded Graph Neural Network, PF-GNN) постає як цілеспрямоване рішення. Такий підхід спрямований на синтез переваг обох напрямів – поєднання фізичної точності та системного розуміння, притаманних аналізу MCE, з обчислювальною ефективністю методів машинного навчання. Кінцевою метою є створення інструменту, здатного забезпечувати швидке та точне довготривале прогнозування надійності всієї конструктивної системи з урахуванням взаємного впливу процесів деградації.

Виклад основного матеріалу

Метод побудови фізично обґрунтованого графа для BIM-моделі з IFC файлу

Метод призначений для перетворення BIM-даних у форматі IFC (Industry Foundation Classes) у графову структуру. У цій структурі кожна вершина представляє фізичний елемент будівлі, а ребра – конструктивні та просторові зв'язки між ними. Ключовою особливістю методу є обчислення фізично обґрунтованих атрибутів для кожної вершини, що моделюють зміну її стану під впливом деградаційних процесів з часом.

Нехай $F = \{e_1, e_2, \dots, e_m\}$ це IFC-файл, що містить множину з m елементів; $C = \{c_1, c_2, \dots, c_k\}$ – множина цільових IFC-класів стандарту IFC4, що описують будівельні елементи [10].

$R = \{\text{"CONNECTS_STRUCTURAL"}, \text{"ADJACENT"}\}$ – множина типів ребер, які моделюють передачу навантаження між несучими елементами ("CONNECTS_STRUCTURAL") та просторову суміжність ("ADJACENT"); $F_{\text{phys}} = \{f_1(\cdot), f_2(\cdot), \dots, f_k(\cdot)\}$ – множина функцій для обчислення фізичних характеристик конструктивних елементів BIM-моделі.

Метою методу є побудова зваженого графа $G = \langle V, E, X \rangle$, де $V = \{v_1, v_2, \dots, v_n\}$ – множина вершин (IFC-елементів); $n = |V|$ – кількість вершин; $E \subseteq V \times V$ – множина ребер (зв'язків); $X \in \mathbb{R}^{n \times d}$ – матриця атрибутів вершин; d – розмірність вектора атрибутів.

Метод складається з чотирьох послідовних етапів: ідентифікація вершин, визначення зв'язків, обчислення фізичних характеристик та побудова графа.

Ідентифікація вершин графа (рядки 1–9): виконується шляхом ітерації по всіх класах із попередньо визначеної множини C та відбору з IFC-файлу всіх екземплярів цих класів. Кожен вибраний IFC-елемент стає вершиною в графі. Для забезпечення швидкого доступу та відображення створюється словник (Map), що пов'яже унікальний ідентифікатор кожного елемента (GlobalId) з його індексом у графі.

Визначення зв'язків графа (рядки 10–89): складається з трьох блоків для гібридної ідентифікації зв'язків: 1) спеціалізований аналіз для балок; 2) загальний геометричний аналіз; 3) аналіз семантичних зв'язків з IFC.

1. Спеціалізований аналіз для лінійних елементів (рядки 22–40): до елементів типу "балка" (IfcBeam) застосовується окремий аналіз. Для всіх потенційних опор (стіни, колони, плити) застосовується серія спеціалізованих функцій для аналізу типів з'єднань: DetectBeamSupport (балка-пряма опора), DetectBeamTConnection (балка-T-подібне примикання) та DetectBeamColConnection (балка-колона). Усі ідентифіковані таким чином з'єднання класифікуються як CONNECTS_STRUCTURAL і додаються до множини E_1 .

2. Загальний аналіз геометричних перетинів (рядки 41–56): заснований на аналізі обмежувальних паралелепіпедів (bounding boxes) та об'ємів елементів. Для кожної пари з перетином BBox функція DetectInteraction визначає тип контакту: дотик (PERFECT_CONTACT) або об'ємний перетин (VOLUMETRIC_INTERSECTION). Ребра створюються лише у випадку фізичного контакту. Тип ребра визначається на основі класів елементів: якщо обидва елементи належать до списку несучих типів (StructTypes), ребро класифікується як CONNECTS_STRUCTURAL; в іншому випадку – як ADJACENT_TO. Результати цього аналізу формують множину E_2 .

3. Аналіз семантичних зв'язків з IFC (рядки 57–88): використовує семантичну інформацію, явно наявну в IFC-файлі, для ідентифікації елементів, пов'язаних відношенням (IfcRelationship). Якщо відношення має тип IfcRelConnectsStructuralMember, зв'язок класифікується як CONNECTS_STRUCTURAL; у всіх інших випадках – як ADJACENT_TO. Щоб уникнути дублювання, до множини E_3 додаються лише ті ребра, які не були знайдені на попередніх геометричних етапах.

Множина ребер E отримується шляхом об'єднання унікальних ребер із множин E_1 , E_2 та E_3 .

Обчислення фізичних характеристик (рядки 90–108) виконується за функціями з множини F_{phys} , вхідними параметрами для яких є геометричні параметри елемента ('ExtractBaseProps'). Ці функції моделюють зміну фізичного стану елемента з часом, враховуючи час симуляції (t_{sim_years}). Отримані дані об'єднуються в матрицю атрибутів X . Для елементів, що не мають визначеної моделі деградації, відповідний рядок матриці заповнюється нулями.

Побудова графа (рядки 109–124) для кожного елемента з множини V створюється вершина з унікальним індексом та атрибутами (ідентифікатори, тип IFC, назва, вектор фізичних характеристик з матриці X). Потім для кожного кортежу ($u, v, edge_type$) із множини E між відповідними вершинами u та v додається ребро із зазначеним типом.

Результатом роботи методу є об'єкт графа G , який є фізично обґрунтованою моделлю будівельного об'єкта, придатною для подальшого аналізу за допомогою графових нейронних мереж.

Псевдокод методу побудови фізично обґрунтованого графа BIM-моделі виглядає так:

```
FUNCTION BuildIFCGraph(IFC_File F, IFC_Classes
C, t_sim_years):
1.  $V \leftarrow []$ 
2.  $Map \leftarrow \emptyset$ 
3. FOR each class  $c$  IN  $C$ :
4.    $V \leftarrow V \cup \{e \in F \mid type(e) = c\}$ 
5. END FOR
6.  $n \leftarrow |V|$ 
7. FOR  $i = 0$  TO  $n-1$ :
8.    $Map[V[i].GlobalId] \leftarrow i$ 
9. END FOR
10.  $StructTypes \leftarrow \{IfcBeam, IfcColumn, IfcSlab, IfcWall, IfcWallStandardCase\}$ 
11.  $IdxBeams \leftarrow \{i \mid V[i].type = IfcBeam\}$ 
12.  $IdxSupport \leftarrow \{i \mid V[i].type \in \{IfcWall, IfcColumn, IfcSlab, IfcBeam\}\}$ 
13.  $IdxWall \leftarrow \{i \mid V[i].type \in \{IfcWall, IfcWallStandardCase\}\}$ 
14.  $IdxColumn \leftarrow \{i \mid V[i].type = IfcColumn\}$ 
15.  $ALL\_BBox \leftarrow$  array of size  $|V|$ 
16. FOR  $i = 0$  TO  $|V|-1$ :
```

```
17.  $ALL\_BBox[i] \leftarrow bounding\_box(V[i])$ 
18. END FOR
19.  $E_1 \leftarrow \emptyset$ 
20.  $E_2 \leftarrow \emptyset$ 
21.  $E_3 \leftarrow \emptyset$ 
22. FOR  $i$  IN  $IdxBeams$ :
23.   FOR  $j$  IN  $IdxSupport$ :
24.     IF  $i = j$ : CONTINUE
25.      $connected \leftarrow False$ 
26.     IF  $j \in IdxWall$  AND not  $connected$ :
27.        $info\_t \leftarrow DetectBeamTConnection(V[i], V[j], ALL\_BBox[i], ALL\_BBox[j])$ 
28.       IF  $info\_t \neq \perp$  AND  $info\_t.valid = True$ :
29.          $E_1 \leftarrow E_1 \cup \{(\min(i,j), \max(i,j), CONNECTS\_STRUCTURAL)\}$ 
30.          $connected \leftarrow True$ 
31.     IF  $j \in IdxColumn$  AND not  $connected$ :
32.        $info\_c \leftarrow DetectBeamColumnConnection(V[i], V[j], ALL\_BBox[i], ALL\_BBox[j])$ 
33.       IF  $info\_c \neq \perp$  AND  $info\_c.valid = True$ :
34.          $E_1 \leftarrow E_1 \cup \{(\min(i,j), \max(i,j), CONNECTS\_STRUCTURAL)\}$ 
35.          $connected \leftarrow True$ 
36.     IF not  $connected$ :
37.        $info\_m \leftarrow DetectBeamSupportConnection(V[i], V[j], ALL\_BBox[i], ALL\_BBox[j])$ 
38.       IF  $info\_m \neq \perp$  AND  $info\_m.valid = True$ :
39.          $E_1 \leftarrow E_1 \cup \{(\min(i,j), \max(i,j), CONNECTS\_STRUCTURAL)\}$ 
40.          $connected \leftarrow True$ 
41. FOR  $i = 0$  TO  $|V|-1$ :
42.   IF  $V[i].type = IfcBeam$ : CONTINUE
43.   FOR  $j = i+1$  TO  $|V|-1$ :
44.     IF  $V[j].type = IfcBeam$ : CONTINUE
45.     IF  $intersects(ALL\_BBox[i], ALL\_BBox[j]) = False$ : CONTINUE
46.      $r \leftarrow DetectInteraction(V[i], V[j], ALL\_BBox[i], ALL\_BBox[j])$ 
47.     IF  $r.type = NO\_CONTACT$ : CONTINUE
48.     IF  $V[i].type \in StructTypes$  AND  $V[j].type \in StructTypes$ :
49.        $edge\_type \leftarrow CONNECTS\_STRUCTURAL$ 
50.     ELSE:
51.        $edge\_type \leftarrow ADJACENT\_TO$ 
52.     IF  $r.type = PERFECT\_CONTACT$ :
53.        $E_2 \leftarrow E_2 \cup \{(\min(i,j), \max(i,j), edge\_type)\}$ 
54.     ELSE IF  $r.type = VOLUMETRIC\_INTERSECTION$ :
55.       IF  $r.v_1 > 0.05$  OR  $r.v_2 > 0.05$ :
56.          $E_2 \leftarrow E_2 \cup \{(\min(i,j), \max(i,j), edge\_type)\}$ 
57. Relationships  $\leftarrow F.by\_type("IfcRelationship")$ 
58. FOR each  $r$  IN Relationships:
59.    $src\_id \leftarrow \perp$ 
```



```

60.  tgt_id ← ⊥
61.  TRY:
62.    IF hasattr(r, "RelatingElement") AND hasattr(r,
"RelatedElement"):
63.      src_id ← r.RelatingElement.GlobalId
64.      tgt_id ← r.RelatedElement.GlobalId
65.    ELSE IF hasattr(r, "RelatingStructure") AND
hasattr(r, "RelatedElements"):
66.      IF r.RelatedElements ≠ ∅:
67.        src_id ← r.RelatingStructure.GlobalId
68.        tgt_id ← r.RelatedElements[0].GlobalId
69.    ELSE IF hasattr(r, "RelatingObject") AND
hasattr(r, "RelatedObjects"):
70.      IF r.RelatedObjects ≠ ∅:
71.        src_id ← r.RelatingObject.GlobalId
72.        tgt_id ← r.RelatedObjects[0].GlobalId
73.    ELSE IF hasattr(r,
"RelatingStructuralMember") AND hasattr(r,
"RelatedStructuralConnection"):
74.      src_id ←
r.RelatingStructuralMember.GlobalId
75.      tgt_id ←
r.RelatedStructuralConnection.GlobalId
76.  EXCEPT:
77.    CONTINUE
78.  IF src_id = ⊥ OR tgt_id = ⊥: CONTINUE
79.  IF src_id ∉ Map OR tgt_id ∉ Map: CONTINUE
80.  u ← Map[src_id]
81.  v ← Map[tgt_id]
82.  IF u > v: swap(u,v)
83.  IF r.is_a() = "IfcRelConnectsStructuralMember":
84.    etype ← CONNECTS_STRUCTURAL
85.  ELSE:
86.    etype ← ADJACENT_TO
87.  IF (u,v,etype) ∉ (E1 ∪ E2):
88.    E3 ← E3 ∪ { (u,v,etype) }
89.  E ← E1 ∪ E2 ∪ E3
90.  nfeatures ← 10

91.  X ← zeros(|V|, nfeatures)
92.  FOR i = 0 TO |V|-1:
93.    elem ← V[i]
94.    elem_type ← elem.type
95.    params ← ExtractBaseProps(elem, F)
96.    env ← EnvironmentProps(t_sim_years)
97.    SWITCH elem_type:
98.      CASE "IfcColumn":
99.        physical_state ← f1(params, env,
t_sim_years)
100.     CASE "IfcBeam":
101.       physical_state ← f2(params, env,
t_sim_years)
102.     CASE "IfcSlab":
103.       physical_state ← f3(params, env,
t_sim_years)
104.     CASE "IfcWall", "IfcWallStandardCase":
105.       physical_state ← f4(params, env,
t_sim_years)
106.     DEFAULT:
107.       physical_state ← zeros(nfeatures)
108.  X[i,:] ← [params.geometric | physical_state]
109.  G ← Graph()
110.  FOR i = 0 TO |V|-1:
111.    element ← V[i]
112.    features ← X[i,:]
113.    node_attrs ← {
114.      'node_id' : i,
115.      'global_id' : element.GlobalId,
116.      'ifc_type' : element.is_a(),
117.      'name' : element.Name
118.    }
119.    FOR j = 0 TO |features|-1:
120.      node_attrs["feature_" + str(j)] ← features[j]
121.    G.add_node(i, **node_attrs)
122.  FOR (u,v,edge_type) IN E:
123.    G.add_edge(u, v, edge_type)
124.  RETURN G

```

Часова складність методу побудови графа

Нехай n – кількість будівельних елементів (вершин), ідентифікованих в IFC-файлі; $|R|$ – загальна кількість сутностей *IfcRelationship* у файлі; g – обчислювальна вартість однієї операції виявлення геометричного перетину або з'єднання; $|E|$ – кількість ребер у побудованому графі. Складність методу аналізується шляхом розгляду його чотирьох основних блоків:

1. Ідентифікація вершин графа (рядки 1–9) включає визначення IFC-класів ($|C|$) які включені в IFC-файл має складність пропорційну кількості знайдених елементів $O(n)$. Складність побудови хеш-таблиці для зберігання відповідей *GlobalId* на індекс складає $O(n)$ так як вставка в хеш-таблицю має середню часову складність $O(1)$.

2. Визначення зв'язків графа (рядки 10-89) складається з створення індексних підмножин та попереднього обчислення обмежувальних паралелепіпедів, припускаючи, що функція *bounding_box* має постійну вартість b , тоді складність операцій становить $O(n)$. Цикл обходу по всіх балках (*IdxBeams*) та всіх потенційних опорах (*IdxSupport*) (рядки 22–40) у найгіршому випадку має складність $O(n^2 \cdot g)$. Для кожної пари, чиї обмежувальні паралелепіпеди перетинаються, викликається функція виявлення геометричної взаємодії (*DetectInteraction*) з вартістю g . Таким чином складність рядків 41–56 складає $O(n^2 \cdot g)$. Обхід відношень по всіх $|R|$ має складність $O(|R|)$ є лінійно пропорційною кількості елементів n , таким чином, цей крок має складність $O(n)$. Домінуючим терміном для блоку 2 є $O(n^2 \cdot g)$.

3. Обчислення фізичних характеристик (рядки 90–108) для кожної вершини викликає функції для обчислення фізичних атрибутів. Складність цих фізичних розрахунків залежить від параметрів симуляції, але не залежить від n . Таким чином, робота, що виконується для одного елемента, є постійною, $O(1)$, відносно розміру графа. Загальна складність цього етапу — $O(n)$

4. Побудова графа (Рядки 109–124) виконується шляхом додавання вершин та ребер до графової структури. В середньому часова складність такої операції складає $O(1)$, значить загальна складність цього блоку – $O(n+|E|)$.

Загальна часова складність методу BuildIFCGraph є сумою складностей його етапів: $T(n)=O(n)+O(n^2 \cdot g)+O(n)+O(n+|E|) \approx O(n^2 \cdot g)$.

Критерії для оцінювання коректності запропонованого методу

Для оцінювання коректності розробленого методу було обрано еталонні BIM-моделі «AC20-FZK-Haus», "AC20-Institute-Var-2", "Smiley West - terraced houses" [11]. Ці моделі є загальнодоступною та входять до бібліотеки Karlsruhe Institute of Technology (KIT) [12]. Збережена у форматі IFC4, модель представляє житловий будинок, багатоповерхову офісну будівлю та багатоквартирну житлову будівлю відповідно (рис. 4-6).

Обрані еталонні моделі містять конструктивні, архітектурні, огорожувальні та допоміжні елементи. Кількісний склад цільових елементів для кожної моделі наведено нижче.

Конструктивні елементи (основа несучої системи будівлі, що сприймає та передає навантаження):

«AC20-FZK-Haus»: IfcWallStandardCase (13), IfcSlab (4), IfcBeam (4), IfcStair (1).

«AC20-Institute-Var-2»: IfcWallStandardCase (121), IfcSlab (26), IfcColumn (2), IfcStair (4).

«AC-20-Smiley-West-10-Bldg»: IfcWallStandardCase (270), IfcSlab (120), IfcColumn (20), IfcBeam (10), IfcStair (30), IfcWall (11).

Архітектурні та огорожувальні елементи (формують зовнішній вигляд та внутрішній простір, не виконуючи основної несучої функції):

«AC20-FZK-Haus»: IfcWindow (11), IfcDoor (5), IfcRailing (2).

«AC20-Institute-Var-2»: IfcWindow (206), IfcDoor (77), IfcRailing (12).

«AC-20-Smiley-West-10-Bldg»: IfcWindow (80), IfcDoor (170), IfcRailing (120).

Загальна кількість конструктивних та архітектурних елементів, що розглядалися, для моделей становить: 40 («AC20-FZK-Haus»), 448 («AC20-Institute-Var-2») та 831 («AC-20-Smiley-West-10-Bldg»).

Допоміжні сутності, які не є фізичними елементами і тому не враховуються запропонованим методом при побудові графа: IfcOpeningElement (отвори), IfcAnnotation (анотації), IfcSpace (простори), IfcFurnishingElement (меблі та обладнання), IfcMember (деталізовані компоненти) та IfcVirtualElement (концептуальні зв'язки).

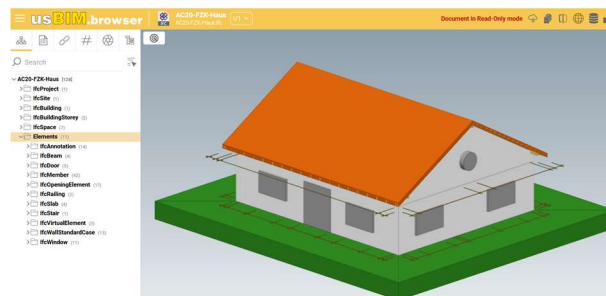


Рисунок 1 – BIM-модель «AC20-FZK-Haus»



Рисунок 2 – BIM-модель "AC20-Institute-Var-2"



Рисунок 3 – BIM-модель «AC-20-Smiley-West-10-Bldg»

Для перевірки методу побудови графа визначені три критерії.

Критерій 1: Коректність ідентифікації вершин. Характеризує здатність методу точно виділяти фізичні елементи будівлі, що належать до заданої множини цільових IFC-класів C , та коректно ігнорувати всі інші типи сутностей. Оцінювання здійснюється шляхом порівняння множини ідентифікаторів GlobalId, отриманих методом, з еталонною множиною, отриманою з моделі, із забезпеченням повної відповідності як за складом, так і за кількістю.

Критерій 2: Точність ідентифікації конструктивних ребер. Спрямований на перевірку здатності методу коректно ідентифікувати множину ребер типу CONNECTS_STRUCTURAL, оскільки саме ці зв'язки відповідають за моделювання несучої системи будівлі. Еталонна множина конструктивних ребер визначається на основі гібридного підходу:

1) аналіз семантики IFC (зокрема, відношень IfcRelConnectsStructuralMember, IfcRelConnectsElements та IfcRelAggregates);

2) геометричний аналіз, що підтверджує фізичний контакт між парами несучих елементів.

Критерій 3: Якісна перевірка типів ребер. Передбачає перевірку правильності класифікації зв'язків між окремими елементами шляхом візуального аналізу вибірки ідентифікованих ребер з використанням програмного засобу usBIM.browser [12]. Для ребер CONNECTS_STRUCTURAL візуальне підтвердження має засвідчувати наявність безпосереднього фізичного контакту між несучими елементами. Для ребер ADJACENT_TO підтвердження є просторові відношення, такі як включення одного елемента в інший або їхня безпосередня близькість.

Метод було реалізовано з використанням бібліотек мови програмування Python 3.10: ifcopenshell: для читання IFC-файлу та доступу до його сутностей, атрибутів і геометричних представлень; NetworkX: для створення, обробки та аналізу графових структур; Matplotlib: для візуалізації отриманих результатів, зокрема структури графа та статистичних розподілів його властивостей.

Результати

Запропонований метод (BuildIFCGraph) було застосовано до трьох еталонних BIM-моделей для створення фізично обґрунтованих графів. Перевірка отриманих результатів проводилася за трьома критеріями: точність ідентифікації вершин, точність ідентифікації конструктивних ребер та якісна візуальна перевірка.

Для базової моделі «AC20-FZK-Haus» було проведено повний цикл валідації, що включав кількісну та якісну оцінку.

Кількісна перевірка. Згенерований граф (рис. 4) складається з 40 вершин, що повністю відповідає кількості цільових фізичних елементів в еталонній моделі.

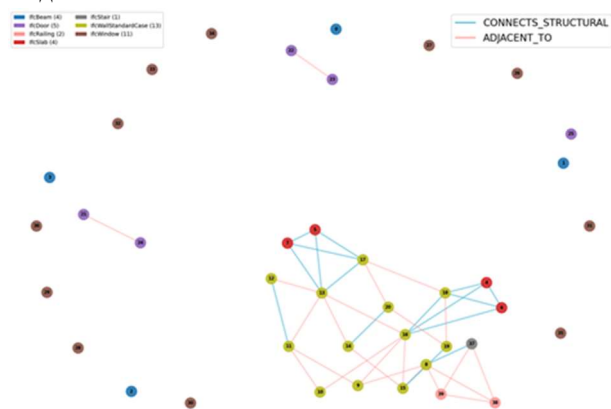


Рисунок 4 – Фізично обґрунтований граф, створений з BIM-моделі «AC20-FZK-Haus»

Статистичний розподіл типів вершин наведено в табл. 1.

Аналіз ребер (табл. 2) показав, що метод ідентифікував 16 конструктивних (CONNECTS_STRUCTURAL) ребер, цей показник збігається з результатами, отриманими за допомогою еталонних правил перевірки, що підтверджує виконання критеріїв точності ідентифікації вершин та конструктивних ребер.

Таблиця 1 – Статистичний розподіл типів вершин у згенерованому графі з BIM-моделі «AC20-FZK-Haus»

Тип елемента (IFC)	Кількість	Частка (%)
IfcWallStandardCase	13	32.5
IfcWindow	11	27.5
IfcDoor	5	12.5
IfcBeam	4	10
IfcSlab	4	10
IfcRailing	2	5
IfcStair	1	2.5
Усього	40	100

Таблиця 2 – Статистичний розподіл типів ребер у згенерованому графі BIM-моделі «AC20-FZK-Haus»

Тип ребра	Кількість	Частка (%)
ADJACENT TO	23	59
CONNECTS_STRUCTURAL	16	41
Усього	39	100

Для підтвердження фізичної основи ідентифікованих зв'язків було проведено візуальну інспекцію вибірки ребер (табл. 3, 4) з використанням usBIM.browser. Цей процес передбачає ізоляцію в 3D-моделі конкретних елементів, що відповідають ребру згенерованого графа, та підтвердження їхньої фізичної взаємодії.

Таблиця 3 – Конструктивні зв'язки з BIM-моделі «AC20-FZK-Haus»

N	u - GlobalId	u-Type	v-GlobalId	v-Type
1	1pPHnf7cX CpPsNEnQ f8_6B	IfcSlab	07Enbsqm9C7 AQc9iyBwfS D	IfcSlab
2	1pPHnf7cX CpPsNEnQ f8_6B	IfcSlab	1bzfVsJqn8De 5PukCrqylz	IfcWallStandardCase
3	1pPHnf7cX CpPsNEnQ f8_6B	IfcSlab	25OWQvmXj 5BPgyergP43t Y	IfcWallStandardCase
4	2RGlQk4x H47RHK93 zcTzUL	IfcSlab	2IxUUNUVP B6Ob\$eicCfP 2N	IfcSlab
5	2RGlQk4x H47RHK93 zcTzUL	IfcSlab	3rPX_Juz59pe XXY6wDJ18	IfcWallStandardCase

Закінчення табл. 3

N	u - GlobalId	u-Type	v-GlobalId	v-Type
6	2RGIQk4xH47RHK93zcTzUL	IfcSlab	0knNIAVBPBFvBy_m5QVHsU	IfcWallStandardCase
7	07Enbsqm9C7AQC9iyBwfSD	IfcSlab	1bzfVsJqn8De5PukCrqylz	IfcWallStandardCase
8	07Enbsqm9C7AQC9iyBwfSD	IfcSlab	25OWQvmXj5BPgyergP43tY	IfcWallStandardCase
9	2IxUUNUVPB6Ob\$eicCfP2N	IfcSlab	3rPX_Juz59peXXY6wDJl18	IfcWallStandardCase
10	2IxUUNUVPB6Ob\$eicCfP2N	IfcSlab	0knNIAVBPBFvBy_m5QVHsU	IfcWallStandardCase
11	2XPyKWY018sA1ygZKgQPtU	IfcWallStandardCase	38a9vdh9bF5Qg28GWyHhlr	IfcStair
12	3jjW3rL656ex34Gws22EfM	IfcWallStandardCase	1\$wmdwWPjDYuku_ghVkyne	IfcWallStandardCase
13	3rPX_Juz59peXXY6wDJl18	IfcWallStandardCase	0knNIAVBPBFvBy_m5QVHsU	IfcWallStandardCase
14	16DNNqzfP2thtfaOfIvsKAsKA	IfcWallStandardCase	3Ttjr\$59XEWfWN1WUHjelZ	IfcWallStandardCase
15	25fsbPyk15VvuXISyNKenK	IfcWallStandardCase	3VCarUKGH1buLo22Ozxe6J	IfcWallStandardCase
16	1bzfVsJqn8De5PukCrqylz	IfcWallStandardCase	25OWQvmXj5BPgyergP43tY	IfcWallStandardCase

Таблиця 4 – Суміжні зв'язки з BIM-моделі «AC20-FZK-Haus»

N	u - GlobalId	u-Type	v-GlobalId	v-Type
1	1Oms875aH3Wg\$9l65H2ZGw	IfcDoor	2jTRqchjif7oB0yhQ6462T0	IfcDoor
2	0pGAjJLMP3ifYPATVF5xAR	IfcDoor	2qiPPF3FrF8OIqfrKiSUqm	IfcDoor
3	2XPyKWY018sA1ygZKgQPtU	IfcWallStandardCase	0o5vgCKyTBzO5\$QJcl2YDP	IfcRailin g
4	2XPyKWY018sA1ygZKgQPtU	IfcWallStandardCase	09axaKU7X1_O8n6dmmiZ1E	IfcRailin g
5	38a9vdh9bF5Qg28GWyHhlr	IfcStair	0o5vgCKyTBzO5\$QJcl2YDP	IfcRailin g
6	38a9vdh9bF5Qg28GWyHhlr	IfcStair	09axaKU7X1_O8n6dmmiZ1E	IfcRailin g
7	0o5vgCKyTBzO5\$QJcl2YDP	IfcRailin g	09axaKU7X1_O8n6dmmiZ1E	IfcRailin g

Закінчення табл. 4

N	u - GlobalId	u-Type	v-GlobalId	v-Type
8	2XPyKWY018sA1ygZKgQPtU	IfcWallStandardCase	3Pfs_Y_DBAfq5naM6zD2Z	IfcWallStandardCase
9	2ptk1k7qn8_Qk22vjh\$0DE	IfcWallStandardCase	3jjW3rL656ex34Gws22EfM	IfcWallStandardCase
10	3Pfs_Y_DBAfq5naM6zD2Z	IfcWallStandardCase	3jjW3rL656ex34Gws22EfM	IfcWallStandardCase
11	1\$wmdwWPjDYuku_ghVkyne	IfcWallStandardCase	3rPX_Juz59peXXY6wDJl18	IfcWallStandardCase
12	3jjW3rL656ex34Gws22EfM	IfcWallStandardCase	3rPX_Juz59peXXY6wDJl18	IfcWallStandardCase
13	3rPX_Juz59peXXY6wDJl18	IfcWallStandardCase	16DNNqzfP2thtfaOfIvsKAsKA	IfcWallStandardCase
14	16DNNqzfP2thtfaOfIvsKAsKA	IfcWallStandardCase	25fsbPyk15VvuXISyNKenK	IfcWallStandardCase
15	2XPyKWY018sA1ygZKgQPtU	IfcWallStandardCase	25fsbPyk15VvuXISyNKenK	IfcWallStandardCase
16	3rPX_Juz59peXXY6wDJl18	IfcWallStandardCase	1bzfVsJqn8De5PukCrqylz	IfcWallStandardCase
17	25fsbPyk15VvuXISyNKenK	IfcWallStandardCase	1bzfVsJqn8De5PukCrqylz	IfcWallStandardCase
18	2ptk1k7qn8_Qk22vjh\$0DE	IfcWallStandardCase	1bzfVsJqn8De5PukCrqylz	IfcWallStandardCase
19	3Pfs_Y_DBAfq5naM6zD2Z	IfcWallStandardCase	1bzfVsJqn8De5PukCrqylz	IfcWallStandardCase
20	0knNIAVBPFvBy_m5QVHsU	IfcWallStandardCase	25OWQvmXj5BPgyergP43tY	IfcWallStandardCase
21	25OWQvmXj5BPgyergP43tY	IfcWallStandardCase	3VCarUKGH1buLo22Ozxe6J	IfcWallStandardCase
22	3VCarUKGH1buLo22Ozxe6J	IfcWallStandardCase	3Ttjr\$59XEWfWN1WUHjelZ	IfcWallStandardCase
23	0knNIAVBPFvBy_m5QVHsU	IfcWallStandardCase	3Ttjr\$59XEWfWN1WUHjelZ	IfcWallStandardCase

Приклади перевірки конструктивних ребер (CONNECTS_STRUCTURAL): стіна-плита (рис. 5): підтверджено пряме спірання стіни на плиту фундаменту, що є умовою передачі навантаження. Плита-плита (рис. 6): показано, що плити різних поверхів, хоч і не мають прямого контакту, функціонально пов'язані через вертикальні несучі стіни, що забезпечує передачу навантажень. Метод коректно ідентифікував цей зв'язок як конструктивний.

Стіна-сходи (рис. 7): продемонстровано, що сходова площадка вбудована у несучу стіну, яка забезпечує її підтримку. Стіна-стіна (рис. 8): підтверджено Т-подібне з'єднання двох стін, що забезпечує бічну стійкість.

Приклади перевірки суміжних ребер (ADJACENT_TO): Було підтверджено коректність виявлення зв'язків просторової близькості між елементами, що не виконують несучу функцію, наприклад, між двома дверима на одному фасаді (рис. 9), сегментами поручнів (рис. 10) та між стіною і прилеглим поручнем (рис. 11). На рис. 12, де метод правильно класифікував зв'язок між двома паралельними, але не з'єднаними конструктивно стінами як ADJACENT_TO, на відміну від Т-подібних перетинів. Це демонструє здатність методу розрізняти пряму несучу взаємодію та просту просторову близькість.

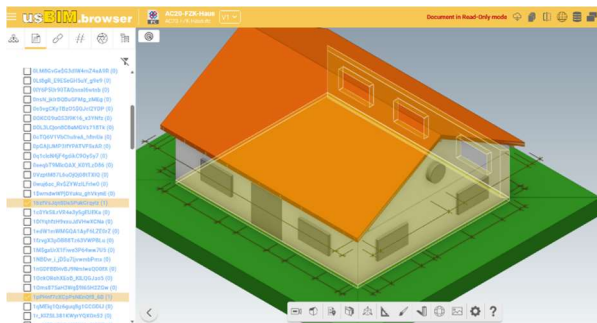


Рисунок 5 – Візуальна перевірка зв'язку між IfcSlab (GlobalId: 1pPHnf7cXCpPsNEnQf8_6B) та IfcWallStandardCase (GlobalId: 1bzfvJqn8De5PukCrqylz)

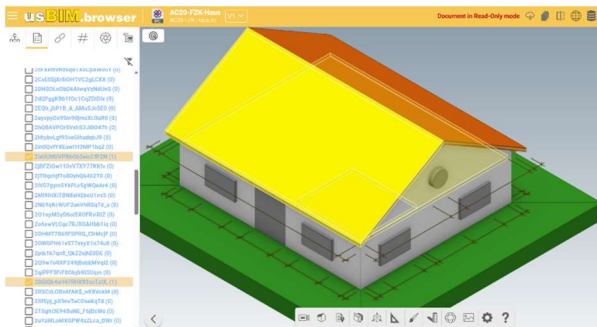


Рисунок 6 – Візуальна перевірка зв'язку між двома елементами IfcSlab (GlobalId: 2RGlQk4xH47RHK93zcTzUL та GlobalId: 2IxUUNUVPB6ObSeicCfP2N)

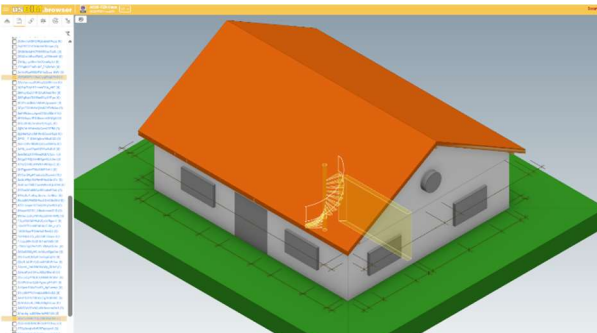


Рисунок 7 – Візуальна перевірка зв'язку між IfcWallStandardCase (GlobalId: 2XPYKWY018sAlygZKgQPtU) та IfcStair (GlobalId: 38a9vdh9bF5Qg28GWyHhIrl)

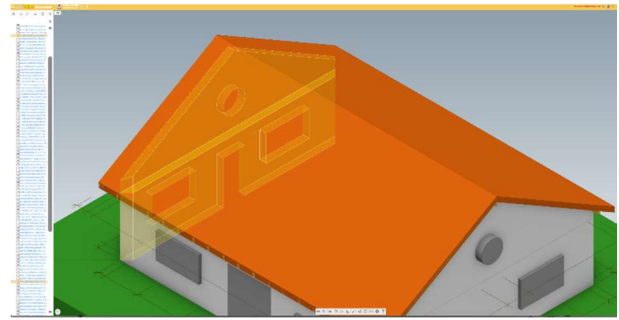


Рисунок 8 – Візуальна перевірка зв'язку між двома елементами IfcWallStandardCase (GlobalId: 3rPX_Juz59peXXY6wDJl18 та GlobalId: 0knNIAVBPFvBy_m5QVHsU)

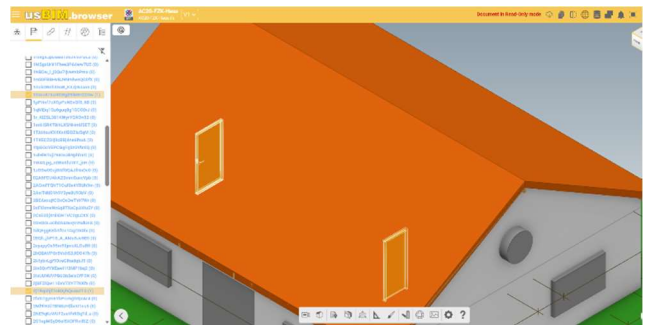


Рисунок 9 – Візуальна перевірка зв'язку між двома елементами IfcDoor (GlobalId: 1Oms875aH3Wg\$9l65H2ZGw та GlobalId: 2jTRqchif7oB0yhQ6462T0)



Рисунок 10 – Візуальна перевірка зв'язку між двома елементами IfcRailing (GlobalId: 0o5vgCKyTBzO5\$QJcI2YDP та GlobalId: 09axaKU7X1_O8n6dmmiZIE)



Рисунок 11 – Візуальна перевірка зв'язку між IfcRailing та IfcWall (GlobalId: 0o5vgCKyTBzO5\$QJcI2YDP та GlobalId: 2XPYKWY018sAlygZKgQPtU)

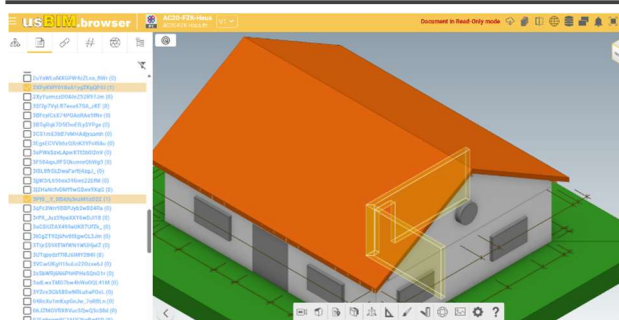


Рисунок 12 – Візуальна перевірка зв'язку між двома елементами IfcWall (GlobalId: 2XPYKWy018sA1ygZKgQPtU та GlobalId: 3PfS_Y_DBAfq5naM6zD2Z).

Для більш складних та масштабних моделей «AC20-Institute-Var-2» та «AC-20-Smiley-West-10-Bldg» було проведено перевірку за кількісними критеріями. Якісна візуальна перевірка всієї вибірки ребер була ускладнена через велику кількість елементів. Згенеровані графи для моделей «AC20-Institute-Var-2» та «AC-20-Smiley-West-10-Bldg» показано на рис. 13 та рис. 14 відповідно. «AC20-Institute-Var-2»: метод успішно ідентифікував 448 вершин, що відповідає кількості цільових елементів у BIM-моделі (табл. 5). Кількість визначених конструктивних ребер склала 360, що збігається з очікуваним результатом, розрахованим за еталонними правилами (табл. 6). «AC-20-Smiley-West-10-Bldg»: Для цієї моделі було згенеровано граф з 831 вершини (табл. 7). Кількість ідентифікованих конструктивних ребер становить 1653, що також відповідає еталонному значенню (табл. 8).

Таблиця 5 – Статистичний розподіл типів вершин у згенерованому графі з BIM-моделі «AC20-Institute-Var-2»

Тип елемента (IFC)	Кількість	Частка (%)
IfcWindow	206	46
IfcWallStandardCase	121	27
IfcDoor	77	17.2
IfcSlab	26	5.8
IfcRailing	12	2.7
IfcStair	4	0.9
ifcColumn	2	0.4
Усього	448	100

Таблиця 6 – Статистичний розподіл типів ребер у згенерованому графі BIM-моделі «AC20-Institute-Var-2»

Тип ребра	Кількість	Частка (%)
ADJACENT_TO	2449	87
CONNECTS_STRUCTURAL	360	13
Усього	2809	100

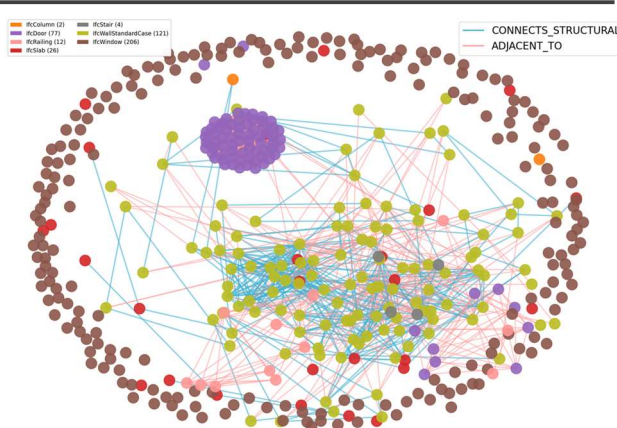


Рисунок 13 – Фізично обґрунтований граф, створений з BIM-моделі "AC20-Institute-Var-2"

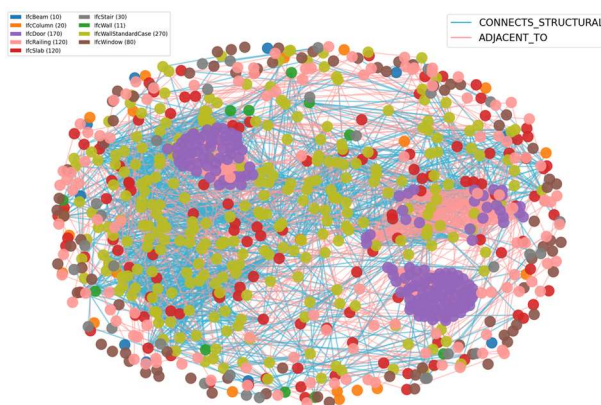


Рисунок 14 – Фізично обґрунтований граф, створений з BIM-моделі «AC-20-Smiley-West-10-Bldg»

Таблиця 7 – Статистичний розподіл типів вершин у згенерованому графі з BIM-моделі «AC-20-Smiley-West-10-Bldg»

Тип елемента (IFC)	Кількість	Частка (%)
IfcWallStandardCase	270	32.5
IfcDoor	170	20.5
IfcSlab	120	14.4
IfcRailing	120	14.4
IfcWindow	80	9.6
IfcStair	30	3.6
ifcColumn	20	2.4
ifcWall	11	1.3
IfcBeam	10	1.2
Усього	831	100

Таблиця 8 – Статистичний розподіл типів ребер у згенерованому графі BIM-моделі «AC-20-Smiley-West-10-Bldg»

Тип ребра	Кількість	Частка (%)
ADJACENT_TO	6137	78.8
CONNECTS_STRUCTURAL	1653	21.2
Усього	7790	100

Таким чином, фізично обґрунтовані графи є структурою даних для наступного етапу предиктивного моделювання. Зв'язки в графі будуть використовуватися для призначення відповідних вагових коефіцієнтів, що контролюють ступінь впливу однієї вершини на її сусідів при прогнозуванні майбутніх станів. Вершини, ідентифіковані як незв'язані, будуть розглядатися як ізольовані компоненти в рамках GNN-моделі. Під час навчання ці вершини матимуть петлі на себе зі стандартною вагою, тоді як їхні зв'язки з усіма іншими вершинами матимуть нульову вагу.

Подяка

Дослідження було виконане в рамках реалізації науково-дослідної роботи «Методологія визначення тональності та класифікації мультимодального контенту в проєктах ревіталізації територій на основі нейромережових методів» (№ 5 ДБ-2025), номер державної реєстрації 0125U001683, що фінансується державним бюджетом України.

Висновки

У дослідженні запропоновано методологічні основи для застосування графових нейронних мереж у сфері моніторингу будівельних активів. Для цього було розроблено та перевірено новий метод автоматизованого перетворення статичних BIM-даних з формату IFC у фізично обґрунтовані графові структури.

Розроблений метод поєднує геометричний аналіз для типових конструктивних вузлів та оброблення семантичної інформації з IFC, що дозволяє ідентифікувати конструктивні і просторові взаємозв'язки.

У статті сформульовано та обґрунтовано систему критеріїв для верифікації згенерованих графів, що включає: точність ідентифікації вершин, точність ідентифікації конструктивних ребер та якісну візуальну перевірку типів зв'язків. Сформульовані критерії дали змогу провести оцінку коректності запропонованого методу.

Проведено експериментальну апробацію методу на трьох еталонних BIM-моделях різного масштабу та функціонального призначення («AC20-FZK-Haus», «AC20-Institute-Var-2», «AC-20-Smiley-West-10-Bldg»). Результати перевірки підтвердили точність роботи запропонованого методу за визначеними критеріями.

Практична значущість роботи полягає у створеному та перевіреному програмному рішенні, яке слугує основою для подальшого розроблення GNN-моделей, здатних прогнозувати технічний стан та довговічність будівельних активів. Запропонований підхід може бути використаний для створення динамічних цифрових двійників та впровадження систем проактивного управління безпекою в будівельній галузі.

Для подальших досліджень передбачається виконання двох взаємопов'язаних етапів. По-перше, розроблення моделей для визначення фізичного стану конструктивних елементів залежно від часу та умов експлуатації, що надасть змогу розширити характеристики вершин графа фізичними атрибутами. По-друге, на основі отриманих характеристик вершин передбачається розробка архітектури графової нейронної мережі, здатної на фізично обґрунтованому графі прогнозувати поширення деградаційних процесів.

Список літератури / References

1. Solihin, W., Liu, Z., Lu, Y., & Wei, L. (2024). BIM-based automated rule-checking in the AECO industry: Learning from semiconductor manufacturing. *Automation in Construction*, 162, Article 105406. <https://doi.org/10.1016/j.autcon.2024.105406>.
2. Solovei, O., Honcharenko, T., & Fesan, A. (2024). Technologies to manager big data of urban building projects. *Management of Development of Complex Systems*, 60, 121–128. <https://doi.org/10.32347/2412-9933.2024.60.121-128>.
3. Zhu, J., Chong, H. Y., Zhao, H., Wu, J., Tan, Y., & Xu, H. (2022). The application of graph in BIM/GIS integration. *Buildings*, 12(12), Article 2162. <https://doi.org/10.3390/buildings12122162>.
4. Liu, B., Xue, J., Lehan, B. M., & Yin, Z. Y. (2024). Time-dependent soil–structure interaction analysis using a macro-element foundation model. *Engineering Structures*, 308, Article 118046. <https://doi.org/10.1016/j.engstruct.2024.118046>.
5. Kenshel, O. M., O'Connor, A. J., Suleiman, M. F., & Jarushi, F. E. (2021). Role of spatial variability in the service life prediction of RC bridges affected by corrosion. *Journal of Civil & Environmental Engineering*, 11 (2), 1–12.
6. Shao, C., Xu, Y., Chen, H., Zheng, S., & Qin, X. (2023). Ordinary Kriging interpolation method combined with FEM for arch dam deformation field estimation. *Mathematics*, 11(5), Article 1106. <https://doi.org/10.3390/math11051106>.
7. Radbakhsh, S. H., Nik-Bakht, M., & Zandi, K. (2023). Structural Digital Twin of concrete infrastructure powered with physics-informed neural networks. In *Interdisciplinary Symposium on Smart & Sustainable Infrastructures* (pp. 1101–1113). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-53389-1_97.
8. Maurizi, M., Gao, C., & Berto, F. (2022). Predicting stress, strain and deformation fields in materials and structures with graph neural networks. *Scientific Reports*, 12, Article 21834. <https://doi.org/10.1038/s41598-022-26424-3>.
9. Wu, X., Liu, Z., & Wang, L. (2025). Spatio-temporal degradation model with graph neural network and structured state space model for remaining useful life prediction. *Reliability Engineering & System Safety*, 256, Article 110770. <https://doi.org/10.1016/j.ress.2024.110770>.

10. BuildingSMART. (2023). *Industry Foundation Classes (IFC) 4.3.2 Documentation*. <https://ifc43-docs.standards.buildingsmart.org/>
11. Karlsruhe Institute of Technology. (2023). *KIT IFC Examples*. https://www.ifcwiki.org/index.php?title=KIT_IFC_Examples.
12. Wikipedia. (2024). *Karlsruhe Institute of Technology (KIT)*. https://en.wikipedia.org/wiki/Karlsruhe_Institute_of_Technology.
13. ACCA Software. (2023). *BIM Viewer software application*. <https://www.accasoftware.com/en/bim-viewer>.
14. Honcharenko, T. (2020). Cluster method of forming metadata of multidimensional information systems for solving general planning problems. *Management of Development of Complex Systems*, 42, 93–101. <https://doi.org/10.32347/2412-9933.2020.42.93-101>.
15. Dolhopolov, S., Honcharenko, T., Savenko, V., Balina, O., Bezklubenko, I. S., & Liashchenko, T. (2023). Construction site modeling objects using artificial intelligence and BIM technology: A multi-stage approach. In *2023 IEEE International Conference on Smart Information Systems and Technologies (SIST)* (pp. 174–179). IEEE. <https://doi.org/10.1109/SIST58284.2023.10223543>.

Стаття надійшла до редколегії 08.12.2025

Olga Solovei

PhD in Technical Sciences, Associate Professor, Doctoral Student at the Department of Information Technology,
<https://orcid.org/0000-0001-8774-7243>,

Kyiv National University of Construction and Architecture, Kyiv

Honcharenko Tetiana

DSc (Eng.), Head of the Department of Information Technology,
<https://orcid.org/0000-0003-2577-6916>

Kyiv National University of Construction and Architecture, Kyiv

METHOD FOR CREATING GRAPHS BASED ON BIM MODELS FOR PREDICTIVE MONITORING OF BUILDINGS

Abstract. A methodology for the automated transformation of BIM models into physics-based graph representations suitable for further application in building durability forecasting tasks has been developed and verified. The following tasks were solved: development of a method that transforms BIM data into a graph structure where vertices represent physical elements and edges represent their structural and spatial interconnections; creation of a system of criteria for evaluating the correctness of the method's performance, including quantitative and qualitative verification; experimental testing of the developed method on a set of reference BIM models of various scales to confirm its reliability and accuracy. The BuildIFCGraph method was developed, which combines specialized geometric analysis for typical structural nodes, general intersection analysis for all elements, and processing of semantic information from the IFC schema. The verification of the method was performed on three reference models from the KIT library: "AC20-FZK-Haus" (40 vertices), "AC20-Institute-Var-2" (448 vertices), and "AC20-Smile-West-10-Bldg" (831 vertices). Quantitative assessment showed full compliance with the criteria for identifying vertices and structural edges for all models. Qualitative visual inspection using the "AC20-FZK-Haus" model confirmed that the method correctly distinguishes between load-bearing interactions and spatial adjacency links. The developed method serves as a methodological toolkit for the automated creation of physics-based graphs from BIM models, which form the basis for developing graph neural networks (GNN) capable of predicting the propagation of degradation processes. This opens prospects for implementing proactive safety and life-cycle management systems for construction assets within modern information technology platforms. Directions for further research include the development of mathematical models to determine the physical state of structural elements depending on time and operating conditions, as well as the creation of GNN architectures for predictive analysis.

Keywords: graph neural network; physics-based graph; data transformation; BIM model; IFC file

Посилання на публікацію

- | | |
|------|---|
| APA | Solovei, O., & Honcharenko, T. (2025). Method for creating graphs based on BIM models for predictive monitoring of buildings. <i>Management of Development of Complex Systems</i> , 64, 247–258, dx.doi.org/10.32347/2412-9933.2025.64.247-258 . |
| ДСТУ | Соловей О. Л., Гончаренко Т. А. Метод створення графів на основі ВІМ-моделей для предиктивного моніторингу будівель. <i>Управління розвитком складних систем</i> . Київ, 2025. № 64. С. 247 – 258, dx.doi.org/10.32347/2412-9933.2025.64.247-258 . |