

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Кваліфікаційна наукова праця  
на правах рукопису

**БІЛИЙ ВІТАЛІЙ ВІКТОРОВИЧ**

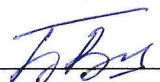
УДК 004.9:519.816:69.059.25

**ДИСЕРТАЦІЯ**

**ПІДТРИМКА ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДНОВЛЕННЯ  
ПОШКОДЖЕНИХ ОБ'ЄКТІВ ЗА УМОВ НЕВИЗНАЧЕНОСТІ НА  
ОСНОВІ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

124 – Системний аналіз

Подається на здобуття наукового ступеня доктора філософії  
Дисертація містить результати власних досліджень. Використання ідей,  
результатів і текстів інших авторів мають посилання на відповідне джерело

  
\_\_\_\_\_ В.В. Білий

Науковий керівник

Бакурова Анна Володимирівна, доктор економічних наук, професор

Запоріжжя – 2026

## АНОТАЦІЯ

**Білий В.В. Підтримка прийняття рішень щодо відновлення пошкоджених об'єктів за умов невизначеності на основі інтелектуальних технологій.** – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 124 – «Системний аналіз» – Національний університет «Запорізька політехніка», Запоріжжя, 2026.

Дисертаційне дослідження присвячено розробці інтелектуальної інформаційно-аналітичної системи, що забезпечує вирішення науково-практичної проблеми підтримки прийняття рішень щодо відбудови пошкоджених об'єктів за умов невизначеності. Вирішення цієї проблеми пов'язане з подоланням розриву між етапами фіксації пошкоджень, оцінки вартості відновлення та ухвалення рішень про вибір альтернативи відбудови, що виникає внаслідок фрагментарності наявних інструментів та неповноти вхідних даних.

**Об'єктом дослідження** є процеси фіксації пошкоджень та оцінювання вартості відбудови пошкоджених об'єктів.

**Предметом дослідження** є інформаційно-аналітична система для підтримки прийняття рішень про відбудову пошкоджених об'єктів.

**Мета дослідження** полягає у розробці інтелектуальної інформаційно-аналітичної системи на основі веб-технологій, що вирішує науково-практичну проблему підтримки прийняття рішень про відбудову пошкоджених об'єктів за умов невизначеності та містить модулі аналітики, оцінювання вартості, прогнозування та збереження пам'яті культурної спадщини.

Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків.

У вступі обґрунтовано актуальність теми дисертаційного дослідження, наведено мету, об'єкт та предмет дослідження, описано методи дослідження,

сформульовано наукову новизну та практичне значення одержаних результатів, подано відомості про апробацію та публікації за темою дисертації.

У першому розділі «Роль інформаційно-аналітичних систем у підтримці прийняття рішень щодо відновлення пошкоджених об'єктів» проаналізовано актуальність проблеми фіксації руйнувань та відбудови в умовах збройної агресії. Здійснено огляд існуючих ІАС підтримки прийняття рішень, проаналізовано підходи до побудови цифрових двійників та застосування методів машинного навчання для прогнозування вартості відновлення. Обґрунтовано необхідність створення комплексної системи, яка б інтегрувала функції обліку, аналітики, прогнозування та сценарного моделювання.

У другому розділі «Концептуальні основи побудови інформаційно-аналітичної системи підтримки прийняття рішень щодо відновлення пошкоджених об'єктів» розроблено концепцію ІАС на основі гібридної архітектури, що поєднує монолітний веб-застосунок із відокремленим ML-мікросервісом. Формалізовано багатокритеріальну задачу прийняття рішення про відбудову пошкодженого об'єкта як задачу вибору з множини альтернатив в умовах невизначеності з урахуванням належності об'єкта до культурної спадщини. Формалізовано задачу прогнозування вартості відновлення як регресійну задачу машинного навчання із застосуванням комбінованого кодування категоріальних ознак.

У третьому розділі «Розробка аналітичного та прогнозного забезпечення підтримки прийняття рішень щодо відновлення пошкоджених об'єктів у межах інформаційно-аналітичної системи» описано реалізацію ІАС, що охоплює модулі обліку пошкоджених об'єктів, інтерактивну карту з геопросторовою візуалізацією, підсистему OLAP-аналітики та модуль віртуальних турів для збереження пам'яті про об'єкти культурної спадщини. Розроблено цифровий двійник пошкодженого об'єкта, що поєднує цифровий профіль з прогностичним шаром на основі моделі XGBoost та механізмом сценарного (what-if) перерахунку вартості відновлення з SHAP-інтерпретацією результатів.

У четвертому розділі «Тестування системи та експериментальна верифікація моделей прогнозування» проведено порівняльний аналіз моделей машинного навчання (Linear Regression, Random Forest, XGBoost) на даних публічних закупівель системи Prozorro. Проведено тестування системи на трьох рівнях: модульне тестування серверної частини, модульне тестування ML-мікросервісу та інтеграційне тестування взаємодії між ними.

Для досягнення поставленої мети необхідно було вирішити такі **завдання**:

1. Проаналізувати існуючі підходи до фіксації руйнувань, прогнозування вартості відновлення, побудови цифрових двійників пошкоджених об'єктів та обґрунтувати необхідність створення комплексної інформаційно-аналітичної системи підтримки прийняття рішень щодо відбудови (розділ 1).

2. Розробити концепцію інтелектуальної інформаційно-аналітичної системи підтримки прийняття рішень, що об'єднує підсистеми фіксації пошкоджень, багатовимірної аналітики, прогнозування вартості та сценарного моделювання в межах спільної архітектури (розділ 2).

3. Формалізувати багатокритеріальну задачу прийняття рішення про відбудову пошкоджених об'єктів різної природи за умов невизначеності (розділ 2).

4. Формалізувати задачу прогнозування вартості відновлення пошкоджених об'єктів як регресійну задачу машинного навчання та обґрунтувати вибір методів кодування категоріальних ознак і найбільш точної моделі прогнозування (розділ 2).

5. Розробити цифровий двійник пошкодженого об'єкта з підтримкою сценарного перерахунку вартості відновлення та інтерпретацією результатів прогнозування на основі SHAP-аналізу (розділ 3).

6. Провести експериментальну верифікацію розробленої системи та оцінити точність моделей прогнозування на даних публічних закупівель (розділ 4).

### **Наукова новизна роботи.**

- Вперше запропоновано концепцію інформаційно-аналітичної системи підтримки прийняття рішень щодо відбудови на основі гібридної архітектури, в якій монолітний веб-застосунок забезпечує функції фіксації пошкоджень та багатовимірної OLAP-аналітики, а також відокремлений ML-мікросервіс – прогнозування вартості та сценарне моделювання, що дозволяє незалежно оновлювати прогнозні моделі без зупинки основної системи та забезпечує наскрізний цикл підтримки прийняття рішень щодо відбудови в умовах неповноти та невизначеності вхідних даних.

- Вперше розроблено цифровий двійник пошкодженого об'єкта як інструмент підтримки прийняття рішень в умовах невизначеності, що поєднує цифровий профіль об'єкта (паспортні характеристики, ступінь пошкодження, фотофакти) з прогностичним шаром на основі моделі XGBoost та механізмом сценарного (what-if) перерахунку вартості відновлення з SHAP-інтерпретацією результатів, що дозволяє порівнювати альтернативні варіанти відбудови та оцінювати вплив окремих характеристик об'єкта на прогнозовану вартість.

- Набув подальшого розвитку підхід до підтримки прийняття рішень щодо відбудови пошкоджених об'єктів за рахунок формалізації багатокритеріальної задачі вибору альтернативи відновлення (капітальний ремонт, поточний ремонт, реставрація, знесення та нове будівництво, консервація) в умовах невизначеності з урахуванням належності об'єкта до культурної спадщини та використанням критеріїв оцінки на основі даних ІАС та механізму сценарного перерахунку.

- Набув подальшого розвитку підхід до прогнозування вартості відновлення пошкоджених об'єктів за рахунок формалізації цієї задачі як регресійної задачі машинного навчання із застосуванням комбінованого кодування категоріальних ознак: one-hot кодування для номінальних змінних та фазифікації з трикутними функціями належності для порядкових змінних, що

дозволяє коректно передати внутрішню градацію ознак і підвищити точність прогнозування.

- Набув подальшого розвитку підхід до експериментальної верифікації розробленої системи та оцінки точності моделей прогнозування за рахунок порівняльного аналізу ансамблевих моделей машинного навчання на даних публічних закупівель системи Prozorro.

**Практична значимість отриманих результатів.** Розроблено діючу веб-систему, що забезпечує: а) реєстрацію пошкоджених об'єктів із фотофактами; б) візуалізацію руйнувань на інтерактивній карті з трьома рівнями деталізації та кольоровою градацією ступеня пошкоджень; в) багатовимірний OLAP-аналіз за вимірами часу, категорії, типу об'єкта та географії; г) автоматичне формування орієнтовної вартості відновлення через ML-мікросервіс за характеристиками об'єкта; д) порівняння альтернативних сценаріїв відбудови із SHAR-інтерпретацією внесків ознак та інфляційним коригуванням кошторису. Це дозволяє скоротити час попередньої оцінки вартості відновлення, виявляти потенційно завищені суми у заявках на відбудову на ранніх етапах та підвищити обґрунтованість розподілу обмежених фінансових ресурсів між об'єктами.

Наукові положення та рекомендації, викладені в дисертації, використано при розробці матеріалів лабораторних робіт з дисципліни «Інтелектуальні системи підтримки прийняття рішень» для здобувачів другого (магістерського) рівня вищої освіти на кафедрі системного аналізу та обчислювальної математики Національного університету «Запорізька політехніка» (акт впровадження від 28.04.2026). Результати дисертаційної роботи апробовано та впроваджено у діяльність Запорізької районної ради Запорізької області.

**Ключові слова:** підтримка прийняття рішень, невизначеність, багатокритеріальна задача вибору, системний аналіз, інформаційно-аналітична система, математичне та комп'ютерне моделювання, машинне навчання, нечітка логіка, прогнозування вартості, експертні оцінки, відбудова, культурна спадщина, цифровий двійник.

## ABSTRACT

**Bilyi V.V. Decision support for the restoration of damaged objects under uncertainty based on intelligent technologies.** – Qualifying scientific work with manuscript rights.

Dissertation for the Doctor of Philosophy degree in specialty 124 – «System Analysis» – National University « Zaporizhzhia Polytechnic», Zaporizhzhia, 2026.

The dissertation research is devoted to the development of an intelligent information-analytical system that addresses the scientific and practical problem of decision support for the restoration of damaged objects under uncertainty. Solving this problem involves bridging the gap between the stages of damage recording, restoration cost estimation, and decision-making regarding the choice of restoration alternative, which arises due to the fragmentation of existing tools and the incompleteness of input data.

The **object of the research** is the processes of damage recording and restoration cost estimation for damaged objects.

The **subject of the research** is the information-analytical system for decision support regarding the restoration of damaged objects.

The **aim of the research** is to develop an intelligent web-based information-analytical system that addresses the scientific and practical problem of decision support for the restoration of damaged objects under uncertainty and includes modules for analytics, cost estimation, forecasting, and cultural heritage memory preservation.

The dissertation consists of an introduction, four chapters, conclusions, a list of references, and appendices.

The introduction substantiates the relevance of the dissertation research topic, presents the aim, object, and subject of the research, describes the research methods, formulates the scientific novelty and practical significance of the obtained results, and provides information on approbation and publications related to the dissertation topic.

Chapter 1, "The role of information-analytical systems in decision support for the restoration of damaged objects," analyzes the relevance of the problem of damage recording and reconstruction under conditions of armed aggression. A review of existing IAS is conducted, approaches to building digital twins and applying machine learning methods for restoration cost forecasting are analyzed. The necessity of creating a unified system integrating the functions of recording, analytics, forecasting, and scenario modeling is substantiated.

Chapter 2, "Conceptual foundations for building an information-analytical decision support system for the restoration of damaged objects," develops the concept of the IAS based on a hybrid architecture combining a monolithic web application with a separate ML microservice. The decision-making task regarding the restoration of a damaged object is formalized as a selection problem from a set of alternatives under uncertainty, taking into account the object's cultural heritage status. The restoration cost forecasting task is formalized as a regression machine learning problem using combined encoding of categorical features: one-hot encoding for nominal variables and fuzzification with triangular membership functions for ordinal variables.

Chapter 3, "Development of analytical and predictive support for decision-making on the restoration of damaged objects within the information-analytical system," describes the implementation of the IAS, covering modules for recording damaged objects, an interactive map with geospatial visualization, an OLAP analytics subsystem, and a virtual tours module for preserving the memory of cultural heritage objects. A digital twin of a damaged object is developed, combining a digital profile with a predictive layer based on the XGBoost model and a scenario (what-if) restoration cost recalculation mechanism with SHAP interpretation of results.

Chapter 4, "System testing and experimental verification of forecasting models," presents a comparative analysis of machine learning models (Linear Regression, Random Forest, XGBoost) using public procurement data from the Prozorro system. The system was tested at three levels: unit testing of the backend, unit testing of the ML microservice, and integration testing of the interaction between them.

To achieve the stated aim, the following **tasks** had to be solved:

1. Analyze existing approaches to damage recording, restoration cost forecasting, and building digital twins of damaged objects, and substantiate the necessity of creating a comprehensive information-analytical decision support system for restoration (Chapter 1).

2. Develop the concept of an intelligent information-analytical decision support system that integrates the subsystems of damage recording, multidimensional analytics, cost forecasting, and scenario modeling within a shared information environment (Chapter 2).

3. Formalize the multi-criteria decision-making task regarding the restoration of damaged objects of various natures under uncertainty (Chapter 2).

4. Formalize the restoration cost forecasting task for damaged objects as a regression machine learning problem and substantiate the choice of categorical feature encoding methods and the forecasting model (Chapter 2).

5. Develop a digital twin of a damaged object with support for scenario-based restoration cost recalculation and interpretation of forecasting results based on SHAP analysis (Chapter 3).

6. Conduct experimental verification of the developed system and evaluate the accuracy of forecasting models using public procurement data (Chapter 4).

**Scientific novelty of the work:**

- For the first time, the concept of an information-analytical decision support system for restoration based on a hybrid architecture is proposed, in which a monolithic web application provides damage recording and multidimensional OLAP analytics functions, while a separate ML microservice provides cost forecasting and scenario modeling, enabling independent updating of predictive models without stopping the main system and ensuring an end-to-end decision support cycle for restoration under conditions of incomplete and uncertain input data.

- For the first time, a digital twin of a damaged object has been developed as a decision support tool under uncertainty, combining a digital profile of the object

(passport characteristics, degree of damage, photographic evidence) with a predictive layer based on the XGBoost model and a scenario (what-if) restoration cost recalculation mechanism with SHAP interpretation of results, enabling comparison of alternative restoration options and assessment of the impact of individual object characteristics on the predicted cost.

- The approach to decision support for the restoration of damaged objects has been further developed through the formalization of the multi-criteria task of selecting a restoration alternative (capital repair, current repair, restoration, demolition and new construction, conservation) under uncertainty, taking into account the object's cultural heritage status and using evaluation criteria based on IAS data and a scenario recalculation mechanism.

- The approach to forecasting the restoration cost of damaged objects has been further developed through the formalization of this task as a regression machine learning problem using combined encoding of categorical features: one-hot encoding for nominal variables and fuzzification with triangular membership functions for ordinal variables, which allows correctly conveying the internal gradation of features and improving forecasting accuracy.

- The approach to experimental verification of the developed system and evaluation of forecasting model accuracy has been further developed through a comparative analysis of ensemble machine learning models using public procurement data from the Prozorro system.

**Practical significance of the obtained results.** A functioning web system has been developed that provides: (a) registration of damaged objects with photographic evidence; (b) visualization of destruction on an interactive map with three levels of detail and color-coded damage grading; (c) multidimensional OLAP analysis across dimensions of time, category, object type, and geography; (d) automatic generation of an approximate restoration cost via an ML microservice based on object characteristics; (e) comparison of alternative restoration scenarios with SHAP interpretation of feature contributions and inflation adjustment of the cost estimate. This enables reducing the

time for preliminary restoration cost estimation, identifying potentially inflated amounts in restoration applications at early stages, and improving the substantiation of limited financial resource allocation among objects.

The scientific provisions and recommendations presented in the dissertation were used in the preparation of laboratory work materials for the discipline «Intelligent Decision Support Systems» for Master's students at the Department of System Analysis and Computational Mathematics of Zaporizhzhia Polytechnic National University. The results of the dissertation have been tested and implemented in the activities of the Zaporizhzhia District Council of Zaporizhzhia Oblast.

**Keywords:** decision support, uncertainty, multi-criteria decision-making, systems analysis, information-analytical system, mathematical and computer modeling, machine learning, fuzzy logic, cost prediction, expert evaluations, reconstruction, cultural heritage, digital twin.

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

**1. Наукові праці, в яких опубліковано основні наукові результати дисертації:**

**1.1. Статті у наукових фахових виданнях України:**

1. Бакурова А. В., Білий В. В. Розробка інформаційно-аналітичної системи для фіксації руйнувань внаслідок російської агресії. *Комунальне господарство міст*. 2025. Т. 3, Вип. 191. С. 2–10. DOI: <https://doi.org/10.33042/2522-1809-2025-3-191-2-10>

2. Бакурова А. В., Білий В. В. Сценарний розрахунок вартості відбудови пошкодженого об'єкта інфраструктури: підхід на основі цифрового двійника. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2026. № 363(2). С. 324–334. DOI: <https://doi.org/10.31891/2307-5732-2026-363-45>

**1.2. Статті у наукових фахових виданнях України, які входять до міжнародних наукометричних баз:**

3. Bakurova A., Bilyi V. Development of an approach for predicting the cost of damaged infrastructure recovery with microservice implementation. *Technology Audit and Production Reserves*. 2025. Vol. 5, No. 2(85). P. 33–39. DOI: <https://doi.org/10.15587/2706-5448.2025.339773> (Індексовано в міжнародній НМБД Scopus)

**2. Наукові праці, які засвідчують апробацію матеріалів дисертації:**

**2.1. Матеріали міжнародних наукових та науково-практичних конференцій, збірники яких входять до міжнародних наукометричних баз:**

4. Bilyi V. V., Bakurova A. V., Tereschenko E. V., Didenko A. V. Analytics Module for the System for Recording Destruction Due to Russian Aggression. *European Association of Geoscientists & Engineers: 17th International Conference Monitoring of Geological Processes and Ecological Condition of the Environment*, Nov. 2023. 2023. P. 1–5. DOI: <https://doi.org/10.3997/2214-4609.2023520232> (Індексовано в міжнародній НМБД Scopus)

## **2.2. Матеріали та тези доповідей наукових та науково-практичних конференцій:**

5. Білий В. В. Система фіксації руйнувань внаслідок російської агресії. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій* : тези доповідей XI Міжнародної науково-практичної конференції, Запоріжжя, 12–14 грудня 2022. Запоріжжя : НУ «Запорізька політехніка», 2022. С. 132.

6. Білий В. В., Бакурова А. В. Організація аналізу багатовимірних даних в системі фіксації руйнувань внаслідок російської агресії. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій* : тези доповідей XII Міжнародної науково-практичної конференції, Запоріжжя, 10–12 грудня 2024. Запоріжжя : НУ «Запорізька політехніка», 2024. С. 426.

7. Білий В. В., Бакурова А. В. Огляд підходів побудови та приклад реалізації інформаційно-аналітичної системи. *Тиждень науки-2025. Факультет комп'ютерних наук і технологій* : тези доповідей науково-практичної конференції, Запоріжжя, 14–18 квітня 2025. Запоріжжя : НУ «Запорізька політехніка», 2025. С. 403.

8. Білий В. В. Формалізація задачі прийняття рішення щодо відбудови пошкоджених об'єктів інфраструктури в умовах невизначеності. *Інформаційні технології: теорія і практика* : тези доповідей III (IX) Міжнародної науково-практичної конференції, Харків – Запоріжжя – Дніпро, 25–27 березня 2026. С. 37.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	17
ВСТУП.....	19
1. РОЛЬ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ У ПІДТРИМЦІ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДНОВЛЕННЯ ПОШКОДЖЕНИХ ОБ’ЄКТІВ .....	27
1.1 Актуальність дослідження .....	27
1.2 Інформаційно-аналітичні системи підтримки прийняття рішень .....	27
1.3 Нормативно-правове забезпечення фіксації пошкоджень та відновлення об’єктів .....	31
1.4 Аналіз існуючих підходів та програмних рішень .....	35
1.4.1 Платформи фіксації та візуалізації руйнувань .....	35
1.4.2 Підходи до прогнозування вартості відновлення .....	40
1.4.3 Технології цифрових двійників для інфраструктурних об’єктів .....	42
1.5 Висновки за розділом 1 .....	44
2. КОНЦЕПТУАЛЬНІ ОСНОВИ ПОБУДОВИ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДНОВЛЕННЯ ПОШКОДЖЕНИХ ОБ’ЄКТІВ .....	46
2.1 Формалізація задачі прийняття рішень щодо відновлення об’єктів інфраструктури .....	46
2.2 Формалізація задачі прогнозування вартості відновлення .....	50
2.3 Інфляційне коригування прогнозованої вартості відновлення .....	54
2.4 Методи кодування категоріальних ознак .....	56
2.5 Ансамблеві моделі машинного навчання.....	58
2.6 Інтерпретація прогнозів на основі SHAP-аналізу.....	60
2.7 Концепція цифрового двійника зі сценарним аналізом .....	64
2.8 Висновки за розділом 2.....	67

3. РОЗРОБКА АНАЛІТИЧНОГО ТА ПРОГНОЗНОГО ЗАБЕЗПЕЧЕННЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДНОВЛЕННЯ ПОШКОДЖЕНИХ ОБ'ЄКТІВ У МЕЖАХ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ .....	69
3.1 Принципи побудови веб-орієнтованих інформаційних систем.....	69
3.2 Вибір фреймворків для розробки .....	72
3.3 Архітектура системи .....	74
3.4 Проектування REST API.....	80
3.5 Проектування бази даних .....	83
3.6 Аналітичний блок системи.....	87
3.6.1 Інтерактивна карта руйнувань .....	87
3.6.2 Графічна аналітика.....	88
3.6.3 OLAP-куб .....	89
3.7 Облік пошкоджених об'єктів .....	91
3.8 Цифровий двійник об'єкта.....	94
3.8.1 Цифровий профіль об'єкта.....	94
3.8.2 Сценарний розрахунок та інтерпретація прогнозів.....	96
3.8.3 Інфляційне коригування вартості відновлення в інтерфейсі цифрового двійника.....	99
3.9 Віртуальні тури.....	101
3.10 Висновки за розділом 3.....	104
4. ТЕСТУВАННЯ СИСТЕМИ ТА ЕКСПЕРИМЕНТАЛЬНА ВЕРИФІКАЦІЯ МОДЕЛЕЙ ПРОГНОЗУВАННЯ.....	106
4.1 Налаштування веб-додатку.....	106
4.1.1 Конфігурація середовища.....	106
4.1.2 Налаштування підключення до бази даних.....	107
4.1.3 Міграції бази даних.....	108
4.1.4 Налаштування аутентифікації та авторизації .....	111
4.1.5 Налаштування взаємодії з ML-мікросервісом .....	112

4.1.6 Налаштування CORS та маршрутизації API .....	113
4.2 Формування навчальної вибірки.....	114
4.3 Результати порівняння моделей прогнозування.....	117
4.4 Тестування модулів системи .....	120
4.4.1 Модульне тестування серверної частини.....	120
4.4.2 Модульне тестування ML-мікросервісу .....	122
4.4.3 Інтеграційне тестування .....	124
4.5 Обмеження та перспективи розвитку.....	126
4.6 Висновки за розділом 4.....	128
ВИСНОВКИ.....	130
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	133
ДОДАТОК А. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації .....	141
ДОДАТОК Б. Акти про використання та впровадження результатів дисертаційної роботи .....	144
ДОДАТОК В. Фрагменти коду розроблених алгоритмів прогнозування, інтерпретації та сценарного аналізу вартості відновлення.....	146

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API (Application Programming Interface) – програмний інтерфейс взаємодії між компонентами системи

BIM (Building Information Modeling) – інформаційне моделювання будівель

БД – база даних

IAC – інформаційно-аналітична система

CORS (Cross-Origin Resource Sharing) – механізм спільного використання ресурсів між різними доменами

CRUD (Create, Read, Update, Delete) – базові операції створення, зчитування, оновлення та видалення даних

CSV (Comma-Separated Values) – формат зберігання табличних даних у текстовому файлі

DT (Digital Twin) – цифровий двійник

HTTP (HyperText Transfer Protocol) – протокол передачі гіпертексту

JSON (JavaScript Object Notation) – текстовий формат обміну даними

MAE (Mean Absolute Error) – середня абсолютна похибка

MAPE (Mean Absolute Percentage Error) – середня абсолютна відсоткова похибка

ML (Machine Learning) – машинне навчання

MPA (Multi-Page Application) – багатосторінковий веб-застосунок

OLAP (Online Analytical Processing) – оперативна аналітична обробка даних

ОПР – особа, що приймає рішення

ORM (Object-Relational Mapping) – об'єктно-реляційне відображення

RBAC (Role-Based Access Control) – розмежування доступу на основі ролей

REST (Representational State Transfer) – архітектурний стиль взаємодії компонентів розподіленої системи

RMSE (Root Mean Square Error) – середньоквадратична похибка

SHAP (SHapley Additive exPlanations) – метод інтерпретації прогнозів на основі значень Шеплі

SPA (Single-Page Application) – односторінковий веб-застосунок

SQL (Structured Query Language) – мова структурованих запитів

СУБД – система управління базами даних

TreeSHAP – алгоритм обчислення SHAP-значень для деревоподібних моделей

XGBoost (eXtreme Gradient Boosting) – алгоритм градієнтного бустингу на деревах рішень

## ВСТУП

**Актуальність теми дисертації.** Повномасштабна збройна агресія Російської Федерації проти України призвела до безпрецедентних руйнувань інфраструктури: знищено та пошкоджено десятки міст, тисячі житлових будинків, об'єктів соціального призначення, промислових підприємств. Масштаб руйнувань ставить перед органами влади та міжнародними донорами комплексну задачу: необхідно одночасно зафіксувати збитки, оцінити вартість відновлення та ухвалити обґрунтовані рішення щодо пріоритетності відбудови в умовах обмежених фінансових ресурсів і значної невизначеності.

Сучасні дослідження у галузі системного аналізу та підтримки прийняття рішень засвідчують зростаючий інтерес до застосування інтелектуальних технологій – машинного навчання, цифрових двійників, сценарного моделювання – для задач управління складними інфраструктурними системами. Водночас критичний аналіз існуючих рішень виявляє їхню фрагментарність: платформи фіксації руйнувань (Russia Will Pay, Україна Recovery Plan, картографічні проєкти Bellingcat) виконують переважно функції обліку та візуалізації даних, але не забезпечують аналітичної підтримки для порівняння альтернатив відновлення. Підходи до прогнозування вартості будівельних робіт, описані в літературі, зосереджені здебільшого на нормальних умовах будівництва та не адаптовані до специфіки відбудови пошкоджених об'єктів із притаманною їй невизначеністю. Технології цифрових двійників активно розвиваються у сфері моніторингу технічного стану та управління життєвим циклом об'єктів, проте їх застосування для прогнозування вартості відновлення з підтримкою сценарного перерахунку залишається практично не дослідженим.

Таким чином, існує науково-практична проблема: відсутність інтегрованого інструменту, який поєднував би реєстрацію пошкоджень, багатовимірну аналітику, прогнозування вартості відновлення та сценарне моделювання в єдиній системі підтримки прийняття рішень. Розв'язання цієї

проблеми потребує формалізації задачі вибору способу відновлення як задачі прийняття рішень в умовах невизначеності, розробки відповідних моделей та інформаційно-аналітичної системи, що їх реалізує. Це й зумовлює актуальність дисертаційного дослідження.

### **Зв'язок роботи з науковими програмами, планами, темами.**

Дисертаційне дослідження виконане у контексті державної політики фіксації пошкоджень та відновлення інфраструктури, визначеної Постановою Кабінету Міністрів України від 20 березня 2022 р. № 326 «Про затвердження Порядку визначення шкоди та збитків, завданих Україні внаслідок збройної агресії Російської Федерації», Постановою КМУ від 26 березня 2022 р. № 380 «Про збір, обробку та облік інформації про пошкоджене та знищене нерухоме майно внаслідок бойових дій, терористичних актів, диверсій, спричинених військовою агресією Російської Федерації», Законом України від 23 лютого 2023 р. № 2923-ІХ «Про компенсацію за пошкодження та знищення окремих категорій об'єктів нерухомого майна внаслідок бойових дій, терористичних актів, диверсій, спричинених збройною агресією Російської Федерації проти України» та Законом України «Про охорону культурної спадщини» (№ 1805-III), вимоги якого щодо обмеження видів робіт на об'єктах спадщини безпосередньо враховано у формалізації задачі прийняття рішень.

Робота виконана відповідно до планів науково-дослідних робіт (НДР) Національного університету «Запорізька політехніка» у межах тем: «Математичне моделювання соціально-економічних процесів та систем» (номер реєстрації – 05038), «Розвиток методів дослідження складних соціально-економічних систем на основі інтелектуальних технологій» (номер державної реєстрації – 0121U113264), а також «Прийняття ефективних рішень на основі інтелектуальних технологій та відкритих даних» (номер реєстрації – 05014). У зазначених НДР здобувач брав участь як виконавець, розробив інформаційно-аналітичну систему фіксації руйнувань, ML-мікросервіс прогнозування вартості

відновлення та цифровий двійник зі сценарним перерахунком для підтримки прийняття рішень щодо відбудови пошкоджених об'єктів за умов невизначеності.

**Об'єкт дослідження** – процеси фіксації пошкоджень та оцінювання вартості відбудови пошкоджених об'єктів.

**Предмет дослідження** – інформаційно-аналітична система для підтримки прийняття рішень про відбудову пошкоджених об'єктів.

**Мета дослідження** – розробити інтелектуальну інформаційно-аналітичну систему на основі веб-технологій, що вирішує науково-практичну проблему підтримки прийняття рішень про відбудову пошкоджених об'єктів за умов невизначеності та містить модулі аналітики, оцінювання вартості, прогнозування та збереження пам'яті культурної спадщини.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

1. Проаналізувати існуючі підходи до фіксації руйнувань, прогнозування вартості відновлення, побудови цифрових двійників пошкоджених об'єктів та обґрунтувати необхідність створення комплексної інформаційно-аналітичної системи підтримки прийняття рішень щодо відбудови (розділ 1).

2. Розробити концепцію інтелектуальної інформаційно-аналітичної системи підтримки прийняття рішень, що об'єднує підсистеми фіксації пошкоджень, багатовимірної аналітики, прогнозування вартості та сценарного моделювання в межах спільної архітектури (розділ 2).

3. Формалізувати багатокритеріальну задачу прийняття рішення про відбудову пошкоджених об'єктів різної природи за умов невизначеності (розділ 2).

4. Формалізувати задачу прогнозування вартості відновлення пошкоджених об'єктів як регресійну задачу машинного навчання та обґрунтувати вибір методів кодування категоріальних ознак і найбільш точної моделі прогнозування (розділ 2).

5. Розробити цифровий двійник пошкодженого об'єкта з підтримкою сценарного перерахунку вартості відновлення та інтерпретацією результатів прогнозування на основі SHAP-аналізу (розділ 3).

6. Провести експериментальну верифікацію розробленої системи та оцінити точність моделей прогнозування на даних публічних закупівель (розділ 4).

**Методи дослідження.** Для вирішення поставлених задач у роботі використано:

- методи теорії прийняття рішень – для формалізації задачі вибору способу відновлення пошкодженого об'єкта як задачі вибору з множини альтернатив в умовах невизначеності з урахуванням обмежень, зумовлених статусом культурної спадщини;

- методи машинного навчання – для побудови ансамблевих регресійних моделей (XGBoost, Random Forest) прогнозування вартості відновлення на основі структурованих даних публічних закупівель системи Prozorro;

- методи фазифікації – для коректного представлення порядкових категоріальних ознак (ступінь пошкодження, тип ремонту) із застосуванням трикутних функцій належності, що зберігає інформацію про внутрішню градацію ознак;

- методи інтерпретації моделей машинного навчання – зокрема SHAP-аналіз (SHapley Additive exPlanations) та його оптимізовану версію TreeSHAP – для визначення внеску окремих характеристик об'єкта у прогнозовану вартість відновлення;

- методи архітектурного проєктування програмних систем – для побудови гібридної веб-архітектури, що поєднує тришарову структуру основного додатку з мікросервісною організацією прогнозного модуля.

**Наукова новизна отриманих результатів** полягає у розробці інформаційно-аналітичної системи підтримки прийняття рішень щодо відновлення пошкоджених об'єктів інфраструктури за умов невизначеності на

основі інтелектуальних технологій. Основні результати, що визначають наукову новизну:

- Вперше запропоновано концепцію інформаційно-аналітичної системи підтримки прийняття рішень щодо відбудови на основі гібридної архітектури, в якій монолітний веб-застосунок забезпечує функції фіксації пошкоджень та багатовимірної OLAP-аналітики, а також відокремлений ML-мікросервіс – прогнозування вартості та сценарне моделювання, що дозволяє незалежно оновлювати прогнозні моделі без зупинки основної системи та забезпечує наскрізний цикл підтримки прийняття рішень щодо відбудови в умовах неповноти та невизначеності вхідних даних.

- Вперше розроблено цифровий двійник пошкодженого об'єкта як інструмент підтримки прийняття рішень в умовах невизначеності, що поєднує цифровий профіль об'єкта (паспортні характеристики, ступінь пошкодження, фотофакти) з прогностичним шаром на основі моделі XGBoost та механізмом сценарного (what-if) перерахунку вартості відновлення з SHAP-інтерпретацією результатів, що дозволяє порівнювати альтернативні варіанти відбудови та оцінювати вплив окремих характеристик об'єкта на прогнозовану вартість.

- Набув подальшого розвитку підхід до підтримки прийняття рішень щодо відбудови пошкоджених об'єктів за рахунок формалізації багатокритеріальної задачі вибору альтернативи відновлення (капітальний ремонт, поточний ремонт, реставрація, знесення та нове будівництво, консервація) в умовах невизначеності з урахуванням належності об'єкта до культурної спадщини та використанням критеріїв оцінки на основі даних ІАС та механізму сценарного перерахунку.

- Набув подальшого розвитку підхід до прогнозування вартості відновлення пошкоджених об'єктів за рахунок формалізації цієї задачі як регресійної задачі машинного навчання із застосуванням комбінованого кодування категоріальних ознак: one-hot кодування для номінальних змінних та фазифікації з трикутними функціями належності для порядкових змінних, що

дозволяє коректно передати внутрішню градацію ознак і підвищити точність прогнозування.

- Набув подальшого розвитку підхід до експериментальної верифікації розробленої системи та оцінки точності моделей прогнозування за рахунок порівняльного аналізу ансамблевих моделей машинного навчання на даних публічних закупівель системи Prozorro.

**Практичне значення отриманих результатів.** У ході дисертаційного дослідження створено діючу веб-систему, яка реалізує наскрізний цикл роботи з пошкодженими об'єктами – від первинної реєстрації із фотофактами та геопросторовою прив'язкою до формування прогнозної оцінки вартості відновлення та сценарного порівняння альтернатив відбудови. Система забезпечує відображення масштабів руйнувань на інтерактивній карті з кількома рівнями деталізації, багатовимірний аналіз даних за ключовими вимірами (час, географія, категорія та тип об'єкта, ступінь пошкоджень), а також автоматизоване прогнозування вартості з інтерпретацією факторів впливу та інфляційним коригуванням. Практична цінність отриманих результатів полягає у скороченні часу попередньої кошторисної оцінки, можливості раннього виявлення потенційно завищених сум у заявках на відбудову та підвищенні обґрунтованості рішень щодо розподілу обмежених фінансових ресурсів між об'єктами.

Окремо реалізований модуль віртуальних турів забезпечує документування стану зруйнованих об'єктів культурної спадщини у форматі інтерактивної фотогалереї з аудіосупроводом та історичними довідками, що дозволяє привернути громадську увагу до наслідків війни та сприяє збереженню пам'яті про втрачені об'єкти.

Наукові положення та рекомендації, викладені в дисертації, використано при розробці матеріалів лабораторних робіт з дисципліни «Інтелектуальні системи підтримки прийняття рішень» для здобувачів другого (магістерського) рівня вищої освіти на кафедрі системного аналізу та обчислювальної математики Національного університету «Запорізька політехніка» (акт впровадження від

28.04.2026). Результати дисертаційної роботи апробовано та впроваджено у діяльність Запорізької районної ради Запорізької області.

**Особистий внесок здобувача.** Усі основні наукові результати дисертаційної роботи отримано автором особисто. У роботах, опублікованих зі співавторами, здобувачеві належать такі результати: [14] – розробка архітектури інформаційно-аналітичної системи, реалізація модулів обліку, аналітики та інтерактивної карти; [62] – формалізація задачі прогнозування вартості, побудова та порівняння регресійних моделей, реалізація ML-мікросервісу; [46] – розробка цифрового двійника, реалізація модуля сценарного перерахунку та SHAR-інтерпретації. Науковому керівнику належить постановка загальної проблеми дослідження та консультації щодо методологічних підходів.

**Апробація результатів дисертації.** Основні результати дисертаційної роботи доповідалися й обговорювалися на наукових конференціях: XI Міжнародна науково-практична конференція «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій», 12–14 грудня 2022 р., м. Запоріжжя; науково-популярний захід «Ніч молодіжної науки – 2023 в умовах війни»; 17th International Conference Monitoring of Geological Processes and Ecological Condition of the Environment, Nov 2023; XII Міжнародна науково-практична конференція «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій», 10–12 грудня 2024 р., м. Запоріжжя; International Scientific and Practical Conference «Business Ecosystems: Sustainable Development 2025» (BESD'2025); щорічна науково-практична конференція «Тиждень науки – 2025», НУ «Запорізька політехніка»; III (IX) Міжнародна науково-практична конференція «Інформаційні технології: теорія і практика», 25–27 березня 2026 р., Харків – Запоріжжя – Дніпро.

**Публікації.** Матеріали дисертації викладені у 8 публікаціях, з яких 3 наукові публікації розкривають основний зміст дисертації, зокрема опубліковано 3 статті у наукових виданнях, включених до Переліку наукових фахових видань України, 2 публікації включено у міжнародну наукометричну базу Scopus.

**Структура й обсяг роботи.** Кваліфікаційна наукова праця на здобуття наукового ступеня доктора філософії складається з анотації, переліку умовних позначень, вступу, чотирьох розділів, висновків, списку використаних джерел і додатків. Загальний обсяг дисертаційної роботи з додатками складає 149 сторінок. Основний зміст роботи викладено на 116 сторінках друкованого тексту. Дисертація містить 30 рисунків, 10 таблиць і 3 додатки. Список використаних джерел налічує 62 найменування.

**Використання засобів штучного інтелекту.** Під час підготовки тексту дисертації автором у допустимих межах використано технології штучного інтелекту, зокрема генеративну мовну модель Claude (Anthropic). За допомогою цієї моделі здійснювалося попереднє формулювання окремих текстових фрагментів у вступі, розділах 1–4 та висновках. Застосування ШІ було спрямоване на структурування матеріалу, покращення мовного стилю та логічної послідовності викладу. Усі текстові фрагменти, попередньо сформовані за допомогою ШІ, були ретельно перевірені, змістовно доповнені та відредаговані автором перед включенням до остаточної редакції дисертації.

# **1. РОЛЬ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ У ПІДТРИМЦІ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДНОВЛЕННЯ ПОШКОДЖЕНИХ ОБ'ЄКТІВ**

## **1.1 Актуальність дослідження**

Внаслідок повномасштабної війни, розв'язаної Російською Федерацією проти України, було знищено та пошкоджено десятки міст, зруйновано сотні підприємств, тисячі житлових будинків та об'єктів соціальної інфраструктури. Для подальшої оцінки збитків та їх відшкодування вся ця інформація має бути збережена та систематизована. Перший крок у процесі відновлення – зафіксувати руйнування, а вже потім оцінити збитки та залучити кошти для відбудови.

Водночас нестача спеціалізованих інструментів обмежує спроможність оперативно ухвалювати обґрунтовані фінансові рішення щодо відновлення. Недостатньо дослідженим залишається підхід до прогнозування вартості відбудови та виявлення потенційно завищених сум на ранніх етапах аналізу заявок, що могло б суттєво посилити механізми прозорості та контролю в розподілі ресурсів.

Наявні рішення функціонують фрагментарно: системи збору даних не взаємодіють із прогнозними моделями, а інструменти машинного навчання не мають механізмів сценарного перерахунку та інтерпретації результатів. Це створює розрив між етапами фіксації, аналізу та ухвалення рішень. Відповідно, виникає потреба у комплексній інформаційно-аналітичній системі, яка б поєднувала облік пошкоджених об'єктів, прогнозування вартості відновлення та підтримку сценарного аналізу в межах єдиного інструменту.

## **1.2 Інформаційно-аналітичні системи підтримки прийняття рішень**

Сучасні виклики, пов'язані з кризовими ситуаціями, зокрема повномасштабною війною в Україні, загострили потребу в системному зборі,

обробці та аналізі великого обсягу даних для оперативного реагування, оцінки збитків та планування подальших дій. У таких умовах інформаційно-аналітичні системи (ІАС) відіграють ключову роль у забезпеченні прийняття обґрунтованих рішень, оскільки дозволяють виявляти приховані закономірності в даних, прогнозувати наслідки рішень і на основі цього формувати ефективні стратегії реагування [1].

Інформаційно-аналітична система – це комплекс програмних засобів, призначених для збору, збереження та обробки даних. Вона використовується для аналізу великих обсягів інформації з метою отримання нових знань, формулювання висновків і прийняття рішень на їх основі [2]. Такі системи підвищують ефективність управління, сприяють створенню реалістичних прогнозів щодо тенденцій розвитку ринку та дозволяють оптимізувати внутрішні процеси організацій.

Одним з найважливіших завдань використання ІАС є забезпечення доступу до максимально достовірних, актуальних і оперативних даних. Аналітичні системи також надають широкий спектр інструментів для подальшого аналізу – від класичної статистики до застосування методів машинного навчання, прогнозування трендів і моделювання складних ситуацій.

Використання інформаційно-аналітичних систем охоплює низку важливих функціональних напрямів, серед яких:

- Збір і збереження даних. ІАС дозволяють автоматизовано збирати великі обсяги даних із різних джерел – баз даних, файлів, веб-сервісів тощо. Уся зібрана інформація зберігається у відповідних сховищах, де стає доступною для перегляду, аналізу та подальшого використання уповноваженими працівниками.

- Обробка та аналіз інформації. Системи мають у своєму складі інструменти для багатовимірного аналізу, побудови моделей машинного навчання, прогнозування трендів та моделювання ситуацій. Це дозволяє виявляти патерни, закономірності та приймати обґрунтовані управлінські рішення.

- Візуалізація даних. ІАС забезпечують представлення результатів аналізу у зручному для сприйняття вигляді – за допомогою графіків, діаграм, дашбордів і таблиць. Така візуалізація полегшує розуміння складних зв'язків між показниками та сприяє прозорому інформуванню всіх зацікавлених сторін.

- Підтримка прийняття рішень. Аналітичні інструменти дозволяють оцінювати альтернативні сценарії вирішення проблем та вибирати найоптимальніші з них, що суттєво підвищує ефективність управління.

- Моніторинг і контроль. Завдяки можливості відстеження змін у ключових показниках ІАС сприяють оперативному реагуванню на відхилення та запровадженню коригувальних дій. Це забезпечує гнучкість і адаптивність організацій до змін у середовищі.

Для реалізації вищезазначених функцій інформаційно-аналітичні системи мають складну внутрішню структуру, яка складається з кількох функціональних компонентів:

- Джерела даних. Це внутрішні або зовнішні інформаційні ресурси, з яких система отримує дані. Внутрішні джерела включають корпоративні бази даних, а зовнішні – відкриті чи комерційні сервіси, файли та API, надані іншими організаціями. Важливо, щоб ІАС підтримувала підключення до всіх релевантних джерел, що містять корисну інформацію.

- Системи збору даних. Вони відповідають за вилучення інформації з джерел і передачу її у систему для подальшої обробки. Однією з найпоширеніших технологій є ETL (Extract, Transform, Load), яка дозволяє трансформувати дані у зручний формат та завантажити їх у внутрішнє сховище.

- Сховище даних. Це компонент, де зберігається вся зібрана інформація. Воно може бути реалізоване у вигляді реляційних баз даних, data lake або спеціалізованих аналітичних платформ. Основними вимогами є ефективне управління, безпечне збереження та швидкий доступ до даних.

- Аналітичні інструменти. Вони включають широкий набір засобів для обробки даних: статистичні методи, візуалізація, машинне навчання, штучний

інтелект тощо. Ці інструменти забезпечують можливість виявлення закономірностей, трендів і формування практичних висновків.

- Користувацький інтерфейс. Це елемент, що забезпечує зручний доступ до функцій ІАС для різних категорій користувачів. Інтерфейс може бути реалізований у вигляді веб- або десктопного застосунку і має забезпечувати можливість фільтрації, побудови звітів і виконання аналізу відповідно до прав доступу.

Під час проєктування ІАС важливо дотримуватися ключових принципів, що забезпечують їх ефективне функціонування. До основних принципів належать: інтеграція даних з різноманітних джерел в єдину структуру; автоматизація процесів збору, обробки та аналізу; гнучкість і масштабованість для адаптації до змін у потребах організації; візуалізація даних для полегшення сприйняття та інтерпретації результатів; забезпечення безпеки даних від несанкціонованого доступу та втрати.

Залежно від призначення та функціональних можливостей, ІАС класифікуються за кількома ознаками. За функціональним призначенням виділяють системи підтримки прийняття рішень (DSS), що допомагають у виборі оптимальних рішень у складних ситуаціях через сценарне моделювання та оцінку ризиків; системи бізнес-аналітики (BI), зосереджені на зборі, інтеграції та візуалізації даних про діяльність організації; аналітичні системи з елементами прогнозування, що включають інструменти машинного навчання для побудови прогнозів на майбутнє.

За архітектурою розрізняють монолітні системи, побудовані як єдине програмне рішення, та модульні або сервісно-орієнтовані системи (SOA, мікросервіси), що складаються з незалежних компонентів і легко масштабуються через API. За характером оброблюваних даних виділяють системи для структурованих даних, що працюють з таблицями та реляційними базами, та системи для неструктурованих даних, що обробляють тексти, зображення, геодані із застосуванням методів обробки природної мови та комп'ютерного зору.

У реальних умовах часто використовуються гібридні системи, які поєднують риси кількох типів – наприклад, система, що одночасно виконує функції моніторингу, прогнозування та візуалізації, інтегруючись із зовнішніми державними сервісами та відкритими джерелами.

### **1.3 Нормативно-правове забезпечення фіксації пошкоджень та відновлення об'єктів**

Процеси фіксації пошкоджень та організації відбудови в Україні регулюються низкою нормативно-правових актів, прийнятих переважно після початку повномасштабного вторгнення. Їх аналіз є необхідним для розуміння того, які вимоги до збору, обліку та оцінки даних про пошкоджені об'єкти має враховувати інформаційно-аналітична система підтримки прийняття рішень.

Базовим документом є Постанова Кабінету Міністрів України від 20 березня 2022 р. № 326 «Про затвердження Порядку визначення шкоди та збитків, завданих Україні внаслідок збройної агресії Російської Федерації» [3]. Цей документ окреслив поняття «шкоди» та «збитків», встановив класифікацію категорій пошкоджень і визначив порядок подання інформації до відповідних органів. Фактично Постанова № 326 заклала фундамент для систематичного обліку наслідків руйнувань на державному рівні.

Механізм збору первинних даних було конкретизовано Постановою КМУ від 26 березня 2022 р. № 380 «Про збір, обробку та облік інформації про пошкоджене та знищене нерухоме майно внаслідок бойових дій, терористичних актів, диверсій, спричинених військовою агресією Російської Федерації». Вона затвердила порядок подання інформаційних повідомлень про пошкоджене майно та визначила процедуру обстеження об'єктів органами місцевого самоврядування [4].

Ключовим законодавчим актом у сфері компенсації став Закон України від 23 лютого 2023 р. № 2923-IX «Про компенсацію за пошкодження та знищення

окремих категорій об'єктів нерухомого майна внаслідок бойових дій, терористичних актів, диверсій, спричинених збройною агресією Російської Федерації проти України». Закон визначив правові засади надання компенсації, механізми її фінансування (державний та місцеві бюджети, міжнародна допомога, репарації), а також коло осіб, які мають право на її отримання [5]. Порядок практичної реалізації компенсаційного механізму деталізовано у Постанові КМУ від 21 квітня 2023 р. № 381.

Окремий нормативний блок стосується визначення ступеня пошкоджень. Методика проведення обстеження та оформлення його результатів, затверджена наказом Мінрегіону від 6 серпня 2022 р., встановлює категорії пошкоджень об'єктів залежно від стану несучих та огорожувальних конструкцій. Ці категорії безпосередньо впливають на вибір типу відновлювальних робіт: від ремонту (за незначних пошкоджень) до визнання об'єкта непридатним для використання (за повного руйнування). У контексті дисертації ця класифікація відповідає ознаці «ступінь пошкодження», яка є однією з ключових вхідних змінних ML-моделі прогнозування вартості відновлення.

Для об'єктів, що мають статус культурної спадщини, діє спеціальний правовий режим. Закон України «Про охорону культурної спадщини» (№ 1805-III, ст. 26) встановлює обмеження щодо видів робіт, дозволених на таких об'єктах [6]. Зокрема, знесення об'єктів національного значення є, як правило, недопустимим, а пріоритетними є реставраційні та консерваційні роботи. Ці обмеження безпосередньо враховано у формалізації задачі прийняття рішень, запропонованій у розділі 2: статус культурної спадщини визначає множину допустимих альтернатив відновлення.

Розглянуті вище нормативні акти регулюють процедури обстеження пошкоджень та надання компенсацій, проте окреме питання становить те, чи можуть зібрані дані бути публічно оприлюднені й за яких умов. Із введенням правового режиму воєнного стану в Україні діє низка обмежень щодо публічного поширення відомостей про конкретні пошкоджені об'єкти – їх точних координат,

дати та обставин ураження, деталізованих фотофіксацій, попередніх оцінок збитків. Логіка цих обмежень очевидна: відкрите оприлюднення такої інформації може бути використане противником для верифікації результатів ударів та коригування подальших дій. Внаслідок цього самостійне формування та публічне оприлюднення баз даних із первинною інформацією про руйнування є, як правило, неправомірним. Винятком, що визнається законодавством, є дані, отримані з офіційно відкритих державних реєстрів та публічних інформаційних систем. Саме такий статус і робить їх правомірним джерелом для відкритих наукових досліджень у цій сфері. Ці правові обмеження враховано при проектуванні інформаційно-аналітичної системи, представленої у дисертації: доступ до первинних даних розмежовано між користувачами, а для навчання моделі прогнозування вартості використано лише відкриті джерела.

У міжнародній практиці методологічним орієнтиром для роботи з відкритими цифровими джерелами в умовах збройного конфлікту став Протокол Берклі з цифрових розслідувань на основі відкритих джерел (Berkeley Protocol on Digital Open Source Investigations), розроблений Управлінням Верховного комісара ООН з прав людини спільно з Центром прав людини Школи права Університету Каліфорнії в Берклі. Англomовну версію документа представлено у грудні 2020 р., офіційні переклади всіма мовами ООН – у 2022 р. [7]. Протокол визначає мінімальні професійні стандарти ідентифікації, збору, збереження, верифікації та аналізу цифрової інформації з відкритих джерел – соціальних мереж, супутникових знімків, відеоматеріалів, веб-архівів, відкритих реєстрів. До його ключових принципів належать методологічна обґрунтованість, відтворюваність результатів, технічна та правова коректність ланцюга збереження даних, а також мінімізація шкоди для тих, чії відомості документуються. Сьогодні ці принципи активно використовуються як методологічний орієнтир для українських слідчих та прокурорів при документуванні воєнних злочинів.

Хоча Протокол Берклі орієнтований передусім на завдання правосуддя, його методологічна основа є цінною й для аналітичних завдань підтримки прийняття рішень: вона задає рамку, в межах якої первинні дані про пошкодження можна використовувати опосередковано – через офіційні агреговані відкриті джерела, без ризику дискредитації результатів дослідження або порушення режиму інформаційної безпеки.

В українському правовому полі засади відкритості даних закріплено Законом України «Про доступ до публічної інформації» № 2939-VI [8] та Постановою Кабінету Міністрів України № 835 від 21 жовтня 2015 р., якою затверджено Положення про набори даних, що підлягають оприлюдненню у формі відкритих даних. Серед таких наборів – інформація про публічні закупівлі, оператором якої виступає електронна система Prozorro. Система функціонує з 2016 р. на підставі Закону України «Про публічні закупівлі» № 922-VIII [9] і надає вільний доступ до повного циклу закупівельних процедур: оголошень, тендерної документації, протоколів розкриття, укладених договорів, проектно-кошторисних розрахунків.

Для цього дослідження Prozorro має ключове значення. З одного боку, це офіційно відкрите джерело, дані з якого можуть бути правомірно використані для побудови емпіричної бази без додаткових погоджень чи знеособлення. З іншого – фактично єдиний на сьогодні доступний масив реальних, верифікованих експертними оцінками та закріплених у договорах значень вартості ремонтно-відновлювальних робіт, що дозволяє провести навчання моделей прогнозування на фактичному матеріалі, а не на синтетичних або експертно сконструйованих наборах. Водночас Prozorro має і суттєві обмеження як емпіричне джерело: воно фіксує лише ті об'єкти, відновлення яких стало предметом публічної закупівлі (отже, переважно соціальну, освітню та комунальну інфраструктуру), не містить деталізованих конструктивно-технічних характеристик об'єктів, а проектно-кошторисна документація доступна не для всіх процедур. Ці обмеження прямо позначилися на репрезентативності навчальної вибірки.

Аналіз законодавчого забезпечення дозволяє зробити кілька висновків, що мають значення для дисертаційного дослідження. По-перше, держава створила розгалужену систему обліку пошкоджень та механізмів компенсації, проте ці інструменти орієнтовані переважно на реєстрацію та адміністрування – вони не містять засобів прогнозування вартості відновлення чи аналітичної підтримки порівняння альтернативних рішень. По-друге, класифікація категорій пошкоджень та правовий режим об'єктів культурної спадщини формують вимоги до набору ознак, які має враховувати система підтримки прийняття рішень. По-третє, в умовах воєнного стану доступ до первинних даних про пошкодження є обмеженим, тому офіційні відкриті джерела, зокрема електронна система Prozorro, є найбільш доступним правомірним джерелом для формування емпіричної бази. Саме це й визначило вибір цього джерела даних для навчання моделі прогнозування вартості відновлення.

#### **1.4 Аналіз існуючих підходів та програмних рішень**

З метою обґрунтування доцільності створення власного програмного продукту та визначення його функціональних вимог було проведено аналіз наявних рішень у трьох взаємопов'язаних напрямках: платформи фіксації та візуалізації руйнувань, підходи до прогнозування вартості відновлення та застосування технологій цифрових двійників у сфері інфраструктури.

##### **1.4.1 Платформи фіксації та візуалізації руйнувань**

На першому етапі аналізу було розглянуто програмні рішення, які використовуються для фіксації, обліку та аналізу руйнувань в Україні. Для порівняння було обрано чотири платформи, які на момент проведення дослідження мали відкритий доступ та охоплювали різні аспекти роботи з даними про пошкодження: веб-сайт проєкту «Росія заплатить», мапу руйнувань

від ГО «Антикорупційний штаб», мапу руйнувань проекту «План відновлення України» та платформу спільноти журналістів «Bellingcat». Кожну з цих платформ було оцінено за такими критеріями: спосіб візуалізації даних, наявність механізмів збору інформації від користувачів, аналітичні можливості та підтримка прогнозування вартості відновлення.

Проект «Росія заплатить» [10] функціонує як централізований ресурс для збору та опрацювання даних про збитки, яких зазнали держава та громадяни України внаслідок російської агресії. На сайті реалізовано форму для внесення інформації про завдану матеріальну шкоду, а також окремий розділ для перегляду статистичних звітів щодо руйнувань (рис. 1.1). Сильною стороною проекту є масштаб залучення даних і співпраця з міжнародними організаціями, проте методологія оцінки збитків носить переважно описовий характер, а доступна аналітика обмежується агрегованою статистикою без деталізації до рівня окремих об'єктів.

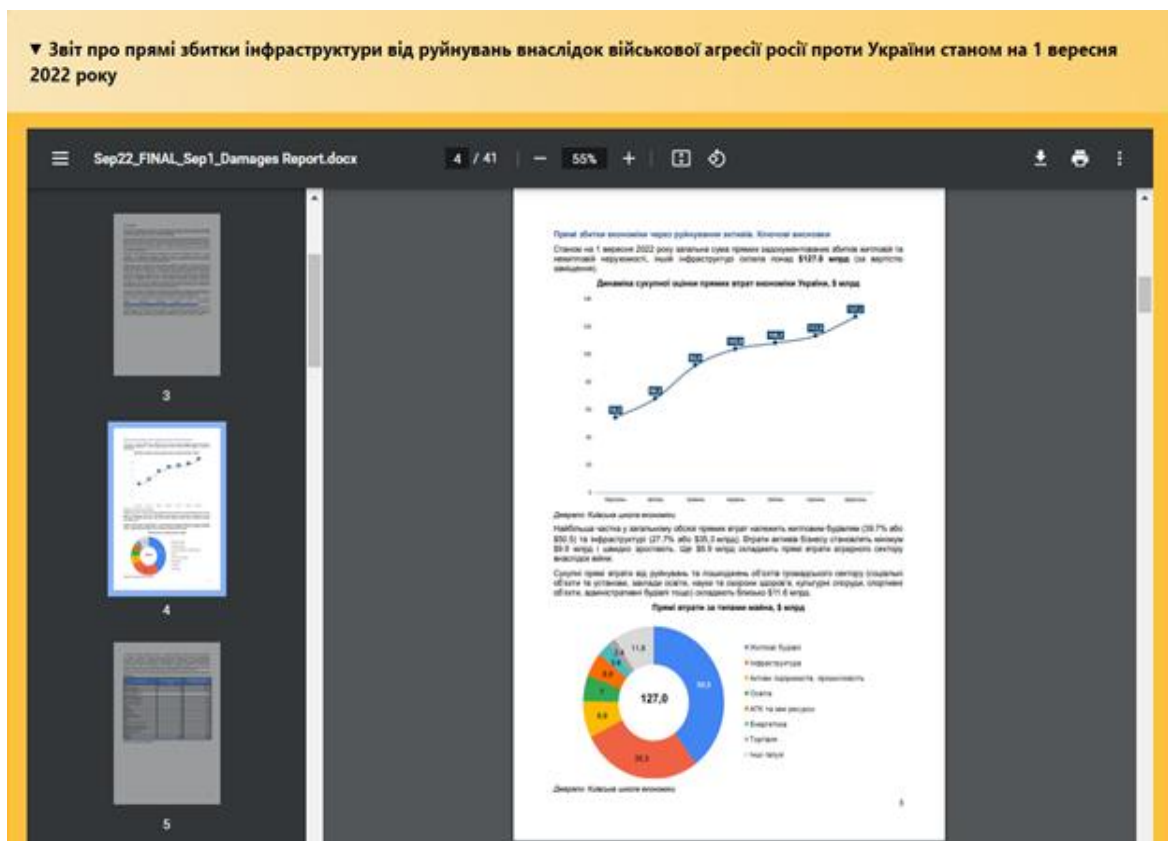


Рисунок 1.1 – Звіт від проекту «Росія заплатить»

Мапа руйнувань від ГО «Антикорупційний штаб» [11] дозволяє відобразити об'єкти інфраструктури, пошкоджені внаслідок вторгнення, з прив'язкою до географічних координат. Кожен об'єкт супроводжується фотоматеріалами – до та після обстрілу. Важливою особливістю є можливість для будь-якого користувача внести інформацію про руйнування через відповідну форму, після чого модератори перевіряють дані та додають їх на карту (рис. 1.2). Водночас платформа орієнтована виключно на картографічне відображення і не містить засобів для аналітичної обробки зібраних даних.

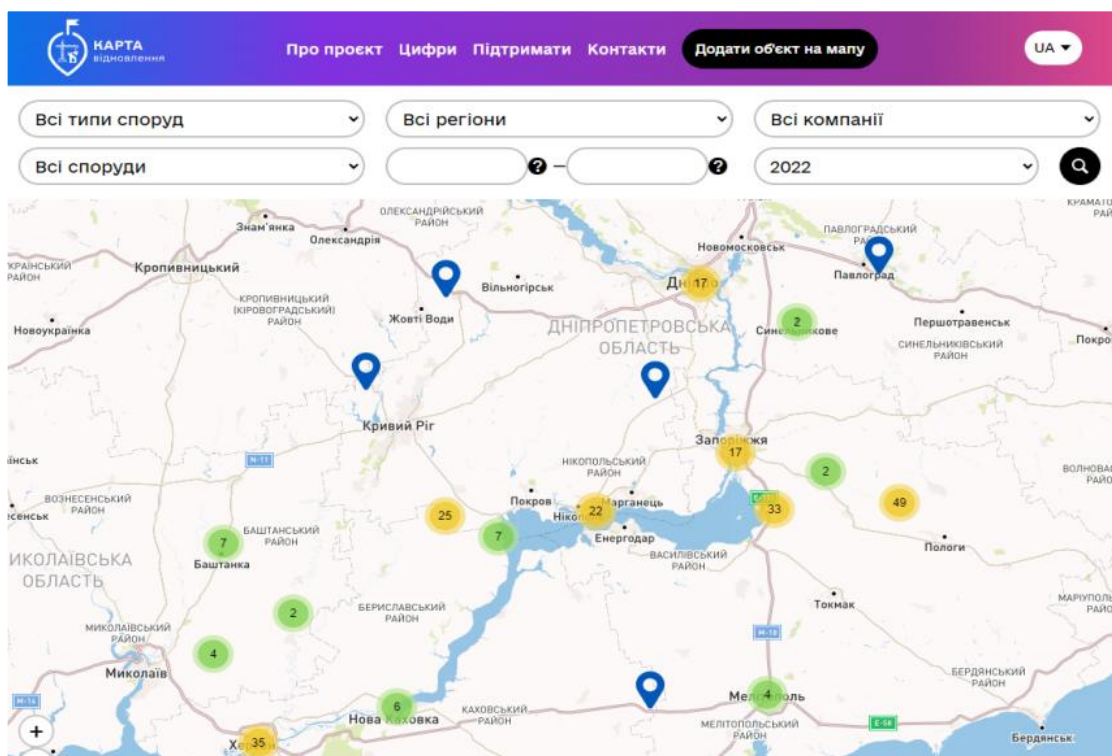


Рисунок 1.2 – Мапа руйнувань від ГО «Антикорупційний штаб»

Проект «План відновлення України» [12] надає детальну кількісну інформацію про зруйновані об'єкти в розрізі категорій: кількість шкіл, лікарень, кілометрів доріг, квадратних метрів житлової площі тощо (рис. 1.3). Перевагою є наявність структурованих числових показників на рівні регіонів, однак ці дані

використовуються виключно для описового представлення без інструментів багатовимірного аналізу чи моделювання.



Рисунок 1.3 – Мапа руйнувань проєкту «План відновлення України»

Платформа спільноти журналістів «Bellingcat» [13] пропонує інтерактивну візуалізацію з хронологією подій – обстрілів, руйнувань та загибелі мирних жителів, побудовану на основі відкритих джерел (рис. 1.4). Ця платформа зорієнтована насамперед на документування воєнних подій у часовому вимірі, а не на облік інфраструктурних об’єктів, тому не передбачає ані класифікації за ступенем пошкодження, ані фінансових оцінок.

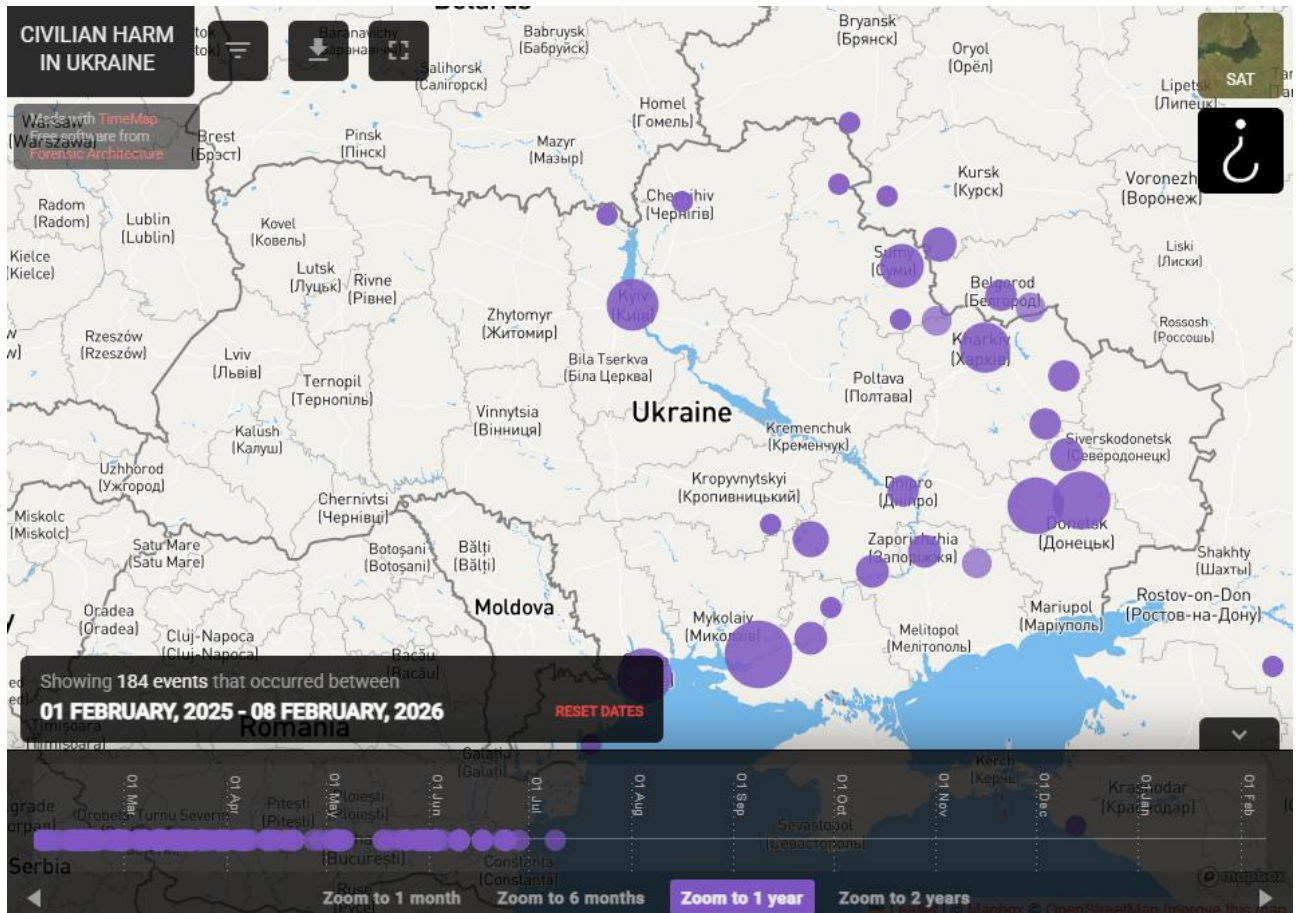


Рисунок 1.4 – Платформа спільноти журналістів «Bellingcat»

Проведений аналіз дозволив виявити спільну для всіх розглянутих платформ рису: кожна з них зосереджується на одній або кількох суміжних функціях – картографічному відображенні, опрацюванні заяв від громадян, агрегованій статистиці або хронологічній фіксації подій. Жодна з них не поєднує ці можливості у єдиному середовищі, яке б одночасно забезпечувало збір даних, їх багатовимірний аналіз та прогнозування вартості відновлення пошкоджених об'єктів. Відсутній також механізм сценарного моделювання, який дозволив би оцінювати вплив зміни характеристик об'єкта на орієнтовну вартість відбудови [14].

Узагальнені результати порівняння наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння існуючих програмних рішень для фіксації та аналізу

Критерій	«Росія заплатить»	«Антикорупц. штаб»	«План відновл. України»	«Bellingcat»
Інтерактивна карта	-	+	+	+
Збір заяв від користувачів	+	+	-	-
Статистичні звіти	+	-	+	-
Багатовимірний аналіз	-	-	-	-
Прогнозування вартості відновлення	-	-	-	-
Сценарний аналіз	-	-	-	-

#### 1.4.2 Підходи до прогнозування вартості відновлення

Окрім платформ фіксації руйнувань, було проаналізовано наукові та прикладні роботи, що стосуються оцінки вартості відновлення пошкоджених об'єктів та суміжних задач прогнозування.

Тематика відбудови, фіксації та моніторингу пошкоджень привертає значну увагу в сучасних дослідженнях. В одній із робіт детально розглянуто математичне забезпечення проєктів відновлення історичних пам'яток, однак питання фінансового прогнозування в ній не охоплюється [15]. Автори дослідження [16] описують застосування ГІС-технологій для картографування збитків, а в роботі [17] увагу зосереджено на підходах до управління ризиками та

механізмах залучення інвестицій для відбудови. У роботі [18] запропоновано підхід до оцінки збитків на основі супутникових і польових даних, що є доцільним в умовах обмеженого фізичного доступу до об'єктів.

Водночас стратегічні документи з відбудови формулюють загальні принципи та критерії відповідно до міжнародних вимог (IFIs, ESG), але не пропонують прикладних процедур прогнозування вартості на рівні конкретних об'єктів [19]. Методологічні підходи гуманітарного планування, зокрема HNRP та JAF 2.0, орієнтовані на оцінку потреб населення та ступеня тяжкості наслідків, а не на обчислення грошових витрат для окремих інфраструктурних одиниць [20]. Бюджетні дослідження фіксують значний дефіцит державних фінансів та високу залежність від зовнішньої допомоги, що додатково підкреслює актуальність інструментів попереднього кошторисного прогнозування [21].

У прикладних технічних роботах продемонстровано доцільність використання ансамблевих моделей машинного навчання, зокрема градієнтного підсилення та Random Forest, для задач будівельно-технічної експертизи. Проте ці моделі застосовуються здебільшого для класифікаційних задач – визначення категорії пошкодження, ідентифікації дефектів, а не для прогнозу вартості відновлення у грошовому виразі [22]. Локальні програми комплексного відновлення територіальних громад, зі свого боку, містять значні масиви структурованих даних про об'єкти, ступінь їх пошкоджень та заплановані заходи, однак використовуються переважно для описового планування без автоматизованого алгоритмічного прогнозування витрат [23].

Таким чином, більшість наявних рішень обмежується збором та геопросторовою візуалізацією наслідків руйнувань і генерацією звітів без розробки моделей прогнозування для оцінки вартості відновлення пошкоджених об'єктів. Попри активний розвиток цифрових інструментів для обліку та аналізу збитків, залишається невирішеним завдання створення інтегрованих рішень, що дозволяють автоматично прогнозувати очікувану вартість відновлення об'єктів

на основі їх характеристик (тип, площа, ступінь пошкодження тощо). Нестача таких інструментів обмежує спроможність оперативно ухвалювати обґрунтовані фінансові рішення. Крім того, недостатньо дослідженим залишається підхід до виявлення потенційно завищених сум на ранніх етапах аналізу заявок на відбудову, що могло б суттєво посилити механізми прозорості та контролю в розподілі ресурсів на відновлення.

#### 1.4.3 Технології цифрових двійників для інфраструктурних об'єктів

Окремої уваги заслуговують технології цифрових двійників (Digital Twin, DT), інтерес до яких помітно зростає в контексті моделювання інфраструктурних об'єктів та підтримки прийняття управлінських рішень. Специфіка задач відновлення зруйнованої інфраструктури підтверджує доцільність такого підходу, оскільки потребує інструментів, здатних поєднати дані про фактичний стан об'єктів із прогностичними моделями та забезпечити порівняння альтернативних сценаріїв відбудови.

У ряді оглядових досліджень, присвячених застосуванню технологій цифрових двійників у сфері інфраструктури [24], [25], [26], розглянуто підходи до моніторингу стану об'єктів, або діагностиці інженерних систем поза контекстом прямої оцінки витрат. Наприклад, дослідження [24] узагальнює методики створення DT для цивільної інфраструктури, однак не включає кошторисний прогноз вартості відновлення. Робота [25] розвиває семантичну онтологію DT для інфраструктури, але не пропонує механізмів перетворення характеристик об'єкта (площа, тип ремонту, ступінь пошкодження) у вартісні сценарії. В огляді [26] основну увагу приділено транспортній інфраструктурі та управлінню її життєвим циклом, проте зазначено, що практичні приклади реалізації таких систем залишаються поодинокими навіть на рівні окремих об'єктів.

Існує також група робіт, які безпосередньо звертаються до теми витрат: у дослідженні [27] запропоновано методіку оцінювання витрат на впровадження DT у промисловому контексті, але не для реконструкції пошкоджених об'єктів. Стаття [28] поєднує DT з моделями штучного інтелекту в модульному будівництві, але з акцентом на зведення нових споруд, а не на відбудову пошкоджених. У дослідженні [29] проаналізовано DT у контексті цивільної інфраструктури, але головний акцент – на структурному стані (SHM) і сенсорах, а не на грошовій оцінці чи управлінні сценаріями витрат.

Ще одна група публікацій присвячена застосуванню сценарного (what-if) аналізу в концепції цифрових двійників. У роботі [30] представлено архітектуру DT із підтримкою сценарного моделювання, однак її застосування обмежується медичною галуззю і не охоплює сферу відновлення інфраструктурних об'єктів. Дослідження [31] робить огляд DT у будівництві, охоплюючи велику кількість тематик (проектування, енергетика, FM), проте приділяє недостатню увагу до кошторисної оцінки та фінансового прогнозування. В [32] розглянуто економічні аспекти впровадження DT, проте запропонована методологія не адаптована до задач відбудови пошкодженої нерухомості. В публікації [33] проаналізовано застосування DT у міському середовищі та визначено ключові бар'єри впровадження, серед яких – відсутність інтегрованої моделі, що пов'язує фактичний стан об'єкта з його фінансовими параметрами.

Отже, попри активний розвиток технологій цифрових двійників у сфері інфраструктури, їх застосування для прогнозування вартості відновлення пошкоджених об'єктів та порівняння альтернативних сценаріїв відбудови залишається практично не дослідженим. Більшість робіт зосереджена на моніторингу технічного стану, діагностиці інженерних систем або управлінні життєвим циклом об'єктів, тоді як задача перетворення характеристик пошкодженого об'єкта у вартісні сценарії з можливістю їх інтерактивного перерахунку не отримала належного висвітлення у літературі.

## 1.5 Висновки за розділом 1

У цьому розділі обґрунтовано актуальність дисертаційного дослідження, проаналізовано нормативно-правове забезпечення фіксації пошкоджень та відновлення об'єктів, а також проведено критичний аналіз існуючих підходів та програмних рішень: платформ фіксації руйнувань, підходів до прогнозування вартості відновлення та технологій цифрових двійників.

Аналіз платформ фіксації руйнувань (проект «Росія заплатить», мапа руйнувань ГО «Антикорупційний штаб», «План відновлення України», Bellingcat) засвідчив їхню фрагментарність: кожна зосереджується на окремих функціях – картографічному відображенні, агрегованій статистиці або хронологічній фіксації подій, – але жодна не поєднує збір даних, багатовимірний аналіз та прогнозування вартості відновлення в єдиному середовищі.

Огляд наукових робіт щодо прогнозування вартості відновлення виявив, що ансамблеві моделі машинного навчання застосовуються переважно для класифікаційних задач, а не для регресійного прогнозу вартості у грошовому виразі. Описовим плануванням обмежуються і локальні програми відновлення громад, які містять структуровані дані про об'єкти, але не використовують алгоритмічне прогнозування витрат. Залишається невирішеним завдання автоматичного прогнозування очікуваної вартості відновлення на основі характеристик об'єкта.

Аналіз технологій цифрових двійників у сфері інфраструктури показав, що, попри активний розвиток цієї галузі, застосування ДТ для прогнозування вартості відновлення пошкоджених об'єктів та порівняння альтернативних сценаріїв відбудови залишається практично не дослідженим. Більшість робіт зосереджена на моніторингу технічного стану та управлінні життєвим циклом, тоді як задача перетворення характеристик пошкодженого об'єкта у вартісні сценарії не отримала належного висвітлення.

На підставі проведеного аналізу сформульовано науково-практичну проблему: відсутність інтегрованого інструменту, який поєднував би реєстрацію пошкоджень, багатовимірну аналітику, прогнозування вартості відновлення та сценарне моделювання в єдиній системі підтримки прийняття рішень. Визначено задачі дослідження та обґрунтовано необхідність розробки комплексної інформаційно-аналітичної системи.

Результати розділу 1 опубліковані в роботах автора: [14].

## 2. КОНЦЕПТУАЛЬНІ ОСНОВИ ПОБУДОВИ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДНОВЛЕННЯ ПОШКОДЖЕНИХ ОБ'ЄКТІВ

### 2.1 Формалізація задачі прийняття рішень щодо відновлення об'єктів інфраструктури

Наявні інформаційні системи фіксації руйнувань виконують переважно облікову функцію, не надаючи аналітичної підтримки для порівняння альтернатив відновлення. Водночас задача вибору способу відбудови конкретного об'єкта є типовою задачею прийняття рішень у складних системах за умов невизначеності [1].

Для забезпечення систематичного та обґрунтованого підходу до цього вибору необхідно формалізувати задачу, визначивши множину альтернатив, вектор критеріїв їх оцінки та обмеження, що накладаються на простір допустимих рішень. Оскільки кожену альтернативу необхідно одночасно оцінювати за кількома різнорідними показниками, сформульована задача належить до класу багатокритеріальних задач вибору в умовах невизначеності.

Формально задачу описано кортежем:

$$D = \langle A, K, H, A^*(h) \rangle, \quad (2.1)$$

де  $A = \{a_1, \dots, a_5\}$  – множина альтернатив відновлення:  $a_1$  – капітальний ремонт,  $a_2$  – поточний ремонт,  $a_3$  – реставрація,  $a_4$  – знесення з подальшим новим будівництвом,  $a_5$  – консервація (тимчасове збереження об'єкта у поточному стані);

$K$  – вектор критеріїв оцінки альтернатив, що включає: прогнозовану вартість відновлення  $V$  та її інфляційно скориговане значення  $V(t)$  для планового місяця початку робіт, ознаки з найбільшим внеском у прогнозовану вартість

(визначені за SHAP-аналізом) та результати сценарного (what-if) перерахунку при зміні параметрів об'єкта;

$H$  – множина рівнів культурної спадщини об'єкта: без статусу, місцевого, регіонального, національного або світового значення;

$A^*(h) \subseteq A$  – допустима підмножина альтернатив, що залежить від статусу спадщини конкретного об'єкта [34].

Введення елемента  $A^*(h)$  є принциповим: не всі альтернативи є доступними для кожного об'єкта. Статус культурної спадщини виступає обмежуючим фактором, що звужує простір допустимих рішень. Зокрема, відповідно до Закону України «Про охорону культурної спадщини» (№ 1805-III, ст. 26), для об'єктів національної культурної спадщини знесення може бути виключене з множини допустимих рішень, а пріоритетною стає реставрація.

Обмеження множини допустимих альтернатив залежно від рівня культурної спадщини наведено у таблиці 2.1.

Таблиця 2.1 – Допустимі альтернативи залежно від статусу культурної спадщини

Рівень спадщини ( $h$ )	Капіт. ремонт	Поточ. ремонт	Реставрація	Знесення + нове буд.	Консервація
Без статусу	+	+	+	+	+
Місцевого значення	+	+	+	+	+
Регіонального знач.	+	+	+	-	+
Національного знач.	-	-	+	-	+
Світового значення	-	-	+	-	-

Таким чином, для об'єктів без статусу спадщини доступні всі п'ять альтернатив. З підвищенням статусу множина звужується: для об'єктів регіонального значення виключається знесення, для національного значення допускається лише реставрація та консервація, а для об'єктів світового значення – виключно реставрація.

Невизначеність у сформульованій задачі зумовлена кількома факторами: неповнотою даних про фактичний стан об'єкта (ступінь пошкодження може бути уточнений після додаткового обстеження), мінливістю цін на будівельні матеріали та роботи, можливістю повторних пошкоджень внаслідок продовження бойових дій, а також варіативністю обсягу необхідних робіт. Ці фактори враховуються через механізм сценарного перерахунку, який є складовою вектора критеріїв  $K$ .

Узагальнену послідовність кроків, у якій елементи кортежу  $D$  перетворюються на конкретний розв'язок задачі для окремого об'єкта, наведено на рис. 2.1. Алгоритм охоплює формування допустимої підмножини  $A^*(h)$  на основі статусу спадщини, циклічне обчислення критеріїв для кожної допустимої альтернативи (кодування ознак, ML-прогноз  $V$ , інфляційне коригування  $V(t)$ , SHAR-розрахунок ознак з найбільшим внеском у прогноз) та інтерактивний сценарний перерахунок, що дозволяє ОПР оцінити вплив змін окремих характеристик об'єкта на результат.

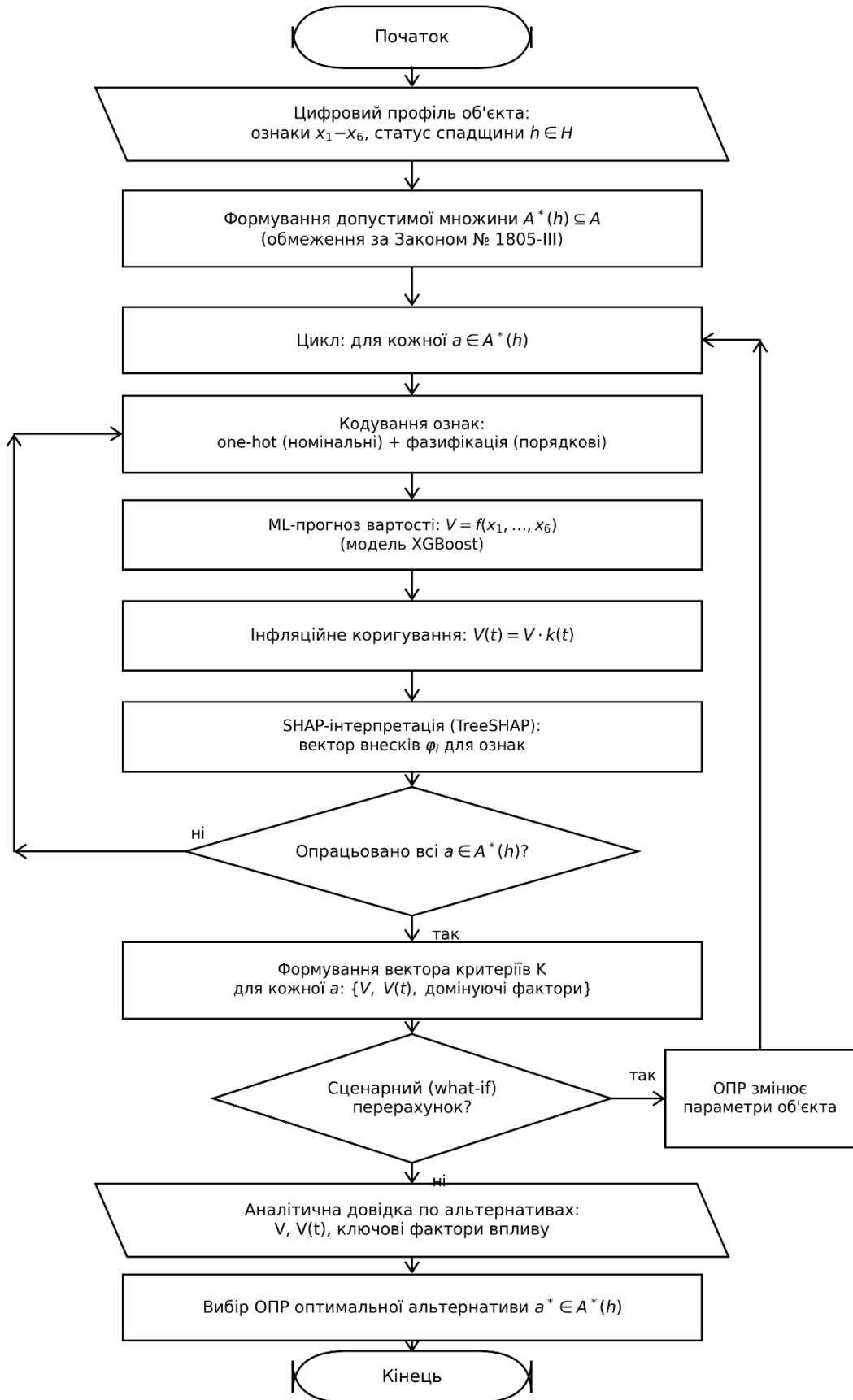


Рисунок 2.1 – Алгоритм прийняття рішення щодо відновлення об'єкта

На виході алгоритму для кожної допустимої альтернативи  $a \in A^*(h)$  система формує аналітичну довідку, що містить: прогнозовану вартість  $V$  та перелік ознак з найбільшим внеском. Фінальне рішення залишається за особою, що приймає рішення (ОПР), – система забезпечує структуровану інформаційну основу для порівняння альтернатив, залишаючи вибір конкретного варіанту за ОПР.

Практичну відповідність запропонованої формалізації реальному процесу прийняття рішень підтверджує випадок житлового будинку по вул. Паралельній, 4 у м. Запоріжжі. Виконавчий комітет міської ради (рішення від 12.03.2026 № 177) ухвалив демонтаж будівлі на підставі попереднього технічного звіту, що зафіксував аварійний стан конструкцій (категорія «4»). У термінах запропонованого кортежу  $D = \langle A, K, H, A^*(h) \rangle$  цю ситуацію можна описати так: будівля не має статусу культурної спадщини, отже  $A^*(h) = A$  і доступні всі п'ять альтернатив. Рішення про демонтаж (альтернатива  $a_4$  – знесення з подальшим новим будівництвом) було обране за критерієм економічної доцільності: витрати на капітальний ремонт або реконструкцію перевищували б вартість нового будівництва аналогічного об'єкта. Саме такий тип порівняння – зіставлення прогнозованої вартості  $V$  для різних альтернатив – становить основу вектора критеріїв  $K$  у запропонованій моделі. При цьому рішення приймалося на основі розрахунків «у довільній формі», без стандартизованого аналітичного інструменту, що підкреслює потребу в системі підтримки прийняття рішень, яка б забезпечувала порівняння альтернатив на основі формалізованих критеріїв.

## 2.2 Формалізація задачі прогнозування вартості відновлення

Ключовим критерієм для оцінки альтернатив є прогнозована вартість відновлення  $V$ . Задачу визначення цієї величини сформульовано як регресійну задачу машинного навчання: модель на основі набору числових і категоріальних

ознак, що описують характеристики пошкодженого об'єкта, прогнозує очікувану вартість його відновлення.

Вибір саме регресійної постановки задачі потребує окремого обґрунтування, оскільки існують альтернативні підходи до прогнозування вартості. Один із них – класифікація за діапазонами: вартість відновлення розбивається на кілька інтервалів (наприклад, «до 30 млн грн», «30–60 млн грн», «понад 60 млн грн»), і модель визначає, до якого з них належить об'єкт. Такий підхід спрощує задачу і може забезпечити вищу точність на малих вибірках, проте він принципово непридатний для сценарного аналізу. Коли ОПР змінює характеристику об'єкта і обидва сценарії потрапляють у той самий діапазон, класифікатор не дає підстав для порівняння – різниця між вартістю 35 і 55 млн грн у межах одного класу залишається невидимою. Регресія, на відміну від класифікації, повертає неперервне числове значення, що дозволяє фіксувати навіть незначні зміни прогнозу при варіюванні параметрів.

Іншою альтернативою є ймовірнісна регресія, яка замість точкового прогнозу формує розподіл можливих значень вартості – наприклад, у вигляді довірчого інтервалу. Такий підхід дав би ОПР додаткову інформацію про міру невизначеності прогнозу. Однак його коректна реалізація потребує значно більшого обсягу навчальних даних для надійної оцінки дисперсії, тоді як наявна вибірка ( $\approx 100$  записів) є недостатньою для цього. З огляду на зазначені обмеження альтернативних підходів, у роботі обрано класичну регресійну постановку, для якої цільовою змінною слугує оціночна вартість, зафіксована у проектно-кошторисній документації.

Навчання моделі здійснюється на основі оціночної вартості  $S$  – вартості відновлення, визначеної експертами та зафіксованої у проектно-кошторисній документації системи Prozorro. Ця величина виступає цільовою змінною, яку модель навчається відтворювати, мінімізуючи похибку між прогнозованим та фактичним значенням. Результатом роботи моделі є прогнозована вартість  $V$  –

числове значення орієнтовної вартості відновлення, сформоване на основі характеристик пошкодженого об'єкта.

Для побудови прогнозу було сформовано набір незалежних змінних  $x_1, x_2, \dots, x_6$ , які відображають фізичні характеристики об'єкта, умови його розташування та контекст завданих пошкоджень:

- $x_1$  – площа об'єкта ( $m^2$ ): числова ознака, яка прямо впливає на вартість матеріалів і робіт;
- $x_2$  – поверховість: числова ознака, що враховує конструктивну складність об'єкта;
- $x_3$  – тип будівлі: категоріальна ознака, яка позначає функціональний тип об'єкта (приватний будинок, багатоквартирна будівля, гуртожиток тощо);
- $x_4$  – ступінь пошкодження: категоріальна ознака, що задається рівнями «легке», «середнє» або «тяжке»;
- $x_5$  – географічне розташування: категоріальна ознака, яка відображає область або населений пункт, що може впливати на вартість робіт і логістику;
- $x_6$  – тип ремонту: категоріальна ознака, що описує спосіб відновлення («поточний ремонт», «капітальний ремонт», «повна реконструкція»).

Таким чином, модель можна представити у вигляді функції:

$$V = f(x_1, x_2, x_3, x_4, x_5, x_6) + \varepsilon, \quad (2.2)$$

де  $\varepsilon$  – випадкова похибка моделі.

Серед наведених ознак  $x_3$ – $x_6$  є категоріальними, що потребує їх числового представлення для використання у моделі машинного навчання.

Реалізація сформульованої регресійної задачі передбачає цілісний процес, у якому навчання моделі та її застосування для нового об'єкта виконуються у двох логічно різних режимах. Узагальнену схему цього процесу наведено на рис. 2.2.

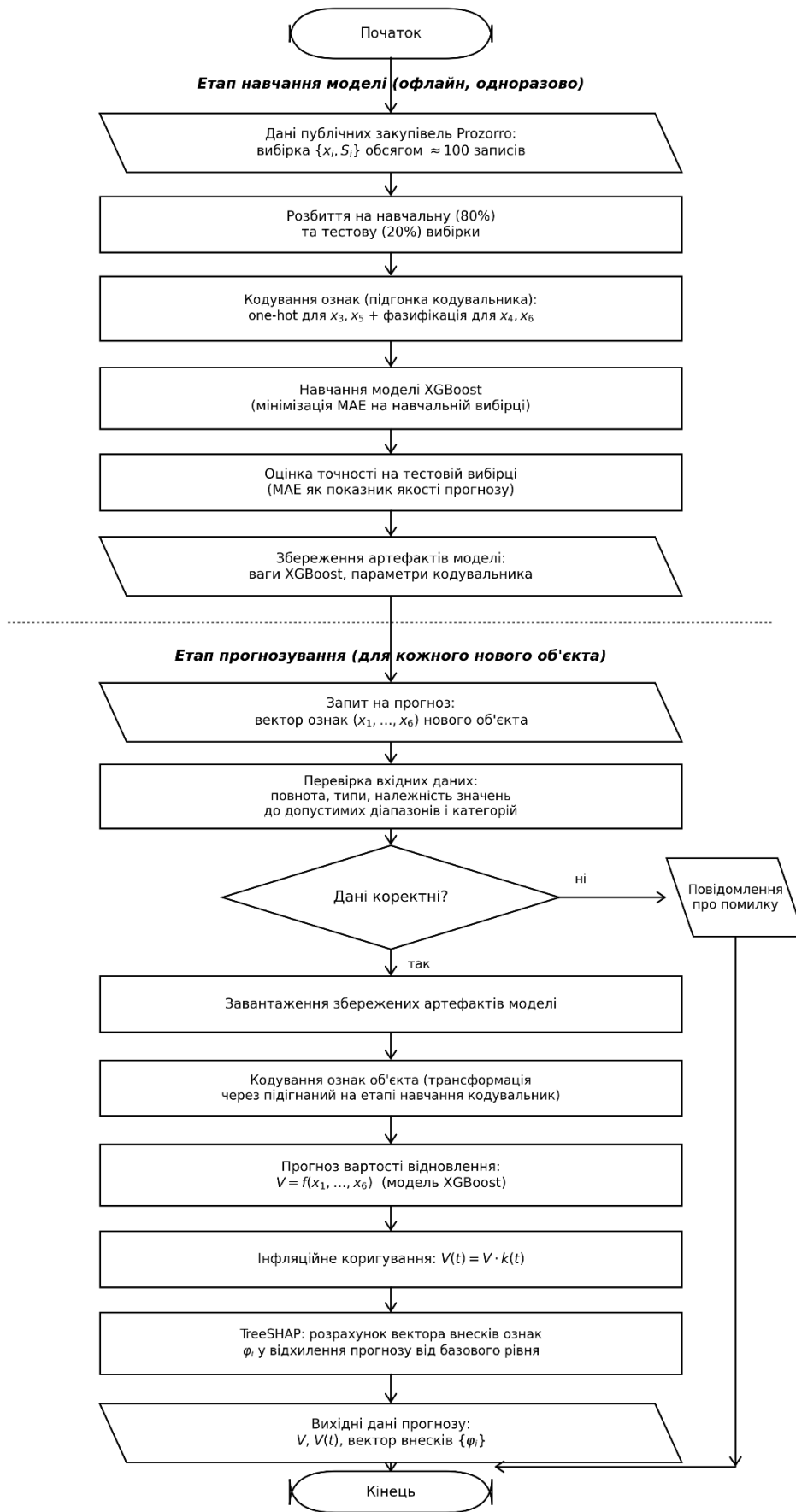


Рисунок 2.2 – Блок-схема процесу прогнозування вартості відновлення

Етап навчання моделі виконується один раз в офлайн режимі. Вибірка з Prozorro попередньо розбивається на навчальну і тестову частини. На навчальній частині визначаються параметри кодування ознак та будується регресійна модель, після чого її точність оцінюється на тестовій вибірці. Отримані в результаті артефакти серіалізуються для подальшого використання.

Етап прогнозування виконується для кожного нового об'єкта і починається з валідації вхідних даних на повноту, типи та належність до допустимих значень. У разі некоректних даних запит відхиляється з повідомленням про помилку. Для коректних запитів ознаки кодуються за параметрами, збереженими на етапі навчання, після чого модель повертає прогнозовану вартість  $V$ , яка коригується на інфляцію та супроводжується SHAP-розрахунком внесків окремих ознак. Деталі кожного етапу розглянуто у підрозділах 2.3–2.6.

Окрім підтримки вибору альтернативи відновлення отримане значення  $V$  може бути використане як орієнтовна оцінка для виявлення можливих завищень у заявлених витратах на відновлення. Прогнозоване значення  $V$  порівнюється з фактичною сумою, зазначеною у звітах чи контрактах ( $S$ ). Якщо  $S$  суттєво перевищує  $V$ , система може класифікувати об'єкт як такий, що потребує додаткової аналітичної або експертної перевірки. Таке застосування є особливо актуальним в умовах масштабної відбудови, коли ручний контроль обґрунтованості кошторисів за тисячами об'єктів є практично неможливим, а автоматизована попередня фільтрація дозволить зосередити експертний ресурс на випадках із найбільшими розбіжностями між прогнозом і заявленими витратами.

### **2.3 Інфляційне коригування прогнозованої вартості відновлення**

Прогнозована вартість  $V$ , що формується ML-моделлю, відповідає рівню цін на будівельно-монтажні роботи станом на базовий період навчання. Це принципове обмеження: модель «заморожує» цінову картину на момент, коли

вона навчалась, і не може самостійно враховувати подальшу динаміку цін. Між моментом оцінювання об'єкта та фактичним початком відновлювальних робіт, як правило, минає від кількох місяців до кількох років – і за цей час будівельна інфляція може суттєво змінити реальний кошторис. Ігнорування цього чинника призводить до систематичного недофінансування проєктів на етапі формування бюджетних заявок.

Мінливість цін на будівельні матеріали та роботи є одним з чинників невизначеності у задачі прийняття рішення. Без інфляційного коригування прогноз  $V$  є коректним лише для базового місяця навчання і втрачає точність при плануванні відбудови на майбутнє. Саме тому в роботі запропоновано механізм коригування, що дає змогу перерахувати прогнозовану вартість для конкретного місяця початку робіт і порівняти між собою альтернативи  $a \in A^*(h)$  з урахуванням фактору часу.

Скоригована вартість відновлення для планового місяця  $t$  визначається за формулою:

$$V(t) = V \cdot k(t), \quad (2.3)$$

де  $V$  – базовий прогноз ML-моделі,  $k(t)$  – коефіцієнт інфляційного коригування для місяця  $t$ , що показує відношення рівня цін у будівництві в місяць  $t$  до рівня базового місяця навчання моделі (січень 2025 = 1,0000).

Коефіцієнти  $k(t)$  визначено на основі офіційних місячних індексів цін на будівельно-монтажні роботи Державної служби статистики України та прогнозних оцінок Національного банку України [35]. Для місяців, що вже завершилися, використовуються фактичні опубліковані дані Держстату; для майбутніх – прогнозні значення, які адміністратор системи може оновлювати в міру публікації нових офіційних даних. Місячний крок обрано відповідно до джерела даних: Держстат публікує індекси цін у будівництві саме у місячному

розрізі, що дозволяє отримати найбільш деталізовані та верифіковані значення без необхідності довільної агрегації чи екстраполяції.

Введення  $V(t)$  розширює вектор критеріїв оцінки альтернатив  $K$  у формалізованій задачі  $D = \langle A, K, H, A^*(h) \rangle$ : тепер оцінка вартості є функцією не лише характеристик об'єкта, але й строків реалізації альтернативи відновлення. Це дозволяє ОПР кількісно оцінити додаткові витрати, зумовлені відкладанням відбудови, і використовувати цей показник як самостійний критерій при пріоритизації об'єктів у бюджетному плануванні.

## 2.4 Методи кодування категоріальних ознак

Серед визначених у попередньому підрозділі ознак  $x_3$ – $x_6$  є категоріальними, що потребує їх числового представлення для використання у моделі машинного навчання. Вибір методу кодування суттєво впливає на точність прогнозування, оскільки різні підходи по-різному передають структуру та взаємозв'язки між категоріями.

Стандартним підходом є one-hot кодування, за якого кожна категорія перетворюється на окрему бінарну колонку (0/1). Цей метод є ефективним для номінальних ознак, де категорії не мають природного порядку. Для ознак  $x_3$  (тип будівлі) та  $x_5$  (географічне розташування), які є номінальними, застосовується саме one-hot кодування.

Проте для ознак  $x_4$  (ступінь пошкодження) та  $x_6$  (тип ремонту) такий підхід є неоптимальним, оскільки ці змінні мають внутрішню градацію: категорії «легке», «середнє», «тяжке» утворюють природний порядок за рівнем пошкодження, а «поточний ремонт», «капітальний ремонт», «повна реконструкція» – за обсягом відновлювальних робіт. One-hot кодування не зберігає інформацію про цей порядок і розглядає всі категорії як рівновіддалені, що може знижувати точність прогнозування.

Альтернативним підходом є ординальне (порядкове) кодування, за якого кожній категорії присвоюється числовий ранг (1, 2, 3 тощо). Цей метод зберігає інформацію про порядок, проте нав'язує рівні відстані між категоріями, що не завжди відповідає реальності: різниця між «легким» та «середнім» пошкодженням не обов'язково дорівнює різниці між «середнім» та «тяжким».

Для коректного представлення порядкових ознак у роботі застосовано методи фазифікації, а саме представлення категорій у вигляді нечітких множин із трикутними функціями належності [36]. Цей підхід має кілька переваг: він зберігає градаційну структуру порядкових ознак, дозволяє моделювати плавні переходи між категоріями та не нав'язує жорстких рівних відстаней. Аналогічний підхід до кодування параметрів за допомогою нечіткої логіки продемонстрував ефективність у задачах визначення економічних параметрів виконання проєктів [37].

Кожній категорії присвоюється числова позиція  $p$  на нормованій шкалі  $[0; 1]$ , що відображає її місце у градації. На основі цієї позиції будується трикутна функція належності  $\mu(x; a, b, c)$ , яка визначає ступінь належності значення  $x$  до відповідної категорії. Функція задається трьома параметрами:  $a$  – ліва межа ( $\mu = 0$ ),  $b$  – вершина ( $\mu = 1$ ),  $c$  – права межа ( $\mu = 0$ ):

$$\mu(x; a, b, c) = \begin{cases} 0, & x \leq a \text{ або } x \geq c \\ \frac{x - a}{b - a}, & a < x \leq b \\ \frac{c - x}{c - b}, & b < x < c \end{cases} . \quad (2.4)$$

Параметри  $(a, b, c)$  для кожної категорії визначаються автоматично на основі позицій сусідніх категорій: вершина  $b$  відповідає позиції категорії, а межі  $a$  та  $c$  – позиціям сусідніх категорій. Для крайніх категорій відповідна межа симетрично розширюється. Параметри фазифікації для ознак  $x_4$  та  $x_6$  наведено у табл. 2.2.

Таблиця 2.2 – Параметри фазифікації порядкових ознак

Ознака	Категорія	$p$	$a$	$b$	$c$
$x_4$	Легке	0,0	-0,5	0,0	0,5
$x_4$	Середнє	0,5	0,0	0,5	1,0
$x_4$	Тяжке	1,0	0,5	1,0	1,5
$x_6$	Поточний	0,0	-0,5	0,0	0,5
$x_6$	Капітальний	0,5	0,0	0,5	1,0
$x_6$	Повна реконструкція	1,0	0,5	1,0	1,5

Таким чином, у роботі застосовано комбінований підхід до кодування: one-hot кодування для номінальних ознак ( $x_3$ ,  $x_5$ ) та фазифікацію з трикутними функціями належності для порядкових ознак ( $x_4$ ,  $x_6$ ), що зберігає інформацію про їх внутрішню градацію.

## 2.5 Ансамблеві моделі машинного навчання

Після формалізації задачі та визначення методів кодування ознак наступним кроком є вибір алгоритму машинного навчання для побудови регресійної моделі. Специфіка задачі накладає ряд вимог на модель: необхідність працювати з невеликою вибіркою (~100 записів), наявність як числових, так і категоріальних ознак після кодування, а також потреба в інтерпретації результатів для підтримки прийняття рішень. З огляду на ці вимоги для порівняльного аналізу було обрано три моделі, що охоплюють широкий спектр підходів – від простих статистичних до потужних ансамблевих методів.

*Linear Regression* – як базова лінійна модель, що дозволяє оцінити загальну структуру залежностей між ознаками і цільовим значенням. Її включення до порівняння обґрунтоване необхідністю мати еталон: якщо складніші моделі не

перевершують лінійну, це свідчить або про лінійну природу залежності, або про недостатність даних.

Аналітично модель описується рівнянням:

$$V = \beta_0 + \sum_{i=1}^n \beta_i x_i + \varepsilon, \quad (2.5)$$

де  $V$  – прогнозоване значення;  $x_i$  – вхідні ознаки;  $\beta_i$  – коефіцієнти, які показують, наскільки сильно кожна ознака впливає на результат;  $\beta_0$  – вільний член;  $\varepsilon$  – випадкова похибка.

*Random Forest Regressor* – як класична ансамблева модель, здатна виявляти складні залежності без надмірної чутливості до шуму [38]. Вибір цієї моделі обумовлений її здатністю ефективно працювати з невеликими вибірками та категоріальними ознаками, а також вбудованим механізмом регуляризації через агрегацію прогнозів множини дерев.

Аналітична форма моделі має вигляд:

$$V = \frac{1}{M} \sum_{i=1}^M f_i(x), \quad (2.6)$$

де  $f_i(x)$  – прогноз  $i$ -го дерева для вектора ознак  $x$ ;  $M$  – загальна кількість дерев.

*XGBoost Regressor* – як сучасне високопродуктивне рішення, яке зарекомендувало себе як одне з найкращих для задач на табличних даних [39]. Його перевагою є послідовне навчання дерев, де кожне нове дерево виправляє помилки попередніх, що особливо ефективно на малих вибірках [40].

Аналітична форма моделі наведена нижче:

$$\hat{V}^{(t)} = \sum_{k=1}^t f_k(x), \quad f_k \in F, \quad (2.7)$$

де  $t$  – кількість дерев, які поступово додаються;  $f_k(x)$  – нове дерево, що навчається на помилках попередніх;  $F$  – множина дерев рішень.

На кожному кроці  $t$  модель мінімізує функцію втрат з регуляризаційним доданком:

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t), \quad (2.8)$$

де  $l$  – функція втрат (наприклад, квадратична похибка), а  $\Omega$  – регуляризація, що запобігає перенавчанню.

Для оцінки точності моделей обрано показник MAE (Mean Absolute Error) – середню абсолютну помилку, яка показує середню різницю між фактичними та передбаченими значеннями вартості відновлення. На відміну від інших метрик, таких як RMSE чи  $R^2$ , MAE є легко інтерпретованою в прикладному контексті: вона вказує, на скільки в середньому гривень модель помиляється при оцінці одного об'єкта.

Порівняльний аналіз ефективності цих моделей на реальних даних наведено у розділі 4.

## 2.6 Інтерпретація прогнозів на основі SHAP-аналізу

Ансамблеві моделі машинного навчання, зокрема XGBoost, забезпечують високу точність прогнозування, проте їх внутрішня структура є складною для інтерпретації. У контексті формалізованої задачі прийняття рішень це створює проблему: ОПР отримує числову оцінку вартості  $V$ , але не розуміє, які саме

характеристики об'єкта найбільше вплинули на цю оцінку. Без такого розуміння порівняння альтернатив відновлення та аналіз сценаріїв втрачають свою обґрунтованість.

Серед існуючих підходів до інтерпретації моделей можна виділити кілька основних. Методи на основі важливості ознак (feature importance), вбудовані у дерева рішень, показують лише загальну значущість ознаки, але не пояснюють її внесок у конкретний прогноз. Методи LIME генерують локальні лінійні апроксимації, проте їх результати можуть бути нестабільними для різних запусків [41]. Часткові залежності (Partial Dependence Plots) демонструють глобальний вплив ознак, але не адаптовані до пояснення індивідуальних прогнозів.

З огляду на обмеження розглянутих методів, у роботі обрано метод SHAP (SHapley Additive exPlanations), який є сучасним стандартом інтерпретації моделей машинного навчання для табличних даних [42]. Вибір обґрунтовано кількома перевагами: SHAP забезпечує як локальну (для конкретного об'єкта), так і глобальну інтерпретацію; має строге теоретичне обґрунтування на основі теорії кооперативних ігор та значень Шеплі [43]; гарантує адитивність (сума внесків дорівнює прогнозу); а його оптимізована версія TreeSHAP працює за поліноміальний час для дерев рішень [44], що забезпечує придатність для онлайн-пояснень у веб-системі.

Метод SHAP спирається на кооперативну теорію ігор, де кожна ознака розглядається як «гравець», що робить внесок у спільний результат моделі, а прогноз інтерпретується як їх сумарний «виграш». На основі цього обчислюються значення Шеплі, які відображають середній маржинальний внесок кожної ознаки у зміну прогнозу відносно базового рівня. Для заданого об'єкта з вектором ознак  $x$  прогноз моделі  $f(x)$  подається у вигляді адитивної декомпозиції:

$$f(x) = \phi_0 + \sum_{i=1}^n \phi_i, \quad (2.9)$$

де  $\phi_0$  – базове значення (середній прогноз моделі на навчальній вибірці),  $\phi_i$  – значення Шеплі для  $i$ -тої ознаки (локальний внесок  $i$ -тої ознаки у відхилення прогнозу  $V$  конкретного об’єкта від базового рівня  $\phi_0$ ),  $n$  – загальна кількість ознак з урахуванням кодування категоріальних змінних  $x_3$ – $x_6$ .

Значення Шеплі  $\phi_i$  для  $i$ -тої ознаки визначається як середній маржинальний внесок цієї ознаки по всіх можливих підмножинах інших ознак:

$$\phi_i = \sum_{T \subseteq N \setminus i} \frac{|T|! (n - |T| - 1)!}{n!} (f(T \cup i) - f(T)), \quad (2.10)$$

де  $N$  – множина всіх ознак після кодування,  $T$  – підмножина ознак, що не містить  $i$ ,  $f(T \cup i)$  – прогноз моделі з урахуванням ознак підмножини  $T$  та ознаки  $i$ ,  $f(T)$  – прогноз моделі лише з ознаками підмножини  $T$ , а  $f(T \cup i) - f(T)$  – маржинальний внесок ознаки  $i$  при додаванні її до підмножини  $T$ .

Така формула гарантує три ключові властивості пояснення:

- локальна точність (сума базового значення  $\phi_0$  та внесків  $\phi_i$  дорівнює прогнозу  $V$ ),
- симетричність (рівнозначні ознаки мають рівні внески),
- узгодженість/монотонність (якщо ознака стає важливішою в моделі, її SHAP-внесок не зменшується).

Оскільки прямий розрахунок за формулою має експоненційну складність, у роботі використано спеціалізований алгоритм TreeSHAP, оптимізований для ансамблевих дерев. TreeSHAP обчислює значення Шеплі за поліноміальний час, використовуючи структуру дерев і розподіл шляхів у них, що робить метод придатним для онлайн-пояснень у веб-системі.

З огляду на те, що частина ознак є категоріальними і подається у вигляді one-hot кодування, у двійнику застосовано агрегацію SHAP-внесків за групами ознак. Наприклад, усі one-hot колонки, пов'язані з ознакою "тип ремонту", підсумовуються в одну групу. Після цього для кожної групи ознак  $g$  формується абсолютний локальний внесок групи ознак у прогнозовану вартість відновлення  $V$  для конкретного об'єкта (або сценарію):

$$C_g = \sum_{i \in g} \phi_i, \quad (2.11)$$

де  $g$  – група ознак (наприклад, усі one-hot колонки для однієї категоріальної змінної).

Для наочного представлення внесків в інтерфейсі цифрового двійника розраховується також відносна частка впливу кожної групи:

$$P_g = \frac{|C_g|}{\sum_{k=1}^K |C_k|} \cdot 100\%, \quad (2.12)$$

де  $K$  – загальна кількість груп ознак,  $|C_k|$  – абсолютне значення внеску  $k$ -ї групи.

На практиці використовується локальна інтерпретація SHAP, яка формується для кожного окремого прогнозу або сценарію. Система повертає вектор внесків, який показує, як кожна характеристика об'єкта збільшує або зменшує прогнозовану вартість. Це дозволяє відповісти на питання, чому саме для даного об'єкта отримано таку оцінку, а також наочно порівнювати альтернативні сценарії відбудови.

## 2.7 Концепція цифрового двійника зі сценарним аналізом

Формалізована задача прийняття рішень передбачає можливість порівняння альтернатив за різних припущень щодо стану об'єкта. На практиці це означає, що ОПР повинна мати змогу змінити будь-яку характеристику об'єкта – ступінь пошкодження, тип ремонту, площу відновлення – та одразу отримати оновлений прогноз вартості разом із новою SHAP-інтерпретацією. Саме для реалізації цієї можливості у роботі запропоновано концепцію цифрового двійника пошкодженого об'єкта інфраструктури.

У літературі виокремлюють кілька класів цифрових двійників залежно від способу відтворення поведінки фізичного об'єкта [45]. Фізико-орієнтовані двійники спираються на математичні моделі процесів (термодинаміка, механіка деформацій) і потребують детальних вхідних параметрів, які для масово пошкоджених об'єктів часто недоступні. Двійники на основі даних (data-driven) будують прогнозну модель безпосередньо за статистичними спостереженнями, що робить їх придатними для ситуацій, коли фізична модель або надто складна, або неможлива через брак інженерних обстежень. Гібридні підходи поєднують обидва класи, проте їхня практична реалізація потребує водночас і фізичної моделі, і достатнього обсягу даних. Запропонований у дисертації цифровий двійник належить до класу data-driven: його прогностичним ядром є ансамблева модель XGBoost, навчена на реальних даних Prozorro, а не на фізичному моделюванні процесу відновлення. Такий вибір зумовлений специфікою предметної області – для переважної більшості пошкоджених об'єктів відсутні детальні інженерні моделі, натомість доступні структуровані ознаки з державних реєстрів.

Слід також зазначити відмінність запропонованого підходу від BIM-орієнтованих цифрових двійників, які набули поширення у будівельній галузі. BIM-двійник оперує тривимірною геометричною моделлю об'єкта і відстежує стан його окремих конструктивних елементів. Для нового будівництва чи

планового обслуговування це виправданий підхід. Проте у контексті масової відбудови пошкоджених об'єктів ВІМ-моделі, як правило, не існує – вона або ніколи не створювалась (для радянської забудови), або втрачена внаслідок руйнувань. Більше того, задача, яку вирішує ОПР на етапі бюджетного планування, полягає не у детальному проектуванні, а у швидкому порівнянні альтернатив за вартістю. Тому у даній дисертаційній роботі цифровий двійник побудовано на рівні абстракції структурованих ознак об'єкта (площа, поверховість, тип будівлі, ступінь пошкодження, тип ремонту, регіон), що відповідає реально доступній інформації та потребам ОПР на етапі порівняння альтернатив відбудови.

Цифровий двійник у контексті даної роботи – це динамічна цифрова репрезентація конкретного об'єкта, що складається з двох взаємопов'язаних компонентів:

1. Цифровий профіль об'єкта – структурований модуль зберігання та управління даними про об'єкт (паспортні характеристики, ступінь пошкодження, фотофакти, геопросторова прив'язка).

2. Модуль сценарного розрахунку – інструмент, що дозволяє моделювати зміни: користувач варіює характеристики об'єкта, і система миттєво перераховує вартість відновлення з SHAR-інтерпретацією результату.

Цифровий профіль у сукупності з модулем сценарного розрахунку формує цифровий двійник. Двійник не лише зберігає дані, а й дозволяє моделювати зміни: кожен сценарій – це окрема комбінація значень характеристик об'єкта, для якої виконується розрахунок вартості відновлення [46].

Життєвий цикл окремого сценарію можна описати послідовністю кроків. Спочатку ОПР обирає об'єкт і бачить його поточний цифровий профіль із базовим прогнозом вартості. Далі вона модифікує одну або кілька ознак – наприклад, змінює тип ремонту з «поточний» на «капітальний» або коригує площу відновлення. Система формує новий вектор ознак, виконує кодування (one-hot для номінальних, фазифікація для порядкових) і передає його ML-

мікросервісу. Мікросервіс повертає оновлений прогноз вартості разом із вектором SHAP-значень, які показують внесок кожної ознаки у зміну прогнозу відносно базового рівня. ОПР порівнює кілька сценаріїв між собою і обирає варіант, який найкраще відповідає наявним ресурсам та пріоритетам відновлення. Принципово, що цей процес є ітеративним: ОПР може повернутися до будь-якого попереднього сценарію, уточнити параметри і створити нову гілку порівняння.

Типові сценарії включають три категорії:

- зміна типу ремонту: як зміниться кошторис при виконанні капітального ремонту замість повної реконструкції;
- уточнення ступеня пошкодження: як вплине перекваліфікація пошкодження на оцінку вартості (наприклад, після додаткового обстеження);
- коригування обсягу робіт: який ефект матиме варіювання площі відновлення на  $\pm 10\text{--}20\%$ .

Кожен сценарій супроводжується SHAP-інтерпретацією, що дозволяє ОПР не лише бачити нову вартість, а й розуміти, які саме зміни параметрів спричинили зміну прогнозу. Це забезпечує прозорість і обґрунтованість процесу порівняння альтернатив.

Варто підкреслити, що сценарний механізм цифрового двійника є також інструментом роботи з невизначеністю. Коли точне значення ознаки невідоме (наприклад, ступінь пошкодження визначений попередньо й може бути уточнений після детального обстеження), ОПР має змогу розглянути кілька варіантів і оцінити діапазон можливих вартостей. Це не замінює повноцінний аналіз чутливості із статистичними довірчими інтервалами, проте на етапі первинного бюджетного планування дає достатнє уявлення про вплив невизначеності окремих параметрів на загальний кошторис.

Таким чином, цифровий двійник є тим інформаційним інструментом, що безпосередньо забезпечує роботу з формалізованою задачею прийняття рішень  $D = \langle A, K, H, A^*(h) \rangle$ : він зберігає дані для визначення  $H$  та  $A^*(h)$ , а через модуль

сценарного розрахунку формує значення вектора критеріїв  $K$  для кожної допустимої альтернативи.

Програмна реалізація цифрового двійника та його інтеграція в архітектуру ІАС описана у підрозділі 3.8.

## 2.8 Висновки за розділом 2

У цьому розділі описано формалізацію задач та обґрунтовано вибір методів для підтримки прийняття рішень щодо відновлення пошкоджених об'єктів інфраструктури.

Формалізовано задачу прийняття рішень щодо відбудови як багатокритеріальну задачу вибору з множини альтернатив в умовах невизначеності, описану кортежем  $D = \langle A, K, H, A^*(h) \rangle$ . Визначено п'ять альтернатив відновлення (капітальний ремонт, поточний ремонт, реставрація, знесення з новим будівництвом, консервація), вектор критеріїв оцінки  $K$  та механізм обмеження допустимих альтернатив залежно від статусу культурної спадщини об'єкта відповідно до Закону України «Про охорону культурної спадщини».

Формалізовано задачу прогнозування вартості відновлення як регресійну задачу машинного навчання  $V = f(x_1, \dots, x_6) + \varepsilon$  з шістьма вхідними ознаками (площа, поверховість, тип будівлі, ступінь пошкодження, регіон, тип ремонту). Запропоновано комбінований підхід до кодування категоріальних ознак: one-hot кодування для номінальних змінних (тип будівлі, регіон) та фазифікацію з трикутними функціями належності для порядкових (ступінь пошкодження, тип ремонту), що дозволяє зберегти інформацію про внутрішню градацію ознак.

Запропоновано механізм інфляційного коригування  $V(t) = V \cdot k(t)$ , де  $k(t)$  – коефіцієнт інфляції для місяця  $t$  відносно базового місяця навчання моделі, що дозволяє перераховувати прогнозовану вартість для конкретного місяця початку

робіт та кількісно оцінювати додаткові витрати, зумовлені відкладанням відбудови.

Обґрунтовано вибір трьох регресійних моделей для порівняльного аналізу: Linear Regression як базова модель, Random Forest Regressor та XGBoost Regressor як ансамблеві методи, здатні ефективно працювати з невеликими вибірками та категоріальними ознаками. Обрано метод SHAP (зокрема оптимізований алгоритм TreeSHAP) для інтерпретації прогнозів, що забезпечує адитивну декомпозицію прогнозу та визначення внеску кожної характеристики об'єкта у вартість відновлення.

Розроблено концепцію цифрового двійника пошкодженого об'єкта як інструменту підтримки прийняття рішень, що складається з цифрового профілю (паспортні характеристики, фотофакти, геопросторова прив'язка) та модуля сценарного розрахунку з SHAP-інтерпретацією. Цифровий двійник безпосередньо забезпечує роботу з формалізованою задачею  $D$ : зберігає дані для визначення  $H$  та  $A^*(h)$ , а через сценарний перерахунок формує значення вектора критеріїв  $K$  для кожної допустимої альтернативи.

Результати розділу 2 опубліковані в роботах автора: [34], [46], [62].

### 3. РОЗРОБКА АНАЛІТИЧНОГО ТА ПРОГНОЗНОГО ЗАБЕЗПЕЧЕННЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДНОВЛЕННЯ ПОШКОДЖЕНИХ ОБ'ЄКТІВ У МЕЖАХ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ

#### 3.1 Принципи побудови веб-орієнтованих інформаційних систем

Для забезпечення гнучкості при роботі з системою було прийнято рішення реалізувати її у вигляді веб-додатку. Такий підхід дозволить вільно користуватися можливостями системи з будь-якого пристрою, що має доступ до мережі Інтернет.

Програмне забезпечення такого роду функціонує за принципами «клієнт-серверної» архітектури, де в якості клієнта виступає браузер, а сервера – web-сервер. Особливість даного підходу полягає в тому, що сам додаток знаходиться і виконується на сервері – клієнт при цьому отримує тільки результати роботи. Основним завданням останнього є коректне відображення завантаженої з сервера інформації, а також відправка на сервер запитів і даних користувача. Передача запитів і результатів їх обробки відбувається через Інтернет (рис. 3.1).

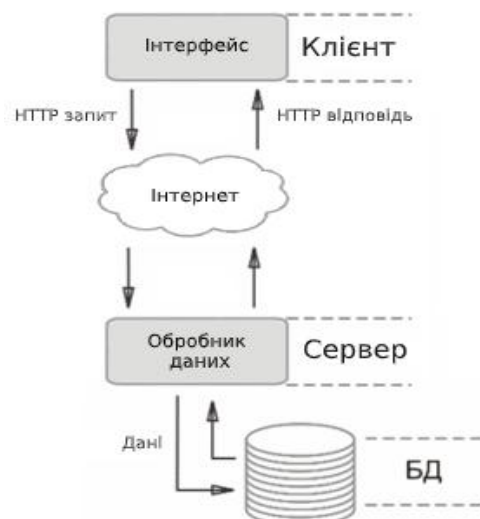


Рисунок 3.1 – Принцип функціонування веб-додатків

За способами розробки веб-додатки розділяють на два основних типи: односторінкові (SPA) та багатосторінкові (MPA). Розглянемо відмінності між ними, а також переваги і недоліки кожного типу.

Односторінкові додатки дозволяють імітувати роботу настільних програм, так як вони реагують на дії користувача без затримки. Концепція SPA полягає в тому, що всі ресурси, необхідні для роботи (елементи CSS, скрипти та ін.), знаходяться на одній сторінці і завантажуються при ініціалізації. При такому підході немає необхідності в перезавантаженні або завантаженні додаткових сторінок – контент оновлюється динамічно за допомогою JavaScript (AJAX-запитів) [47]. На рисунку 3.2 наведено принцип роботи односторінкового додатку.



Рисунок 3.2 – Принцип роботи односторінкового додатку

Переваги:

- при роботі завантажуються тільки необхідні дані (без повторюваних елементів і блоків), що дозволяє розвантажити ресурси сервера;
- відсутність повного перезавантаження сторінки підвищує швидкість роботи.
- код серверної частини (бекенду) можна повторно використати для створення власної мобільної програми.
- концепція SPA дозволяє ефективно кешувати будь-які дані. Програма

відправляє всього один запит, збирає дані, а після цього може функціонувати в offline-режимі.

Недоліки:

- початкове завантаження триває досить довго, через великий розмір JavaScript коду і допоміжних бібліотек.

- для коректної роботи потрібен JavaScript. Якщо користувач вручну відключить JavaScript в браузері, він не зможе повноцінно використовувати весь функціонал програми.

- через особливості архітектури більшість сторінок просто недоступні для сканування пошуковими роботами, що ускладнює SEO-оптимізацію.

Багатосторінкові додатки працюють за традиційною схемою. Це означає, що при незначній зміні даних чи переході на іншу сторінку браузер робить новий запит до сервера і знову завантажує всі ресурси, навіть ті компоненти, які повторюються на всіх сторінках (заголовки, нижній колонтитул, тощо). Таким чином, продуктивність витрачається на завантаження тих самих елементів. Принцип роботи даного підходу зображено на рисунку 3.3.

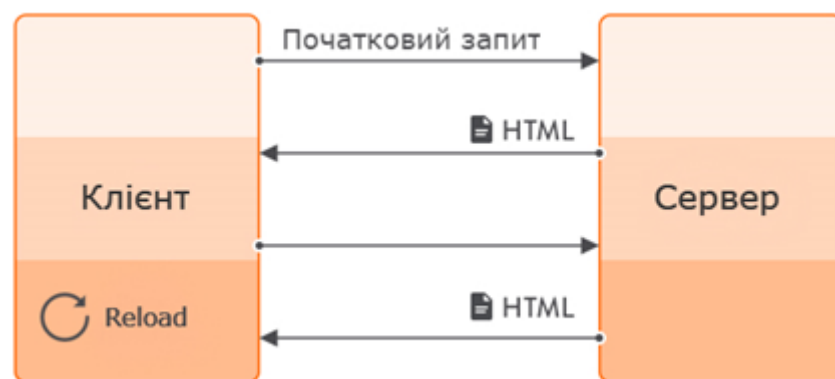


Рисунок 3.3 – Принцип роботи багатосторінкового додатку

Переваги:

- МРА архітектура дозволяє легко оптимізувати web-додаток для пошукових систем. Є можливість додати мета-теги для кожної сторінки окремо;

- як правило, для розробки багатосторінкових додатків потрібен менший стек технологій, таким чином їх вартість виходить дешевшою.

Недоліки:

- розробка як десктопної, так і мобільної версії займає значно більше часу;
- тісний зв'язок між бекендом і фронтендом ускладнює процес додавання нових функціональних можливостей.

Підвівши підсумки, можна зробити висновок, що кожна архітектура має свої переваги для проектів певного типу. Однак завдяки вищій швидкості роботи було прийнято рішення застосувати саме концепцію SPA при розробці системи.

### **3.2 Вибір фреймворків для розробки**

Розробка веб-додатку з нуля вимагає багато роботи. При цьому в більшості випадків доводиться витратити час на відтворення функцій, що вже були виконані тисячі разів. Фреймворки допомагають обійти цю проблему, надаючи свого роду каркас майбутньої програми та набір додаткових інструментів, покликаних прискорити роботу зі створення продукту.

Виходячи з того, що концепція SPA передбачає чітке розділення відповідальності між зовнішнім представленням (front-end) і внутрішньою реалізацією (back-end) варто окремо потурбуватися про вибір фремоворка як для серверної, так і для клієнтської частин.

Так, для визначення оптимального серверного фреймворку були встановлені наступні критерії:

- документація – наявність зрозумілої документації для швидкого старту;
- продуктивність – швидкість обробки вхідних запитів;
- безпека – ступінь захисту від веб-загроз;
- поріг освоєння – кількість часу, необхідного для освоєння фреймворка людиною з мінімальним досвідом у веб-розробці;

– гнучкість і масштабування – можливість впровадження допоміжних бібліотек для розширення базових можливостей.

Під продуктивністю тут мається на увазі швидкість обробки вхідних запитів, а гнучкість означає можливість впровадження допоміжних бібліотек для розширення базових можливостей.

Для порівняння було обрано 3 найпопулярніших back-end фреймворки: Django, Laravel та Ruby on Rails. За результатами проведеного аналізу можна зробити висновок, що кожен з цих фреймворків має гарні показники відносно таких важливих характеристик, як продуктивність і гнучкість. Однак в плані безпеки все ж лідирує Laravel. До його переваг також можна віднести використання PHP – найбільш популярної мови у сфері веб-розробки, що забезпечує її підтримку будь-якими хостинг-провайдерами. Тому для написання серверної частини CRM-системи було обрано Laravel [48].

Вибір фреймворку для клієнтської частини проходив за такими критеріями:

- простота – ступінь складності освоєння;
- документація – наявність детально описаного керівництва по використанню;
- швидкість рендерінгу, тобто кількість часу, необхідного для відображення на сторінці оновлених даних.

В даному випадку між собою порівнювались Vue.js, React та Angular. Виходячи з результатів аналізу React і Vue.js випереджають Angular за швидкістю роботи. Також Angular виявився більш складним в освоєнні. Інші два фреймворки чудово підходять для реалізації поставленої задачі. Але виходячи з особистого досвіду роботи при розробці клієнтської частини було використано саме Vue.js [49].

В наведених таблицях зазначено відповідність розглянутих фреймворків встановленим критеріям.

Таблиця 3.1 – Аналіз серверних фреймворків

	Django	Laravel	Ruby on Rails
Документація	+	+	–
Продуктивність	+	+	+
Безпека	–	+	–
Поріг освоєння	±	+	–
Гнучкість і масштабування	+	+	+

Таблиця 3.2 – Аналіз клієнтських фреймворків

	Vue.js	React	Angular
Простота	+	+	–
Документація	+	+	+
Швидкість рендерінгу	+	+	–

### 3.3 Архітектура системи

Вибір архітектури є одним із ключових рішень при проєктуванні інформаційно-аналітичної системи, оскільки визначає її масштабованість, гнучкість та можливості подальшого розвитку. Протягом останнього десятиліття в архітектурі ІАС спостерігається перехід від монолітних централізованих рішень до модульних, децентралізованих платформ з відкритими інтерфейсами обміну даними [50]. Найпоширенішими архітектурними підходами є тришарова архітектура, мікросервісна архітектура, а також гібридні моделі, які поєднують елементи обох підходів із використанням інтерфейсів REST для гнучкої взаємодії між компонентами системи.

Тришарова архітектура є класичним підходом до побудови ІАС, що передбачає поділ системи на три логічно незалежні рівні:

- Презентаційний рівень відповідає за взаємодію з користувачем через графічний інтерфейс (веб або мобільний застосунок). Цей рівень надсилає запити до серверу та відображає отримані результати.

- Рівень бізнес-логіки відповідає за обробку запитів і прийняття рішень на основі введених даних. На цьому рівні реалізуються основні правила обробки даних – те, як інформація має бути перетворена, відфільтрована, агрегована відповідно до потреб користувача.

- Рівень доступу до даних відповідає за зберігання, пошук, оновлення та видалення даних у базах (SQL, NoSQL тощо).

Перевагами тришарової архітектури є чітке розділення відповідальності між рівнями (інтерфейс, логіка, дані), що спрощує розробку, тестування та підтримку системи. Однак цей підхід має обмеження у масштабованості та гнучкості в умовах зростаючих обсягів даних і підвищених вимог до швидкості їх обробки.

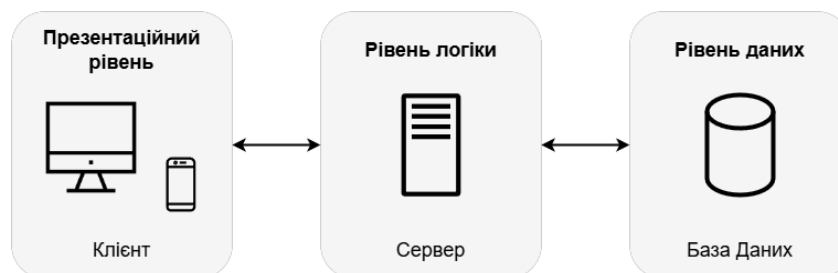


Рисунок 3.4 – Тришарова архітектура

Мікросервісна архітектура базується на ідеї поділу системи на набір незалежних сервісів, кожен з яких виконує одну конкретну бізнес-функцію і взаємодіє з іншими через чітко визначені інтерфейси (зазвичай REST або GraphQL). Кожен мікросервіс може бути реалізований різними мовами програмування, а також мати свою базу даних. Цей підхід є більш гнучким і

краще масштабується, оскільки є можливість проводити розробку і оновлення модулів незалежно один від одного. Однак через таку розподілену структуру значно зростає складність впровадження та підтримки такої системи, оскільки потрібно налаштування взаємодію між усіма сервісами та слідкувати за їх роботою окремо [51].

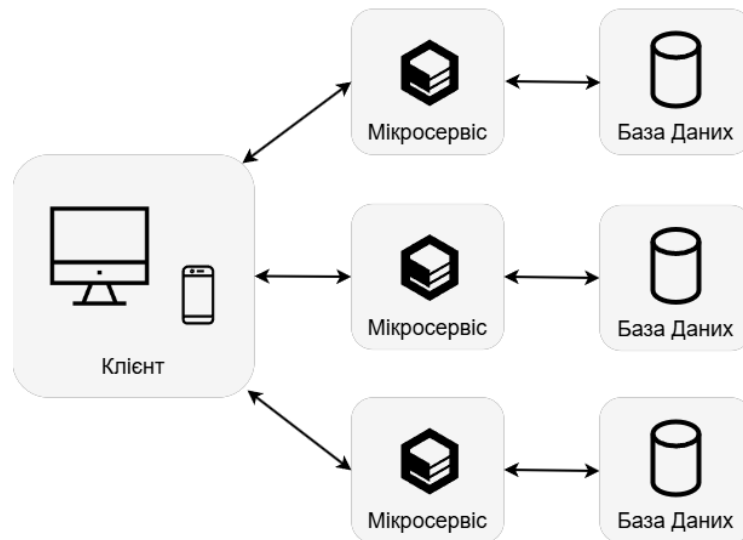


Рисунок 3.5 – Мікросервісна архітектура

Гібридні архітектури об'єднують кращі риси попередніх підходів, використовуючи мікросервіси для певних функцій, де потрібно забезпечити високу масштабованість та незалежність, але зберігаючи централізовану модель для інших частин системи. Такі архітектури застосовуються, коли потрібно поєднати переваги масштабованості мікросервісів із зручністю управління та підтримки централізованої моделі.

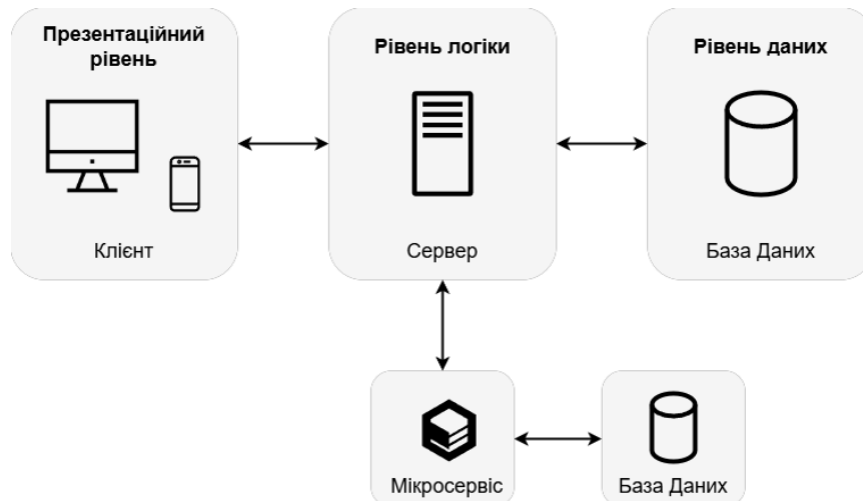


Рисунок 3.6 – Гібридна архітектура

До особливостей гібридних архітектур також належить підтримка потокової (streaming) аналітики - тобто обробки даних у режимі реального часу або з мінімальною затримкою, одразу в момент їх надходження. Замість накопичення даних у сховищі для подальшого аналізу, система миттєво реагує на події і, наприклад, може автоматично оновлювати аналітичні звіти, чи надсилати сповіщення відповідальним особам [52].

Враховуючи переваги гібридного підходу, система була побудована за архітектурою, що поєднує монолітний веб-застосунок (Laravel + Vue.js SPA) та мікросервіс машинного навчання (Python/Flask). Такий підхід забезпечує простоту розгортання та підтримки основного застосунку, незалежне масштабування та оновлення ML-компоненту, а також можливість використання спеціалізованих технологій для кожного компоненту.

Клієнтська частина розроблена у вигляді односторінкового додатка (SPA) із використанням фреймворку Vue.js, що забезпечує динамічну взаємодію користувача з даними, зокрема з інтерактивною картою пошкоджень. Серверна частина реалізована на PHP-фреймворку Laravel, що виконує роль API-шлюзу: відповідає за авторизацію, маршрутизацію запитів, обробку бізнес-логіки та формування відповіді у форматі JSON.

Модуль прогнозування вартості відновлення та інтерпретації прогнозів винесено в окремий мікросервіс на Python. Таке архітектурне рішення зумовлене необхідністю використання спеціалізованих бібліотек машинного навчання (scikit-learn, XGBoost, SHAP, Pandas, NumPy) [53], які є стандартом у цій галузі та не мають повноцінних аналогів у PHP-екосистемі.

Мікросервіс побудований за принципом багатосарової архітектури з чітким розділенням відповідальностей:

1. Шар REST API – реалізований на фреймворку Flask, забезпечує прийом HTTP-запитів, валідацію вхідних даних та формування JSON-відповідей.
2. Шар обробки даних – відповідає за кодування ознак: one-hot кодування для номінальних категоріальних змінних та fuzzy-кодування з трикутними функціями належності для порядкових ознак (методи кодування описано у підрозділах 2.2.1–2.2.2).
3. Шар моделей ML – містить попередньо натреновані моделі машинного навчання та забезпечує виконання інференсу.
4. Шар даних – серіалізовані артефакти моделей, конфігураційні файли та навчальний датасет.

Мікросервіс надає два ендпоінти для прогнозування: базове прогнозування (POST /predict), що повертає числову оцінку вартості відновлення, та прогнозування з поясненням (POST /predict\_explain), що додатково повертає SHAP-внески кожної групи ознак у прогнозовану вартість. Структура вхідних даних включає шість ознак: площа об'єкта, кількість поверхів, тип будівлі, ступінь пошкодження, регіон та тип ремонту.

Для забезпечення якості вхідних даних до складу мікросервісу включено блок автоматизованої валідації, що виконує перевірку наявності обов'язкових ознак, допустимості значень та відповідності категоріальних параметрів заданим перелікам. Усі ендпоінти, окрім перевірки стану, захищені аутентифікацією через API-ключ.

Взаємодія між компонентами системи здійснюється через REST API. Клієнтська частина надсилає HTTP-запити до серверної частини Laravel, яка обробляє їх, взаємодіє з базою даних MySQL та за потреби перенаправляє запити до ML-мікросервісу для отримання прогнозів вартості відновлення. Результати повертаються клієнту у форматі JSON для відображення в інтерфейсі користувача. Структура побудованої інформаційно-аналітичної системи представлена на рисунку 3.7.



Рисунок 3.7 – Структура системи

Винесення обчислень у незалежний мікросервіс забезпечує технологічну ізоляцію компонентів системи та спрощує її супровід. Зокрема, модель прогнозування можна оновлювати або перенавчати без змін у основному веб-додатку, а розрахунки масштабувати незалежно від інших частин системи.

Особливу увагу приділено механізму розмежування прав доступу, що запобігає збереженню неперевіреної інформації. Таким чином, доступ до даних

про пошкодження та можливість внесення записів надається виключно уповноваженим особам.

### 3.4 Проектування REST API

У веб-додатку, розробленому за архітектурою SPA, серверна частина взаємодіє з клієнтом через обмежений набір кінцевих точок - API (Application Programming Interface). Клієнт надсилає запити до цих точок, а сервер обробляє їх і повертає відповідь.

Для створення RESTful API при розробці системи було використано фреймворк Laravel. Позначення "RESTful" означає, що інтерфейс дотримується принципів REST. У цьому підході тіло запиту та відповіді від сервера передається у стандартному текстовому форматі, заснованому на синтаксисі JavaScript Object Notation (JSON).

```
{
  "id": 1,
  "object_category_id": 1,
  "name": "Багатоповерховий будинок",
  "created_at": "2025-02-01T19:00:19.000000Z",
  "updated_at": "2025-02-01T19:00:19.000000Z",
  "object_category": {
    "id": 1,
    "name": "Житловий фонд",
    "created_at": "2025-02-01T19:00:19.000000Z",
    "updated_at": "2025-02-01T19:00:19.000000Z"
  }
}
```

Згідно з архітектурними принципами REST, основні операції з сутностями реалізуються щонайменше чотирма HTTP-методами: отримання даних (GET), збереження (POST), оновлення (PUT/PATCH) та видалення (DELETE). Окрім цього, REST також передбачає обробку помилок у запитах, контроль доступу та валідацію вхідних даних [54]. Структура HTTP-методів, використаних у процесі розробки, наведена в таблиці 3.3.

Таблиця 3.3 – HTTP-методи, що були використані при розробці системи

Метод	Операція	Тіло запиту	Тіло відповіді
GET	Отримати інформацію	NA	JSON об'єкт
POST	Створити новий запис	JSON об'єкт	JSON об'єкт
PUT/PATCH	Оновити запис	JSON об'єкт	JSON об'єкт
DELETE	Видалити запис	NA	JSON об'єкт

Після завершення проєктування JSON-об'єктів необхідно створити унікальні URL-адреси для кожної сутності, що дозволяють ідентифікувати REST-запити. Зазвичай такі адреси мають формат "назва сутності/ідентифікатор", де елементи розділяються символом "/". Ідентифікатор, який виконує роль первинного ключа для визначення конкретного об'єкта, є необов'язковим параметром.

На рис. 3.8 представлено структуру кінцевих точок, що використовуються клієнтською частиною для взаємодії з сервером на прикладі даних про пошкодження об'єктів нерухомості.

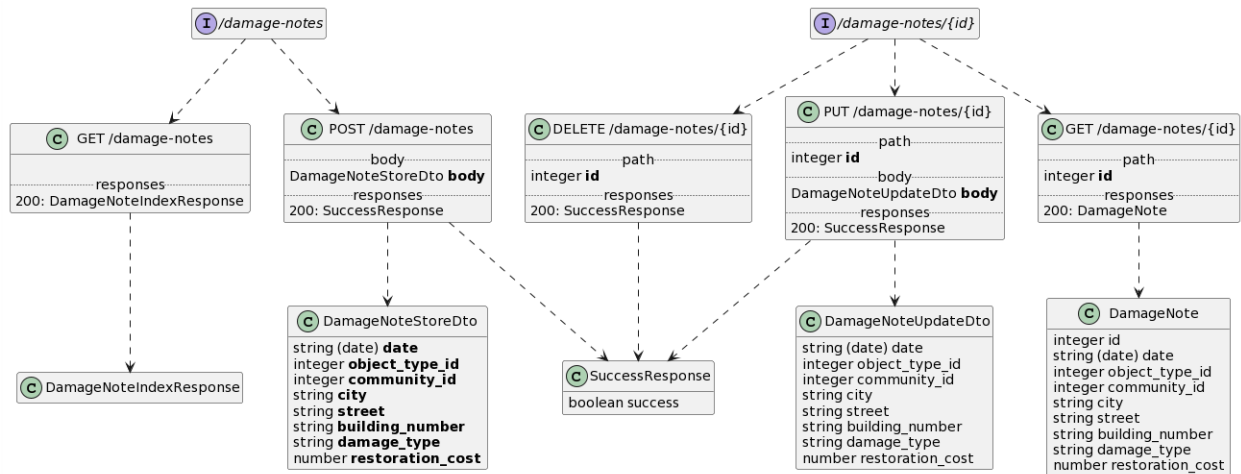


Рисунок 3.8 – Кінцеві точки для роботи з даними про пошкодження об’єктів

Кінцеві точки API згруповано за функціональними модулями системи. Перша група – довідники – охоплює маршрути для отримання переліків областей, районів, громад, типів об’єктів та типів ремонту (`GET /api/regions`, `GET /api/object-types` тощо). Ці маршрути є публічними та не потребують аутентифікації, оскільки довідникові дані використовуються для заповнення форм і побудови фільтрів. Друга група – облік пошкоджень – включає CRUD-операції для записів про пошкоджені об’єкти та процес обробки заявок (створення, схвалення, відхилення). Маршрути цієї групи захищені аутентифікацією через `Laravel Sanctum` і доступні лише користувачам з відповідними ролями. Третя група – аналітика – об’єднує маршрути для агрегованих запитів: статистику пошкоджень за часовими періодами, розподіл за типами об’єктів та OLAP-зрізи за кількома вимірами одночасно.

Окрему групу кінцевих точок становлять маршрути взаємодії з ML-мікросервісом. Клієнтська частина не звертається до Python-мікросервісу напряму – замість цього `Laravel`-додаток виконує роль проксі. Коли користувач ініціює прогнозування або сценарний перерахунок, `Vue`-клієнт надсилає запит до `Laravel` (`POST /api/predict-restoration-cost`), серверна частина формує `payload` з характеристиками об’єкта та масивом інфляційних індексів з бази даних, а потім перенаправляє його до `Flask`-мікросервісу (`POST /predict` або `POST`

/predict\_explain). Мікросервіс повертає прогнозовану вартість  $i$ , за потреби, SHAP-внески кожної групи ознак, які Laravel передає клієнту у стандартному JSON-форматі. Така схема дозволяє централізувати валідацію, аутентифікацію та логування на рівні основного додатку, не розкриваючи адресу мікросервісу зовнішнім клієнтам.

### 3.5 Проектування бази даних

Центральним елементом системи є база даних, проектуванню якої було приділено особливу увагу через необхідність обробки значних обсягів інформації. З огляду на це, під час розробки її структури були застосовані механізми нормалізації [55].

Структуру бази даних приведено до третьої нормальної форми (3НФ), що передбачає усунення транзитивних залежностей між неключовими атрибутами. Розглянемо, як це реалізовано на прикладі основних груп таблиць.

Адміністративно-територіальну ієрархію винесено у три окремі довідникові таблиці – regions, districts, communities – замість збереження назв області, району та громади безпосередньо у записі про пошкодження. Між ними діють зв'язки «один-до-багатьох»: кожна область містить кілька районів, кожен район – кілька громад. Якби ці назви зберігалися як текстові поля в основній таблиці, зміна назви району (наприклад, внаслідок адміністративної реформи) потребувала б оновлення всіх записів, де ця назва фігурує. Винесення у довідник зводить таке оновлення до зміни одного рядка.

Аналогічно, класифікацію об'єктів розділено на два рівні: object\_categories (верхній рівень – житлова нерухомість, соціальна інфраструктура тощо) та object\_types (конкретний тип – багатоквартирний будинок, школа, лікарня). Зв'язок між ними також «один-до-багатьох»: одна категорія об'єднує кілька типів. Такий поділ дозволяє будувати аналітичні звіти як за окремими типами об'єктів, так і за укрупненими категоріями, не дублюючи назву категорії в

кожному записі. Типи ремонту винесено в окрему таблицю `repair_types` з тієї ж причини – набір допустимих значень (поточний ремонт, капітальний ремонт, реконструкція) є фіксованим довідником, а не довільним текстом.

Така декомпозиція усуває дублювання текстових значень, спрощує оновлення довідників і забезпечує узгодженість даних при формуванні аналітичних зрізів.

Для формування бази даних визначено такі основні сутності: «Користувач», «Область», «Район», «Громада», «Категорія об'єкта», «Тип об'єкта», «Тип ремонту», «Запис про пошкодження», «Зображення запису», «Заявка на модерацію», «Віртуальний тур» та «Індекс інфляції». Для кожної з них створено окрему таблицю з необхідними атрибутами.

Довідникові таблиці адміністративно-територіальної ієрархії: `regions` (поля: `id`, `name`), `districts` (поля: `id`, `region_id`, `name`) та `communities` (поля: `id`, `district_id`, `name`), а також таблиці класифікації об'єктів: `object_categories` (поля: `id`, `name`), `object_types` (поля: `id`, `object_category_id`, `name`) та `repair_types` (поля: `id`, `name`) – описані вище в контексті нормалізації.

Таблиця `users` зберігає дані про зареєстрованих користувачів системи (поля: `id`, `name`, `email`, `password`, `api_token`, `region_id`, `district_id`, `community_id`, `deleted_at`). Прив'язка користувача до адміністративно-територіальної одиниці через зовнішні ключі дозволяє обмежити область видимості даних відповідно до його компетенції. Розмежування прав доступу реалізовано засобами пакету `Spatie Laravel Permission` через таблиці відповідності ролей та дозволів.

Центральною сутністю є таблиця `damage_notes`, яка зберігає відомості про пошкоджені об'єкти. Її структура містить: ідентифікатор запису (`id`), дату фіксації пошкодження (`date`), зовнішні ключі на тип об'єкта (`object_type_id`), громаду (`community_id`) та вид ремонту (`repair_type_id`), адресні поля (`city`, `street`, `building_number`), характеристики будівлі (`floors`, `area`), ступінь пошкодження (`damage_type` – перелічуваний тип зі значеннями `low`, `medium`, `high`), оціночну вартість відновлення з проєктно-кошторисної документації (`restoration_cost`),

прогнозовану вартість, сформовану ML-моделлю (`predicted_restoration_cost`), та текстовий коментар (`comment`).

До кожного запису може бути прикріплено довільну кількість фотофактів через таблицю `damage_note_images` (поля: `id`, `damage_note_id`, `file_name`, `hash_file_name`).

Модерація записів реалізована через таблицю `damage_note_requests`, яка пов'язує запис із заявником (`full_name`, `email`, `phone`), фіксує автора створення (`creator_id`) та модератора (`approver_id`, `approver_comment`) і зберігає мітки часу ухвалення або відхилення заявки (`approved_at`, `declined_at`).

Таблиця `virtual_tours` забезпечує збереження матеріалів віртуальних турів для об'єктів культурної спадщини (поля: `id`, `damage_note_id`, `title`, `description`, `image_file_name`, `audio_file_name`).

Таблиця `inflation_indices` зберігає помісячні індекси інфляції (поля: `id`, `year`, `month`, `index_value`, `source_type`, `source_description`) та використовується модулем інфляційного коригування для перерахунку вартості відновлення відносно базового місяця.

Загалом база даних системи налічує 18 таблиць, які можна розділити на чотири функціональні групи: довідники адміністративно-територіального устрою та класифікації об'єктів, таблиці предметної області (записи про пошкодження з пов'язаними зображеннями, заявками та віртуальними турами), таблиці підсистеми управління доступом та таблиці інфляційного коригування. Повну структуру бази даних із зазначенням зв'язків між таблицями наведено на рис. 3.9.



## 3.6 Аналітичний блок системи

### 3.6.1 Інтерактивна карта руйнувань

Карта руйнувань (рис. 3.10) забезпечує наочне відображення пошкоджень соціальної та житлової інфраструктури, заподіяних росією. В залежності від обраного масштабу є можливість оцінити руйнування як у межах всієї України, так і для кожної області окремо.

Карта має три рівні деталізації: спочатку демонструються загальні дані по регіонах, а при збільшенні масштабу можна отримати детальнішу інформацію щодо окремих районів і територіальних громад. Для візуального позначення ступеня ушкоджень використовується кольорова градація – від світлих до темніших відтінків, що відображають рівень руйнувань: від нешкоджених територій до найбільш постраждалих зон.

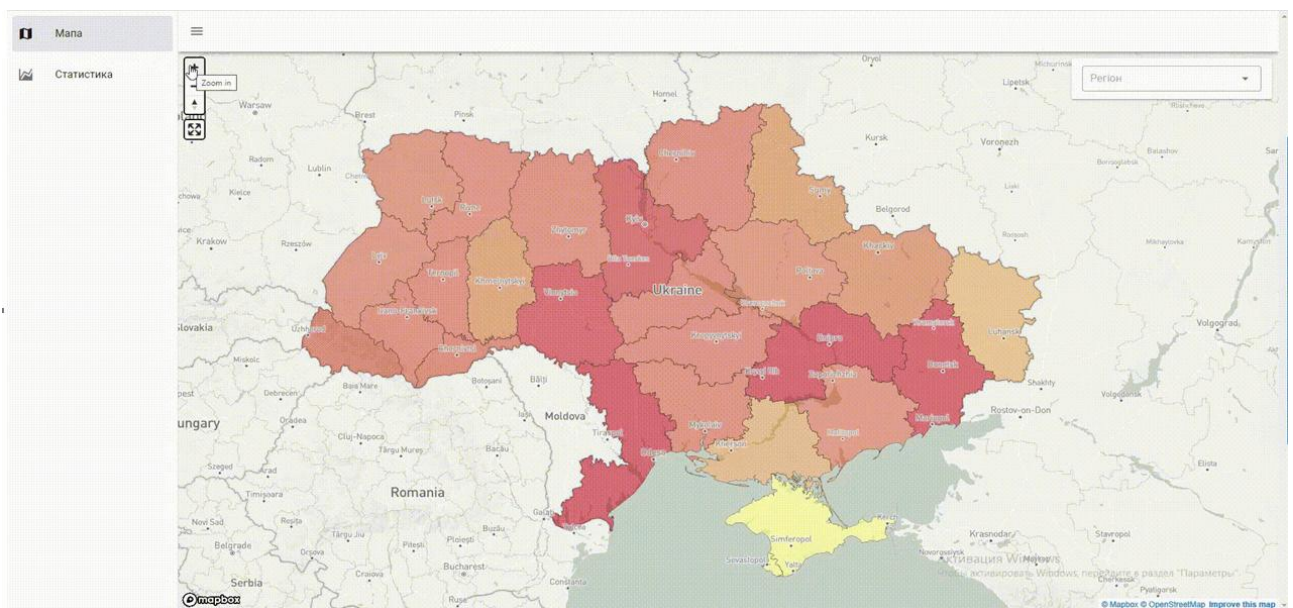


Рисунок 3.10 – Інтерактивна карта руйнувань

### 3.6.2 Графічна аналітика

Модуль графічної аналітики призначений для детальної оцінки рівня пошкоджень та надає можливість візуалізувати дані у вигляді інтерактивних графіків і діаграм. Для побудови візуалізацій використано бібліотеку Chart.js, інтегровану з Vue.js-компонентами.

Модуль забезпечує виконання таких аналітичних операцій:

1. Генерація графічних звітів у часовому розрізі. Система дозволяє будувати графіки динаміки вартості відновлення або кількості пошкоджених об'єктів із різною деталізацією: по днях, тижнях або місяцях. Це дає змогу відстежувати тенденції та виявляти періоди найбільшої інтенсивності руйнувань.

2. Візуалізація розподілу за категоріями. Кругові та кільцеві діаграми відображають співвідношення пошкоджених об'єктів за категоріями (житловий фонд, нежитловий фонд, природні ресурси) або за типами об'єктів у межах обраної категорії. Такий розподіл дозволяє визначити, які типи інфраструктури постраждали найбільше.

3. Гнучка система фільтрації. Користувач може обмежити вибірку даних за обраним періодом часу, регіоном, районом або громадою. Фільтри застосовуються динамічно без перезавантаження сторінки, що забезпечує оперативний аналіз даних у різних розрізах.

4. Вибір метрики відображення. Система підтримує два основних показники: кількість пошкоджених об'єктів та оціночна вартість їх відновлення. Користувач може переключатися між метриками для отримання різних аналітичних зрізів.

Інтерфейс модуля графічної аналітики складається з двох основних блоків: панелі фільтрів у верхній частині екрану та області візуалізації, що містить графік динаміки (стовпчикова або лінійна діаграма) та кругову діаграму розподілу за типами. Графіки оновлюються автоматично при зміні параметрів фільтрації [56].

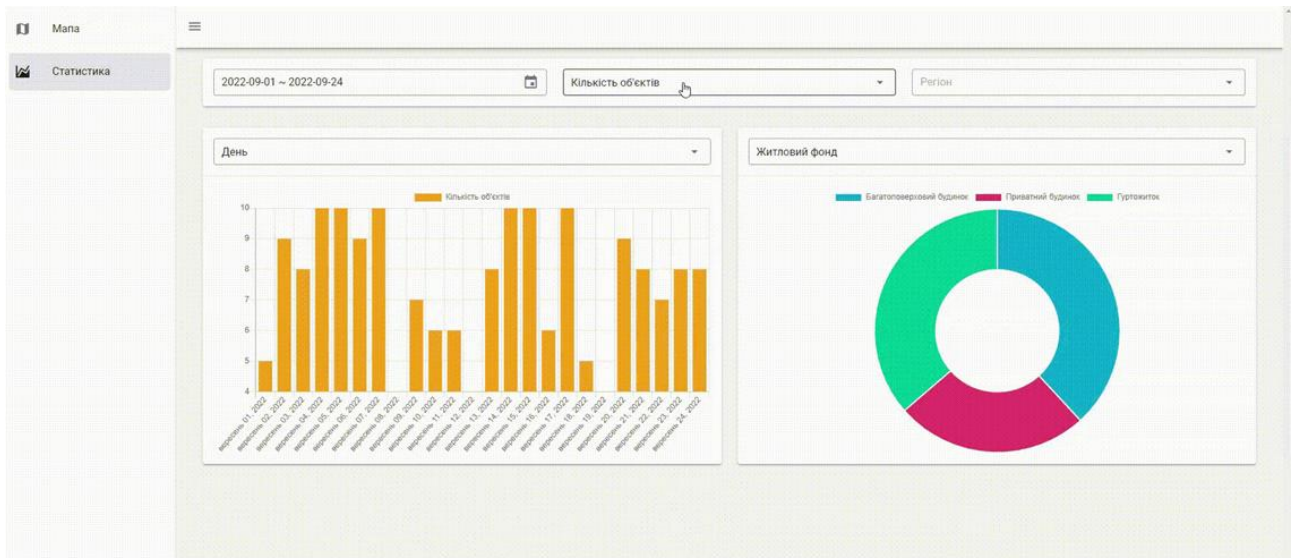


Рисунок 3.11 – Аналітичний модуль

### 3.6.3 OLAP-куб

Окрім графічних звітів, аналітичний блок системи містить OLAP-куб (Online Analytical Processing) – інструмент багатовимірного аналізу даних про кількість пошкоджених об'єктів та витрати на їх відновлення. На відміну від графічних звітів, що візуалізують дані в одному розрізі, OLAP-куб дозволяє оперативно переключатися між різними вимірами та отримувати агреговані значення метрик для кожного з них [57].

Розроблений OLAP-куб охоплює такі виміри (dimensions):

- Часові виміри. OLAP-куб включає часові виміри, такі як день, тиждень і місяць, що дозволяє проаналізувати тенденції пошкоджень у різні періоди.
- Категорія об'єкта. Пошкодження класифіковані за трьома основними категоріями: «Житловий фонд», «Нежитловий фонд» та «Природні ресурси». Кожна категорія допомагає визначити, які типи інфраструктури постраждали найбільше.
- Тип об'єкта. Конкретні типи в кожній категорії додають деталізації. Наприклад, у категорії «Житловий фонд» є багатоповерхові будинки, приватні будинки та гуртожитки; у категорії «Нежитловий фонд» - об'єкти критичної

інфраструктури, адміністративні будівлі, бізнес-центри тощо; у категорію «Природні ресурси» включено ліси, поля та водні ресурси.

- Географічне положення. OLAP-куб охоплює кілька географічних вимірів в межах України, таких як регіон, район і територіальна громада. Ця структура допомагає локалізувати та категоризувати пошкодження за адміністративними поділами.

- Рівень пошкодження. Система розрізняє незначні, середні та повні руйнування, надаючи уявлення про ступінь пошкодження об'єктів.

Метриками OLAP-кубу (measures) є:

- Кількість пошкоджених об'єктів - ця метрика визначає масштаби руйнувань у кожному вимірі, що дозволяє краще розуміти вплив на інфраструктуру.

- Вартість відновлення - ця метрика оцінює фінансові витрати на відбудову пошкодженої інфраструктури. Поєднуючи цю метрику з географічними та категорійними вимірами, система допомагає пріоритезувати розподіл фінансування та планування ресурсів.

OLAP-куб є безпосередньою частиною систему та був реалізований засобами PHP фреймворку Laravel, як і інші модулі. За відображення даних кубу відповідає таблиця з трьома колонками: показник виміру (назва показнику в залежності від виміру, що розглядається), кількість об'єктів та вартість відновлення (агреговані значення для кожного з показників). Для зміни виміру у верхній частині таблиці реалізовано відповідний перемикач. Поряд з цим перемикачем додатково було реалізовано фільтр по даті, що дозволяє обрати конкретний часовий діапазон для аналізу. Зовнішній вигляд таблиці та фільтрів представлено на рис. 3.12.

Показник виміру	Кількість об'єктів	Вартість відновлення
Багатоповерховий будинок	1	4161086.00
Приватний будинок	1	14334168.00
Гуртожиток	1	10891874.00
Адміністративна будівля	1	17976145.00
Бізнес-центр	1	5248934.00
ТРЦ	1	18678917.00
Інше	1	7274231.00

Елементів на сторінці: 15 1-737

Рисунок 3.12 – Реалізація OLAP-кубу

OLAP-куб у системі фіксації руйнувань є незамінним інструментом для аналізу складних, багатовимірних даних, оскільки підтримує швидкий та динамічний аналіз, дозволяючи фільтрувати та пріоритезувати записи про пошкодження за різними вимірами [58].

### 3.7 Облік пошкоджених об'єктів

Модуль обліку даних про пошкодження призначений для реєстрації, збереження та управління записами про завдані збитки [59]. Він забезпечує повний цикл роботи з даними та надає можливість виконувати такі операції:

1. Перегляд списку записів – відображення переліку зареєстрованих пошкоджень у вигляді таблиці з можливістю сортування за датою, типом пошкодження, вартістю відновлення та іншими параметрами. Для полегшення роботи з великими обсягами даних реалізовано посторінкову навігацію.

2. Додавання нового запису – внесення інформації про пошкоджений об'єкт до бази даних через спеціалізовану форму.

3. Редагування записів – можливість коригування раніше внесених даних адміністраторами системи.

4. Видалення записів – вилучення помилкових або неактуальних записів з бази даних.

5. Експорт даних – вивантаження записів у форматі CSV для подальшого аналізу в зовнішніх програмах, зокрема Microsoft Excel.

6. Імпорт даних – завантаження кількох записів одночасно з файлу Excel, що суттєво прискорює процес наповнення бази даних.

ID	Дата	Тип об'єкта	Територіальна громада	Місто / село/за	Вулиця	Відомості	Тип пошкодження	Вартість відновлення
8	2022-02-25	Багатоповерховий будинок	Каблянець-Подільська міська громада	Lake Helenamouth	Renner Tunnel	7838	Слабке руйнування	6278590.00
25	2022-02-28	Багатоповерховий будинок	Чинадівська сільська громада	Pagamouth	Walsh Row	80983	Слабке руйнування	19551839.00
44	2022-03-02	Багатоповерховий будинок	Радомишльська міська громада	Stakesbury	Rosenbaum Station	1445	Слабке руйнування	1100946.00
45	2022-03-02	Багатоповерховий будинок	Чечельницька сільська громада	New Jadenberg	Brandi Via	697	Сильне руйнування	1528339.00
50	2022-03-03	Багатоповерховий будинок	Великомиколаївська сільська громада	South Estelville	Lori Square	36404	Повне руйнування	14400043.00
67	2022-03-05	Багатоповерховий будинок	Ободівська сільська громада	North Taniaport	Becker Gateway	31593	Сильне руйнування	12318780.00
75	2022-03-06	Багатоповерховий будинок	Ушмирська сільська громада	Lake Rheaton	Zelda Avenue	968	Сильне руйнування	1075951.00
90	2022-03-06	Багатоповерховий будинок	Берездська міська громада	West Friedrich	Zboncak Key	9989	Слабке руйнування	7207313.00
97	2022-03-10	Багатоповерховий будинок	Миколаївська міська громада	Hermistonbury	Jennyfer Trail	2382	Слабке руйнування	12022986.00
102	2022-03-11	Багатоповерховий будинок	Наравська сільська громада	Gradyville	Dylan Wells	57488	Слабке руйнування	21501310.00

Рисунок 3.13 – Відомості про пошкодження

Форма внесення записів підтримує два способи додавання даних: обробка кожного запису окремо через інтерфейс системи або одночасне завантаження кількох записів через імпорт із файлу.

При ручному внесенні запису необхідно заповнити такі поля: дата фіксації пошкодження; категорія об'єкта (житловий фонд, нежитловий фонд); тип об'єкта (багатоповерховий будинок, приватний будинок, гуртожиток, адміністративна будівля тощо); територіальна громада; населений пункт; вулиця та номер будинку; кількість поверхів; площа об'єкта; ступінь пошкодження (незначне,

середнє, повне руйнування); тип ремонту; оціночна вартість відновлення; коментар; фотографії пошкодженого об'єкта.

Form for recording damage records:

- ЗАВАНТАЖИТИ ФАЙЛ
- Дата пошкодження: 2026-04-15
- Категорія об'єкта
- Тип об'єкта
- Територіальна громада
- Місто / селище
- Вулиця
- Номер будівлі
- Кількість поверхів
- Площа (м<sup>2</sup>) м<sup>2</sup>
- Тип пошкодження
- Тип ремонту
- Вартість відновлення
- Коментар
- Photo upload area (+)
- ЗБЕРЕГТИ

Рисунок 3.14 – Форма внесення записів

Особливу увагу приділено механізму розмежування прав доступу. Перегляд списку записів доступний авторизованим користувачам з роллю «Аналітик» та вище. Створення, редагування та видалення записів дозволено лише користувачам з роллю «Адміністратор» або «Суперадміністратор». При цьому адміністратори мають доступ лише до даних у межах призначеної їм

території (регіону, району або громади), тоді як суперадміністратори працюють без територіальних обмежень.

Для забезпечення прозорості та залучення громадян до процесу фіксації руйнувань реалізовано публічну форму подання заявок. Будь-який користувач, навіть неавторизований, може надіслати інформацію про пошкоджений об'єкт, вказавши свої контактні дані (ПІБ, email, телефон) та характеристики пошкодження. Такі заявки потрапляють до черги на модерацію та очікують розгляду адміністратором. Після перевірки заявка може бути схвалена (дані переносяться до основної бази) або відхилена з відповідним коментарем.

### **3.8 Цифровий двійник об'єкта**

Ключовим елементом розробленої інформаційно-аналітичної системи є цифровий двійник пошкодженого об'єкта, який забезпечує прогнозування орієнтовної вартості відновлення, сценарний перерахунок при зміні характеристик об'єкта та інтерпретацію результатів прогнозу.

#### **3.8.1 Цифровий профіль об'єкта**

Центральним компонентом цифрового двійника є модуль ведення цифрового профілю, що формує структуровану картку конкретного об'єкта та забезпечує повний цикл роботи з ним: створення запису, його уточнення та підтримку актуальної інформації.

Під час реєстрації об'єкта користувач вносить його паспортні характеристики (адреса, тип будівлі, площа, поверховість), фіксує ступінь пошкодження, додає фотофакти та текстові коментарі. У результаті формується структурована картка об'єкта, яка відображає його поточний стан і слугує базою для подальших розрахунків.

Стрийська 45, Львів			
<ul style="list-style-type: none"> <li><b>ЗАГАЛЬНА ІНФОРМАЦІЯ</b></li> <li>ВАРТІСТЬ ВІДНОВЛЕННЯ</li> <li>ФІНАНСУВАННЯ</li> <li>ГАЛЕРЕЯ</li> </ul>	Дата пошкодження	Категорія об'єкта	Тип об'єкта
	2025-12-27	Житловий фонд	Багатоповерховий будинок
	Регіон	Район	Територіальна громада
	Львівська область	Львівський район	Львівська міська громада
	Місто / селище	Вулиця	Будівля
	Львів	Стрийська	45
	Кількість поверхів	Площа (м²)	
9	7200.00 м²		
Тип пошкодження	Тип ремонту		
Середнє	Капітальний		
Оціночна вартість відновлення	Прогнозована вартість відновлення		
<b>48 600 000,00 грн</b>	<b>37 750 956,00 грн</b>		

Рисунок 3.15 – Інтерфейс цифрового профілю об'єкта нерухомості

Для забезпечення актуальності цифрового профілю реалізовано механізм редагування запису при надходженні нової інформації, наприклад після повторного технічного обстеження або уточнення паспортних даних. Редагування здійснюється через окрему форму оновлення цифрового профілю, у якій користувач може змінити ключові характеристики об'єкта, додати нові фотофакти чи коментарі, а також скоригувати ступінь пошкодження або тип запланованих робіт.


Дата пошкодження 2024-03-12	Категорія об'єкта Житловий фонд	Тип об'єкта Багатоповерховий...
Територіальна громада Львівська міська гром	Місто / селище Львів	Вулиця Стрийська
Номер будівлі 45	Кількість поверхів 9	Площа (м²) 7200 м²
Тип пошкодження Середнє	Тип ремонту Капітальний	Вартість відновлення € 48600000
Коментар Пошкоджено фасад і частково покрівлю внаслідок вибухової хвилі. Вибито вікна у під'їздах та кількох квартирах. Частково пошкоджено інженерні мережі (водопостачання).		
		
<input type="button" value="ЗБЕРЕГТИ"/>		

Рисунок 3.16 – Форма оновлення цифрового профілю

### 3.8.2 Сценарний розрахунок та інтерпретація прогнозів

На основі цифрового профілю працює модуль сценарного розрахунку, який перетворює двійник на інструмент варіативного планування відбудови. Він дозволяє змінювати одну або кілька характеристик об'єкта, що впливають на кошторис, і отримувати перераховану орієнтовну вартість відновлення.

Модуль реалізує типові сценарії, визначені у підрозділі 2.6: зміну типу ремонту, уточнення ступеня пошкодження та коригування обсягу робіт. Користувач змінює одну або кілька характеристик об'єкта у формі сценарного розрахунку та отримує оновлений прогноз вартості.

Технічна реалізація сценарного розрахунку передбачає надсилання HTTP-запиту до ML-мікросервісу з параметрами обраного сценарію. Мікросервіс

виконує кодування ознак, інференс моделі XGBoost та повертає прогнозовану вартість разом із SHAP-внесками для кожної групи ознак. Інтерфейс модуля сценарного розрахунку представлено на рисунку 3.17.

Категорія об'єкта Житловий фонд	
Тип об'єкта Багатоповерховий будинок	
Територіальна громада Львівська міська громада	
Кількість поверхів 9	
Площа (м <sup>2</sup> ) 7200	м <sup>2</sup>
Тип пошкодження Легке	
Тип ремонту Поточний	

---

Оціночна вартість відновлення	Прогнозована вартість відновлення
<b>48 600 000,00 грн</b>	<b>9 142 633,00 грн</b>
<a href="#">РОЗРАХУВАТИ</a>	

Рисунок 3.17 – Інтерфейс модулю сценарного розрахунку

Для забезпечення пояснюваності результатів реалізовано модуль інтерпретації на основі методу SHAP (описаного у підрозділі 2.5). Система повертає вектор внесків, який показує, як кожна характеристика об'єкта збільшує або зменшує прогнозовану вартість. Для наочного представлення внесків в інтерфейсі цифрового двійника розраховується відносна частка впливу кожної групи ознак, що слугує основою для побудови кругової діаграми (рис. 3.18). Це дозволяє швидко визначити, які параметри об'єкта (наприклад, ступінь пошкодження або тип ремонту) роблять найбільший внесок у формуванні оцінки вартості відновлення.

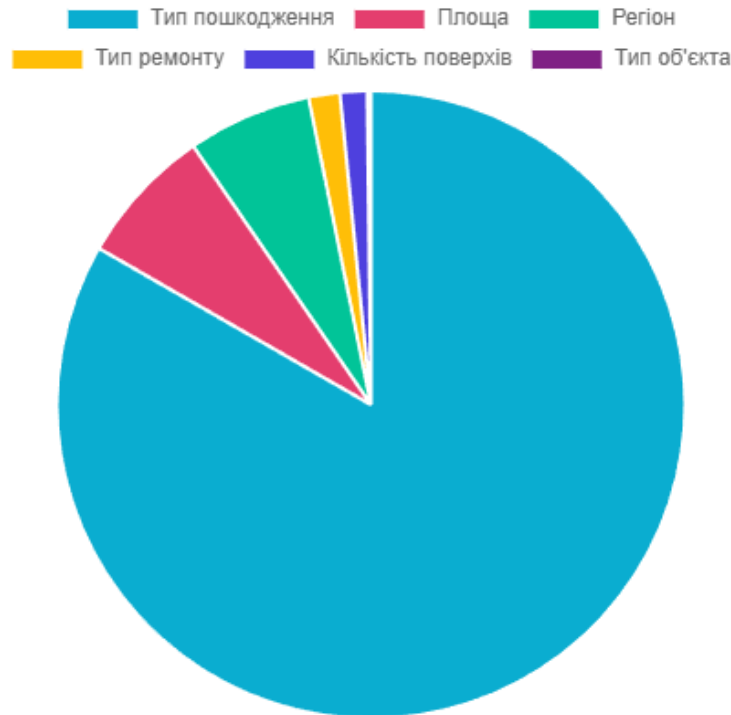


Рисунок 3.18 – Візуалізація внесків груп ознак у прогнозовану вартість

Розглянемо гіпотетичне застосування цифрового двійника до ситуації, аналогічної випадку будинку по вул. Паралельній, 4 у м. Запоріжжі. Користувач створює цифровий профіль об'єкта з такими характеристиками: рік побудови – 1957, тип конструкції – складна конфігурація, ступінь пошкодження – тяжкий (категорія «4»). Система визначає, що об'єкт не належить до культурної спадщини, і формує множину допустимих альтернатив  $A^*(h) = \{a_1, a_2, a_3, a_4, a_5\}$ . Для кожної альтернативи ML-модель генерує прогноз вартості  $V$ , а SHAP-аналіз визначає ознаки з найбільшим внеском. Далі користувач може виконати сценарний перерахунок: наприклад, змінити ступінь пошкодження з «тяжкого» на «середній» (моделюючи ситуацію, коли частина конструкцій виявиться збереженою після детальнішого обстеження) і порівняти, чи стає капітальний ремонт ( $a_1$ ) дешевшим за знесення з новим будівництвом ( $a_4$ ). Якщо навіть за оптимістичного сценарію вартість ремонту перевищує вартість нового будівництва, система надає ОПР обґрунтовану рекомендацію на користь

демонтажу – результат, що збігається з фактичним рішенням міської ради у цьому випадку.

### 3.8.3 Інфляційне коригування вартості відновлення в інтерфейсі цифрового двійника

Механізм інфляційного коригування, описаний у підрозділі 2.3, реалізований безпосередньо в інтерфейсі цифрового двійника. Він активується після виконання базового розрахунку: щойно модель повертає прогнозовану вартість  $V$ , користувачу стає доступний блок вибору планового місяця початку робіт – два пов’язані селектори «рік» та «місяць». Місяць залишається заблокованим до обрання року, що унеможлиблює некоректне заповнення форми.

Коефіцієнти  $k(t)$  зберігаються у таблиці `inflation_indices` бази даних системи та передаються до ML-мікросервісу разом із кожним запитом. Мікросервіс застосовує коефіцієнт до базового прогнозу та повертає як  $V$ , так і  $V(t)$ . Такий розподіл відповідальності забезпечує єдине джерело істини: актуальність значень визначається таблицею в базі даних, яку адміністратор може оновлювати в міру публікації нових офіційних даних Держстату.

При виборі місяця система виконує один запит до ендпоінту сценарного порівняння, який повертає значення  $V(t)$  одразу для всіх доступних місяців. На основі цих даних будується стовпчикова діаграма (рис. 3.19): по горизонтальній осі – місяці від базового до найвіддаленішого прогнозного, по вертикальній – скоригована вартість у гривнях. Обраний місяць підсвічується окремим кольором, решта відображаються нейтральним тоном.

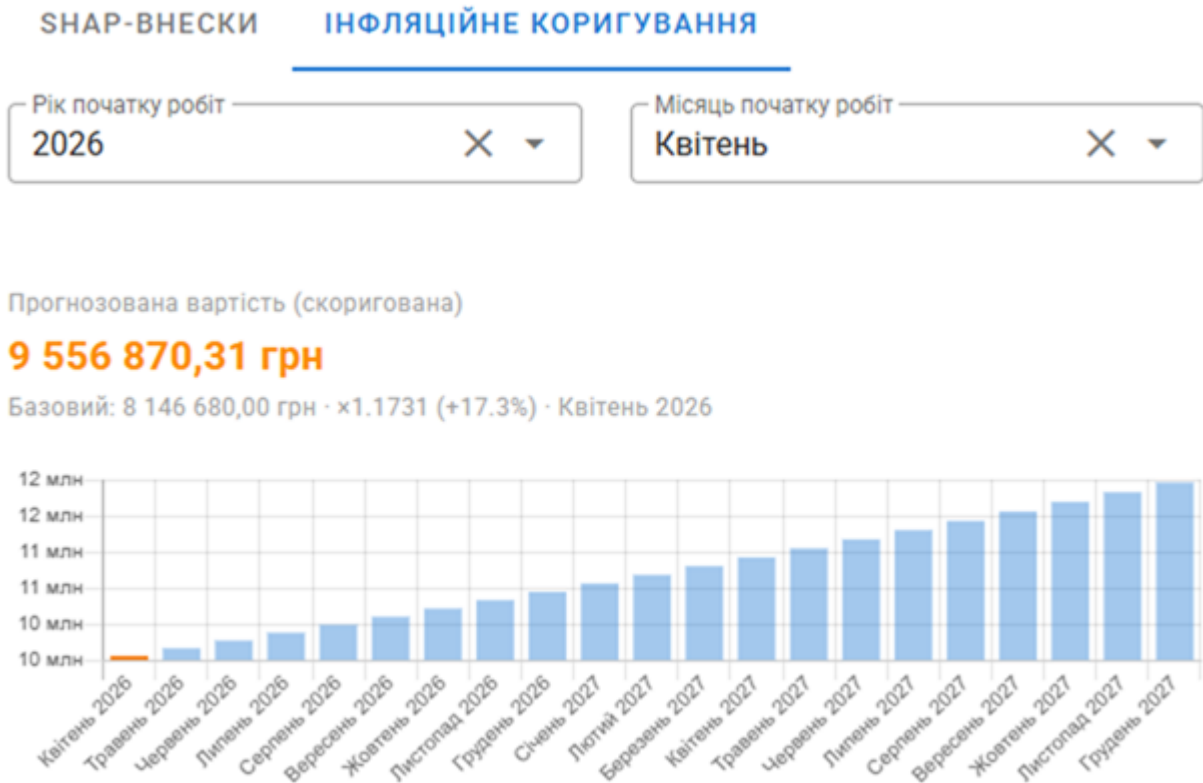


Рисунок 3.19 – Діаграма порівняння скоригованої вартості відновлення по місяцях

Діаграма дає ОПР наочне розуміння двох речей одночасно. По-перше – актуальної скоригованої вартості для обраного терміну початку робіт. По-друге – траєкторії зміни кошторису в часі: для прогнозних місяців графік показує очікувані індекси цін на підставі даних НБУ, тобто ОПР бачить не лише поточну оцінку, а й те, якою вона стане за рік чи два при збереженні поточних тенденцій. Це перетворює діаграму на інструмент аргументації: відкладання відбудови конкретного об'єкта на рік можна проілюструвати не словами, а конкретною різницею у мільйонах гривень.

SHAP-інтерпретація при цьому залишається прив'язаною до базового прогнозу  $V$  та пояснює внесок характеристик об'єкта. Коефіцієнт  $k(t)$  є зовнішнім параметром масштабування і не змінює відносних внесків ознак – тому кругова діаграма SHAP-внесків залишається актуальною незалежно від обраного місяця.

### 3.9 Віртуальні тури

Окрім основних функцій обліку та аналізу пошкоджень, інформаційно-аналітична система містить модуль віртуальних турів, призначений для збереження пам'яті про пошкоджені або зруйновані об'єкти культурної спадщини. Він забезпечує не лише збереження та відображення даних про пошкодження, але й інтерактивне занурення користувачів, що сприяє глибшому розумінню масштабів та наслідків руйнувань. Таким чином, модуль став важливим доповненням до основної функціональності системи.

Основна ідея модуля «Віртуальні тури» полягає у створенні інтерактивного каталогу пошкоджених об'єктів, який слугує зручним інструментом для дослідження наслідків руйнувань. При створенні нового туру можна використовувати дані із загальної бази об'єктів, зафіксованих в інформаційно-аналітичній системі. Для цього реалізовано відповідний фільтр, що дозволяє обрати об'єкт з переліку вже зареєстрованих пошкоджень. Інтерфейс каталогу віртуальних турів представлено на рисунку 3.20.

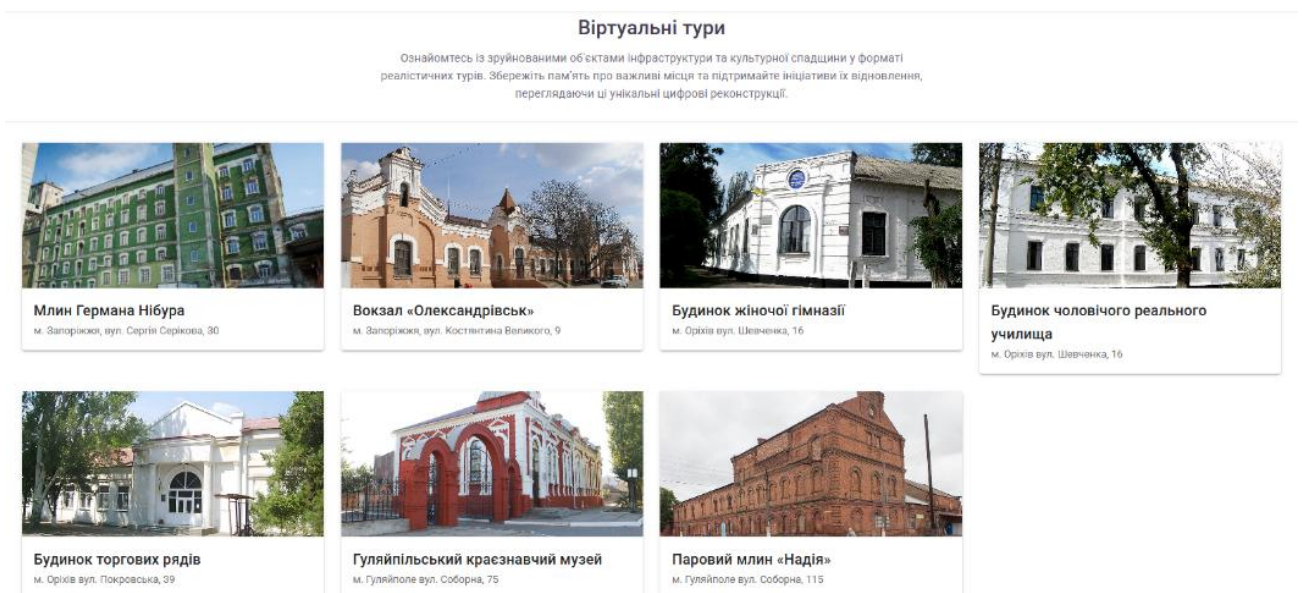


Рисунок 3.20 – Фрагмент каталогу віртуальних турів

Для кожного об'єкта в каталозі надається детальна інформація. Насамперед це опис стану об'єкта до та після пошкодження. Зазначається, яким був об'єкт до російської військової агресії: його тип (житловий будинок, адміністративна споруда, історична пам'ятка тощо), розташування, основні характеристики та функціональне призначення. Це допомагає зрозуміти початкову роль об'єкта в інфраструктурі громади. Крім того, описується ступінь пошкодження, його причини та наслідки.

Особливе місце в кожному турі відведено візуальному контенту. Кожен об'єкт доповнюється фотографіями та відеоматеріалами, що демонструють його стан до та після пошкодження. Фотографії до пошкодження показують естетичну та функціональну цінність об'єкта, що допомагає підкреслити масштаб втрат. Фотографії після пошкодження дозволяють оцінити рівень руйнувань, візуалізувати наслідки війни та сформувати чітке розуміння необхідного обсягу відновлювальних робіт. Відеоматеріали, включаючи панорамні зйомки або зйомку з дронів, надають можливість оглянути об'єкт з різних ракурсів, забезпечуючи високореалістичне представлення. Приклад представлення візуального контенту наведено на рисунку 3.21.

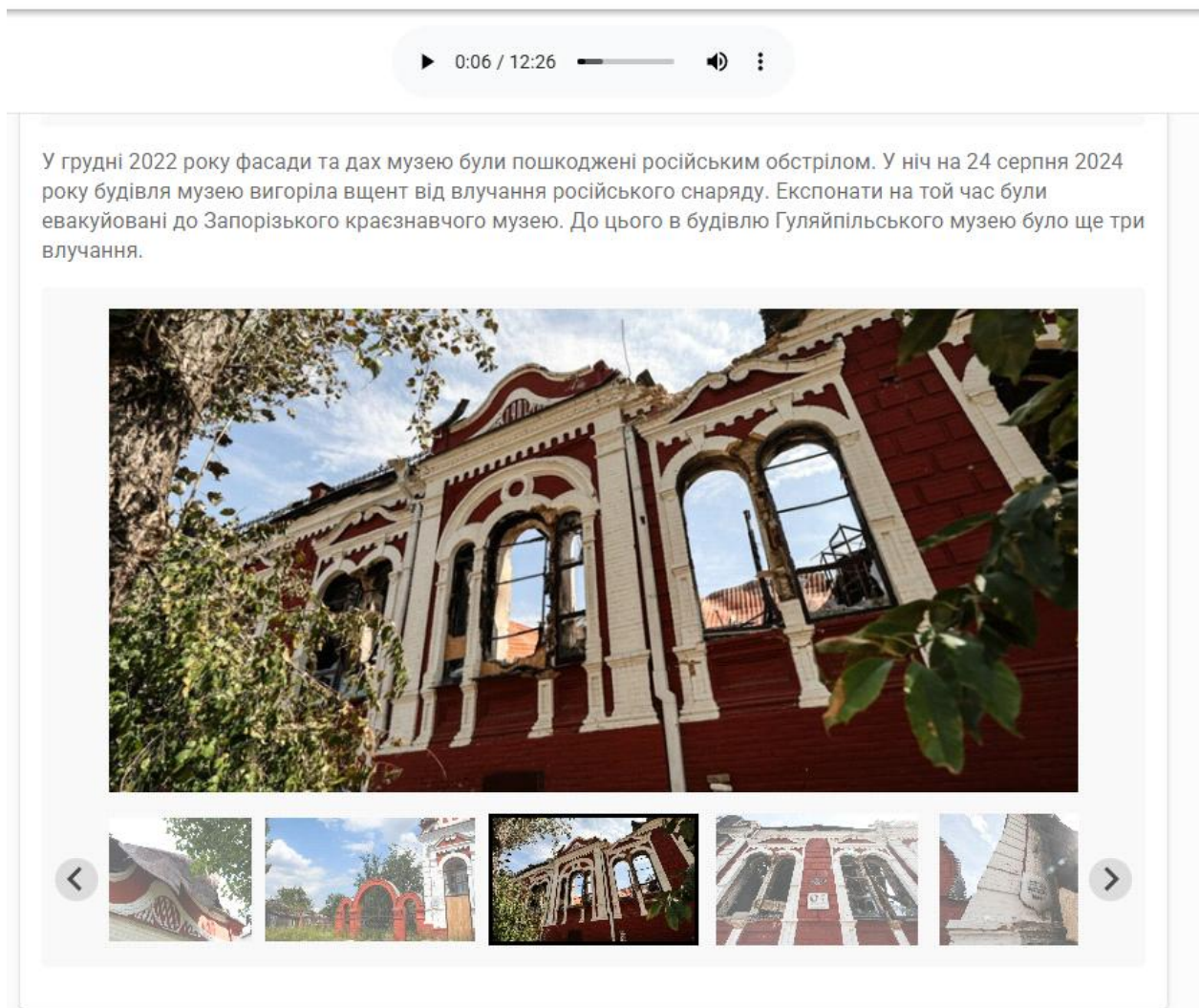


Рисунок 3.21 – Приклад представлення візуального контенту у віртуальному турі

Важливою частиною кожного туру є історична довідка про об'єкт. Вона включає таку інформацію, як рік будівництва, архітектурні особливості та значення об'єкта для місцевої громади або країни. Наприклад, якщо це історична пам'ятка, довідка висвітлює її культурну або архітектурну цінність та роль у житті громади. Якщо це житловий або адміністративний будинок, довідка містить відомості про кількість людей або організацій, які використовували приміщення. Історична довідка може також включати інформацію про попередні реконструкції або ремонтні роботи, що дає уявлення про тривалість та характер використання об'єкта.

Сам віртуальний тур проводиться через інтерактивну фотогалерею, що дозволяє детально оглянути стан об'єкта до та після пошкодження. Система також підтримує можливість додавання аудіосупроводу до туру з додатковими поясненнями та історичною інформацією про об'єкт. Такий підхід забезпечує глибше розуміння масштабів та наслідків руйнувань.

Управління віртуальними турами (створення, редагування, видалення) доступне лише користувачам з роллю «Суперадміністратор». Перегляд турів є публічним і не потребує авторизації, що забезпечує широкий доступ до інформації про наслідки руйнувань.

Загалом модуль «Віртуальні тури» не лише сприяє детальному документуванню пошкоджень, але й виконує важливу соціальну функцію, дозволяючи користувачам побачити наслідки війни. Це підвищує громадську обізнаність та сприяє залученню фінансування на відновлення через візуалізацію фактичного стану пошкоджених об'єктів. Розроблена система сприятиме створенню стратегій відновлення громад та збереженню історичних територій, залученню інвесторів та полегшенню процесу отримання репарацій у майбутньому.

### **3.10 Висновки за розділом 3**

У цьому розділі описано проєктування та програмну реалізацію інформаційно-аналітичної системи підтримки прийняття рішень щодо відновлення пошкоджених об'єктів.

Обґрунтовано вибір гібридної архітектури, що поєднує монолітний веб-застосунок (Laravel + Vue.js SPA) для функцій обліку, аналітики та візуалізації з відокремленим ML-мікросервісом (Python/Flask) для прогнозування вартості та інтерпретації результатів. Таке рішення забезпечує технологічну ізоляцію компонентів: модель прогнозування можна оновлювати або перенавчати без зупинки основного додатку.

Спроектовано REST API та реляційну базу даних MySQL, що забезпечують зберігання й обробку інформації про пошкоджені об'єкти з підтримкою адміністративно-територіальної ієрархії (область – район – громада) та категоризації об'єктів за типом.

Реалізовано аналітичний блок системи, що включає: інтерактивну карту руйнувань із трьома рівнями деталізації та кольоровою градацією ступеня пошкоджень; модуль графічної аналітики з динамічною фільтрацією за часом, регіоном та категорією об'єктів; OLAP-куб для багатовимірного аналізу даних.

Реалізовано цифровий двійник пошкодженого об'єкта, що включає цифровий профіль з паспортними характеристиками та фотофактами, модуль сценарного розрахунку вартості з SHAP-інтерпретацією внесків груп ознак, а також механізм інфляційного коригування з візуалізацією траєкторії зміни кошторису в часі.

Додатково розроблено модуль віртуальних турів, призначений для збереження пам'яті про пошкоджені об'єкти культурної спадщини та привернення громадської уваги до наслідків руйнувань через інтерактивну фотогалерею з аудіосупроводом та історичними довідками.

Результати розділу 3 опубліковані в роботах автора: [46], [52], [56], [58], [59], [62].

## 4. ТЕСТУВАННЯ СИСТЕМИ ТА ЕКСПЕРИМЕНТАЛЬНА ВЕРИФІКАЦІЯ МОДЕЛЕЙ ПРОГНОЗУВАННЯ

### 4.1 Налаштування веб-додатку

Перед початком розробки функціональних модулів системи необхідно виконати низку підготовчих дій, пов'язаних із конфігурацією середовища виконання, підключенням до бази даних, налаштуванням механізмів аутентифікації та забезпеченням взаємодії між компонентами розподіленої архітектури.

#### 4.1.1 Конфігурація середовища

Фреймворк Laravel використовує механізм змінних оточення для зберігання параметрів конфігурації, які можуть відрізнятися залежно від того, де саме запущено додаток – на локальній машині розробника, тестовому чи робочому сервері. Параметри визначаються у файлі `.env`, який розміщується в кореневому каталозі проєкту і не включається до системи контролю версій. Такий підхід дозволяє зберігати окремі конфігурації для середовищ розробки, тестування та експлуатації, не змінюючи при цьому вихідний код додатку.

Основні параметри, що задаються через змінні оточення, включають: режим роботи додатку (`APP_ENV`), ключ шифрування (`APP_KEY`), параметри підключення до бази даних, адресу ML-мікросервісу та налаштування кешування. Ключ шифрування генерується командою `php artisan key:generate` і використовується для захисту сесій, токенів та інших чутливих даних. Без коректно згенерованого ключа робота додатку неможлива.

У робочому середовищі параметр `APP_DEBUG` встановлюється у значення `false`, щоб запобігти відображенню трасування помилок кінцевим користувачам. Це важливо з погляду безпеки, оскільки трасування може містити конфіденційну

інформацію: структуру файлової системи, параметри підключення до бази даних тощо.

Для підвищення продуктивності в робочому середовищі також застосовується кешування конфігурації. Команда `php artisan config:cache` об'єднує всі конфігураційні файли в один кешований файл, що скорочує кількість звернень до файлової системи під час обробки запитів.

#### 4.1.2 Налаштування підключення до бази даних

За збереження, організацію та управління доступом до даних у системі відповідає серверна частина, реалізована на фреймворку Laravel. Однією з його переваг є наявність Eloquent ORM – об'єктно-реляційного відображення, що дозволяє описувати сутності предметної області та їх зв'язки у вигляді PHP-класів. Eloquent бере на себе генерацію SQL-запитів, завантаження даних та приведення типів бази даних до типів PHP. У результаті безпосередня робота з SQL зводиться до мінімуму – достатньо лише коректно налаштувати підключення. Водночас для складних запитів, де можливостей Eloquent ORM не вистачає, Laravel надає Query Builder – зручний інтерфейс для побудови довільних SQL-конструкцій.

Файл з основними параметрами підключення до БД розташований за шляхом `config/database.php`. У ньому визначаються варіанти підключення, обирається СУБД за замовчуванням та задаються параметри з'єднання. На момент розробки фреймворк підтримує MySQL, PostgreSQL, SQLite та SQL Server. Під час розробки обрано MySQL у зв'язку з її надійністю, швидкодією та широкою поширеністю у веб-розробці.

Ключові параметри конфігурації підключення наведено нижче:

```
'mysql' => [ 'driver' => 'mysql', 'host' => env('DB_HOST', '127.0.0.1'), 'port'
=> env('DB_PORT', '3306'), 'database' => env('DB_DATABASE', 'damage_map'),
'username' => env('DB_USERNAME', 'root'), 'password' =>
```

```
env('DB_PASSWORD', ''), 'charset' => 'utf8mb4', 'collation' =>
'utf8mb4_unicode_ci', 'engine' => 'InnoDB' ]
```

Параметр `driver` визначає драйвер СУБД; `host` – доменне ім'я або IP-адресу сервера; `database` – назву бази даних; `username` і `password` – облікові дані користувача з правами доступу. Кодування `utf8mb4` обрано свідомо: на відміну від стандартного `utf8` у MySQL, воно підтримує повний діапазон символів Unicode, включаючи кирилицю та спеціальні символи. Параметр `collation` зі значенням `utf8mb4_unicode_ci` забезпечує коректне сортування та порівняння рядків для україномовного контенту. Як движок таблиць використовується InnoDB, що підтримує транзакції, зовнішні ключі та блокування на рівні рядків – характеристики, необхідні для забезпечення цілісності даних при одночасній роботі кількох користувачів.

Варто зазначити, що всі чутливі параметри підключення зчитуються з файлу `.env` за допомогою функції `env()`. Це дозволяє використовувати один і той самий конфігураційний файл для різних середовищ розгортання.

Для адміністрування бази даних на етапі розробки використовувалось допоміжне програмне забезпечення `phpMyAdmin`, що надає зручний веб-інтерфейс для управління таблицями, виконання SQL-запитів та перегляду структури БД. При створенні бази даних `damage_map` було обрано кодування `utf8mb4_unicode_ci`, що узгоджується з налаштуваннями серверної частини.

### 4.1.3 Міграції бази даних

Після створення бази даних необхідно наповнити її таблицями відповідно до спроектованої структури (підрозділ 3.5). З цією метою використано механізм міграцій `Laravel` – систему контролю версій для схеми бази даних.

Міграції вирішують практичну проблему, що виникає при супроводі системи: необхідність синхронізації змін у структурі БД із змінами у вихідному

кодi. Кожна мiграцiя являє собою PHP-клас iз двома методами: `up()` для застосування змiн (створення таблицi, додавання стовпця, створення iндексу) та `down()` для їх скасування. Файли мiграцiй зберiгаються у каталозi `database/migrations` i мiстять мiтку часу в назвi, що визначає порядок їх застосування.

Для створення мiграцiї використовується консольна команда: *`php artisan make:migration create_damage_notes_table --create=damage_notes`*

Параметр `--create` вказує, що мiграцiя має створити нову таблицю, а не модифiкувати iснуючу. Розглянемо приклад мiграцiї для центральної таблицi системи `damage_notes`, яка зберiгає записи про пошкодженi об'єкти:

```
return new class extends Migration
{
    public function up()
    {
        Schema::create('damage_notes', function (Blueprint $table) {
            $table->id();
            $table->date('date');
            $table->bigInteger('object_type_id')->unsigned()->index()->nullable();
            $table->bigInteger('community_id')->unsigned()->index()->nullable();
            $table->string('city')->nullable();
            $table->string('street')->nullable();
            $table->string('building_number')->nullable();
            $table->unsignedInteger('floors')->nullable();
            $table->decimal('area', 12, 2)->nullable();
            $table->enum('damage_type',
array_keys(DamageNote::DAMAGE_TYPES_MAPPING))
                ->nullable();
            $table->bigInteger('repair_type_id')->unsigned()->index()->nullable();
            $table->decimal('restoration_cost', 15, 2)->nullable();
```

```

        $table->decimal('predicted_restoration_cost', 15, 2)->nullable();
        $table->text('comment')->nullable();
        $table->timestamps();
    });
}

public function down()
{
    Schema::dropIfExists('damage_notes');
}
};

```

У наведеному прикладі метод `up()` використовує фасад `Schema::create()` для створення таблиці. Другий аргумент – функція-замикання, що отримує екземпляр `Blueprint` для визначення структури. Тут задіяно кілька типів стовпців: `id()` для автоінкрементного первинного ключа, `date` для дати фіксації пошкодження, `decimal` для числових значень вартості з фіксованою точністю, `enum` для обмеженого набору значень ступеня пошкодження. Допустимі значення `enum` визначаються через константу `DAMAGE_TYPES_MAPPING`, що дозволяє уникнути дублювання переліку у різних частинах коду. Для стовпців із зовнішніми ключами додатково створюються індекси через ланцюг `->unsigned()->index()`, що прискорює виконання JOIN-запитів.

Застосування всіх міграцій виконується однією командою `php artisan migrate`. Для скасування останніх змін використовується команда `php artisan migrate:rollback`. Загалом база даних системи містить 21 таблицю, що охоплюють географічну ієрархію (регіони, райони, громади), класифікацію об'єктів, облік пошкоджень, управління доступом, контент та системні потреби Laravel.

#### 4.1.4 Налаштування аутентифікації та авторизації

Система працює з даними про пошкоджені об'єкти інфраструктури, що можуть містити конфіденційну інформацію, тому забезпечення контролю доступу є одним із пріоритетних завдань. Контроль доступу складається з двох етапів: аутентифікації (перевірка особи користувача – хто він) та авторизації (визначення дозволених дій – що він може робити).

Для аутентифікації використовується пакет Laravel Sanctum, який дозволяє ідентифікувати користувача за допомогою API-токенів. Токен зберігається на стороні клієнта у cookie-файлі та передається разом із кожним HTTP-запитом до API. Термін дії токена за замовчуванням – 1 година; при активації параметра «запам'ятати мене» він подовжується до 1 року.

Авторизація реалізована за допомогою пакету Spatie Laravel Permission, який реалізує модель рольового розмежування прав доступу (RBAC). У системі визначено три ролі:

- `super_admin` – повний доступ до всіх функцій системи без обмежень;
- `admin` – адміністратор, що має доступ до управління записами про пошкодження та модерації заявок у межах закріпленої території;
- `analyst` – роль із правом перегляду аналітичних звітів та роботи з модулем прогнозування.

Для реалізації RBAC пакет Spatie використовує окремий набір таблиць: `roles`, `permissions`, `model_has_roles`, `model_has_permissions` та `role_has_permissions`. Зв'язок між користувачами та ролями є поліморфним, що дозволяє при потребі поширити систему прав на інші сутності.

Особливістю розробленої системи є територіальне обмеження повноважень адміністраторів. Модель користувача містить поля `region_id`, `district_id` та `community_id`, що визначають територію відповідальності. При формуванні запитів до бази даних контролер автоматично застосовує фільтрацію за цими полями:

```
$query->when(isset($user->region_id), function($query) use (&$user) {
    $query->where('districts.region_id', '=', $user->region_id);
});
```

Такий механізм гарантує, що адміністратор бачить і може модерувати лише записи, що належать до його адміністративно-територіальної одиниці.

#### 4.1.5 Налаштування взаємодії з ML-мікросервісом

Модуль прогнозування вартості відновлення реалізовано у вигляді окремого мікросервісу на Python. Laravel-додаток надсилає до нього HTTP-запити з характеристиками об'єкта і отримує у відповідь прогнозовану вартість. Щоб ця взаємодія працювала, необхідно вказати адресу мікросервісу, ключ доступу та допустимий час очікування відповіді. Ці параметри зберігаються у файлі `config/services.php`:

```
'restoration' => [
    'url' => env('PY_PREDICT_URL', 'http://127.0.0.1:5000'),
    'api_key' => env('PY_API_KEY'),
    'timeout' => env('PY_TIMEOUT', 8),
],
```

Адреса мікросервісу, API-ключ та таймаут зчитуються зі змінних оточення, що дозволяє легко змінювати ці параметри при переході між середовищами розробки та експлуатації.

Таймаут у 8 секунд обрано з урахуванням того, що мікросервіс виконує запити різної складності. Базове прогнозування працює швидко – інференс XGBoost-моделі займає десятки мілісекунд. Проте прогнозування з поясненням додатково обчислює SHAP-внески кожної групи ознак у результат, що потребує значно більше часу. Встановлений таймаут забезпечує достатній запас для обох

сценаріїв, водночас не змушуючи користувача очікувати надто довго у разі недоступності мікросервісу.

Оскільки мікросервіс доступний по мережі, необхідно захистити його від несанкціонованих запитів. Для цього використовується аутентифікація через HTTP-заголовок X-API-Key. При кожному зверненні Laravel додає до запиту заголовок із ключем, який зберігається у змінній оточення `PY_API_KEY`. На стороні мікросервісу Flask-обробник порівнює отриманий ключ зі значенням, що зберігається у змінній оточення `API_KEY` самого мікросервісу. Якщо ключі не збігаються або заголовок відсутній, мікросервіс повертає HTTP-код 401 і не виконує обчислення. Такий підхід є простим, але достатнім для внутрішньої взаємодії між компонентами, що працюють у межах одного сервера або захищеної мережі.

Обробка помилок при взаємодії з мікросервісом включає декілька рівнів: перевірку доступності сервісу, валідацію HTTP-коду відповіді та перевірку структури отриманого JSON. У разі недоступності мікросервісу або отримання помилки система логує інцидент і повідомляє користувача, зберігаючи при цьому працездатність решти функцій.

#### 4.1.6 Налаштування CORS та маршрутизації API

Оскільки клієнтська частина побудована як односторінковий додаток на Vue.js і може обслуговуватися з іншого домену або порту, необхідно налаштувати механізм Cross-Origin Resource Sharing (CORS). Без такого налаштування браузер блокуватиме запити клієнта до API через політику одного джерела (Same-Origin Policy).

Laravel надає вбудований middleware для обробки CORS-запитів. Конфігурація визначається у файлі `config/cors.php`, де задаються дозволені джерела (origins), HTTP-методи та заголовки. Для даної системи дозволено всі основні методи REST API – GET, POST, PUT, PATCH і DELETE, а серед

заголовків – Content-Type, X-Requested-With, Authorization (для передачі токена аутентифікації) та Accept.

Маршрутизація API визначена у файлі routes/api.php. Всі маршрути автоматично отримують префікс /api та проходять через middleware-групу api, що забезпечує обмеження частоти запитів (rate limiting) і обробку JSON-відповідей. Маршрути, що потребують аутентифікації, додатково захищені middleware auth:api:

```
Route::group(['middleware' => ['auth:api']], function () {
    Route::get('/user', [AuthController::class, 'user']);
    Route::post('/logout', [AuthController::class, 'logout']);
    // ...
});
```

Водночас частина маршрутів залишається доступною без аутентифікації – зокрема, довідники регіонів, типів об'єктів та громад, статистика, а також створення заявок про пошкодження. Це дозволяє громадянам подавати заявки без попередньої реєстрації в системі.

## 4.2 Формування навчальної вибірки

Для проведення навчання моделей машинного навчання було зібрано табличні дані про об'єкти нерухомості, що зазнали пошкоджень. Вони охоплювали як безпосередні характеристики об'єктів, так і додаткову аналітичну інформацію, зокрема географічне розташування, ступінь пошкоджень і тип будівлі.

При формуванні навчальної вибірки була використана інформація з державного сервісу Prozorro, який містить відкриті дані про публічні закупівлі, пов'язані з ремонтами та відновленням будівель [60]. Це дозволило отримати реальні значення вартості ремонтних робіт на основі офіційних контрактів, що

підвищило якість і достовірність навчального набору даних. Використання відкритих даних публічних закупівель закриває виявлену у літературі прогалину та формує основу для забезпечення більшої відкритості й ефективності фінансового управління процесами відбудови.

Після збору інформації було здійснено етап передобробки даних, під час якого всі категоріальні ознаки було перетворено на числові. Для моделі Linear Regression застосовано порядкове (ordinal) кодування, при якому кожній категорії присвоювався унікальний числовий ранг (1, 2, 3 тощо). Такий підхід дає змогу врахувати можливу градацію значень та зберегти інтерпретованість коефіцієнтів. У випадку Random Forest та XGBoost використано метод one-hot encoding, оскільки ці ансамблеві алгоритми краще працюють із бінарно закодованими ознаками.

Однак для частини ознак (зокрема `damage_level` – ступінь пошкодження та `repair_type` – тип ремонту) існує внутрішній порядок або градація, що не враховується при one-hot кодуванні. Це може призводити до втрати інформації про відносні відмінності між категоріями й, відповідно, знижувати точність моделі. Для вирішення цієї проблеми було застосовано методи фазифікації з трикутними функціями належності (fuzzy-кодування), що дозволяє більш коректно формалізувати градаційні переходи між станами та підвищити точність прогнозування (детальний опис методів кодування наведено у підрозділах 2.2.1–2.2.2).

Для оцінки ефективності fuzzy-кодування у порівнянні з традиційним one-hot кодуванням застосовано метод 5-кратної крос-валідації. Суть методу полягає у поділі вибірки на 5 рівних частин: на кожній ітерації одна частина слугує тестовою вибіркою, а решта 4 – навчальною, після чого результати усереднюються по всіх ітераціях. Такий підхід дозволяє отримати більш надійну оцінку якості моделі та зменшити вплив випадкового розподілу даних.

Для порівняння використано модель XGBoost Regressor та чотири метрики якості: MAE (середня абсолютна помилка), RMSE (корінь середньоквадратичної

помилки),  $R^2$  (коефіцієнт детермінації) та MAPE (середня абсолютна відсоткова помилка) [61]. Результати порівняння наведено у таблиці 4.1.

Таблиця 4.1 – Порівняння методів кодування порядкових ознак

Метрика	One-hot кодування	Fuzzy-кодування	$\Delta$
MAE, грн	16 976 183	16 957 077	-0,1%
RMSE, грн	31 922 795	31 913 844	-0,0%
$R^2$	0,3312	0,3315	+0,1%
MAPE, %	41,2	41,1	-0,1%

Результати свідчать, що нечітке кодування забезпечує незначне покращення всіх метрик якості прогнозування: MAE зменшується на 0,1%, що в абсолютному вимірі становить близько 19 тис. грн економії похибки на один об'єкт. Водночас статистичний аналіз (парний t-тест,  $p = 0,39$ ) показує, що ця різниця не є статистично значущою на рівні  $\alpha = 0,05$ , що пояснюється обмеженим обсягом навчальної вибірки.

Додатково проведено тест монотонності для перевірки того, що прогнози зростають із збільшенням рівня пошкодження (Легке > Середнє > Тяжке). Fuzzy-кодування успішно пройшло цей тест, що підтверджує коректність збереження градаційної структури порядкових ознак.

Незважаючи на відсутність статистично значущої переваги за наявного обсягу даних, fuzzy-кодування було обрано для фінальної моделі з таких міркувань: воно коректно відображає семантику порядкових ознак, забезпечує кращу інтерпретованість SHAP-внесків та має потенціал до більш суттєвого покращення точності при збільшенні навчальної вибірки.

Щоб забезпечити об'єктивну оцінку точності моделі та перевірити її здатність до узагальнення, весь набір даних було поділено на тренувальну та

тестову вибірки у співвідношенні 80:20. Для забезпечення відтворюваності результатів було зафіксовано параметр `random_state = 42`.

### 4.3 Результати порівняння моделей прогнозування

Для комплексної перевірки ефективності обраних регресійних алгоритмів навчання здійснювалося у два етапи. Спочатку було використано вибірку обсягом 50 записів. Зразок даних цієї вибірки наведено у табл. 4.2.

Таблиця 4.2 – Зразок даних

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$S$ , грн
4000	5	Багатоквар тирна	Тяжке	м. Одеса	Капітал ьний	32788802
7200	9	Багатоквар тирна	Тяжке	Київська обл.	Капітал ьний	39493328
7200	9	Адміністр ативна	Середнє	Київська обл.	Поточни й	27781429
7200	9	Багатоквар тирна	Тяжке	Київська обл.	Капітал ьний	48975673
4000	5	Багатоквар тирна	Тяжке	Київська обл.	Капітал ьний	49790360
4000	5	Багатоквар тирна	Тяжке	Київська обл.	Капітал ьний	63354904
8000	10	Багатоквар тирна	Тяжке	Київська обл.	Повна реконст рукція	96045497
7200	9	Багатоквар тирна	Тяжке	Київська обл.	Капітал ьний	321652285

Продовження таблиці 4.2

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$S$ , грн
4000	5	Багатоквар тирна	Тяжке	Київська обл.	Капітал ьний	86432499
3200	4	Освітня	Середнє	м. Харків	Капітал ьний	60558384

Результат навчання на вибірці з 50 записів можна побачити у табл. 4.3.

Таблиця 4.3 – Результати навчання на вибірці обсягом 50 записів

Назва моделі	MAE (млн грн)	Приблизна похибка, % від вартості відновлення
Linear Regression	20,81	~ 40–45%
Random Forest Regressor	14,54	~ 30–35%
XGBoost Regressor	14,86	~ 30–35%

На цьому етапі модель Random Forest показала найнижче середнє абсолютне відхилення (MAE), однак всі три моделі мали відносно високі помилки через обмежений обсяг вибірки. Помилки моделі становили в середньому 30–40% від фактичної вартості відновлення, що є недостатнім для практичного застосування в оцінці ризиків чи формуванні бюджетних рішень. Тому було прийнято рішення збільшити навчальну вибірку до 100 записів, частково за рахунок аугментованих варіантів наявних даних.

Для коректного відображення вартості у загальноприйнятому валютному еквіваленті слід зазначити, що на момент формування вибірки (дата 05.06.2025) офіційний курс НБУ становив 1 USD = 41,4829 грн, 1 EUR = 47,2013 грн. Таким чином, наведені суми у гривнях можна інтерпретувати у доларах чи євро. Результат навчання на вибірці зі 100 записів можна побачити у табл. 4.4.

Таблиця 4.4 – Результати навчання на вибірці обсягом 100 записів

Назва моделі	MAE (млн грн)	Приблизна похибка, % від вартості відновлення
Linear Regression	15,94	~ 25–30%
Random Forest Regressor	9,65	~ 15–20%
XGBoost Regressor	7,27	~ 15–20%

Після проведеного навчання також було отримано аналітичні форми моделей, що дає змогу інтерпретувати їх роботу та оцінити внесок окремих ознак у формування прогнозованої вартості відновлення.

Аналітична форма моделі Linear Regression:

$$V = -69339774.63 - 8483.34 \cdot x_1 + 9257139.60 \cdot x_2 + 2320479.86 \cdot x_3 + 39763722.59 \cdot x_4 - 981582.33 \cdot x_5 + 13325011.68 \cdot x_6, \quad (4.1)$$

Аналітична форма одного дерева моделі Random Forest (4.2). Наведені значення є прогнозами лише одного дерева, тоді як загальний результат обчислюється як середнє значення результатів усіх дерев ансамблю:

$$f_1(x) = \begin{cases} 2948233, & \text{якщо } (x_4 \neq \text{Тяжке}) \wedge (x_4 \neq \text{Середнє}) \wedge (5662.5 < x_1 \leq 7312.5), \\ 28600000, & \text{якщо } (x_4 \neq \text{Тяжке}) \wedge (x_4 \neq \text{Середнє}) \wedge (x_1 > 7312.5), \\ 96045497, & \text{якщо } (x_4 = \text{Тяжке}) \wedge (x_6 \neq \text{Капітальний}) \wedge (x_5 \neq \text{м.Запоріжжя}) \wedge (x_5 \neq \text{м.Ірпінь}), \\ 91000000, & \text{якщо } (x_4 = \text{Тяжке}) \wedge (x_6 \neq \text{Капітальний}) \wedge (x_5 = \text{м.Ірпінь}), \\ 158272516, & \text{якщо } (x_4 = \text{Тяжке}) \wedge (x_6 \neq \text{Капітальний}) \wedge (x_5 = \text{м.Запоріжжя}), \\ 38500000, & \text{якщо } (x_4 = \text{Тяжке}) \wedge (x_6 = \text{Капітальний}) \wedge (x_5 = \text{м.Суми}), \\ \text{інші випадки,} & \text{значення визначається іншими гілками дерева.} \end{cases} \quad (4.2)$$

Аналітична форма одного дерева моделі XGBoost (4.3). Наведені значення є прогнозами лише одного дерева, тоді як загальний результат обчислюється як сума внесків усіх дерев ансамблю:

$$f_1(x) = \begin{cases} -3875224.25, & \text{якщо } (x_4 = \text{Тяжке}) \wedge (x_1 \geq 7410) \wedge (x_1 < 8250), \\ -1068228.88, & \text{якщо } (x_4 = \text{Тяжке}) \wedge (x_1 \geq 8250), \\ 4872042, & \text{якщо } (x_4 \neq \text{Тяжке}) \wedge (x_5 = \text{м.Запоріжжя}) \wedge (x_5 \neq \text{Київська обл.}) \wedge (x_1 < 3300), \\ 3640291.25, & \text{якщо } (x_4 \neq \text{Тяжке}) \wedge (x_5 = \text{Одеська обл.}) \wedge (x_1 < 7410), \\ 21266848, & \text{якщо } (x_4 \neq \text{Тяжке}) \wedge (x_5 \neq \text{м.Запоріжжя}) \wedge (x_3 = \text{Адмін.}) \wedge (x_1 \geq 3300), \\ \text{інші випадки,} & \text{значення визначається іншими гілками дерева.} \end{cases} \quad (4.3)$$

Відповідно до отриманих результатів навчання зі збільшенням вибірки до 100 записів точність моделей суттєво зросла. Найкращий результат продемонстрував алгоритм XGBoost Regressor, який забезпечив найнижче середнє абсолютне відхилення з помилкою, що в середньому склала близько 15–20% від вартості відновлення. Враховуючи складність структури даних і наявність категоріальних змінних, дерева рішень виявилися ефективнішими за класичну лінійну модель [62].

#### 4.4 Тестування модулів системи

Для забезпечення якості та стабільності роботи системи проведено автоматизоване тестування її ключових компонентів. Тестування охоплює три рівні: модульне тестування серверної частини (Laravel), модульне тестування ML-мікросервісу (Python) та інтеграційне тестування взаємодії між ними.

##### 4.4.1 Модульне тестування серверної частини

Для тестування API серверної частини використовується фреймворк PHPUnit, вбудований у Laravel. Кожен тест-клас використовує трейт RefreshDatabase, що забезпечує чисту базу даних перед кожним тестом – таблиці створюються через міграції, а після виконання тесту всі зміни скасовуються.

Тести серверної частини організовані у три файли відповідно до функціональних модулів:

1. Тестування аутентифікації (AuthTest). Файл містить чотири тести, що перевіряють процес входу в систему та роботу із сесіями: успішний логін з валідними обліковими даними (перевіряється наявність `access_token` у відповіді), спроба входу з невірним паролем (очікується код помилки 401), отримання даних поточного користувача через захищений ендпоінт `GET /api/user`, а також вихід із системи з очищенням токена.

2. Тестування модуля заявок (DamageNoteRequestTest). Шість тестів покривають повний життєвий цикл заявки. Перевіряється, що неаутентифікований користувач може подати заявку з валідними даними – це відповідає публічному характеру ендпоінту `POST /api/damage-note-requests`. Окремий тест перевіряє валідацію: при відсутності обов'язкових полів або некоректному форматі дати сервер повертає код 422 зі списком помилок. Для тестування схвалення та відхилення заявок створюється користувач із роллю `super_admin`, після чого перевіряється, що відповідні поля (`approved_at`, `declined_at`, `approver_comment`) коректно заповнюються у базі даних. Два тести перевіряють розмежування доступу: користувач із роллю `analyst` отримує код 403 при спробі схвалити або відхилити заявку.

3. Тестування модуля записів про пошкодження (DamageNoteTest). П'ять тестів охоплюють CRUD-операції та контроль доступу. Перевіряється, що неаутентифікований користувач не може отримати список записів (код 401), а аутентифікований адміністратор отримує коректну JSON-структуру відповіді. Тестуються операції оновлення та видалення записів з перевіркою стану бази даних після виконання. Окремої уваги заслуговує тест територіального обмеження: створюються два регіони з окремими ієрархіями (регіон > район > громада), адміністратору призначається `region_id` одного з них, після чого перевіряється, що він бачить лише записи свого регіону.

Результат виконання модульних тестів серверної частини наведено на рисунку 4.1.

```
PS D:\projects\php\damage-map> php artisan test
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Feature\AuthTest
✓ login with valid credentials returns 200 with access token
✓ login with invalid credentials returns 401
✓ authenticated user can get their own data
✓ logout clears api token and returns 200

PASS Tests\Feature\DamageNoteRequestTest
✓ unauthenticated user can create damage note request with valid data
✓ creating damage note request with invalid data returns 422
✓ super admin can approve damage note request
✓ super admin can decline damage note request with comment
✓ analyst cannot approve damage note request returns 403
✓ analyst cannot decline damage note request returns 403

PASS Tests\Feature\DamageNoteTest
✓ unauthenticated user cannot access approved damage notes returns 401
✓ authenticated admin gets approved damage notes with correct json structure
✓ admin can update damage note
✓ admin can delete damage note
✓ admin with region id sees only notes from their region

Tests: 15 passed
Time: 0.99s
```

Рисунок 4.1 – Результат виконання модульних тестів серверної частини

#### 4.4.2 Модульне тестування ML-мікросервісу

Тестування Python-мікросервісу реалізовано за допомогою фреймворку pytest з використанням вбудованого тестового клієнта Flask. Тести організовані у п'ять класів за функціональним призначенням:

1. Тестування аутентифікації (TestAuthentication). Чотири тести перевіряють роботу захисту через заголовок X-API-Key: запит без заголовка повертає 401, запит з невалідним ключем також повертає 401, а з валідним – 200. Окремо перевіряється аутентифікація для ендпоінту /predict\_explain.

2. Тестування валідації вхідних даних (TestValidation). Вісім тестів покривають різні сценарії некоректного введення: відсутність обов'язкового поля `area`, передача рядка замість числа, недопустиме значення `building_type`, від'ємні значення площі та кількості поверхів. Окрема група тестів перевіряє правило парності параметрів: `work_year` без `work_month` (і навпаки) повертає 400, так само як і передача цих параметрів без масиву `inflation_indices`.

3. Тестування базового прогнозування (TestPredict). Дев'ять тестів перевіряють ендпоінт `/predict`. Базовий тест переконується, що `predicted_cost` є додатним числом. Перевіряється наявність усіх обов'язкових полів у відповіді (`predicted_cost`, `adjusted_cost`, `inflation_k`, `currency`, `model`), а також що валюта завжди дорівнює UAH. Для запитів без інфляційного коригування перевіряється рівність `adjusted_cost` і `predicted_cost` та значення `inflation_k = 1.0`. При передачі параметрів інфляції перевіряється, що скоригована вартість не менша за базову та що коефіцієнт відповідає переданим індексам. Два додаткових тести перевіряють граничні випадки: мінімальні значення (площа 1 м<sup>2</sup>, 1 поверх) та великі значення (100 000 м<sup>2</sup>, 50 поверхів).

4. Тестування прогнозування з поясненням (TestPredictExplain). Вісім тестів перевіряють ендпоінт `/predict_explain`. Окрім базових перевірок (код 200, наявність поля `contributions`), тести контролюють структуру кожного елемента масиву внесків (поля `group`, `contribution`, `percent`), а також що сума відсотків дорівнює 100%. Окремий тест перевіряє фундаментальну властивість SHAP-декомпозиції: сума `base_value` та всіх `contribution` має приблизно дорівнювати `predicted_cost` (з допуском 1%). Перевіряється також узгодженість прогнозів: значення `predicted_cost` у відповідях `/predict` та `/predict_explain` для однакових вхідних даних повинні збігатися.

5. Тестування перевірки стану (TestHealth). Два тести перевіряють ендпоінт `/health`, що не потребує ані моделі, ані аутентифікації.

Результат виконання модульних тестів мікросервісу наведено на рисунку 4.2.

```

platform win32 -- Python 3.12.0, pytest-9.0.3, pluggy-1.6.0 -- D:\projects\python\corruption-risks\.venv\Scripts\python.exe
tests/test_api.py::TestValidation::test_invalid_building_type_returns_400 PASSED
tests/test_api.py::TestValidation::test_work_year_without_work_month_returns_400 PASSED
tests/test_api.py::TestValidation::test_work_month_without_work_year_returns_400 PASSED
tests/test_api.py::TestValidation::test_work_year_month_without_inflation_indices_returns_400 PASSED
tests/test_api.py::TestValidation::test_negative_area_returns_400 PASSED
tests/test_api.py::TestValidation::test_negative_floors_returns_400 PASSED
tests/test_api.py::TestPredict::test_basic_predict_returns_positive_cost PASSED
tests/test_api.py::TestPredict::test_basic_predict_response_has_required_fields PASSED
tests/test_api.py::TestPredict::test_basic_predict_currency_is_uah PASSED
tests/test_api.py::TestPredict::test_predict_without_inflation_adjusted_cost_equals_predicted PASSED
tests/test_api.py::TestPredict::test_predict_with_inflation_returns_adjusted_cost PASSED
tests/test_api.py::TestPredict::test_predict_with_inflation_coefficient_matches_payload PASSED
tests/test_api.py::TestPredict::test_predict_minimal_values_returns_200 PASSED
tests/test_api.py::TestPredict::test_predict_large_values_returns_200 PASSED
tests/test_api.py::TestPredictExplain::test_predict_explain_returns_200 PASSED
tests/test_api.py::TestPredictExplain::test_predict_explain_has_contributions_field PASSED
tests/test_api.py::TestPredictExplain::test_predict_explain_contributions_have_correct_structure PASSED
tests/test_api.py::TestPredictExplain::test_predict_explain_percentages_sum_to_100 PASSED
tests/test_api.py::TestPredictExplain::test_predict_explain_shap_decomposition_property PASSED
tests/test_api.py::TestPredictExplain::test_predict_explain_has_base_value_field PASSED
tests/test_api.py::TestPredictExplain::test_predict_explain_predicted_cost_matches_predict PASSED
tests/test_api.py::TestPredictExplain::test_predict_explain_with_inflation PASSED
tests/test_api.py::TestHealth::test_health_returns_200 PASSED
tests/test_api.py::TestHealth::test_health_returns_ok_status PASSED

```

Рисунок 4.2 – Результат виконання модульних тестів ML-мікросервісу

#### 4.4.3 Інтеграційне тестування

Інтеграційні тести перевіряють взаємодію між Laravel-додатком і Python-мікросервісом. Оскільки під час тестування мікросервіс не запущено, зовнішні HTTP-запити мокуються за допомогою фасаду `Http::fake()`, який перехоплює всі вихідні запити і повертає заздалегідь визначені відповіді.

Файл `RestorationCostIntegrationTest` містить 17 тестів, що можна розділити на кілька груп:

1. Взаємодія при створенні заявки. Ключовий тест перевіряє повну послідовність дій: при створенні заявки через `POST /api/damage-note-requests` Laravel формує `payload` з характеристиками об'єкта та масивом індексів інфляції, надсилає його до мікросервісу і зберігає отримане значення `predicted_restoration_cost` у базі даних. За допомогою `Http::assertSent()` тест перевіряє, що запит до мікросервісу містив правильні параметри: площу,

кількість поверхів, тип будівлі, рівень пошкодження та назву регіону (яка визначається через зв'язки `community > district > region`).

2. Обробка помилок мікросервісу. Три тести перевіряють стійкість системи до збоїв: таймаут з'єднання (`ConnectionException`), HTTP-код 500 та код 400 від мікросервісу. У всіх випадках заявка все одно створюється, але поле `predicted_restoration_cost` залишається порожнім, а користувач отримує відповідне повідомлення.

3. Ендпоінт прогнозування. Тести перевіряють `POST /api/predict-restoration-cost`: успішний прогноз з мокуванням Flask-відповіді, перевірка структури JSON з SHAP-внесками, обробка помилок мікросервісу та повернення коду 401 для неаутентифікованих запитів.

4. Сценарне порівняння. Тестується ендпоінт `POST /api/predict-restoration-cost/scenario-comparison`, що приймає масив часових періодів і для кожного виконує окремий запит до мікросервісу. Перевіряється структура відповіді (масив `scenarios` із полями `predicted_cost`, `adjusted_cost`, `inflation_k`, `delta_pct`), кількість викликів мікросервісу (`Http::assertSentCount(3)` для трьох періодів) та те, що перший сценарій є базовим (`delta_pct = 0`).

5. Незалежність схвалення від ML. Два тести підтверджують, що операція схвалення заявки (`POST /api/damage-note-requests/{id}/approve`) не звертається до мікросервісу – прогноз вже збережено на етапі створення заявки. Це перевіряється через `Http::assertNothingSent()`.

Результат виконання інтеграційних тестів наведено на рисунку 4.3.

```

PS D:\projects\php\damage-map> php artisan test --filter=RestorationCostIntegrationTest
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Feature\RestorationCostIntegrationTest
✓ store request calls ml service and saves predicted cost
✓ store request sends correct region in ml payload
✓ store request sends api key header to ml service
✓ store request handles ml service 500 error
✓ store request handles ml service 400 validation error
✓ store request handles ml service connection timeout
✓ store request includes inflation indices in ml payload
✓ predict restoration cost endpoint
✓ predict endpoint with work year and month
✓ predict endpoint handles ml service unavailable
✓ predict endpoint requires authentication
✓ predict endpoint validates request fields
✓ scenario comparison endpoint
✓ scenario comparison handles ml service failure
✓ scenario comparison validates periods
✓ approve request sets approved at and approver id
✓ approve does not overwrite predicted restoration cost

Tests: 17 passed
Time: 0.99s

```

Рисунок 4.3 – Результат виконання інтеграційних тестів

#### 4.5 Обмеження та перспективи розвитку

Обмеження дослідження стосуються насамперед репрезентативності навчальних даних та точності прогнозування вартості відновлення.

Навчальні вибірки сформовано на основі відкритих даних системи Prozorro, які охоплюють відносно невелику кількість об'єктів (100 записів), і відображають виключно ті випадки, що стали предметом публічних закупівель. Об'єкти, відновлення яких фінансувалося з інших джерел (донорські кошти, місцеві бюджети, приватні інвестиції), у вибірці не представлені. Це обмежує репрезентативність даних та може спричиняти систематичні похибки, пов'язані зі зміщенням вибірки.

Результати валідовано лише в діапазоні вартостей від 20 до 90 млн грн, тому прогнози для об'єктів із суттєво меншою або більшою вартістю відновлення потребують додаткової перевірки і можуть мати значно вищу похибку. Окрім

того, точність прогнозу значною мірою залежить від якості та повноти вхідних даних. Відсутність або викривлення ключових параметрів – наприклад, площі чи ступеня пошкодження – може призводити до некоректних оцінок. З огляду на зазначене, моделі машинного навчання, інтегровані в цифровий двійник, слід розглядати як інструмент попереднього аналізу, що доповнює, але не замінює проектно-кошторисну оцінку експертів.

Перспективи подальших досліджень охоплюють два напрями: розширення функціональних можливостей цифрового двійника та підвищення точності прогнозування.

Щодо розвитку цифрового двійника, доцільним є впровадження механізму історії та версійності сценаріїв. Ідея полягає у збереженні результатів кожного сценарного (what-if) перерахунку як окремої «версії рішення» з фіксацією дати, автора та короткого опису. Такий підхід дозволить відстежувати, які саме сценарії розглядалися раніше, які оцінки вони давали, і на підставі чого було обрано фінальний варіант відбудови. За потреби користувач зможе повернутися до попередніх розрахунків і порівняти їх між собою, що підвищить прозорість процесу ухвалення рішень.

Перспективним є також застосування глобальної інтерпретації SHAP для порівняльного аналізу структури впливів у різних адміністративних розрізах. Наразі інтерпретація реалізована лише на рівні окремого об'єкта (локальний SHAP). Перехід до агрегованого аналізу дозволив би виявляти відмінності у найвпливовіших ознаках при формуванні вартості між регіонами, районами, громадами або типами об'єктів. Це, у свою чергу, надасть можливість обґрунтовувати диференційовані підходи до планування відбудови залежно від специфіки конкретної території.

Підвищення точності прогнозів вартості відновлення потребує роботи у кількох напрямках. Насамперед необхідно розширити навчальну вибірку, залучивши більшу кількість реальних заявок із верифікованими даними з різних джерел фінансування, не лише публічних закупівель. Важливим кроком є також

інженерія ознак – доповнення набору параметрів обчислюваними характеристиками, серед яких вік будівлі, орієнтовна вартість 1 м<sup>2</sup> для відповідного регіону, відстань до лінії зіткнення, тип забудови, наявність комунікацій тощо. В умовах обмеженого обсягу реальних даних доцільно застосувати аугментацію – генерацію штучних записів на основі незначних перетворень наявних даних із збереженням статистичних властивостей розподілу.

Реалізація зазначених заходів дозволить підвищити точність прогнозування та розширити діапазон застосування розробленої інформаційно-аналітичної системи для підтримки прийняття рішень щодо відбудови.

#### **4.6 Висновки за розділом 4**

У цьому розділі виконано налаштування веб-додатку, проведено експериментальну верифікацію моделей прогнозування вартості відновлення та автоматизоване тестування модулів системи.

Виконано налаштування веб-додатку, що включає конфігурацію середовища через змінні оточення, підключення до бази даних MySQL, створення 21 таблиці за допомогою механізму міграцій Laravel, налаштування аутентифікації через Laravel Sanctum та рольового розмежування доступу за допомогою пакету Spatie Permission. Також забезпечено взаємодію з ML-мікросервісом через HTTP-клієнт та налаштовано CORS для коректної роботи односторінкового додатку.

Для верифікації моделей прогнозування сформовано навчальну вибірку обсягом 100 записів на основі відкритих даних системи Prozorro у діапазоні вартостей від 20 до 90 млн грн. Навчання проведено у два етапи – на вибірках 50 та 100 записів, що дозволило оцінити вплив обсягу даних на якість прогнозування.

Порівняльний аналіз трьох регресійних моделей на вибірці зі 100 записів показав, що найкращий результат продемонстрував алгоритм XGBoost Regressor із MAE = 7,27 млн грн (похибка ~15–20% від вартості відновлення). Random Forest Regressor посів друге місце з MAE = 9,65 млн грн, а Linear Regression – третє з MAE = 15,94 млн грн. Збільшення вибірки з 50 до 100 записів суттєво підвищило точність усіх моделей: для XGBoost MAE знизилася вдвічі.

Проведено порівняння методів кодування порядкових ознак за допомогою 5-кратної крос-валідації. Fuzzy-кодування забезпечило незначне покращення всіх метрик (MAE зменшилася на 0,1%), проте різниця не є статистично значущою ( $p = 0,39$ ). Водночас fuzzy-кодування успішно пройшло тест монотонності, підтвердивши коректність збереження градаційної структури порядкових ознак, і було обрано для фінальної моделі з огляду на кращу інтерпретованість SHAP-внесків та потенціал до покращення при збільшенні вибірки.

Проведено автоматизоване тестування системи на трьох рівнях: модульне тестування серверної частини (15 тестів PHPUnit), модульне тестування ML-мікросервісу (31 тест pytest) та інтеграційне тестування взаємодії між Laravel-додатком і Python-мікросервісом (17 тестів з мокуванням HTTP-запитів). Тести покривають аутентифікацію, валідацію вхідних даних, CRUD-операції із записами про пошкодження, розмежування доступу за ролями та територіями, коректність прогнозування та SHAP-пояснень, а також стійкість системи до збоїв мікросервісу.

Визначено обмеження дослідження: відносно невеликий обсяг навчальної вибірки, обмежена репрезентативність (лише публічні закупівлі), валідованість моделі у вузькому діапазоні вартостей. Окреслено перспективи подальших досліджень, що охоплюють два напрями: розширення функціональних можливостей цифрового двійника (версійність сценаріїв, глобальна SHAP-інтерпретація в адміністративних розрізах) та підвищення точності прогнозування (розширення вибірки, інженерія ознак, аугментація даних).

Результати розділу 4 опубліковані в роботах автора: [62].

## ВИСНОВКИ

У дисертаційній роботі розроблено інтелектуальну інформаційно-аналітичну систему підтримки прийняття рішень щодо відбудови пошкоджених об'єктів за умов невизначеності, що забезпечує вирішення науково-практичної проблеми наскрізної підтримки процесу відбудови – від фіксації пошкоджень до формування прогностичної оцінки вартості відновлення та сценарного порівняння альтернатив.

Проведені дослідження дозволили зробити такі висновки.

1. Проведено аналіз існуючих рішень у трьох взаємопов'язаних напрямках: фіксації руйнувань, прогнозуванні вартості відновлення та побудові цифрових двійників. В результаті була виявлена фрагментарність наявних рішень, оскільки вони охоплюють лише окремі функції, такі як збір даних, візуалізація або статистична звітність. На підставі цього обґрунтовано доцільність інтеграції зазначених напрямів у єдину систему, що забезпечує безперервний інформаційний цикл від фіксації пошкоджень до прогнозування вартості та сценарного аналізу, підвищуючи обґрунтованість управлінських рішень щодо відбудови.

2. Розроблено концепцію інформаційно-аналітичної системи підтримки прийняття рішень на основі гібридної архітектури, що поєднує монолітний веб-застосунок та відокремлений ML-мікросервіс, які взаємодіють через REST API. Система інтегрує чотири підсистеми: фіксацію пошкоджень з розмежуванням прав доступу, багатовимірну OLAP-аналітику, прогнозування вартості відновлення та сценарне моделювання на основі цифрового двійника. Наскрізним елементом є цифровий профіль об'єкта, дані якого безпосередньо використовуються для прогнозування та сценарного аналізу, що забезпечує безперервний інформаційний цикл підтримки прийняття рішень щодо відбудови.

3. Формалізовано задачу прийняття рішення щодо відбудови пошкодженого об'єкта як багатокритеріальну задачу вибору з множини

альтернатив: капітальний ремонт, поточний ремонт, реставрація, знесення та нове будівництво, консервація. Визначено, що вибір альтернативи залежить від належності об'єкта до культурної спадщини (місцевого, регіонального, національного або світового значення), що обмежує множину допустимих рішень – зокрема, для об'єктів культурного фонду пріоритетною є реставрація, а знесення може бути виключене. Для кожної альтернативи визначено критерії оцінки: прогнозована вартість відновлення, ступінь пошкодження, тип ремонту та статус культурної спадщини. Інформаційною основою для прийняття рішення є результати роботи ІАС: прогнозована вартість для кожного сценарію, SHAP-інтерпретація факторів впливу та порівняння альтернатив через цифровий двійник. Невизначеність параметрів об'єкта враховується через механізм what-if перерахунку, що дозволяє ОПР оцінити чутливість вартості до зміни характеристик об'єкта.

4. Формалізовано задачу прогнозування вартості відновлення пошкоджених об'єктів як регресійну задачу машинного навчання з визначенням вектора ознак (площа, поверховість, тип будівлі, ступінь пошкодження, регіон, тип ремонту) та цільової змінної – оціночної вартості з проектно-кошторисної документації ProzoGo. Для кодування категоріальних ознак обґрунтовано застосування комбінованого підходу: one-hot кодування для номінальних змінних та фазифікації з трикутними функціями належності для порядкових, що зберігає інформацію про їх внутрішню градацію. Побудовано та порівняно три регресійні моделі (Linear Regression, Random Forest, XGBoost) на реальних даних. Найкращий результат продемонстрував XGBoost Regressor із найнижчим MAE, що обґрунтовує його вибір як основної моделі прогнозування.

5. Розроблено цифровий двійник пошкодженого об'єкта, що складається з двох взаємопов'язаних компонентів. Перший – цифровий профіль, який забезпечує ведення структурованої картки об'єкта з можливістю оновлення при надходженні нової інформації. Другий – модуль сценарного розрахунку, що дозволяє змінювати характеристики об'єкта (тип ремонту, ступінь пошкодження,

площу) та отримувати перераховану вартість відновлення. Кожен сценарій супроводжується SHAP-інтерпретацією, що показує внесок кожної характеристики у прогнозовану вартість та візуалізується у вигляді кругової діаграми часток впливу.

6. Проведено експериментальну верифікацію системи на даних публічних закупівель із системи Prozorro. Навчання здійснювалось у два етапи: на вибірці з 50 записів та розширеній вибірці зі 100 записів, що підтвердило зростання точності при збільшенні обсягу даних. Найкращий результат продемонстрував XGBoost Regressor із середньою абсолютною похибкою 15-20% від вартості відновлення. Також виконано автоматизоване тестування розробленої системи на трьох рівнях: модульне тестування серверної частини, модульне тестування ML-мікросервісу та інтеграційне тестування взаємодії між веб-застосунком і мікросервісом, що покривають аутентифікацію, валідацію вхідних даних, CRUD-операції, розмежування доступу за ролями та територіями, коректність прогнозування та SHAP-пояснень, а також стійкість системи до збоїв мікросервісу.

Визначено обмеження дослідження: навчальна вибірка охоплює 100 записів виключно з публічних закупівель Prozorro, результати валідовано лише у діапазоні вартостей від 20 до 90 млн грн, а точність прогнозу суттєво залежить від повноти вхідних даних. З огляду на це, моделі машинного навчання, інтегровані в цифровий двійник, слід розглядати як інструмент попереднього аналізу, що доповнює, але не замінює проєктно-кошторисну оцінку експертів.

Перспективними напрямками продовження досліджень є: розширення функціональних можливостей цифрового двійника (впровадження версійності сценаріїв, перехід до глобальної SHAP-інтерпретації в адміністративних розрізах) та підвищення точності прогнозування (розширення навчальної вибірки за рахунок даних з різних джерел фінансування, інженерія ознак – вік будівлі, вартість 1 м<sup>2</sup>, відстань до лінії зіткнення, аугментація даних).

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бідюк П. І., Тимошук О. Л., Коваленко А. Є., Коршевніук Л. О. Системи і методи підтримки прийняття рішень : підручник. Київ : КПІ ім. Ігоря Сікорського, 2022. 616 с.
2. Hashem I. A. T., Yaqoob I., Anuar N. B., Mokhtar S., Gani A., Khan S. U. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*. 2015. Vol. 47. P. 98–115. DOI: <https://doi.org/10.1016/j.is.2014.07.006>
3. Про затвердження Порядку визначення шкоди та збитків, завданих Україні внаслідок збройної агресії Російської Федерації : Постанова Кабінету Міністрів України від 20.03.2022 № 326. URL: <https://zakon.rada.gov.ua/go/326-2022-п> (дата звернення: 05.01.2025).
4. Про збір, обробку та облік інформації про пошкоджене та знищене нерухоме майно внаслідок бойових дій, терористичних актів, диверсій, спричинених військовою агресією Російської Федерації : Постанова Кабінету Міністрів України від 26.03.2022 № 380. URL: <https://zakon.rada.gov.ua/go/380-2022-п> (дата звернення: 05.01.2025).
5. Про компенсацію за пошкодження та знищення окремих категорій об'єктів нерухомого майна внаслідок бойових дій, терористичних актів, диверсій, спричинених збройною агресією Російської Федерації проти України : Закон України від 23.02.2023 № 2923-IX. URL: <https://zakon.rada.gov.ua/go/2923-20> (дата звернення: 05.01.2025).
6. Про охорону культурної спадщини : Закон України від 08.06.2000 № 1805-III. URL: <https://zakon.rada.gov.ua/go/1805-14> (дата звернення: 05.01.2025).
7. Berkeley Protocol on Digital Open Source Investigations: A Practical Guide on the Effective Use of Digital Open Source Information in Investigating Violations of International Criminal, Human Rights and Humanitarian Law / Office of the United Nations High Commissioner for Human Rights ; Human Rights Center, UC Berkeley

School of Law. – New York ; Geneva : United Nations, 2022. – 124 p. – URL: [https://www.ohchr.org/sites/default/files/2024-01/OHCHR\\_BerkeleyProtocol.pdf](https://www.ohchr.org/sites/default/files/2024-01/OHCHR_BerkeleyProtocol.pdf) (дата звернення: 05.04.2025).

8. Про доступ до публічної інформації : Закон України від 13 січня 2011 р. № 2939-VI : редакція станом на 31.03.2023 // Відомості Верховної Ради України. – 2011. – № 32. – Ст. 314. – URL: <https://zakon.rada.gov.ua/laws/show/2939-17> (дата звернення: 05.04.2025).

9. Про публічні закупівлі : Закон України від 25 грудня 2015 р. № 922-VIII : редакція станом на 19.10.2024 // Відомості Верховної Ради України. – 2016. – № 9. – Ст. 89. – URL: <https://zakon.rada.gov.ua/laws/show/922-19> (дата звернення: 05.04.2025).

10. Russia will pay: Project for collecting, assessing, and analyzing information on Ukraine's material losses from the war with Russia. Kyiv School of Economics. URL: <https://kse.ua/russia-will-pay/> (last accessed: 05.01.2025).

11. Destruction map. Anti-Corruption Headquarters. URL: <https://reukraine.shtab.net/> (last accessed: 05.01.2025).

12. Recovery plan: Destruction map. Ministry for Communities, Territories and Infrastructure Development of Ukraine. URL: <https://recovery.gov.ua/> (last accessed: 05.01.2025).

13. Civilian harm in Ukraine Timemap. Bellingcat. URL: <https://ukraine.bellingcat.com/> (last accessed: 05.01.2025).

14. Бакурова А. В., Білий В. В. Розробка інформаційно-аналітичної системи для фіксації руйнувань внаслідок російської агресії. Комунальне господарство міст. 2025. Т. 3, Вип. 191. С. 2–10. DOI: <https://doi.org/10.33042/2522-1809-2025-3-191-2-10>

15. Novozhylova M. V., Chub O. I. Mathematical support of restoration projects for historical monuments. Information Systems in Project and Program Management : Proceedings of the International Scientific and Practical Conference, Koblevo, September 9–13, 2024. Kharkiv : KhNURE, 2024. P. 171–175.

16. Singh R. The role of geographic information systems (GIS) in disaster management and planning. *International Journal of Geography, Geology and Environment*. 2024. Vol. 6, Issue 2, Part C. P. 195–205. DOI: <https://doi.org/10.22271/27067483.2024.v6.i2c.305>

17. Puri A., Elkhaboutly M., Ameer Ali N. Identifying major challenges in managing post-disaster reconstruction projects: A critical analysis. *International Journal of Disaster Risk Reduction*. 2024. Vol. 107. Article 104491. DOI: <https://doi.org/10.1016/j.ijdr.2024.104491>

18. Lozano J.-M., Tien I. Data collection tools for post-disaster damage assessment of building and lifeline infrastructure systems. *International Journal of Disaster Risk Reduction*. 2023. Vol. 94. Article 103819. DOI: <https://doi.org/10.1016/j.ijdr.2023.103819>

19. UNDP, KSE. *Green Recovery of Ukraine: Guidelines and Tools for Decision-Makers*. Kyiv : UNDP, 2023. 64 p.

20. OCHA. *Ukraine Humanitarian Needs and Response Plan 2025: Annex 4.3 – Analysis Methodology, Data Sources & Findings*. Geneva : UN OCHA, 2025. 58 p.

21. Centre for Economic Strategy. *Budget-2025: Analytical Review*. Kyiv : CES, 2024. 34 p.

22. Босенко І. В. Моделі і методи штучного інтелекту в процесі виконання будівельно-технічної експертизи. *Управління розвитком складних систем*. 2025. № 61. С. 180–186. DOI: <https://doi.org/10.32347/2412-9933.2025.61.180-186>

23. Програма комплексного відновлення Михайло-Коцюбинської територіальної громади. Чернігівська область, 2024. 112 с.

24. Liu C., Zhang P., Xu X. Literature review of digital twin technologies for civil infrastructure. *Journal of Infrastructure Intelligence and Resilience*. 2023. Vol. 2, No. 3. Article 100050. DOI: <https://doi.org/10.1016/j.iintel.2023.100050>

25. Comprehensive digital twin for infrastructure: A novel ontology and graph-based modelling paradigm / T. Li et al. *Advanced Engineering Informatics*. 2024. Vol. 62, Part B. Article 102747. DOI: <https://doi.org/10.1016/j.aei.2024.102747>

26. Digital twin in transportation infrastructure management: a systematic review / B. Yan et al. *Intelligent Transportation Infrastructure*. 2023. Vol. 2. Article liad024. DOI: <https://doi.org/10.1093/iti/liad024>

27. Oetl F., Eckart L., Schilp J. Cost estimation approach of a digital twin implementation in industry. *Procedia CIRP*. 2023. Vol. 118. P. 318–323. DOI: <https://doi.org/10.1016/j.procir.2023.06.055>

28. Serugga J. Digital Twins and AI Decision Models: Advancing Cost Modelling in Off-Site Construction. *Eng*. 2025. Vol. 6, No. 2. Article 22. DOI: <https://doi.org/10.3390/eng6020022>

29. Digital Twin for Civil Engineering Systems: An Exploratory Review for Distributed Sensing Updating / M. F. Bado et al. *Sensors*. 2022. Vol. 22, No. 9. Article 3168. DOI: <https://doi.org/10.3390/s22093168>

30. What-If Scenarios for the BedreFlyt Digital Twin / Å. A. A. Kløvstad et al. *Lecture Notes in Computer Science*. Vol. 15760. Springer, Cham, 2026. P. 360–381. DOI: [https://doi.org/10.1007/978-3-031-97439-7\\_18](https://doi.org/10.1007/978-3-031-97439-7_18)

31. Digital Twins in the Construction Industry: A Comprehensive Review of Current Implementations, Enabling Technologies, and Future Directions / H. Omrany et al. *Sustainability*. 2023. Vol. 15, No. 14. Article 10908. DOI: <https://doi.org/10.3390/su151410908>

32. Infrastructure digital twin technology: A new paradigm for future construction industry / T. D. Moshood et al. *Technology in Society*. 2024. Vol. 77. Article 102519. DOI: <https://doi.org/10.1016/j.techsoc.2024.102519>

33. Kaveh H., Alhajj R. Advancing civil infrastructure with digital twins: a review of applications and challenges. *Journal of Civil Engineering and Management*. 2025. Vol. 31, No. 8. P. 828–842. DOI: <https://doi.org/10.3846/jcem.2025.24921>

34. Білий В. В. Формалізація задачі прийняття рішення щодо відбудови пошкоджених об'єктів інфраструктури в умовах невизначеності. Інформаційні технології: теорія і практика : тези доповідей III (IX) Міжнародної науково-практичної конференції, Харків – Запоріжжя – Дніпро, 25–27 березня 2026. С. 37.

35. Державна служба статистики України. Індекси споживчих цін на товари та послуги (щомісячні дані). URL: [https://ukrstat.gov.ua/operativ/menu/menu\\_u/cit.htm](https://ukrstat.gov.ua/operativ/menu/menu_u/cit.htm) (дата звернення: 05.01.2025).
36. Zadeh L. A. Fuzzy Sets. *Information and Control*. 1965. Vol. 8, No. 3. P. 338–353. DOI: [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
37. Матвієнко О. М., Закутній С. О. Нечітка логіка в задачах визначення економічних параметрів виконання проєктів. Сучасний стан наукових досліджень та технологій в промисловості. 2024. № 1(27). С. 96–108. DOI: <https://doi.org/10.30837/ITSSI.2024.27.096>
38. Breiman L. Random Forests. *Machine Learning*. 2001. Vol. 45, No. 1. P. 5–32. DOI: <https://doi.org/10.1023/A:1010933404324>
39. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York : ACM, 2016. P. 785–794. DOI: <https://doi.org/10.1145/2939672.2939785>
40. Friedman J. H. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*. 2001. Vol. 29, No. 5. P. 1189–1232. DOI: <https://doi.org/10.1214/aos/1013203451>
41. Ribeiro M. T., Singh S., Guestrin C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York : ACM, 2016. P. 1135–1144. DOI: <https://doi.org/10.1145/2939672.2939778>
42. Lundberg S. M., Lee S.-I. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*. 2017. Vol. 30. P. 4768–4777. DOI: <https://doi.org/10.48550/arXiv.1705.07874>
43. Shapley L. S. A Value for n-Person Games. *Contributions to the Theory of Games II* / eds. H. W. Kuhn, A. W. Tucker. Princeton : Princeton University Press, 1953. P. 307–317.

44. From local explanations to global understanding with explainable AI for trees / S. M. Lundberg et al. *Nature Machine Intelligence*. 2020. Vol. 2, No. 1. P. 56–67. DOI: <https://doi.org/10.1038/s42256-019-0138-9>

45. Grieves M., Vickers J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. *Transdisciplinary Perspectives on Complex Systems*. Springer, Cham, 2017. P. 85–113. DOI: [https://doi.org/10.1007/978-3-319-38756-7\\_4](https://doi.org/10.1007/978-3-319-38756-7_4)

46. Бакурова А. В., Білий В. В. Сценарний розрахунок вартості відбудови пошкодженого об'єкта інфраструктури: підхід на основі цифрового двійника. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2026. № 363(2). С. 324–334. DOI: <https://doi.org/10.31891/2307-5732-2026-363-45>

47. Овчарук І., Горбенко О. Single Page Application для застосунку відстеження погодних умов. *Цифрова платформа: інформаційні технології в соціокультурній сфері*. 2021. Т. 4, № 2. С. 241–249. DOI: <https://doi.org/10.31866/2617-796X.4.2.2021.247490>

48. Laravel Documentation. URL: <https://laravel.com/docs> (last accessed: 05.01.2025).

49. Vue.js Documentation. URL: <https://vuejs.org/> (last accessed: 05.01.2025).

50. Анісімов В. Г., Кунанець Н. Е. Перехід від монолітної до мікросервісної архітектури: методологія та досвід впровадження. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2024. № 55. С. 30–41. DOI: <https://doi.org/10.36910/6775-2524-0560-2024-55-03>

51. Di Francesco P., Lago P., Malavolta I. Architecting with microservices: A systematic mapping study. *Journal of Systems and Software*. 2019. Vol. 150. P. 77–97. DOI: <https://doi.org/10.1016/j.jss.2019.01.001>

52. Білий В. В., Бакурова А. В. Огляд підходів побудови та приклад реалізації інформаційно-аналітичної системи. *Тиждень науки-2025. Факультет комп'ютерних наук і технологій : тези доповідей науково-практичної*

конференції, Запоріжжя, 14–18 квітня 2025. Запоріжжя : НУ «Запорізька політехніка», 2025. С. 403.

53. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830.

54. Prayogi A. A., Niswar M., Indrabayu, Rijal M. Design and implementation of REST API for academic information system. *IOP Conference Series: Materials Science and Engineering*. 2020. Vol. 875. P. 012047. DOI: <https://doi.org/10.1088/1757-899X/875/1/012047>

55. Bui D. B., Puzikova A. V. Theory of normalization in relational databases: Current state. *Scientific Notes of NaUKMA. Computer Science*. 2015. Vol. 177. P. 83–92.

56. Bilyi V. V., Bakurova A. V., Tereschenko E. V., Didenko A. V. Analytics Module for the System for Recording Destruction Due to Russian Aggression. *European Association of Geoscientists & Engineers: 17th International Conference Monitoring of Geological Processes and Ecological Condition of the Environment*, Nov. 2023. 2023. P. 1–5. DOI: <https://doi.org/10.3997/2214-4609.2023520232>

57. Codd E. F., Codd S. B., Salley C. T. Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate. E. F. Codd & Associates, 1993. 24 p.

58. Білий В. В., Бакурова А. В. Організація аналізу багатовимірних даних в системі фіксації руйнувань внаслідок російської агресії. Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій : тези доповідей XII Міжнародної науково-практичної конференції, Запоріжжя, 10–12 грудня 2024. Запоріжжя : НУ «Запорізька політехніка», 2024. С. 426.

59. Білий В. В., Бакурова А. В. Система фіксації руйнувань внаслідок російської агресії. Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій : тези доповідей XI Міжнародної науково-практичної конференції, Запоріжжя, 12–14 грудня 2022. Запоріжжя : НУ «Запорізька політехніка», 2022. С. 132.

60. Prozorro : система публічних електронних закупівель України. URL: <https://prozorro.gov.ua/> (дата звернення: 05.01.2025).

61. Hyndman R. J., Koehler A. B. Another Look at Measures of Forecast Accuracy. *International Journal of Forecasting*. 2006. Vol. 22. P. 679–688. DOI: <https://doi.org/10.1016/j.ijforecast.2006.03.001>

62. Bakurova A., Bilyi V. Development of an approach for predicting the cost of damaged infrastructure recovery with microservice implementation. *Technology Audit and Production Reserves*. 2025. Vol. 5, No. 2(85). P. 33–39. DOI: <https://doi.org/10.15587/2706-5448.2025.339773>

## ДОДАТОК А.

### Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації

#### 1. Наукові праці, в яких опубліковані основні наукові результати дисертації:

##### 1.1. Статті у наукових фахових виданнях України:

1. Бакурова А. В., Білий В. В. Розробка інформаційно-аналітичної системи для фіксації руйнувань внаслідок російської агресії. *Комунальне господарство міст*. 2025. Т. 3, Вип. 191. С. 2–10. DOI: <https://doi.org/10.33042/2522-1809-2025-3-191-2-10>

2. Бакурова А. В., Білий В. В. Сценарний розрахунок вартості відбудови пошкодженого об'єкта інфраструктури: підхід на основі цифрового двійника. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2026. № 363(2). С. 324–334. DOI: <https://doi.org/10.31891/2307-5732-2026-363-45>

##### 1.2. Статті у наукових фахових виданнях України, які входять до міжнародних наукометричних баз:

3. Bakurova A., Bilyi V. Development of an approach for predicting the cost of damaged infrastructure recovery with microservice implementation. *Technology Audit and Production Reserves*. 2025. Vol. 5, No. 2(85). P. 33–39. DOI: <https://doi.org/10.15587/2706-5448.2025.339773> (Індексовано в міжнародній НМБД Scopus)

#### 2. Наукові праці, які засвідчують апробацію матеріалів дисертації:

##### 2.1. Матеріали міжнародних наукових та науково-практичних конференцій, збірники яких входять до міжнародних наукометричних баз:

4. Bilyi V. V., Bakurova A. V., Tereschenko E. V., Didenko A. V. Analytics Module for the System for Recording Destruction Due to Russian Aggression. *European Association of Geoscientists & Engineers: 17th International Conference Monitoring of Geological Processes and Ecological Condition of the Environment*,

Nov. 2023. 2023. P. 1–5. DOI: <https://doi.org/10.3997/2214-4609.2023520232>  
(Індексовано в міжнародній НМБД Scopus)

## **2.2. Матеріали та тези доповідей наукових та науково-практичних конференцій:**

5. Білий В. В. Система фіксації руйнувань внаслідок російської агресії. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій* : тези доповідей XI Міжнародної науково-практичної конференції, Запоріжжя, 12–14 грудня 2022. Запоріжжя : НУ «Запорізька політехніка», 2022. С. 132.

6. Білий В. В., Бакурова А. В. Організація аналізу багатовимірних даних в системі фіксації руйнувань внаслідок російської агресії. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій* : тези доповідей XII Міжнародної науково-практичної конференції, Запоріжжя, 10–12 грудня 2024. Запоріжжя : НУ «Запорізька політехніка», 2024. С. 426.

7. Білий В. В., Бакурова А. В. Огляд підходів побудови та приклад реалізації інформаційно-аналітичної системи. *Тиждень науки-2025. Факультет комп'ютерних наук і технологій* : тези доповідей науково-практичної конференції, Запоріжжя, 14–18 квітня 2025. Запоріжжя : НУ «Запорізька політехніка», 2025. С. 403.

8. Білий В. В. Формалізація задачі прийняття рішення щодо відбудови пошкоджених об'єктів інфраструктури в умовах невизначеності. *Інформаційні технології: теорія і практика* : тези доповідей III (IX) Міжнародної науково-практичної конференції, Харків – Запоріжжя – Дніпро, 25–27 березня 2026. С. 37.

### **Відомості про апробацію результатів дисертації**

1. XI Міжнародна науково-практична конференція «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій», Запоріжжя, 12–14 грудня 2022 р., дистанційно.

2. Науково-популярний захід «Ніч молодіжної науки – 2023 в умовах війни» (Рада молодих учених при МОН України), 18 травня 2023 р., дистанційно.
3. 17th International Conference “Monitoring of Geological Processes and Ecological Condition of the Environment” (European Association of Geoscientists & Engineers), листопад 2023 р., дистанційно.
4. XII Міжнародна науково-практична конференція «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій», Запоріжжя, 10–12 грудня 2024 р., дистанційно.
5. International Scientific and Practical Conference “Business Ecosystems: Sustainable Development 2025” (BESD-2025), 3–7 лютого 2025 р., дистанційно.
6. Науково-практична конференція «Тиждень науки-2025. Факультет комп’ютерних наук і технологій», Запоріжжя, 14–18 квітня 2025 р., заочно.
7. III (IX) Міжнародна науково-практична конференція «Інформаційні технології: теорія і практика», Харків – Запоріжжя – Дніпро, 25–27 березня 2026 р., дистанційно.

**ДОДАТОК Б.****Акти про використання та впровадження результатів дисертаційної роботи**

## АКТ

про впровадження теоретичних результатів дисертаційної роботи

Білого Віталія Вікторовича

«Підтримка прийняття рішень щодо відновлення пошкоджених об'єктів за умов невизначеності на основі інтелектуальних технологій»

Цим актом підтверджується, що в діяльність Запорізької районної ради Запорізької області впроваджено теоретичні результати дисертаційного дослідження Білого Віталія Вікторовича, аспіранта кафедри системного аналізу та обчислювальної математики Національного університету «Запорізька політехніка», в частині інформаційно-аналітичної підтримки прийняття управлінських рішень щодо фіксації пошкоджень та планування відновлення об'єктів, пошкоджених внаслідок збройної агресії Російської Федерації.

Система може використовуватися при обґрунтуванні пріоритетності включення пошкоджених об'єктів до переліків, що подаються на розгляд у межах програм відбудови району. Серед можливостей системи, які запропоновані до роботи районної ради, – внесення даних про пошкоджені об'єкти з фотофіксацією для формування систематизованої інформаційної бази, отримання аналітичних зведень за періодом, категорією та типом об'єктів при підготовці інформаційних довідок, а також розрахунок орієнтовних оцінок вартості відновлення для обґрунтування попередньої потреби у фінансуванні – до розробки проектно-кошторисної документації.

Доступ до даних системи обмежено за рольовою моделлю і надається виключно уповноваженим особам відповідно до вимог законодавства України про захист персональних даних та інформацію з обмеженим доступом.

Використання зазначених результатів дасть змогу підвищити обґрунтованість рішень щодо планування відбудови об'єктів в умовах обмеженості фінансових ресурсів та неповноти вхідних даних, а також забезпечити аналітичну підтримку при плануванні заходів з відновлення.

Даний акт не є підставою для взаємних фінансових розрахунків.

Голова ЗРРЗО



Філін О. О.



## ДОДАТОК В.

### Фрагменти коду розроблених алгоритмів прогнозування, інтерпретації та сценарного аналізу вартості відновлення

Фрагмент коду трикутної функції належності та обчислення її параметрів:

```
def triangular_mf(x, a, b, c):
    if x <= a or x >= c:
        return 0.0
    elif x <= b:
        return (x - a) / (b - a) if b != a else 1.0
    else:
        return (c - x) / (c - b) if c != b else 1.0

def compute_fuzzy_params(positions):
    params = []
    n = len(positions)
    for i, center in enumerate(positions):
        if n == 1:
            params.append((center - 0.5, center, center + 0.5))
        elif i == 0:
            right_dist = positions[1] - center
            params.append((center - right_dist, center, positions[1]))
        elif i == n - 1:
            left_dist = center - positions[i - 1]
            params.append((positions[i - 1], center, center + left_dist))
        else:
            params.append((positions[i - 1], center, positions[i + 1]))
    return params
```

Фрагмент коду для fuzzy-кодування порядкових ознак:

```
class FuzzyEncoder:
    def __init__(self, config):
        self.config = config
        self.category_to_position = {
            cat: pos for cat, pos in zip(config.categories, config.positions)
        }
        self.fuzzy_params = compute_fuzzy_params(config.positions)
        self.output_columns = [
            f"{config.name}_{cat}" for cat in config.categories
        ]

    def encode_value(self, category):
        x = self.category_to_position[category]
        memberships = [
            triangular_mf(x, *params) for params in self.fuzzy_params
        ]
        return np.array(memberships)

    def encode_series(self, series):
        encoded = np.vstack([self.encode_value(v) for v in series])
        return pd.DataFrame(
            encoded, columns=self.output_columns, index=series.index
```

)

### Фрагмент коду для навчання моделі XGBoost:

```
df = pd.read_csv("restoration_data.csv", encoding="utf-8-sig")
y = df["restoration_cost"]
X_raw = df.drop(columns=["restoration_cost"])
X, encoder = preprocess_for_training(X_raw)

x_train, x_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=1
)

model = xgb.XGBRegressor(
    objective="reg:squarederror", random_state=1
)
model.fit(x_train, y_train)

mae = mean_absolute_error(y_test, model.predict(x_test))
joblib.dump(model, MODEL_PATH)
```

### Фрагмент коду агрегації локальних SHAP-значень за групами ознак:

```
def aggregate_local_contrib(shap_vec, feature_columns, feature_groups):
    per_feature = {
        feat: float(val)
        for feat, val in zip(feature_columns, shap_vec)
    }
    per_group = {}
    for g, cols in feature_groups.items():
        if not isinstance(cols, list):
            continue
        per_group[g] = sum(per_feature.get(c, 0.0) for c in cols)

    denom = sum(abs(v) for v in per_group.values()) or 1.0
    per_group_pct = {
        g: (abs(v) / denom) * 100.0 for g, v in per_group.items()
    }
    ordered = sorted(
        per_group.items(), key=lambda kv: abs(kv[1]), reverse=True
    )
    return ordered, per_group_pct
```

### Фрагмент коду інфляційного коригування вартості:

```
BASE_YEAR = 2025
BASE_MONTH = 1

class InflationService:
    def get_coefficient(self, work_year, work_month, indices=None):
        if work_year is None and work_month is None:
            return 1.0
        if not indices:
            raise ValueError(
                "Не передано таблицю інфляційних коефіцієнтів"
```

```

    )

    for row in indices:
        y = row['year']          if isinstance(row, dict) else row.year
        m = row['month']         if isinstance(row, dict) else row.month
        v = row['index_value']   if isinstance(row, dict) else row.index_value
        if y == work_year and m == work_month:
            return float(v)

    raise ValueError(
        f"Коефіцієнт для {work_year}-{work_month:02d} відсутній"
    )

def apply(self, base_cost, work_year, work_month, indices=None):
    k = self.get_coefficient(work_year, work_month, indices)
    return {
        "adjusted_cost": round(base_cost * k, 2),
        "inflation_k": k,
        "base_year": BASE_YEAR,
        "base_month": BASE_MONTH,
        "work_year": work_year or BASE_YEAR,
        "work_month": work_month or BASE_MONTH,
    }

```

### Фрагмент коду REST-ендпоінту прогнозування з SHAP-поясненням:

```

@app.post("/predict_explain")
def predict_explain():
    if not check_key(request):
        return jsonify({"error": "unauthorized"}), 401

    payload = Payload.model_validate(request.get_json(force=True))
    input_data = payload.model_dump(
        exclude={"work_year", "work_month", "inflation_indices"}
    )
    X = preprocess_input(input_data)
    y_hat = float(MODEL.predict(X)[0])

    inflation = inflation_svc.apply(
        y_hat,
        payload.work_year,
        payload.work_month,
        payload.inflation_indices,
    )

    explainer = shap.TreeExplainer(MODEL)
    shap_values = explainer.shap_values(X)
    local = shap_values[0]
    base_value = float(explainer.expected_value)

    feature_groups = get_feature_groups_for_aggregation()
    ordered, per_group_pct = aggregate_local_contrib(
        local, FEATURE_COLUMNS, feature_groups
    )

    contributions = [
        {

```

```

        "group": g,
        "contribution": float(val),
        "percent": float(per_group_pct[g]),
    }
    for g, val in ordered
]

return jsonify({
    "predicted_cost": round(y_hat, 2),
    "adjusted_cost": inflation["adjusted_cost"],
    "inflation_k": inflation["inflation_k"],
    "base_value": base_value,
    "contributions": contributions,
})

```

### Фрагмент коду крос-валідації для порівняння методів кодування:

```

def evaluate_encoding(df, encode_func, n_folds=5):
    y = df["restoration_cost"].values
    X = encode_func(df)

    kf = KFold(n_splits=n_folds, shuffle=True, random_state=42)
    mae_folds, rmse_folds, r2_folds = [], [], []

    for train_idx, test_idx in kf.split(X):
        X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
        y_train, y_test = y[train_idx], y[test_idx]

        model = xgb.XGBRegressor(
            objective="reg:squarederror",
            random_state=42,
            n_estimators=100,
            max_depth=6,
            learning_rate=0.1,
        )
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)

        mae_folds.append(mean_absolute_error(y_test, y_pred))
        rmse_folds.append(np.sqrt(mean_squared_error(y_test, y_pred)))
        r2_folds.append(r2_score(y_test, y_pred))

    return {
        "mae_mean": np.mean(mae_folds),
        "rmse_mean": np.mean(rmse_folds),
        "r2_mean": np.mean(r2_folds),
        "mae_folds": mae_folds,
    }

onehot = evaluate_encoding(df, encode_onehot)
fuzzy = evaluate_encoding(df, encode_fuzzy)
t_stat, p_value = stats.ttest_rel(onehot["mae_folds"], fuzzy["mae_folds"])

```