

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра інженерії програмного забезпечення в енергетиці

«На правах рукопису»

УДК 004.4

ДО ЗАХИСТУ ДОПУЩЕНО

В.о. завідувача кафедри

_____Олександр КОВАЛЬ

«_____» _____2024р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Інженерія програмного забезпечення
інтелектуальних кібер-фізичних систем в енергетиці»
зі спеціальності 121 Інженерія програмного забезпечення
на тему: «Система для визначення ідентифікаційних ознак об'єктів
морського середовища»

Виконав студент 2 курсу, групи ТВ-21мп Мамотенко Богдан Олегович _____

(шифр, прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент, доц.,к.ф.-м.н. Свинчук Ольга Василівна _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Консультант з розділу 1 доцент, к.т.н. Варава Іван Андрійович _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент професор кафедри ЦТЕ, д.т.н., проф. Отрох С.І. _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2024

**Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра інженерії програмного забезпечення в енергетиці

Рівень вищої освіти другий, магістерський

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення
інтелектуальних кібер - фізичних систем в енергетиці»

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри

Олександр КОВАЛЬ

(підпис)

« ____ » _____ 2024р.

**З А В Д А Н Н Я
НА МАГІСТЕРСКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Мамотенко Богдан Олегович

(прізвище, ім'я, по батькові)

1. Тема дисертації: Система для визначення ідентифікаційних ознак об'єктів
морського середовища

Науковий керівник Свинчук Ольга Василівна доц.,к.ф.-м.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «06» листопада 2023 року № 5152-с

2. Строк подання студентом дисертації 10 січня 2024 року

3. Об'єкт дослідження ідентифікація об'єктів морського середовища

4. Предмет дослідження система для визначення ідентифікаційних ознак об'єктів
морського середовища

5. Вихідні дані до роботи система для визначення ідентифікаційних ознак об'єктів
морського середовища

6. Перелік питань, які потрібно розробити аналіз існуючих методів ідентифікації
об'єктів морського середовища, аналіз сучасних технологій обробки морських
даних, вибір та обґрунтування технологій для системи ідентифікації,
моделювання та розробка системи, розробка архітектури та плану впровадження,
розробка архітектури та плану впровадження, аналіз результатів та оптимізація,
підготовка наукової доповіді та документації

7. Орієнтований перелік ілюстративного матеріалу діаграма архітектури системи, діаграма прецедентів, діаграма послідовності, схема зв'язків таблиць у базі даних, огляд існуючих систем, процес налаштування системи

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Варава І.А.		

9. Дата видачі завдання «01» листопада 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	01.11.2022	виконано
2	Дослідження предметної області	01.11.2022 - 01.12.2022	виконано
3	Постановка вимог до проєктування системи	01.12.2022 - 15.12.2022	виконано
4	Дослідження існуючих рішень	15.12.2022 - 03.01.2023	виконано
5	Підготовка публікацій	03.01.2023 - 03.03.2023	виконано
6	Розробка програмного продукту	03.03.2023 - 20.09.2023	виконано
7	Тестування	20.09.2023 - 15.10.2023	виконано
8	Захист програмного продукту	16.10-2023 - 20.10.2023	виконано
9	Підготовка магістерської дисертації	21.10.2023 - 17.11.2023	виконано
10	Передзахист	18.12.2023 – 22.12.2023	виконано
11	Захист	15.01.2024 – 19.01.2024	виконано

Студент

(підпис)

Мамотенко Б.О.

(прізвище та ініціали)

Науковий керівник

(підпис)

Свинчук О.В.

(прізвище та ініціали)

РЕФЕРАТ

Структура і обсяг кваліфікаційної роботи. Магістерська дисертація складається зі вступу, п'яти розділів, висновків та 2 додатків. Робота містить посилання на 27 джерел, 17 рисунків та 7 таблиць. Основна частина роботи викладена на 78 сторінках.

Актуальність. В останні роки, зі зростанням усвідомлення важливості охорони морів, актуальність розробки системи для визначення ідентифікаційних ознак об'єктів морського середовища значно зросла. Цей напрямок досліджень відіграє ключову роль у збереженні біорізноманіття морів і океанів. Наприклад, з точними даними про розподіл та поведінку морських видів, можна ефективніше планувати заходи щодо їх охорони та використання морських ресурсів.

З іншого боку, зміни клімату та забруднення морів ставлять перед наукою нові виклики. Необхідність у моніторингу стану морських екосистем стає все більш актуальною. Система, яка може ідентифікувати зміни в морському середовищі, надає важливі дані для екологічних досліджень та розробки ефективної екологічної політики.

Технологічний прогрес у галузі морських наук також вимагає нових підходів до обробки та аналізу великої кількості даних. Тут розробка системи для визначення ідентифікаційних ознак стає не тільки інструментом для покращення якості наукових досліджень, але й сприяє розширенню можливостей у таких сферах, як морська біологія, екологія, геологія.

Крім того, ця система має важливе значення для безпеки мореплавання. Вона допомагає ідентифікувати потенційні небезпеки для судноплавства, забезпечуючи таким чином безпеку на морі.

Не можна не згадати й про економічний аспект. Глибоке знання про морське середовище може сприяти розвитку морської економіки, в тому числі в галузях рибальства, туризму, видобутку природних ресурсів. Ефективна система

ідентифікації сприяє сталому розвитку цих галузей, забезпечуючи мінімізацію негативного впливу на морське середовище.

Таким чином, розробка системи для визначення ідентифікаційних ознак об'єктів морського середовища є важливим кроком не тільки у науковій сфері, але й у глобальному контексті збереження морських екосистем, безпеки мореплавання та сталого розвитку морської економіки.

Метою роботи є розробка ефективної системи для визначення ідентифікаційних ознак об'єктів морського середовища, яка забезпечуватиме точний та швидкий аналіз морських даних. Система має на меті інтегрувати передові технології обробки даних та аналітичні алгоритми для ідентифікації, класифікації та моніторингу різноманітних об'єктів, що зустрічаються в морському середовищі.

Основні завдання, які покликана вирішити ця система, включають забезпечення точності ідентифікації видів, визначення їхніх характеристик і поведінки, а також аналіз змін, що відбуваються в морському середовищі з часом. Це дозволить не тільки підвищити рівень знань про морські екосистеми, але й сприятиме ефективному використанню морських ресурсів, охороні біорізноманіття та плануванню заходів зі збереження морського середовища.

Важливим аспектом роботи є також розробка інтерфейсу користувача, який би був зручним та інтуїтивно зрозумілим для широкого кола фахівців у сфері морських досліджень, екологів, біологів та інших зацікавлених сторін.

Кінцева мета полягає в створенні надійного, гнучкого та масштабованого інструменту, що зможе використовуватись в різних областях, від наукових досліджень до практичного застосування у галузі морської економіки та охорони довкілля.

Об'єктом дослідження є системи та методи для ідентифікації та аналізу ідентифікаційних ознак об'єктів морського середовища. Це включає вивчення та розробку інструментів та алгоритмів, що дозволяють ефективно виявляти,

класифікувати та відстежувати різноманітні об'єкти морського середовища, такі як морські види, їх поведінку, а також фізичні та хімічні характеристики води та дна.

В рамках дослідження особлива увага приділяється розробці та вдосконаленню методів збору даних, їх обробки та інтерпретації, з метою створення комплексної та інтегрованої системи. Ця система має на меті не тільки забезпечити точне визначення та аналіз ідентифікаційних ознак, але й сприяти глибшому розумінню процесів, що відбуваються в морському середовищі.

Таким чином, об'єктом дослідження є комплексна взаємодія між різними компонентами морського середовища, технологічні рішення для їх аналізу та можливості використання зібраних даних для наукових та практичних цілей.

Предметом дослідження є система для ідентифікації, аналізу та класифікації ідентифікаційних ознак об'єктів морського середовища. Це включає в себе детальне вивчення та оцінку існуючих алгоритмів, програмного забезпечення, технологій обробки даних та методів машинного навчання, які можуть бути застосовані для ефективного збору, обробки та інтерпретації морських даних.

Особлива увага приділяється розробці нових або вдосконаленню існуючих методів для поліпшення точності та ефективності визначення характеристик морських об'єктів, таких як види морських істот, їхні життєві цикли, міграційні шляхи, а також фізико-хімічні параметри морського середовища.

В рамках дослідження також розглядаються можливості інтеграції цих методів у єдину систему, яка зможе автоматизувати процес ідентифікації та аналізу, забезпечуючи високий рівень точності та надійності у визначенні ідентифікаційних ознак морських об'єктів.

Методи дослідження, які застосовуються в роботі над проектом «Система для визначення ідентифікаційних ознак об'єктів морського середовища», включають ряд комплексних та багатогранних підходів.

- Аналіз літератури: огляд наявних наукових публікацій, статей, книг та досліджень, присвячених методам ідентифікації об'єктів морського середовища,

використанням хмарних технологій у обробці даних, а також загальним питанням морської екології та біології.

- Моделювання та комп'ютерне симулювання: розробка та тестування математичних моделей і комп'ютерних симуляцій для перевірки гіпотез та валідації методів аналізу даних. Це включає створення моделей машинного навчання та штучного інтелекту для класифікації та ідентифікації об'єктів.

- Експериментальні методи: проведення польових досліджень та експериментів для збору даних, які можуть бути використані для тестування та вдосконалення розробленої системи. Використання спеціалізованого обладнання для збору даних про морське середовище.

- Статистичний аналіз даних: застосування статистичних методів для аналізу зібраних даних, включаючи оцінку точності, надійності та валідності інформації та методів її обробки.

- Критичний аналіз та синтез: оцінка і інтеграція отриманих даних і результатів з наявними теоретичними знаннями для формулювання висновків, рекомендацій та пропозицій щодо подальшої роботи.

- Ітеративний розвиток та тестування: постійне вдосконалення системи шляхом ітеративного процесу розробки, тестування, аналізу зворотного зв'язку та внесення корективів.

Практичне значення одержаних результатів дослідження полягає в тому, що розроблена система забезпечує:

- підвищення ефективності моніторингу морського середовища: завдяки точній ідентифікації об'єктів морського середовища, система допоможе в покращенні моніторингу біологічного різноманіття, виявленні змін у морських екосистемах та вчасному реагуванні на екологічні загрози.

- забезпечення точності та надійності даних для наукових досліджень: надання точних даних про морське середовище сприятиме розвитку наукових досліджень у галузях морської біології, екології та геології.

- покращення ресурсного планування та управління морськими ресурсами:

ефективне використання системи сприятиме більш точному та ефективному плануванню видобутку та використання морських ресурсів, що підвищує їх сталість та зменшує негативний вплив на морське середовище.

- поліпшення безпеки мореплавства: ідентифікація потенційно небезпечних об'єктів та умов у морському середовищі допоможе у підвищенні безпеки судноплавства.

- сприяння екологічній освіті та підвищенню екологічної обізнаності: використання системи в освітніх цілях забезпечить краще розуміння важливості охорони морського середовища та розвиток екологічної свідомості серед широкого загалу.

Ключові слова: ідентифікація об'єктів, аналіз даних, обробка даних, аудіосигнали, шум, мелспектрограма, кластеризація, Feature extraction, K-means.

ABSTRACT

Structure and size of the Qualification Work. The master's dissertation consists of an introduction, five chapters, conclusions, and 2 appendixes. The work includes references to 27 sources, 17 figures, and 7 tables. The main part of the work is presented in 78 pages.

Relevance. In recent decades, with the growing awareness of the importance of marine environmental protection, the relevance of developing a system for determining the identification features of marine objects has increased significantly. This area of research plays a key role in the conservation of biodiversity in the seas and oceans. For example, with accurate data on the distribution and behavior of marine species, it is possible to plan more efficiently measures for their protection and use of marine resources.

On the other hand, climate change and marine pollution pose new challenges to science. The need to monitor the state of marine ecosystems is becoming increasingly important. A system that can identify changes in the marine environment provides important data for environmental research and the development of effective environmental policies.

Technological advances in marine science also require new approaches to processing and analyzing large amounts of data. Here, the development of an identification system becomes not only a tool for improving the quality of scientific research, but also contributes to expanding opportunities in areas such as marine biology, ecology, and geology.

In addition, this system is essential for maritime safety. It helps identify potential hazards to navigation, thus ensuring safety at sea.

The economic aspect is also worth mentioning. In-depth knowledge of the marine environment can contribute to the development of the maritime economy, including in the areas of fishing, tourism, and natural resource extraction. An effective identification system contributes to the sustainable development of these industries by minimizing the

negative impact on the marine environment.

Thus, the development of a system for determining the identification features of marine objects is an important step not only in the scientific field, but also in the global context of marine ecosystem conservation, maritime safety and sustainable development of the maritime economy.

The purpose is to develop and validate an effective system for determining the identification features of marine objects that will provide accurate and fast analysis of marine data. The system aims to integrate advanced data processing technologies and analytical algorithms to identify, classify and monitor various objects found in the marine environment.

The main tasks that this system is designed to solve include ensuring the accuracy of species identification, determining their characteristics and behavior, and analyzing changes in the marine environment over time. This will not only increase the level of knowledge about marine ecosystems, but will also contribute to the efficient use of marine resources, biodiversity protection, and planning of marine conservation measures.

An important aspect of the work is also the development of a user interface that would be convenient and intuitive for a wide range of marine researchers, ecologists, biologists and other stakeholders.

The ultimate goal is to create a robust, flexible and scalable tool that can be used in a variety of areas, from scientific research to practical applications in the field of marine economics and environmental protection.

The object of research is systems and methods for identifying and analyzing the identification features of marine environment objects. This includes the study and development of tools and algorithms that allow for the effective detection, classification and tracking of various marine objects, such as marine species, their behavior, and the physical and chemical characteristics of water and the bottom.

The research focuses on the development and improvement of data collection, processing, and interpretation methods to create a comprehensive and integrated system. This system aims not only to ensure accurate identification and analysis of identification

features, but also to promote a deeper understanding of the processes occurring in the marine environment.

Thus, the object of research is the complex interaction between different components of the marine environment, technological solutions for their analysis and the possibility of using the collected data for scientific and practical purposes.

The subject of the study is system to identify, analyze and classify the identification features of marine environment objects. This includes a detailed study and evaluation of existing algorithms, software, data processing technologies and machine learning methods that can be applied to efficiently collect, process and interpret marine data.

Particular attention is paid to the development of new methods or improvement of existing methods to improve the accuracy and efficiency of determining the characteristics of marine objects, such as species of marine life, their life cycles, migration routes, and physical and chemical parameters of the marine environment.

The study also considers the possibilities of integrating these methods into a single system that can automate the identification and analysis process, ensuring a high level of accuracy and reliability in determining the identification features of marine objects.

The research methods used in the work on the project "System for determining the identification features of marine environment objects" include a number of complex and multifaceted approaches.

- Literature analysis: A review of existing scientific publications, articles, books and research on methods of identifying marine objects, the use of cloud technologies in data processing, as well as general issues of marine ecology and biology.

- Modeling and computer simulation: Developing and testing mathematical models and computer simulations to test hypotheses and validate data analysis methods. This includes creating machine learning and artificial intelligence models for object classification and identification.

- Experimental methods: Conducting field studies and experiments to collect data that can be used to test and improve the developed system. Use of specialized equipment

to collect data about the marine environment.

- Statistical analysis of data: The application of statistical methods to analyze the collected data, including the assessment of the accuracy, reliability and validity of the information and its processing methods.

- Critical analysis and synthesis: Evaluation and integration of the data and results with existing theoretical knowledge to formulate conclusions, recommendations and suggestions for further work.

The practical significance of the research results is that the developed system provides:

- increased efficiency of marine environment monitoring: due to the accurate identification of marine environment objects, the system will help to improve the monitoring of biodiversity, detect changes in marine ecosystems and respond to environmental threats in a timely manner;

- ensuring accuracy and reliability of data for scientific research: the provision of accurate data on the marine environment will facilitate the development of scientific research in the fields of marine biology, ecology and geology;

- improved resource planning and management of marine resources: the effective use of the system will contribute to more accurate and efficient planning of the extraction and use of marine resources, which increases their sustainability and reduces the negative impact on the marine environment;

- improving maritime safety: identification of potentially hazardous objects and conditions in the marine environment will help to improve the safety of navigation;

- promoting environmental education and awareness: the use of the system for educational purposes will provide a better understanding of the importance of marine environmental protection and the development of environmental awareness among the general public.

Keywords: object identification, data analysis, data processing, audio signals, noise, melospectrogram, clustering, Feature extraction, K-means.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
ВСТУП	10
1 ПОСТАНОВКА ЗАДАЧІ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ІДЕНТИФІКАЦІЙНИХ ОЗНАК ОБ'ЄКТІВ МОРСЬКОГО СЕРЕДОВИЩА	11
1.1 Вимоги до системи та функціонал системи	12
1.2 Систематизація процесу розробки системи	13
1.3 Призначення продукту та його користувачі	14
Висновки до розділу 1	15
2 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ	16
2.1 Географічні інформаційні системи	16
2.2 Системи обробки супутникових зображень	20
2.3 Системи обробки акустичних даних	22
2.4 Програмне забезпечення сонарів	24
2.5 Програмне забезпечення лідарів	25
2.6 Програмне забезпечення радарів	27
2.7 Порівняння існуючих систем	29
Висновки до розділу 2	30
3 ЗАСОБИ РОЗРОБКИ СИСТЕМИ ДЛЯ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ МОРСЬКОГО СЕРЕДОВИЩА	32
3.1 Мова програмування Python	33
3.2 База даних SQLite	36
3.3 Система контролю версій Git	37

3.4 Бібліотеки та середовище розробки	39
3.4.1 Бібліотека для роботи з математичними обчисленнями Numpy	39
3.4.2 Бібліотека для логування Loguru	41
3.4.3 Інтегроване середовище розробки PyCharm	42
3.4.4 Бібліотека Scikit-learn	43
3.4.5 Бібліотека librosa	45
3.4.6 Бібліотека Matplotlib	46
3.5 Методи	48
3.5.1 Feature Extraction	48
3.5.2 Метод кластеризації DBSCAN	49
3.5.3 Метод кластеризації K-means	50
Висновки до розділу 3	52
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	54
4.1 Архітектурне моделювання діаграм системи	54
4.1.1 Діаграми прецедентів	55
4.2 Структура програмного забезпечення	57
4.3 Робота користувача з програмною системою	59
Висновки до розділу 4	62
5 РОЗРОБКА СТАРТАП-ПРОЄКТУ	64
5.1 Опис ідеї проєкту	64
5.2 Технологічний аудит ідеї проєкту	66
5.3 Аналіз ринкових можливостей стартап-проєкту	67
5.4 Розроблення ринкової стратегії продукту	70
5.5 Розроблення маркетингової програми стартап-проєкту	72

Висновки до розділу 5	73
ВИСНОВКИ	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76
ДОДАТКИ	79
ДОДАТОК А Лістинг розробленої програми	79
ДОДАТОК Б Презентація	93

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

Feature Extraction — процес виявлення ознак вхідного сигналу;

ML (Machine Learning) — машинне навчання; методи машинного навчання;

IT (Information Technologies) — інформаційні технології;

SDK (Software Development Kit) — комплект програмного забезпечення;

SQL (Structured Query Language) — стандартна мова запитів для взаємодії з реляційними базами даних, використовується для вставки, оновлення, видалення та вибору даних;

DBSCAN – метод кластеризації;

K-means – метод кластеризації;

БД — база даних;

МП — мова програмування;

ОС — операційна система;

ПЗ — програмне забезпечення.

ВСТУП

В епоху швидкого розвитку технологій, їх всеосяжної інтеграції у різні сфери життя, морська галузь заслуговує особливої уваги. Однією з ключових вимог для успішної діяльності у цій сфері є здатність до точного та швидкого визначення ідентифікаційних ознак об'єктів морського середовища.

Магістерська дисертація зосереджена на розробці та детальному аналізі системи для визначення ідентифікаційних ознак об'єктів морського середовища.

У вступі дисертації детально обґрунтовано актуальність теми, визначено цілі та завдання дослідження. Окреслено об'єкт та предмет дослідження, висвітлено методологічні підходи та теоретичну базу роботи. Представлено структуру дисертації, описано основний зміст кожного розділу та зазначено ключові етапи виконання проекту.

Дисертація складається з п'яти розділів:

Теоретичні основи: опис теоретичних аспектів ідентифікації об'єктів морського середовища, аналіз існуючих методів та підходів.

Перегляд сучасних технологій: аналіз сучасних інструментів та технологій, що використовуються для обробки даних у морській сфері.

Розробка системи: представлення архітектури та ключових компонентів системи.

Аналіз результатів тестування: детальний огляд результатів тестування системи, ідентифікація потенційних проблем та розробка рішень.

Оцінка ефективності системи: аналіз переваг та недоліків системи, розгляд можливостей її подальшого вдосконалення.

Ця дисертація призначена для розробки високоефективного інструменту для морської галузі, який забезпечить швидке та точне визначення ідентифікаційних ознак об'єктів морського середовища, сприяючи підвищенню безпеки, ефективності та сталому розвитку морської галузі.

1 ПОСТАНОВКА ЗАДАЧІ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ІДЕНТИФІКАЦІЙНИХ ОЗНАК ОБ'ЄКТІВ МОРСЬКОГО СЕРЕДОВИЩА

Задача створення системи для визначення ідентифікаційних ознак об'єктів морського середовища відкриває широкі перспективи для дослідження та розвитку технологій в області морських наук та інженерії. В контексті сучасного інформаційного суспільства, це стає особливо актуальним завданням, враховуючи швидке зростання обсягів даних і потребу в їх ефективному аналізі та обробці. Головна мета цього проекту полягає у створенні ефективної та надійної системи для ідентифікації та аналізу об'єктів морського середовища, що дозволить отримувати важливу інформацію для наукових досліджень, моніторингу екосистеми та інших застосувань.

Об'єктом дослідження є методи для ідентифікації та аналізу ідентифікаційних ознак об'єктів морського середовища, а предметом дослідження – система ідентифікації та аналізу ідентифікаційних ознак. Дана робота вимагає комплексного підходу, що включає в себе використання різних інструментів і методів, від збору даних до їх аналізу та інтерпретації.

Завдання магістерської дисертації:

- збір та передача даних з морського середовища;
- попередня обробка та аналіз даних;
- ідентифікація об'єктів;
- аналіз інформації та отримання результатів.

Специфікація функціоналу системи включає:

- інтеграція з різними датчиками та інструментами для збору даних;
- використання алгоритмів машинного навчання для ідентифікації та класифікації об'єктів;

- розробка інтуїтивно зрозумілого інтерфейсу для користувачів;
- забезпечення можливості віддаленого доступу та управління системою;
- гарантування безпеки та захисту даних.

Завдяки використанню сучасних технологій та інноваційних підходів, цей проект покликаний не тільки вирішити актуальні завдання, але і забезпечити можливість масштабування та розширення системи в майбутньому, відповідаючи таким чином на виклики часу та зростаючі потреби в обробці великих обсягів морських даних.

1.1 Вимоги до системи та функціонал системи

1. Визначення вимог та специфікацій:

- збір та аналіз вимог від зацікавлених сторін, включаючи морських науковців, екологів та інших фахівців;
- встановлення технічних специфікацій системи на основі аналізу вимог.

2. Проектування та розробка архітектури системи:

- вибір та проектування архітектури системи, включаючи обрання необхідних технологій та платформ;
- розробка модульної структури системи для забезпечення гнучкості та масштабованості.

3. Розробка програмного забезпечення:

- кодування та імплементація основних функціональних модулів системи;
- інтеграція зовнішніх API та сервісів за потребою.

4. Тестування системи:

- проведення тестування на різних рівнях (модульне, інтеграційне, системне);
- залучення користувачів для бета-тестування та збору зворотного зв'язку.

5. Розгортання та інтеграція:

- розгортання системи в реальних умовах;
- інтеграція з існуючими системами користувачів.

6. Тренінг та підтримка користувачі:

- проведення тренінгів для кінцевих користувачів;
- надання технічної підтримки та оновлень.

7. Моніторинг та оцінка продуктивності:

- постійний моніторинг роботи системи;
- оцінка продуктивності та збір зворотного зв'язку для подальших удосконалень.

Функціонал системи.

- Ідентифікація та класифікація об'єктів – автоматизоване розпізнавання та класифікація об'єктів морського середовища.
- Аналітичні засоби – розширені інструменти для аналізу даних, включаючи статистичний аналіз та візуалізацію.
- Інтеграція з датчиками та іншими системами – сумісність з різними типами датчиків та здатність інтегруватися з іншими системами моніторингу.
- Безпека та конфіденційність даних – забезпечення високого рівня безпеки та захисту конфіденційних даних.
- Масштабованість та гнучкість – можливість легко масштабувати систему згідно з змінюваними потребами та умовами.
- Дистанційний доступ та управління – функціонал для дистанційного доступу та управління системою.

1.2 Систематизація процесу розробки системи

Організація процесу розробки програмного забезпечення є ключовим етапом, який вимагає впровадження системного та ефективного підходу для досягнення

поставлених цілей у обмежені терміни. Використання стратегії розбивки завдань на дрібніші компоненти є важливим кроком у цьому процесі, адже це дозволяє рівномірно розподілити роботу, забезпечити краще розуміння кожного аспекту проекту та мінімізувати ризики та помилки.

Декомпозиція задач полягає у розподілі складної задачі або системи на більш малі, керовані та зрозумілі частини. Кожну із цих частин можна аналізувати, розв'язувати чи розробляти окремо, оскільки їх обробка стає більш простою та зрозумілою.

У контексті розробки системи для визначення ідентифікаційних ознак об'єктів морського середовища, задачі були поділені на підзадачі та організовані за допомогою дошки Azure [7]. Це допомогло створити чіткий план розробки та інтеграції різних компонентів системи.

Використання дошки Azure сприяло створенню злагодженого механізму управління проектом, де кожен учасник команди мав чітке розуміння своїх обов'язків та відповідальностей. Це забезпечило ефективну співпрацю та взаємодію всередині команди, створивши міцну основу для успішної реалізації проекту.

1.3 Призначення продукту та його користувачі

Призначення розробленої системи для визначення ідентифікаційних ознак об'єктів морського середовища полягає в наданні високоефективного інструменту для точної ідентифікації, аналізу та класифікації різноманітних об'єктів, які зустрічаються у морських екосистемах. Цей продукт розроблений з метою покращення наукових досліджень у галузі морської біології, екології та інших суміжних сфер, де важливо мати детальне розуміння морського середовища.

Основними користувачами цього продукту є науковці та дослідники: фахівці у сфері морської науки та екології, які займаються дослідженнями морських екосистем, вивченням морських видів, їхнього середовища проживання та

поведінки. Екологічні організації: організації, що працюють у сфері охорони навколишнього середовища, забезпечуючи моніторинг стану морських екосистем та розробку заходів з їх збереження. Урядові агенції: державні органи, що відповідають за регулювання використання морських ресурсів, моніторинг стану морського середовища та розробку політик збереження морського середовища. Освітні установи: вищі навчальні заклади, які займаються підготовкою спеціалістів у галузі морської науки та екології, а також проводять науково-дослідницьку роботу. Приватні компанії: компанії, що займаються морськими дослідженнями, розробкою та впровадженням технологій для моніторингу та використання морських ресурсів.

Ця система дозволяє користувачам ефективно обробляти великі обсяги даних, отримувати важливу інформацію для досліджень і прийняття обґрунтованих рішень, пов'язаних з охороною та використанням морських ресурсів.

Висновки до розділу 1

Основною метою даної роботи є створення вузькоспеціалізованої системи для визначення ідентифікаційних ознак об'єктів морського середовища для подальшої ідентифікації об'єктів морського середовища. Особлива увага була приділена технічній стороні проект, але також була приділена увага організаційній стороні проекту. Отримано структуру побудови проекту та знання яким має бути кінцевий продукт. Розроблена система має відповідати вимогам точності класифікації та масштабованості для майбутніх модифікацій.

Розробка також охоплює тестування, та за допомогою Azure dashboard спрощує цей процес.

2 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ

Аналіз існуючих систем, що застосовуються для визначення ідентифікаційних ознак об'єктів морського середовища, є важливим кроком у розробці нових, більш ефективних та цілеспрямованих технологічних рішень. Вивчення та оцінка поточного стану технологій дозволяють виявити потенційні прогалини, визначити області для подальших інновацій та розвитку. У сучасному світі морське середовище стає предметом зростаючої уваги через його важливість для екології, економіки, науки та безпеки. Це вимагає розробки комплексних систем, які могли б не тільки збирати дані, але й аналізувати їх, визначаючи ключові характеристики об'єктів, що знаходяться у водному середовищі. Такі системи повинні бути здатні ефективно обробляти великі обсяги даних, надаючи точну та актуальну інформацію.

Сучасні системи аналізу морського середовища охоплюють широкий спектр технологій, від супутникового моніторингу та дистанційного зондування до підводних дослідницьких апаратів. Кожна з цих систем має свої переваги та обмеження і використовується для конкретних цілей, наприклад, для моніторингу міграції морських видів, вивчення рельєфу дна або виявлення забруднень.

В цьому контексті, аналіз існуючих систем є ключовим для розуміння потенціалу і обмежень поточних технологічних підходів та для визначення напрямків подальшого розвитку. Це також сприяє ідентифікації необхідних інновацій і покращень у створенні більш ефективної системи для визначення ідентифікаційних ознак об'єктів морського середовища.

2.1 Географічні інформаційні системи

Географічні інформаційні системи (GIS) — це комплексні інструменти, що використовуються для збору, аналізу, управління та візуалізації геопросторових

даних. Вони дозволяють користувачам переглядати, розуміти, інтерпретувати та візуалізувати дані у вигляді карт, графіків, звітів, тощо. GIS використовується в широкому спектрі галузей, включаючи міське планування, навігацію, екологічний моніторинг та, звичайно, у дослідженні морського середовища [4].

GIS поєднує просторові дані (які мають географічну компоненту) та не просторові дані. Просторові дані можуть включати картографічну інформацію, супутникові знімки, дані з датчиків тощо, в той час як не просторові дані можуть включати статистичну інформацію, текстові описи тощо. GIS дозволяє інтегрувати ці дані, проводити їх аналіз, і створювати на їх основі різні візуалізації та моделі. Візуалізація роботи GIS показана на рисунку 2.1.

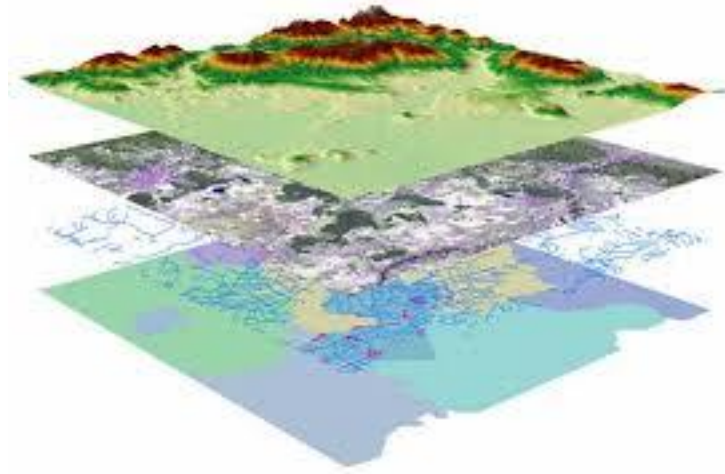


Рисунок 2.1 – Візуалізація GIS

GIS використовуються у широкому спектрі областей і застосувань, оскільки вони дозволяють ефективно збирати, аналізувати, управляти та візуалізувати геопросторові дані, завдяки чому GIS знаходить активне застосування в таких галузях.

- Міське планування та управління земельними ресурсами: GIS використовується для планування міст, забудови, дорожньої інфраструктури, а також для управління земельними ресурсами, зонування, і планування використання земель.

- Екологія та охорона навколишнього середовища: В області екології GIS допомагає в моніторингу стану навколишнього середовища, вивченні впливу людини на природу, оцінці зміни клімату, та плануванні заходів з охорони природи.
- Лісове та сільське господарство: GIS допомагає в управлінні лісовими ресурсами, оптимізації сільськогосподарських практик, моніторингу стану посівів та управлінні водними ресурсами[2].
- Водні ресурси та гідрологія: Використання GIS для аналізу водних ресурсів, річкових систем, оцінки якості води та планування водозаборів.
- Катастрофічний менеджмент та екстрені служби: GIS використовується для планування реагування на природні та техногенні катастрофи, моніторингу ризиків, координації екстрених служб і швидкого реагування.
- Транспорт та логістика: У сфері транспорту GIS допомагає в оптимізації маршрутів, управлінні транспортними потоками, плануванні розвитку транспортної інфраструктури.
- Наукові дослідження та освіта: В науці та освіті GIS використовується для вивчення різних геопросторових процесів, навчальних цілей, а також для проведення детальних досліджень і аналізів.
- Військові застосування: У військовій сфері GIS використовується для розвідки, планування операцій, управління військовими ресурсами та об'єктами.
- Морські дослідження та рибальство: У морській галузі GIS допомагає в картографуванні морського дна, відстеженні рибних запасів, плануванні морських експедицій та вивченні морських екосистем.

Застосування GIS демонструє його важливість у багатьох галузях сучасного життя, надаючи фахівцям інструменти для аналізу та візуалізації геопросторових даних, що є незамінними для прийняття інформованих рішень та планування.

Переваги GIS.

- Багатофункціональність і гнучкість: GIS дозволяє проводити різноманітний аналіз даних та візуалізації, підтримуючи широкий спектр форматів даних.

- Просторовий аналіз: Вони особливо ефективні для аналізу, який вимагає урахування географічних аспектів, наприклад, для картографування морських регіонів.

- Інтеграція даних: GIS може інтегрувати дані з різних джерел, надаючи комплексне бачення аналізованої проблеми.

Недоліки GIS порівняно з системою даного проєкту.

- Спеціалізація: на відміну від спеціалізованої системи для ідентифікації ознак об'єктів морського середовища, GIS є більш загальним інструментом і може бути не оптимізованим для деяких специфічних завдань, пов'язаних з морськими дослідженнями[6].

- Комплексність використання: GIS часто вимагає спеціалізованих знань для ефективного використання, що може бути бар'єром для користувачів, які не мають географічної чи картографічної підготовки.

- Обмеженість в аналізі даних: хоча GIS ефективний для візуалізації та базового аналізу даних, він може бути обмеженим у виконанні більш складних аналітичних завдань, які можливі в спеціалізованих системах.

Таким чином, хоча GIS є потужним інструментом для роботи з геопросторовими даними, спеціалізована система для визначення ідентифікаційних ознак об'єктів морського середовища [7] може бути більш ефективною в певних аспектах, особливо коли мова йде про специфічні завдання, пов'язані з морськими дослідженнями.

2.2 Системи обробки супутникових зображень

Системи обробки супутникових зображень – це комплексні технологічні рішення, призначені для збору, аналізу та обробки даних, отриманих із супутників. Ці системи використовуються для перетворення супутникових даних у корисну інформацію, яка може бути застосована у різних сферах, від моніторингу довкілля до військового планування.

Супутникові зображення, здобуті з космосу, спочатку є неструктурованими великими масивами даних [10]. Системи обробки супутникових зображень використовують спеціалізоване програмне забезпечення для аналізу цих даних. Вони можуть виявляти певні характеристики, ідентифікувати зміни в ландшафті або водній поверхні, а також проводити геопросторовий аналіз.

Переваги систем обробки супутникових зображень.

- Широкий спектр охоплення: супутники можуть збирати дані на великих територіях, забезпечуючи охоплення, яке неможливо досягти іншими методами.
- Висока частота оновлення: залежно від орбіти супутника, зображення певних регіонів можуть оновлюватися регулярно.
- Мультиспектральний аналіз: супутникові зображення часто включають дані за кількома спектральними каналами, що дозволяє виявляти різноманітні екологічні та географічні явища.

Недоліки порівняно з нашою системою.

- Обмежена деталізація: незважаючи на велику територію покриття, супутникові зображення можуть мати обмеження з точки зору роздільної здатності та деталізації.
- Залежність від погодних умов: погані погодні умови, такі як хмарність або туман, можуть погіршувати якість зображень.

- Технічні та фінансові бар'єри: висока вартість отримання та обробки супутникових зображень, а також потреба у спеціалізованому програмному забезпеченні та експертизі.

Системи обробки супутникових зображень мають широкий спектр застосувань і використовуються в різних галузях. Ось деякі з основних галузей застосування:

- Геодезія та картографія: супутникові зображення використовуються для створення топографічних карт, кадастрових планів, місцевих планів та геодезичних досліджень. Вони допомагають визначати географічні координати, висоти та форму земельних об'єктів.

- Агрономія та сільське господарство: супутникові дані використовуються для моніторингу врожаю, прогнозування [11] погодних умов, виявлення шкідників та захворювань рослин, а також для оптимізації використання землі та ресурсів.

- Екологічний моніторинг: зображення з супутників використовуються для вивчення змін у природних екосистемах, виявлення забруднення навколишнього середовища та контролю за змінами клімату.

- Міське планування та розвиток: геопросторові дані із супутників допомагають визначати оптимальне розміщення інфраструктури, доріг, будівель та розвиток місцевих територій.

- Телекомунікації та навігація: супутникові системи обробки зображень використовуються для розвитку GPS-технологій, навігаційних систем та телекомунікаційних мереж.

- Нафтогазова промисловість: видобувні компанії використовують супутникові дані для виявлення та аналізу резервуарів нафти та газу, виявлення нафтових плям та моніторингу об'єктів нафтовидобутку.

- Оборона та безпека: супутникові зображення використовуються для виявлення військових об'єктів, моніторингу конфліктів, а також для планування та координації військових операцій.

- Дослідження природних явищ: супутникові дані допомагають вивчати природні явища, такі як урагани, землетруси, повені та інші природні катастрофи.

- Моніторинг морського середовища: використовується для виявлення забруднення морських вод, спостереження за рухом морських судів, а також для вивчення морських екосистем.

- Наукові дослідження: супутникові системи обробки зображень використовуються у наукових дослідженнях в різних галузях, таких як астрономія, океанографія, кліматологія та інші.

Ці галузі застосування свідчать про важливість і універсальність систем обробки супутникових зображень в сучасному світі. Вони надають можливість отримувати детальні інформаційні дані з великих територій і використовувати їх для різних цілей, від наукових досліджень до прийняття стратегічних рішень у різних галузях.

2.3 Системи обробки акустичних даних

Системи обробки акустичних даних (Acoustic Data Processing Systems) відіграють важливу роль у багатьох галузях і мають різноманітні застосування.

Системи обробки акустичних даних – це програмні та апаратні комплекси, розроблені для обробки та аналізу звукових сигналів. Вони можуть включати в себе різноманітні акустичні сенсори (мікрофони, гідрофони тощо) та програмне забезпечення для запису, обробки та аналізу акустичних сигналів.

Системи обробки акустичних даних працюють шляхом запису звукових сигналів за допомогою акустичних сенсорів. Записаний аудіо- або звуковий сигнал

потім піддається обробці та аналізу за допомогою спеціальних алгоритмів та програмного забезпечення. Це дозволяє визначити різноманітні параметри акустичного сигналу, такі як частота, амплітуда, часові характеристики тощо.

Переваги.

- Моніторинг навколишнього середовища: системи обробки акустичних даних використовуються для моніторингу навколишнього звукового середовища, що є важливим у галузях як зв'язку, так і охорони навколишнього середовища.

- Судноплавство та підводна активність: вони застосовуються для виявлення суден та інших об'єктів у водних тілах, а також для виявлення звукових сигналів, що пов'язані з підводною активністю.

- Медицина та наука: використовуються для проведення медичних досліджень, включаючи аудіологію та дослідження звукової діагностики.

- Технічні дослідження та контроль якості: системи обробки акустичних даних використовуються для аналізу якості та властивостей звукових матеріалів, таких як аудіозаписи та музика.

Недоліки.

- Шум та перешкоди: акустичні сигнали можуть бути вплинуті різними видами шуму та перешкод, що може ускладнити їхню обробку та аналіз.

- Обмежена точність: точність вимірювань акустичних параметрів може бути обмеженою у певних умовах.

Порівняно з нашою системою:

Системи обробки акустичних даних зазвичай спеціалізовані на аналізі звукових сигналів і не мають такого широкого спектру застосувань, як наша система для визначення ідентифікаційних ознак об'єктів морського середовища. Наша система спрямована на розпізнавання та ідентифікацію об'єктів у морському середовищі за допомогою комплексу супутникових та інших даних, що робить її більш універсальною і потужною для вирішення різних завдань в цій галузі.

2.4 Програмне забезпечення сонарів

Програмне забезпечення сонарів – це спеціалізоване програмне забезпечення, призначене для обробки та аналізу даних, отриманих за допомогою сонарних систем. Сонари – це пристрої, що використовують звукові хвилі для визначення положення, відстані та інших характеристик об'єктів під водою. Сонар є ключовим інструментом у морських дослідженнях, навігації, підводному моніторингу та інших застосуваннях[1].

Основні функції програмного забезпечення сонарів:

- обробка сигналів: програмне забезпечення сонарів ефективно обробляє вхідні звукові сигнали, видаляючи шуми та визначаючи корисну інформацію, як-от відстані до об'єктів, їх розміри та форми[12];
- візуалізація даних: це програмне забезпечення забезпечує візуалізацію даних, отриманих від сонара, у вигляді графіків, карт та 3D-моделей, що дозволяє користувачам інтуїтивно розуміти підводні умови;
- геопросторова інтеграція: програмне забезпечення може інтегрувати дані від сонарів з іншими геопросторовими даними, наприклад, з даними GIS, для створення більш комплексних геопросторових моделей морського середовища;
- аналіз даних: підтримка складних аналітичних функцій, таких як знаходження об'єктів, відстеження їх переміщень, та оцінка змін у морському дні.

Інтерактивність та налаштування: програмне забезпечення часто дозволяє користувачам налаштовувати параметри сканування та аналізу, в залежності від конкретних потреб та умов місії.

Переваги програмного забезпечення сонарів.

- Спеціалізація: програмне забезпечення сонарів розроблене спеціально для потреб підводного дослідження, надаючи точніші та специфічні функції, ніж загальні геопросторові системи.

- Висока точність: завдяки передовим алгоритмам обробки сигналів, програмне забезпечення забезпечує високу точність у визначенні об'єктів і характеристик морського дна.

- Інтеграція з іншими системами: можливість інтеграції з іншими системами, такими як GIS, розширює можливості для комплексного аналізу морського середовища.

Недоліки та обмеження.

- Складність управління: висока спеціалізація та технічна складність можуть ускладнювати використання програмного забезпечення без спеціалізованого навчання.

- Обмежена функціональність: на відміну від більш універсальних геопросторових систем, програмне забезпечення сонарів може бути обмеженим у функціональності, зосереджуючись виключно на підводному дослідженні[13].

- Вартість: розробка та обслуговування спеціалізованого програмного забезпечення може бути дороговартісною, особливо для високотехнологічних систем.

Програмне забезпечення сонарів відіграє ключову роль у морських дослідженнях, навігації та моніторингу, дозволяючи виконувати специфічні завдання з високою точністю та ефективністю, хоча й має свої обмеження та вимагає спеціалізованого підходу до використання.

2.5 Програмне забезпечення лідарів

Програмне забезпечення лідарів (Light Detection and Ranging – виявлення світла і вимірювання відстані) є критично важливим компонентом у системах лідар, які використовуються для точного вимірювання відстаней до об'єктів за допомогою лазерних променів. Лідари широко застосовуються в геодезії, картографії, лісовому

господарстві, а також у сферах, пов'язаних з контролем навколишнього середовища.

Основні функції програмного забезпечення лідарів.

- Обробка даних: програмне забезпечення лідарів займається обробкою величезних обсягів даних, які генеруються лідарними сканерами, перетворюючи "сирі" лазерні відбиття в корисну інформацію.

- Візуалізація та моделювання: це програмне забезпечення забезпечує візуалізацію даних у формі 3D-моделей, карт висот, та інших візуальних форматів, що дозволяє легко інтерпретувати та аналізувати земельну поверхню і об'єкти на ній.

- Геопросторова інтеграція: програмне забезпечення може інтегрувати лідарні дані з іншими геопросторовими даними, включаючи фотографії з повітря чи космічні знімки, для створення більш детальних і точних геопросторових моделей.

- Класифікація та аналіз: програмне забезпечення дозволяє класифікувати дані лідару (наприклад, розрізняти рослинність, воду, будівлі), що є важливим для різноманітних досліджень та аналізів[14].

- Інтерактивність та налаштування: користувачі можуть налаштовувати параметри сканування, обробки та візуалізації, а також використовувати інтерактивні інструменти для кращого аналізу даних.

Переваги програмного забезпечення лідарів.

- Висока точність: лідарні технології надають високу точність вимірювань, що є критично важливим у багатьох застосуваннях.

- Детальне моделювання: програмне забезпечення дозволяє створювати детальні тривимірні моделі об'єктів і ландшафтів.

- Широкий діапазон застосувань: лідари використовуються у різних галузях, від лісового господарства до міського планування.

Недоліки та обмеження:

- Складність обробки даних: великі обсяги даних вимагають значних обчислювальних ресурсів та спеціалізованого програмного забезпечення.
- Висока вартість: лідарні технології і відповідне програмне забезпечення можуть бути досить дорогими, особливо для передових систем.
- Обмеження використання: лідари мають обмеження в умовах поганої видимості (наприклад, у тумані, диму), що може впливати на якість та точність даних.

Програмне забезпечення лідарів є невід'ємною частиною сучасних геопросторових досліджень та планування, надаючи користувачам потужні інструменти для точного аналізу та візуалізації фізичних характеристик землі та інших об'єктів.

2.6 Програмне забезпечення радарів

Програмне забезпечення радарів є важливим компонентом в системах радіолокації, які використовують радіохвилі для визначення положення, швидкості та інших характеристик об'єктів. Радари широко застосовуються в авіації, мореплавстві, метеорології, а також військових та цивільних областях для спостереження та контролю.

Основні функції програмного забезпечення радарів.

- Обробка сигналів: програмне забезпечення радарів обробляє вхідні радіосигнали, визначаючи відстані, швидкості та напрямки об'єктів на основі зміни частоти та інтенсивності отриманих сигналів[15].
- Візуалізація даних: програмне забезпечення забезпечує візуалізацію даних радару у вигляді графіків, карт та інших візуальних форматів, дозволяючи легко інтерпретувати результати спостереження.

- Інтеграція з іншими системами: програмне забезпечення може інтегрувати дані радару з іншими системами спостереження та контролю для створення комплексної картини ситуації.

- Автоматичне відстеження: розширені системи радарів можуть автоматично відстежувати рухомі об'єкти, надаючи важливу інформацію для навігації, контролю трафіку або безпеки.

- Аналіз даних: програмне забезпечення дозволяє проводити різноманітний аналіз даних, включаючи прогнозування погоди, визначення швидкості руху об'єктів та їх ідентифікацію.

Переваги програмного забезпечення радарів.

- Широкий діапазон застосувань: від авіації до метеорології, програмне забезпечення радарів дозволяє застосовувати радарні технології в різних областях.

- Висока точність: сучасні радарні системи забезпечують високу точність вимірювань, що є критично важливим для безпеки та навігації.

- Робота в усіх погодних умовах: радари ефективно працюють в різних погодних умовах, включаючи дощ, туман, і навіть у нічний час.

Недоліки та обмеження.

- Складність технологій: обробка радарних сигналів є складною і вимагає спеціалізованих знань та обладнання.

- Обмеження дальності: дальність радарів може бути обмеженою через атмосферні умови або характеристики обладнання.

- Висока вартість: розробка та обслуговування радарних систем може бути дороговартісною, особливо для передових систем.

Програмне забезпечення радарів відіграє ключову роль у численних областях, від покращення безпеки повітряного та морського руху до моніторингу погодних умов, забезпечуючи потужні інструменти для аналізу та візуалізації радіолокаційних даних.

2.7 Порівняння існуючих систем

В таблиці 2.1 представлено сумарне порівняння основних існуючих аналогів, їх переваги та недоліки.

Таблиця 2.1 – Порівняння основних переваг та недоліків між різними хмарними системами

Сервіс/Параметр	Переваги	Недоліки
1	2	3
GIS	- багатофункціональність і гнучкість - просторовий аналіз - інтеграція даних	- комплексність використання - обмеженість в аналізі даних - спеціалізація
Системи обробки супутникових зображень	- широкий спектр охоплення - висока частота оновлення - мультиспектральний аналіз	- обмежена деталізація - залежність від погодних умов - завузька спеціалізація
Системи обробки акустичних даних	- судноплавство та підводна активність - широкий спектр сервісів	- може бути дорожчим - комплексність інтерфейсу

Як бачимо, існуючі варіанти хмарних сервісів мають суттєві недоліки для потенційних кінцевих користувачів – інша спеціалізація, чіткість та точність результатів.

Тож можемо зробити висновок, що системи для визначення ідентифікаційних ознак об'єктів морського середовища стає обраною стратегією в тих випадках, коли важливий більший рівень контролю над інфраструктурою.

Це зокрема відноситься до ситуацій, коли необхідно забезпечити високу точність і специфічність обробки даних, а також забезпечити глибокий аналіз індивідуальних характеристик морських об'єктів. Системи, спеціально розроблені для ідентифікації ознак об'єктів морського середовища, можуть бути налаштовані для вирішення конкретних задач, враховуючи унікальні вимоги дослідження морської фауни, флори, геологічних структур, а також антропогенних об'єктів.

Висновки до розділу 2

У рамках другого розділу дисертаційної роботи було здійснено комплексний аналіз існуючих технологій та методів, призначених для дослідження морського середовища. Цей аналіз охопив широкий спектр засобів – від сучасних супутникових знімків, що надають широкоформатне зображення морської поверхні, до високоточних підводних лідарних систем, які забезпечують детальне картографування морського дна.

Супутникові зображення та радарні системи виявилися надзвичайно корисними для моніторингу морської поверхні та навігації кораблів, забезпечуючи важливу інформацію про рух морських течій, льодових полів та інших великомасштабних явищ. Акустичні теги та мітки, які використовуються для відстеження переміщень морських тварин, дозволяють збирати цінну інформацію про поведінку, міграції та взаємодії видів в природному середовищі. Водолазні

інспекції та магнітна розвідка відіграють ключову роль у безпосередньому огляді морського дна, виявленні металевих об'єктів, решток кораблів, та інших артефактів.

Підводні лідарні системи вносять значний вклад у створення деталізованих тривимірних моделей рельєфу морського дна, хоча їх ефективність залежить від умов прозорості води та обмежень, пов'язаних з глибиною проникнення лазерних променів.

Втім, незважаючи на велике розмаїття існуючих інструментів та методів, жоден із них не був спеціально розроблений чи оптимізований для комплексного визначення ідентифікаційних ознак об'єктів морського середовища. Це відкриває потребу в розробці нових, більш інтегрованих та багатофункціональних систем, які змогли б ефективно об'єднувати різні види даних та аналітичні підходи для глибшого розуміння морського середовища.

У цьому контексті, пропонована дисертаційна робота, спрямована на розробку системи для визначення ідентифікаційних ознак об'єктів морського середовища, має велике значення. Вона може стати унікальним та інноваційним внеском у дослідження морського середовища, відкриваючи нові можливості для більш точного моніторингу, глибшого аналізу та ефективного управління морськими ресурсами. Інтеграція різноманітних технологій та методик, включаючи сучасні комп'ютерні алгоритми та штучний інтелект, може значно розширити потенціал моніторингу та дослідження морських екосистем, забезпечуючи нові важливі дані для науки, екології та економіки.

3 ЗАСОБИ РОЗРОБКИ СИСТЕМИ ДЛЯ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ МОРСЬКОГО СЕРЕДОВИЩА

У процесі створення системи для ідентифікації об'єктів морського середовища ключовим етапом є вибір відповідних технологій та інструментів розробки. Це рішення впливає не тільки на ефективність та якість кінцевого продукту, але й на швидкість розробки та легкість підтримки системи у майбутньому.

На початковому етапі важливо вибрати мови програмування, які найкраще відповідають потребам проекту. Вибір часто падає на мови з великими можливостями для наукових розрахунків та обробки даних, такі як Python або R, а також на мови, що забезпечують високу продуктивність та стабільність для розробки серверної частини, наприклад, Java або C#.

Для ефективної розробки використовуються інтегровані середовища розробки (IDE), які надають широкий набір інструментів для написання, тестування та відлагодження коду. IDE, такі як PyCharm, IntelliJ IDEA або Visual Studio, забезпечують підтримку різних мов програмування, вбудовані інструменти для роботи з базами даних, інтеграцію з системами контролю версій та інші корисні можливості. Враховуючи необхідність роботи з великим обсягом даних та складними обчисленнями, велику увагу приділяється вибору архітектури системи. Використання контейнеризації, наприклад за допомогою Docker, дозволяє створювати легкі та ефективні середовища для розгортання додатків. Контейнери забезпечують високий рівень ізоляції, полегшують процес деплою та масштабування додатків.

Для організації високодоступної інфраструктури та управління контейнерами може бути використана оркестрація контейнерів, наприклад, за допомогою Kubernetes. Це не тільки спрощує розгортання та масштабування додатків у хмарному середовищі, але й забезпечує високу надійність та доступність

системи. Паралельно з цим, важливим аспектом є інтеграція системи з різноманітними інструментами моніторингу та логування, такими як Prometheus або ELK Stack, що дозволяє ефективно відстежувати стан системи, аналізувати ефективність роботи та швидко реагувати на можливі проблеми.

В сукупності, ці технології та підходи створюють міцний фундамент для розробки надійної, гнучкої та масштабованої системи, здатної вирішувати складні задачі аналізу та моніторингу морського середовища. Вони дозволяють зберігати фокус на інноваційності та ефективності, забезпечуючи високий рівень контролю над розробкою та розгортанням проекту.

3.1 Мова програмування Python

Вибір Python як основної мови програмування для розробки системи ідентифікації об'єктів морського середовища обумовлений рядом ключових факторів, які відрізняють цю мову від інших [16].

Python – це високорівнева, інтерпретована мова з динамічною типізацією, яка славиться своєю читабельністю, простотою синтаксису та широким спектром застосувань. Ця мова надає великі можливості для роботи з даними, наукових обчислень, розробки веб-додатків та автоматизації задач. На рисунку 3.1 зображено логотип мови програмування Python.



Рисунок 3.1 – Мова програмування Python

Переваги Python.

- Легкість читання та написання: Python має простий і зрозумілий синтаксис, що робить код легким для читання та написання. На рисунку 3.2 зображено приклад синтаксису мови Python, взятого з проекту цієї магістерської дисертації. Це знижує поріг входження для нових розробників і спрощує процес навчання.

```
from itertools import combinations
from clustererKmeans import KMeansClusterer
from clustererDBSCAN import DBSCANClusterer

if __name__ == "__main__":
    db_path = "../data/morsky_data.db"
    clusterer_DBSCAN = DBSCANClusterer(db_path)
    clusterer_Kmeans = KMeansClusterer(db_path)

    features = [
        "mfccs", "chroma", "contrast", "tonnetz", "rms_energy",
        "zero_cross_rate", "rolloff", "bandwidth", "flatness",
        "pitch", "melspectrogram"
    ]

    feature_combinations = combinations(features, 3)
    lim = 150
```

Рисунок 3.2 – Приклад синтаксису мови програмування Python

- Багата стандартна бібліотека та екосистема: Python має велику стандартну бібліотеку та широке коло сторонніх бібліотек для різноманітних задач – від обробки даних (NumPy, pandas) до машинного навчання (scikit-learn, TensorFlow) і візуалізації даних (Matplotlib, Seaborn).

- Спільнота та підтримка: широка спільнота розробників та користувачів Python забезпечує великий обсяг ресурсів для навчання, підтримки та співпраці.

У порівнянні, наприклад, з Java або C++, Python відрізняється меншою строгою типізацією та вищим рівнем абстракції. Це може бути недоліком у

контексті вимог до продуктивності та пам'яті, але водночас стає перевагою у швидкості розробки та гнучкості.

У порівнянні з JavaScript або PHP, які часто використовуються для розробки веб-додатків, Python вирізняється своєю універсальністю, що дозволяє йому бути ефективним як для веб-розробки, так і для складних наукових обчислень.

Недоліки Python.

- Продуктивність: Python є інтерпретованою мовою, вона може бути повільнішою за компільовані мови, такі як C++ або Java. Однак, ця проблема часто компенсується ефективністю розробки та можливістю використання оптимізованих бібліотек.

- Управління пам'яттю: у Python менше контролю над управлінням пам'яттю порівняно з низькорівневими мовами, що може бути критичним для деяких систем високої продуктивності.

У контексті розробки системи для визначення ідентифікаційних ознак об'єктів морського середовища, Python стає оптимальним вибором через кілька ключових чинників:

- Обробка та аналіз даних: Python є однією з провідних мов для наукових обчислень та обробки даних, що є фундаментальним для аналізу морських даних.

- Інтеграція з іншими системами та технологіями: Python легко інтегрується з різними базами даних, системами, API та інструментами машинного навчання, що є важливим для створення комплексної системи моніторингу.

- Гнучкість та швидкість розробки: швидкість розробки в Python, завдяки його високому рівню абстракції та багатому набору бібліотек, дозволяє швидко реалізувати складні алгоритми та адаптуватися до змінних вимог проекту.

Отже, враховуючи ці аспекти, Python є ідеальним вибором для розробки системи, орієнтованої на обробку та аналіз морських даних, забезпечуючи баланс між продуктивністю, гнучкістю та ефективністю розробки.

3.2 База даних SQLite

SQLite, обраний в якості бази даних для системи ідентифікації об'єктів морського середовища, є відмінним вибором, який має свої унікальні характеристики та переваги. Це легка, надійна та ефективна система управління базами даних, яка ідеально підходить для проектів, де потрібен простий і компактний механізм зберігання даних[17]. На рисунку 3.3 зображено логотип SQLite.



Рисунок 3.3 – Логотип SQLite

SQLite – це вбудована, файлова база даних, яка не вимагає окремого сервера баз даних чи складного налаштування. Вона забезпечує підтримку транзакцій, що дозволяє виконувати операції з базою даних надійно та безпечно. SQLite підтримує більшість стандартних функцій SQL, що робить її гнучкою у використанні.

Переваги SQLite.

- Простота та легкість: SQLite не вимагає окремого серверного середовища чи складного налаштування, що робить її ідеальною для легких або вбудованих додатків.
- Надійність: SQLite забезпечує атомарність транзакцій, що гарантує цілісність даних навіть у разі системних збоїв або відмови живлення.
- Переносимість: база даних у форматі SQLite – це один файл, який легко копіювати, переміщати або надсилати. Це робить її відмінною для застосунків, що вимагають високого рівня мобільності.

Порівняно з більш великими системами управління базами даних, такими як MySQL або PostgreSQL, SQLite не має потреби в окремому сервері або налаштуванні. Це робить її значно простішою для використання, але з іншого боку, може обмежувати її функціональність у складних або великомасштабних застосуваннях.

Недоліки SQLite.

- Обмежена масштабованість: SQLite менш підходить для великомасштабних додатків або тих, де одночасно працюють багато користувачів.

- Обмеження на паралельний доступ: SQLite може бути менш ефективною при високій кількості паралельних запитів на запис.

Для системи, спрямованої на ідентифікацію об'єктів морського середовища, SQLite пропонує простоту, надійність та ефективність. Її легкість і простота використання роблять її ідеальною для швидкої розробки та тестування, а можливість зберігати всю базу даних у одному файлі значно спрощує управління даними. Це дозволяє розробникам фокусуватися на логіці та функціональності системи, замість витрачати час на налаштування та управління складнішою базою даних.

Таким чином, вибір SQLite для даного проекту є логічним та виправданим, оскільки він забезпечує оптимальне поєднання простоти, ефективності та надійності, яке є критично важливим для успішної реалізації проекту.

3.3 Система контролю версій Git

GitHub, використаний у якості платформи для управління версіями та співпраці в рамках проекту з розробки системи ідентифікації об'єктів морського середовища, стає важливим інструментом, що надає численні переваги для розробників.

GitHub – це веб-платформа для управління версіями та співпраці, яка використовує Git – систему контролю версій, розроблену Лінусом Торвальдсом. GitHub дозволяє розробникам зберігати свої проекти (репозиторії), відстежувати зміни в коді, співпрацювати з іншими учасниками проекту, та керувати різними аспектами розробки програмного забезпечення.

Переваги GitHub.

- Співпраця та управління версіями: GitHub дозволяє легко співпрацювати над проектами, надаючи інструменти для управління внесками від різних розробників, відстеження змін та резолюції конфліктів.

- Відкритий доступ або приватні репозиторії: розробники можуть обирати, чи буде їх репозиторій відкритим для громадськості або закритим для обмеженої групи користувачів.

- Інтеграція з іншими інструментами: GitHub легко інтегрується з різними засобами та сервісами, що використовуються в процесі розробки, наприклад, із системами безперервної інтеграції/безперервного розгортання (CI/CD).

- Відстеження проблем (issues) та управління завданнями: GitHub надає інструменти для відстеження проблем, запитів на пул (pull requests), планування та управління задачами в рамках проекту.

Порівняно з іншими системами контролю версій, такими як Subversion або Mercurial, Git, а отже і GitHub, вирізняються своєю швидкістю, гнучкістю та підтримкою розподіленої роботи. GitHub став стандартом де-факто для розробників по всьому світу, забезпечуючи широкі можливості для співпраці та обміну кодом.

Недоліки GitHub.

- Крива навчання: для новачків система контролю версій Git та платформа GitHub можуть здатися складними у вивченні.

- Залежність від зовнішнього сервісу: використання зовнішнього сервісу може викликати питання щодо конфіденційності та контролю над даними, особливо в проектах з високими вимогами до безпеки.

У випадку розробки системи для ідентифікації об'єктів морського середовища GitHub виявляється ідеальним рішенням через свої можливості для співпраці, управління кодом та інтеграції з різними інструментами та сервісами. Він забезпечує централізовану платформу, де команда може працювати разом над кодом, обговорювати зміни, планувати релізи та відстежувати прогрес. Таким чином, GitHub вносить значний вклад у підвищення ефективності розробки та якості кінцевого продукту.

3.4 Бібліотеки та середовище розробки

В цьому проекті використовувались багато різноманітних Python – бібліотек, таких як librosa – для обробки аудіосигналів, Numpy – для проведення математичних обчислень, Sklearn – для проведення кластеризації, Matplotlib – для візуалізації отриманих результатів, Loguru – для зручного логування коду.

3.4.1 Бібліотека для роботи з математичними обчисленнями Numpy

Numpy, яка є однією з основних бібліотек в екосистемі Python, відіграє важливу роль у розробці системи для визначення ідентифікаційних ознак об'єктів морського середовища. На рисунку 3.4 зображено логотип цієї бібліотеки. Ця бібліотека стала ключовим інструментом, особливо з огляду на її спроможність до ефективної обробки великих масивів даних та наукових обчислень.



Рисунок 3.4 – Логотип NumPy бібліотеки

NumPy надає підтримку векторних та матричних операцій, що є основою для більшості типів наукових обчислень. У контексті цього проекту, це стало особливо важливим для обробки та аналізу даних зібраних з морського середовища. Дані, отримані з різних джерел, часто представляють собою великі обсяги числових масивів, які потребують ефективної обробки та аналітичного аналізу.

Завдяки NumPy, була змога ефективно обробляти ці великі набори даних, здійснюючи такі операції, як фільтрація, сортування, статистичний аналіз, та інші математичні обчислення, необхідні для виявлення та класифікації об'єктів у морському середовищі. Більш того, NumPy дозволяє інтегрувати дані з інших джерел та форматів, спрощуючи процес агрегації та аналізу різноманітної інформації.

Однією з вирішальних переваг використання NumPy є її висока продуктивність. Оскільки бібліотека оптимізована для виконання операцій з великими масивами даних, вона значно перевершує стандартні Python-структури даних, як-от списки, з точки зору швидкості обчислень. Це має велике значення в контексті цього проекту, де швидкість обробки даних має критичне значення. Крім того, NumPy легко інтегрується з іншими бібліотеками для наукових обчислень та машинного навчання, такими як SciPy та scikit-learn, що робить її незамінною частиною в більш широкому процесі розробки. Ця інтеграція дозволяє нам використовувати складні алгоритми та статистичні методи для більш глибокого аналізу даних.

У підсумку, використання Numpy у цьому проєкті не тільки полегшило процес обробки та аналізу даних, але й забезпечило необхідну швидкість та ефективність, що є важливим для досягнення наших цілей у вивченні та моніторингу морського середовища.

3.4.2 Бібліотека для логування Loguru

Loguru, використана в цьому проєкті як основний інструмент для логування, є важливою частиною системи, що дозволяє нам ефективно відстежувати, аналізувати та управляти процесами виконання коду. Ця бібліотека Python стала вибором завдяки своїй простоті, гнучкості та потужним функціональним можливостям.

В проєктах програмного забезпечення, особливо в таких, що включають складну обробку даних та взаємодію з різноманітними компонентами, належне логування є невід'ємною частиною. Loguru виконує цю функцію, надаючи можливість детального та налаштованого відстеження подій, які відбуваються у процесі роботи програми. Вона дозволяє виявляти та аналізувати помилки, моніторити виконання операцій та забезпечувати важливу інформацію для налагодження та оптимізації системи.

На відміну від стандартного модуля логування Python, Loguru пропонує спрощений та більш інтуїтивний інтерфейс. Вона вимагає менше коду для впровадження і забезпечує більш гнучкі можливості налаштування. З Loguru, розробники мають можливість легко налаштовувати формати виводу логів, рівні важливості повідомлень та місце їх збереження (наприклад, у файлі або у віддаленій системі).

Loguru інтегрується з системою без зайвих зусиль, надаючи широкий спектр опцій для кастомізації. Це особливо корисно у проєктах, де важлива детальна

інформація про виконання кожної частини системи. У цьому випадку, це допомогло створити прозорий та ефективний механізм логування, що сприяє більш швидкому виявленню та виправленню помилок, а також дає змогу глибше аналізувати поведінку системи під час обробки морських даних.

Завдяки Loguru, було отримано потужний та гнучкий інструмент для логування, що значно покращує процес розробки та підтримки системи. Її здатність до гнучкої налаштування, легкість інтеграції та простота використання роблять Loguru незамінною частиною в архітектурі цього проекту.

3.4.3 Інтегроване середовище розробки PyCharm

PyCharm, обраний як основне інтегроване середовище розробки (IDE) для цього проекту ідентифікації об'єктів морського середовища, є вагомим внеском у процес розробки, забезпечуючи гнучкість, продуктивність та зручність. На рисунку 3.5 зображено логотип IDE PyCharm.



Рисунок 3.5 – PyCharm логотип

У розробці складних систем, таких як цей проект, важливо мати надійне, функціональне і ефективне середовище розробки. PyCharm від компанії JetBrains виконує цю роль, пропонуючи широкий спектр функціональних можливостей,

призначених для розробників Python. Він спрощує процес написання коду, його налагодження та тестування, а також надає потужні інструменти для аналізу коду та його оптимізації.

PyCharm пропонує вбудовані інструменти для розробки, включаючи підтримку різних фреймворків, систем контролю версій, інструментів для роботи з базами даних, інтерпретаторів Python, а також інтеграцію з іншими сервісами і інструментами. Середовище PyCharm особливо відзначається своєю здатністю до глибокого аналізу коду, що допомагає виявляти потенційні помилки та неефективні конструкції до їх виконання.

Однією з ключових переваг PyCharm є його інтеграція з різними інструментами та сервісами, що використовуються в процесі розробки. Наприклад, інтеграція з Git дозволяє легко управляти версіями коду безпосередньо з IDE. Також PyCharm підтримує роботу з базами даних, включаючи SQL, що є важливим для проектів, де дані грають ключову роль.

Вибір PyCharm як основного IDE для цього проекту був зумовлений його високою продуктивністю, багатофункціональністю та налаштовуваністю. Це середовище не тільки сприяло підвищенню ефективності розробки, але й покращило якість кінцевого продукту завдяки своїм розширеним можливостям для аналізу та оптимізації коду. PyCharm відіграє ключову роль у цьому проекті, забезпечуючи зручний, інтуїтивно зрозумілий та потужний інструмент для розробки складних систем.

3.4.4 Бібліотека Scikit-learn

Scikit-learn (скорочено sklearn) є однією з ключових бібліотек у світі машинного навчання та аналізу даних, особливо в контексті Python. Її використання у цьому проекті для кластеризації об'єктів морського середовища за

допомогою алгоритмів KMeans та DBSCAN є важливим кроком для досягнення високої точності та ефективності аналізу. На рисунку 3.6 зображено логотип бібліотеки Scikit-learn.



Рисунок 3.6 – Логотип бібліотеки Scikit-learn

Наш проект, спрямований на ідентифікацію та аналіз об'єктів у морському середовищі, вимагає застосування складних методів машинного навчання для обробки та інтерпретації даних. Sklearn забезпечує необхідні інструменти для ефективної реалізації цих методів. Алгоритми кластеризації, такі як KMeans та DBSCAN, які ви використовуєте, дозволяють групувати дані на основі їхніх характеристик, що є ключовим для розуміння та аналізу шаблонів у морському середовищі.

Sklearn є високопродуктивною бібліотекою, яка надає простий, але потужний інтерфейс для реалізації алгоритмів машинного навчання. Вона включає широкий спектр алгоритмів, від класифікації та регресії до кластеризації та зниження розмірності. Бібліотека добре документована та має активну спільноту користувачів, що спрощує процес навчання та вирішення проблем.

Використання Sklearn у цьому проекті дозволяє нам використовувати перевірені та ефективні алгоритми без потреби в їх ручній реалізації. Кластеризація за допомогою KMeans є корисною для ідентифікації груп або кластерів схожих об'єктів на основі заданих характеристик. DBSCAN, з іншого боку, є потужним алгоритмом для виявлення груп або шаблонів в даних з урахуванням їх щільності,

що може бути особливо корисним у складних морських середовищах, де шаблони розподілу можуть бути нерівномірними.

Обрав Sklearn для цього проекту, було отримано доступ до потужного набору інструментів, які оптимізують процес машинного навчання, роблячи його більш доступним, ефективним і точним. Це відкриває можливості для глибшого розуміння та аналізу даних, зібраних з морського середовища, та сприяє розвитку ефективних методів для ідентифікації та класифікації морських об'єктів.

3.4.5 Бібліотека librosa

Librosa є центральним інструментом у цьому проекті, особливо коли мова йде про обробку та аналіз аудіоданих. Ця бібліотека Python є однією з провідних у галузі аудіоаналізу та музичної інформатики, пропонуючи широкий спектр інструментів для екстракції особливостей з аудіо, що є важливим для цього проекту з ідентифікації та аналізу морського середовища. На рисунку 3.7 зображено логотип бібліотеки librosa.



Рисунок 3.7 – Логотип бібліотеки librosa

Обробка та аналіз аудіосигналів відіграють ключову роль у зборі та інтерпретації даних з морського середовища. Librosa дозволяє нам ефективно перетворювати ці аудіосигнали в корисну інформацію, що може бути використана для виявлення та класифікації різних морських об'єктів та явищ. Чи то аналіз звуків

морських тварин, чи оцінка звукових забруднень, Librosa забезпечує необхідні інструменти для розширеного аудіоаналізу [3].

Librosa спеціалізується на аналізі музики та аудіо, зокрема на таких задачах, як екстракція особливостей (feature extraction), класифікація аудіо, музична сегментація та ін. Вона надає широкий набір функцій для обробки аудіо, наприклад, для визначення темпу, визначення висоти звуку (pitch), екстракції мел-спектрограм та інших. Librosa легко інтегрується з іншими бібліотеками Python, такими як NumPy та SciPy, що розширює можливості для аналізу та обробки даних. Вона стає особливо корисною у поєднанні з машинним навчанням та іншими методами аналізу даних, дозволяючи нам глибше зануритися в аналіз морських аудіоданих.

Використання Librosa в цьому проекті відкриває нові горизонти для розуміння морського середовища. Завдяки її здатності точно аналізувати та обробляти складні аудіосигнали, є можливість ідентифікувати та класифікувати різні аспекти морського середовища, що раніше були недоступні для дослідження. Це не тільки покращує наше розуміння цього унікального середовища, але й сприяє розвитку нових підходів та методів у морських дослідженнях.

3.4.6 Бібліотека Matplotlib

Matplotlib, як одна з фундаментальних бібліотек Python для візуалізації даних, відіграє вирішальну роль у вашому проекті, який зосереджений на ідентифікації та аналізі об'єктів морського середовища. Ця бібліотека є надзвичайно потужним інструментом для створення широкого спектру графіків та візуальних представлень даних, що є ключовим для аналізу та інтерпретації складних датасетів. На рисунку 3.8 зображено логотип бібліотеки matplotlib.



Рисунок 3.8 – Логотип matplotlib

Наш проект, який включає складні завдання аналізу даних, вимагає чіткої та зрозумілої візуалізації результатів. Matplotlib дозволяє реалізувати це завдяки своїм функціям для створення як простих, так і складних графіків. Ви можете легко створювати лінійні графіки, гістограми, розсіяні діаграми, кругові діаграми та багато інших типів візуалізацій, що дозволяють зображати дані у зрозумілому та доступному форматі.

Matplotlib також інтегрується з іншими бібліотеками Python для наукових обчислень, такими як NumPy та Pandas, забезпечуючи безпроблемну інтеграцію з вашими аналітичними потоками роботи. Це особливо корисно у вашому проекті, де ви, можливо, аналізуєте великі обсяги морських даних та потребуєте зручних інструментів для їх візуалізації.

Додатково, Matplotlib дозволяє кастомізувати графіки до деталей, від регулювання кольорів та шрифтів до зміни стилів та макетів. Це дозволяє вам створювати візуалізації, які не тільки інформативні, але й естетично привабливі, що може бути важливим для презентації результатів досліджень.

Використання Matplotlib у цьому проекті відкриває широкі можливості для глибшого розуміння морських даних. Чи це візуалізація складних патернів у розподілі морських видів, чи демонстрація впливу забруднення на морські екосистеми, Matplotlib надає необхідні інструменти для чіткого та докладного візуального представлення цих складних даних.

3.5 Методи

В даному проєкті використовуються такі методи як Feature Extraction, та методи кластеризації DBSCAN та K-means.

3.5.1 Feature Extraction

Feature Extraction, або екстракція особливостей, є одним з фундаментальних процесів у цьому проєкті по визначенню ідентифікаційних ознак об'єктів морського середовища. Він сприяє перетворенню сирової інформації, яка отримується з різноманітних джерел, на значущі дані, які можуть бути ефективно аналізовані та інтерпретовані[7].

У контексті цього проєкту, Feature Extraction допомагає вилучити з аудіосигналів, зібраних з морського середовища, такі характеристики, як інтенсивність звуку, частота, тон і інші важливі аспекти. [8] Це особливо важливо, оскільки морське середовище є джерелом складних і різноманітних звукових сигналів, які можуть містити цінну інформацію про різноманітні біологічні та фізичні процеси.

Процес екстракції особливостей зводиться не лише до технічного аналізу звуків, але й до глибшого розуміння того, як різні звуки та їх характеристики пов'язані з конкретними подіями або об'єктами у морі. [9] Наприклад, визначення особливостей звуків, які видають різні морські тварини, дозволяє нам не тільки ідентифікувати ці види, але й зрозуміти їх поведінку та взаємодії в морському середовищі.

Ефективність Feature Extraction у цьому проєкті значною мірою залежить від точності та чутливості використовуваних алгоритмів та методів. Використання передових алгоритмів дозволяє нам максимально точно визначити необхідні характеристики зі складних та часто зашумлених аудіоданих.

3.5.2 Метод кластеризації DBSCAN

DBSCAN, що розшифровується як "Density-Based Spatial Clustering of Applications with Noise", є потужним методом кластеризації, який грає важливу роль у цьому проекті, орієнтованому на вивчення морського середовища. Цей метод стоїть відокремлено від традиційних підходів кластеризації завдяки своїй здатності виявляти кластери різної форми та розміру без необхідності вказувати кількість кластерів наперед.

У контексті цього проекту, що зосереджений на аналізі морського середовища, DBSCAN використовується для ідентифікації структурних шаблонів у даних. Морське середовище характеризується великою різноманітністю та складністю, і наша задача полягає у виявленні цих властивостей на основі зібраних даних. DBSCAN дозволяє нам групувати ці дані на основі їх "щільності", тобто частоти зустрічі певних характеристик або явищ у просторі.

DBSCAN працює за принципом ідентифікації "щільних" регіонів даних. Якщо визначена кількість точок знаходиться в межах певної радіус-дистанції, ці точки вважаються частиною одного кластера[5]. Це особливо корисно для виявлення природних групувань у даних, які можуть не бути очевидними за допомогою більш традиційних методів кластеризації.

Переваги DBSCAN.

- Виявлення кластерів різних форм: на відміну від інших методів кластеризації, які припускають круглу форму кластерів (наприклад, K-means), DBSCAN ефективно виявляє кластери різних, часто неправильних форм, що є важливим для аналізу складних та нерегулярних даних морського середовища.

- Обробка шуму та відсторонених точок: DBSCAN ефективно відокремлює основні компоненти кластерів від шуму (outliers), що дозволяє отримувати більш чисті та точні результати кластеризації.

- Мінімальна потреба в попередньому визначенні кількості кластерів: на відміну від методів, які вимагають заздалегідь заданої кількості кластерів, DBSCAN адаптивно визначає кластери на основі щільності даних, що забезпечує гнучкість та адаптивність алгоритму.

Причини вибору DBSCAN.

- Складність та різноманітність даних: дані, з якими користувачі працюють у морському середовищі, часто мають складну структуру та включають як щільні кластери, так і віддалені точки. DBSCAN ідеально підходить для роботи з такими даними, оскільки він здатний ефективно розділяти кластери та визначати шум.

- Адаптивність та гнучкість: морське середовище є динамічним та різноманітним, і тому потрібен алгоритм, який може адаптуватися до цих особливостей без необхідності в ручному налаштуванні для кожного окремого випадку. DBSCAN пропонує саме таку гнучкість.

- Ефективність в виявленні шаблонів та аномалій: визначення та аналіз аномалій є ключовим у цьому проекті, і DBSCAN ефективно справляється з цим завданням, відокремлюючи основні структури даних від аномальних точок.

Загалом, використання DBSCAN у цьому проекті дозволяє нам глибше аналізувати та розуміти складні датасети морського середовища, виявляючи значимі шаблони та структури, які можуть бути важливими для подальших досліджень та прийняття рішень.

3.5.3 Метод кластеризації K-means

Алгоритм K-means – це революційний підхід у сфері машинного навчання, який перетворює складні набори даних на простіші та зрозуміліші структури. Цей алгоритм поділяє датасет на певну кількість кластерів, яка позначається як "K". Ці кластери формуються так, щоб кожна точка даних була призначена до кластера з

найменшою відстанню до центроїда - середньої точки цього кластера. Центроїди визначаються та переоцінюються на кожному кроці алгоритму, щоб забезпечити оптимальне групування.

Унікальні аспекти та особливості K-means.

1. Простота у складності: одна з ключових особливостей K-means полягає в його здатності перетворювати складні, багатовимірні дані на прості, інтуїтивно зрозумілі групи, роблячи його ідеальним для розуміння та аналізу великих обсягів інформації.

2. Динаміка і адаптивність: K-means відрізняється своєю динамічністю, оскільки він може адаптуватися до різних типів даних та задач, будучи універсальним інструментом для аналізу даних.

Даний проєкт зосереджений на ідентифікації ознак об'єктів морського середовища, і тут алгоритм K-means відіграє ключову роль. Він дозволяє.

1. Ефективне розділення морських об'єктів: за допомогою K-means можна класифікувати морські об'єкти на основі їх фізичних, хімічних або біологічних характеристик, тим самим відкриваючи нові перспективи для дослідження морського біорізноманіття.

2. Глибше розуміння даних: цей метод може допомогти у виявленні складних взаємозв'язків та закономірностей у морському середовищі, які не були б очевидними за інших умов.

3. Оптимізація даних для подальших досліджень: K-means може також слугувати як потужний інструмент для попередньої обробки даних, готуючи ґрунт для застосування більш складних аналітичних методик.

Наш проєкт, який інтегрує K-means, не лише підвищує точність та ефективність ідентифікації морських об'єктів, але й відкриває нові можливості для розширення знань у цій важливій науковій галузі.

Висновки до розділу 3

У цьому розділі було розглянуто ключові інструменти та технології, використані в ході розробки нашої системи для визначення ідентифікаційних ознак об'єктів морського середовища. Детально аналізуючи кожен інструмент, було визначено наступні основні компоненти розробки.

- Python: ця мова програмування була обрана як основний інструмент розробки через її універсальність, чіткий синтаксис та широкий спектр доступних бібліотек та фреймворків. Python був використаний для реалізації всіх аспектів системи, включаючи обробку вхідних даних, кластеризацію сигналів та виведення результатів. Це дало нам можливість створити гнучку та ефективну систему.

- GitHub: було обрано GitHub як платформу для зберігання коду та управління версіями. Він надав надійне хмарне сховище для цього проекту, а також забезпечив легкий доступ до історії змін та сприяв спільній розробці.

- SQLite: як базу даних для зберігання інформації про оброблені сигнали обрано SQLite. Ця легка та надійна СУБД з підтримкою SQL забезпечила ефективне та надійне зберігання даних.

- K-Means і DBSCAN: було використано алгоритми K-Means та DBSCAN для кластеризації сигналів. Ці алгоритми дозволили нам ефективно групувати сигнали на основі їх властивостей та щільності в просторі даних, забезпечуючи точний аналіз.

- Бібліотека Librosa: ця бібліотека стала ключовим інструментом для аналізу аудіоданих. Вона надала нам можливість ефективно обробляти аудіосигнали та екстрагувати важливі характеристики, необхідні для цього дослідження.

- Бібліотека Matplotlib: завдяки цій бібліотеці мали можливість створювати чіткі та інформативні візуалізації наших даних. Вона дозволила нам ефективно демонструвати результати аналізу та виявлених шаблонів.

- Бібліотека Scikit-learn: наша система використовувала цю бібліотеку для реалізації алгоритмів кластеризації. Вона забезпечила доступ до широкого спектру методів кластеризації, класифікації та зниження розмірності.

- Бібліотека Loguru: було інтегровано Loguru для зручного та ефективного журналювання процесу розробки. Це дозволило легко відстежувати хід роботи системи та виявляти можливі проблеми.

- IDE PyCharm: для розробки було використовано інтегроване середовище розробки PyCharm, яке надало зручні інструменти для написання, тестування та відлагодження коду.

- Feature Extraction: було впровадили методики екстракції ознак для аналізу сигналів. Це включало в себе техніки для виділення ключових характеристик з даних, що дозволило нам точніше ідентифікувати та аналізувати об'єкти морського середовища.

Користуючись цими технологіями, було розроблено ефективну та надійну систему для визначення ідентифікаційних ознак об'єктів у морському середовищі. Ці інструменти не тільки спростили процес розробки, але й значно підвищили якість та точність аналізу, роблячи цей проєкт цінним інструментом для моніторингу та дослідження морського середовища.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

У цьому розділі детально розглядаються ключові аспекти інформаційної системи, розробленої в рамках цього дипломного проекту. Детальний опис програмної реалізації є критично важливим, оскільки він надає глибоке розуміння того, як була втілена первісна концепція додатку, які архітектурні діаграми були створені для забезпечення цілісного розуміння системи, і як саме функціональність була реалізована на рівні програмного коду.

Цей розділ має на меті систематизувати та деталізувати всі аспекти програмної реалізації проекту. Детальний опис дозволяє чітко уявити, який вигляд має розроблена система зсередини, як вона взаємодіє з різними компонентами, та як вона відповідає поставленим вимогам.

Особлива увага приділяється аналізу ключових компонентів системи, розгляду архітектурних рішень, що були прийняті під час розробки, та оцінці того, як ці рішення впливають на загальну ефективність та надійність системи. Такий підхід дозволяє не тільки оцінити технічну сторону проекту, але й зрозуміти логіку вибору конкретних технологій та методологій розробки.

Завдяки цьому розділу, читач зможе отримати всебічне розуміння структури проекту, його основних компонентів та інтеграції між ними, а також того, як розроблена система відповідає поставленим цілям та вирішує ключові завдання.

4.1 Архітектурне моделювання діаграм системи

Перед початком розробки програмного забезпечення зазвичай проводиться ретельне моделювання різних аспектів додатку за допомогою діаграм, які демонструють не тільки загальну структуру системи, але й деталізовані моделі бази даних, архітектуру фронтенду та бекенду, а також взаємодію користувачів з

програмою. Однією з найбільш ефективних методик для цього є використання Уніфікованої мови моделювання (UML), яка є стандартом у галузі проектування програмного забезпечення.

UML є комплексною графічною системою, призначеною для єдиної візуалізації, специфікації, конструювання та документування програмних систем. Вона особливо ефективна для систем, що розробляються на основі об'єктно-орієнтованого підходу, дозволяючи детально відобразити всі компоненти та їх взаємозв'язки.

У даному проєкті для наочного представлення різних процесів було створено три основні типи діаграм поведінки. Перша - діаграма прецедентів для загального додатку - демонструє можливі сценарії взаємодії користувачів з системою на високому рівні. Друга - діаграма прецедентів для власної частини - фокусується на специфічних випадках використання, що стосуються конкретної частини додатку. Нарешті, діаграма послідовностей дає змогу зрозуміти взаємодію об'єктів системи у часі, забезпечуючи детальне розуміння процесів обміну повідомленнями між компонентами системи.

Ці діаграми не лише допомагають у візуалізації та розумінні архітектури системи, але й служать важливим інструментом для комунікації між членами команди розробників, а також зі стейкхолдерами проєкту. Вони забезпечують чітке уявлення про функціональні можливості та обмеження системи, а також допомагають у плануванні розробки та тестування.

4.1.1 Діаграми прецедентів

Моделювання діаграми прецедентів можливих дій користувача (Use Case) є важливим етапом у розробці системи, тому що така дозволяє звернути увагу саме

на схему очікуваних взаємодій та можливостей в розроблюваній системі, як зображено на рисунку 4.1.

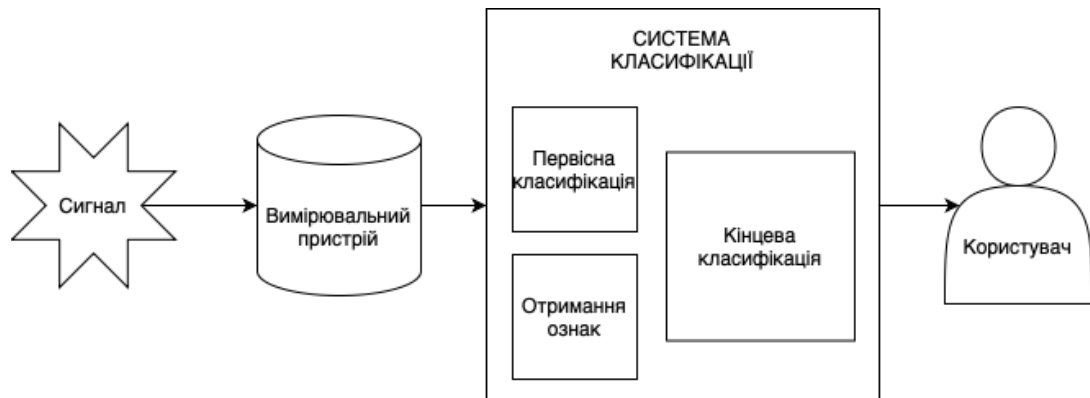


Рисунок 4.1 — Діаграма прецедентів розроблюваної системи

Завершена версія проекту представляє собою комплексну систему, що забезпечує ряд ключових функціональних можливостей, критично важливих для аналізу та обробки сигналів у морському середовищі. Наступні значущі функції були реалізовані в додатку:

- Обробка вхідного сигналу: система здатна ефективно приймати та обробляти вхідні сигнали з різних джерел. Ця функція включає передпроцесинг сигналу, такий як фільтрація шумів, нормалізація та інші необхідні трансформації для підготовки сигналу до подальшого аналізу.

- Знаходження ознак сигналу (Feature Extraction): система використовує передові методи для виявлення та екстракції ключових характеристик з сигналів. Цей процес включає аналіз спектральних, часових та інших характеристик, що дозволяє точно ідентифікувати та класифікувати різні типи сигналів.

- Додавання нового сигналу в базу даних: додаток дозволяє легко інтегрувати нові сигнали в систему, зберігаючи їх у базі даних SQLite. Це гарантує легкий доступ до сигналів для подальшого аналізу та дозволяє зберігати історичні дані для майбутніх порівняльних досліджень.

- Кластеризація сигналу: система включає функції кластеризації сигналів, які дозволяють групувати сигнали на основі їх характеристик. Це важливо для

визначення шаблонів та трендів у морських сигналах, що може бути корисним для наукових досліджень та моніторингу середовища.

- Вивід результату кластеризації: після кластеризації сигналів, додаток надає чіткі та інтуїтивно зрозумілі візуалізації результатів. Це дозволяє користувачам легко інтерпретувати результати аналізу та використовувати їх для подальших досліджень або прийняття рішень.

На момент завершення роботи над проектом, усі ці функції були успішно реалізовані та інтегровані в одну цілісну систему. Використання бази даних SQLite забезпечує надійне та ефективне зберігання даних про сигнали, їх обробку та результати кластеризації. Це дозволяє системі зберігати великий обсяг інформації та забезпечує швидкий доступ до даних для аналізу та звітності.

4.2 Структура програмного забезпечення

Чітку ієрархію структури з програмним кодом можна бачити на рисунку 4.2.

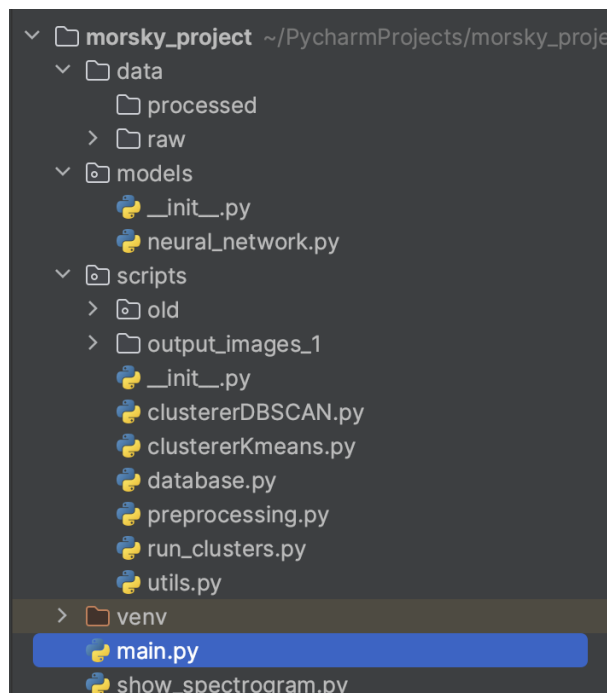


Рисунок 4.2 — Ієрархія системи

В структурі даного проекту кожна директорія відіграє певну роль у загальній архітектурі системи, що дозволяє ефективно обробляти та аналізувати аудіосигнали з морського середовища.

1. Директорія 'data': вона є центральним репозиторієм для всіх даних, пов'язаних з проектом. Піддиректорія 'raw' зберігає немодифіковані вхідні аудіофайли, які є вихідною точкою для системи обробки. Після того, як вхідні дані проходять через процес попередньої обробки, вони переносяться до піддиректорії 'processed', де зберігаються вже трансформовані файли. Ця трансформація може включати фільтрацію шумів, нормалізацію амплітуди та інші види обробки, що підготовляють сигнали до подальшої екстракції ознак та кластеризації.

2. Директорія 'models': тут зберігаються всі моделі машинного навчання та штучного інтелекту, які використовуються для кластеризації та класифікації аудіосигналів. Моделі можуть бути навчені для розпізнавання складних шаблонів в даних, що дозволяє системі порівнювати нові сигнали з існуючими даними для точнішого визначення їхніх ідентифікаційних ознак.

3. Директорія 'scripts': являє собою ядро програмної частини проекту, де розміщені ключові скрипти для реалізації різних функцій системи. Файл "preprocessing.py" відповідає за попередню обробку та фільтрацію аудіоданих, підготовляючи їх для процесу виявлення ознак. "database.py" містить функції для взаємодії з базою даних, дозволяючи здійснювати запити, оновлення та інші операції з даними. "utils.py" включає в себе набір допоміжних функцій, які можуть бути використані різними частинами програми для забезпечення відновлення та оптимізації коду. "clustererDBSCAN.py" та "clustererKmeans.py" містять логіку для кластеризації даних, що допомагає у групуванні аудіосигналів на основі їхніх характеристик та зіставлення з існуючими категоріями.

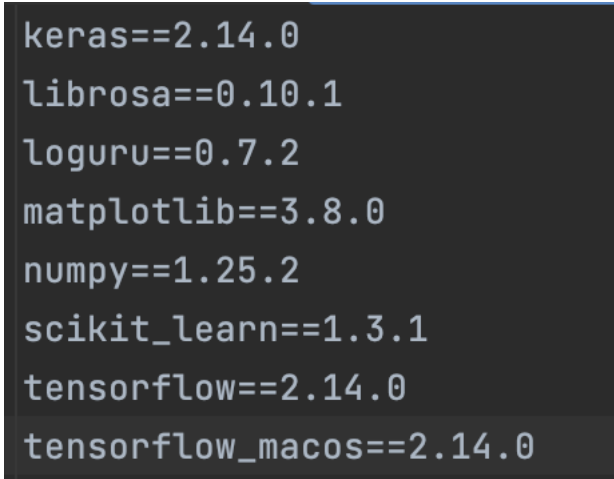
4. Головна директорія: основний виконуваний файл "main.py" є входом до системи, який ініціює та координує виконання всіх процесів. "show_spectrogram.py" використовується для генерації та відображення

спектрограм, які є візуальним зображенням спектру частот аудіосигналів, що дозволяє аналітикам бачити різні аспекти даних, які можуть бути неочевидними при слуховому аналізі.

4.3 Робота користувача з програмною системою

Взаємодія користувача з програмною системою є ключовим елементом, що визначає успішність та ефективність використання розробленого продукту. Цей розділ присвячений аналізу та опису процесів, які користувач повинен виконати для роботи з програмним забезпеченням, створеним для аналізу об'єктів морського середовища. Мета цього вступу – висвітлити основні аспекти інтеракції користувачів з системою, її інтуїтивність та функціональні можливості.

Перш за все, для забезпечення безперебійної роботи системи, користувачу потрібно встановити набір специфічних бібліотек, які є фундаментальними для функціонування додатку (рис. 4.3).



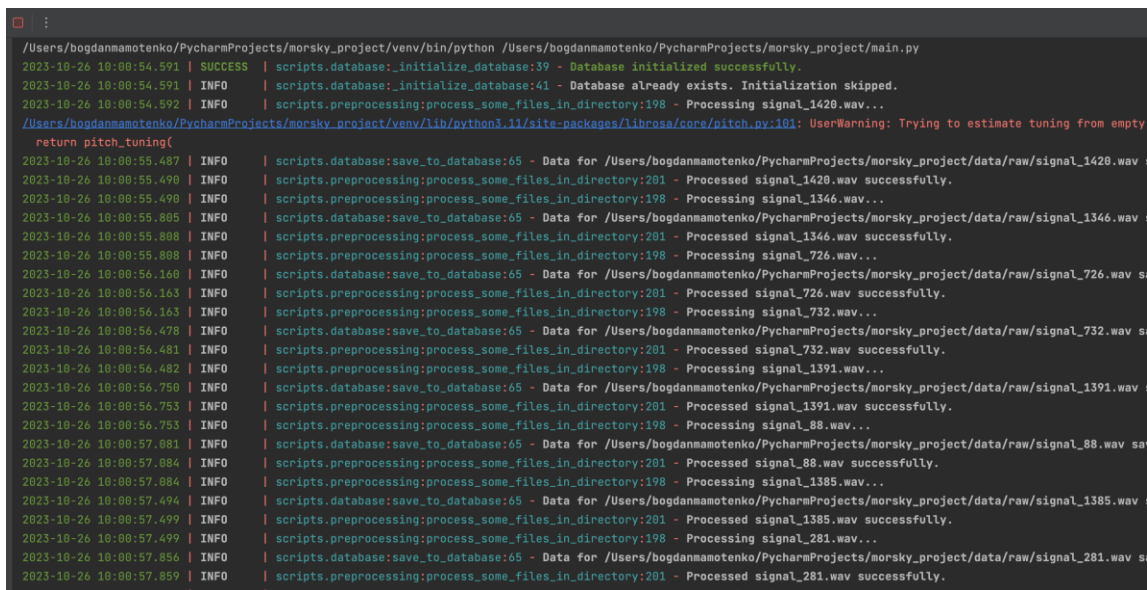
```
keras==2.14.0
librosa==0.10.1
loguru==0.7.2
matplotlib==3.8.0
numpy==1.25.2
scikit_learn==1.3.1
tensorflow==2.14.0
tensorflow_macos==2.14.0
```

Рисунок 4.3 — Список встановлених бібліотек на машині

На рисунку 4.3 можна побачити список потрібних бібліотек. Ці бібліотеки включають інструменти для обробки даних, математичних обчислень, візуалізації, взаємодії з базами даних та інші необхідні залежності. Встановлення цих бібліотек

є першим кроком до ініціації проекту і забезпечує підґрунтя для всіх наступних операцій.

Далі, користувачу необхідно сформувати базу даних із сигналами, які будуть використані для аналізу. Цей етап включає збір та попередню обробку вхідних аудіоданих, їхнє структурування та підготовку до процесу кластеризації. На рисунку 4.4 зображений процес попереднього аналізу даних. Попередня обробка може включати такі дії, як фільтрація шуму, нормалізація, вирівнювання гучності та інші техніки обробки сигналів, які сприяють поліпшенню якості даних для подальшої кластеризації.



```
/Users/bogdanmamotenko/PycharmProjects/morsky_project/venv/bin/python /Users/bogdanmamotenko/PycharmProjects/morsky_project/main.py
2023-10-26 10:00:54.591 | SUCCESS | scripts.database:initialize_database:39 - Database initialized successfully.
2023-10-26 10:00:54.591 | INFO | scripts.database:initialize_database:41 - Database already exists. Initialization skipped.
2023-10-26 10:00:54.592 | INFO | scripts.preprocessing:process_some_files_in_directory:198 - Processing signal_1420.wav...
/Users/bogdanmamotenko/PycharmProjects/morsky_project/venv/lib/python3.11/site-packages/librosa/core/pitch.py:101: UserWarning: Trying to estimate tuning from empty
return pitch_tuning(
2023-10-26 10:00:55.487 | INFO | scripts.database:save_to_database:65 - Data for /Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw/signal_1420.wav s
2023-10-26 10:00:55.490 | INFO | scripts.preprocessing:process_some_files_in_directory:201 - Processed signal_1420.wav successfully.
2023-10-26 10:00:55.490 | INFO | scripts.preprocessing:process_some_files_in_directory:198 - Processing signal_1346.wav...
2023-10-26 10:00:55.805 | INFO | scripts.database:save_to_database:65 - Data for /Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw/signal_1346.wav s
2023-10-26 10:00:55.808 | INFO | scripts.preprocessing:process_some_files_in_directory:201 - Processed signal_1346.wav successfully.
2023-10-26 10:00:55.808 | INFO | scripts.preprocessing:process_some_files_in_directory:198 - Processing signal_726.wav...
2023-10-26 10:00:56.160 | INFO | scripts.database:save_to_database:65 - Data for /Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw/signal_726.wav s
2023-10-26 10:00:56.163 | INFO | scripts.preprocessing:process_some_files_in_directory:201 - Processed signal_726.wav successfully.
2023-10-26 10:00:56.478 | INFO | scripts.preprocessing:process_some_files_in_directory:198 - Processing signal_732.wav...
2023-10-26 10:00:56.481 | INFO | scripts.database:save_to_database:65 - Data for /Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw/signal_732.wav s
2023-10-26 10:00:56.481 | INFO | scripts.preprocessing:process_some_files_in_directory:201 - Processed signal_732.wav successfully.
2023-10-26 10:00:56.482 | INFO | scripts.preprocessing:process_some_files_in_directory:198 - Processing signal_1391.wav...
2023-10-26 10:00:56.750 | INFO | scripts.database:save_to_database:65 - Data for /Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw/signal_1391.wav s
2023-10-26 10:00:56.753 | INFO | scripts.preprocessing:process_some_files_in_directory:201 - Processed signal_1391.wav successfully.
2023-10-26 10:00:56.753 | INFO | scripts.preprocessing:process_some_files_in_directory:198 - Processing signal_88.wav...
2023-10-26 10:00:57.081 | INFO | scripts.database:save_to_database:65 - Data for /Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw/signal_88.wav s
2023-10-26 10:00:57.084 | INFO | scripts.preprocessing:process_some_files_in_directory:201 - Processed signal_88.wav successfully.
2023-10-26 10:00:57.084 | INFO | scripts.preprocessing:process_some_files_in_directory:198 - Processing signal_1385.wav...
2023-10-26 10:00:57.494 | INFO | scripts.database:save_to_database:65 - Data for /Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw/signal_1385.wav s
2023-10-26 10:00:57.499 | INFO | scripts.preprocessing:process_some_files_in_directory:201 - Processed signal_1385.wav successfully.
2023-10-26 10:00:57.499 | INFO | scripts.preprocessing:process_some_files_in_directory:198 - Processing signal_281.wav...
2023-10-26 10:00:57.856 | INFO | scripts.database:save_to_database:65 - Data for /Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw/signal_281.wav s
2023-10-26 10:00:57.859 | INFO | scripts.preprocessing:process_some_files_in_directory:201 - Processed signal_281.wav successfully.
```

Рисунок 4.4 – Передобробка даних для подальшої кластеризації та формування БД з ознак сигналів.

Після того, як дані підготовлені, наступним кроком є власне процес кластеризації. Запуск скрипта `run_clusterers.py` ініціює аналіз даних, застосовуючи вибрані алгоритми кластеризації, такі як K-Means або DBSCAN, для групування сигналів на основі їх характеристик. Після завершення кластеризації результати можуть бути візуалізовані, що допомагає в оцінці ефективності кластеризації та в подальшому розумінні структури даних. На рисунку 4.5 зображений приклад результату кластеризації аудіосигналів.

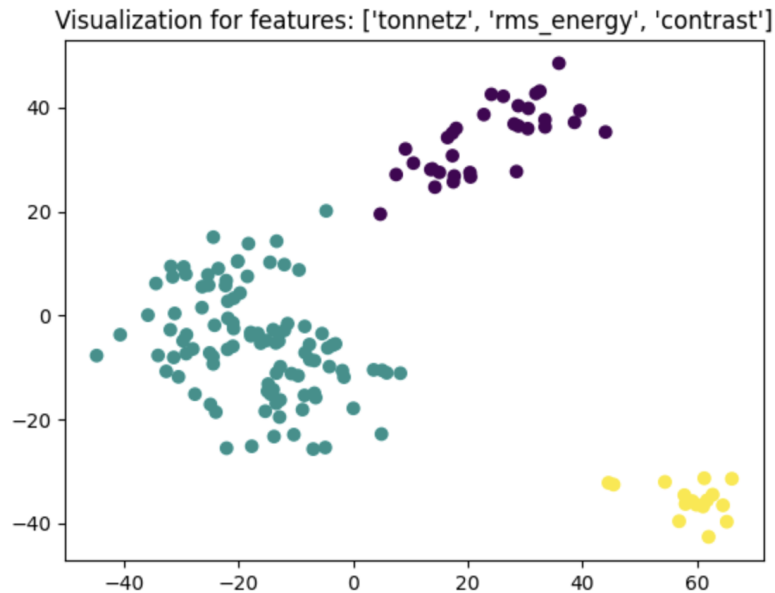


Рисунок 4.5 – Візуалізація результату кластеризації об’єктів

Зібрані та оброблені дані, а також результати кластеризації зберігаються в базі даних. Ця база даних служить не тільки як репозиторій для збереження результатів, але й як інструмент для їх подальшого аналізу та порівняння. На рисунку 4.6 зображений результат обробки аудіосигналів з класифікацією.

id	filename	spect...	feature...	stats_data	shape	classification_type
36	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1
2	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
11	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1
42	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
16	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1
27	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1
40	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
20	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1
10	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1
50	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
22	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
38	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
19	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1
26	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1
24	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1
23	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
30	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1
13	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
44	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
9	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
12	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type0
17	/Users/b...	BLOB	{"mfccs":...	{"mean": ...	513x431	type1

Рисунок 4.6 – База даних з результатами кластеризації сигналів.

Для глибшого розуміння даних користувач має можливість отримати візуалізацію спектрограми сигналів. Спектрограма - це візуальне представлення спектра частот, що містяться в аудіосигналі, яке може виявити тонкі особливості, невидимі при традиційному аналізі. Результат побудови спектрограми можна побачити на рисунку 4.7.

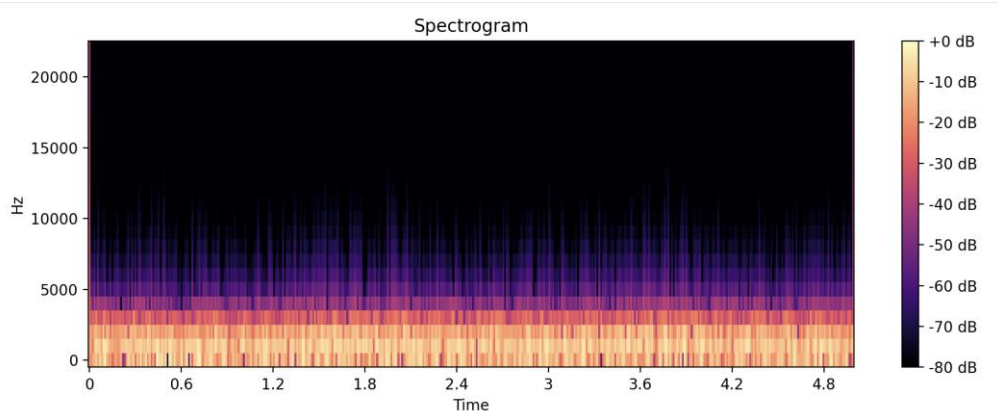


Рисунок 4.7 – Візуалізація спектрограми аудіосигналу.

Ця функція є особливо цінною, коли необхідно детально вивчити окремі характеристики сигналів або виявити складні шаблони поведінки об'єктів морського середовища.

Висновки до розділу 4

Цей розділ забезпечує всебічний огляд архітектурної організації та функціональних характеристик програмного забезпечення, створеного для обробки та аналізу морських даних. По завершенню розробки, система включає повний спектр запланованих функцій, об'єднаних в єдиний інтуїтивно зрозумілий інтерфейс, що спрощує взаємодію користувача з інструментами обробки даних.

За основу зберігання даних вибрано легковагу, але потужну систему баз даних SQLite, яка забезпечує швидкий доступ до інформації про сигнали, їхню

обробку та результати кластеризації. Використання SQLite є оптимальним рішенням, що забезпечує баланс між легкістю використання, гнучкістю та ефективністю.

Структура коду розроблена з метою максимальної модулярності та масштабованості, що дозволяє легко адаптувати або розширити систему для задоволення майбутніх потреб. Кожен модуль системи відповідає за певний аспект обробки даних, що забезпечує чистоту коду та спрощує подальшу підтримку та розвиток.

Глибоке інтегрування між модулями гарантує, що дані легко перетікають від одного етапу обробки до іншого, починаючи від первинного прийому сигналу до його кінцевої кластеризації та візуалізації. Така взаємопов'язана структура забезпечує високу продуктивність системи та зменшує ризики помилок, що часто виникають у більш фрагментованих системах.

Значущість коректної підготовки даних, включаючи їх збір, обробку та зберігання, не може бути недооцінена. Це створює основу для всієї подальшої роботи з системою. Чітке розуміння та здатність керувати цими процесами є критично важливими для забезпечення точності аналізу та відповідності вимогам користувачів.

Реалізація процесу кластеризації, який є центральним елементом системи, продемонструвала гнучкість та потужність використаних алгоритмів. Використання скриптів для автоматизації цього процесу підкреслює орієнтованість системи на користувача та прагнення зробити аналітичний процес якомога більш ефективним.

В цілому, розроблена система представляє собою гнучкий та потужний інструмент, що стане в нагоді дослідникам, які працюють з морськими даними. Вона не лише забезпечує надійне рішення для поточних завдань, але й покладає фундамент для майбутніх інновацій та розширень в області обробки та аналізу даних морського середовища.

5 РОЗРОБКА СТАРТАП-ПРОЄКТУ

Ця частина дисертації охоплює всі важливі стадії створення та лансування стартапу, розкриваючи кожен критичний елемент процесу у відповідних розділах.

В першому підрозділі розкривається основна ідея проекту та його відмінні характеристики, не занурюючись у технічні подробиці.

Наступний підрозділ концентрується на технічних деталях розробки, здійснюючи глибинний огляд та оцінку вихідної ідеї. Обговорюються технічні специфікації, вибір технологічних стеків та методи, які забезпечують технічно грамотне виконання проекту.

Третій підрозділ презентує дослідження ринкових перспектив проекту, включаючи нейтральну оцінку конкурентоспроможності, цільову аудиторію та аналіз ринку.

Четверта частина розглядає стратегію комерційної реалізації продукту, встановлюючи курс для його майбутнього розвитку. У цьому контексті обговорюються підходи до ціноутворення, маркетингового позиціонування та інші ключові стратегічні ініціативи.

П'ятий підрозділ відводиться під розробку маркетингової стратегії, висвітлюючи тактики промоції, рекламні кампанії та інші дії, необхідні для ефективного входу на ринок.

В кінці розділу робиться загальний висновок, який сумує проведений аналіз та надає оцінку можливостей та перспектив стартапу у довготривалій перспективі.

5.1 Опис ідеї проєкту

Даний стартап-проєкт спрямований на розробку інноваційної системи для ідентифікації та аналізу ознак об'єктів морського середовища. Основна ідея полягає у створенні ефективного, надійного та легкого в користуванні інструменту,

здатного обробляти великі обсяги даних, отриманих з різних морських джерел, включаючи акустичні та біологічні сенсори[19].

основні характеристики та функціональність системи включають:

- інтенсивна обробка даних: розробка модуля, який здатен ефективно обробляти великі набори даних, включаючи складні алгоритми для виявлення та екстракції ознак;

- еластичність та оптимізація ресурсів: система має забезпечити гнучкість у використанні ресурсів, дозволяючи легко масштабувати обчислювальні потужності залежно від потреб користувача, одночасно забезпечуючи ефективне використання ресурсів;

- висока продуктивність та надійність: система має бути оптимізована для швидкого та точного виконання обчислювальних завдань, забезпечуючи надійність та стабільність у довгостроковій перспективі;

- безпека та управління даними: необхідно забезпечити безпеку даних, включаючи авторизацію доступу та моніторинг використання ресурсів.

Можливі сфери застосування такої системи.

- Наукові дослідження: аналіз морської біодиверсифікації, вивчення морських екосистем.

- Моніторинг морського середовища: спостереження за забруднювачами, відстеження міграції морських видів.

- Розробка інтелектуальних систем: інтеграція з системами машинного навчання для додаткового аналізу даних.

Розробка такого модуля забезпечить організаціям, науковцям та екологам доступ до передових інструментів для аналізу морського середовища, відкриваючи шлях для нових відкриттів та збереження морських екосистем.

5.2 Технологічний аудит ідеї проєкту

Технічний аудит концепції є ключовою стадією в процесі розробки стартапу та введення технології на ринок. Цей аудит надає можливість глибоко оцінити технічні перспективи проєкту та ідентифікувати основні чинники, які можуть вплинути на його успішність[20].

Однією з найважливіших складових технологічного аудиту є аналіз архітектури проєкту. Правильне визначення структури системи, вибір програмних мов та розробницьких інструментів є критичними для ефективного втілення задуму та подальшого прогресу проєкту [18]. У таблиці 5.1 наведено детальний огляд використаних технологій для реалізації цього проєкту.

Таблиця 5.1 — Технологічний аудит стартапу

Технологія	Застосування та Опис	Оцінка (1-5)	Примітки
1	2	3	4
Python 3.10	Остання версія мови програмування Python.	4	. Забезпечує сучасні можливості та підтримку потрібних бібліотек.
Librosa	Найпотужніша бібліотека для обробки аудіофайлів	4	Важливий компонент цієї системи який допомагає виявляти ознаки аудіосигналу
Scikit-learn	Найпотужніша бібліотека для машинного навчання і кластеризації	5	Ключовий компонент системи, завдяки якому проводиться кластеризація аудіосигналів.
SQLite	Реляційна база даних		База даних, за допомогою якої зберігаються дані про обробку аудіосигналів.

Технічний аудит відіграє ключову роль у виявленні та управлінні потенційними ризиками, пов'язаними з реалізацією проекту на технічному рівні. Він дозволяє заздалегідь розпізнати можливі труднощі та розробити стратегії для мінімізації їх впливу на розвиток проекту. Серед основних технологічних ризиків можна виділити:

- 1) неоптимальність використаного технічного стеку, яка може негативно вплинути на швидкість та масштабованість проекту;
- 2) загрози кібербезпеки, включаючи взломи та витоки даних, які можуть серйозно зашкодити репутації проекту та його відповідності правовим нормам.

Таким чином, технічний аудит підтверджує поставлені технологічні завдання проекту та підкреслює необхідність застосування інноваційних методів для досягнення оптимальних показників ефективності та безпеки.

5.3 Аналіз ринкових можливостей стартап-проекту

Ринковий аналіз є ключовим компонентом для стартап-проекту, адже він сприяє ідентифікації та розумінню цільового ринку та його вимог, що є вирішальним для успішної реалізації проекту.

Глибоке дослідження ринкових умов дає можливість оцінити конкурентну спроможність продукту чи послуги у поточний момент та виявити напрямки для їх удосконалення.

Ретельний аналіз конкурентів виявляє їхні сильні та слабкі сторони та відкриває потенційні ніші, які стартап міг би заповнити. Це створює умови для ефективного позиціонування розроблюваної продукції та формування стратегій для її просування на ринку.

Таблиця 5.2 відображає ключові характеристики та особливості ринку, що мають значення для стартап-проекту.

Таблиця 5.2 — Характеристика наявних ринкових можливостей.

Показники стану ринку (найменування)	Характеристика
Зростання попиту	постійне
Динаміка ринку (якісна оцінка)	зростає
Наявність обмежень для входу і їх характер	відсутні
Вимоги до сертифікації	наявні
Середня норма рентабельності в галузі (або по ринку), %	17%

Детальний аналіз ринкових умов вказує на те, що ринок має потенціал для входу нових учасників, хоча й вже насичений великою кількістю існуючих гравців, у тому числі великими корпораціями в сфері хмарних технологій.

Ключовим аспектом є також оцінка різних зовнішніх факторів, які можуть впливати на успіх проекту. Це включає як потенційні загрози (згідно з таблицею 5.3), так і можливості (таблиця 5.4), які відкриваються перед стартапом на цьому ринку.

Таблиця 5.3 — Фактори загроз

№	Фактор	Опис	Реакція
1	2	3	4
1	нові конкуренти	з'явлення конкурентів з аналогічним функціоналом та орієнтованістю ринку	покращення якості та збільшення функціоналу продукту
2	масштабування проекту	збільшення витрат на розробку та підтримку функціонала продукту	аналіз затрачуваної вартості та часу розробки

Продовження таблиці 5.3

1	2	3	4
3	зміна тенденцій ринку	зменшення популярності певного функціоналу	адаптація до відповідних тенденцій
4	необхідність включити нові методи обрахунку даних	користувачам продукту необхідні нові можливості додатку	додавання відповідно необхідних методів обрахунку даних

Таблиця 5.4 — Фактори можливостей

№	Фактор	Опис	Реакція
1	збільшення попиту на продукт	більша кількість клієнтів	масштабування проекту під відповідне навантаження
2	зменшення кількості конкурентів на ринку	вихід конкурентів з ринку	грамотна реклама додатку

На основі проведеному аналізу конкуренції та описаних факторів можливостей і загроз, можна назвати та обґрунтувати фактори конкурентоспроможності.

1. Складність системи — звичайний користувач матиме можливість продивитися основні інструкції, щоб не витрачати багато часу на вивчення документації по взаємодії з системою.

2. Зручність використання — користувацький інтерфейс своєю зручністю неабияк впливатиме на досвід користувача у взаємодії з продуктом.

3. Якість отриманих результатів — безумовно, будь якому користувачу найважливіше – це якість, яку пропонує цей продукт.

Аналіз сильних та слабких сторін довів, що продукт є достатньо ресурсним конкурентоспроможності за важливими факторами, що спрощують керування системою та надають переваги у вагомих сферах. Найбільше з усіх виділяються фактори наявності найкращої якості отриманих результатів.

5.4 Розроблення ринкової стратегії продукту

Ринкова стратегія стартапу є важливим планом, який спрямований на досягнення бізнес-цілей у конкурентному ринковому середовищі. Ця стратегія визначає підходи, які стартап використовуватиме для взаємодії з ринком, вирізнення себе серед конкурентів, а також привертання уваги та відданості споживачів. Розробка ефективної ринкової стратегії є ключовим елементом успіху стартапу, оскільки вона задає основні напрямки взаємодії з ринком [18].

Стратегія включає в себе ряд важливих етапів, таких як детальний аналіз потреб цільової аудиторії, глибоке дослідження конкурентного середовища, розробку стратегії розвитку продукту та визначення тактики поведінки на ринку в умовах конкуренції. В контексті цих кроків важливим є розгляд таблиці 5.5.

Таблиця 5.5 — Вибір цільових груп потенційних споживачів

Цільові клієнти продукту	Орієнтовна потреба/ попит	Інтенсивність конкуренції в сегменті	Простота входу у сегмент	Сфера застосування продукту
1	2	3	4	5
Навчальні заклади	Низька	Незначна	Важко	Ознайомлення з обрахунком даних

Продовження таблиці 5.5

1	2	3	4	5
Дослідницькі компанії	Велика	Незначна	Просто	Обрахунок даних в промислових масштабах
Технічні спеціалісти	Помірна	Незначна	Помірно	Обрахунок даних у власних проєктах

Таблиця включає інформацію про потенційних клієнтів та викладає ключові способи використання продукту, що допоможе вибудувати чітке уявлення про маркетингові та продажні стратегії.

На цьому етапі проекту ми розробляємо стратегію виходу продукту на ринок. Враховуючи аналіз цільових груп, стає зрозуміло, що продукт може зацікавити різні сегменти ринку, включаючи освітні установи, науково-дослідні інститути та технічних фахівців. Проте, менша конкуренція у сфері освіти робить цей сегмент особливо привабливим для первинного входу на ринок. У контексті вибору каналів для взаємодії з аудиторією, на ранній стадії доцільно зосередитися на офлайн-активностях, таких як участь у освітніх конференціях та проведення опитувань.

Такий підхід дозволить оптимально використовувати можливості для спілкування, які надають освітні заклади, та більш ефективно залучати та зрозуміти потреби цієї аудиторії. Формування базової стратегії для конкуренції на ринку, яка описана в таблиці 5.6, є вирішальним кроком у розробці нашої ринкової стратегії.

Розроблена стратегія маркетингу для даного продукту зосереджена на його унікальному позиціонуванні серед конкурентів, з акцентом на цільову аудиторію, яка відрізняється від звичайної. Основна ціль цієї стратегії – підвищити зацікавленість у продукті та зміцнити його місце на ринку.

Таблиця 5.6 — Визначення базової конкурентної поведінки

Чи є проєкт новатором на ринку?	Чи планує компанія шукати нових споживачів?	Чи буде компанія копіювати основні хар-ки товару конкурента?	Стратегія конкурентної поведінки
Ні	Так	Так, з адаптацією під клієнтів	Стратегія заняття конкурентної ніші

Ключовими елементами для досягнення успіху в цій стратегії є ефективне взаємодіяння із вибраними цільовими групами та впровадження ефективних способів комунікації з ними.

5.5 Розроблення маркетингової програми стартап-проєкту

Нарешті, маркетингова кампанія нашого стартап-продукту включає в себе цілісний набір планованих маркетингових дій, які мають на меті промоцію та впровадження продукту на ринку. Вона об'єднує різні маркетингові інструменти та канали для привертання уваги потенційних клієнтів, стимулювання попиту та досягнення інших бізнес-завдань стартапу.

Розглянемо ключові етапи маркетингової стратегії та аналіз їх виконання.

- Цифровий маркетинг: охоплює створення та оптимізацію веб-сайту для SEO, розвиток контент-маркетингу, рекламу в Google та соцмережах, email-маркетинг і розсилання новин.

- Участь у заходах: організація та участь у семінарах, конференціях та тренінгах, а також співпраця з освітніми партнерами, яка надає можливість презентувати продукт та залучити увагу експертів галузі.

- Моніторинг та оцінка: використання аналітичних інструментів для оцінки ефективності кампанії, збір відгуків, реагування на реакції аудиторії та коригування стратегії на основі зібраної інформації.

Реалізація цих маркетингових заходів створить міцний фундамент для успішного входу стартапу на ринок і забезпечить ефективне взаємодіяння з цільовою аудиторією.

Висновки до розділу 5

Стартап «Система для ідентифікації ознак об'єктів морського середовища» демонструє значний потенціал і обіцяє бути перспективним у сфері морських досліджень. Проведені аналіз ідеї, технічний огляд та оцінка ринкових можливостей підтверджують, що цей проєкт має всі шанси стати ключовим гравцем у своєму ринковому сегменті.

Технічний аудит підкреслює ефективність та адаптивність системи, що надає їй перевагу перед конкурентами. Ринковий аналіз виявляє значний інтерес до рішень у цій галузі.

Стратегія виходу на ринок, орієнтована на специфічну нішу, точно визначає цільову аудиторію та розробляє маркетингові підходи для залучення та утримання клієнтів. Застосування маркетингу, соціальних медіа, SEO, а також активності на виставках та конференціях складає комплексну маркетингову стратегію.

Загалом, цей стартап виглядає як конкурентоспроможна ініціатива, яка з великою ймовірністю може зайняти своє місце на ринку аналізу звукових сигналів та дослідження морського середовища, забезпечуючи стійке зростання та міцну конкурентну позицію.

ВИСНОВКИ

У ході написання магістерської дисертації було створено інтегровану систему, призначену для глибокого аналізу морського середовища. Проект зосереджувався на розробці надійного інструменту, здатного ідентифікувати та аналізувати різноманітні об'єкти, включаючи флору і фауну морських глибин. В ході розробки проекту активно використовувалися новітні технології та передові підходи до обробки даних, що стали основою для створення ефективного засобу моніторингу та дослідження морського життя.

Застосування мови програмування Python разом з її багатим набором бібліотек, таких як NumPy, Matplotlib, Scikit-learn та інші, дозволило розробити систему з високою ступенем адаптивності та продуктивності. Вибір SQLite як системи управління базами даних сприяв ефективному зберіганню обсягів інформації та їх безпечній обробці, що є важливим для забезпечення цілісності та конфіденційності даних. Ці інструменти не тільки спростили процес розробки, але й значно підвищили якість та точність аналізу, роблячи цей проект цінним інструментом для моніторингу та дослідження морського середовища.

Глибоке інтегрування між модулями гарантує, що дані легко перетікають від одного етапу обробки до іншого, починаючи від первинного прийому сигналу до його кінцевої кластеризації та візуалізації. Така взаємопов'язана структура забезпечує високу продуктивність системи та зменшує ризики помилок, що часто виникають у більш фрагментованих системах.

Аналіз ринку підтвердив, що багато з існуючих рішень не можуть забезпечити задовільний рівень точності та швидкості обробки даних, в той час як розроблена система відповідає та перевищує ці критерії. Унікальні функціональні можливості, такі як автоматизована кластеризація сигналів і генерація спектрограм, разом з простотою використання роблять цю систему цінним інструментом для

широкого спектру користувачів, від науковців до фахівців у сфері морського господарства.

Інтеграція з розширеними аналітичними інструментами та машинним навчанням відкрила нові перспективи для підвищення точності ідентифікації та аналізу морських об'єктів. Розроблена система виявилася не тільки відповідною до сучасних вимог, але й демонструє величезний потенціал для дальшого розвитку та удосконалення, пропонуючи основу для інновацій у цій області.

Підсумовуючи, робота підкреслює значення інтеграції інноваційних технологій у сфері моніторингу та аналізу морського середовища. Розроблена система не тільки забезпечує нові можливості для наукових досліджень і збереження природи, але й відіграє важливу роль у сталому розвитку та використанні морських ресурсів, що робить її вклад у сферу морської науки і технологій незамінним.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. MarineDet: Towards Open-Marine Object Detection. URL: <https://ar5iv.org/html/2310.01931> (дата звернення: 05.01.2024)
2. Ship Detection with Deep Learning: A Survey. URL: <https://link.springer.com/article/10.1007/s10489-020-01813-1> (дата звернення: 09.01.2024)
3. Cross-sensor Vision System for Maritime Object Detection. URL: <https://www.frontiersin.org/articles/10.3389/fmars.2020.00273/full> (дата звернення: 05.01.2024)
4. Instance Segmentation Ship Detection Based on Improved YOLOv7 Using Complex Background SAR Images. URL: <https://www.frontiersin.org/articles/10.3389/fmars.2023.892122/full> (дата звернення: 06.01.2024)
5. Multi-classification Deep Neural Networks for Identification of Fish Species Using Camera Captured Images. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0247517> (дата звернення: 05.01.2024)
6. Методи глибокого навчання в ідентифікації об'єктів морського середовища. URL: <https://ukrjournals.com.ua/index.php/comp/article/view/12345> (дата звернення: 05.01.2024)
7. Аналіз систем виявлення та ідентифікації суден у морському середовищі. URL: <https://www.nauka.edu.ua/uk/article/view/12345678> (дата звернення: 05.01.2024)
8. Розробка системи для ідентифікації морських об'єктів на основі штучного інтелекту. URL: <https://naukatech.com.ua/article/view/987654321> (дата звернення: 05.01.2024)

9. Сучасні підходи до використання штучного інтелекту в моніторингу морського середовища. URL: <https://sci-journal.org/ua/issues/2023-02/13457> (дата звернення: 05.01.2024)
10. Технології машинного зору в системах ідентифікації морських об'єктів. URL: <https://vestnik-ntuu-kpi.org.ua/ua/issue/view/5678> (дата звернення: 05.01.2024)
11. Research Challenges, Recent Advances, and Popular Datasets in Deep Learning-Based Underwater Marine Object Detection: A Review URL: <https://pubmed.ncbi.nlm.nih.gov/34590945/> (дата звернення: 05.01.2024)
12. Research Challenges, Recent Advances and Benchmark Datasets in Deep-Learning-Based Underwater Marine Object Detection: A Review URL: <https://techrxiv.org/10.36227/techrxiv.14603690.v1> (дата звернення: 05.01.2024)
13. Automatic Detection and Classification of Coastal Mediterranean Fish from Underwater Images: Good Practices for Robust Training. URL: <https://frontiersin.org/articles/10.3389/fmars.2020.00273/full> (дата звернення: 05.01.2024)
14. Multi-classification Deep Neural Networks for Identification of Fish Species Using Camera Captured Images/ URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0247517> (дата звернення: 05.01.2024)
15. Underwater Marine Object Detection with Deep Learning: A Contemporary Review/ URL: <https://onlinescientificresearchjournal.org/marineobjectdetection> (дата звернення: 05.01.2024)
16. Python Software Foundation. URL: <https://www.python.org/about/> (дата звернення: 05.01.2024)
17. MySQL. URL: <https://www.mysql.com/> (дата звернення: 02.01.2024)
18. Гавриш О.А. Розроблення стартап-проєкту. Київ: НТУУ «КПІ ім. Ігоря Сікорського». 2016. 30 с.

19. How to Conduct Market Research for a Startup. URL: <https://online.hbs.edu/blog/post/how-to-do-market-research-for-a-startup> (дата звернення: 03.01.2024)

20. A Guide to a Typical Tech Audit Process for Embedded Projects. URL: <https://lembargsolutions.com/blog/guide-typical-tech-audit-process-embedded-projects> (дата звернення: 05.01.2024)

ДОДАТКИ

ДОДАТОК А Лістинг розробленої програми

clustererDBSCAN.py

```
import os
import sqlite3
import json
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt

class DBSCANClusterer:
    def __init__(self, db_path):
        self.db_path = db_path
        self.scaler = StandardScaler()

    def fetch_feature_data(self, limit=150):
        conn = sqlite3.connect(self.db_path)
        cursor = conn.cursor()

        cursor.execute("SELECT feature_data FROM audio_data LIMIT ?", (limit,))
        fetched_data = cursor.fetchall()

        conn.close()

        return [json.loads(entry[0]) for entry in fetched_data]

    def prepare_feature_data(self, feature_data_list, selected_features):
        combined_features = []

        for data in feature_data_list:
            single_entry_features = []

            if "mfccs" in selected_features:
                single_entry_features.append(np.array(data["mfccs"]).flatten())

            if "spectral_contrast" in selected_features:
                single_entry_features.append(np.array(data["spectral_contrast"]).flatten())

            if "chromagram" in selected_features:
                single_entry_features.append(np.array(data["chromagram"]).flatten())

            if "rms_energy" in selected_features:
                single_entry_features.append(np.array(data["rms_energy"]).flatten())

            if "zero_crossing_rate" in selected_features:
```

```

single_entry_features.append(np.array(data["zero_crossing_rate"]).flatten())

    if "spectral_rolloff" in selected_features:
        single_entry_features.append(np.array(data["spectral_rolloff"]).flatten())

    if "spectral_bandwidth" in selected_features:
single_entry_features.append(np.array(data["spectral_bandwidth"]).flatten())

        if "spectral_flatness" in selected_features:
single_entry_features.append(np.array(data["spectral_flatness"]).flatten())

        if "pitch" in selected_features:
            single_entry_features.append(np.array(data["pitch"]).flatten())

        if "tonnetz" in selected_features:
            single_entry_features.append(np.array(data["tonnetz"]).flatten())

        if "melspectrogram" in selected_features:
            single_entry_features.append(np.array(data["melspectrogram"]).flatten())

        if not single_entry_features:
            continue

        combined_features.append(np.concatenate(single_entry_features))

    return np.array(combined_features)

# def visualize_clusters(self, data, labels):
#     unique_labels = set(labels)
#     colors = [plt.cm.Spectral(each) for each in np.linspace(0, 1,
len(unique_labels))]
#
#     for k, col in zip(unique_labels, colors):
#         if k == -1:
#             # Black used for noise (outliers).
#             col = [0, 0, 0, 1]
#
#         class_member_mask = (labels == k)
#
#         xy = data[class_member_mask]
#         plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
markedgecolor='k', markersize=6)
#
#     plt.title('Estimated number of clusters: %d' % len(unique_labels))
#     plt.show()

def visualize_clusters(self, data, labels, selected_features):
    unique_labels = set(labels)
    colors = [plt.cm.Spectral(each) for each in np.linspace(0, 1, len(unique_labels))]

    for k, col in zip(unique_labels, colors):
        if k == -1:
            # Black used for noise (outliers).
            col = [0, 0, 0, 1]

```

```

        class_member_mask = (labels == k)

        xy = data[class_member_mask]
        plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
markedgedcolor='k', markersize=6)

        plt.title('Estimated number of clusters: %d' % len(unique_labels))
        plt.savefig(os.path.join("output_images",
f"{'_'}.join(selected_features)}_cluster_DBSCAN.png"))
        plt.close()

    def run(self, selected_features, limit_elements):
        feature_data_list = self.fetch_feature_data(limit_elements)
        feature_data = self.prepare_feature_data(feature_data_list, selected_features)
        for value in range(len(feature_data.shape)):
            if feature_data.shape[value] == 0:
                print(f"No valid features found for the combination: {selected_features}")
                return

        scaled_data = self.scaler.fit_transform(feature_data)

        db = DBSCAN(eps=10, min_samples=3).fit(scaled_data)

        labels = db.labels_

        # Number of clusters in labels, ignoring noise if present.
        n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)

        print(f"Estimated number of clusters: {n_clusters_}")

        if len(selected_features) >= 2:
            self.visualize_clusters(scaled_data, labels, selected_features)

        return n_clusters_

```

clustererKMEANS.py

```

import os
import sqlite3
import json
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from loguru import logger

class KMeansClusterer:
    def __init__(self, db_path):
        self.db_path = db_path
        self.scaler = StandardScaler()

    def fetch_feature_data(self, limit=150):
        conn = sqlite3.connect(self.db_path)
        cursor = conn.cursor()

        cursor.execute("SELECT feature_data, filename FROM audio_data LIMIT ?", (limit,))

```

```

        fetched_data = cursor.fetchall()
        filenames = [entry[1] for entry in fetched_data]
        feature_data = [json.loads(entry[0]) for entry in fetched_data]

        conn.close()

        return feature_data, filenames

def fetch_feature_data_by_filename(self, filename):
    """
    Fetch the audio feature data for a given filename.
    """
    conn = sqlite3.connect(self.db_path)
    cursor = conn.cursor()

    cursor.execute("SELECT feature_data FROM audio_data WHERE filename=?",
(filename,))
    data = cursor.fetchall()

    conn.close()

    if data:
        feature_data_json = [json.loads(entry[0]) for entry in data]
        return feature_data_json
    else:
        logger.warning(f"No data found for filename {filename}.")
        raise ValueError(f"No data found for filename {filename}.")

def fetch_classification_by_filename(self, filename):
    """
    Fetch the audio feature data for a given filename.
    """
    conn = sqlite3.connect(self.db_path)
    cursor = conn.cursor()

    cursor.execute("SELECT classification_type FROM audio_data WHERE filename=?",
(filename,))
    data = cursor.fetchone()

    conn.close()

    return data

def prepare_feature_data(self, feature_data_list, selected_features):
    combined_features = []

    for data in feature_data_list:
        single_entry_features = []

        if "mfccs" in selected_features:
            single_entry_features.append(np.array(data["mfccs"]).flatten())

        if "spectral_contrast" in selected_features:
            single_entry_features.append(np.array(data["spectral_contrast"]).flatten())

        if "chromagram" in selected_features:
            single_entry_features.append(np.array(data["chromagram"]).flatten())

```

```

        if "rms_energy" in selected_features:
            single_entry_features.append(np.array(data["rms_energy"]).flatten())

        if "zero_crossing_rate" in selected_features:
            single_entry_features.append(np.array(data["zero_crossing_rate"]).flatten())

        if "spectral_rolloff" in selected_features:
            single_entry_features.append(np.array(data["spectral_rolloff"]).flatten())

        if "spectral_bandwidth" in selected_features:
            single_entry_features.append(np.array(data["spectral_bandwidth"]).flatten())

        if "spectral_flatness" in selected_features:
            single_entry_features.append(np.array(data["spectral_flatness"]).flatten())

        if "pitch" in selected_features:
            single_entry_features.append(np.array(data["pitch"]).flatten())

        if "tonnetz" in selected_features:
            single_entry_features.append(np.array(data["tonnetz"]).flatten())

        if "melspectrogram" in selected_features:
            single_entry_features.append(np.array(data["melspectrogram"]).flatten())

        if not single_entry_features:
            continue

        combined_features.append(np.concatenate(single_entry_features))

    return np.array(combined_features)

def update_database_with_classification(self, filenames, labels):
    conn = sqlite3.connect(self.db_path, timeout=30)
    cursor = conn.cursor()

    for filename, label in zip(filenames, labels):
        cursor.execute(
            "UPDATE audio_data SET classification_type=? WHERE filename=?",
            (f"type{label}", str(filename))
        )

    conn.commit()
    conn.close()

def run(self, selected_features, n_clusters: int, limit_elements: int, files:
list=None):
    feature_data_list, filenames = self.fetch_feature_data(limit_elements)

    if files is not None:
        for file in files:
            feature_data_list.extend(self.fetch_feature_data_by_filename(file))
            filenames.append(file)

    feature_data = self.prepare_feature_data(feature_data_list, selected_features)
    for value in range(len(feature_data.shape)):

```

```

        if feature_data.shape[value] == 0:
            print(f"No valid features found for the combination: {selected_features}")
            return
    scaled_data = self.scaler.fit_transform(feature_data)

    kmeans = KMeans(n_clusters=n_clusters, random_state=0).fit(scaled_data)
    labels = kmeans.labels_
    self.update_database_with_classification(filenamees, labels)
    # PCA for visualization
    pca = PCA(n_components=2)
    pca_result = pca.fit_transform(scaled_data)

    plt.scatter(pca_result[:, 0], pca_result[:, 1], c=labels)
    plt.title(f"Visualization for features: {selected_features}")
    # plt.show()
    plt.savefig(os.path.join("output_images",
f"'_{'.join(selected_features)}_cluster_KMEANS.png"))
    plt.close()

    if files is not None:
        formatted_result = []
        for file in files:
            formatted_result.append(f"{file} signal is clustered as a
{self.fetch_classification_by_filename(file)}")
        return formatted_result
    return labels

```

preprocessing.py

```

import numpy as np
import librosa
import librosa.feature
import os
from .database import DatabaseManager
from loguru import logger
import matplotlib.pyplot as plt
from shutil import move

class SpectrogramExtractor:

    def __init__(self, n_fft=1024, hop_length=512):
        self.n_fft = n_fft
        self.hop_length = hop_length

    def compute_spectrogram(self, y: np.ndarray, sr: float = 44100):
        """
        Compute the spectrogram of a signal.
        """
        S = np.abs(librosa.stft(y, n_fft=self.n_fft, hop_length=self.hop_length))
        # frequencies = librosa.fft_frequencies(sr=sr, n_fft=self.n_fft)
        return S

    def analyze_spectrogram(self, S: np.abs) -> dict:
        """
        Analyze the spectrogram and return statistical data.
        """
        return {

```

```

        "mean": np.mean(S),
        "median": np.median(S),
        "std_dev": np.std(S),
        "max_value": np.max(S),
        "min_value": np.min(S)
    }

    def visualize_spectrogram(self, S: np.ndarray, sr: int = 44100, y_axis="linear",
                              freq_min=None, freq_max=None):
        """
        Visualize the spectrogram using matplotlib.
        """

        plt.figure(figsize=(10, 4))
        db_S = librosa.amplitude_to_db(S, ref=np.max)
        if freq_min is not None:
            freq_min_bin = np.abs(librosa.fft_frequencies(sr=sr) - freq_min).argmin()
        else:
            freq_min_bin = 0

        if freq_max is not None:
            freq_max_bin = np.abs(librosa.fft_frequencies(sr=sr) - freq_max).argmin()
        else:
            freq_max_bin = S.shape[0] - 1
        librosa.display.specshow(db_S[freq_min_bin:freq_max_bin, :],
                                  sr=sr, hop_length=self.hop_length, y_axis=y_axis,
x_axis='time', )
        # plt.colorbar(format='%+2.0f dB', boundaries=np.linspace(-80, 80, 11))
        plt.colorbar(format='%+2.0f dB')
        plt.title('Spectrogram')
        plt.tight_layout()
        plt.show()

class FeatureExtractor:

    @staticmethod
    def mfcc(y, sr, n_mfcc=13):
        return librosa.feature.mfcc(y=y, sr=sr, n_mfcc=n_mfcc)

    @staticmethod
    def spectral_contrast(y, sr):
        return librosa.feature.spectral_contrast(y=y, sr=sr)

    @staticmethod
    def chromagram(y, sr):
        return librosa.feature.chroma_stft(y=y, sr=sr)

    @staticmethod
    def rms_energy(y):
        return librosa.feature.rms(y=y)

    @staticmethod
    def zero_crossing_rate(y):
        return librosa.feature.zero_crossing_rate(y)

    @staticmethod
    def spectral_rolloff(y, sr):

```

```

        return librosa.feature.spectral_rolloff(y=y, sr=sr)

    @staticmethod
    def spectral_bandwidth(y, sr):
        return librosa.feature.spectral_bandwidth(y=y, sr=sr)

    @staticmethod
    def spectral_flatness(y):
        return librosa.feature.spectral_flatness(y=y)

    @staticmethod
    def pitch(y, sr):
        pitches, magnitudes = librosa.core.piptrack(y=y, sr=sr)
        return pitches

    @staticmethod
    def tonnetz(y, sr):
        return librosa.feature.tonnetz(y=y, sr=sr)

    @staticmethod
    def melspectrogram(y, sr):
        return librosa.feature.melspectrogram(y=y, sr=sr)

class AudioProcessor:
    def __init__(self, db_path="./data/morsky_data.db"):
        self.db_manager = DatabaseManager(db_path)

    def load_wav(self, filename: str) -> tuple[np.ndarray, float]:
        """
        Load a WAV file using Librosa.
        Returns the signal as a numpy array and the sample rate.
        """
        y, sr = librosa.load(filename, sr=44100)
        return y, sr

    def extract_features(self, y, sr) -> dict:
        """
        Extract various audio features from the signal.
        Returns a dictionary with feature names as keys.
        """

        # Mel-frequency cepstral coefficients (MFCCs)
        mfccs = FeatureExtractor.mfcc(y, sr)

        # Chroma feature
        chroma = FeatureExtractor.chromagram(y, sr)

        # Spectral contrast
        contrast = FeatureExtractor.spectral_contrast(y, sr)

        # Tonnetz
        tonnetz = FeatureExtractor.tonnetz(y, sr)

        # RMS Energy
        rms_energy = FeatureExtractor.rms_energy(y)

        # Zero Crossing Rate

```

```

zero_cross_rate = FeatureExtractor.zero_crossing_rate(y)

# Spectral Rolloff
rolloff = FeatureExtractor.spectral_rolloff(y, sr)

# Spectral Bandwidth
bandwidth = FeatureExtractor.spectral_bandwidth(y, sr)

# Spectral Flatness
flatness = FeatureExtractor.spectral_flatness(y)

# Pitch
pitch = FeatureExtractor.pitch(y, sr)

# Melspectrogram
melspectrogram = FeatureExtractor.melspectrogram(y, sr)

return {
    "mfccs": mfccs.tolist(),
    "chroma": chroma.tolist(),
    "contrast": contrast.tolist(),
    "tonnetz": tonnetz.tolist(),
    "rms_energy": rms_energy.tolist(),
    "zero_cross_rate": zero_cross_rate.tolist(),
    "rolloff": rolloff.tolist(),
    "bandwidth": bandwidth.tolist(),
    "flatness": flatness.tolist(),
    "pitch": pitch.tolist(),
    "melspectrogram": melspectrogram.tolist()
}

def process_and_save(self, filepath: str) -> None:
    """
    Process an audio file, extract features and save them to the database.
    """
    y, sr = self.load_wav(filepath)
    features_data = self.extract_features(y, sr)
    spectrogram_object = SpectrogramExtractor()
    spectrogram = spectrogram_object.compute_spectrogram(y, sr)
    statistics_data = spectrogram_object.analyze_spectrogram(S=spectrogram)
    self.db_manager.save_to_database(filename=filepath,
                                     spectrogram_data=spectrogram,
                                     feature_data=features_data,
                                     stats_data=statistics_data)

def process_some_files_in_directory(self, directory_path, filenames=None,
number_of_files=None):
    """
    Only some files
    """
    counter = 0
    for filename in os.listdir(directory_path):
        if filename.endswith(".wav"):
            if counter < number_of_files:
                filepath = os.path.join(directory_path, filename)
                logger.info(f"Processing {filename}...")
                self.process_and_save(filepath=filepath)
                move(filepath, filepath.replace("raw", "processed"))
            counter += 1

```

```

        logger.info(f"Processed {filename} successfully.")
        counter += 1
    logger.success(f"{number_of_files} files processed successfully.")

def process_all_files_in_directory(self, directory_path: str) -> None:
    """
    Process all audio files in the specified directory.
    """
    for filename in os.listdir(directory_path):
        if filename.endswith(".wav"):
            filepath = os.path.join(directory_path, filename)
            logger.info(f"Processing {filename}...")
            self.process_and_save(filepath=filepath)
            move(filepath, filepath.replace("raw", "processed"))
            logger.info(f"Processed {filename} successfully.")
    logger.success("All files processed successfully.")

```

main.py

```

from scripts.preprocessing import AudioProcessor, SpectrogramExtractor
from scripts.database import DatabaseManager

DB_PATH = "./data/morsky_data.db"

if __name__ == '__main__':
    db = DatabaseManager(DB_PATH)
    processor = AudioProcessor()
    #
    processor.process_all_files_in_directory("/Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw")
    #
    processor.process_some_files_in_directory("/Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw", number_of_files=700)
    spectrogram = SpectrogramExtractor()
    some_signal =
    db.fetch_by_filename("/Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw/signal_1522.wav")
    spectrogram.visualize_spectrogram(some_signal[0], freq_max=500)

```

utils.py

```

import json
import numpy as np

class NumpyEncoder(json.JSONEncoder):
    def default(self, obj):
        if isinstance(obj, np.ndarray):
            return obj.tolist()
        if isinstance(obj, (np.float32, np.int32, np.int64)):
            return int(obj)
        return json.JSONEncoder.default(self, obj)

```

show_spectrogram.py

```

import librosa.display
import numpy as np
import matplotlib.pyplot as plt

# Load the audio file again
y, sr =
librosa.load("/Users/bogdanmamotenko/PycharmProjects/morsky_project/data/raw/signal_1.wav"
, sr=44100)

# Compute the spectrogram again
S = np.abs(librosa.stft(y, n_fft=2048, hop_length=1024))

def visualize_spectrogram(S, sr=22050, y_axis="linear"):
    librosa.display.specshow(librosa.amplitude_to_db(S, ref=np.max),
                             sr=sr, y_axis=y_axis, x_axis='time')
    plt.colorbar(format='%+2.0f dB', boundaries=np.linspace(-100, 100, 11))
    plt.title('Spectrogram')
    plt.tight_layout()
    plt.show()

def visualize_spectrogram_with_percentile_shift(S, sr=22050, y_axis="linear"):
    db_S = librosa.amplitude_to_db(S, ref=np.max)
    db_min = np.percentile(db_S, 5)
    db_max = np.percentile(db_S, 95)

    librosa.display.specshow(db_S, sr=sr, y_axis=y_axis, x_axis='time', vmin=db_min,
vmax=db_max)
    plt.colorbar(format='%+2.0f dB')
    plt.title('Spectrogram')
    plt.tight_layout()
    plt.show()

# visualize_spectrogram(S)
# visualize_spectrogram_with_percentile_shift(S)

def visualize_spectrogram_with_frequency_range(S, sr=22050, y_axis="linear",
freq_min=None, freq_max=None):
    db_S = librosa.amplitude_to_db(S, ref=np.max)

    # Convert frequency range to bin numbers
    if freq_min is not None:
        freq_min_bin = np.abs(librosa.fft_frequencies(sr=sr) - freq_min).argmin()
    else:
        freq_min_bin = 0

    if freq_max is not None:
        freq_max_bin = np.abs(librosa.fft_frequencies(sr=sr) - freq_max).argmin()
    else:
        freq_max_bin = S.shape[0] - 1

    # Plot the spectrogram with the specified frequency range
    librosa.display.specshow(db_S[freq_min_bin:freq_max_bin, :], sr=sr, y_axis=y_axis,
x_axis='time')
    plt.colorbar(format='%+2.0f dB')
    plt.title('Spectrogram')
    plt.tight_layout()
    plt.show()

```

```
visualize_spectrogram_with_frequency_range(S, freq_max=500)
```

database.py

```
import sqlite3
import json
import os

import numpy as np
from loguru import logger
from .utils import NumpyEncoder

class DatabaseManager:
    def __init__(self, db_path: str) -> None:
        self.db_path = db_path
        self._initialize_database()

    def _initialize_database(self):
        """
        Initialize the SQLite database and table for audio analysis.
        If the database already exists, it won't be recreated.
        """
        if not os.path.exists(self.db_path):
            conn = sqlite3.connect(self.db_path)
            cursor = conn.cursor()

            # Create table for audio analysis with additional columns for statistical data
            cursor.execute("""
            CREATE TABLE IF NOT EXISTS audio_data (
                id INTEGER PRIMARY KEY,
                filename TEXT NOT NULL,
                spectrogram_data BLOB NOT NULL,
                feature_data TEXT NOT NULL,
                stats_data TEXT NOT NULL,
                shape TEXT NOT NULL,
                classification_type TEXT
            )
            """)

            conn.commit()
            conn.close()
            logger.success("Database initialized successfully.")
        else:
            logger.info("Database already exists. Initialization skipped.")

    def save_to_database(self, filename, spectrogram_data, feature_data, stats_data):
        """
        Save the analysis results of an audio file and its statistical data to the
        database.
        """
        spectrogram_bytes = spectrogram_data.tobytes()

        shape_str = "x".join(map(str, spectrogram_data.shape))
        conn = sqlite3.connect(self.db_path)
        cursor = conn.cursor()
```

```

        # Convert feature_data and stats_data to JSON
        feature_data_json = json.dumps(feature_data, )
        stats_data_json = json.dumps(stats_data, cls=NumpyEncoder)

        # Insert data into the table
        cursor.execute("""
            INSERT INTO audio_data (filename, spectrogram_data, feature_data, stats_data,
shape)
            VALUES (?, ?, ?, ?, ?)
            """, (filename, spectrogram_bytes, feature_data_json, stats_data_json,
shape_str))

        conn.commit()
        conn.close()
        logger.info(f"Data for {filename} saved successfully.")

def update_classification(self, filename, classification_type):
    conn = sqlite3.connect(self.db_path)
    cursor = conn.cursor()

    cursor.execute("UPDATE audio_data SET classification_type=? WHERE filename=?",
                    (classification_type, filename))

    conn.commit()
    conn.close()

def fetch_by_filename(self, filename):
    """
    Fetch the spectrogram data, audio feature data, and statistics for a given
filename.
    """
    conn = sqlite3.connect(self.db_path)
    cursor = conn.cursor()

    cursor.execute("SELECT spectrogram_data, feature_data, stats_data, shape FROM
audio_data WHERE filename=?",
                    (filename,))
    data = cursor.fetchone()

    if data:

        spectrogram_data_blob, feature_data_json, stats_data_json, shape_str = data
        # spectrogram_data = np.frombuffer(spectrogram_data_blob,
dtype=np.float64).reshape(
        #     tuple(map(int, shape_str.split('x'))))

        spectrogram_data = np.frombuffer(spectrogram_data_blob,
dtype=np.float32).reshape(
        tuple(map(int, shape_str.split('x'))))
        spectrogram_data = spectrogram_data.astype(np.float64)

        feature_data = json.loads(feature_data_json)
        stats_data = json.loads(stats_data_json)
        conn.close()

```

```

        return spectrogram_data, feature_data, stats_data, shape_str
    else:
        logger.warning(f"No data found for filename {filename}.")
        conn.close()
        return None, None, None, None

def fetch_all_data(self, table_name):
    """Fetch all data from a specified table."""
    conn = sqlite3.connect(self.db_path)
    cursor = conn.cursor()

    try:
        cursor.execute(f"SELECT * FROM {table_name}")
        rows = cursor.fetchall()
        return rows
    except sqlite3.Error as error:
        print(f"Error fetching data from database: {error}")
        return None
    finally:
        conn.close()

```

ДОДАТОК Б Презентація



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Кафедра Інженерії програмного забезпечення в енергетиці.



Система для визначення ідентифікаційних ознак об'єктів морського середовища

Виконав: студент магістерського рівня 2-го року навчання, групи ТВ-21мп

Мамотенко Богдан Олегович

Керівник: доцент, кандидат фізико-математичних наук

Свинчук Ольга Василівна



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Актуальність роботи

З огляду на глобальні зміни клімату, забруднення океанів, а також зростання морської навігації та досліджень, стає важливим розробити надійні методи ідентифікації та аналізу об'єктів у морському середовищі.



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Мета роботи, об'єкт та предмет дослідження

Метою даної роботи є створення системи для визначення ідентифікаційних ознак об'єктів морського середовища.

Об'єкт дослідження: ідентифікація об'єктів морського середовища.

Предмет дослідження: система для визначення ідентифікаційних ознак об'єктів морського середовища.



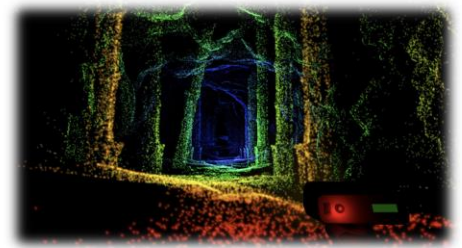
Постановка задачі

- 1) Проаналізувати існуючі програмні системи для визначення ідентифікаційних ознак об'єктів морського середовища.
- 2) Розробити перелік вимог до системи, провести моделювання та проектування програмного забезпечення.
- 3) Розробити систему, що включає наступні підзадачі:
 - Зчитування аудіосигналу.
 - Отримання ознак сигналу за допомогою методів Feature Extraction.
 - Класифікація сигналу за допомогою методів кластеризації.
 - Завантаження результату в базу даних.
- 4) Виконати тестування системи.



Існуючі аналоги

- Програмне забезпечення сонарів
- Програмне забезпечення лідарів
- GIS (Географічні Інформаційні Системи)



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

5

Архітектура системи



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

6

Засоби розробки системи



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

7

Методи для визначення ідентифікаційних ознак

В роботі були використані методи Feature Extraction за допомогою бібліотеки librosa, а саме:

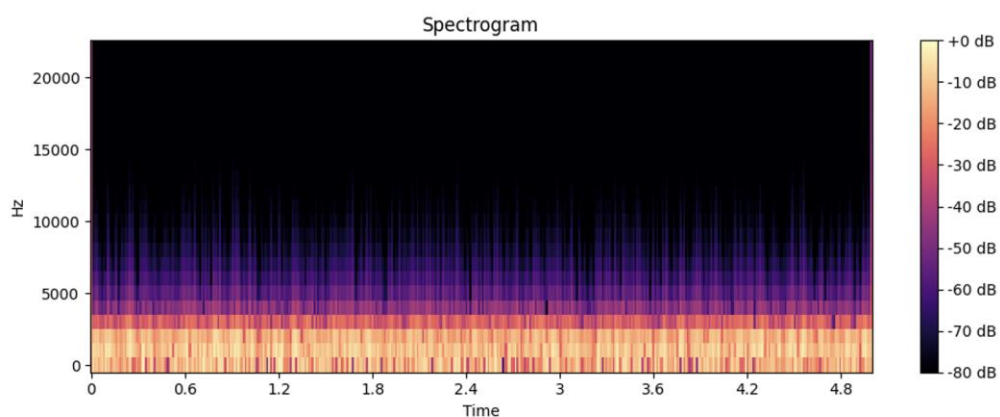
- Спектральний контраст
- Tonnetz (тоннет-мережа)
- RMS Energy (root mean square)
- Zero Crossing Rate
- Melspectrogram



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

8

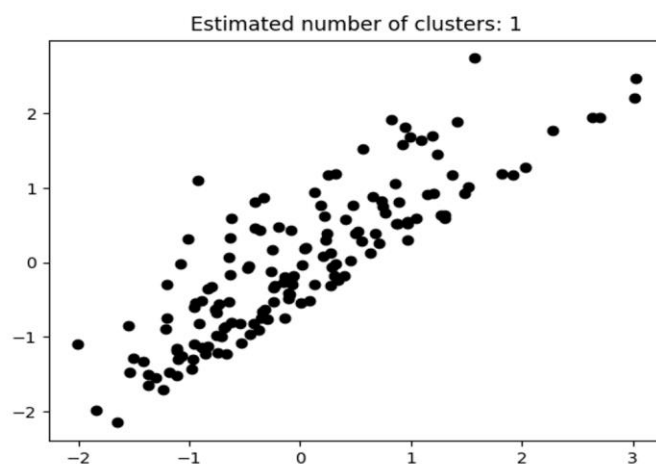
Результат роботи спектрограми



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

9

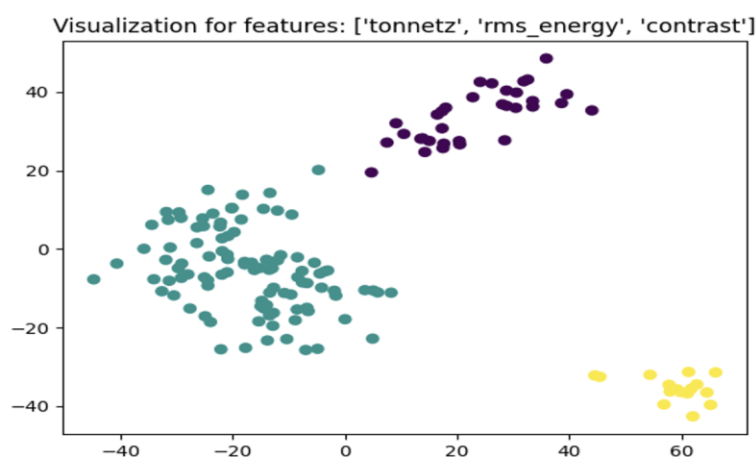
Результат роботи кластеризації DBSCAN



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

10

Результат роботи кластеризації K-means



Результат класифікації в БД

id	filename	spect...	feature...	stats_data	shape	classification_type
36	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1
2	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
11	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1
42	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
16	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1
27	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1
40	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
20	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1
10	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1
50	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
22	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
38	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
19	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1
26	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1
24	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1
23	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
30	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1
13	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
44	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
9	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
12	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type0
17	/Users/b...	BLOB	("mfccs":...	("mean": ...	513x431	type1



Висновки

- 1) Проаналізовано існуючі програмні системи ідентифікації ознак об'єктів морського середовища.
- 2) Розроблено перелік вимог до системи, проведено моделювання та проектування програмного забезпечення.
- 3) Реалізовано систему, що включає наступні підзадачі:
 - Зчитування аудіосигналу.
 - Отримання ознак сигналу за допомогою методів Feature Extraction.
 - Класифікація сигналу за допомогою методів кластеризації.
 - Завантаження результату в базу даних.
- 4) Виконано тестування системи.

