Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка та дослідження моделей машинного навчання для аудіоаналізу в інтелектуальних інформаційних системах
(тема)

Виконав:
здобувач другого року навчання,
групи ДСм-24-1
Тихонов І.О.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Науки про дані (Data Science)
(повна назва спеціалізації)
Керівник доц. Магдаліна І.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Лариса ЧАЛА
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Науки про дані (Data Science)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Тихонову Івану Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та дослідження моделей машинного навчання для аудіоаналізу в інтелектуальних інформаційних системах

затверджена наказом університету від 24 листопада 20 25 р. № 1057Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18 грудня 20 25 р.

3. Вихідні дані до роботи Розробити програмний інструмент для аналізу та класифікації музичних аудіосигналів за жанрами на основі методів машинного навчання та глибинних нейронних мереж. Операційна система Windows ОС (середовище розробки та тестування). Програмне забезпечення та інструменти: мова програмування Python; середовище розробки Visual Studio Code / PyCharm; бібліотеки машинного навчання та аналізу даних scikit-learn, NumPy, Pandas; бібліотеки для обробки аудіосигналів Librosa; фреймворк глибинного навчання TensorFlow / Keras; інструменти візуалізації Matplotlib та Seaborn; датасет Free Music Archive (FMA) для навчання та тестування моделей.

4. Перелік питань, що потрібно опрацювати в роботі

1) Аналіз предметної галузі

2) Розробка вимог до розроблювальної моделі

3) Опис прийнятих проектних рішень

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	24.11.2025	виконано
2	Аналіз застосування інтелектуальних систем в	29.11.2025	виконано
	обробці аудіо		
3	Аналіз аналогічних систем та визначення сфери	02.12.2025	виконано
	застосування		
4	Постановка задачі	03.12.2025	виконано
5	Розробка системних та функціональних вимог до	05.12.2025	виконано
	моделей		
6	Опис і обґрунтування вибору алгоритмів	06.12.2025	виконано
	машинного навчання		
7	Опис архітектури розробленої системи та	08.12.2025	виконано
	програмних засобів		
8	Підготовка та формування датасету аудіо для	09.12.2025	виконано
	аналізу		
9	Розробка алгоритму попередньої обробки	11.12.2025	виконано
	аудіосигналів і вилучення ознак		
10	Навчання та налаштування моделей машинного	12.12.2025	виконано
	навчання		
11	Розробка та навчання згорткової нейронної	12.12.2025	виконано
	мережі CNN		
12	Тестування моделей та аналіз результатів	13.12.2025	виконано
	класифікації		
13	Використання моделей для аналізу реальних	13.12.2025	виконано
	аудіофайлів		
14	Формування висновків та узагальнення	14.12.2025	виконано
	результатів дослідження		
15	Підготовка пояснювальної записки та подання	14.12.2025	виконано
	роботи в ЕК		

Дата видачі завдання 24 листопада 20 25 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Магдаліна І.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 73 с., 26 рис., 3 табл., 1 дод., 12 джерел.

АУДІОАНАЛІЗ, ГЛИБИННЕ НАВЧАННЯ, ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ, КЛАСИФІКАЦІЯ, МАШИННЕ НАВЧАННЯ, МОДЕЛЮВАННЯ ДАНИХ, НЕЙРОННІ МЕРЕЖІ, ОБРОБКА СИГНАЛІВ, ПЕРЕДОБРОБКА АУДІО, ПРОГНОЗУВАННЯ ЖАНРІВ, СПЕКТРОГРАМА.

Об’єкт дослідження – моделі обробки аудіосигналів в інтелектуальних інформаційних системах, що використовуються для аналізу, інтерпретації та класифікації аудіоданих.

Предмет дослідження – алгоритми та методи автоматичної обробки аудіо, зокрема витягнення ознак, моделі машинного і глибинного навчання, а також підходи до оцінювання їх ефективності.

Мета роботи – розробка програмного рішення для підготовки аудіоданих, побудови й навчання моделей машинного навчання, аналізу результатів та визначення ефективних підходів для застосування в інтелектуальних інформаційних системах.

Методи дослідження – методи машинного та глибинного навчання, цифрової обробки сигналів, алгоритми класифікації, методи виділення ознак, нейронні мережі, статистичні методи оцінювання точності.

Результатом роботи є програмний прототип системи аудіоаналізу, що реалізує обробку аудіосигналів, витягнення ознак і класифікацію аудіоданих для автоматизованого визначення характеристик аудіофайлів.

Галузь застосування – інтелектуальні інформаційні системи автоматизованого аудіоаналізу, зокрема рекомендаційні та стрімінгові сервіси, мультимедійні платформи та системи моніторингу звукового середовища.

ABSTRACT

Master’s thesis contains: 73 pp., 26 fig., 3 tabl., 1 ann., 12 references.

AUDIO ANALYSIS, AUDIO PREPROCESSING, CLASSIFICATION, DATA MODELING, DEEP LEARNING, GENRE PREDICTION, INTELLIGENT SYSTEMS, MACHINE LEARNING, NEURAL NETWORKS, SIGNAL PROCESSING, SPECTRAGRAM.

The object of research is audio signal processing models in intelligent information systems used for analysis, interpretation, and classification of audio data.

The subject of research is algorithms and methods of automatic audio processing, in particular feature extraction, machine learning and deep learning models, as well as approaches to evaluating their effectiveness.

The purpose of the work is to develop a software solution for preparing audio data, building and training machine learning models, analyzing results, and determining effective approaches for use in intelligent information systems.

Research methods include machine and deep learning methods, digital signal processing, classification algorithms, feature extraction methods, neural networks, and statistical methods for accuracy assessment.

The result of the work is a software prototype of an audio analysis system that implements audio signal processing, feature extraction, and audio data classification for automated determination of audio file characteristics.

The area of application is intelligent information systems for automated audio analysis, in particular recommendation and streaming services, multimedia platforms, and sound environment monitoring systems.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence – штучний інтелект;

CNN – Convolutional Neural Network – згорткова нейронна мережа, що використовується для обробки зображень, зокрема спектрограм;

FFT – Fast Fourier Transform – швидке перетворення Фур'є, алгоритм отримання спектру сигналу;

LR – Learning Rate – коефіцієнт швидкості навчання моделі;

LSTM – Long Short-Term Memory – тип рекурентної нейронної мережі для роботи з послідовностями;

MFCC – Mel-Frequency Cepstral Coefficients – мел-частотні кепстральні коефіцієнти, популярний набір аудіофічерів;

ML – Machine Learning – машинне навчання, сукупність методів побудови моделей на основі даних;

SNR – Signal-to-Noise Ratio – співвідношення сигнал/шум;

Spectrogram – спектрограма – зображення частотної структури сигналу в часі, побудоване на основі короткочасного спектрального аналізу;

SR – Sampling Rate – частота дискретизації, кількість вимірів сигналу за секунду;

STFT – Short-Time Fourier Transform – короткочасне перетворення Фур'є, основа побудови спектрограм;

TPU/GPU – Tensor/Graphics Processing Unit – прискорювачі обчислень для навчання нейронних мереж.

ВСТУП

За останні роки розвиток інтелектуальних інформаційних систем набув особливої динаміки, відкриваючи широкі можливості для автоматизації процесів, що раніше вимагали значних людських зусиль. Одним із ключових напрямів цього поступу став аналіз аудіо, у якій машинне навчання дозволяє вирішувати складні завдання, починаючи від розпізнавання мови й музичної класифікації до виявлення аномалій у технічних системах. Аудіоаналітика вже широко застосовується у промисловості, медіа, сервісах потокової музики, охоронних системах та інших сучасних технологічних екосистемах, проте значний потенціал залишається нереалізованим через складність обробки звуку та потребу в точних моделях.

У світлі цих тенденцій розробка та дослідження моделей машинного навчання для аудіоаналізу є актуальною та важливою задачею. Стрімке зростання обсягів аудіоданих, підвищення вимог до швидкості обробки та необхідність автоматичного витягнення корисної інформації з сигналів стимулюють пошук нових підходів і рішень. Особливої актуальності набуває створення таких моделей, які здатні працювати на різних типах даних: мовних, музичних, технічних, забезпечуючи високу точність розпізнавання за умов шуму, варіативності джерел та різноманіття жанрів чи звукових подій.

Аудіоаналіз є затребуваним, перш за все, у тих галузях, де потрібна автоматична інтерпретація або класифікація звукових подій. Це можуть бути інтелектуальні мультимедійні сервіси, голосові асистенти, системи моніторингу стану обладнання, музичні рекомендаційні платформи та інші рішення, що працюють з великими потоками даних у реальному часі. В таких середовищах точність моделі та швидкість обчислень безпосередньо впливають на ефективність системи та якість користувацького досвіду.

Актуальність теми також підсилюється потребою у вдосконалених підходах до обробки спектральних та часових характеристик звуку. Методи традиційного аналізу часто не справляються зі складною природою аудіосигналів, а використання нейронних мереж, у тому числі згорткових та рекурентних архітектур, відкриває нові можливості для підвищення точності та стійкості моделей. Це дозволяє будувати ефективні системи, здатні не лише класифікувати звук, а й виявляти закономірності, що раніше були недоступними.

Результати роботи в цьому напрямі можуть бути застосовані в інтелектуальних системах різного призначення, від комерційних аудіосервісів до автоматизованих технічних комплексів, де потрібен аналіз акустичних сигналів. Сучасні тенденції цифровізації, збільшення ролі штучного інтелекту та зростання попиту на інтерактивні аудіосервіси створюють сприятливі умови для впровадження таких рішень, що забезпечують ефективність, точність і масштабованість аудіоаналізу в реальних умовах.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз застосування інтелектуальних систем в обробці аудіо

Аудіоаналіз є складовою великої галузі обробки цифрових сигналів, що охоплює методи перетворення, аналізу та класифікації звукових даних. Його основною метою є вилучення з аудіосигналу інформативних характеристик, які можна використовувати для автоматизованої інтерпретації. Такі процеси відіграють важливу роль у сучасних інтелектуальних інформаційних системах, що працюють з великими масивами аудіоданих, починаючи від музичних сервісів до охоронних платформ та промислового моніторингу.

Аудіосигнали мають складну та багатовимірну структуру. На відміну від текстових або табличних даних, звук характеризується одночасною зміною частоти, амплітуди та спектральних компонентів у часі. Для отримання корисних ознак використовуються методи цифрового аналізу сигналів, зокрема перетворення Фур'є, мел-спектрограми, MFCC-коефіцієнти та інші підходи, які дозволяють перетворити сигнал у форму, придатну для подальшої обробки моделями машинного навчання.

У предметній галузі обробки аудіо також важливу роль відіграє розподіл задач на категорії. Існують декілька типових сценаріїв: розпізнавання мови, класифікація музичних жанрів, виявлення звукових подій, діагностика стану технічних систем за шумом, виявлення аномалій, а також побудова рекомендаційних алгоритмів на основі вмісту. Кожен із цих напрямів має власні виклики, пов’язані з варіативністю джерел звуку, наявністю шумів, відмінностями у тривалості та структурі сигналів.

На відміну від традиційних методів аналізу звуку, які покладалися переважно на ручне виділення ознак, сучасні інтелектуальні системи орієнтуються на застосування моделей машинного навчання та глибинного навчання. Нейронні мережі дозволяють автоматизувати витягнення ознак,

обробляти спектрограми як зображення, враховувати часові послідовності та ефективно узагальнювати закономірності в великих обсягах даних.

Процеси аналізу аудіо в інтелектуальних системах умовно можна поділити на наступні етапи:

– збирання даних аудіо за допомогою різних пристроїв, зокрема це можуть бути мікрофони, музичні записи, поточний запис або вже сформовані датасети;

– попередня обробка, що складається з нормалізації, фільтрації шумів, обрізання чи сегментації сигналу;

– перетворення аудіо в репрезентативну форму, наприклад, побудова спектрограм, MFCC, хрома-ознаків тощо;

– навчання моделей за використанням класичних алгоритмів (Random Forest, SVM, MLP) або нейронних мереж (CNN, RNN, CRNN);

– класифікація або оцінка сигналів, визначення класу, жанру, події чи стану об’єкта за аудіоданими;

– тестування точності, стійкості та узагальнювальної здатності моделей;

– використання отриманих результатів для прийняття рішень, рекомендацій або автоматизованого моніторингу.

Предметна галузь охоплює широкий спектр процесів, пов’язаних зі зчитуванням, обробкою та інтерпретацією аудіосигналів. Аналіз цієї сфери дозволяє визначити ключові вимоги до системи, функціональні можливості майбутнього інструмента та обґрунтувати вибір тих методів і моделей, які забезпечують найвищу точність і ефективність при роботі з такими даними.

1.2 Аналіз аналогічних систем

Спираючись на те, що розроблена система працює насамперед із аналізом музичних файлів за допомогою методів машинного навчання,

доцільно порівняти її з відомими сервісами, які вирішують суміжні завдання, а саме класифікацію, виявлення характеристик треку або формування рекомендацій. Для порівняння було розглянуто чотири поширених рішення: Spotify Recommendation System [9], Last.fm Scrobbling & Recommender [1] та аудіоаналітичні функціонали стрімінгових платформ YouTube Music та SoundCloud [7] (рисунок 1.1).

Рисунок 1.1 – Логотипи систем-аналогів

Spotify використовує складну гібридну архітектуру, що поєднує контентний, поведінковий й соціальний аналіз. Контентний компонент включає базові спектральні та ритмічні характеристики треку – темп, яскравість, енергійність, валентність, що формуються у межах модуля Spotify Audio Analysis. Поведінковий рівень системи побудований на колаборативному фільтруванні й моделюванні подібності між користувачами та їхніми плейлистами. Додатково Spotify застосовує глибокі нейронні мережі для сегментації аудіо, групування користувацьких патернів і побудови рекомендацій. Основні недоліки підходу полягають у тому, що Spotify практично не працює з сирим аудіосигналом, а значна частина його моделей закрита та недоступна для досліджень. На відміну від цього, розроблена система ґрунтується саме на детальному спектральному аналізі, а не на агрегованих фічах або поведінковій статистиці.

Last.fm базується на принципі «scrobbling», тобто зборі історій прослуховування з різних платформ, формуванні тегів і застосуванні колаборативного фільтрування. Система добре працює у випадках, коли жанрова структура стійка та популярна, проте її точність значною мірою залежить від суб'єктивних тегів користувачів. Інформація може містити значний шум, а класифікація фактично не враховує сам аудіосигнал. Такий підхід суттєво обмежує можливості при роботі з рідкісними, змішаними або нечітко визначеними жанрами. Розроблена в межах цієї роботи система, навпаки, спирається на об'єктивні часово-частотні характеристики, а не на поведінкові або соціальні дані.

YouTube Music [12] формує рекомендації, поєднуючи кілька джерел інформації: історію переглядів і прослуховувань, поведінкові патерни користувача, популярність треків, соціальні та регіональні тенденції, а також загальні метадані: жанр, виконавець, альбом, рік, настрій. Для приватних даних Google реалізує масштабні графові моделі подібності між треками, виконавцями та користувачами. Попри вражаючу масштабність, YouTube Music майже не виконує глибинного аналізу аудіо як сирого сигналу. Більшість характеристик формується з метаданих або з високорівневих дескрипторів, що генеруються внутрішніми моделями Google, які працюють із великими колекціями анотацій та поведінкових даних. Справжній часово-частотний аналіз сигналу (наприклад, формування спектрограм, мел-спектрограм чи MFCC для подальшої класифікації) не є ключовим елементом їхніх рекомендаційних систем – це лише допоміжний інструмент для виявлення контенту або авторських прав.

Основною перевагою YouTube Music є надзвичайно потужна персоналізація, здатність швидко адаптуватися до змін у поведінці користувача та оперувати величезними обсягами даних. Однак ці ж фактори є недоліками у контексті порівняння з системами чистого аудіоаналізу: рекомендації YouTube Music залежать від історії взаємодії, а не від

об'єктивної акустичної природи треку; жанрова класифікація часто розмита та контекстно залежна; сам аудіосигнал не завжди визначає рішення моделі.

SoundCloud, хоча й не є класичною системою рекомендацій, має окремі модулі для аналізу аудіо, які використовуються переважно для візуального представлення треку, визначення пікових значень, виявлення тиші та загальної енергії сигналу. Сам сервіс працює з великою кількістю незалежної музики, тому інколи застосовує автоматичні системи виявлення схожості треків чи визначення контенту для боротьби з порушеннями авторських прав. Попри це, аналіз переважно обмежується низькорівневими акустичними фічами, такими як амплітудна огинаюча, енергетичний профіль та спрощені спектральні параметри. Глибинна жанрова класифікація не є основною метою SoundCloud, тому можливості системи в цьому напрямку доволі обмежені.

Отже, якщо Spotify та Last.fm фокусуються на поведінкових та соціальних моделях, а SoundCloud та Youtube Music обмежується поверхневим аналізом, то розроблена система вирізняється тим, що виконує комплексний спектральний аналіз музичних файлів і застосовує спеціалізовані моделі машинного навчання для точного визначення жанру. Це забезпечує більш об’єктивний, стійкий і незалежний від користувацьких уподобань підхід до класифікації аудіо.

1.3 Визначення сфери застосування розроблювальних моделей

Розроблювана система машинного аналізу аудіосигналів має широкий спектр потенційних напрямів застосування, що виходить далеко за межі суто академічного використання. Завдяки здатності автоматично обробляти музичні файли, виділяти спектральні ознаки та класифікувати їх відповідно до навчених моделей, система може бути інтегрована в різні інтелектуальні інформаційні продукти та сервіси.

Перш за все, її можна використовувати у музичних сервісах, що потребують автоматичного визначення жанру або інших характеристик треків для покращення каталогізації та структурування великих аудіобаз. Такі можливості особливо важливі для платформ, на яких користувачі завантажують власний контент, оскільки це дає змогу оперативно визначати жанрову належність і пропонувати релевантні рекомендації.

Система також може знайти застосування у сфері контент-модерації, де необхідно автоматично перевіряти тип аудіоматеріалу, визначати відповідність певним стандартам або виявляти потенційні порушення правил. Її використання є актуальним і в галузях цифрової музикології та дослідницьких проєктів, де потрібний інструмент для об'єктивного аналізу структури музичних композицій, виявлення патернів та оцінки подібності між треками.

Окремим напрямом застосування є розвиток рекомендаційних систем нового покоління, які доповнюють дані користувацької поведінки саме акустичними характеристиками. Сучасні алгоритми рекомендацій часто спираються на метадані та історію прослуховувань, тоді як розроблена система дозволяє покращити точність рекомендацій за рахунок глибшого розуміння аудіосигналу.

1.4 Постановка задачі

1.4.1 Призначення розробки

Багато сучасних інформаційних систем медіаконтенту потребують ефективних засобів аналізу аудіоданих для автоматизації класифікації [11], виявлення закономірностей та генерації рекомендацій на основі звукової частини. Розробка такого інструменту дозволяє досліджувати аудіосигнали за допомогою моделей машинного навчання та створювати автоматизовані процеси обробки великих обсягів музичних чи звукових файлів.

Метою розробки є створення програмного інструменту для автоматизованого аудіоаналізу, здатного виділяти ознаки звукового сигналу, класифікувати аудіо за жанрами, оцінювати подібність треків та проводити експерименти з різними моделями машинного навчання [4]. Такий інструмент дозволяє проводити дослідження в галузі аудіоаналітики та оптимізувати алгоритми для подальшого застосування в інтелектуальних системах.

Основною перевагою є автоматизація обробки аудіофайлів без участі користувача, що зменшує час на аналіз, підвищує точність результатів та дозволяє досліджувати великі колекції даних. Інструмент надає можливість експериментувати з різними алгоритмами та параметрами моделей, оцінювати їх ефективність та порівнювати результати на єдиному наборі даних.

Розроблений інструмент може бути застосований у різних напрямках: від дослідження музичних жанрів та класифікації звукових подій до підготовки даних для систем рекомендацій або подальшого використання у складніших інтелектуальних інформаційних системах.

1.4.2 Мета створення інструментів аналізу

З боку дослідника або розробника системи:

– надання інструменту для автоматизованого аудіоаналізу та класифікації музичних треків за жанрами або характеристиками звуку;

– прискорення процесу обробки великих колекцій аудіофайлів за рахунок автоматизації виділення ознак та побудови моделей машинного навчання;

– забезпечення можливості експериментів з різними алгоритмами та параметрами моделей для підвищення точності класифікації та оцінки ефективності моделей;

– збирання та підготовка структурованих даних для подальшого використання в системах рекомендацій або інших інтелектуальних інформаційних системах;

– зменшення ручної праці при аналізі аудіоконтенту та мінімізація ймовірності людських помилок під час класифікації та обробки.

З боку потенційного користувача системи (інтелектуальних інформаційних систем, дослідницьких проєктів):

– швидке та ефективне отримання характеристик аудіо та жанрової класифікації треків;

– можливість порівняння треків за різними ознаками, такими як темп, спектральні характеристики або стиль виконання;

– доступ до аналітичних даних про музичні колекції для побудови рекомендацій або проведення наукових досліджень;

– підвищення точності систем рекомендацій за рахунок використання моделей машинного навчання, натренованих на аудіофайлах;

– можливість швидкої інтеграції та масштабування інструменту для обробки великих потоків аудіоданих у різних застосунках.

1.4.3 Основні вимоги та функціонал моделей

Розроблені моделі машинного навчання для аудіоаналізу повинні забезпечувати точну та ефективну класифікацію музичних треків за жанрами та іншими характеристиками. Основні вимоги до моделей включають високу точність прогнозів, здатність працювати з великими обсягами аудіоданих, а також гнучкість у налаштуванні параметрів для оптимізації результатів.

Функціонально моделі повинні виконувати кілька ключових завдань: обробку аудіофайлів, виділення ознак звуку, таких як спектрограми, MFCC, темп, ритм та гармонійні характеристики, побудову внутрішніх представлень аудіо та навчання на підготовлених датасетах для класифікації

або прогнозування жанрів. Крім того, моделі повинні підтримувати оцінку ефективності через метрики точності, а також надавати можливість візуалізації результатів для подальшого аналізу.

Серед технічних вимог варто виділити сумісність з популярними бібліотеками та фреймворками машинного навчання, такими як PyTorch, TensorFlow [10] або scikit-learn [8], можливість обробки аудіопотоку в реальному часі, масштабованість для роботи з великими музичними колекціями та гнучкість інтеграції в інтелектуальні інформаційні системи.

Таким чином, функціонал моделей охоплює весь цикл роботи з аудіоданими: від завантаження та попередньої обробки, через навчання та оцінку, до інтеграції результатів у системи рекомендацій або дослідницькі проекти, що дозволяє ефективно вирішувати завдання класифікації та аналізу музики.

2 РОЗРОБКА ВИМОГ ДО РОЗРОБЛЮВАЛЬНОЇ МОДЕЛІ

2.1 Розробка системних вимог до системи

Для забезпечення ефективної роботи системи аналізу аудіофайлів на основі моделей машинного навчання необхідні певні системні вимоги, що гарантують стабільну обробку великих обсягів аудіоданих, тренування моделей та інтеграцію результатів у дослідницькі або аналітичні інструменти.

Для ефективного аналізу аудіофайлів за допомогою моделей машинного навчання система повинна підтримувати роботу з сучасними бібліотеками для обробки та представлення аудіоданих, такими як librosa для завантаження та аналізу сигналів, pydub для конвертації та обробки аудіоформатів, essentia-tensorflow для виділення специфічних аудіо характеристик, а також ffmpeg для роботи з різними форматами файлів.

Крім цього, система використовує бібліотеки машинного навчання tensorflow та keras для побудови, тренування та оцінки нейронних мереж, включаючи багатошарові перцептрони та згорткові моделі, а також scikit-learn для побудови класичних моделей, таких як Random Forest, SVM або Gradient Boosting, та виконання підготовки даних, нормалізації та кодування міток класів.

Для роботи з великими датасетами та обліку результатів досліджень застосовуються numpy та pandas для чисельних операцій і обробки таблиць, а matplotlib та seaborn дозволяють візуалізувати спектрограми, графіки точності та матриці помилок. Утиліти на кшталт tqdm забезпечують відстеження процесу тренування моделей, а pickle та json використовуються для збереження та завантаження проміжних даних, конфігурацій і результатів аналізу.

Важливими є достатній обсяг оперативної пам’яті та дискового простору для зберігання аудіофайлів і проміжних результатів обробки, а

також наявність процесора або графічного прискорювача (CPU/GPU) для ефективного тренування та роботи моделей. Операційна система може бути Windows, Linux або macOS, а доступ до інтернету необхідний для завантаження датасетів та бібліотек. Система повинна підтримувати основні аудіоформати, такі як WAV, MP3 та FLAC, та мати можливість обробки пакетів аудіофайлів або потокових даних для досліджень у реальному часі. Крім того, важливо забезпечити журналювання та збереження результатів аналізу для подальшої перевірки, оцінки ефективності моделей та інтеграції у додаткові аналітичні модулі. Виконання цих вимог дозволить забезпечити стабільну та гнучку роботу системи, швидке навчання моделей, ефективну обробку великих колекцій аудіоданих та точну класифікацію музичних треків за жанрами й іншими характеристиками.

2.2 Розробка функціональних вимог до системи

Система повинна забезпечувати повний цикл аналізу аудіофайлів із використанням моделей машинного навчання, включаючи завантаження та обробку музичних треків різних форматів, виділення основних характеристик аудіосигналу, таких як спектрограми, мел-спектрограми та інші аудіо-фічі, а також збереження цих даних у структурованому вигляді для подальшого навчання та оцінки моделей. Вона повинна підтримувати підготовку даних, включаючи нормалізацію сигналу, кодування міток класів, поділ на тренувальні, валідаційні та тестові набори. Система також має надавати можливість тренування різних типів моделей машинного навчання, зокрема нейронних мереж, SVM, Random Forest та інших, із використанням відповідних алгоритмів оптимізації та механізмів контролю якості, таких як рання зупинка, збереження найкращих моделей та моніторинг метрик точності, втрат та інших показників. Крім того, функціонал системи передбачає генерацію звітів та візуалізацій результатів класифікації, включаючи графіки точності на тренувальних і тестових

даних, матриці помилок та порівняння результатів для різних моделей, що дозволяє користувачу оцінити ефективність обраної архітектури. Важливим функціональним аспектом є також інтеграція із зовнішніми бібліотеками та інструментами для роботи з аудіо, що забезпечує гнучкість системи у роботі з різними форматами та розмірами аудіофайлів. Нарешті, система повинна підтримувати відтворення аудіо для візуальної та слухової перевірки оброблених сигналів, автоматизацію процесів підготовки та навчання моделей, а також можливість збереження проміжних результатів для повторного використання або подальшого аналізу.

2.2.1 Використання алгоритму Random Forest

Система повинна забезпечувати можливість класифікації аудіофайлів із використанням алгоритму Random Forest, що є ансамблевою моделлю, яка складається з великої кількості окремих дерев рішень. Ця модель дозволяє підвищити стабільність прогнозів та зменшити ризик перенавчання, що особливо важливо при роботі з музичними треками, де аудіо має високу варіативність у частотних характеристиках, темпі, тембрі та інших параметрах. Кожне дерево в ансамблі приймає рішення незалежно, використовуючи випадкову вибірку з тренувальних даних та випадковий підмножину ознак, що дозволяє моделі захоплювати різні аспекти сигналу та зменшувати кореляцію помилок між деревами.

Функціонально система повинна забезпечувати генерацію навчальних наборів дерев із заданими параметрами, такими як кількість дерев у лісі, максимальна глибина дерев, мінімальна кількість зразків у вузлі та критерій поділу. Random Forest дозволяє оцінювати важливість ознак, що є критичною функцією для аудіоаналізу, оскільки вона дає змогу визначити, які спектральні та тимчасові характеристики сигналу найбільше впливають на класифікацію жанру або настрою треку. Важливою функцією системи є можливість автоматичного збереження та завантаження навченої моделі, що

дозволяє повторне використання тренованих дерев для нових аудіофайлів без необхідності повторного навчання.

Визначення Random Forest у відносній формі можна подати як ансамбль дерев рішень $h_1(x)$, $h_2(x)$,..., $h_K(x)$, де K – кількість моделей в ансамблі. Прогноз класу для вхідного аудіосигналу x визначається більшістю голосів дерев:

$$\hat{y} = mode\{h_K(x)\}_{k=1}^{K}. \quad (2.1)$$

Тут $\hat{y}$ – прогноз класу аудіо, а $h_K(x)$ – прогноз k-го дерева. Система повинна забезпечувати обробку багатокласових задач, оскільки аудіо може належати до різних жанрів одночасно. Кожне дерево в Random Forest функціонально відповідає за визначення класу на основі підмножини ознак спектральних характеристик, таких як MFCC, Chroma, spectral contrast та tempo.

Функціональна реалізація Random Forest у системі передбачає інтеграцію з підготовкою аудіо: завантаження файлів, обчислення спектрограм, мел-спектрограм та інших ознак, а також нормалізацію цих ознак. Після цього модель здатна автоматично проводити навчання, оцінювати точність на валідаційних і тестових наборах, формувати матрицю помилок і генерувати графіки порівняння точності для різних налаштувань. Система повинна надавати можливість змінювати параметри Random Forest, такі як n_estimators, max_depth, min_samples_split та criterion, для підвищення ефективності та адаптації до різних типів аудіоданих.

Однією з ключових функцій є оцінка важливості ознак, яка дозволяє аналітику аудіо визначити, які характеристики сигналу найбільше впливають на результат класифікації. Це дозволяє розробникам системи оптимізувати попередню обробку даних та видаляти непотрібні або зайві ознаки, що зменшує час навчання моделі та підвищує точність. Для цієї мети функціонал Random Forest інтегрується з бібліотеками для обробки аудіо,

такими як librosa [5] та essentia, які забезпечують виділення спектральних та тимчасових ознак.

З точки зору функціональних вимог, Random Forest дозволяє системі: автоматично навчатися на аудіоданих, обробляти багатокласові задачі, забезпечувати оцінку важливості ознак, генерувати прогноз для нових аудіофайлів, формувати графічні звіти та інтегруватися із загальною архітектурою системи для подальшого порівняння з іншими моделями, такими як Gradient Boosting, MLP та CNN. Завдяки використанню Random Forest система здатна обробляти великі обсяги аудіоданих та забезпечувати високий рівень точності класифікації без необхідності тонкого налаштування кожного окремого дерева.

Функціональні можливості включають збереження проміжних результатів навчання, автоматичне відновлення моделей та оцінку різних параметрів, що дозволяє користувачу або досліднику швидко тестувати різні конфігурації моделей. Система повинна також надавати інструменти для візуалізації результатів, включаючи побудову графіків точності на тренувальних та тестових даних, генерацію матриць сплутування та порівняння різних варіантів Random Forest.

Таким чином, інтеграція Random Forest як основної функціональної моделі забезпечує систему можливістю ефективно виконувати класифікацію аудіо, автоматично оцінювати якість ознак, адаптуватися до різних типів аудіофайлів і надавати науково обґрунтовану інформацію для подальшого розвитку моделей та інструментів аудіоаналізу.

2.2.2 Використання алгоритму Gradient Boosting

Система повинна забезпечувати можливість використання алгоритму Gradient Boosting, який є потужним ансамблевим методом для класифікації та регресії, що поєднує слабкі прогностичні моделі, найчастіше дерева рішень, у послідовний ансамбль для підвищення точності. Основна ідея

Gradient Boosting полягає в поступовому коригуванні помилок попередніх моделей шляхом оптимізації функції втрат за допомогою градієнтного спуску. У контексті аудіоаналізу, Gradient Boosting дозволяє більш гнучко працювати з багатогранними характеристиками сигналів, такими як спектральні контрасти, мел-спектрограми, темп, тембр та інші акустичні ознаки, які відображають складність музичних треків.

Функціонально система повинна забезпечувати побудову послідовних дерев, де кожне наступне дерево націлене на мінімізацію помилок, залишених попередніми моделями. Це дозволяє ефективно працювати з аудіоданими, які часто містять шум, неповні або суперечливі характеристики, а також з даними високої розмірності. Gradient Boosting забезпечує високу точність класифікації, що особливо важливо при визначенні музичних жанрів або інших категорій аудіо, де точність прогнозу є критичною для дослідницьких та практичних завдань.

Відносне визначення Gradient Boosting можна подати формулою:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x), \tag{2.2}$$

де $F_m(x)$ – прогноз ансамблю на m-му етапі;

$F_{m-1}(x)$ – прогноз попереднього ансамблю;

$h_m(x)$ – слабка модель на поточному етапі;

γ_m – коефіцієнт навчання, що визначає внесок поточного дерева в загальний прогноз. Ця функція дозволяє системі послідовно коригувати результати, покращуючи точність класифікації для кожного аудіотреку. Gradient Boosting здатний працювати з багатокласовими задачами, що необхідно для класифікації музики за жанрами, настроєм, емоційним забарвленням та іншими категоріями.

Функціональні вимоги до інтеграції Gradient Boosting у систему включають можливість автоматичного навчання на підготовлених аудіоданих, зокрема спектрограмах, мел-спектрограмах, MFCC та інших

виділених аудіо-фічах. Система повинна забезпечувати розбиття даних на тренувальні, валідаційні та тестові набори, нормалізацію ознак та кодування міток класів. Gradient Boosting дозволяє системі оптимально працювати із складними розподілами ознак та забезпечує високу стабільність прогнозів, навіть при наявності незначних артефактів у аудіо.

Для ефективної роботи моделі система повинна підтримувати налаштування ключових параметрів, таких як кількість дерев, глибина дерев, коефіцієнт навчання, мінімальна кількість зразків у вузлі та вибір функції втрат (наприклад, log-loss для класифікації). Ці параметри дозволяють адаптувати модель під специфіку аудіоданих, забезпечити баланс між точністю та швидкістю навчання, а також запобігти перенавчанню на невеликих наборах треків.

Gradient Boosting у системі виконує також функцію оцінки важливості ознак, що дозволяє виділяти ключові аудіо-фічі для класифікації жанрів, настрою або темпу треків. Ця функціональна можливість інтегрується з бібліотеками librosa, essentia та іншими інструментами обробки аудіо, що дозволяє автоматично обчислювати спектральні ознаки, їх нормалізувати та передавати на вхід моделі. Система повинна надавати функції збереження проміжних результатів навчання та можливість відновлення навченої моделі для подальшого використання або аналізу нових аудіофайлів.

З точки зору функціональних вимог, Gradient Boosting забезпечує: підвищену точність класифікації в порівнянні з окремими деревами рішень, можливість роботи з багатокласовими та багатовимірними задачами, оцінку важливості ознак, інтеграцію з підготовкою та обробкою аудіоданих, автоматичне тренування та збереження моделей, а також генерацію графіків точності, матриць сплутування та порівняння моделей.

Функціональна реалізація системи передбачає автоматичне налаштування та тестування різних конфігурацій Gradient Boosting, що дозволяє оптимізувати роботу моделі для різних наборів аудіотреків. Система також повинна забезпечувати збереження результатів тренування

у структурованому форматі для подальшого порівняння з іншими моделями, такими як Random Forest, MLP та CNN, що дозволяє досліднику обирати найбільш ефективну архітектуру для конкретної задачі аудіоаналізу.

Для візуалізації результатів роботи Gradient Boosting система інтегрує побудову графіків точності на тренувальних та тестових наборах, відображення матриць помилок та порівняння результатів із попередніми моделями. Це дозволяє користувачу оцінити ефективність налаштувань, виявити слабкі та сильні сторони моделі та адаптувати її під конкретний тип аудіоданих. Система також повинна підтримувати автоматичне масштабування обробки аудіо та паралельне навчання дерев для прискорення процесу тренування на великих датасетах.

Загалом інтеграція Gradient Boosting у функціональні вимоги системи забезпечує можливість ефективного навчання, прогнозування та оцінки моделей на основі спектральних та тимчасових характеристик аудіо, підтримку багатокласових задач, автоматичне керування параметрами моделей та генерацію графічних звітів для аналітики. Алгоритм Gradient Boosting у системі стає ключовим інструментом підвищення точності класифікації та оптимізації процесів аудіоаналізу у рамках розроблюваної інтелектуальної інформаційної системи.

2.2.3 Використання алгоритму MLP

Модель багатошарового персептрону, або MLP, у контексті функціональних вимог системи класифікації аудіосигналів повинна бути реалізована як повністю підключена штучна нейронна мережа, що складається принаймні з вхідного шару, одного або кількох прихованих шарів і вихідного шару з кількістю нейронів, відповідною числу цільових класів задачі. Система повинна забезпечувати можливість довільного налаштування конфігурації MLP, включаючи вибір кількості нейронів у кожному шарі, типу активаційних функцій, методу ініціалізації ваг та

параметрів оптимізації. На функціональному рівні система повинна дозволяти генерувати вхідний вектор ознак на базі аудіофіч, таких як MFCC, мел-спектрограми чи спектральна енергія, після чого ці вектори подаються як вхід у MLP, забезпечуючи можливість моделювання нелінійних залежностей між характеристиками аудіосигналу та їх належністю до певного класу. Формально MLP може бути описаний як композиція перетворень вигляду:

$$y = f(W_n f(W_{n-1} \dots f(W_{1x} + b_1) + b_{n-1}) + b_n, \quad (2.3)$$

де W_i та b_i – параметри відповідних шарів;

$f(\cdot)$ – нелінійна активаційна функція.

Система може підтримувати використання різних типів активаційних функцій, таких як ReLU, sigmoid, tanh або їх модифікацій, а також забезпечувати коректну обробку похідних цих функцій під час процесу оптимізації ваг.

Функціональні вимоги передбачають, що під час навчання MLP система має реалізовувати механізм зворотного поширення помилки з автоматичним обчисленням градієнтів, що дозволяє налаштовувати ваги моделі відповідно до метрик втрат, які визначають якість класифікації аудіофрагментів. Механізм оптимізації повинен реалізовувати алгоритми, такі як SGD, Adam або RMSProp, із можливістю вибору параметрів швидкості навчання, моменту, регуляризації та інших гіперпараметрів, необхідних для стабільного та ефективного процесу навчання. У цьому контексті система повинна забезпечувати можливість автоматичної або ручної зупинки тренування моделі, використовуючи критерії ранньої зупинки, контроль над перенавчанням, а також автоматичне збереження найкращих моделей, що демонструють найнижче значення функції втрат або найвищу точність класифікації на валідаційних даних.

Крім того, функціональні вимоги включають підтримку регуляризаційних методів, таких як L2-регуляризація, дропаути та обмеження ваг, що необхідно для запобігання перенавчанню моделі у випадку, коли набір аудіоданих є обмеженим або має високу внутрішню корельованість. На рівні реалізації система повинна забезпечувати можливість вибору коефіцієнтів регуляризації, частки відключення нейронів у шарах дропаутів, а також алгоритмів контролю складності моделі для забезпечення узагальнюючої здатності MLP.

Функціональні вимоги також передбачають можливість обробки MLP-моделлю великих масивів ознак, що генеруються під час екстракції аудіо, тому система повинна підтримувати пакетний режим тренування, нормалізацію входів, стандартизацію або мін-макс скейлінг для стабілізації навчання. У цьому контексті система має забезпечувати роботу MLP із вхідними векторами високої розмірності, включаючи підтримку операцій матричного множення та прискорення обчислень за допомогою апаратного прискорення, такого як GPU або TPU, залежно від обраної конфігурації розгортання.

Крім тренування, система повинна забезпечувати процес інференсу MLP у режимі реального часу або пакетної класифікації, що включає завантаження збережених параметрів моделі, виконання прямого проходу крізь мережу та отримання ймовірнісних виходів. Система має забезпечувати можливість конфігурації способу інтерпретації цих виходів, включаючи вибір максимального значення ймовірності, порогову класифікацію або зважені рішення, що може бути корисним у задачах із незбалансованими класами або у випадках, коли моделі MLP використовуються як частина більш комплексного ансамблю.

Таким чином, у межах функціональних вимог MLP повинен бути інтегрований у загальну архітектуру системи класифікації аудіосигналів як модель, що забезпечує здатність до навчання на складних багатовимірних ознаках, автоматизацію процесу тренування та оптимізації, підтримку

широкого спектра гіперпараметрів, а також забезпечення повної сумісності з іншими компонентами, що використовуються для обробки аудіо, збереження даних, оцінки якості та формування прогнозів.

2.2.4 Використання алгоритму SVM

Метод опорних векторів (Support Vector Machine, SVM) у межах функціональних вимог системи машинного аналізу аудіо виконує роль високоточного класифікаційного інструмента, здатного працювати з даними високої розмірності та забезпечувати стійкі результати за умов обмежених обсягів вибірки. У контексті розроблюваної системи SVM використовується як один із ключових механізмів класифікації спектральних та мел-спектральних ознак аудіосигналу, забезпечуючи альтернативний підхід до моделювання порівняно з ансамблевими методами та нейронними мережами. Функціонально така модель повинна підтримувати побудову оптимальної роздільної гіперплощини між класами на основі максимізації відстані до найближчих навчальних прикладів, які називаються опорними векторами. Цей принцип реалізується через оптимізаційну задачу мінімізації функціоналу втрат із використанням штрафного коефіцієнта за помилки класифікації. У загальному випадку така задача записується у вигляді мінімізації функції:

$$\frac{1}{2}||w||^2 + C\sum_{i+1}^{n}\xi_i\,, \tag{2.4}$$

за умов

$$y_i(w\ \cdot\ x_i + b) \geq 1 - \xi_i, \xi_i\ \geq 0. \tag{2.5}$$

У системі цей механізм реалізується через параметр C, який визначає компроміс між шириною маржі та кількістю допущених помилок на

тренувальному наборі. Високе значення C робить модель жорсткішою та схильною до переобучення, тоді як менше значення призводить до ширшої маржі та вищої здатності до узагальнення.

Функціональна підтримка ядрових методів є ключовою вимогою для коректної роботи моделі SVM у задачах аудіоаналізу. Через те, що ознаки після обчислення спектрограм, MFCC або мел-спектрограм можуть мати складну нелінійну структуру, система повинна забезпечувати використання ядрової функції радіально-базисного типу (RBF), яка дозволяє відображати дані у простір вищої розмірності без явного обчислення цього перетворення. Відповідно, функціонально модель визначає ядро як

$$K(x_i, x_j) = \exp\left(-\gamma \left| x_i - x_j \right|^2\right), \tag{2.6}$$

де параметр γ контролює масштаб впливу окремих навчальних точок: великі значення γ призводять до високої чутливості моделі до локальних змін, тоді як менші значення забезпечують більш плавну та узагальнену поведінку класифікатора. Для аудіо це дозволяє адаптувати модель до варіацій у спектральних структурах треків, забезпечуючи баланс між точністю й узагальненням.

Функціонально важливим елементом моделі є підтримка обчислення ймовірнісних передбачень, які використовуються системою для подальшої побудови метрик, аналізу впевненості класифікації та формування порівняльних звітів між різними моделями. Хоча класичний SVM не є ймовірнісним методом, система повинна забезпечувати наявність механізму калібрування ймовірностей, що реалізується засобами Platt scaling. Це особливо важливо в умовах багатокласової класифікації аудіо, коли потрібно оцінити не тільки остаточне рішення моделі, але й ступінь упевненості у ньому. У реалізації системи така можливість забезпечується вибором параметра probability=True, який запускає додатковий внутрішній процес калібрування.

У межах функціональних вимог передбачено, що система повинна забезпечувати роботу методу SVM навіть на великих масивах аудіоданих, однак слід враховувати ресурсомісткість алгоритму. Тому модель повинна підтримувати адаптивне зменшення вибірки у випадках, коли кількість аудіофіч сягає десятків тисяч. У реалізації це забезпечено автоматичним обрізанням вибірки до 5000 прикладів у разі надмірно великих тренувальних даних, що дозволяє зберегти стабільність виконання та контрольовану тривалість тренування без втрати загальної функціональності системи. Це узгоджується з концепцією гнучкого управління ресурсами, яка закладена у функціональні вимоги системи обробки аудіоданих.

Модель також повинна функціонально інтегруватися у загальний процес тренування з можливістю обчислення стандартних метрик класифікації, таких як точність, матриця помилок та деталізований класифікаційний звіт. Це дозволяє системі забезпечувати повноцінний моніторинг продуктивності SVM у контексті порівняння з іншими моделями, включаючи Random Forest, Gradient Boosting, MLP та CNN. Функціонально система має формувати структуроване збереження результатів роботи моделі, включно з прогнозами, ймовірностями, часом тренування та згенерованими метриками, що дозволяє виконувати подальший аналіз та повторне використання моделі.

Окремою вимогою є можливість збереження параметрів та стану моделі у форматі, сумісному з іншими компонентами системи. У контексті реалізованої системи така функціональність забезпечується механізмом серіалізації через pickle, що дозволяє інтегрувати SVM у повний цикл обробки, тестування та експлуатації моделей. Завдяки цьому модель опорних векторів стає повністю керованою частиною системи аудіоаналізу, забезпечуючи високоточний і водночас контрольований підхід до класифікації аудіофіч.

Узгодження ядрових методів, можливості керування складністю моделі, підтримки ймовірнісних передбачень та сумісності з підсистемами візуалізації результатів створює повний набір функціональних властивостей, необхідних для застосування SVM у задачах аудіокласифікації. Внаслідок цього метод опорних векторів стає важливим компонентом розроблювальної системи машинного аналізу аудіо, забезпечуючи гнучкість і точність, необхідні для якісної роботи з високовимірними й нерівномірними аудіоданими.

2.2.5 Використання алгоритму CNN

CNN, або згорткова нейронна мережа – це спеціалізований тип нейромереж для роботи з даними, що мають сітчасту структуру, в першу чергу з зображеннями та відео, який дозволяє комп'ютерам «бачити» і розуміти візуальні образи, розпізнаючи патерни та об'єкти завдяки шарам згортків, аналогічно роботі зорової кори головного мозку.

Згорткові нейронні мережі у межах системи виконують критично важливу функцію обробки аудіосигналів у формі спектрограм і мел-спектрограм, забезпечуючи автоматичне вилучення структурних, частотних і просторових ознак, які недоступні для традиційних машинних моделей. Оскільки аудіосигнал у вигляді спектрограми стає по суті зображенням, CNN є незамінним технологічним компонентом для забезпечення точності, масштабованості та стійкості класифікації музичних треків. Система повинна підтримувати повний цикл роботи з CNN, починаючи від перетворення аудіо у двовимірні масиви даних та закінчуючи тренуванням, оптимізацією, збереженням та відтворенням моделей, здатних обробляти великі колекції музичних фрагментів. На відміну від моделей класичного типу, таких як SVM чи Random Forest, які працюють з векторизованими даними і потребують ручного конструювання ознак, CNN не вимагають від користувача визначення того, які саме структури в сигналі є важливими,

оскільки вони автоматично навчаються виявляти закономірності в частотно-часовому просторі. Саме тому система повинна забезпечувати можливість роботи CNN з багатковимірними вхідними даними, зокрема з декількома каналами спектрограм, які можуть включати основний спектр, його нормалізовану версію, похідні чи розширені спектральні ознаки.

Фундаментальною операцією CNN є згортка, яка виконує локальне сканування вхідного простору ознак. Система повинна підтримувати її виконання у форматі:

$$(X * W)(i,j) = \sum_m \sum_n X(i-m, j-n) W(m,n). \quad (2.7)$$

Ця операція забезпечує базову здатність моделі виявляти патерни різних масштабів, короткі спектральні імпульси, гармоніки, переходи, шумові характеристики і формувати стійкі ознаки для класифікації. Використання CNN у системі повинно забезпечувати інваріантність до невеликих варіацій у часі або частоті, що є важливим для аудіосигналів, де зміщення сигналу на декілька мілісекунд або незначні коливання тембру не повинні впливати на результат класифікації. Завдяки цьому CNN є набагато ефективнішими для аналізу звуку, ніж моделі, побудовані на ручних ознаках.

Система повинна дозволяти налаштовувати ключові параметри згорткових шарів, включаючи кількість фільтрів, їхній розмір, типи наповнення, кроку та розширення. Завдяки цьому стає можливим контролювати глибину та складність моделі відповідно до розміру даних і доступних обчислювальних ресурсів. CNN не лише вилучають ознаки, а й комбінують їх у глибших шарах, що дозволяє формувати ієрархію ознак: від простих частотних контурів до глобальних структур спектрограми, які найбільш корисні для моделювання жанрових особливостей музичних треків.

Додатковим функціональним компонентом CNN є pooling – операція субдискретизації, яка дозволяє зменшити розмірність проміжних представлень і водночас зберегти найбільш значущу інформацію. Система повинна забезпечувати підтримку щонайменше таких операцій, як max pooling і average pooling, що дозволяє гнучко налаштовувати баланс між компресією даних та втратою інформації. Операція max pooling формально описується як:

$$Y(i,j) = max_{m,n} X(i+m, j+n). \quad (2.8)$$

Ця функціональна можливість є ключовою, оскільки дозволяє моделі витягувати найсильніші ознаки в локальному вікні спектрограми, що підвищує стабільність класифікації у випадках складних і заплутаних аудіопотоків.

Система повинна також підтримувати механізми регуляризації, оскільки CNN мають багато параметрів і піддаються ризику перенавчання на обмежених наборах даних. Важливим компонентом є dropout – випадкова деактивація нейронів під час навчання, що забезпечує розподіл відповідальності між параметрами моделі:

$$y = x \cdot m, \quad (2.9)$$

де m – маска випадкового занулення.

Цей механізм дає змогу системі забезпечити більш стійке навчання CNN-моделей на аудіоданих, особливо у випадках невеликих або нерівномірних датасетів.

Особливістю роботи CNN у системі є їхня здатність масштабуватись за розміром архітектури. Система повинна підтримувати як компактні моделі, орієнтовані на швидку обробку та демонстраційні експерименти, так і великі, багатошарові мережі, що включають десятки згорткових блоків і

здатні навчатися на великих наборах даних. Завдяки своїй структурі CNN забезпечують високу стійкість до шуму в аудіо, а також здатність виявляти рідкісні або тонкі спектральні патерни, що робить їх інноваційним рішенням порівняно з класичними методами.

Для задач класифікації система повинна підтримувати додавання повнозв'язних шарів після згорткових блоків. Саме ці шари відповідають за остаточне ухвалення рішення щодо належності аудіофайлу до певного жанру або класу. Вихідний шар використовує стандартну функцію softmax нормалізації ймовірностей у багатокласовій класифікації:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_i}}. \tag{2.10}$$

Система має забезпечувати можливість отримання детальних прогнозів та візуального аналізу розподілу ймовірностей по класах.

CNN є ключовим доповненням до інших моделей, таких як Random Forest, Gradient Boosting, SVM чи MLP, оскільки надають спосіб автоматичного вилучення ознак, який радикально скорочує потребу у ручній інженерії. Цей підхід дозволяє комбінувати CNN із класичними моделями, наприклад, використовуючи вектор ознак, сформований CNN, як вхід у традиційний алгоритм класифікації. Така модульність системи є важливою функціональною вимогою, оскільки забезпечує гнучкість і дозволяє оцінювати різні архітектури на спільному наборі даних. Інноваційність CNN у контексті аудіоаналізу полягає в тому, що вони перетворили підхід до задач класифікації звуку: замість перетворення аудіосигналу у набори статистичних параметрів CNN працюють безпосередньо з візуальними представленнями звуку, що відкриває можливість виявлення складних ідей, таких як ритмічні структури, тембральні патерни, форма хвилі, спектральні переходи та стійкі ознаки

жанру. Це дозволяє суттєво підвищувати точність класифікації та забезпечує стабільність моделі навіть у випадках неоднорідних даних.

Система повинна реалізовувати також підтримку автоматичної оптимізації CNN шляхом використання сучасних оптимізаторів, таких як Adam або SGD. Це забезпечує стабільне навчання і можливість працювати з великими наборами аудіоданих.

Завдяки своїй архітектурі CNN є одним з найсучасніших і технологічно ефективних методів аналізу аудіо, що дозволяє доповнювати і підсилювати інші моделі, забезпечуючи системі більш високу точність, стійкість та гнучкість.

3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис архітектури розробленої системи

Розроблена система орієнтована на створення універсального програмного інструменту для аналізу аудіоданих із використанням методів машинного навчання, який може бути інтегрований до складу інтелектуальних інформаційних систем різного призначення [6]. Тому архітектура системи має модульний та інструментально-орієнтований характер і реалізована у вигляді програмного коду з використанням спеціалізованих бібліотек для обробки аудіо та побудови моделей машинного навчання.

Архітектурно система складається з логічно відокремлених компонентів, кожен з яких відповідає за окремий етап обробки даних. На вхід системи подаються аудіофайли різних форматів, які обробляються модулем завантаження та попередньої обробки аудіо. На цьому етапі здійснюється приведення сигналу до єдиної частоти дискретизації, моноформату та заданої тривалості, що забезпечує уніфікацію даних і коректну подальшу обробку.

Наступним архітектурним рівнем є модуль виділення ознак, який відповідає за перетворення аудіосигналу у числове подання, придатне для машинного навчання. У межах цього модуля формуються спектрограми та мел-спектрограми, а також обчислюються інші аудіо-ознаки, що відображають часово-частотну структуру сигналу. Результати цього етапу можуть зберігатися у вигляді структурованих файлів, що дозволяє повторно використовувати підготовлені дані без необхідності повторної обробки аудіо.

Центральним елементом архітектури є модуль машинного навчання, який реалізує навчання та оцінювання різних типів моделей, зокрема Random Forest, Gradient Boosting, MLP та згорткових нейронних мереж.

Кожна модель реалізована у вигляді окремого програмного компонента з чітко визначеними методами навчання, передбачення та оцінювання результатів. Такий підхід забезпечує гнучкість архітектури та можливість порівняння різних алгоритмів у межах єдиного інструменту.

Окремим рівнем архітектури виступає модуль оцінювання та аналізу результатів, який забезпечує обчислення метрик точності, формування матриць помилок, побудову графіків навчання та візуалізацію отриманих результатів. Цей компонент дозволяє здійснювати аналітичне порівняння ефективності моделей та оцінювати їх поведінку на різних етапах навчання.

Важливою архітектурною особливістю системи є відсутність жорсткої прив’язки до конкретного інтерфейсу або середовища виконання. Інструмент може використовуватися як автономний дослідницький модуль, так і бути вбудованим у серверну частину більш складних інтелектуальних інформаційних систем, зокрема рекомендаційних платформ або систем автоматичної класифікації аудіоконтенту. Такий підхід спрощує подальше масштабування, розширення функціоналу та інтеграцію з іншими програмними компонентами.

3.2 Обґрунтування вибору програмних засобів

Для реалізації інструменту аналізу аудіоданих у межах даної роботи було обрано хмарне програмне середовище Google Colab, мову програмування Python [2] та набір спеціалізованих бібліотек для обробки аудіосигналів, машинного навчання та візуалізації результатів. Такий вибір обумовлений як специфікою поставленого завдання, так і вимогами до гнучкості, масштабованості та відтворюваності експериментальних досліджень.

Google Colab було обрано як основне середовище розробки з огляду на можливість швидкого розгортання обчислювального середовища без необхідності локального налаштування програмного забезпечення. Дане

середовище надає доступ до обчислювальних ресурсів, зокрема CPU та GPU, що є критично важливим для навчання моделей машинного навчання, особливо нейронних мереж. Крім того, Google Colab забезпечує зручну інтеграцію з хмарними сховищами, що спрощує роботу з великими наборами аудіоданих, а також сприяє відтворюваності результатів та документуванню етапів дослідження у вигляді інтерактивних ноутбуків.

У якості основної мови програмування було використано Python, який є де-факто стандартом у галузі аналізу даних, машинного навчання та обробки сигналів. Python поєднує простоту синтаксису з високою виразністю, що дозволяє швидко реалізовувати та тестувати алгоритмічні рішення. Важливою перевагою Python є наявність великої кількості відкритих бібліотек, оптимізованих для наукових обчислень і роботи з даними, а також активна спільнота, яка забезпечує постійний розвиток і підтримку інструментів.

Для обробки та аналізу аудіосигналів було використано бібліотеки librosa, essentia-tensorflow та pydub. Бібліотека librosa надає широкий набір інструментів для завантаження аудіофайлів, перетворення сигналів у часово-частотну область та обчислення спектральних ознак, таких як спектрограми та мел-спектрограми. Її застосування дозволяє ефективно реалізувати етапи попередньої обробки аудіо та підготовки ознак для подальшого машинного навчання. Бібліотека essentia-tensorflow поєднує класичні методи аналізу аудіо з можливостями нейронних мереж, що є особливо корисним для дослідження інтелектуальних підходів до аналізу звукових сигналів. Pydub використовується для базових операцій з аудіофайлами, зокрема конвертації форматів і роботи з тривалістю сигналів. Для коректної роботи з різними аудіоформатами було також використано програмний пакет ffmpeg, який забезпечує універсальну підтримку кодування та декодування мультимедійних даних.

Для реалізації моделей машинного навчання та глибокого навчання було використано бібліотеки scikit-learn та TensorFlow з високорівневим API

Keras. Scikit-learn надає зручні та перевірені реалізації класичних алгоритмів машинного навчання, зокрема Random Forest, Gradient Boosting, SVM та MLP, а також інструменти для попередньої обробки даних, масштабування ознак і оцінювання якості моделей. Ця бібліотека дозволяє швидко будувати експериментальні прототипи та здійснювати порівняльний аналіз різних підходів. TensorFlow та Keras були обрані для реалізації згорткових нейронних мереж, оскільки вони забезпечують високу продуктивність, гнучкість у побудові архітектур та можливість ефективного використання апаратного прискорення. Використання callback-механізмів, таких як EarlyStopping та ReduceLROnPlateau, дозволяє контролювати процес навчання та запобігати перенавчанню моделей.

Для роботи з табличними даними та числовими обчисленнями було використано бібліотеки NumPy та pandas, які є базовими інструментами для аналізу даних у Python. Вони забезпечують ефективне зберігання, обробку та трансформацію даних, що є необхідним на етапах підготовки ознак і аналізу результатів. Візуалізація експериментальних даних та результатів навчання моделей реалізована за допомогою бібліотек matplotlib та seaborn, які дозволяють наочно відображати метрики якості, динаміку навчання та структуру помилок класифікації.

Додатково у роботі використовувалися допоміжні бібліотеки для роботи з файловою системою, серіалізації даних, завантаження ресурсів та інтеграції з середовищем виконання, зокрема os, pathlib, pickle, tqdm та IPython.display. Їх застосування підвищує зручність розробки, автоматизацію експериментів та інтерактивність дослідження.

Таким чином, обраний набір програмних засобів повністю відповідає вимогам до розробки інструменту аудіоаналізу на основі методів машинного навчання, забезпечуючи баланс між гнучкістю, продуктивністю та науковою обґрунтованістю результатів, а також створює надійну основу для подальшої інтеграції розробленого рішення в інтелектуальні інформаційні системи.

3.3 Підготовка даних

У межах розробки інструменту для аналізу аудіосигналів із використанням методів машинного навчання одним із ключових етапів є підготовка даних та попередня обробка аудіоінформації. Якість і коректність цього етапу безпосередньо впливають на результати навчання моделей, стабільність експериментів і можливість коректного порівняння різних підходів до класифікації. Оскільки аудіодані мають складну структуру, значну варіативність за тривалістю, частотними характеристиками та рівнем гучності, виникає необхідність у формуванні єдиного стандартизованого конвеєра обробки сигналів, що забезпечує уніфіковане представлення даних для подальшого аналізу.

На початковому етапі було виконано встановлення спеціалізованих системних та програмних бібліотек, необхідних для роботи з аудіофайлами різних форматів. Зокрема, було встановлено мультимедійний фреймворк ffmpeg, який забезпечує коректне декодування та обробку аудіофайлів, а також набір Python-бібліотек для аналізу аудіосигналів, роботи з нейронними мережами та класичними алгоритмами машинного навчання.

Для організації даних та результатів роботи інструменту було сформовано структуровану ієрархію директорій, яка забезпечує логічне розділення сирих даних, метаданих, оброблених ознак, збережених моделей та проміжних результатів експериментів. Такий підхід спрощує повторне використання даних, відтворюваність експериментів та подальшу інтеграцію інструменту в більші інформаційні системи. Окремі каталоги використовуються для зберігання метаданих датасету, підготовлених аудіоознак, навчених моделей та збережених результатів обчислень.

У якості основного джерела даних для дослідження було обрано відкритий музичний датасет Free Music Archive, зокрема його підмножину FMA Medium [3]. Вибір цього датасету обґрунтований кількома факторами. По-перше, FMA є широко відомим і активно використовується у наукових

дослідженнях з аналізу музичних сигналів, що забезпечує можливість порівняння отриманих результатів з іншими роботами. По-друге, підмножина FMA Medium містить значну кількість аудіотреків середньої тривалості, охоплює велику кількість музичних жанрів та має збалансовану структуру класів, що робить її придатною для задач багатокласової класифікації. По-третє, датасет супроводжується детальними метаданими, які можуть бути використані для аналізу, фільтрації та підготовки вибірок.

На етапі підготовки даних було виконано завантаження архіву з метаданими датасету, що містить інформацію про треки, жанри, ієрархію категорій та додаткові атрибути аудіофайлів. Метадані зберігаються у вигляді CSV-файлів і завантажуються у вигляді табличних структур, що дозволяє здійснювати їх аналіз, фільтрацію та зв’язування з відповідними аудіофайлами. Зокрема, файл tracks містить інформацію про кожен трек, включаючи ідентифікатор, жанрову приналежність та інші описові характеристики, тоді як файл genres використовується для аналізу структури жанрів та їх кількості у вибірці.

Після завантаження та аналізу метаданих здійснюється підготовка аудіоданих до подальшої обробки. Архів FMA Medium, що має значний обсяг, розпаковується у відповідний каталог, після чого аудіофайли стають доступними для поетапного зчитування та обробки. Кожен аудіосигнал завантажується з використанням стандартних параметрів частоти дискретизації та приводиться до моноформату, що дозволяє уніфікувати дані незалежно від початкового формату запису. За необхідності сигнали обрізаються або доповнюються до фіксованої тривалості, що забезпечує однакову довжину вхідних даних для моделей машинного навчання.

Важливим етапом попередньої обробки є перетворення аудіосигналу з часової інтерпретації у часово-частотну. Для цього використовуються мел-спектрограми, які дозволяють представити енергетичний розподіл сигналу у частотному просторі з урахуванням особливостей сприйняття звуку людиною. Отримані спектрограми додатково перетворюються у

логарифмічну шкалу та нормалізуються, що сприяє стабільності навчання нейронних мереж і зменшує вплив амплітудних коливань між різними треками.

Таким чином, реалізований етап підготовки даних забезпечує повний цикл роботи з аудіоінформацією, починаючи від завантаження сирих аудіофайлів і метаданих, організації структури зберігання даних та закінчуючи формуванням стандартизованого числового представлення аудіосигналів, придатного для використання в алгоритмах машинного навчання та глибокого навчання. Цей підхід створює основу для подальшого дослідження моделей класифікації та порівняння їх ефективності в задачах аудіоаналізу.

3.4 Формування та перевірка датасету аудіо для аналізу

Підготовка та формування датасету є одним із ключових етапів розробки системи для аудіоаналізу з використанням методів машинного навчання. Для цього було створено клас FMADataProcessor, що реалізує основні функції завантаження метаданих, перевірки наявності аудіофайлів, формування збалансованого набору даних та перевірки цілісності аудіодатасету.

На першому етапі було здійснено завантаження метаданих із FMA Medium, включно з файлами tracks.csv та genres.csv. Метадані містять інформацію про треки, їх жанрову приналежність та додаткові атрибути, що дозволяє здійснювати фільтрацію та аналіз вибірки перед формуванням датасету. Аналіз розподілу жанрів показав наявність 163 унікальних жанрів і 106 574 треків. Було виділено топ-15 жанрів за кількістю треків, серед яких найбільшу представленість мають Rock, Experimental, Electronic та Hip-Hop, що показано на рисунку 3.1.

```
Топ-15 жанрів:
(track, genre_top)
Rock                14182
Experimental        10608
Electronic           9372
Hip-Hop              3552
Folk                 2803
Pop                  2332
Instrumental         2079
International        1389
Classical            1230
Jazz                  571
Old-Time / Historic   554
Spoken                423
Country               194
Soul-RnB              175
Blues                 110
```

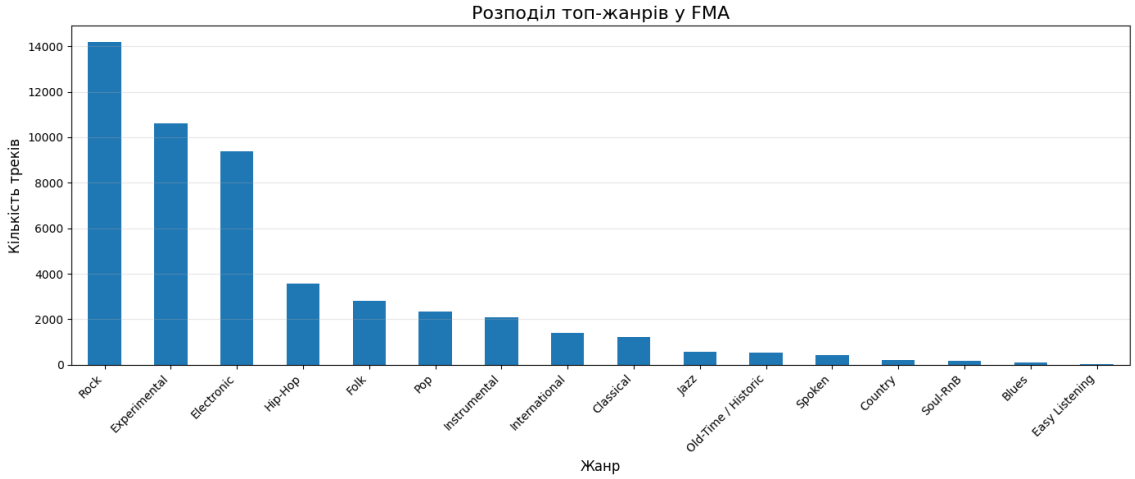

Рисунок 3.1 – Розподіл жанрів у вигляді топу

Наступним етапом стало визначення доступних аудіофайлів у локальному архіві FMA Medium. Для перевірки існування та коректності файлів використовувалися функції класу, що здійснюють пошук MP3-файлів у відповідних директоріях, завантаження перших зразків для тестування та відтворення аудіо. Це дозволяє упевнитися в наявності файлів та їх придатності для подальшої обробки.

Для формування збалансованого датасету було обрано обмежену кількість жанрів (топ-8) та фіксовану кількість треків на жанр (20–30 зразків). У результаті було створено набір із 144 треків, що охоплює жанри Rock, Experimental, Electronic, Hip-Hop, Folk, Pop, Instrumental та International. Дані зберігаються у вигляді таблиці з колонками track_id, genre та audio_path, що забезпечує зручну інтеграцію з подальшими алгоритмами обробки аудіосигналів. Процес формування датасету показано на рисунку 3.2. Розподіл жанрів музики у створеному датасеті показано на рисунку 3.3.

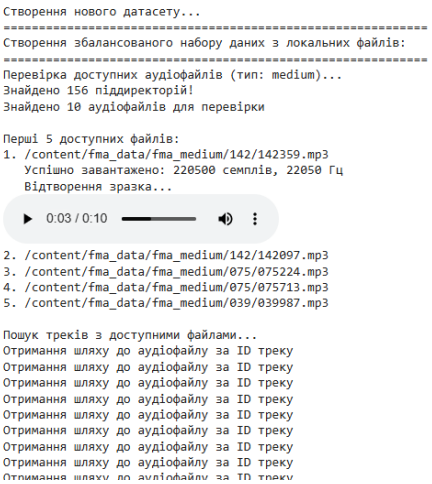

Рисунок 3.2 – Процес формування датасету

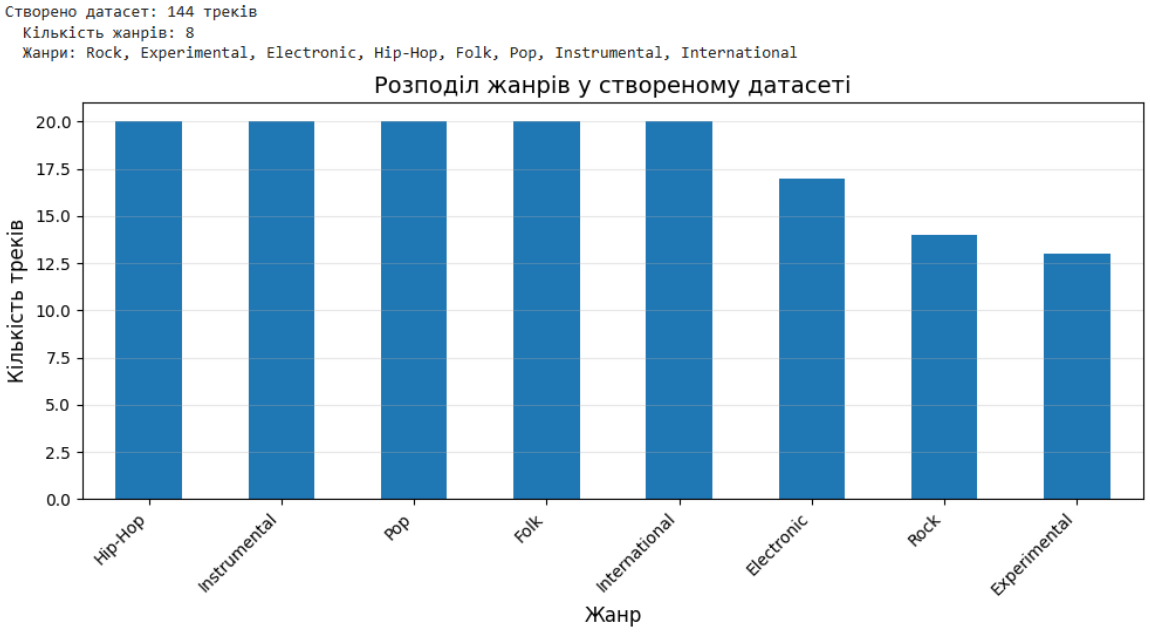

Рисунок 3.3 – Розподіл жанрів музики створеного датасету

Після формування датасету здійснено його валідацію. Кожен аудіофайл перевірявся на наявність та можливість коректного завантаження у Python-бібліотеці Librosa. Результати показали, що всі 144 треки є валідними, відсутніх або некоректних файлів не виявлено. Для тестування завантаження та відтворення аудіо було відібрано один з валідних треків, який успішно завантажився та відтворився протягом 15 сек. (рисунок 3.4).

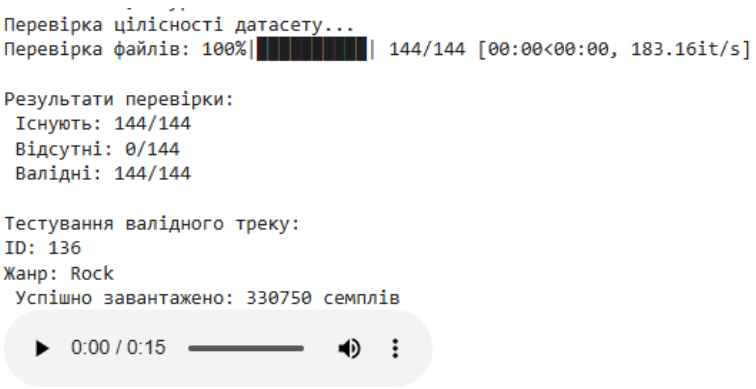

Рисунок 3.4 – Валідація датасету

Формування такого збалансованого аудіодатасету забезпечує стандартизоване представлення даних для подальшого використання у методах машинного та глибокого навчання. Завдяки збереженню структури та перевірці цілісності датасету забезпечується повторюваність експериментів та надійність отриманих результатів, що є критично важливим для досліджень у сфері аудіоаналізу та розробки інтелектуальних інформаційних систем.

3.5 Розробка класу для вилучення ознак з аудіосигналів та підготовка даних для навчання

Одним із ключових етапів побудови системи аудіоаналізу є перетворення сирого аудіосигналу у формалізований набір числових ознак, придатних для подальшого використання в алгоритмах машинного навчання. З цією метою в межах даної роботи було розроблено спеціалізований програмний клас AudioFeatureExtractor, який реалізує комплексний підхід до вилучення акустичних характеристик музичних треків.

Основною ідеєю розробленого підходу є поєднання спектральних, ритмічних, тембральних та мел-кепстральних ознак у єдиному просторі характеристик. Такий підхід дозволяє охопити різні аспекти аудіосигналу та

забезпечує більш повне його представлення порівняно з методами, що використовують лише один тип ознак.

На першому етапі здійснюється завантаження аудіофайлу з приведенням його до стандартних параметрів. Аудіо сигнал приводиться до монофонічного формату, фіксованої частоти дискретизації 22 050 Гц та обмеженої тривалості 30 секунд. Така уніфікація є важливою для забезпечення порівнюваності ознак між різними треками та стабільності навчання моделей машинного навчання.

У випадку помилок завантаження аудіофайлу система обробляє виняткові ситуації, що підвищує надійність обробки великих датасетів. MFCC-ознаки є одним із найбільш поширених інструментів аналізу аудіо, оскільки вони моделюють сприйняття звуку людським слухом. У межах розробленого класу обчислюються середні значення MFCC, стандартні відхилення та середні значення похідних (delta MFCC).

Таке представлення дозволяє зменшити розмірність даних, зберігаючи при цьому ключову інформацію про тембральні характеристики сигналу. Для аналізу частотної структури аудіосигналу використовуються спектральні ознаки, зокрема спектральний центр мас, спектральна ширина, спектральний контраст, спектральний спад та частота нульових перетинів. Ці характеристики дозволяють описати енергетичний розподіл сигналу у частотній границі та є особливо корисними для жанрової класифікації музики.

Ритмічна структура є важливою складовою музичних треків. Для її аналізу використовуються оцінка темпу (beats per minute) та хроматичні ознаки, отримані різними методами (STFT, CQT, CENS). Хроматичні ознаки дозволяють описати гармонічний вміст треку незалежно від октави, що є важливим для аналізу музичних стилів. Водночас, для аналізу тембру та динаміки аудіосигналу використовуються середні та дисперсійні значення RMS-енергії, мел-спектрограми у децибелах, тональна мережа і розділення

гармонічної та перкусивної складових сигналу. Такий підхід дозволяє більш точно описати характер звучання треку та його емоційні особливості.

Усі отримані ознаки об’єднуються в єдиний вектор характеристик, який додатково містить інформацію про тривалість аудіофайлу. Для обробки великих обсягів даних реалізовано пакетну обробку (batch processing) із періодичним збереженням результатів, що дозволяє ефективно працювати з датасетами значного розміру та мінімізувати втрати даних у разі збоїв. Отримані ознаки зберігаються у серіалізованому вигляді, що спрощує їх повторне використання та інтеграцію з іншими компонентами системи. Після вилучення ознак здійснюється етап підготовки даних до навчання моделей машинного навчання. Для цього було розроблено окремий клас DataPreprocessor, який реалізує повний цикл передобробки даних.

На етапі формування навчальних вибірок здійснюється об’єднання векторів ознак з відповідними жанровими мітками, отриманими з метаданих датасету. Для кожного аудіофайлу формується єдиний числовий вектор, що включає всі доступні характеристики. З метою зменшення впливу дисбалансу класів у датасеті реалізовано механізм фільтрації жанрів, кількість зразків яких є меншою за заданий поріг. Це дозволяє підвищити стабільність навчання моделей та уникнути спотворення результатів.

Категоріальні жанрові мітки кодуються у числовий формат за допомогою методу label encoding. Числові ознаки масштабуються до стандартного розподілу, що є критично важливим для алгоритмів, чутливих до масштабів даних, зокрема нейронних мереж та методів опорних векторів.

Для оцінки якості моделей дані поділяються на тренувальну та тестову вибірки із збереженням пропорцій класів. У випадках, коли це неможливо, використовується випадкове розділення. Додатково реалізовано інструменти для аналізу балансу класів та оцінки важливості ознак за допомогою ансамблевих методів. Це дозволяє дослідити внесок окремих характеристик у процес класифікації та зробити висновки щодо

інформативності різних типів ознак. Процес підготовки даних для навчання показаний на рисунку 3.5 та візуалізація розподілу класів на рисунку 3.6 та в таблиці 3.1.

```
============================================================
ПІДГОТОВКА ДАНИХ ДЛЯ НАВЧАННЯ
============================================================
Завантаження ознак...
Завантажено 144 записів ознак
Використання датасету з 144 треків
Ініціалізація препроцесора...
Підготовка ознак для навчання
Обробка ознак: 100%|██████████| 144/144 [00:00<00:00, 787.12it/s]

Початковий розподіл класів:
  Electronic: 17 зразків
  Experimental: 13 зразків
  Folk: 20 зразків
  Hip-Hop: 20 зразків
  Instrumental: 20 зразків
  International: 20 зразків
  Pop: 20 зразків
  Rock: 14 зразків

Фінальна статистика:
  Розмірність ознак: (144, 245)
  Кількість класів: 8
  Загальна кількість зразків: 144
```

Рисунок 3.5 – Підготовка даних для навчання

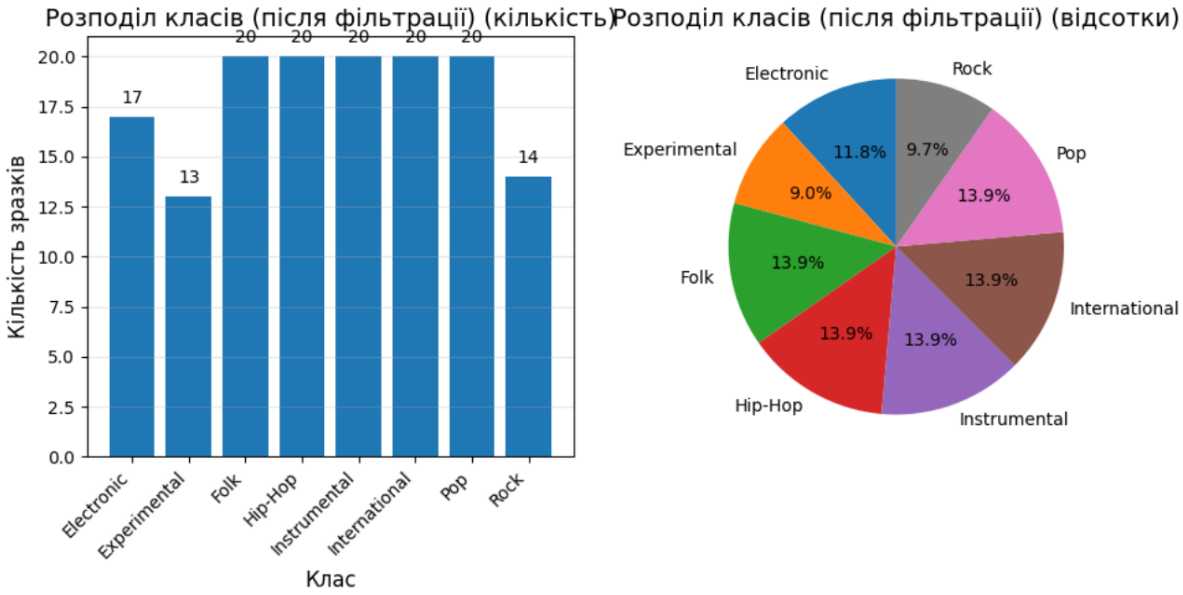

Рисунок 3.6 – Візуалізація розподілу класів

Таблиця 3.1 – Розподіл даних за класами

Розподіл за класами	Початковий розподіл:	Тренувальна вибірка:	Тестова вибірка:
Electronic (клас 0)	17 зразків	14 зразків	3 зразки
Experimental (клас 1)	13 зразків	10 зразків	3 зразки
Folk (клас 2)	20 зразків	16 зразків	4 зразки
Hip-Hop (клас 3)	20 зразків	16 зразків	4 зразки
Instrumental (клас 4)	20 зразків	16 зразків	4 зразки
International (клас 5)	20 зразків	16 зразків	4 зразки
Pop (клас 6)	20 зразків	16 зразків	4 зразки
Rock (клас 7)	14 зразків	11 зразків	3 зразки

Розподіл класів (після фільтрації):

– загальна кількість зразків: 144;
– кількість класів: 8;
– мінімальна кількість на клас: 13;
– максимальна кількість на клас: 20;
– середня кількість на клас: 18.0;
– стандартне відхилення: 2.8;
– коефіцієнт дисбалансу: 1.54;
– масштабування ознак: (144, 245);
– тренувальна вибірка: (115, 245);
– тестова вибірка: (29, 245);
– кількість класів: 8.

3.6 Навчання моделей ML

Після завершення етапів вилучення ознак та підготовки даних наступним ключовим кроком стало навчання класичних моделей машинного навчання, які слугують базовим рівнем для оцінки ефективності розробленого підходу до аудіоаналізу. Для уніфікації процесу навчання, оцінювання та збереження результатів було реалізовано окремий клас

MLModelTrainer, який інкапсулює логіку роботи з різними алгоритмами та забезпечує зручне порівняння їхніх характеристик.

Основною ідеєю цього модуля є використання однієї й тієї самої підготовленої вибірки для навчання кількох моделей, що дозволяє об’єктивно оцінити їхню здатність до класифікації аудіоданих за жанрами. Усі моделі навчаються з фіксованим значенням random_state, що забезпечує відтворюваність експериментів і коректність порівняльного аналізу.

Першою моделлю є Random Forest – ансамблевий метод, заснований на побудові великої кількості дерев рішень і агрегації їхніх результатів. Для даної задачі було використано розширену конфігурацію з підвищеною кількістю дерев, обмеженою глибиною та додатковими параметрами регуляризації, що дозволяє зменшити ризик перенавчання на багатовимірних ознаках аудіосигналу. Навчання моделі супроводжується вимірюванням часу тренування, після чого виконується передбачення на тестовій вибірці та обчислення основних метрик якості, зокрема загальної точності та детального звіту класифікації. Отримані результати зберігаються для подальшого аналізу й порівняння.

Другим підходом є метод опорних векторів (SVM), який традиційно вважається одним із найбільш ефективних класичних алгоритмів для задач класифікації з чітко розділеними класами. У межах даного проєкту використовується SVM із радіально-базисним ядром, що дозволяє моделювати нелінійні залежності між ознаками. Водночас цей метод є обчислювально затратним, особливо на великих вибірках, тому для практичної реалізації було застосовано навчання на зменшеній підмножині тренувальних даних. Такий компроміс дає змогу отримати репрезентативну оцінку якості SVM без надмірних витрат часу та ресурсів. Як і для інших моделей, виконується оцінка точності, формування звіту класифікації та збереження результатів.

Третьою моделлю є Gradient Boosting – ансамблевий метод, який базується на послідовному навчанні слабких моделей із поступовою

мінімізацією помилки. На відміну від Random Forest, де дерева навчаються незалежно, градієнтний бустинг дозволяє більш точно підлаштовувати модель під складні структури даних, що є особливо актуальним для задач аудіоаналізу з великою кількістю статистичних і спектральних ознак. Для цієї моделі було підібрано помірну кількість ітерацій та глибину дерев, що забезпечує баланс між точністю та стабільністю навчання.

Окрему увагу приділено багатошаровій нейронній мережі прямого поширення (MLP), яка хоча й належить до нейромережевих підходів, але в контексті даної роботи розглядається як класична модель машинного навчання, що працює з векторними ознаками. Архітектура мережі складається з кількох прихованих шарів зі зменшенням розмірності, що дозволяє моделі поступово виділяти більш абстрактні представлення аудіоознак. Для стабілізації навчання використовується адаптивна швидкість навчання та механізм ранньої зупинки, який запобігає перенавчанню. Процеси навчання показані на рисунку 3.7.

```
Навчання Random Forest...
[Parallel(n_jobs=-1)]: Using backend ThreadingBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done  46 tasks      | elapsed:    0.1s
[Parallel(n_jobs=-1)]: Done 196 tasks      | elapsed:    0.4s
[Parallel(n_jobs=-1)]: Done 200 out of 200 | elapsed:    0.4s finished
[Parallel(n_jobs=2)]: Using backend ThreadingBackend with 2 concurrent workers.
[Parallel(n_jobs=2)]: Done  46 tasks      | elapsed:    0.0s
[Parallel(n_jobs=2)]: Done 196 tasks      | elapsed:    0.1s
[Parallel(n_jobs=2)]: Done 200 out of 200 | elapsed:    0.1s finished
[Parallel(n_jobs=2)]: Using backend ThreadingBackend with 2 concurrent workers.
[Parallel(n_jobs=2)]: Done  46 tasks      | elapsed:    0.0s
[Parallel(n_jobs=2)]: Done 196 tasks      | elapsed:    0.0s
[Parallel(n_jobs=2)]: Done 200 out of 200 | elapsed:    0.1s finished
Точність Random Forest: 0.4828
Час навчання: 0.50 секунд

Навчання MLP (нейронної мережі)...
Iteration 1, loss = 2.25580219
Validation score: 0.083333
Iteration 2, loss = 1.76651355
Validation score: 0.083333
Iteration 3, loss = 1.52195908
Validation score: 0.166667
Iteration 4, loss = 1.29563325
Validation score: 0.250000
Iteration 5, loss = 1.06321206
Validation score: 0.250000
Iteration 17, loss = 0.05565512
Validation score: 0.416667
Iteration 18, loss = 0.04565204
Validation score: 0.416667
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Точність MLP: 0.3103
Час навчання: 0.28 секунд

Навчання SVM...
[LibSVM]Точність SVM: 0.7241
Час навчання: 0.12 секунд

Навчання Gradient Boosting...
      Iter       Train Loss   Remaining Time
         1           1.4228           25.33s
         2           1.1066           26.38s
         3           0.8860           26.19s
         4           0.7231           26.11s
         5           0.5909           25.94s
         6           0.4863           25.80s
         7           0.4022           25.62s
         8           0.3337           25.75s
         9           0.2776           25.60s
        10           0.2314           25.44s
        20           0.0406           23.49s
        30           0.0076           21.66s
        40           0.0013           20.25s
        50           0.0002           19.31s
        60           0.0000           17.18s
        70           0.0000           15.11s
        80           0.0000           13.09s
        90           0.0000           11.04s
       100           0.0000            8.89s
Точність Gradient Boosting: 0.3448
Час навчання: 18.31 секунд
```

Рисунок 3.7 – Фрагменти логування навчання ML моделей.

Після навчання всіх моделей реалізовано етап візуалізації результатів. Для цього будуються порівняльні графіки точності та часу навчання, що наочно демонструють компроміс між якістю класифікації та обчислювальними витратами для кожного алгоритму. Додатково формуються матриці помилок, які дозволяють детально проаналізувати характер помилок класифікації, виявити жанри, що найчастіше плутаються між собою, та оцінити сильні й слабкі сторони кожної моделі. Візуалізація точності показана на рисунках 3.8 – 3.13.

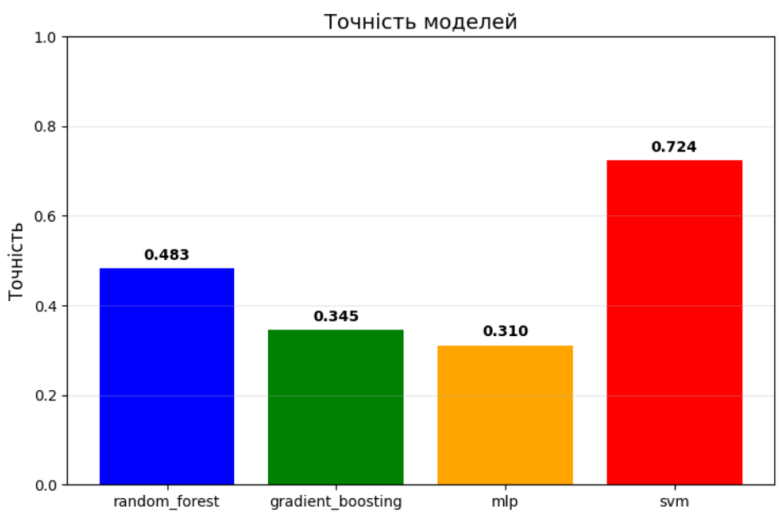

Рисунок 3.8 – Візуалізація точності моделей

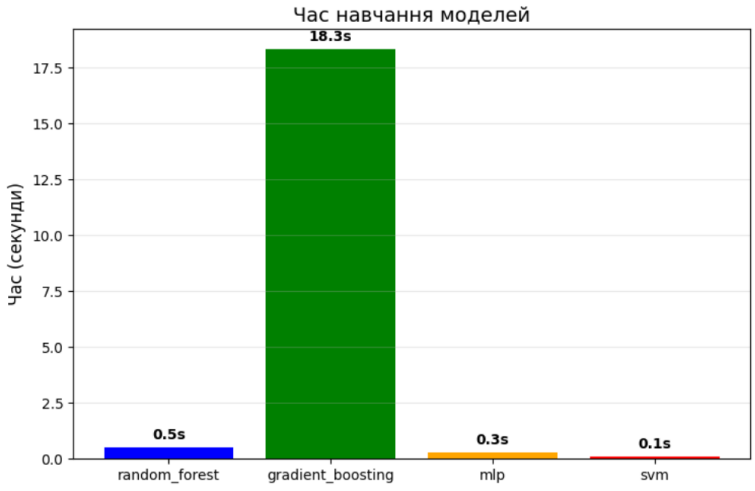

Рисунок 3.9 – Візуалізація часу навчання моделей

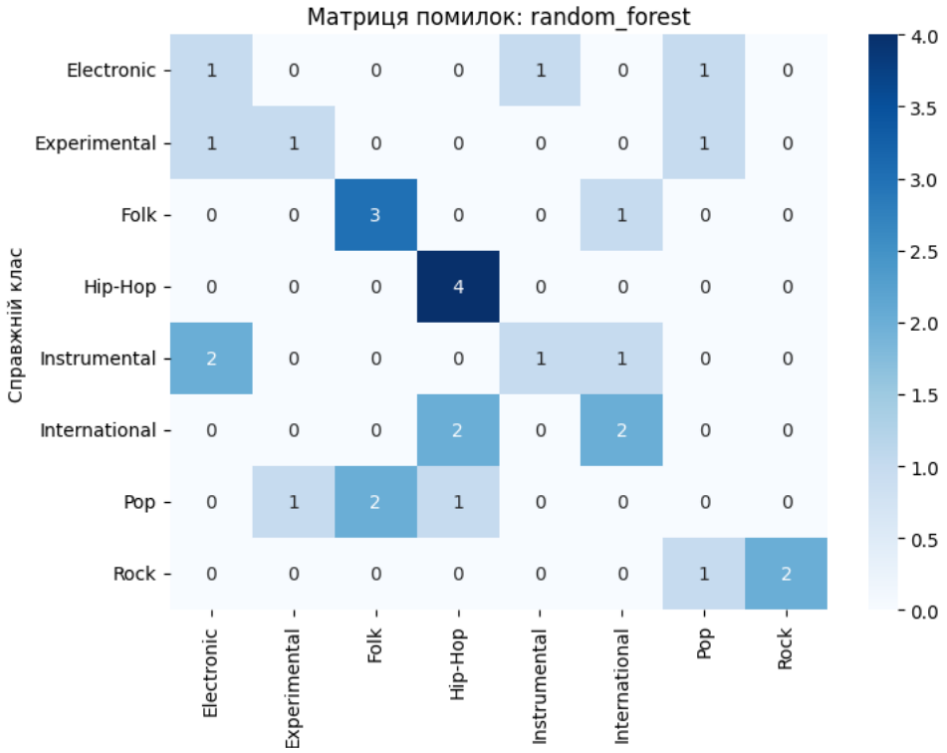

Рисунок 3.10 – Матриця помилок random forest

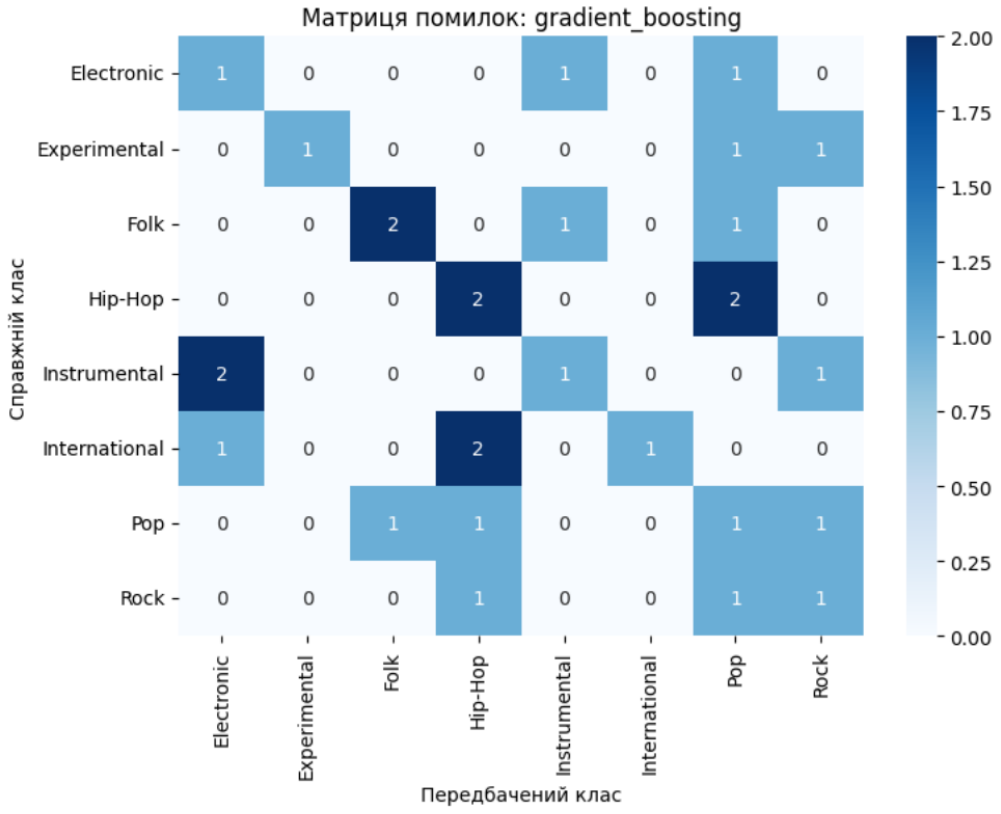

Рисунок 3.11 – Матриця помилок gradient_boosting

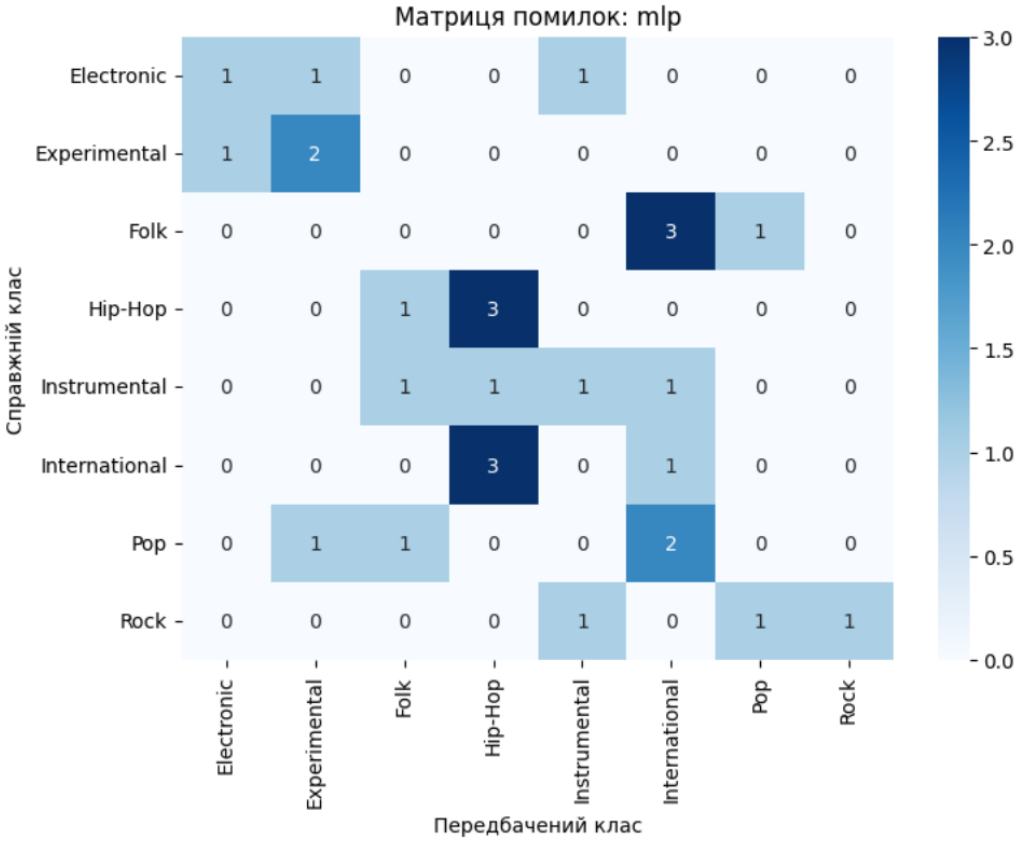

Рисунок 3.12 – Матриця помилок mlp

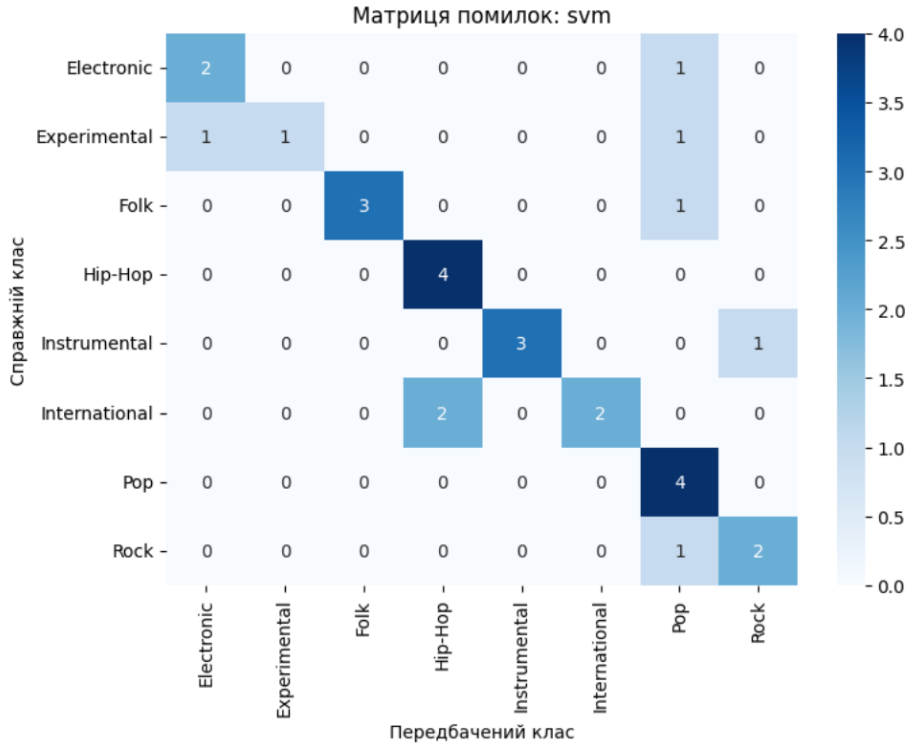

Рисунок 3.13 – Матриця помилок svm

За допомогою візуалізацій, можна побачити, що найточнішою виявилась модель svm, хоча більш за все часу на навчання потребувала модель gradient_boosting.

Завершальним етапом є збереження навчених моделей та результатів експериментів у файлову систему. Це забезпечує можливість подальшого повторного використання моделей без необхідності повторного навчання, а також спрощує інтеграцію розробленого інструменту в інші інформаційні системи або дослідницькі середовища.

3.7 Глибока нейронна мережа CNN

Згорткова нейронна мережа у межах даного проєкту використовується як ключовий інструмент глибокого аналізу аудіосигналів на основі їх спектрального подання. На відміну від класичних моделей машинного навчання, які працюють з попередньо обчисленими числовими ознаками, CNN безпосередньо навчається на двовимірних представленнях сигналу, що дозволяє моделі самостійно виявляти локальні та глобальні закономірності у часово-частотній сфері. У якості такого представлення використовується мел-спектрограма, яка є наближеним відображенням сприйняття звуку людським слухом і широко застосовується в задачах класифікації музики, мовлення та акустичних подій.

Процес побудови та навчання CNN у даній роботі починається з етапу перетворення сирого аудіосигналу у стандартизоване вхідне подання. Для кожного аудіофайлу сигнал приводиться до фіксованої частоти дискретизації та тривалості, після чого обчислюється мел-спектрограма. Отримане спектральне представлення додатково нормалізується, що зменшує вплив різниці в амплітудах та сприяє стабільнішому навчанню нейронної мережі. Для забезпечення однакової розмірності вхідних даних спектрограми або доповнюються нульовими значеннями, або обрізаються

до заданої ширини по часовій осі, після чого перетворюються у тензори з одним каналом, придатні для подачі у CNN.

Підготовлені спектрограми формують вхідну вибірку для навчання, валідації та тестування моделі. Такий підхід дозволяє зберегти просторову структуру даних, де одна вісь відповідає часовій еволюції сигналу, а інша частотному розподілу енергії. Саме ця структура є принципово важливою для згорткових мереж, оскільки згорткові фільтри здатні виявляти характерні патерни, зокрема ритмічні структури, гармоніки та спектральні переходи, які важко формалізувати у вигляді класичних ознак. Приклади спектрограм деяких класів (жанрів) показані на рисунках 3.14 – 3.18

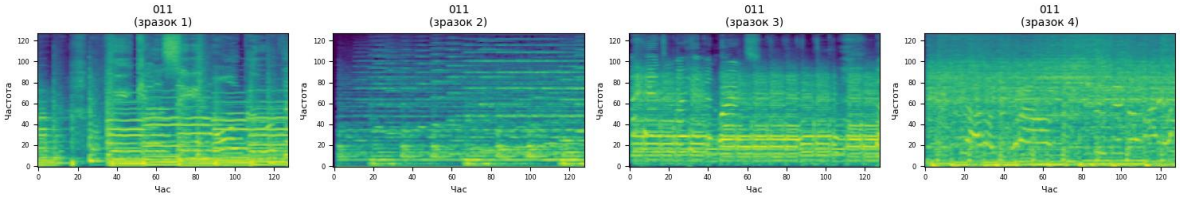

Рисунок 3.14 – приклади спектрограм для класу 0

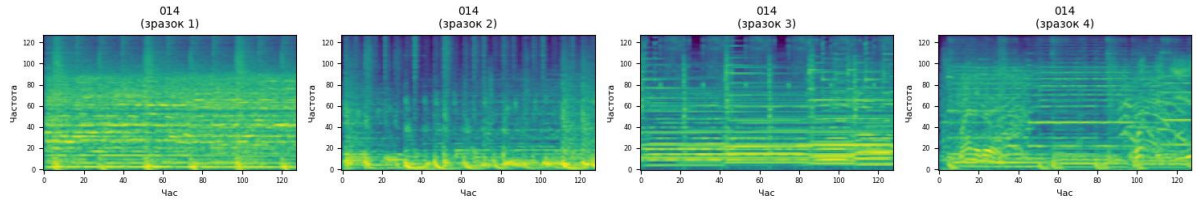

Рисунок 3.15 – приклади спектрограм для класу 1

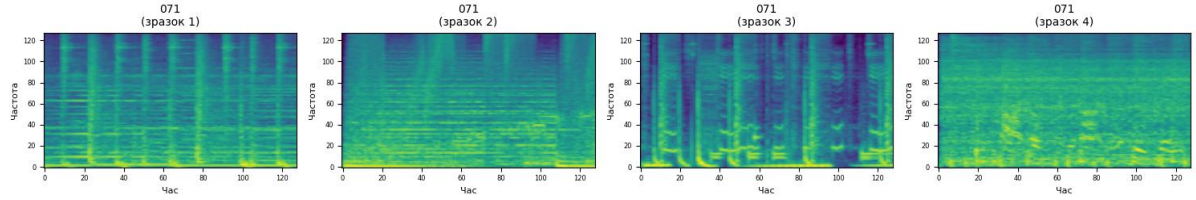

Рисунок 3.16 – приклади спектрограм для класу 2

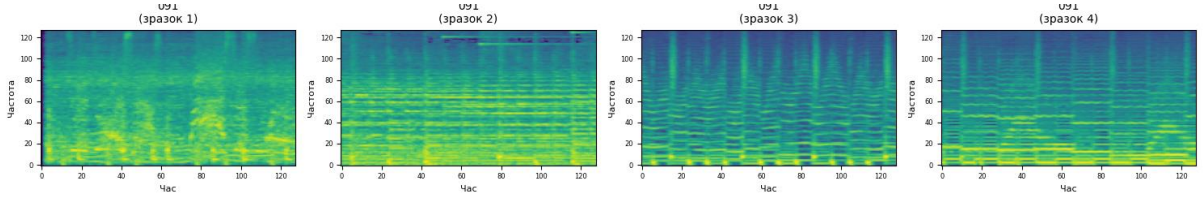

Рисунок 3.17 – приклади спектрограм для класу 3

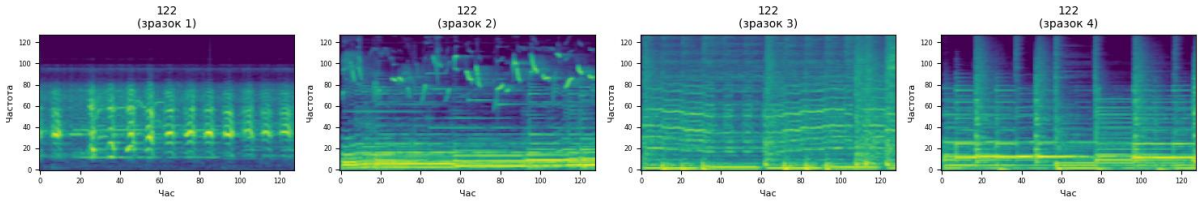

Рисунок 3.18 – приклади спектрограм для класу 4

Архітектура згорткової нейронної мережі у даному проєкті побудована як послідовність кількох згорткових блоків зі зростаючою кількістю фільтрів. На початкових рівнях мережа навчається виявляти прості локальні ознаки, такі як різкі зміни енергії або характерні частотні смуги, тоді як на глибших рівнях формується більш абстрактне уявлення про структуру аудіосигналу. Кожен згортковий блок поєднує операцію згортки з нелінійною функцією активації, нормалізацію та підвибірку, що дозволяє зменшити розмірність представлення і підвищити стійкість до незначних варіацій у даних. Додаткове використання випадкового вимикання нейронів зменшує ризик перенавчання та покращує узагальнювальну здатність моделі.

Після згорткових шарів отримане багатовимірне представлення ознак перетворюється у вектор та передається до повнозв’язних шарів. Ці шари виконують роль інтеграції виявлених ознак і формування фінального рішення щодо належності аудіофрагмента до певного класу. Регуляризація ваг і використання нормалізації на цьому етапі дозволяють стабілізувати процес навчання та зменшити чутливість моделі до шумових компонентів у даних. Отримано необхідну кількість ознак та побудовано модель sequential_2, архітектуру якої показано в таблиці 3.2.

Таблиця 3.2 – Архітектура CNN моделі

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 128, 128, 32)	320
batch_normalization_12 (BatchNormalization)	(None, 128, 128, 32)	128
max_pooling2d_8 (MaxPooling2D)	(None, 64, 64, 32)	0
dropout_12 (Dropout)	(None, 64, 64, 32)	0
conv2d_9 (Conv2D)	(None, 64, 64, 64)	18,496
batch_normalization_13 (BatchNormalization)	(None, 64, 64, 64)	256
max_pooling2d_9 (MaxPooling2D)	(None, 32, 32, 64)	0
dropout_13 (Dropout)	(None, 32, 32, 64)	0
conv2d_10 (Conv2D)	(None, 32, 32, 128)	73,856
batch_normalization_14 (BatchNormalization)	(None, 32, 32, 128)	512
max_pooling2d_10 (MaxPooling2D)	(None, 16, 16, 128)	0
dropout_14 (Dropout)	(None, 16, 16, 128)	0
conv2d_11 (Conv2D)	(None, 16, 16, 256)	295,168
batch_normalization_15 (BatchNormalization)	(None, 16, 16, 256)	1,024
max_pooling2d_11 (MaxPooling2D)	(None, 8, 8, 256)	0
dropout_15 (Dropout)	(None, 8, 8, 256)	0
flatten_2 (Flatten)	(None, 16384)	0
dense_6 (Dense)	(None, 512)	8,389,120
batch_normalization_16 (BatchNormalization)	(None, 512)	2,048
dropout_16 (Dropout)	(None, 512)	0
dense_7 (Dense)	(None, 256)	131,328
batch_normalization_17 (BatchNormalization)	(None, 256)	1,024
dropout_17 (Dropout)	(None, 256)	0
dense_8 (Dense)	(None, 8)	2,056

Total params: 8,915,336 (34.01 MB)
Trainable params: 8,912,840 (34.00 MB)
Non-trainable params: 2,496 (9.75 KB)

Наведена таблиця відображає підсумкову архітектуру згорткової нейронної мережі для класифікації аудіосигналів на основі мел-спектрограм розміром 128×128×1, яка складається з чотирьох послідовних згорткових блоків із поступовим збільшенням кількості фільтрів від 32 до 256 та одночасним зменшенням просторової розмірності до 8×8, що забезпечує ієрархічне виділення спектральних ознак різного рівня абстракції; кожен блок поєднує згортку, batch normalization, max pooling і dropout, що підвищує стабільність навчання та зменшує перенавчання, після чого відбувається перехід до повнозв'язної частини через шар Flatten з формуванням вектора довжиною 16384; основне зосередження параметрів припадає на перший dense-шар із 512 нейронами, який містить понад 8 млн параметрів і відповідає за моделювання складних нелінійних залежностей між ознаками, далі застосовується додатковий повнозв'язний шар на 256 нейронів і вихідний шар із 8 нейронами відповідно до кількості класів; загальна кількість параметрів моделі становить близько 8,9 млн, з яких майже всі є навчуваними, що свідчить про високу ємність моделі та її придатність для глибокого аналізу аудіоданих у межах поставленої задачі.

Навчання CNN здійснюється ітеративно з використанням оптимізаційного алгоритму на основі градієнтного спуску. У процесі навчання відстежується якість моделі як на навчальній, так і на валідаційній вибірках, що дозволяє контролювати баланс між точністю та узагальненням (рисунок 3.19). Для автоматичного регулювання процесу навчання застосовуються механізми дострокової зупинки та адаптивної зміни швидкості навчання, які запобігають надмірному перенавчанню та скорочують час обчислень. Збереження найкращої версії моделі за результатами валідації забезпечує можливість подальшого використання CNN без необхідності повторного навчання.

```
Початок навчання моделі...
Параметри навчання:
Епохи: 50
Розмір батча: 32
Розмір навчальної вибірки: 560
Навчання CNN моделі...
Epoch 1/50
18/18 ━━━━━━━━━━━━━━━━━━━━ 65s 3s/step - accuracy: 0.1426 - loss: 4.5052 - val_accuracy: 0.1250 - val_loss: 4.4554 - learning_rate: 0.0010
Epoch 2/50
18/18 ━━━━━━━━━━━━━━━━━━━━ 71s 3s/step - accuracy: 0.1168 - loss: 4.5761 - val_accuracy: 0.1250 - val_loss: 4.9017 - learning_rate: 0.0010
Epoch 3/50
18/18 ━━━━━━━━━━━━━━━━━━━━ 80s 3s/step - accuracy: 0.1468 - loss: 4.3622 - val_accuracy: 0.1250 - val_loss: 6.0272 - learning_rate: 0.0010
Epoch 4/50
18/18 ━━━━━━━━━━━━━━━━━━━━ 81s 3s/step - accuracy: 0.1910 - loss: 4.2713 - val_accuracy: 0.1250 - val_loss: 7.0877 - learning_rate: 0.0010
Epoch 5/50
18/18 ━━━━━━━━━━━━━━━━━━━━ 85s 3s/step - accuracy: 0.1765 - loss: 4.2706 - val_accuracy: 0.1250 - val_loss: 6.0469 - learning_rate: 0.0010
Epoch 6/50
18/18 ━━━━━━━━━━━━━━━━━━━━ 45s 2s/step - accuracy: 0.1682 - loss: 4.1338 - val_accuracy: 0.1250 - val_loss: 7.2801 - learning_rate: 0.0010
Epoch 7/50
18/18 ━━━━━━━━━━━━━━━━━━━━ 84s 3s/step - accuracy: 0.2405 - loss: 3.9152 - val_accuracy: 0.1250 - val_loss: 8.2954 - learning_rate: 5.0000e-04
Epoch 8/50
18/18 ━━━━━━━━━━━━━━━━━━━━ 82s 3s/step - accuracy: 0.2473 - loss: 3.8653 - val_accuracy: 0.1250 - val_loss: 6.4242 - learning_rate: 5.0000e-04
Epoch 9/50
18/18 ━━━━━━━━━━━━━━━━━━━━ 51s 3s/step - accuracy: 0.1987 - loss: 3.9699 - val_accuracy: 0.1333 - val_loss: 6.1167 - learning_rate: 5.0000e-04
Epoch 10/50
```

Рисунок 3.19 – Процес навчання моделі

Оцінювання навченого згорткового класифікатора проводиться на відкладеній тестовій вибірці, що дозволяє отримати об’єктивну характеристику якості моделі. Окрім інтегральних показників точності, аналізуються передбачені класи та ймовірнісні оцінки (таблиця 3.3), що дає змогу детальніше дослідити поведінку моделі для різних типів аудіосигналів. Додаткова візуалізація історії навчання у вигляді кривих точності та функції втрат (рисунок 3.20) дозволяє інтерпретувати динаміку процесу оптимізації та підтвердити коректність обраної архітектури.

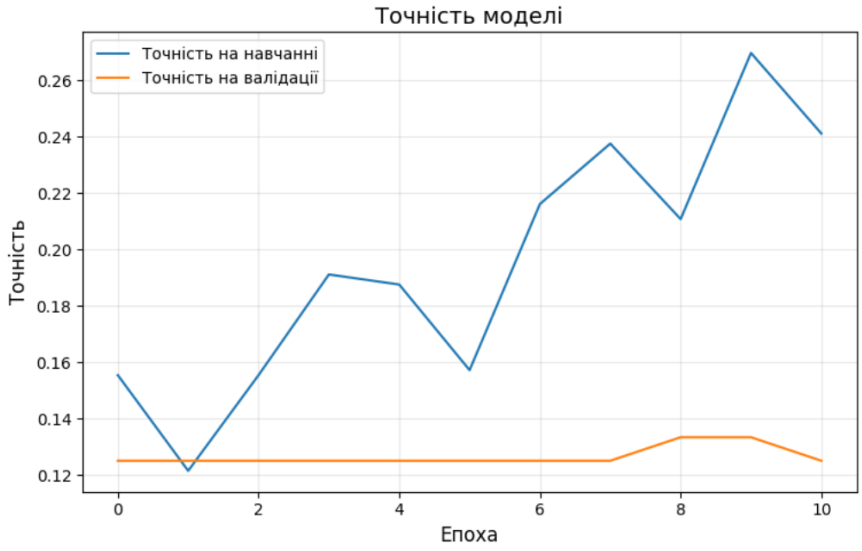

Рисунок 3.20 – Точність моделі

Таблиця 3.3 – Результати тестування моделі

Тест для жанру	Передбачений жанр	Впевненість	Топ-3 передбачення:		
			011	014	071
rock	011	28.79%	28.79%	28.01%	25.93%
pop	011	28.61%	28.61%	28.15%	26.07%
jazz	014	28.33%	28.33%	28.25%	26.20%
classical	011	30.28%	30.28%	27.59%	25.02%

3.8 Використання моделей для аналізу аудіо

Фінальним етапом розробки системи є практичне застосування навчених моделей машинного та глибокого навчання для аналізу довільних музичних аудіофайлів і визначення їх жанрової приналежності. На цьому етапі реалізовано повноцінний програмний модуль, який об’єднує всі попередні компоненти системи: попередню обробку аудіосигналу, видобування ознак, застосування кількох моделей класифікації та інтерпретацію результатів у зручному для користувача вигляді.

Основним елементом цього етапу є клас EnhancedMusicGenrePredictor, який виконує роль центрального керуючого модуля системи. Під час ініціалізації класу відбувається завантаження збережених моделей машинного навчання (Random Forest, Gradient Boosting, MLP, SVM), згорткової нейронної мережі (CNN), а також препроцесора, що містить об’єкти масштабування ознак і кодування класів. Такий підхід забезпечує відтворюваність результатів і повну відповідність процесу аналізу етапу навчання моделей.

Для аналізу музичного файлу система спочатку виконує видобування аудіоознак. За наявності спеціалізованого екстрактора ознак використовується розширений набір характеристик, а у разі його відсутності застосовується базовий підхід, що включає MFCC, хроматичні ознаки, спектральний центр, ширину спектра, roll-off, RMS та частоту

нульових перетинів. Отримані ознаки перетворюються у вектор фіксованої розмірності та масштабуються відповідно до параметрів, використаних під час навчання моделей.

Після цього система виконує паралельне передбачення жанру за допомогою всіх доступних моделей машинного навчання. Кожна модель повертає не лише передбачений клас, але й імовірнісний розподіл по всіх жанрах, що дозволяє оцінити ступінь впевненості окремого класифікатора. Окремо реалізовано модуль передбачення на основі CNN, який працює безпосередньо зі спектрограмами: аудіофайл перетворюється у нормалізовану мел-спектрограму фіксованого розміру, яка подається на вхід нейронної мережі.

Важливою особливістю фінального етапу є реалізація механізму консенсусного передбачення. Результати всіх моделей агрегуються шляхом зваженого голосування, де внесок кожної моделі визначається її ймовірністю передбачення. Такий ансамблевий підхід дозволяє зменшити вплив окремих помилкових рішень і підвищити загальну стабільність та надійність системи порівняно з використанням однієї моделі.

Для підвищення інтерпретованості результатів система формує розширену візуалізацію, яка включає порівняння ймовірностей усіх моделей, кругову діаграму консенсусу, теплову карту впевненості класифікаторів, порівняння CNN та класичних ML-підходів, а також статистичний аналіз результатів. Додатково відображається інформація про сам аудіофайл, включно з тривалістю, частотою дискретизації та розміром.

Фінальний етап також передбачає можливість базової інтерактивної роботи з користувачем: завантаження власних аудіофайлів, їх попереднє прослуховування, автоматичний запуск аналізу та збереження результатів у текстовий файл.

Для тестування розробленої системи жанрової класифікації був використаний класичний представник жанру фолк-року – композиція Animals – House of the Rising Sun, що є показовим прикладом поєднання

фолкових мотивів із рок-аранжуванням і тому добре підходить для перевірки коректності роботи моделей. У процесі аналізу аудіофайлу було виконано повний цикл попередньої обробки та видобування ознак, зокрема завантаження аудіо, витяг MFCC, спектральних, ритмічних і тембральних характеристик, у результаті чого сформовано вектор із 245 ознак, який був поданий на вхід класичним ML-моделям. За результатами їх передбачень Random Forest відніс трек до жанру International з імовірністю 17,10 %, Gradient Boosting з дуже високою впевненістю 96,95 % класифікував його як Folk, нейронна мережа MLP також визначила жанр Folk з імовірністю 25,69 %, а SVM з імовірністю 18,66 %, що свідчить про домінування фолкової інтерпретації серед більшості класичних підходів. Окремо було застосовано CNN-модель, вона класифікувала трек як Experimental з імовірністю 40,05 %, що відображає її чутливість до нетипових спектральних та тембрових особливостей композиції. Узагальнення результатів усіх п’яти моделей у межах консенсусного підходу дало підсумковий жанр Folk з впевненістю 71,2 %, при цьому середня імовірність по моделях склала 39,69 %, максимальна – 96,95 %, мінімальна – 17,10 %, а стандартне відхилення 0,2976. Поетапно результати роботи моделей машинного навчання для аудіоаналізу в інтелектуальних інформаційних системах показані на рисунках 3.21 – 3.25.

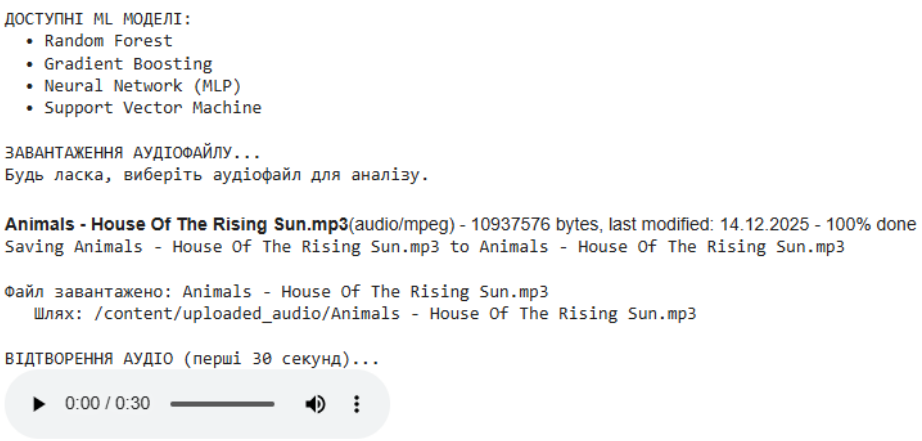

Рисунок 3.21 – Завантаження композиції

```
ПЕРЕДБАЧЕННЯ ML МОДЕЛЯМИ:

Видобування ознак з: Animals - House Of The Rising Sun.mp3
Витяг усіх ознак
Обробка: Animals - House Of The Rising Sun.mp3
Завантаження аудіофайлу
Витяг MFCC ознак
Витяг спектральних ознак
Витяг ритмічних ознак
Витяг тембральних ознак
Видобуто 245 ознак
  Random Forest: International (17.10%)
  Gradient Boosting: Folk (96.95%)
  Neural Network (MLP): Folk (25.69%)
  Support Vector Machine: Folk (18.66%)

  ПЕРЕДБАЧЕННЯ CNN МОДЕЛЛЮ:
[Parallel(n_jobs=2)]: Using backend ThreadingBackend with 2 concurrent workers.
[Parallel(n_jobs=2)]: Done  46 tasks      | elapsed:    0.0s
[Parallel(n_jobs=2)]: Done 196 tasks      | elapsed:    0.0s
[Parallel(n_jobs=2)]: Done 200 out of 200 | elapsed:    0.0s finished
  CNN: Experimental (40.05%)

КОНСЕНСУСНЕ ПЕРЕДБАЧЕННЯ:
     КОНСЕНСУСНИЙ ЖАНР: Folk
     Впевненість: 28.26%
     Кількість моделей: 5
```

Рисунок 3.22 – Обробка та передбачення моделями

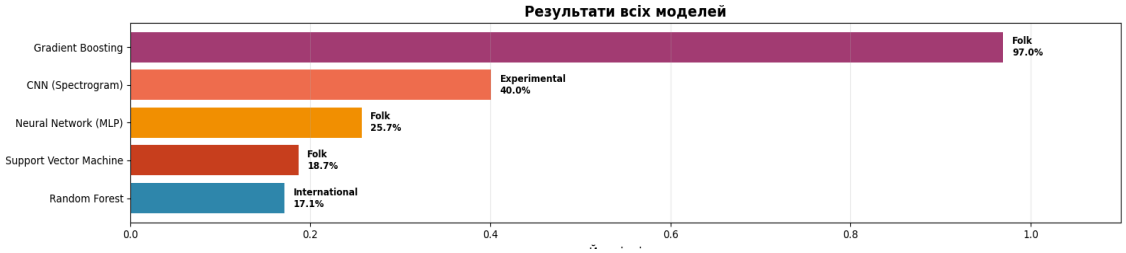

Рисунок 3.23 – Гістограма результатів всіх моделей

Рисунок 3.24 – Консенсусне передбачення

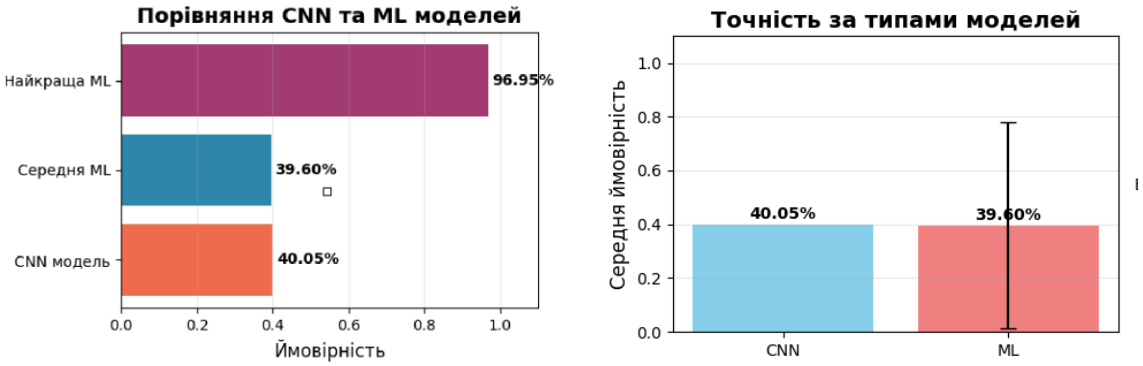

Рисунок 3.25 – Порівняння моделей

Це підтверджує доцільність використання ансамблю моделей для підвищення надійності жанрової класифікації та демонструє, як поєднання класичних ML-методів і CNN дозволяє отримати більш збалансований та інтерпретований результат аналізу аудіо.

ВИСНОВКИ

Під час виконання роботи було реалізовано повний цикл розробки та дослідження інструменту автоматичної класифікації музичних жанрів на основі аналізу аудіосигналів із застосуванням методів машинного навчання та глибинних нейронних мереж. Основний акцент робився не на створенні готової рекомендаційної системи, а на побудові універсального механізму порівняння та інтерпретації аудіотреків, який у подальшому може бути інтегрований у різноманітні прикладні рішення, зокрема системи рекомендацій або музичні пошукові сервіси. У результаті роботи було підтверджено, що коректна попередня обробка аудіоданих та якісне видобування ознак є критично важливими етапами, які суттєво впливають на поведінку всіх моделей незалежно від їх складності.

Під час підготовки даних було розроблено алгоритм обробки аудіосигналів, який включає завантаження та нормалізацію аудіо, а також видобування різних груп ознак, що описують тембральні, спектральні та ритмічні характеристики музики. Такий підхід дозволив сформувати інформативне представлення аудіотреків у числовому вигляді, придатному для використання класичними моделями машинного навчання. Використання датасету FMA було обґрунтованим рішенням, оскільки він містить велику кількість треків різних жанрів, має чітку структуру та супровідні метадані, що робить його зручним для досліджень і порівняння результатів між різними моделями.

У межах роботи було реалізовано та проаналізовано декілька підходів до жанрової класифікації, зокрема класичні моделі машинного навчання та згорткову нейронну мережу. Класичні ML-моделі загалом продемонстрували стабільну та передбачувану поведінку, особливо у випадках жанрів із чітко вираженими статистичними та тембральними ознаками. Такі моделі добре реагують на структуровані числові характеристики та часто забезпечують інтерпретовані результати, що є

важливою перевагою з точки зору аналізу. Водночас було помітно, що різні моделі по-різному сприймають музику, наприклад, у деяких експериментах саме Random Forest демонстрував найкращий результат і впевнено визначав правильний жанр у ситуаціях, коли інші моделі давали розпливчасті або суперечливі відповіді. Це свідчить про здатність ансамблевих методів ефективно працювати з неоднорідними та зашумленими ознаками.

Згорткова нейронна мережа, яка працює безпосередньо зі спектрограмами, показала принципово інший характер поведінки. CNN здатна виявляти складні локальні та глобальні патерни в часово-частотному представленні аудіосигналу, що робить її особливо перспективною для аналізу сучасних або нетипових жанрів, де класичні ознаки можуть бути недостатньо інформативними. У ряді випадків CNN виявлялася точнішою за ML-моделі та краще відображала реальну стилістичну природу треку. Водночас для жанрів із традиційною структурою або чіткими статистичними характеристиками її результати іноді були менш стабільними, що можна пояснити як обмеженим обсягом навчальних даних, так і високою чутливістю моделі до особливостей спектрограми.

Важливим результатом роботи стало спостереження, що жодна з використаних моделей не є універсально найкращою для всіх жанрів. Поведінка моделей значною мірою залежить від характеру музики: деякі жанри краще розпізнаються класичними методами, інші нейронними мережами. Саме тому застосування ансамблевого або консенсусного підходу є виправданим і практично доцільним. Об’єднання результатів кількох моделей дозволяє згладити слабкі сторони окремих підходів і отримати більш збалансоване та надійне фінальне рішення. Навіть у випадках, коли окремі моделі помиляються або демонструють низьку впевненість, колективне рішення часто відображає реальний жанровий характер композиції.

Загалом результати роботи можна вважати позитивними, оскільки поставленої мети було досягнуто: створено гнучкий інструмент аналізу

аудіосигналів, реалізовано кілька підходів до жанрової класифікації та проведено їх порівняльний аналіз. Отримані результати підтверджують, що поєднання класичних методів машинного навчання з глибинними нейронними мережами є перспективним напрямом для задач аналізу музики. Розроблений підхід має потенціал для подальшого розвитку, зокрема шляхом розширення набору ознак, оптимізації архітектури CNN, використання більших обсягів даних або адаптації системи під конкретні жанрові підмножини. Таким чином, виконана робота створює міцну основу для подальших досліджень у сфері автоматичного аналізу музики та аудіосигналів і може бути використана як практичний приклад застосування методів машинного навчання в мультимедійних задачах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. API docs | last.fm. *Last.fm*. URL: https://www.last.fm/api (дата звернення: 04.12.2025).

2. Chollet F. Deep learning with python. Manning Publications Co. LLC, 2017. 384 с.

3. GitHub – mdeff/fma: FMA: a dataset for music analysis. *GitHub*. URL: https://github.com/mdeff/fma (дата звернення: 08.12.2025).

4. Humphrey E. J., Bello J. P., LeCun Y. Feature learning and deep architectures: new directions for music informatics. *Journal of intelligent information systems*. 2013. Т. 41, № 3. С. 461–481. URL: https://doi.org/10.1007/s10844-013-0248-5 (дата звернення: 08.12.2025).

5. Librosa: audio and music signal analysis in python / B. McFee та ін. *Python in science conference*, м. Austin, Texas. 2015. URL: https://doi.org/10.25080/majora-7b98e3ed-003 (дата звернення: 11.12.2025).

6. Müller M. Fundamentals of music processing: audio, analysis, algorithms, applications. Springer, 2016. 516 с.

7. Music discovery and recommendation at soundcloud. *SoundCloud Developers Blog*. URL: https://developers.soundcloud.com/blog (дата звернення: 12.12.2025).

8. Scikit-learn: machine learning in Python. *scikit-learn: machine learning in Python – scikit-learn 0.16.1 documentation*. URL: https://scikit-learn.org (дата звернення: 10.12.2025).

9. Spotify's official technology blog | Spotify Engineering. *Spotify Engineering*. URL: https://engineering.atspotify.com/ (дата звернення: 09.12.2025).

10. TensorFlow. *TensorFlow*. URL: https://www.tensorflow.org (дата звернення: 08.12.2025).

11. Tzanetakis G., Cook P. Musical genre classification of audio signals. *IEEE transactions on speech and audio processing*. 2002. Т. 10, № 5. С. 293–302. URL: https://doi.org/10.1109/tsa.2002.800560 (дата звернення: 06.12.2025).

12. Walter G. A. Algorithmic culture: youtube recommendation system. *Cultural and artistic practices: world and ukrainian context*. 2023. С. 169–184. URL: https://doi.org/10.30525/978-9934-26-322-4-8 (дата звернення: 07.12.2025).