

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

факультет інформатики та обчислювальної техніки  
(повна назва інституту/факультету)

кафедра автоматика та управління в технічних системах  
(повна назва кафедри)

«На правах рукопису»  
УДК 004.89

«До захисту допущено»  
Завідувач кафедри

О. І. Ролік  
(підпис) (ініціали, прізвище)

“ ” \_\_\_\_\_ 2018 р.

## Магістерська дисертація

зі спеціальності (спеціалізації) 121 «Інженерія програмного забезпечення»  
(код і назва спеціальності)

на тему: Інтелектуальна система підбору музикального контенту

Виконав : студент 6 курсу, групи ІТ-73мп  
(шифр групи)

Хапілін Олександр Сергійович \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент, к.т.н. Дорогий Я.Ю. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант \_\_\_\_\_  
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент доцент кафедри ІБ, к.т.н., доц. Демчинський В.В \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

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

Студент  
Хапілін О.С.

\_\_\_\_\_ (підпис)

Київ – 2018 року

## АНОТАЦІЯ

В даній роботі 88 сторінки текстової інформації, 32 рисунки та 47 таблиць.

Магістерська дисертація містить результати розробки інтелектуальної системи підбору музичного контенту. Актуальність даної роботи полягає у різкому збільшенні медіа інформації, доступної користувачу для перегляду онлайн, в особливості музичних записів. Метою даною роботи є розробка інтелектуальної системи пошуку музичного контенту. Об'єктом даної роботи є інтелектуальні алгоритми підбору контенту. Предметом даної роботи є алгоритми машинного навчання для навчання з учителем та без.

Ключові слова: пошук музичного контенту, класифікація, предиктивне кодування, наближений пошук найближчих сусідів.

The master's thesis contains results of intelligent musical content search system development. The relevance of this work is in rapid increase of available media information for users to watch online, especially musical recording. The aim of this work is to develop intellectual musical content search system. The object of this work are intelligent algorithms of content search. The subject of this work are supervised and unsupervised machine learning algorithms.

Keywords: music information retrieval, classification, predictive coding, approximate nearest neighbors search.

## ЗМІСТ

ЗМІСТ .....	4
ВСТУП .....	6
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	7
1.1 Last.fm.....	7
1.2 Spotify .....	8
1.3 Методи колаборативної фільтрації .....	10
1.4 Кластеризація.....	12
1.5 Pandora.....	13
1.6 Word2Vec .....	15
1.7 Shazam .....	18
3 РОЗРОБКА МОДЕЛІ .....	23
3.1 Розробка функціональних вимог до системи.....	23
3.2 Розробка нефункціональних вимог до системи .....	24
3.3 Загальний підхід до поставленої задачі .....	24
3.4 Проста згорткова модель.....	25
3.5 Модель із конволюційними блоками та рекурентністю .....	28
3.6 Класифікація жанрів .....	34
3.7 Контрастивне предиктивне кодування .....	40
4 РОЗРОБКА СИСТЕМИ .....	50
4.1 Сценарії використання системи.....	50
4.2 Розробка структурної схеми .....	67
4.3 Розробка функціональної схеми .....	68
4.4 Вибір та обґрунтування елементів та технологій .....	70
4.5 Розробка бази даних.....	71

	5
4.2 Розгортання системи.....	73
5 ОЦІНКА МОДЕЛІ ТА ФІНАЛЬНИЙ ВАРІАНТ АЛГОРИТМУ .....	74
6 РОЗРОБКА СТАРТАП-ПРОЕКТУ .....	75
6.1 Опис ідеї проекту .....	75
6.2 Технологічний аудит ідеї проекту .....	77
6.3 Аналіз ринкових можливостей стартап проекту .....	78
6.4 Розроблення ринкової стратегії проекту .....	87
6.5 Розроблення маркетингової програми стартап проекту .....	90
ВИСНОВКИ.....	94
СПИСОК ВИКОРСТАНИХ ДЖЕРЕЛ.....	95
ДОДАТОК А.....	97
ДОДАТОК Б .....	98
ДОДАТОК В .....	99
ДОДАТОК Г .....	100
ДОДАТОК Д.....	101
ДОДАТОК Е .....	102
ДОДАТОК Ж.....	103
ДОДАТОК К.....	104
ДОДАТОК Л.....	105

## ВСТУП

З розвитком інформаційних технологій створення і поширення мультимедійної інформації стало просте як ніколи. Разом з цим з'являється потреба обробляти ці дані таким чином, щоб був можливий зручний пошук по цим даним. Якщо розглядати окремо лише музичні файли, то проблема їх індексації полягає у змінних у часі характеристик та доволі нечіткого поняття їх схожості, на відміну від, наприклад, текстових документів. Крім того, жоден існуючий рекомендаційний сервіс не може повністю задовольнити потреби користувачів. Так, наприклад, Last.fm та Spotify не використовують інформацію, що міститься безпосередньо в аудіо, сервіс Pandora, хоча і покладається на розмітку музичних файлів експертами, не може повноцінно поповнюватися в автоматичному режимі, що впливає у найменшій серед своїх конкурентів бібліотеці, а Shazam має можливість шукати лише повне співпадіння композиції. Звідси у користувачів впливає потреба у сервісі, що може вирішити ці проблеми.

Метою дипломної роботи є створення розумної системи, що дозволяє знаходити схожі музичні композиції за контентом. Для цього система повинна мати певний перелік властивостей та вирішувати такі задачі:

- система повинна виробляти рекомендації базуючись лише на інформації, що безпосередньо міститься в аудіо;
- система має мати можливість обробляти дані в автоматичному режимі, в тому числі й ті дані, що раніше не оброблялися системою, а тому вона не має залежати від роботи експертів чи будь-якої іншої розмітки користувачами;
- аудіо інформація повинна бути подана у стислому вигляді для можливості подальшого зручного індексування та кластеризації.

Об'єктом даної роботи є інтелектуальні алгоритми підбору контенту.

Предметом даної роботи є алгоритми машинного навчання для навчання з учителем та без.

# 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

## 1.1 Last.fm

Last.fm - сайт та набір програмного забезпечення та плагинів, що дозволяють прослуховувати музику, переглядати інформацію про виконавців, збирати інформацію про музику певного користувача[1, 2].

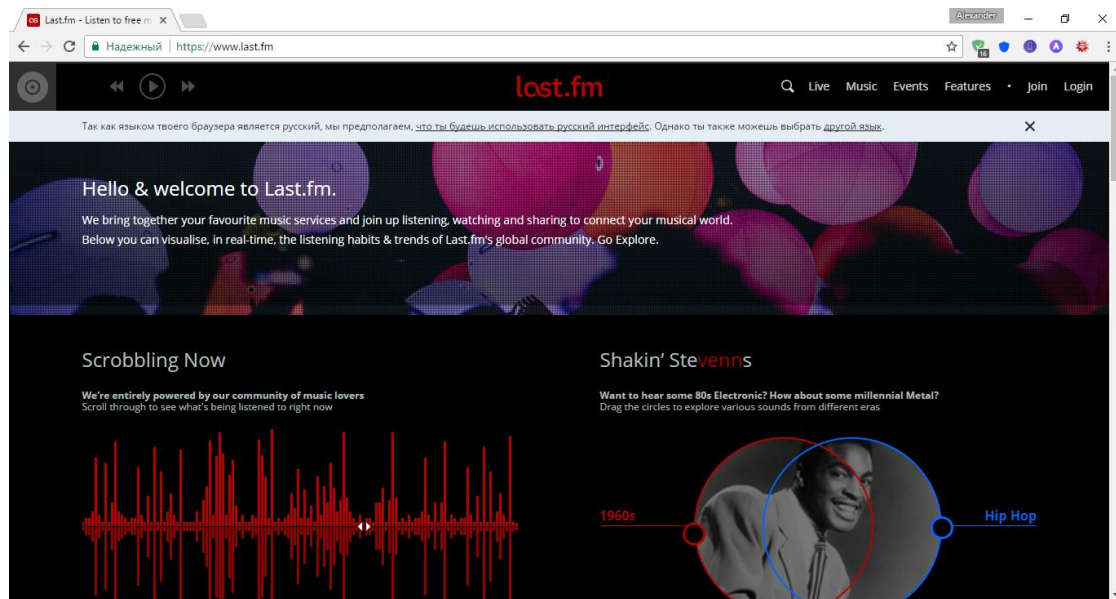


Рисунок 1.1 - Веб-інтерфейс Last.fm

Особливістю даного сервіса є можливість створювати індивідуальні та загальні хіт-паради (“чарти”). Для цього в Last.fm використовують таку техніку, як “скробблінг”. Скробблінг - це метод збору та передачі на сервер інформації про музику, що переслуховує користувач. Це може бути інформація про жанр, виконавця, рік видання та альбом. Для цього не потребувалося постійне підключення до інтернету, програма спочатку зберігала дані, а потім відправляла за можливістю. Для цього можна використовувати спеціальні плагіни, які інтегруються в медіапрогравачі.

Для навігації та групування композицій використовуються теги, які можуть проставляти користувачі.

Завдяки багатому набору інформації, що збирає скробблер, Last.fm побудували систему рекомендацій на основі алгоритмів колаборативної фільтрації [3]. В залежності від історії користувача, йому пропонувалося прослуховувати музичні треки, що популярні серед користувачів зі схожими вподобаннями, переглядати сторінки-профілі цих користувачів (“сусідів”, як їх назвали), пропонувалися концерти, які могли зацікавити користувача. На жаль, після редизайну сайту в 2014 році більша частина цього функціоналу була втрачена.

## 1.2 Spotify

Spotify - музичний стрімінговий сервіс, офіційно запущений 7 жовтня 2008 г. Він розроблений у стартапі Spotify AB в Стокгольмі, Швеція [4]. Spotify є безкоштовним сервісом з деякими платними функціями, а це означає, що основні функції є безкоштовні але з рекламою, в той час як додаткові функції, включаючи поліпшену якість потокового аудіо і в завантаження музики для автономного прослуховування, пропонуються через платні підписки.

Spotify доступний в більшості країн Європи, більшість з Північної та Південної Америки, Австралії, Нової Зеландії та деяких частинах Азії. Він доступний для більшості сучасних пристроїв, включаючи Windows, MacOS, Linux і комп'ютери, а також IOS і Android смартфонів і планшетів. Музика може бути переглянута або знайдена за допомогою різних параметрів, таких як виконавець, альбом, жанр, плейлист або звукозаписний лейбл. Користувачі можуть створювати, редагувати і обмінюватися списками програвання, ділитися треками в соціальних мережах, а також створювати списки з іншими користувачами. Spotify забезпечує доступ до більш ніж 30 мільйонів пісень. Станом на червень 2016 року, він має більше 100 мільйонів активних користувачів щомісяця, а за станом на березень 2017 року, він має більш ніж 50 мільйонів передплатників.

На відміну від фізичних або проданих для завантаження копій, які платять артистам фіксованою ціною за продану пісню або альбом, Spotify платить

ліцензійні платежі на основі кількості прослуховувань виконавців як частка від загальних пісень трансляції на службі. Вони розподіляють близько 70% від загального доходу правовласників, які потім платять виконавцям на основі їх індивідуальних угод.

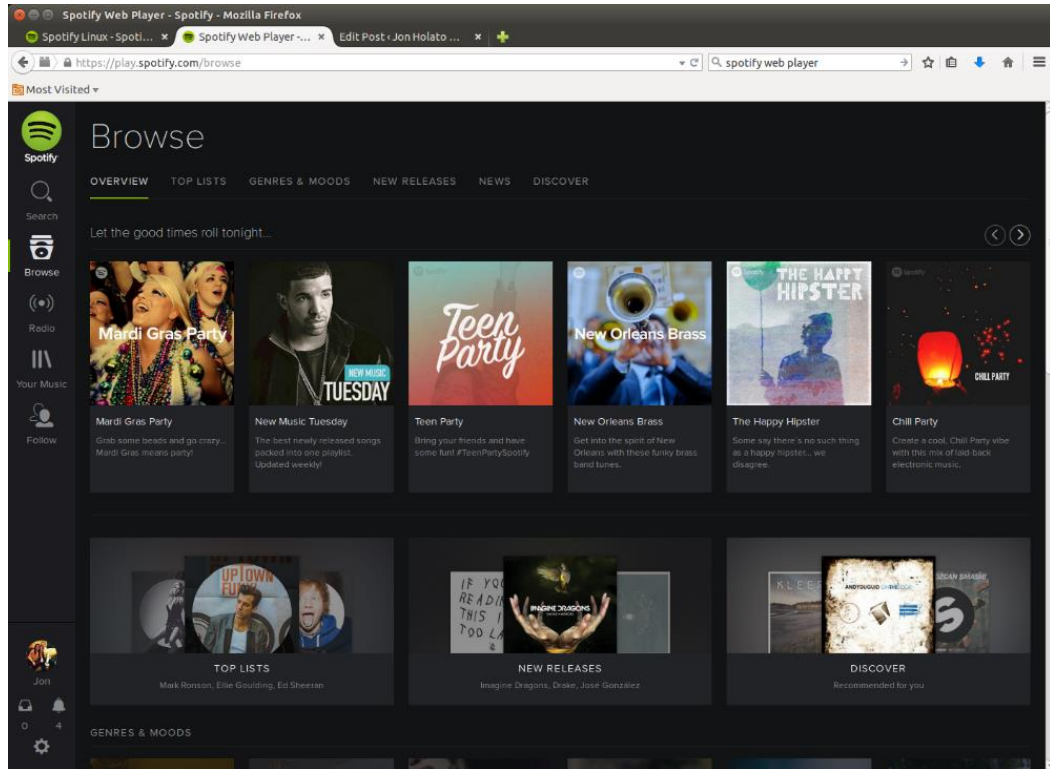


Рисунок 1.2 – Приклад веб-інтерфейсу Spotify

На відміну від Last.fm він дозволяє повноцінно прослуховувати музику, має більше можливостей і більш гнучкий програвач. Користувачі так само можуть створювати плейлисти та подкасти.

Загалом - Spotify це як повноцінний музична бібліотека, але в більш зручному форматі. Spotify так само має рекомендаційну систему. Можна продивлятися рекомендації щодо виконавців, альбомів та пісень. Як і Last.fm використовує методи колаборативної фільтрації.



### 1.3 Методи колаборативної фільтрації

Обидва попередньо розглянуті сервіси мають за основу рекомендаційної системи методи колаборативної фільтрації [4]. Колаборативна фільтрація - процес пошуку рекомендацій для певного користувача на основі інформації про багатьох користувачів. У випадках коли колаборативна фільтрація використовує знання про інших користувачів, використовуються знання про групу для створення рекомендацій на основі подібності користувачів. По своїй суті рекомендації базуються на співпраці багатьох користувачів (колаборація) для відокремлення тих користувачів, що мають схожі смаки або поведінку (фільтрація).

Методи колаборативної фільтрації можна поділити на:

1. Методи пошуку схожості користувачів (User-based)
2. Методи пошуку схожості елементів (Item-based)

Мета обох методів - об'єднання схожих об'єктів в групи на основі матриці оцінок.

Схожість користувачів обчислюється за такою загальною формулою:

$$N(u_0) = \{(u_0, u) > \theta\}, \quad (1.1)$$

де  $u_0, u$  – користувач,  $\text{sim}$  – функція подібності оцінок користувачів (наприклад Евклідова відстань чи коефіцієнт узгодження Пірсон),  $\theta$  – поріг, який визначає, чи вважати даних користувачів схожими. Після вибору схожих користувачів, для користувача  $u_0$  обирається елемент  $r$ , який вже був оцінений схожим користувачем  $u$ , та вираховується передбачення оцінки користувача  $u_0$  елементу  $r$ .

В найпростішому випадку схожість користувачів характеризують якоюсь метрикою на основі спільних оцінок. Це може бути навіть Евклідова відстань, хоча частіше використовують коефіцієнт кореляції Пірсона. Використання кореляції дозволяє уникнути проблеми, коли користувач завжди ставить завищенні або

занижені оцінки. З іншого боку, якщо користувач ставить лише високі або низькі оцінки, дисперсія таких даних буде дорівнювати 0 і кореляцію не можна буде точно визначити. Крім того, коли дані в нас бінарні, має сенс використати міри Жаккара або Танімото. Як видно, вибір оптимальної метрики задача досить нетривіальна.

Якщо потрібно застосувати колаборативну фільтрацію для визначення схожості об'єктів, а не користувачів, то для цього лише треба повторити всі тіж кроки, але рахувати міру схожості на основі оцінок спільних користувачів для якогось певного об'єкта.

До переваг методів колаборативної фільтрації можна віднести:

- простоту реалізації;
- простота збору даних;
- відсутність необхідності якоїсь інформації про порівнювані об'єкти крім оцінок користувачів;

До недоліків можна віднести:

- складність вибору метрики;
- складність вирішення проблем, пов'язаних із даними та метриками (нульова дисперсія, зсув математичного очікування, тощо);
- при пошуку подібності взагалі не враховується ніяка інформація про порівнювані об'єкти

Важливо відмітити, що останній недолік є найкритичнішим, якщо мова йде про порівняння музичних композицій. Оскільки алгоритм не враховує жанри, інструменти, вокал та ін., а тільки оцінки користувачів, користувачі, що мають різноманітний смак або полюбляють переважно сучасні хіти, будуть негативно впливати на адекватність оцінок схожості, в чому можна пересвідчитися після користування сервісами Last.fm або Spotify.

## 1.4 Кластеризація

Інший спосіб отримувати оцінку схожості об'єктів - алгоритми кластеризації [5]. Завдання кластеризації полягає у пошуку схожих об'єктів та об'єднання їх у групи таким чином, щоб близькі за якимось критерієм об'єкти потрапляли в одну групу. Кластеризаційні алгоритми є методами навчання без вчителя. Це значить, що нема потреби власноруч задавати схожість об'єктів. Кластеризація проводиться в просторі ознак об'єктів. Ефективність кластеризації сильно залежить від вибору цього простору. Усі алгоритми в основі своєї роботи покладають так звану гіпотезу "компактності". Гіпотеза полягає у припущенні, що схожі об'єкти повинні розташовуватися в ознаковому просторі ближче ніж не схожі.

До переваг алгоритмів кластеризації можна віднести:

- врахування ключових властивостей об'єктів при правильному виборі ознакового простору;
- навчання проходить без вчителя, що дозволяє вирішувати задачі, що не до кінця формалізовані;

До недоліків можна віднести:

- не існує однозначно найкращого критерія якості розбиття кластерів;
- часто оброто критерій якості розбиття доволі складно;
- метрики схожості мають сильний вплив на розбиття і їх вибір також складна задача;

В контексті даної роботи до вище наведених недоліків можна додати, що для музичних композицій вибір ознакового простору доволі непроста задача, адже музика - протяжна і залежна від часу, рідко різні композиції будуть мати однакові часові характеристики, що ускладнює процес пошуку ознакового простору.

## 1.5 Pandora

Pandora - сервіс інтернет-радіо, що створений спеціально для пошуку нової музики [6]. Користувач обирає виконавця, жанр або пісню, а сайт створює радіостанцію, що буде програвати треки схожих виконавців. Крім того можна прослуховувати станції для жанрів або ті, що були раніше створені іншими користувачами. Для персоналізації рекомендацій можна ставити відмітки сподобалося/не сподобалося для програних пісень. Дана інформація буде використана пізніше для покращення рекомендацій.

Для виконавців, альбомів чи інших атрибутів такої можливості нема. Крім цього, для керування програвачем є такі опції, як перегляд інформації, чому саме ця пісня рекомендована, можливість перемістити композицію до іншої станції, створити нову станцію з цієї пісні, зробити закладку, або придбати.

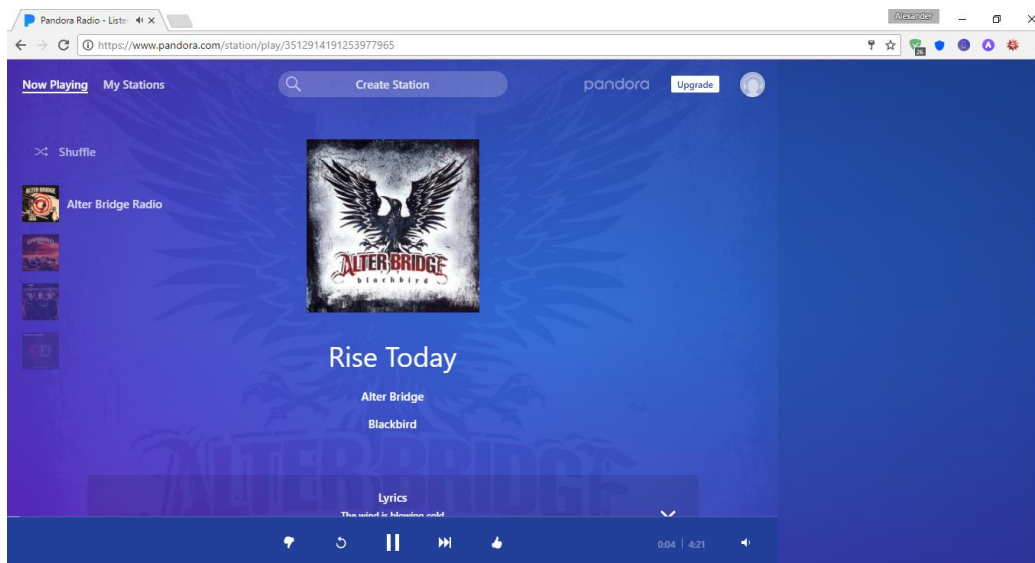


Рисунок 1.3 - Веб-інтерфейс Pandora

На відміну від Last.fm та Spotify, Pandora не використовує інформацію про інших користувачів. В основі алгоритмів Pandora лежить так званий Music Genome Project. Music Genome Project - система, що ставить у відповідність музичним композиціям різні її характеристики, усього близько 400 штук. До цих характеристик можуть відноситися ритм, характеристики голосу вокаліста, тексти

пісень та інші. Цей проект було розпочато в 2000 році і зайняло 5 років створення у 30-ти музичних експертів. Цей проект заснований на ручному аналізі пісень. Експерти тратять близько 20-30 хвилин на одну пісню. Всього сервіс містить близько 400 000 пісень. “Музичний геном” - це не постійний набір ознак. Для різних жанрів необхідний різний підхід. Для цього було виокремлено такі жанрові групи, як поп, джаз, електроніка та інші. Як кажуть засновники Music Genome Project, існує геном, що пов’язує всі жанри, проте, наприклад, народна музика потребує більшого різноманіття інструментів, ніж поп музика. Немає змісту проробляти всю цю роботу для поп музики, якщо 99% цієї праці надлишкові. Тому використовується шаблон, що більше підходить до певного музичного жанру. Реп, наприклад, більше сконцентрований на тексті пісень в своєму геномі, чим поп музика [7, 8].

Серед основних характеристик, що використовуються Pandora можна виокремити:

- аранжировка;
- такт;
- форма;
- гармонія;
- текст;
- мелодія;
- оркестровість;
- ритм;
- темп;
- вокал;

До переваг цього сервісу можна віднести:

- глибокий аналіз кожної композиції фахівцями;
- для кожного жанру використовується свій геном, що більше звертає увагу на ключові характеристики жанру;
- велика кількість ознак дозволяє всесторонньо описати кожну пісню;
- отримані ознаки непогане підґрунтя для кластеризації;

До недоліків можна віднести:

- додавання нових пісень до сервісу потребує обробки їх людьми;
- важко порівнювати пісні різних жанрів, адже їх геноми можуть не збігатися

## 1.6 Word2Vec

Відносно нещодавно, а саме у 2013 році було запропоновано новий метод створення відображення слів - word2vec [9, 10]. Цей метод дозволяє отримати відповідні словам вектори, що розташовуються у векторному просторі таким чином, що слова, які часто зустрічаються в контексті, будуть знаходитися в цьому векторному просторі ближче. Таким чином вирішується доволі давня проблема відображення слів у векторних просторах - втрата семантичного змісту. Вектори, що отримані за допомогою word2vec несуть набагато більше інформації про дане слово, що робить такі вектори ідеальними вхідними даними для алгоритмів кластеризації, класифікації та інших. Навіть більше - такі вектори без додаткової обробки вже формують доволі сформовані кластери лише на основі контекстної інформації, в якій знаходяться слова.



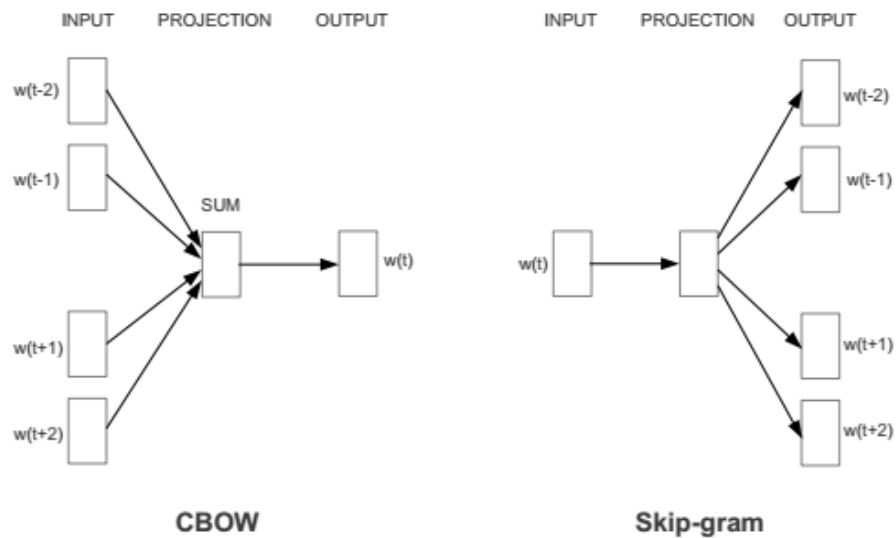


Рисунок 1.5 – Схематичне зображення моделей для CBOW та Skip-gram

В якості класифікатора використовується простий softmax регресор, проте оптимізується спеціальна функція втрат - Noise Contrastive Estimation (NCE), її зміст полягає в тому, що близькі за змістом слова штрафуються менше, ніж далекі. [6, 7, 8]. Хоча цей метод був спочатку розроблений спеціально для слів, він достатньо гнучкий, щоб узагальнити його на будь-що. Якщо говорити про застосування цього методу для пошуку схожих музичних композицій, то замість класифікатора можна використовувати якийсь алгоритм, що отримує проміжні ознаки, а контекст вибирати в межах однієї композиції, протиставляючи багатьом випадково обраним композиціям.

Переваги методу:

- отримані вектори легко кластеризуються;
- вектори будуть мати схоже семантичне значення;
- надає добрі ознаки для інших алгоритмів

Недоліки:

- в контексті пошуку схожих музичних композицій потрібно використовувати алгоритм, який попередньо виокремить ознаки із звуку.



## 1.7 Shazam

Shazam це послуга, яка може ідентифікувати музику, фільми, рекламу і телевізійні шоу, на основі короткого аудіо фрагменту за допомогою мікрофона на пристрої. Програмне забезпечення доступне для Windows, MacOS і смартфонів. Shazam Entertainment Limited була заснована в 1999 році Крісом Бартоном, Філіпом Інґелбрчтем, Ейвері Вангом і Дхіраджем Мукержі.

У грудні 2013 року, згідно з його генеральним директором, Shazam був один з десяти найбільш популярних додатків в світі.

Станом на серпень 2014 року, додаток Shazam мав понад 100 мільйонів активних користувачів щомісяця і був використаний на більш ніж 500 мільйонах мобільних пристроїв.

У жовтні 2014 року Shazam оголосив, що його технологія була використана для ідентифікації 15 мільярдів пісень.

У жовтні 2016 року, Shazam оголосила, що його мобільний додаток було завантажено понад одного мільярду разів, користувачі виконали більше тридцяти мільярдів пошуків з моменту запуску.

В роботі свого алгоритму Shazam використовує аудіо відбиток – коротка репрезентація фрагменту звуку, що використовується для ідентифікації аудіо [11, 12]. Схематично робота алгоритму відповідно до їх статті виглядає так:

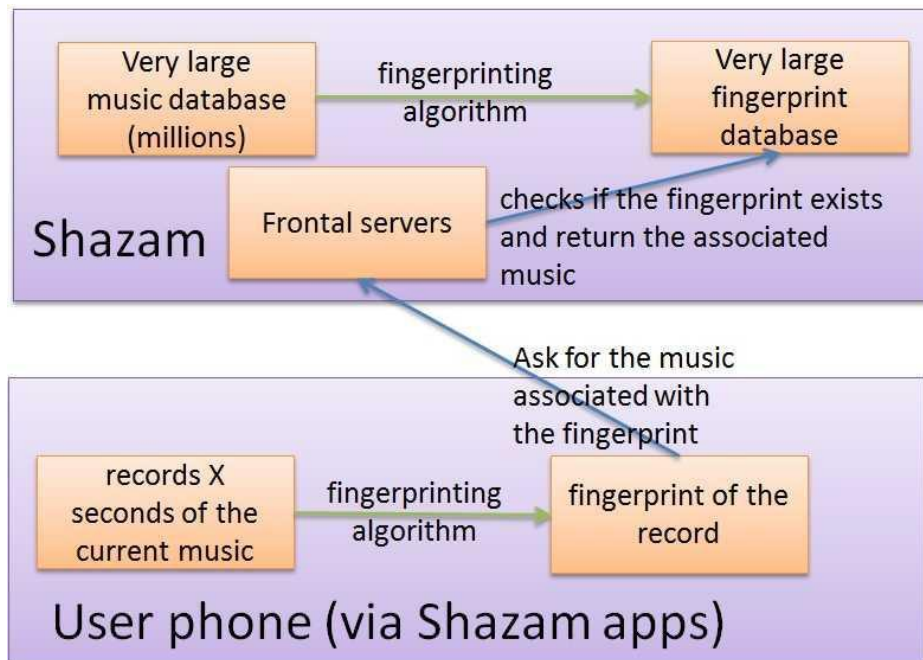


Рисунок 1.6 – Схема пошуку аудіо, що використовується в Shazam

З рисунку видно, що на стороні сервера відбуваються такі процеси:

- для своєї музичної колекції Shazam обраховує аудіо відбитки та додає до бази даних;
- для кожної нової пісні в базі даних додається відбиток, який, в свою чергу додається до бази даних.

На стороні клієнта:

- при використанні додатку за допомогою вбудованого мікрофону записується музичний відбиток;
- на пристрої користувача обчислюється аудіо відбиток;
- цей відбиток відправляється на сервера, де порівнюється з базою даних і при знайденому співпадінні користувачу повертається назва композиції.

Для отримання аудіо відбитків використовується спектрограма. Вони використовують одноканальне аудіо із частотою дискретизації 11025 Гц та вікном у 1024 відліки. Таким чином кожне вікно покриває 0.1 секунд аудіо. Shazam використовує лише найгучніші ноти для аналізу. Проте через ряд причин не можна залишати лише найбільш потужні частоти через наступні причини:

— людське вухо легше розпізнає середні частоти (500 – 2000 Гц) ніж низькі (до 500 Гц) або високі (більше 2000 Гц), як результат часто штучно збільшують гучність низьких частот. Таким чином, якщо брати лише найпотужніші частоти, низькі частоти ударних інструментів перекриватимуть інші інструменти;

— через спектральні витоки поряд з основною частотою можуть з'являтися інші потужні частоти, серед яких потрібно буде виокремити головну.

Тому частотну ділянку ділять на логарифмічні відрізки:

- дуже низькі частоти від 0 до 10 ;
- низькі частоти від 10 до 20;
- низько-середні частоти від 20 до 40;
- середні частоти від 40 до 80;
- середньо-високі від 80 до 160;
- високі частоти від 160 до 511;

Для кожної ділянки виокремлюють лише найпотужніші частоти. Після цього обраховують середнє значення для кожного проміжку. В результатуючому відбитку залишаються лише частоти, що більше обчисленого середнього помноженого на якийсь коефіцієнт. Дане перетворення можна побачити на зображеннях нижче:

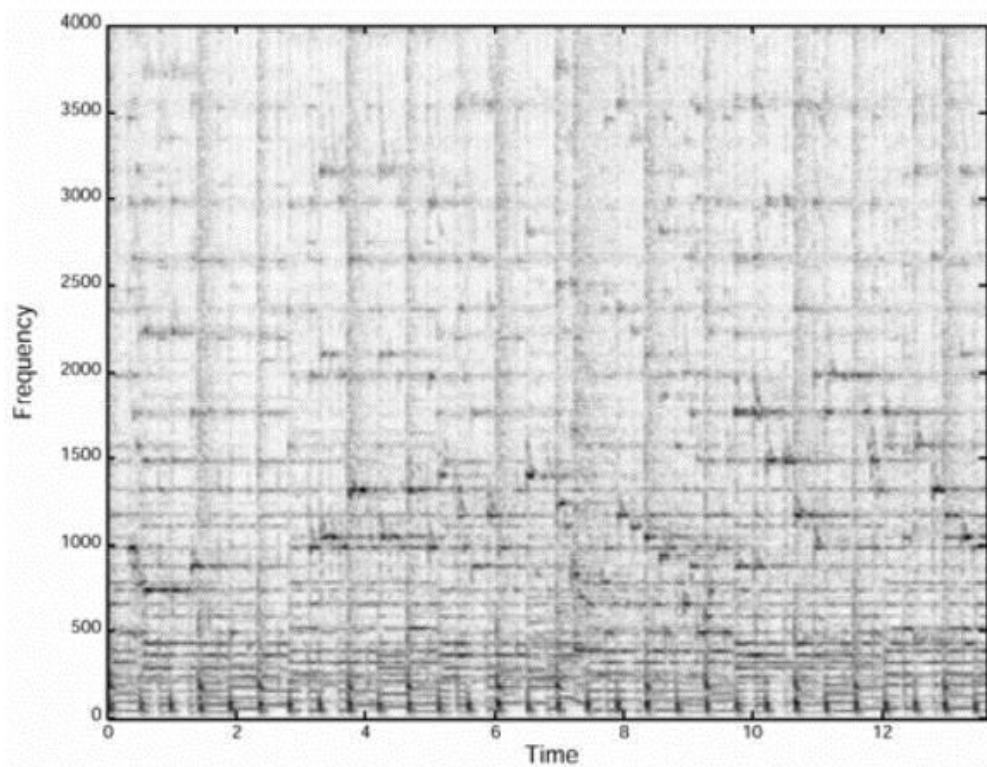


Рисунок 1.7 – Початкова спектрограма

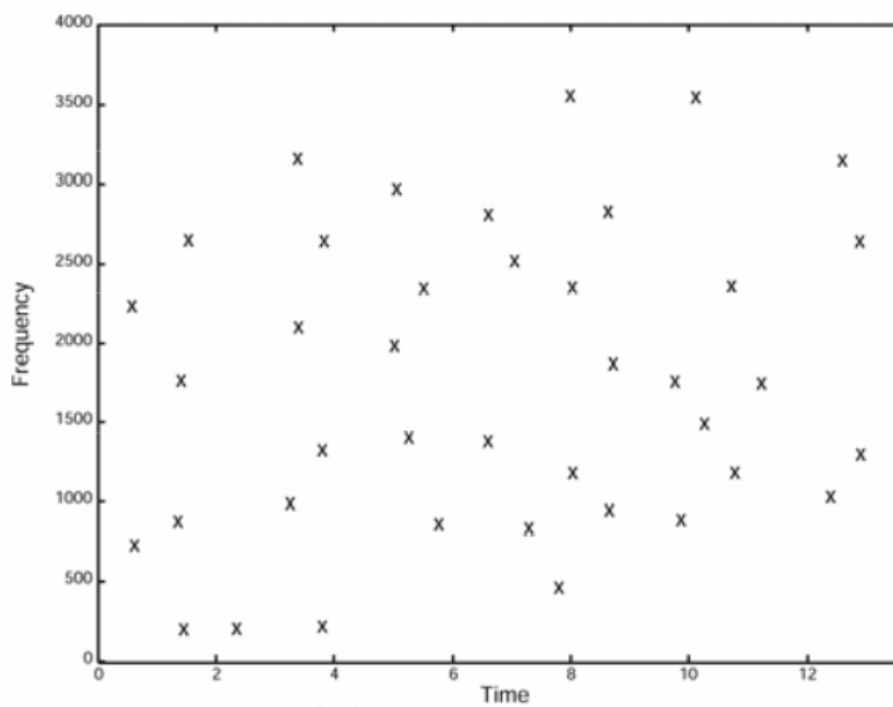


Рисунок 1.8 – Фільтрована спектрограма

Отримана фільтрована спектрограма і є аудіо відбитками. Головний недолік алгоритму Shazam є те, що він здатен шукати лише точні співпадіння пісень,

проте немає можливості використовувати його для оцінки схожості різних композицій.

## 3 РОЗРОБКА МОДЕЛІ

### 3.1 Розробка функціональних вимог до системи

Система розробляється як музичний стрімінговий сервіс із просунутою системою рекомендацій. Функціональні вимоги включають можливість прослуховування та пошуку музичних композицій, перегляд пов'язаної інформації. Більш докладніше функціональні вимоги можна переглянути нижче:

- користувач повинен мати змогу шукати музичні композиції по назві альбому, імені автора, назві композиції;
- користувач повинен мати змогу обирати та прослуховувати аудіо;
- користувач може переглядати власні плейлисти та рекомендації на головній сторінці;
- користувач може шукати схожі музичні композиції, читаючи аудіо з мікрофону або з файлу;
- користувач може формувати персоналізовані автоматичні листи на основі пошукових видач;
- користувач може створювати власні плейлисти, додавати аудіо в обране;
- користувач може сплатити підписку, щоб стати преміумним користувачем;
- преміумний користувач може налаштовувати якість звуку;
- преміумний користувач може користуватися еквайзером;
- преміумний користувач може кешувати аудіо на своєму пристрої;
- рекламодавець може розмістити рекламний банер;
- рекламодавець може розмістити аудіорекламу;
- рекламодавець може видалити аудіорекламу;
- рекламодавець може видалити банер;

- контент менеджер може додавати виконавця, альбом, трек та інформацію про них;
- контент менеджер може редагувати інформацію про виконавця, альбом, тощо;
- контент менеджер може видаляти записи по вимогам правовласника

### 3.2 Розробка нефункціональних вимог до системи

Основні нефункціональні вимоги наведено нижче:

- стримінг аудіо повинен відбуватися не нижче 128 кбіт/с при частоті дискретизації 44100 Гц, за умови, що пропускна здатність мережі дозволяє це;
- точність класифікації мережі для 10 жанрів повинна бути не нижче 80%;
- система повинна мати змогу працювати якнайменше із такими форматами даних, як WAV, MP3, FLAC, ALAC, AAC, OGG, APE;
- індексація одного запису повинна відбуватися не більше півсекунди;
- система повинна обробляти аудіо із частотою дискретизації 22050 Гц;
- формування та видача пошукового запиту повинна займати не більше секунди;
- аудіореклама повинна програватися між треками не частіше ніж раз в час і не довше 10 секунд.

### 3.3 Загальний підхід до поставленої задачі

Для вирішення поставленого завдання було вирішено використовувати методи машинного навчання, оскільки вони розробляються спеціально для задач, де не завжди можливе знайти точне рішення проблеми. В даній роботі було використано одразу два підходи машинного навчання – навчання з учителем та

без учителя. Навчання з учителем проводилося згідно із принципом максимізації правдоподібності з перехресною ентропією як функцією помилки. Навчання без учителя відбувалося методом контрастивної предиктивної, оцінки, суть якого навчитися кодувати сигнал так, щоб коди двох різних сигналів можна було легко відрізнити простим дискримінатором. Було розроблено та порівняно декілька моделей. Остаточне рішення використовує або одну нейромережу для класифікації жанрів, або ансамбль із двох мереж - одна для класифікації - друга допоміжна та навчена без учителя. Головну оцінку схожості виконує мережа для класифікації, а навчена без учителя - допоміжна. Емпірично було встановлено, що в разі, коли перші 50 композицій, передбачені цими двома моделями мають якесь об'єднання, композиції, що входять до нього, суб'єктивно дуже схожі їх можна видати як першими результатами пошуку. У разі обмежень по пам'яті чи швидкості, використовується лише модель для класифікації.

### 3.4 Проста згорткова модель

Згорткові мережі – спеціальний клас нейромереж, які були створені спеціально для обробки зображень. В їх основі полягає припущення, що на зображенні один і той самий об'єкт можна розпізнати в різних місцях використовуючи одні й ті самі ваги. Тому в згорткових мережах ваги поєднані лише з певною обмеженою ділянкою вхідного сигналу, після підрахунку активації ця ділянка зсувається і знову обраховується активація. Іншими словами, до вхідного сигналу застосовується операція згортки з певним фільтром. Для дискретної функції  $F$  та дискретного фільтра  $k$ , дискретна згортка визначається як

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t) \quad (3.1)$$



Як нелінійність між шарами використовувалась кусково-лінійна функція випрямляч (ReLU):

$$z = f(x) = \max(x, 0) \quad (3.2)$$

Оскільки експерименти проводилися і для класифікації, і для навчання без учителя, то в залежності від задачі, використовувалися різні шари. Це пов'язано з тим, що якщо використовувати для класифікації останній шар, то багато інформації губиться через те, що мережа передбачає тільки клас і нехтує якоюсь додатковою інформацією.

Точні параметри простої згорткової мережі можна побачити в таблиці 3.1

Таблиця 3.1 - Точні параметри простої згорткової мережі

Блок	Параметри
Згортковий	Фільтр:10 Зсув:5 Кількість фільтрів: 128
Згортковий	Фільтр:8 Зсув:4 Кількість фільтрів: 128
Згортковий	Фільтр:4 Зсув:2 Кількість фільтрів: 128
Згортковий	Фільтр:4 Зсув:2 Кількість фільтрів: 128
Згортковий	Фільтр:4 Зсув:2 Кількість фільтрів: 128
Повнозв'язний	Фільтр:1 Кількість фільтрів: 128
Повнозв'язний	Фільтр:1 Кількість фільтрів: 10 (кількість жанрів в наборі даних)
Загальна кількість параметрів	614 726

Точну структуру мережі можна побачити в додатку Б

Мережа складається з 5 блоків з різними розмірами фільтрів та зсувів, що починаються із більших значень і поступово зменшуються. Усього через зсуви розміром більше 1 досягається зменшення довжини вхідного сигналу у 160 разів. Для частоти дискретизації 22050 Гц, що використовувалась, це відповідає 7 мс. Дану мережу було протестовано на аудіохвилі та мел-спектрограмі.

### 3.5 Модель із конволюційними блоками та рекурентністю

Ідея даної мережі полягає у використанні апріорних знань про структуру сигналу. Так, кожна людина, навіть без музичної освіти, розуміє, що музика складається із поєднання нот, що змінюються із часом. Тому природним є бажання побудувати модель із урахуванням цієї структури. Для цього використаємо такі припущення про необхідну структуру:

1. Модель повинна спочатку підсумовувати співвідношення між різними частотами
2. Модель повинна підсумовувати швидкі зміни у сигналі
3. Модель повинна враховувати глобальну структуру сигналу.

Для того, щоб виконати ці умови, 3-ох шарова мережа. Перший шар складається із блоку згорткових блоків, подібних до блоків, що використовуються в архітектурі Insertion для класифікації зображень. Ідея в тому, щоб на одному і тому самому вхідному сигналі використати згортки з фільтрами різного розміру. Це дозволяє згорткам вивчати різні залежності в одних і тих самих даних. Проте, щоб зменшити кількість параметрів, замість збільшення розміру фільтру було використано збільшення коефіцієнта розширення. Особливістю згортки із збільшеним коефіцієнтом розширення є застосування фільтру, в якому між елементами з певною періодичністю  $d$  ставлять нулі. Таким чином, без збільшення параметрів легко нарощується поле зору фільтру, що є важливим показником нейронної мережі. Рівняння для розширеної згортки можна переписати так:

$$(F_d * k)(p) = \sum_{s+dt=p} F(s)k(t) \quad (3.3)$$

Традиційна згортка є частковим випадком розширеної згортки з коефіцієнтом розширення  $d=1$ . Інтуїтивно, набір блоків із різним коефіцієнтом розширення повинен навчитися розрізняти різні співвідношення між частотами. Таким чином, вивчені згортки повинні навчитися розрізняти різні типи акордів та інших музичних структур. Аналогічним чином побудований другий набір блоків мережі, проте замість згортки по частотам, вони роблять згортку по часу. Це повинно допомогти розпізнати швидкі чи плавні зміни сигналу, які в свою чергу є виходом попередньої мережі, яка може розпізнавати акорди. Таким чином, дана мережа може розпізнавати швидкі прогресії акордів. Проте, оскільки поле зору другого шару обмежено величиною найбільшого фільтра, дана мережа обмежена для розпізнавання довгих послідовностей, тому вихід другого шару подається рекурентній мережі. В даній роботі були використані вентиляльні рекурентні вузли (GRU (gated recurrent unit))[13]. Люди не починають думати з нуля кожену секунду, коли ми читаємо, ми розуміємо, про що йде мова, базуючись на попередніх словах. Традиційні нейронні мережі прямого розповсюдження не можуть вирішувати таку задачу. Для цього було створено рекурентні нейронні мережі. Основна їх ідея полягає в тому, що для аналізу послідовностей необхідно мати пам'ять, у яку мережа може записувати інформацію про минулі події. Щоб задовольнити ці вимоги рекурентна мережа має два окремі входи та виходи. Один вхід та вихід не відрізняється від звичайних мереж і слугують для передачі фрагменту вхідного сигналу та отримання передбачення. Інший вхід використовується як інформація про минуле, а вихід слугує як місце для запису інформації про теперішнє для наступного кроку. Якщо вхідний фрагмент сигналу на кроці  $t$  позначити, як  $x$ , а вихід мережі позначити як  $h$ , то схематично шар рекурентної мережі  $A$  можна зобразити як на рисунку 3.2:

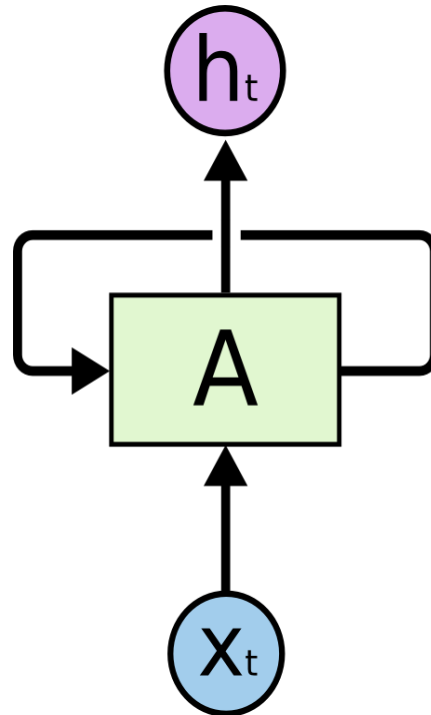


Рисунок 3.2 - Шар рекурентної мережі

Стрілка, що іде циклічно - пам'ять мережі, що переноситься та трансформується самою мережею. Через це з'єднання мережі і називаються рекурентними. Навчання таких мереж не відрізняється від класичних, оскільки будь-яку рекурентну мережу можна розгорнути як послідовність нейронних мереж прямого розповсюдження, що показано на рисунку 3.3.

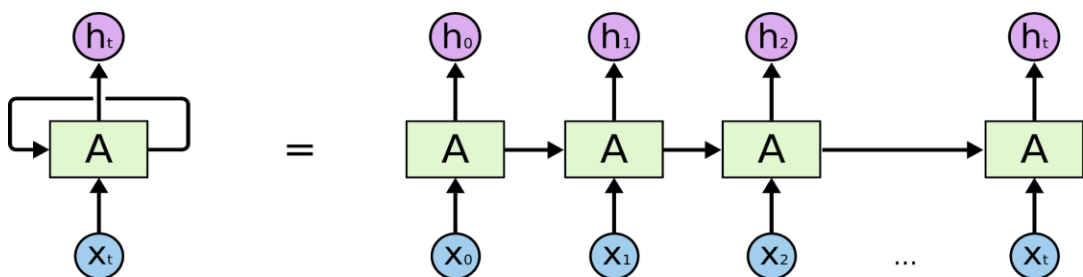


Рисунок 3.3 - Розгорнута рекурентна нейронна мережа

Єдиною різницею є те, що параметри блоків однакові та, оскільки розгортати дуже довгі послідовності доволі коштовно і по часу і по пам'яті,

мережа або розгортається частково, або використовуються короткі послідовності для навчання. У класичному рекурентному блоці використовуються такі функції:

$$s_t = \tanh(Ux_t + Ws_{t-1}) \quad (3.4)$$

$$h_t = Vs_t$$

де:

1.  $s_t$  - пам'ять мережі (яку ще називають прихованим станом)
2.  $s_{t-1}$  - стан мережі на минулому кроці
3.  $U, W, V$  - матриці вагів для передбачення
4.  $h_t$  - передбачення на даному кроці
5.  $x_t$  - вхідний сигнал на даному кроці

Схематично на малюнку це виглядає, як на рисунку 3.4.

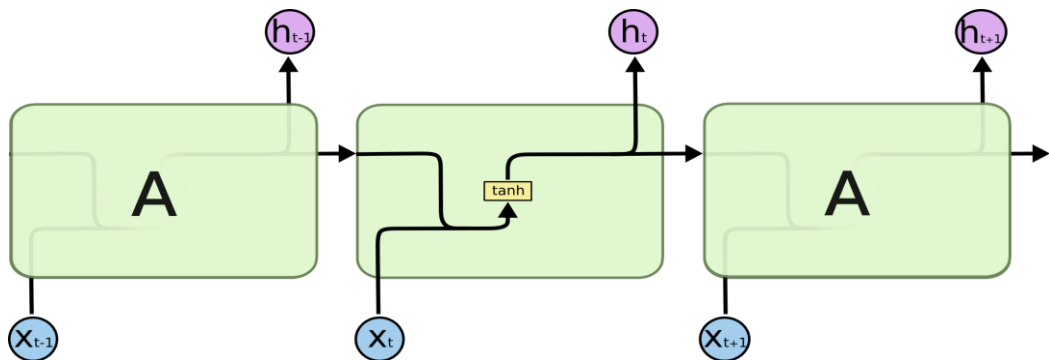


Рисунок 3.4 - Структурна схема класичного рекурентного блоку

Хоча в теорії такий блок може передбачати та запам'ятовувати нескінченно великі послідовності, на практиці такого не відбувається. Крім того, даний шар страждає від зникаючого градієнту, що робить його навчання доволі складним. Для подолання цих недоліків було розроблено інші архітектури, а саме LSTM та GRU. GRU було обрано, оскільки він показав високу якість у завданнях моделювання мови та музики. Структурно цей блок виглядає як на рисунку 3.5

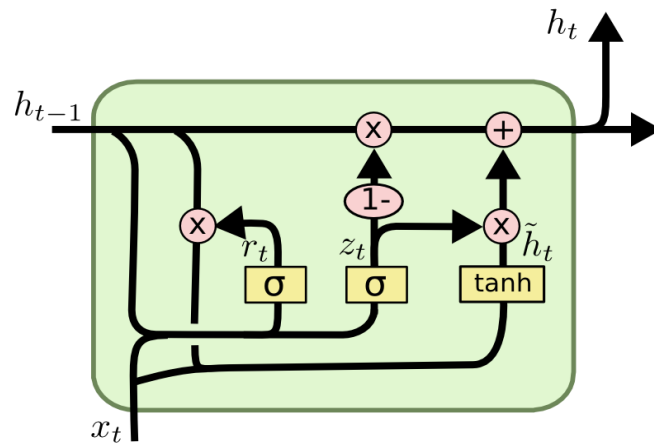


Рисунок 3.5 - Структурна схема вентиляних рекурентних вузлів

У рівняннях це виглядає наступним чином:

$$\begin{aligned}
 z_t &= \sigma(W_z[h_{t-1}, x_t]) \\
 r_t &= \sigma(W_r[h_{t-1}, x_t]) \\
 \hat{h}_t &= \tanh(W[r_t \odot h_{t-1}, x_t]) \\
 h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t
 \end{aligned} \tag{3.5}$$

де  $\odot$  - поелементне множення. В даному випадку  $z$  - оновлюючий вентиль, що вирішує, яка інформація у новому виході мережі оновиться або залишиться,  $\hat{h}$  - кандидат виходу мережі, а  $r$  - скидуючий вентиль, що визначає, скільки інформації вихід-кандидат отримає з минулих кроків.

Крім рекурентної мережі, перші два шари використовували залишкові з'єднання.

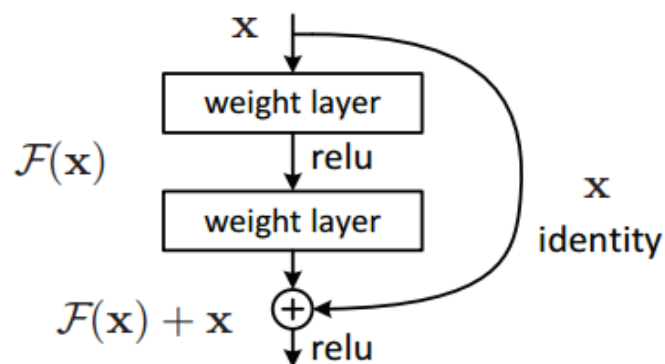


Рисунок 3.6. – Приклад залишкового з'єднання

Таке з'єднання було запропоноване для уникнення проблеми деградації градієнту для дуже глибоких нейронних мереж. Таке з'єднання дозволяє легко поширювати градієнт через мережу, підвищуючи швидкість навчання і узагальнюючу здатність мережі. Повну структуру мережі можна переглянути у таблиці 3.2.

Таблиця 3.2 - Повна структура мережі

Блок	Опис	Параметри
Блок 1	5 згорток по частотним каналам	Фільтри: 3 Кількість фільтрів: 16 Розширюючий коефіцієнт: 1, 2, 4, 8, 16
Блок 2	5 згорток по часу	Фільтри: 3 Кількість фільтрів: 16 Розширюючий коефіцієнт: 1, 2, 4, 8, 16
Рекурентний	GRU	Розмір стану та решіток: 128
Повнозв'язний		Фільтр: 1 Кількість фільтрів: 128
Повнозв'язний		Фільтр: 1 Кількість фільтрів: 10 (кількість жанрів в даних)
Загальна кількість параметрів		351 668



Структурну схему даної мережі можна переглянути в додатку В.

Оскільки ідея за цією архітектурою полягає у знаходженні співвідношень у частотах, то експерименти проводилися на мел-спектрограмах та мел-кепстральних коефіцієнтах.

### 3.6 Класифікація жанрів

Музичний жанр – багатозначне поняття, що характеризує музичну творчість з огляду на її походження, умови виконання, зміст, структуру, засоби виразності, склад виконавців тощо. Для розпізнавання музичних жанрів модель необхідна мати глибоке поняття про музику. Мережі навчались згідно із принципом максимізації правдоподібності. До останнього шару мережі застосовується *softmax* функція:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=0}^I e^{x_j}} \quad (3.6)$$

Ця функція дозволяє наближувати розподіл вірогідностей певної дискретної випадкової величини. Наприклад:

$$P(y = i|x) = \text{softmax}(x_i) \quad (3.7)$$

Ця функція є диференційованою і використовується разом із перехресною ентропією для задач класифікації.

Перехресна ентропія – функція втрат, що визначена як:

$$H(y, x) = -\sum p(y) \log(p(x)) \quad (3.8)$$

де  $p(y)$  – унітарний код об'єкта певного класу,  $p(x)$  – розподіл вірогідностей класів для об'єкта  $x$ .

Оскільки музичний жанр – характеристика музики, що полягає у певних композиційних та стилістичних ознаках, таких як інструментальна та ритмічна структура, характеристика вокалу виконавця, тощо, то вивчення вагами нейронної мережі ознак, що розрізняють жанри є доволі потужною властивістю, що дозволить надалі віднаходити схожі композиції. Було проведення навчання мережі з двома типами музичних ознак: нормованих значень амплітуд аудіохвилі та кепстральних мел-коефіцієнтів. Значення амплітуд зазвичай зчитуються і декодуються із аудіо файлів напряму. Кепстральні коефіцієнти отримуються наступним чином:

- Аудіофайл розбивається на відрізки
- До цих відрізків застосовують Хеммінгове вікно
- Застосовується швидке перетворення Фур'є
- Частоти у герцах шкалюють на мел-шкалу, що майже лінійна в області низьких частот і логарифмічна в області високих (це більше відповідає сприйняттю звуків людиною)
- До отриманого спектра застосовується косинусне перетворення.

В роботі було використано кадри довжиною 23мс та зміщенням 6мс та 40 перших коефіцієнтів.

В якості тренувальної вибірки використовується датасет GTZAN. В ньому міститься 1000 музичних записів 10 жанрів, по 100 пісень в одному жанрі. Частота дискретизації 22050 відліків на секунду. Кожна пісня довжиною 30 секунд. В роботі 800 пісень було використано для навчання, 200 для тестування нейронної мережі.

Усі експериментальні налаштування можна переглянути в таблиці 3.3

Таблиця 3.3 - Співвідношення типу моделі до типу ознак

Тип моделі	Використані ознаки
Проста згорткова модель	Аудіохвиля
Модель із конволюційними блоками та рекурентністю	Мелспектрограми
Модель із конволюційними блоками та рекурентністю	Мел-кепстральні коефіцієнти

Падіння функції втрат та зростання точності можна побачити на рисунку 3.6.



Рисунок 3.6 - Згори - функція помилки - всі 3 моделі сходяться приблизно до одного значення, знизу - точність на тестових даних - два верхніх графіки моделі із конволюційними блоками та рекурентністю, нижній - проста згорткова модель. Видно, що моделі із конволюційними блоками та рекурентністю значно краще за просту модель

Точні значення функції втрат та точності і, порівнянні із WaveNet [14] для класифікації, можна побачити в таблиці 3.4

Таблиця 3.4 - Точні значення функції втрат та точності

Модель	Ознаки	Функція втрат	Точність
Проста згорткова модель	Аудіохвиля	1.083	69%
Модель із конволюційними блоками та рекурентністю	Мел-кепстральні коефіцієнти	0.83	81%
Модель із конволюційними блоками та рекурентністю	Мелспектрограма	0.907	80%
WaveNet	Мел-кепстральні коефіцієнти	-	46.3%

Як видно з таблиці, найкращі результати дають моделі із конволюційними блоками та рекурентністю, і мають приблизно однакову точність. Цікаво відмітити, що всі моделі сильно обганяють у якості WaveNet для класифікації. Хоча WaveNet є доволі потужною моделлю, яка дозволяє генерувати високоякісну мову, велика кількість параметрів (близько мільйона) на невеликій кількості даних призводить до отримання низької якості класифікації. Усі розроблені мережі тренувалися протягом 40 000 ітерацій, що відповідає 40 епохам для даного набору даних. Щодо часу навчання, його можна побачити в таблиці 3.5

Таблиця 3.5 - Час навчання різних мереж

Модель	Ознаки	Час тренування
Проста згорткова модель	Аудіохвиля	35хв, 8с
Модель із конволюційними блоками та рекурентністю	Мел-кепстральні коефіцієнти	1 год 52 хв
Модель із конволюційними блоками та рекурентністю	Мелспектрограма	1 год 53 хв

З таблиці видно, що, незважаючи на меншу кількість параметрів, моделі з блоками тренуються в 1.5 рази довше. Це пов'язано із використанням рекурентного шару, оскільки кожне нове передбачення залежить від попередніх, то цей шар не можна ефективно виконати паралельно. Врешті-решт, з точки зору якості все одно краще використовувати саме їх. Результати кластеризації цих моделей можна побачити на рисунках нижче.





Рисунок 3.7 - T-SNE відображення результати кластеризації моделі, навченої на мелспектрограмах

З рисунків видно, що хоча кластери не дуже яскраво виділені, але все-таки присутні. Особливо добре відокремлюються кластери жанрів метал та класика.

### 3.7 Контрастивне предиктивне кодування

Покращення якості репрезентацій потребує ознак, які більше спеціалізовані для кодування загальної інформації, ніж для класифікації. Наприклад для зображень, ознаки отримані моделям для класифікації гарно переносять знання, проте вони можуть втрачати таку інформацію, як колір, специфічні форми, вміння

рахувати або сегментувати [15]. У випадку із звуком класифікація фонем навряд чи допоможе класифікувати музику. Таким чином, основних задач глибокого навчання є вивчення корисних репрезентацій, загальна ідея методів, що намагаються вирішити дану задачу — тренування спеціальної функції-відображення (яку також називають енкодером) , яка максимізує взаємну інформацію між вхідними даними та результатом застосування функції-енкодера. Зазвичай взаємну інформацію досить важко обчислювати, особливо для високорозмірних та неперервних розподілів. Одна із найпопулярніших стратегій для навчання без учителя - передбачення майбутнього, пропущених або контекстних даних. Такі техніки називають предиктивним кодуванням, і є одними із перших технік в теорії обробки сигналів для компресії даних. У нейронауках багато теорій стверджують, що мозок передбачає свої спостереження на різних рівнях абстракції. Метод для створення репрезентацій слів у word2vec також є окремим випадком предиктивного кодування. Такі підходи корисні, оскільки для передбачення контексту зазвичай потрібно моделювати розподіл умовної вірогідності даних відносно якихось прихованих змін. Основна ідея методу контрастивного предиктивного кодування полягає у вивченні репрезентацій зі спільною інформацією різних частин сигналу. Під час моделювання послідовностей, підходи, що моделюють наступний крок, використовують локальні характеристики сигналу. Зі збільшенням відстані у часі модель повинна вивчити більш глобальну структуру. Ці "повільні ознаки" , які охоплюють багато етапів часу, часто є більш цікавими (наприклад, фонемі та інтонація в мові, об'єкти у зображеннях або сюжети в книгах). Однією з перешкод прогнозування високорозмірних даних є те, що унімодальні функції втрат, такі як середньоквадратична помилка та крос-ентропія, не є дуже корисними, і зазвичай потрібні потужні умовні генеративні моделі, які потребують реконструкції кожної деталі в даних. Але ці моделі є обчислювально інтенсивними, а втрачають ємність при моделюванні даним, не зважаючи на контекст. Наприклад, зображення можуть містити тисячі бітів інформації, тоді як високорівневі змінні, такі як мітка класу, містять набагато менше інформації (10 біт для 1024 категорій). Це



припускає, що моделювання  $(x | c)$  безпосередньо не може бути оптимальним для вивчення загальної інформації між  $x$  та  $c$ . Прогнозуючи майбутнє, ми замість цього кодуємо ціль  $x$  (майбутнє) та контекст  $c$  (теперішнє) у компактні розподілені векторні представлень (через нелінійні вивчені відображення) таким чином, що максимально зберігаємо взаємну інформацію про вихідні сигнали і визначаємо її як:

$$I(x; c) = \sum_x p(x, c) \log \left( \frac{p(x, c)}{p(x)} \right) \quad (3.9)$$

Максимізуючи спільну інформацію між енкодером (яка пов'язана із спільною інформацією входів енкодера), можна виокремити приховані змінні.

У даному методі наступні передбачення  $x_{t+1}$  не передбачаються напряму генеративною моделлю  $p_k(x_{t+k} | c_t)$ . Замість цього моделюється відношення щільностей імовірностей, яка зберігає спільну інформацію між  $x_{t+k}$  та  $c_t$

$$f_k(x_{t+k}, c_t) \simeq \frac{p(x_{t+k}, c_t)}{p(x_{t+k})} \quad (3.10)$$

Хоча будь-яка позитивна функція може бути використана, було використано таку експоненційну функцію

$$f(x_{t+k}, c_t) = \exp(z_{t+k}^T W_k c_t) \quad (3.11)$$

Лінійна трансформація  $W$  використовується для передбачень із різними  $W$  для різних  $k$ . Використовуючи відношення щільностей  $f(x_{t+k}, c_t)$  та обчислюючи енкодером  $z_t$  ми уникаємо необхідності моделювати високорозмірні дані напряму. Хоча ми не можемо обчислити  $p(x)$  та  $p(x|c)$  напряму, ми можемо робити вибірку з цих розподілів та використовувати контрастивну оцінку за шумом, які

дозволяють оцінювати цільове значення із випадково обраним. Для репрезентацій можна використовувати  $z$  і  $c$ ,  $c$  може бути корисне для завдань, де додаткова інформація з минулого може бути корисною. Енкодер та декодер тренуються одночасно для оптимізації функції втрат, заснованої на контрастивній оцінці за шумом. Маючи вибірку даних  $X = \{x_1, \dots, x_n\}$  із  $N$  випадкових спостережень, маючи 1 позитивне спостереження  $p(x_{t+k} \vee c)$  та  $N-1$  негативних із “шумного” розподілу  $p(x_{t+k} \vee c)$ , ми оптимізуємо негативну контрастивну оцінку

$$L_N = -E \left[ \log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right] \quad (3.12)$$

Якість моделей, навчених таким чином, можна оцінити лише непрямыми способами. До таких можна віднести порівняння значень функції помилки та візуалізацію представлень.

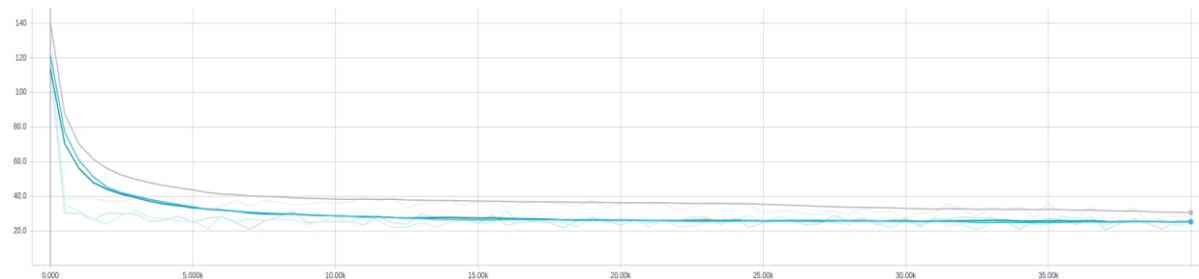


Рисунок 3.8 - Функції помилок, верхня - для простої згорткової моделі, нижні - згорткові моделі з блоками та рекурентністю

Таблиця 3.6 - Значення функції втрат для різних моделей

Модель	Ознаки	Функція втрат
Проста згорткова модель	Аудіохвиля	30.65
Модель із конволюційними блоками та рекурентністю	Мел-кепстральні коефіцієнти	26.01
Модель із конволюційними блоками та рекурентністю	Мелспектрограма	24.48

Таблиця 3.7 - Час тренування різних моделей

Модель	Ознаки	Час тренування
Проста згорткова модель	Аудіохвиля	1 год 24 хв
Модель із конволюційними блоками та рекурентністю	Мел-кепстральні коефіцієнти	2 год 42 хв
Модель із конволюційними блоками та рекурентністю	Мелспектрограма	2 год 44 хв

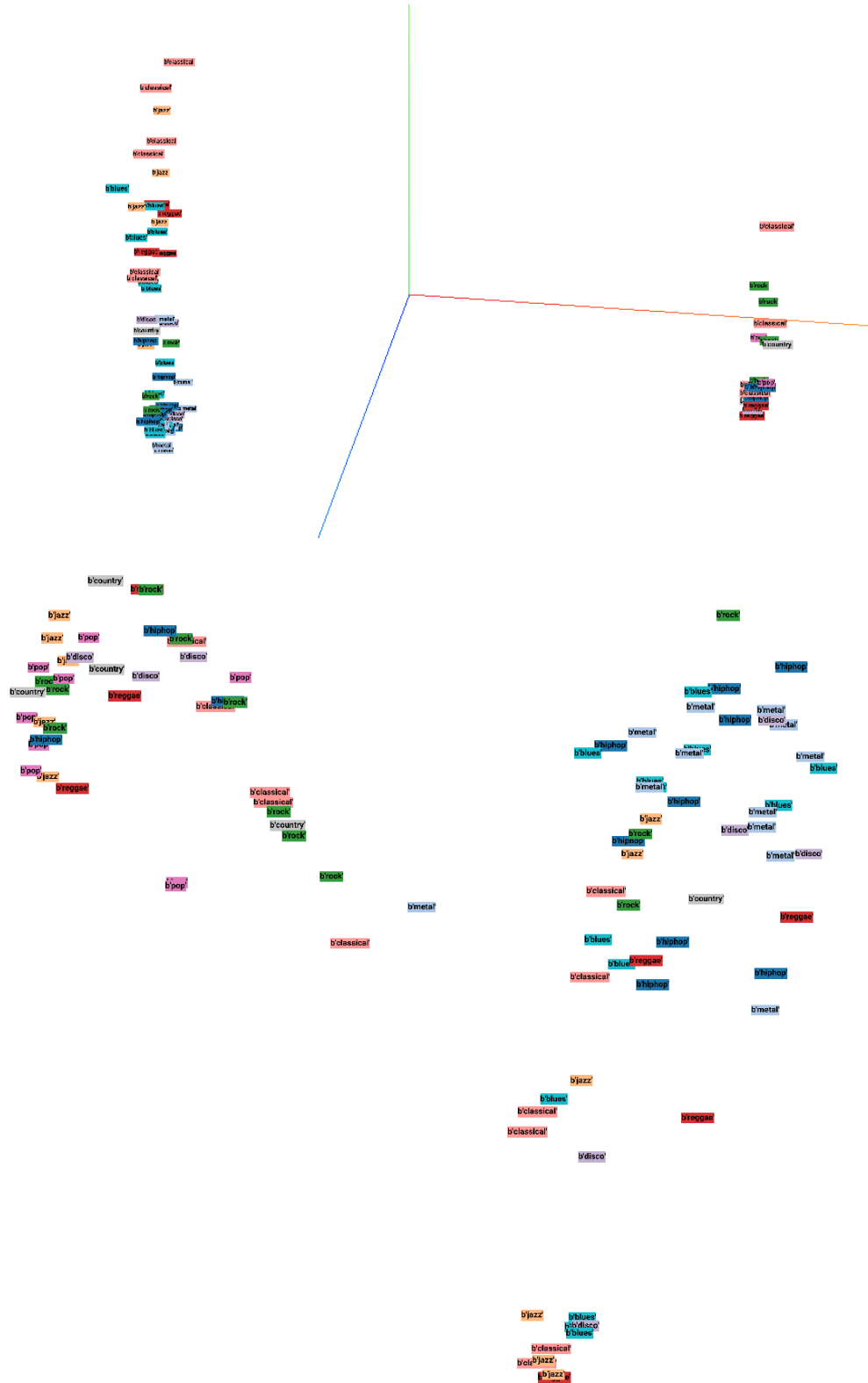


Рисунок 3.9 - Представлення  $s$  та  $z$  для простої згорткової мережі

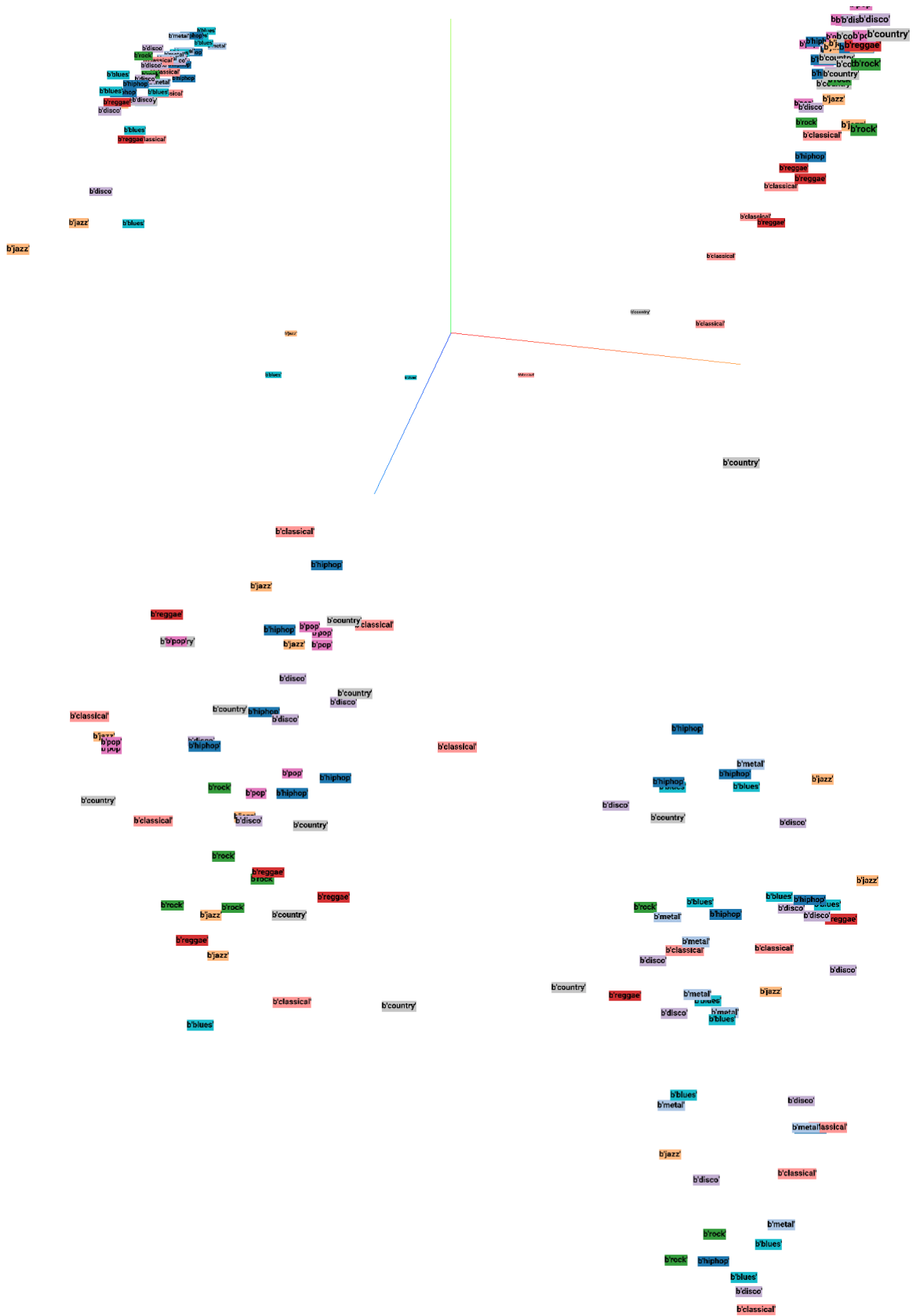


Рисунок 3.10 - Представлення  $s$  та  $z$  для згорткової мережі з блоками та рекурентністю навченної на кепстральних коефіцієнтах

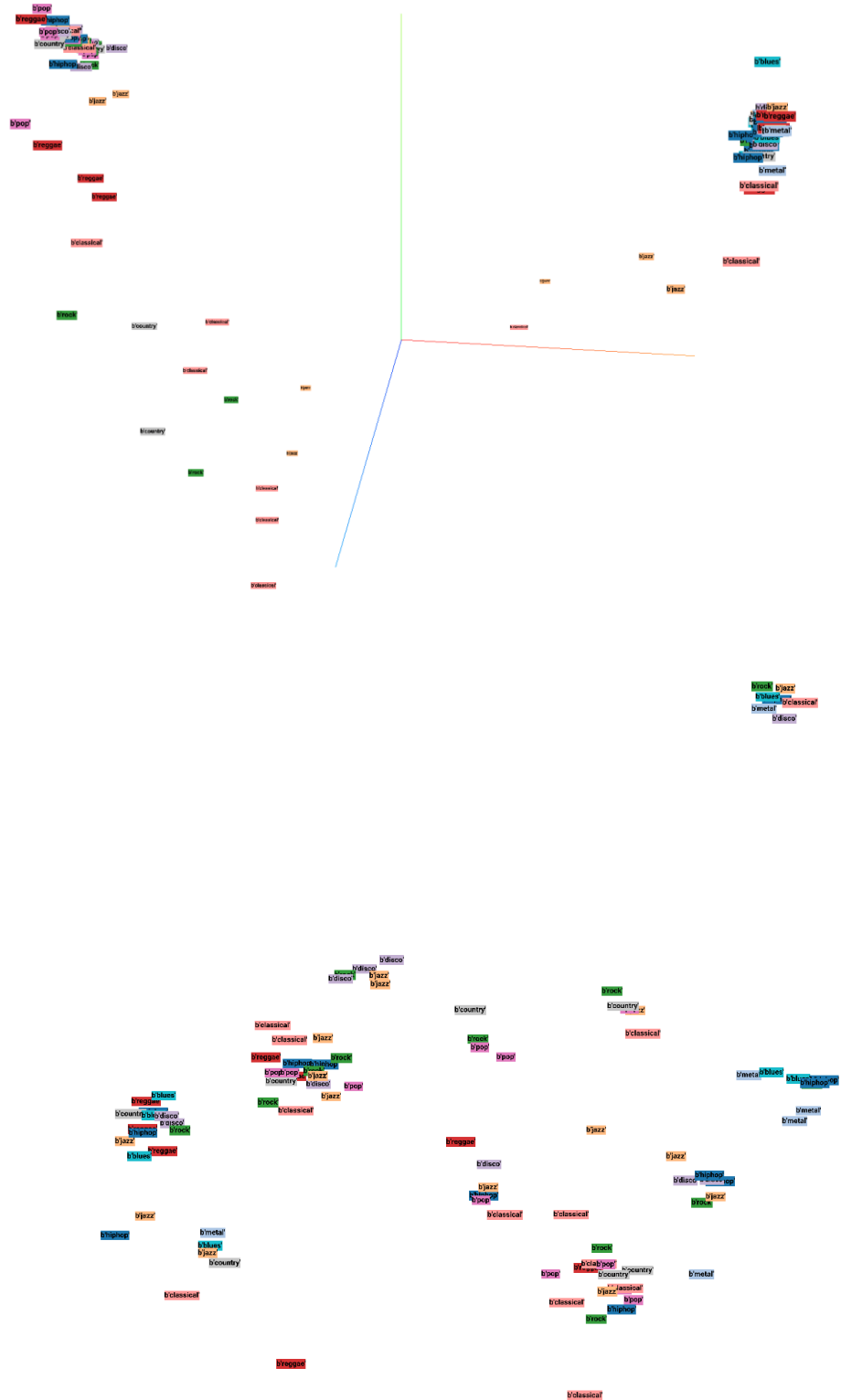


Рисунок 3.11 - Представлення  $s$  та  $z$  для згорткової мережі з блоками та рекурентністю навченної на мелспектрограмах

З рисунків видно, що отримані векторні представлення погано кластеризуються у порівнянні із представленнями отриманими з класифікації.

Серед усіх представлень суб'єктивно найкращими для оцінки схожості є вектори  $z$  отримані із моделі зі згортковими блоками та рекурентністю. Далі в роботі використовуються саме вони, разом із представленнями отриманими з класифікації такою ж моделлю.



## 4 РОЗРОБКА СИСТЕМИ

## 4.1 Сценарії використання системи

Діаграму сценаріїв із функціональними вимогами наведено у додатку Г, детальні описи прецедентів наведено у таблицях нижче.

Таблиця 4.1 - Сценарій використання

Назва	Пошук по виконавцю, альбому, назві композиції
ID	1
Опис	Користувач, використовуючи поле для пошуку, шукає аудіо, що відповідає критерію пошуку по виконавцю, альбому або назві
Актори	Користувач, Преміумний користувач
Вигоди компанії	Якщо користувачі не можуть шукати вже відому їм музику, то вони навряд чи продовжать користуватися сервісом
Частота користування	Постійно
Тригери	Користувач вводить пошуковий запит у полі для пошуку
Передумови	Пошукове поле доступне у будь-якому вікні
Постумови	Користувач потрапляє на вікно з результатами пошуку
Основний розвиток	Користувач вводить запит у пошукову строку, натискає на кнопку пошуку чи Enter
Альтернативні розвитки	-
Винятки	-

Таблиця 4.2 - Сценарій використання

Назва	Прослуховування обраних композицій
ID	2
Опис	Користувач обирає трек, який хоче прослухати і слухає
Актори	Користувач, Преміумний користувач
Вигоди компанії	Стримінговий сервіс без можливості прослуховування не буде користуватися попитом
Частота користування	Постійно
Тригери	Користувач натискає на кнопку “Грати” біля треку
Передумови	У вікні є елементи треків, які можна програти
Постумови	Музика грає
Основний розвиток	Користувач натискає на кнопку “Грати” Музика грає
Альтернативні розвитки	-
Винятки	-

Таблиця 4.3 - Сценарій використання

Назва	Перегляд головної сторінки із плейлистами та рекомендаціями
ID	3
Опис	Користувач переглядає головну сторінку із плейлистами та рекомендаціями
Актори	Користувач, Преміумний користувач
Вигоди компанії	Головна сторінка важлива для зацікавлення нових користувачів, та актуальна інформація буде утримувати старих
Частота користування	Постійно
Тригери	Користувач переходить на сайт/запускає додаток
Передумови	Немає
Постумови	Користувач потрапляє на головну сторінку
Основний розвиток	Користувач запускає додаток/переходить на сайт
Альтернативні розвитки	-
Винятки	Сервіс недоступний у країні користувача, тоді показується повідомлення про помилку

Таблиця 4.3 - Сценарій використання

Назва	Пошук композицій по наданому користувачем аудіо із мікрофону або файлу
ID	4
Опис	Користувач може знайти схожі треки, надавши запис з мікрофону або файл як запит
Актори	Користувач, Преміумний користувач
Вигоди компанії	Така стратегія сприяє кращому пошуку нової музики, ніж класичні моделі, тому повинна стати головною цікавинкою компанії
Частота користування	Невідомо, як часто ця опція буде використовуватися на практиці
Тригери	Користувач натискає на відповідні кнопки у пошуковій строці
Передумови	Користувач знаходиться на будь-якій сторінці із пошуковим запитом
Постумови	Користувач потрапляє на сторінку із пошуковими видачами
Основний розвиток	Користувач натискає на відповідні кнопки у пошуковій строці Обирає файл. Отримує пошукову видачу
Альтернативні розвитки	Користувач натискає на відповідні кнопки у пошуковій строці. Записує з мікрофону Отримує пошукову видачу
Винятки	-

Таблиця 4.4 - Сценарій використання

Назва	Формування персоналізованих плейлистів на основі результатів видачі
ID	5
Опис	Формуються персоналізовані плейлисти на основі результатів видачі
Актори	Користувач, Преміумний користувач
Вигоди компанії	Легке створення якісних нових плейлистів потрібно утримати існуючих користувачів
Частота користування	Користувачи зазвичай досить часто використовують плейлисти
Тригери	Користувач здійснив пошук
Передумови	Користувач знаходиться у вікні із результатами пошуку
Постумови	Користувач зберігає новий плейлист
Основний розвиток	У вікні із результатами пошуку користувач натискає на спеціальну кнопку
Альтернативні розвитки	-
Винятки	-

Таблиця 4.5 - Сценарій використання

Назва	Додавання композицій в плейлисти/обране
ID	6
Опис	Користувач додає композиції в плейлисти/обране
Актори	Користувач, Преміумний користувач
Вигоди компанії	Плейлисти потрібні для утримання існуючих користувачі, як необхідний функціонал
Частота користування	Часто
Тригери	Користувач натискає на відповідну кнопку
Передумови	Користувач знаходиться на сторінці із треками, обирає якийсь із них
Постумови	Трек додано до відповідного плейлисту
Основний розвиток	Користувач натискає на відповідну кнопку біля треку для додавання його до обраного
Альтернативні розвитку	Користувач натискає на відповідну кнопку біля треку для додавання. Відкривається віконце із існуючими плейлистами або варіантом створення ного, до якого додасться трек
Винятки	Спроба створити плейлист з тією ж назвою

Таблиця 4.6 - Сценарій використання

Назва	Додавання композицій в плейлисти/обране
ID	6
Опис	Користувач додає композиції в плейлисти/обране
Актори	Користувач, Преміумний користувач
Вигоди компанії	Плейлисти потрібні для утримання існуючих користувачі, як необхідний функціонал
Частота користування	Часто
Тригери	Користувач натискає на відповідну кнопку
Передумови	Користувач знаходиться на сторінці із треками, обирає якийсь із них
Постумови	Трек додано до відповідного плейлисту
Основний розвиток	Користувач натискає на відповідну кнопку біля треку для додавання його до обраного
Альтернативні розвитку	Користувач натискає на відповідну кнопку біля треку для додавання. Відкривається віконце із існуючими плейлистами або варіантом створення ного, до якого додасться трек
Винятки	Спроба створити плейлист з тією ж назвою

Таблиця 4.7 - Сценарій використання

Назва	Сплата платної підписки
ID	7
Опис	Користувач може сплатити підписку, що надає більше можливостей
Актори	Користувач, Преміумний користувач
Вигоди компанії	Платні підписки основний дохід компанії
Частота користування	Приблизно третина користувачів сплачує підписку на Spotify, проте вони генерують найбільший прибуток
Тригери	Користувач вибирає відповідну опцію в особистому кабінеті
Передумови	Користувач знаходиться в особистому кабінеті
Постумови	Користувач стає преміумним користувачем, або продовжує свою підписку, якщо він уже преміумний
Основний розвиток	Користувач вибирає відповідну опцію в особистому кабінеті, проходить процедуру оплати
Альтернативні розвитки	-
Винятки	Користувач відміняє сплату, відмова оплати банком, виводиться повідомлення про невдалу транзакцію, поповнення не відбувається



Таблиця 4.8 - Сценарій використання

Назва	Налаштування якості звуку
ID	8
Опис	Преміумний користувач може вибрати більш високу якість звуку
Актори	Преміумний користувач
Вигоди компанії	Висока якість звучання необхідна для залучення деяких клієнтів
Частота користування	Близько третини користувачів
Тригери	Користувач натискає меню поряд із строкою програвання
Передумови	Користувач що-небудь прослуховує
Постумови	Змінюється якість передаваного звуку
Основний розвиток	Користувач натискає меню поряд із строкою програвання, вибирає бажану якість із списку
Альтернативні розвитки	-
Винятки	Інтернет користувача занадто повільний для обраної якості, виводиться попереджувальне повідомлення

Таблиця 4.9 - Сценарій використання

Назва	Підстройка аудіо еквалайзером
ID	9
Опис	Користувач налаштовує АЧХ звуку за допомогою еквалайзера
Актори	Преміумний користувач
Вигоди компанії	Висока якість звучання необхідна для залучення деяких клієнтів
Частота користування	Близько третини користувачів
Тригери	Користувач натискає меню поряд із строкою програвання
Передумови	Користувач знаходиться на сторінці із треками, обирає якийсь із них
Постумови	Змінюються АЧХ передаваного звуку
Основний розвиток	Користувач натискає меню поряд із строкою програвання, відкривається меню із слайдерами налаштування, користувач може їх змінювати
Альтернативні розвитки	-
Винятки	

Таблиця 4.10 - Сценарій використання

Назва	Кешування аудіо на пристрої користувача
ID	10
Опис	Користувач кешує обрані треки
Актори	Преміумний користувач
Вигоди компанії	Можливість прослуховування офлайн повинна мотивувати користувачів сплачувати підписку
Частота користування	Третина користувачів
Тригери	Користувач натискає на відповідну кнопку
Передумови	Користувач знаходиться на сторінці із треками, обирає якийсь із них
Постумови	Трек кешується на пристрої користувача
Основний розвиток	Користувач натискає на відповідну кнопку біля треку для додавання до кешу
Альтернативні розвитку	-
Винятки	Не вистачає вільного місця на пристрої, виводиться повідомлення про помилку

Таблиця 4.11 - Сценарій використання

Назва	Розмістити банер
ID	11
Опис	Рекламодавець може придбати місця для банеру, щоб він інколи відображався клієнтам
Актори	Рекламодавець
Вигоди компанії	Таким чином можна монетизувати користувачів без платної підписки
Частота користування	Часто
Тригери	Рекламодавець із особистого кабінету вибирає банер, який хоче розмістити і сплачує розміщення
Передумови	Рекламодавець знаходиться у власному кабінеті
Постумови	Банер буде випадково показано користувачам
Основний розвиток	Рекламодавець із особистого кабінету вибирає банер, який хоче розмістити і сплачує розміщення
Альтернативні розвитки	-
Винятки	Відбувається помилка під час сплати, показується повідомлення про помилку

Таблиця 4.12 - Сценарій використання

Назва	Розмістити аудіорекламу
ID	12
Опис	Рекламодавець може придбати місця для аудіореклами, щоб вони інколи програвалися клієнтам
Актори	Рекламодавець
Вигоди компанії	Таким чином можна монетизувати користувачів без платної підписки
Частота користування	Часто
Тригери	Рекламодавець із особистого кабінету вибирає аудіо, який хоче розмістити і сплачує розміщення
Передумови	Рекламодавець знаходиться у власному кабінеті
Постумови	Аудіорекламу буде випадково програно користувачам
Основний розвиток	Рекламодавець із особистого кабінету вибирає аудіо, яке хоче розмістити і сплачує розміщення
Альтернативні розвитки	-
Винятки	Відбувається помилка під час сплати, показується повідомлення про помилку. Аудіо занадто довге

Таблиця 4.13 - Сценарій використання

Назва	Видалити аудіорекламу
ID	13
Опис	Рекламодавець може видалити аудіорекламу
Актори	Рекламодавець
Вигоди компанії	Необхідний функціонал рекламодавця
Частота користування	Часто
Тригери	Рекламодавець із особистого кабінету вибирає аудіо, який хоче видалити
Передумови	Рекламодавець знаходиться у власному кабінеті
Постумови	Аудіорекламу буде видалено
Основний розвиток	Рекламодавець із особистого кабінету вибирає аудіо, який хоче видалити, натискає відповідну кнопку
Альтернативні розвитки	-
Винятки	-

Таблиця 4.14 - Сценарій використання

Назва	Видалити банер
ID	14
Опис	Рекламодавець може видалити банер
Актори	Рекламодавець
Вигоди компанії	Необхідний функціонал рекламодавця
Частота користування	Часто
Тригери	Рекламодавець із особистого кабінету вибирає банер, який хоче видалити
Передумови	Рекламодавець знаходиться у власному кабінеті
Постумови	Банер буде видалено
Основний розвиток	Рекламодавець із особистого кабінету вибирає банер, який хоче видалити, натискає відповідну кнопку
Альтернативні розвитки	-
Винятки	-

Таблиця 4.15 - Сценарій використання

Назва	Редагування інформації про виконавця, альбом тощо
ID	15
Опис	Контент менеджер редагує інформацію про виконавця, альбом тощо
Актори	Контент менеджер
Вигоди компанії	Можливість виправляти помилки
Частота користування	Часто
Тригери	Контент менеджер вибирає, що він бажає редагувати
Передумови	Контент менеджер знаходиться на сторінці із треками, виконавцями або альбомами
Постумови	Зміна інформації
Основний розвиток	Контент менеджер знаходиться на сторінці із треками, виконавцями або альбомами, вибирає, що він бажає редагувати, редагує та зберігає зміни
Альтернативні розвитки	-
Винятки	-



Таблиця 4.16 - Сценарій використання

Назва	Додавання виконавця, альбому, аудіотреку
ID	16
Опис	Контент менеджер додає інформацію про виконавця, альбом тощо
Актори	Контент менеджер
Вигоди компанії	Можливість додавати нове
Частота користування	Часто
Тригери	Контент менеджер вибирає, що він бажає додати
Передумови	Контент менеджер знаходиться на сторінці власного кабінету
Постумови	Зміна інформації
Основний розвиток	Контент менеджер знаходиться на сторінці власного кабінету, контент менеджер вибирає, що він бажає додати, додає
Альтернативні розвитки	-
Винятки	-

Таблиця 4.17 - Сценарій використання

Назва	Видалення записів за вимогами правовласників
ID	17
Опис	Контент менеджер видаляє треки
Актори	Контент менеджер
Вигоди компанії	Можливість видаляти старе або незаконне
Частота користування	Часто
Тригери	Контент менеджер вибирає, що він бажає видалити
Передумови	Контент менеджер знаходиться на сторінці із треками, виконавцями або альбомами
Постумови	Зміна інформації
Основний розвиток	Контент менеджер знаходиться на сторінці із треками, виконавцями або альбомами, вибирає, що він бажає видалити, видаляє та зберігає зміни
Альтернативні розвитки	-
Винятки	-

#### 4.2 Розробка структурної схеми

Розроблену структурну схему можна побачити в додатку Д. Система складається з таких модулів:

- інтерфейса користувача;
- відображення музики;
- пошуку контенту;
- БД;
- пошуку по найближчим сусідам.

Модуль інтерфейса користувача слугує для того, щоб користувач міг зручно взаємодіяти із системою, цей модуль безпосередньо чи опосередковано спілкується із іншими модулями.

Модуль відображення музики визначає, який музичний контент було відображено користувачу в модулі інтерфейса користувача, чи буде музика програватись, та як.

Модуль пошуку контенту використовується для пошуку музичного контенту за текстовим описом чи аудіо запитом. У другому випадку цей модуль взаємодіє із модулем пошуку найближчих сусідів, який в свою чергу спілкується із базою даних.

Модуль бази даних зберігає інформацію про музичні треки, виконавців, користувачів та їх плейлисти, тощо.

Модуль інтерфейсу користувача спілкується з іншими модулями засобами протоколів TCP/IP. Модуль пошуку контенту, отримуючи запит, після обробки передає його модулю пошуку по найближчим сусідам. Цей модуль в свою чергу спілкується з БД.

#### 4.3 Розробка функціональної схеми

Розроблену функціональну схему можна побачити в додатку Е. Система складається із таких основних модулів:

- модуль інтерфейсу користувача;
- модуль пошуку контенту;

- модуль пошуку по найближчим сусідам;
- модуль бази даних;

Модуль інтерфейсу користувача слугує інтерфейсом, з яким взаємодіє користувач, цей модуль складається із окремих підсистем:

- система пошуку та рекомендацій;
- налаштування;
- відображення/програвання контенту, музики та реклами;
- управління плейлистами

Система пошуку та рекомендацій взаємодіє із модулем для пошуку контенту. Цей модуль складається із двох підсистем:

- система отримання векторних представлень;
- система текстового пошуку

В залежності від типу отриманих даних використовуються різні підсистеми. Система отримання векторних представлень обробляє звукові сигнали та використовує нейромережі та алгоритми, які будуть розглянуті докладніше далі у роботі. Система текстового пошуку робить прості запити до БД і шукає серед текстових даних про назви, імена тощо. Система отримання векторних представлень працює взаємодіє із базою даних через модуль пошуку по найближчим сусідам, про цей модуль буде мова далі у роботі. Це пов'язано із особливістю зберігання векторних даних.

Підсистема налаштувань дозволяє користувачу налаштовувати різні налаштування відображення, облікового запису та програвання музики. Підсистема програвання музики та реклами відповідає за розташування даних на сторінці сайту чи додатку. Ця підсистема взаємодіє із базою даних, яка повертає інформацію та контент на відповідні запити. Система управління плейлистами взаємодіє із системами пошуку та рекомендацій, і формує автоматичні списки програвання, та із системою відображення, для правильного розташування даних у вікні.

#### 4.4 Вибір та обґрунтування елементів та технологій

Мовою програмування для даної роботи було обрано мову програмування загального призначення Python. Це високорівнева скриптова мова програмування із компактним синтаксисом та динамічною типізацією. В даній роботі використовувалася саме ця мова програмування через такі переваги:

- висока швидкість розробки додатків;
- велика екосистема бібліотек для наукових обчислень;
- кросплатформеність;
- велика кількість фреймворків для веб-розробки

Для розробки неймереж в цій роботі використовувався фреймворк Tensorflow, розроблений компанією Google для ефективного обчислення матричних операцій та диференціювання за допомогою побудови графів обчислень. Спільними для всіх фреймворків машинного навчання є можливість виконання розрахунків на GPU, що значно прискорює матричні операції. Відмінними і ключовими властивостями Tensorflow у виборі є такі:

- наявність компоненту Tensorflow Serving, що створює сервер, який ефективно обробляє запити до моделі;
- наявність компоненту Tensorflow Lite, що дозволяє використовувати моделі на мобільних пристроях

В даній роботі використовувалася ORM peewee для мови програмування Python, особливістю якої є простота використання, що пришвидшує розробку, потужні інструменти та підтримку трьох популярних баз даних - MySQL, PostgreSQL, SQLite, між якими є можливість динамічно переключатися. Для прототипування було використано SQLite, оскільки це доволі проста база даних, яка зберігає всі дані у єдиному файлі, підтримується Python по замовчуванню і не потребує якогось встановлення. Для забезпечення роботи повноцінного сервісу краще використовувати PostgreSQL або MySQL.

## 4.5 Розробка бази даних

Оскільки для оцінки схожості в даній роботі використовуються векторні представлення, для яких використання реляційної бази даних є недоцільним, в даній роботі буде використано два різні типи баз даних. Перший, це реляційна БД. Хоча такі БД не можуть ефективно зберігати та шукати векторні представлення, вони ідеально підходять для зберігання табличних даних із залежностями. Оскільки дані із музичної галузі про виконавця, альбоми і тому подібне легко представляється реляційною БД, то для цього вона і буде використана. Другий тип буде використано для зберігання векторних представлень. Цей тип баз даних використовує алгоритми приблизного пошуку найближчого сусіда. Такі алгоритми дозволяють швидко шукати вектори з мінімальної відстаню за якоюсь метрикою (евклідова, косинусна відстань тощо) до вектору-запиту. Недоліком таких БД є те, що зазвичай вони зберігаються у вигляді єдиного файлу-індексу, у який записується лише вектор та його ідентифікатор і нічого більше.

Схему сутність-зв'язок у нотації воронячої лапки можна побачити у додатку Ж. Усі сутності мають атрибут ID в якості унікального ідентифікатора. БД має такі сутності:

- виконавець;
- альбом;
- музичний трек;
- користувач;
- плейлист;
- індекс найближчих сусідів;

Виконавець поєднан із треком зв'язком один до багатьох, оскільки один виконавець може мати багато пісень. Ситуація, де виконавець фактично представлений декількома виконавцями або гуртами розв'язуються створенням нового виконавця, адже зазвичай це короточасні утворення для створення одного

альбому чи синглу. Таким саме зв'язком виконавець поєднан із альбомом. Користувач Поєднан із плейлистом через відношення один до багатьох, бо кожен користувач може створювати багато плейлистів, або не створювати взагалі. Плейлист у свою чергу зв'язан із музичним треком відношенням один до багатьох. Кожен музичний трек пов'язаний із індексом найближчих сусідів відношенням один до одного. Індекс найближчих сусідів - це таблиця відповідностей унікального ключа кожного окремого треку в реляційній БД, до унікального ключа векторного представлення цього ж треку в індексі найближчих сусідів.

Пошук найближчих сусідів є фундаментальною та необхідною операцією у багатьох завданнях, найяскравішим прикладом є бази даних та машинне навчання. Метод найближчих сусідів полягає у знаходженні об'єктів, які описані векторами фіксованої розмірності, найближчих до цільового згідно з якоюсь функцією відстані. Точне знаходження таке об'єктів потребує повного перебору БД, що при наявності великої кількості записів є недоцільним. Для ефективного пошуку розроблені наближені методи найближчих сусідів. Умовно алгоритми для наближеного пошуку сусідів можна поділити на 3 категорії:

- локально чутливе хешування (locality-sensitive hashing - LSH);
- просторово-розподілені алгоритми;
- засновані на сусідстві

В даній роботі використовувався алгоритм annou на базі LSH, поєднаного із просторовими розподіленими деревами. Ідея даного методу полягає у відображенні даних великої розмірності у простір меншої розмірності, використовуючи підібрані випадкові проектуючі функції. LSH функція  $H$  для функції відстані  $f$  визначається як  $(r_1, r_2, p_1, p_2)$ -sensitive, де для будь-яких 2 точок  $x, y$  існує два пороги для відстані  $r_1, r_2$ , і два вірогіднісні пороги  $p_1, p_2$ , які задовольняють умови, що відображення двох точок мають одні й ті самі координати з ймовірністю не менше  $p_1$ , якщо відстань між ними менше  $r_1$ , і що

відображення двох точок мають одні й ті самі координати з ймовірністю не більшою, ніж  $r_2$ , якщо відстань між ними більша, ніж  $r_2$ .

Ця функція визначає, що вірогідність співставлення  $x$ ,  $y$  в одну категорію росте із зменшенням відстані між ними. Значення точки даних співставляється із хешем(або, відро) на основі випадкового проектування та методу дискретизації. Для забезпечення гарантій якості виконання використовується певна кількість хеш-функцій, або різні схеми, такі як схема AND-OR, схема OR-AND, інвертована схема на основі списків, або схема дерев.

Дерево просторовий розділ широко використовується для вирішення проблеми точного та приблизного найближчого сусіду. Як правило, цей простір розподіляється ієрархічно, і існує дві основні схеми поділу: поворотні та компактні схеми розподілу. Механізми обертання розбивають простір, спираючись на відстанях від точки до точки обертання, тоді як компактні методи розподілу або розділяють дані точки на кластери, або випадково.

## 4.2 Розгортання системи

Діаграму розгортання системи можна побачити в додатку К. Перший вузол цієї діаграми - пристрій користувача, який може бути як персональним комп'ютером, так і мобільним пристроєм. На його пристрої розгортається сайт мобільний додаток чи веб-сайт, який спілкується із веб-сервером застосунку. Веб-сервер відповідає за видачу контенту та запити до серверів моделі рекомендацій та БД. Спілкування з сервером рекомендаційної моделі відбувається засобами gRPC - система віддаленого виводу процедур розроблена в Google. На сервері рекомендаційної моделі розгорнуто інстанцію Tensorflow Serving - сервер, що використовуючи збережені обчислювальні графи Tensorflow керує взовами до них і повертає результат обчислення моделі. Крім цього веб-сервер застосунку спілкується із сервером БД, на якому розгорнуті PostgreSQL сервер та індекс найближчих сусідів Annoy. Індекс найближчих сусідів залежить від результатів обробки запиту сервером рекомендаційної моделі.



## 5 ОЦІНКА МОДЕЛІ ТА ФІНАЛЬНИЙ ВАРІАНТ АЛГОРИТМУ

Для оцінки моделі було опитано 9 людей, для кожного було сгенеровано 10 трійок музичних записів. Перший запис - композиція, до якої опитувані повинні були визначити більш схожу за звучанням. Дві інших - випадковим чином перемішані випадково обрана композиція та композиція обрана алгоритмом класифікації. Метою було визначити, чи алгоритм здатен перевершити хоча б випадковий вибір. Фактично усього було прослухано 90 трійок пісень, де в 73 з яких опитувані віддали перевагу алгоритму, а у 8 випадках не змогли точно визначитись. Це значить, що у 81% випадків люди віддавали перевагу алгоритму. Крім того до алгоритму було додано невелику модифікацію. Замість використання лише алгоритму, заснованого на класифікації даних, було використано поєднання його із методом предиктивного контрастивного кодування. Діаграму діяльності цього алгоритму можна побачити в додатку Л. Ідеєю алгоритму є те, що найбільш суб'єктивно схожі результати зустрічаються доволі близько у просторі векторних представлень, проте віддаленіше від неправильних, тому використання одразу двох моделей повинно допомогти віднайти такі випадки. Алгоритм працює наступним чином: спочатку переданий файл оброблюється алгоритмом для класифікації жанрів і робиться вибірка із 100 (евристично визначене число) найближчих сусідів, після чого файл обробляється алгоритмом контрастивного предиктивного кодування, і робиться вибірка ще із 100 наборів, якщо в цих вибірках є перетин, він підіймається вгору у списку видачі, інакше видається результат першого алгоритму. Для оцінки цього алгоритму було опитано трьох людей на десяти трійках, де було представлено кращий результат алгоритму для класифікації, та алгоритму, описаного вище. Змішаному алгоритму було надано перевагу у 18 із 30 випадків, що відповідає 60%, проте для такої малої вибірки даний результат не є статистично значущим.

## 6 РОЗРОБКА СТАРТАП-ПРОЕКТУ

### 6.1 Опис ідеї проекту

Ідея проекту даної роботи - розробити сервіс, який дозволяє користувачам прослуховувати музику, одночасно з цим, надавати гнучкі та доцільні рекомендації на основі історії їх прослуховувань, створення персоналізованих плейлистів. Користувач повинен мати можливість автоматично створити плейлист, який буде складатися на основі запропонованих ним композицій. Узагальнення цих ідей можна побачити в таблиці 6.1.

Таблиця 6.1 Опис ідеї стартап проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	1. Автоматичне створення списків програвання на основі певних композицій	Швидкий доступ, до музики, що хочеться прослуховувати саме зараз
	2. Пошук музики за записаними фрагментами, або із особистої колекції користувача	Легкий пошук нової музики, можливо маловідомої / маловідомі групи мають більше шансів стати відомими

Оскільки на ринку музики існує багато сервісів для простого стримінгу музики, доведеться конкурувати і з ними, а не тільки з тими, що пропонують рекомендації. Тому із найкрупніших конкурентів можна виокремити Spotify, Google Music та Apple Music. Порівняння даного проекту із ними можна побачити в таблиці 6.2

Таблиця 6.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Техніко-економічні характеристики ідеї	товари/концепції конкурентів				Слабка сторона	Нейтральна сторона	Сильна сторона
	Мій проект	Spotify	Apple Music	Google Music			
Рекомендаційна система	Заснована на музичному контенті пошукова система	Система без урахування контенту, а лише користувацьких вподобань	відсутня	відсутня			+
Автоматичні плейлисти	Заснована на музичному контенті пошукова система	Система без урахування контенту, а лише користувацьких вподобань	Случайне перемішування бібліотеки				+
Віднаходження нового	Заснована на музичному контенті пошукова система	Система без урахування контенту, а лише користувацьких вподобань	Базові рекомендації	Базові рекомендації			+
Розмір бібліотеки	Заснована на музичному	Система без урахування контенту, а			+		

	контенті пошукова система	лише користувачьких вподобань					
Розмір клієнтської бази	Жодного	83 мільйони	30 мільйонів	22 мільйони	+		

## 6.2 Технологічний аудит ідеї проекту

Оскільки проект є більше програмною проблемою, то більшість технологій для його створення є доступною для розробки. Хоча головну ідею проекту можна реалізувати і без додатків, фінальний продукт має мати мобільний додаток, сайт, десктопний додаток або все одразу, тому ці пункти і вказані в таблиці. Технології, що доцільно використовувати можна побачити на таблиці 6.3

Таблиця 6.3 Технологічна здійсненність ідеї проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Рекомендаційна система	мови загального програмування, фреймворки машинного навчання	У відкритому доступі є, наприклад, такі: Python, C++/Tensorflow, Pytorch	Свободні для використання
Автомашичні плейлисти	мови загального програмування, фреймворки машинного навчання	У відкритому доступі є, наприклад, такі: Python, C++/Tensorflow, Pytorch	Свободні для використання
Мобільний додаток	мови та середовища розробки мобільних додатків	Android Studio, Java, Xcode, Swift	Свободні для використання
Веб-сайт	мови розмітки та спеціальні мови програмування	HTML, Java Script, Angular, React	Свободні для використання

Десктоп ний додаток	мови та середовища розробки	C++, Qt	Свободні для використання
Обрані технології реалізації: Python, Tensorflow, React Native, C++			

### 6.3 Аналіз ринкових можливостей стартап проекту

Таблиця 6.4 Попередня характеристика потенційного ринку стартап-проекту

п/п	Показники стану ринку (найменування)	Характеристика
	Кількість головних гравців, од	5
	Загальний обсяг продаж, грн/ум.од	1 блн. ум. од.
	Динаміка ринку (якісна оцінка)	Зростає
	Наявність обмежень для входу (вказати характер обмежень)	Недискримінаційні якісні
	Специфічні вимоги до стандартизації та сертифікації	Угоди із правовласниками музичного контенту
	Середня норма рентабельності в галузі (або по ринку), %	30%

Таблиця 6.5 Характеристика потенційних клієнтів стартап-проекту

<i>Потреба, що формує ринок</i>	<i>Цільова аудиторія (цільові сегменти ринку)</i>	<i>Відмінності у поведінці різних потенційних цільових груп клієнтів</i>	<i>Вимоги споживачів до товару</i>
Потреба до прослуховування музики	1. Аудіофіли, 2. Підлітки 3. Працівники із сидячою роботою	Потребують різну якість прослуховування, зацікавленість у різних записах	Легкість доступу до музики, гнучкі налаштування
Потреба для пошуку нової музики	4. Аудіофіли, 5. Підлітки 6. Працівники із сидячою роботою	Можливі відмінності у вподобаннях до різних типів пошуку	Зручна, інтуїтивна та розумна система

Таблиця 6.6 Фактори загроз

Фактор	Зміст загрози	Можлива реакція компанії
Отримання несанкціонованого доступу сторонніми особами	Хакерська атака що може призвести до компрометації даних клієнтів	Залучення спеціалістів з інформаційної безпеки Використання засобів шифрування та резервного копіювання
Висока конкуренція	Відсутність можливості залучити нових клієнтів	Ретельний розгляд проблем потенційних клієнтів Спробо переорієнтувати рішення Консультації із спеціалістами
Недостача капіталовкладень	Витрачені усі кошти до моменту виходу на ринок	Пошук нових джерел інвестицій

Таблиця 6.7 Фактори можливостей

Фактор	Зміст можливості	Можлива реакція компанії
Отримання інвестицій	Отримання капіталу що необхідний для реалізації продукту	Розробка продукту
Успішна маркетингова політика	В результаті проведеної маркетингової політики отримана висока зацікавленість користувачів	Підтримка стабільної роботи системи та проведення масштабування системи Збільшення цін на використання сервісу Використання подібної маркетингової стратегії надалі для залучення нових користувачів
Поглинання конкурентами	Пропозиція купівлі проекту або розроблених технологій одним із конкурентів	Розвиток розроблених технологій Оцінка вартості розроблених технологій



Таблиця 6.8 Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Олігополія	Велика кількість конкурентів Велика ринкова сила Схожість використовуваних технологій	Інформування ринку щодо появи нової платформи управління хмарною інфраструктурою
Глобальний	Присутні не залучені ринки, на залучених висока лояльність до конкурентів	Інформування ринку щодо якості використовуваної новаторської технології, вихід на незаховлені ринки
Внутрішньогалузева	Діяльність в одній галузі економіки Надання сервісів одного типу	Зменшення вартості сервісу, покращення його якості
Товарно-видова	Надання різних сервісів одного типу	Маркетингова політика
Нецінова	Використання наданих послуг для залучення аудиторії	Покращення якості сервісу
Марочна	Спільна цільова аудиторія	Інформування ринку щодо появи нового стрімінгового сервісу

Таблиця 6.9 Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Spotify, Apple Music Google Music	Розмір капіталовкладень, Забезпечення гнучких цін, Доступ до каналів розподілу, Витрати на масштабах Розмір бібліотеки	Праводаці	Змінні витрати: Виробничі непрямі дегресивні -Системи інформації: пропаганда, реклама та директ-маркетинг, -Рівень чутливості до цін: споживачі орієнтовані на цінність продукту -Продуктова диференціація: якість, Методи контролю якості: тестування та профілювання,	Копіювання функціоналу, Монополізація дистриб'юторів, Демпінгування

Висновки	Інтенсивна конкуренція	Можливість входу на ринок забезпечить мінімізація цін,	Формування ціни залежить від вартості купівлі ліцензії	Клієнти диктують якість забезпечення сервісу, його доступність	Забезпечення захисту інтелектуальної власності, гнучкість цінової політики
----------	------------------------	--	--	--	--

Таблиця 6.10 Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
Унікальність сервісу	Розроблений продукт має унікальні характеристик
Цінова політика	Отримання прибутку здійснюється за рахунок гнучкої моделі оплати

Таблиця 6.11 Порівняльний аналіз сильних та слабких сторін “Simsong”

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з “Simsong”						
		3	2	1		1	2	3
Унікальність сервісу	15							

Цінова політика	10							
-----------------	----	--	--	--	--	--	--	--

Таблиця 6.12 SWOT- аналіз стартап-проекту

Сильні сторони: якість пошуку, зручність користування	Слабкі сторони: відсутність бази ліцензій, відсутність капіталовкладень
Можливості: Нові ринки, Нові постачальники (незалежні виконавці)	Загрози: Відсутність попиту

Таблиця 6.13 Альтернативи ринкового впровадження стартап-проекту

п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
.	Розробка системи мультиmodalного пошуку	Ймовірне	24 місяці
.	Маркетингова кампанія для приваблювання користувачів	Малоймовірне	6 місяці
.	Пропонування безкоштовних тарифів	Малоймовірне	2 місяці
Обрана альтернатива: Розробка системи мультиmodalного пошуку			

## 6.4 Розроблення ринкової стратегії проекту

Таблиця 6.14 Вибір цільових груп потенційних споживачів

п/п	Опис профілю цільової групи потенційних клієнтів	Гото вність споживачів сприйняти продукт	Орієн товний попит межах цільової групи (сегменту)	Інтенс ивність конкуренції в сегменті	Пр остота входу у сегмент
.	Аудіофі ли	Висо ка	50%	Висок а	Ви сокі бар'єри входу
.	Підлітк и	Сере дня	30%	Висок а	Ви сокі бар'єри входу
.	Працівн ики із сидячою роботою	Низь ка	15%	Висок а	Ви сокі бар'єри входу
Які цільові групи обрано: аудіофіли, підлітки					

Таблиця 6.15 Визначення базової стратегії розвитку

п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
	Надання послуг користувачам	Вихід на незайняті ринки	Здатність протистояти прямим конкурентам	Стратегія диференціації

Таблиця 6.16 Визначення базової стратегії конкурентної поведінки

п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
	Ні	Забирати та залучати нових	Ні	Стратегія лідера. Розширення первинного попиту

Таблиця 6.17 Визначення стратегії позиціонування

п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
	Простий інтерфейс, велика бібліотека, функціональність	Стратегія диференціації	Формування регулярного попиту Збільшення разового використання послуги Виявлення нових груп споживачів Нові напрями застосування існуючої послуги	Інноваційність технології Проста та використання



## 6.5 Розроблення маркетингової програми стартап проекту

Таблиця 6.18 Визначення ключових переваг концепції потенційного товару

п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
	Потреба до прослуховування музики	Користувач може зручно прослуховувати улюблену музику	Зручність користування, якість обслуговування
	Потреба для пошуку нової музики	Мотивація постійно шукати нове, чим простіше це робити, ти частіше цим будешь користуватися	Якість надання послуг, інноваційність методів, простота

Таблиця 6.19 Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Програмний продукт що надає можливість прослуховувати та шукати нову музика		
	Властивості/характеристики	М/Нм	Вр/Тх /Гл/Е/Ор
	Кількість		1 шт.
	Якість: стандарти якості постачання програмних продуктів		
	Пакування: через веб інтерфейс		
	Марка: Sinsong		
	Програмний продукт		
	Програмний продукт, технічна підтримка та підписка на оновлення		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності			

Таблиця 6.20 Визначення меж встановлення ціни

п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	~300 грн/місяць	~300 грн/місяць	20 000 грн./міс.	100 - 400 грн./місяць

Таблиця 6.21 Формування системи збуту

п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глиби на каналу збуту	Оптиміальна система збуту
1	Оплата підписки іде через веб-сайт	Інформування користувачів Доступ до додаткових функцій сервісу	Канал одного рівня	Селективна з використанням комбінованого каналу збуту

Таблиця 6.22 Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Стримінг та рекомендації і пошук нової музики	Прямі офіційні	Доступність та об'єктивність інформації про фірму і товар Унікальність послуги	Формування у цільовій аудиторії обізнаності про появу нового продукту Інформування користувачів про властивості та переваги продукту Пояснення	Раціоналістична стратегія реклами

				цільовій аудиторії принципу роботи платформи	
--	--	--	--	--	--

## ВИСНОВКИ

Метою дисертації було створення інтелектуальної системи для пошуку схожого музичного контенту. Для розв'язання даної задачі було використано моделі на основі нейромереж. Одна із моделей спеціалізується на класифікації жанрів. Завдяки отриманим проміжним представленням можна вирішувати задачу пошуку схожого контенту. Друга неймережа навчена без учителя і, хоча не дає достатньої якості як окрема модель для пошуку схожого контенту, все одно є корисною, коли працює в ансамблі із першою моделлю.

Оцінку якості моделі було виконано шляхом візуалізації векторного представлення композицій та опитування. В ході опитування було отримано статистично значимі результати щодо уподобання рекомендацій, запропонованих алгоритмом.

## СПИСОК ВИКОРСТАНИХ ДЖЕРЕЛ

1. About Last.fm [Електронний ресурс]. Режим доступу: <https://www.last.fm/about>
2. Last.fm – Wikipedia [Електронний ресурс]. Режим доступу <https://en.wikipedia.org/wiki/Last.fm>
3. Spotify – [Електронний ресурс]. Режим доступу: Wikipedia <https://en.wikipedia.org/wiki/Spotify>
4. М. Тім Джонс. Рекомендательные системы: Часть 1. Введение в подходы и алгоритмы [Електронний ресурс]. Режим доступу: <https://www.ibm.com/developerworks/ru/library/os-recommender1/>
5. Clustering
6. Pandora Radio [Електронний ресурс]. Режим доступу: <https://www.pandora.com>
7. How Pandora Radio Works [Електронний ресурс]. Режим доступу: <http://computer.howstuffworks.com/internet/basics/pandora3.htm>
8. VV Show #54 - Tim Westergren of Pandora [Електронний ресурс]. Режим доступу: [http://www.venturevoice.com/2009/03/vv\\_show\\_54\\_tim\\_westergren\\_of\\_p.html](http://www.venturevoice.com/2009/03/vv_show_54_tim_westergren_of_p.html)
9. Mikolov et al. Distributed Representations of Words and Phrases and their Compositionality [Електронний ресурс]. Режим доступу: [http://web2.cs.columbia.edu/~blei/seminar/2016\\_discrete\\_data/readings/MikolovSutskeverChenCorradoDean2013.pdf](http://web2.cs.columbia.edu/~blei/seminar/2016_discrete_data/readings/MikolovSutskeverChenCorradoDean2013.pdf)
10. Vector Representations of Words [Електронний ресурс]. Режим доступу: <https://www.tensorflow.org/tutorials/word2vec>
11. How does Shazam work [Електронний ресурс]. Режим доступу: <http://coding-geek.com/how-shazam-works/>

12. Li-Chun Wang. An Industrial-Strength Audio Search Algorithm [Електронний ресурс]. Режим доступу: <https://www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf>
13. Kyunghyun Cho, Bart van Merriënboer, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation [Електронний ресурс]. Режим доступу: <https://arxiv.org/abs/1406.1078>
14. Yaroslav Yuriiiovych Dorogyi, Vasyl Vasylovych Tsurkan, Oleksandr S Kharilin Автоматичне розпізнавання музичних жанрів глибокими згортковими нейронними мережами [Електронний ресурс]. Режим доступу: <http://feltran.kpi.ua/article/view/105159/119678>
15. Aaron van den Oord, Yazhe Li, Oriol Vinyals Representation Learning with Contrastive Predictive Coding [Електронний ресурс]. Режим доступу: <https://arxiv.org/abs/1807.03748>

## ДОДАТОК А

## Стаття за темою дисертації

ISSN 2524-2725. ElectronAcoustEng, 2018, vol. 1, no. 1

45

УДК 004.89

## Автоматичне розпізнавання музичних жанрів глибокими згортковими нейронними мережами

Дорогий Я. Ю., к.т.н., доц., ORCID [0000-0003-3848-9852](https://orcid.org/0000-0003-3848-9852)e-mail [argusyk@gmail.com](mailto:argusyk@gmail.com)Цуркан В. В., к.т.н., ORCID [0000-0003-1352-042X](https://orcid.org/0000-0003-1352-042X)e-mail [v.v.tsurkan@gmail.com](mailto:v.v.tsurkan@gmail.com)Хапілін О. С., ORCID [0000-0002-3888-584X](https://orcid.org/0000-0002-3888-584X)e-mail [khapilins@yandex.ua](mailto:khapilins@yandex.ua)

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського" [kpi.ua](http://kpi.ua)

Київ, Україна

*Реферат*—В статті розглядаються алгоритми для автоматичного розпізнавання музичних жанрів та пропонується використання глибоких згорткових нейронних мереж для цієї задачі. Спираючись на реальні дані, окреслено архітектуру мережі та оцінено її якість. Робота виконана з використанням дата-сету GTZAN. Було розглянуто задачу класифікації для чотирьох та десяти жанрів з використанням мел-кепстральних коефіцієнтів та аудіо хвилі в якості ознак. Якість запропонованого алгоритму було протестовано на відкладених даних для чотирьох та десяти різних жанрів та порівняно з використанням обмеженої машини Больцмана для чотирьох жанрів.

Бібл. 11, рис. 3, табл. 2.

*Ключові слова* — глибокі нейронні мережі; згорткові мережі; пошук музичної інформації; класифікація.

### I. ВСТУП

Музичний жанр – характеристика музики, що полягає у певних композиційних та стилістичних ознаках, таких, як інструментальна та ритмічна структура, характеристика вокалу виконавця тощо. Зазвичай жанри використовуються для організації музичних композицій, наприклад, для розміщення різних альбомів в музичних магазинах чи для пропозиції в інтернет-радіо. Із швидким ростом кількості музики ручна класифікація композицій стає недоцільною та ресурсомною. Оскільки існуючих рішень мало, а їх якість ще не досягла достатнього для масового впровадження рівня, дана область досліджень є перспективною. Крім того, все більш широкого поширення набувають такі музичні сервіси, як Spotify, iTunes та Pandora, що з їх кількістю музичних даних повинні бути зацікавлені в їх автоматичній обробці. Крім того, натреновані на класифікації жанрів мережі можна спробувати використати як основу для інших завдань, наприклад, для пошуку схожих музичних композицій. В цій роботі розглянуто метод на основі глибоких згорткових мереж для автоматичної класифікації музичних жанрів.

### II. АНАЛІЗ ЛІТЕРАТУРНИХ ДАНИХ І ПОСТАНОВКА ПРОБЛЕМИ

Хоча завдання класифікації музичних жанрів

менш поширене, ніж, наприклад, зображень, кількість підходів досить значна. На практиці зазвичай використовують як методи навчання без вчителя [1], [2], так і з учителем [3]. В першому випадку дослідники використовували глибокі згорткові обмежені машини Больцмана [4] для отримання похідних від спектру ознак.

У другому джерелі використовувалися кепстральні мел-коефіцієнти та обмежена машина Больцмана для отримання ознак, що потім відправлялися до нейронної мережі прямого поширення.

Структура машин Больцмана складається з видимого та прихованого шару. В разі глибоких машин Больцмана прихованих шарів декілька, і кожен прихований шар є видимим для наступного шару. Така структура дозволяє створювати складні ієрархічні ознаки без учителя, які можна буде використовувати в подальшому для класифікації. Конфігурація такої системи описується її енергією:

$$E(v, h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{i,j},$$

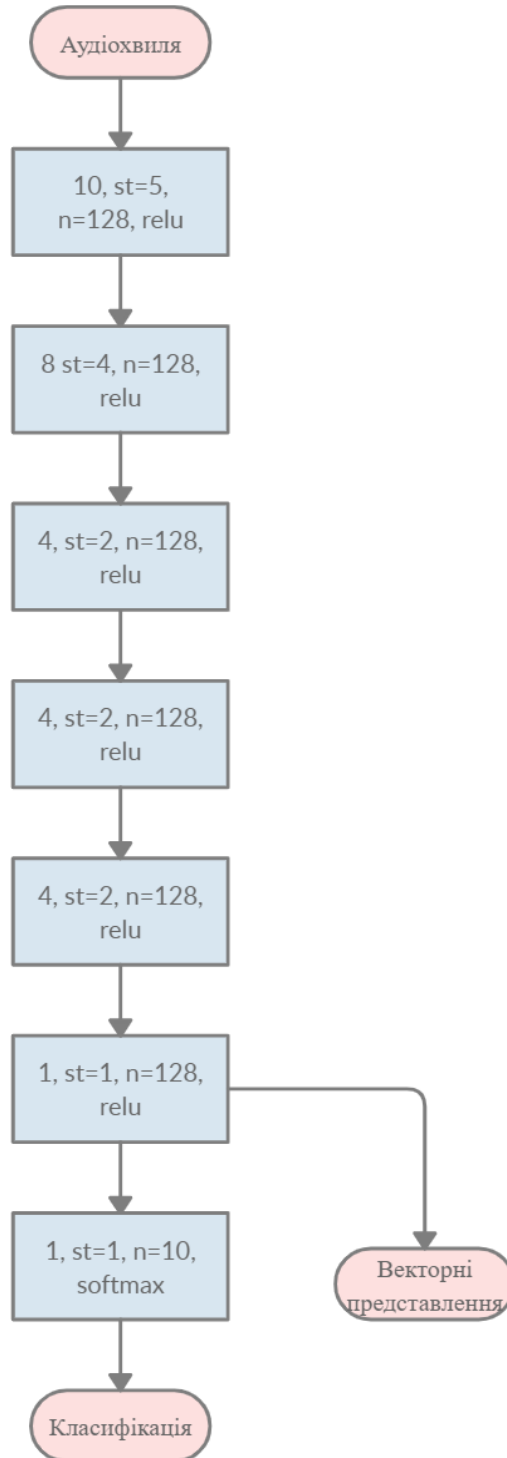
де  $v, h$  – вектори видимого та прихованого шарів,  $w$  – вагові коефіцієнти між видимим та прихованим шаром,  $a, b$  – коефіцієнти зсуву для видимого та прихованого шарів.





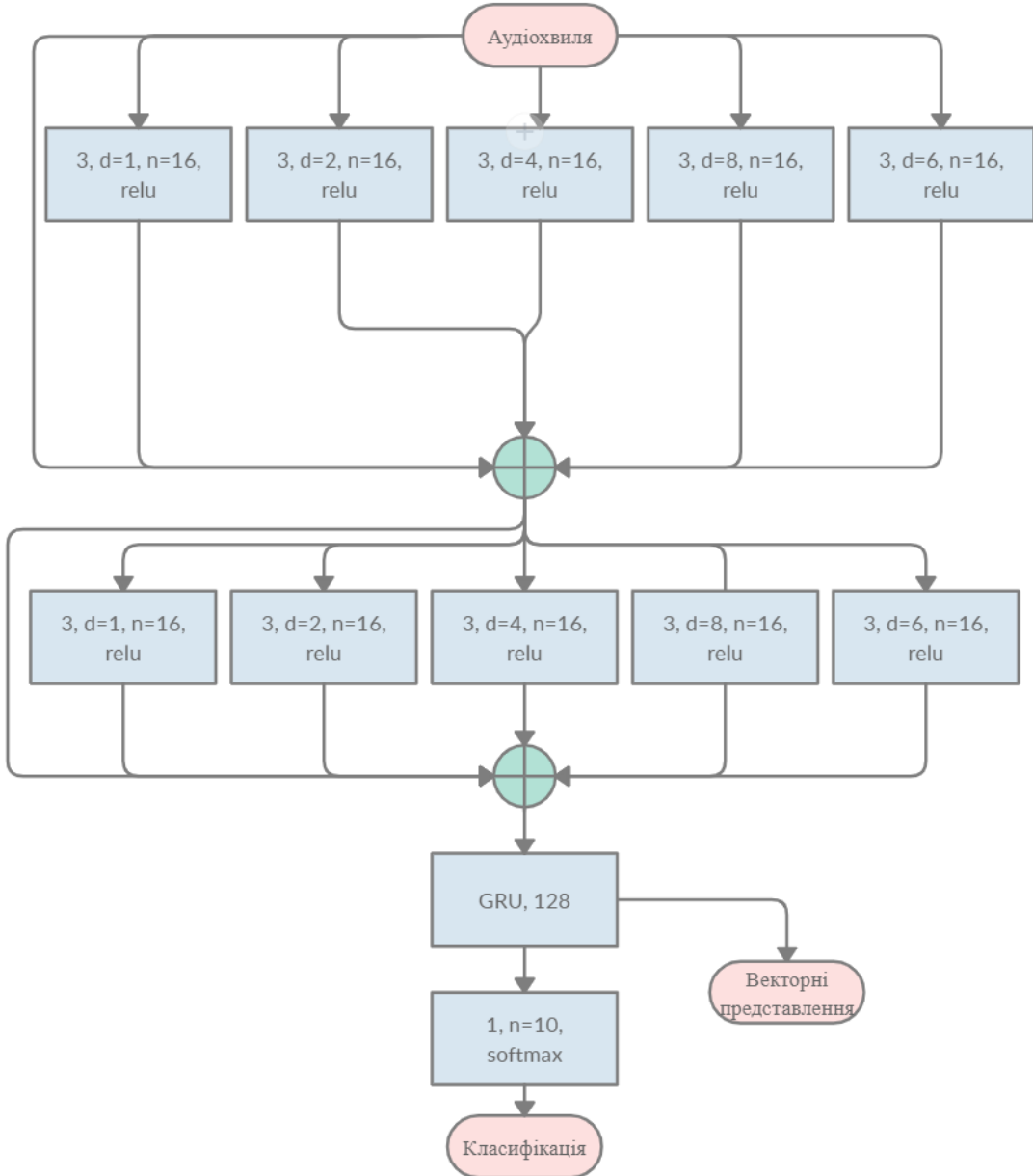
## ДОДАТОК Б

## Структурна схема простої згорткової мережі



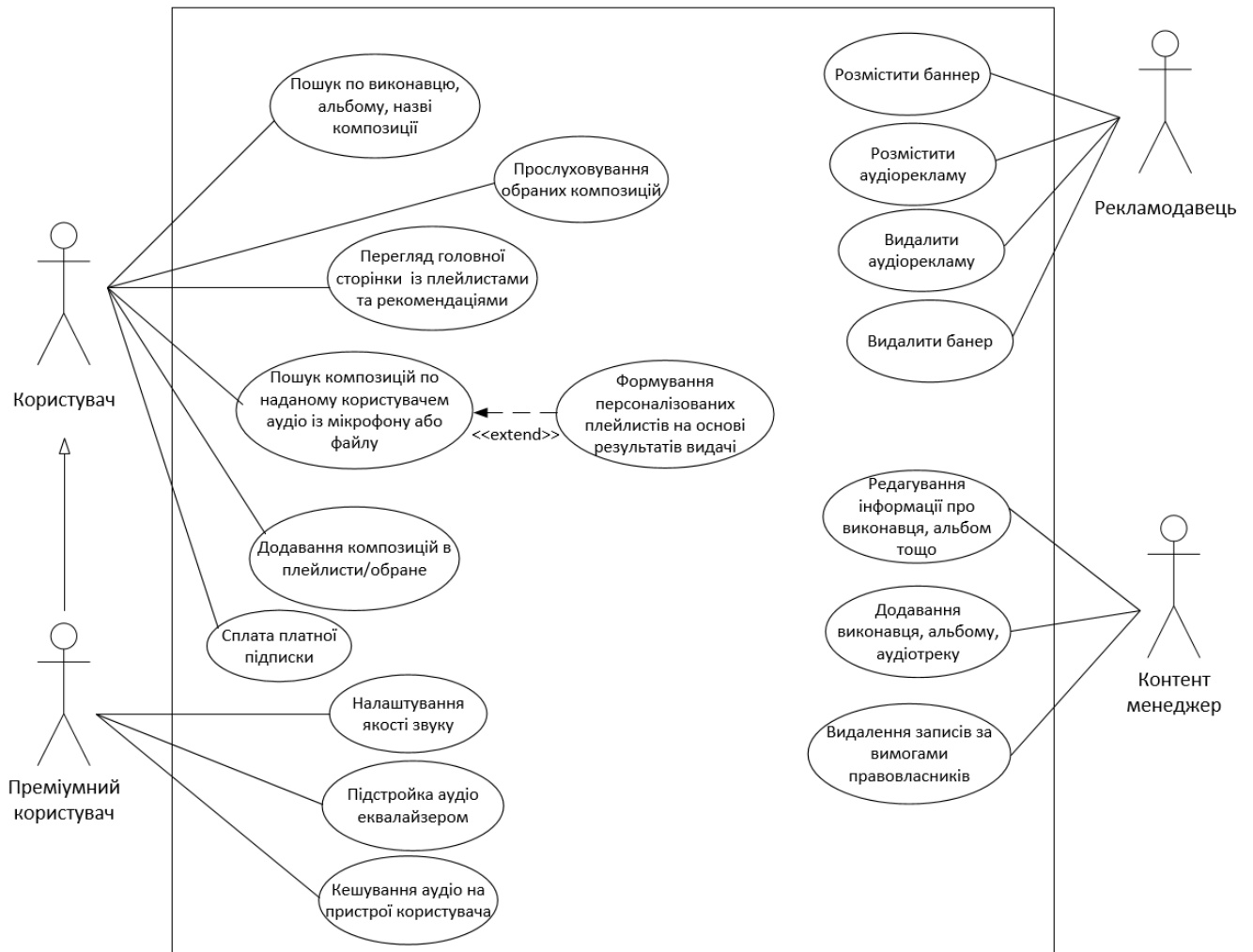
## ДОДАТОК В

## Структурна схема мережі із блоками та рекурентністю



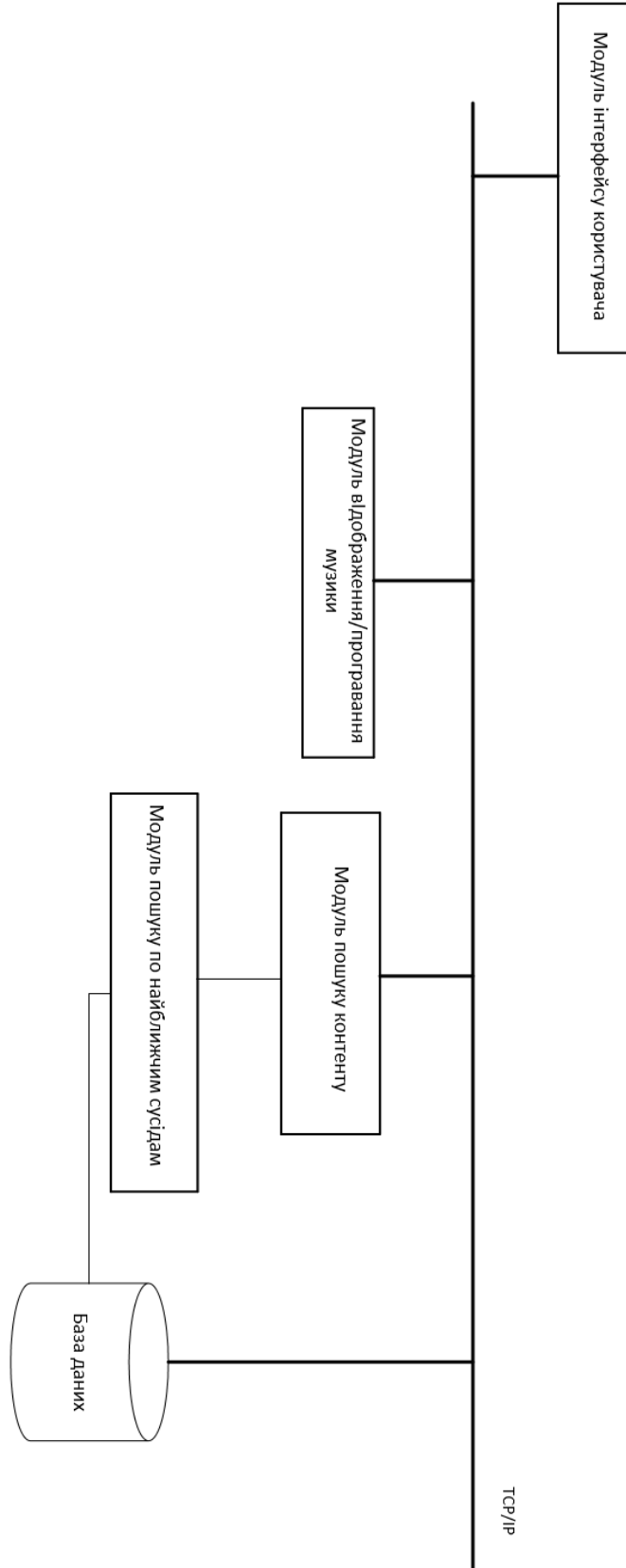
## ДОДАТОК Г

## Діаграма сценаріїв використання



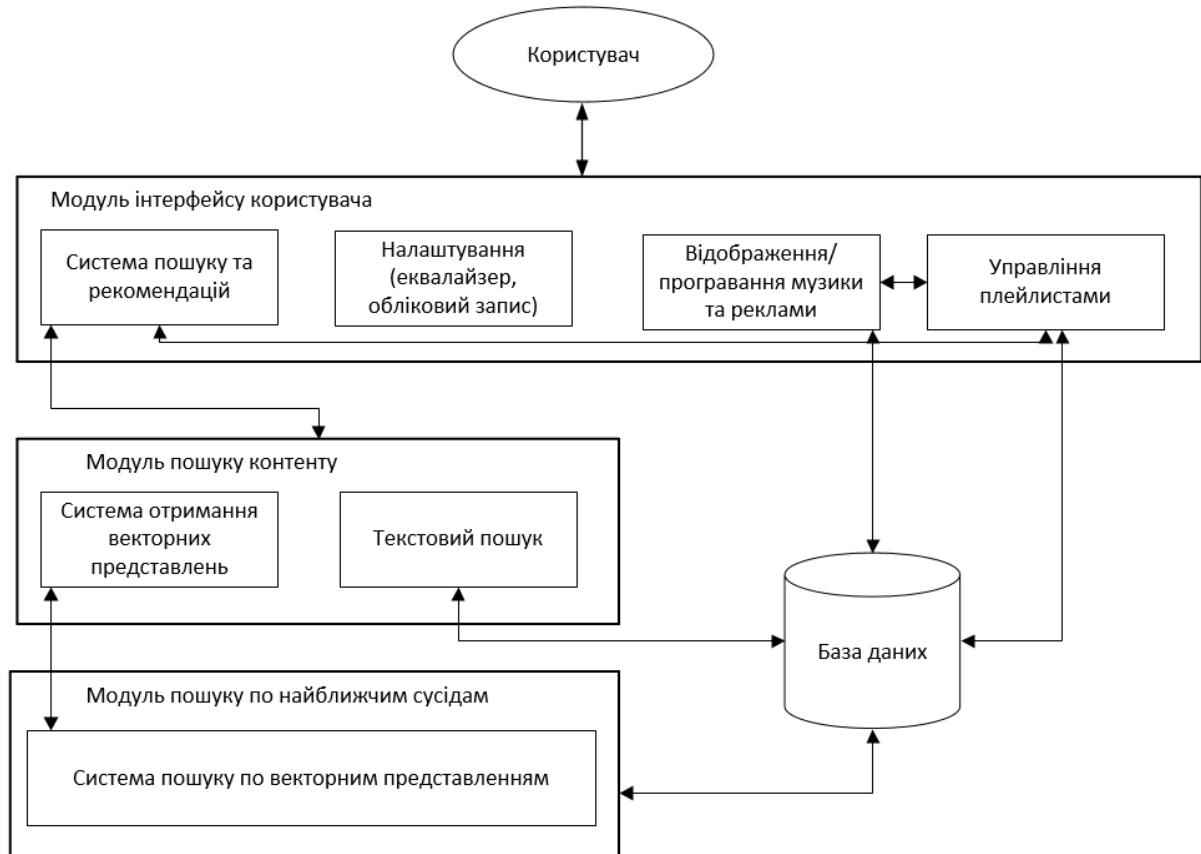
## ДОДАТОК Д

### Структурна схема системи



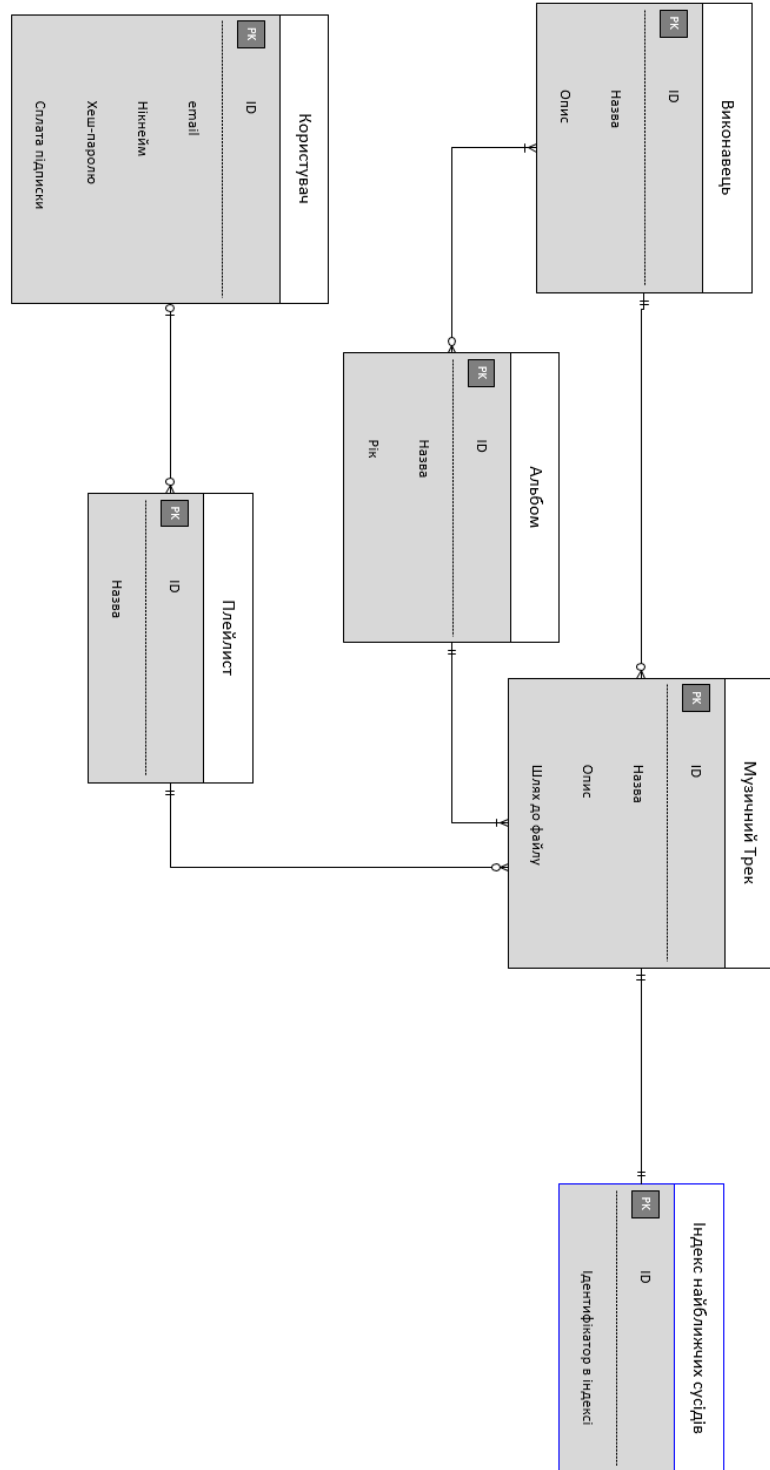
## ДОДАТОК Е

### Функціональна схема системи



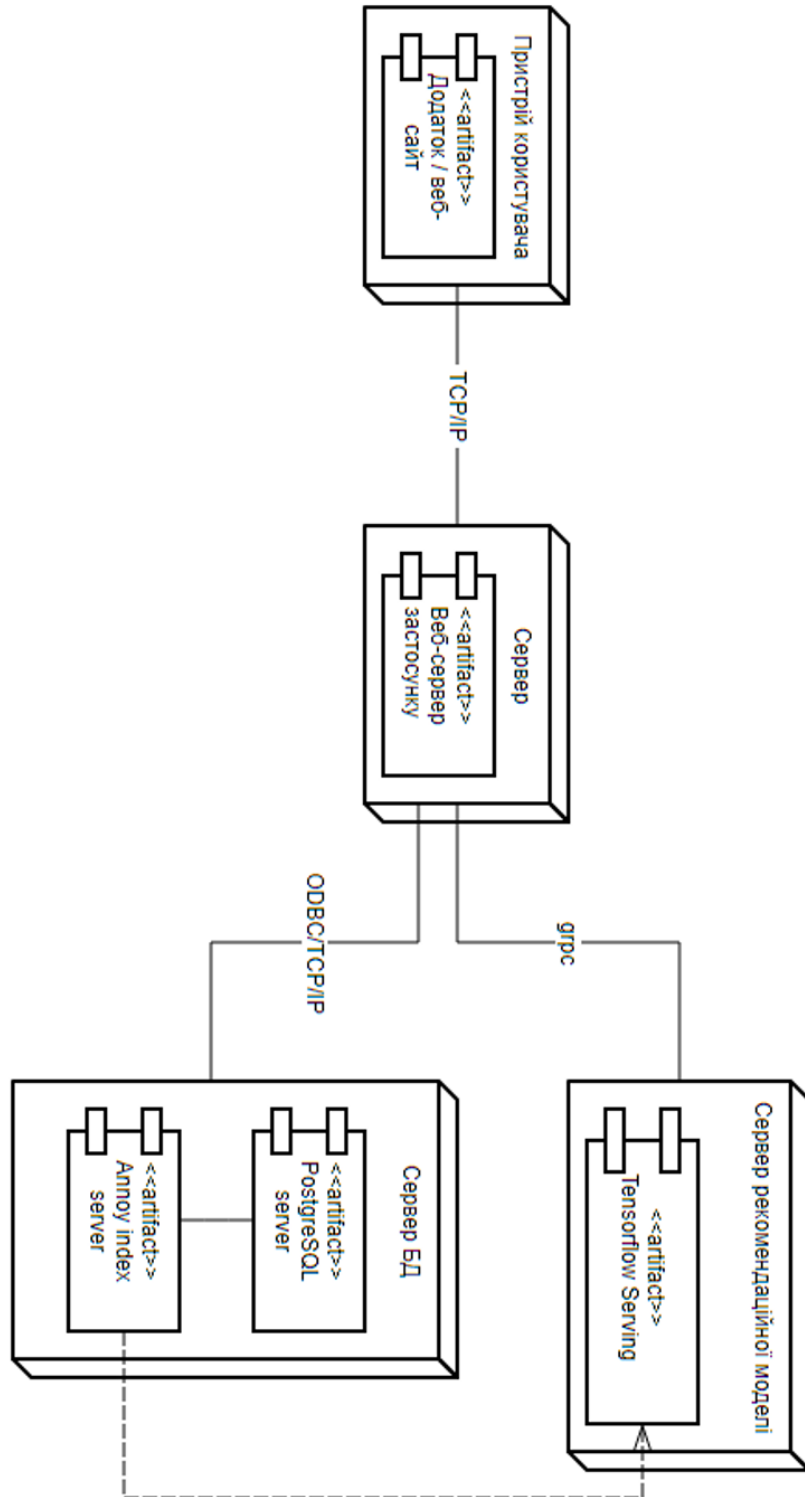
## ДОДАТОК Ж

### ER-діаграма



## ДОДАТОК К

## Діаграма розгортання системи



## ДОДАТОК Л

## Діаграма діяльності остаточного алгоритму

