

Міністерство освіти та науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Факультет інформатики

Кафедра інформатики

## **Кваліфікаційна робота**

освітній ступінь – бакалавр

на тему: **«РОЗРОБКА СИСТЕМИ ДЛЯ ГЕНЕРАЦІЇ МУЗИКИ НА ОСНОВІ  
ФІЗІОЛОГІЧНИХ ПАРАМЕТРІВ»**

Виконав: студент 4-го року навчання,

Спеціальності

121 Інженерія програмного забезпечення

Літвінчук Данило Вікторович

Керівник Франків О. О.

Рецензент \_\_\_\_\_

Кваліфікаційна робота захищена

з оцінкою \_\_\_\_\_

Секретар ЕК \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

Київ – 2023

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Факультет інформатики  
Кафедра інформатики

ЗАТВЕРДЖУЮ

Викладач кафедри інформатики ,  
канд. фіз-мат. наук, доц. \_\_\_\_\_ Гороховський С.С.  
(підпис)  
„\_\_\_\_\_” \_\_\_\_\_ 2023р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на дипломну роботу  
студенту Літвінчуку Данилові Вікторовичу  
факультету інформатики 4 курсу бакалаврської програми  
ТЕМА: Розробка системи для рекомендацій на основі медичних показників.

Зміст ТЧ до дипломної роботи:

Індивідуальне завдання

Вступ

Огляд теоретичного матеріалу і здійснення дослідження

Висновки

Список посилань

Дата видачі „\_\_\_\_\_” \_\_\_\_\_ 2023 р.

Керівник \_\_\_\_\_  
(підпис)

Завдання отримав \_\_\_\_\_  
(підпис)

**Тема: Розробка системи для рекомендацій на основі медичних показників**

| №<br>п/п | Назва етапу курсової роботи                                                                          | Термін виконання<br>етапу | Примітка |
|----------|------------------------------------------------------------------------------------------------------|---------------------------|----------|
| 1.       | Отримання завдання на дипломну роботу.                                                               | 22.09.2023                |          |
| 2.       | Огляд технічної літератури за темою роботи.                                                          | 23.10. 2023               |          |
| 3.       | Виконати аналіз сучасних рішень.                                                                     | 22.11.2023                |          |
| 4.       | Аналіз медичних показників, які можуть бути отримані за допомогою Apple Watch.                       | 26.12.2023                |          |
| 5.       | Аналіз алгоритмів генерації мелодій.                                                                 | 16.02. 2024               |          |
| 6.       | Написання пояснювальної роботи.                                                                      | 02.04.2024                |          |
| 7.       | Аналіз отриманих результатів з керівником.                                                           | 19.04. 2024               |          |
| 8.       | Коригування роботи.                                                                                  | 02.05. 2024               |          |
| 9.       | Створення слайдів для доповіді та написання доповіді.                                                | 13.05.2024                |          |
| 10.      | Аналіз отриманих результатів з керівником, написання доповіді та попередній захист дипломної роботи. | 14.05.2024                |          |
| 11.      | Коригування роботи за результатами попереднього захисту.                                             | 15.05.2024                |          |
| 12.      | Остаточне оформлення пояснювальної роботи та слайдів.                                                | 20.05.2024                |          |
| 13.      | Захист дипломної роботи (проекту)                                                                    | 27.05.2024                |          |

Студент \_\_\_\_\_

Керівник \_\_\_\_\_

“ ”  
\_\_\_\_\_

## ЗМІСТ

|                                                                                                                             |           |
|-----------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>ЗМІСТ.....</b>                                                                                                           | <b>4</b>  |
| <b>АНОТАЦІЯ .....</b>                                                                                                       | <b>5</b>  |
| <b>ВСТУП .....</b>                                                                                                          | <b>6</b>  |
| <b>1 ТЕОРЕТИЧНІ ЗАСАДИ КЛАСИФІКАЦІЇ ЗВУКОЗАПИСІВ ТА ЛЮДСЬКИХ ЕМОЦІЙ В КОНТЕКСТІ ЖИТТЄВИХ ПОКАЗНИКІВ.....</b>                | <b>8</b>  |
| 1.1 Людські емоції та їх вплив на життєдіяльність.....                                                                      | 8         |
| 1.2 Поділ емоцій за фізіологічними ознаками .....                                                                           | 9         |
| 1.3 Огляд принципу аналізу звукозаписів в контексті емоцій .....                                                            | 12        |
| 1.4 Опис датасету для використання.....                                                                                     | 13        |
| Висновки до розділу 1 .....                                                                                                 | 14        |
| <b>2..... ІМПЛЕМЕНТАЦІЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ЗВУКОЗАПИСІВ ТА ПОСЛІДОВНОЇ ГЕНЕРАЦІЇ МЕЛОДІЙ .....</b> | <b>15</b> |
| 2.1 Підходи до аналізу звукозаписів .....                                                                                   | 15        |
| 2.2 Огляд будови рекурентних нейронних мереж .....                                                                          | 17        |
| 2.3 Огляд підходів до класифікації звукозаписів .....                                                                       | 19        |
| 2.4 Формати зберігання аудіозаписів та особливості формату MIDI.....                                                        | 21        |
| 2.5 Створення моделі послідовної генерації нот .....                                                                        | 23        |
| Висновки до розділу 2 .....                                                                                                 | 24        |
| <b>3 СТВОРЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ІМПЛЕМЕНТАЦІЇ МОДЕЛІ ПОСЛІДОВНОЇ ГЕНЕРАЦІЇ МЕЛОДІЙ .....</b>                       | <b>26</b> |
| 3.1 Огляд використаних технологій.....                                                                                      | 26        |
| 3.2 Особливості розробки додатка-компаньйона для watchOS .....                                                              | 28        |
| 3.3 Особливості імплементации моделі для генерації мелодії на iOS.....                                                      | 31        |
| 3.4 Опис алгоритму генерації мелодії за HRV .....                                                                           | 32        |
| Висновки до розділу 3 .....                                                                                                 | 33        |
| <b>ВИСНОВКИ .....</b>                                                                                                       | <b>35</b> |
| <b>СПИСОК ПОКЛИКАНЬ.....</b>                                                                                                | <b>37</b> |
| <b>ДОДАТКИ.....</b>                                                                                                         | <b>39</b> |
| Додаток А. Порівняння способів обчислення HRV .....                                                                         | 39        |
| Додаток Б. Обрахування варіабельності ритму серцебиття.....                                                                 | 40        |

## АНОТАЦІЯ

Ця робота покликана дослідити вплив прослуховуваної музики на HRV (англ. BSR, варіабельність серцевого ритму) та продуктивність праці людини. В ході дослідження було створено класифікатор звукозаписів за збудженістю та позитивністю, послідовну модель, яка здатна генерувати нові ноти на основі вхідних, та два застосунки: для операційної системи watchOS з метою вимірювання ритму серцебиття користувача, та для iOS з метою обробки вхідних даних та програвання згенерованої музики. Серед використаних технологій мова програмування Swift та фреймворки: SwiftUI, WatchConnectivity, TensorFlowLite. Для машинного навчання та аналізу звукозаписів також TensorFlow та openSMILE.

## ВСТУП

Музика здавна слугувала не лише способом насолоди чи розваги, а й забезпечувала терапевтичний ефект. Сучасні дослідження доводять, що це також інструмент, що здатен стимулювати як розумову, так і фізичну активність [14]. Завдяки інформаційним технологіям музика та способи її прослуховування набули чималого поширення, що ледь не кожен пересічний насолоджується улюбленою мелодією під час праці чи занять спортом. Дані активності супроводжуються відповідним емоційним станом, що також впливає на вибір музики користувачем. Таким чином, розробка мобільної системи, що здатна генерувати мелодію з огляду на емоційний стан, має потенціал підвищити продуктивність праці користувача. У цій роботі були вивчені зв'язок між життєвими показниками людини та її емоційним станом, способи послідовної генерації мелодій, можливість їх класифікації за збудженням та позитивністю, а також розроблено додаток для смарт-годинника Apple Watch та смартфона iPhone, що мають на меті імплементувати зазначені практики.

Перший розділ роботи присвячений дослідженню та класифікації емоцій людини за характером, що забезпечить можливість відповідним чином аналізувати звукозаписи. На основі наукових праць розглянуті архітектури моделей машинного навчання та їх ефективність у даній предметній області, а також алгоритми генерації мелодій.

Другий розділ зосереджений на створенні та аналізі моделей машинного навчання для класифікації звукозаписів, а також для генерації мелодій з метою їх застосування в мобільному додатку.

Третій розділ ставить за мету створення мобільного додатку для операційних систем iOS та watchOS, що включатиме прикладне застосування моделей

машинного навчання з огляду на специфіку платформ, створення відповідних користувацьких інтерфейсів за використання нативних технологій та фреймворків.

Сфера дослідження полягає в застосуванні прикладної інформатики та музичних технологій і фокусується на розробці програмних рішень, які використовують машинне навчання для аналізу та впливу на взаємозв'язок між прослуховуванням музики, варіабельністю серцевого ритму (BCR) та загальною працездатністю людини.

Метою цього дослідження є вивчення та розуміння впливу музики на варіабельність серцевого ритму та працездатність людини. Крім того, дослідження спрямоване на розробку технологічних рішень, зокрема, моделей машинного навчання та мобільних додатків, які можуть генерувати музику, адаптовану до емоційного стану користувача, щоб підвищити його продуктивність у різних видах діяльності.

Наукова новизна полягає в розробці класифікатора звукозаписів на основі збудження та позитивності, послідовної моделі для генерації музичних нот, адаптованих до вхідних ознак, та впровадження їх у додатки для iOS та watchOS.

Практичне застосування має потенціал підвищити продуктивність та оптимізувати фізичні тренування шляхом узгодження музики з емоційним та фізіологічним станом користувача.

# 1 ТЕОРЕТИЧНІ ЗАСАДИ КЛАСИФІКАЦІЇ ЗВУКОЗАПИСІВ ТА ЛЮДСЬКИХ ЕМОЦІЙ В КОНТЕКСТІ ЖИТТЄВИХ ПОКАЗНИКІВ

## 1.1 Людські емоції та їх вплив на життєдіяльність

Людські емоції – це складний психологічний стан, що характеризується певним почуттям, наприклад, радість, смуток, гнів, страх, а також фізіологічними змінами, як-от зміна частоти серцевих скорочень, дихання, м'язової напруги. Зазвичай емоції поділяють на позитивні та негативні. Вони можуть виникати у відповідь на певний стимул, зокрема зовнішній: подія, людина, чи внутрішній: думка, спогад. Емоції виконують низку важливих функцій, спонукаючи нас до дії, допомагаючи нам спілкуватися з іншими людьми чи регулювати власну поведінку, а також оцінювати події чи приймати рішення, чим демонструють мотиваційну, комунікативну, регулятивну та оціночну складові.

Існують теорії, що емоції та фізіологічний стан можуть виникати як одночасно, так і передувати одне одному, наприклад, коли ми бачимо щось страхітливе, наше серце починає битися швидше, ми починаємо пітніти, м'язи напружуються, що й змушує нас відчувати страх. Незалежно від того, якій теорії слідувати, можна зрозуміти, що ці двоє тісно пов'язані між собою. Таким чином, позитивні емоції володіють такими властивостями:

- підвищення продуктивності; коли ми відчуваємо себе щасливими та бадьорими, ми більш мотивовані, зосереджені та креативні;
- покращення фізичного та психічного здоров'я; дослідження показали, що позитивні емоції можуть зміцнювати імунну систему, знижувати рівень стресу та ризик хронічних захворювань;
- підвищення стійкості; коли ми стикаємося з труднощами, позитивний настрій може допомогти нам впоратися з ними та швидше відновитися.



Водночас негативні емоції, спричиняють пригніченість та демотивованість, підсилюють роздратованість [2].

## 1.2 Поділ емоцій за фізіологічними ознаками

Емоції, які люди переживають під час взаємодії із навколишнім середовищем асоціюють зі зміною міри фізіологічного збудження. З огляду на це, Джеймс Рассел запропонував зображати їх у вигляді кругової моделі, де вертикальна вісь відповідає за збудженість (arousal), а горизонтальна – за позитивність (valence) (Рис 1.1):

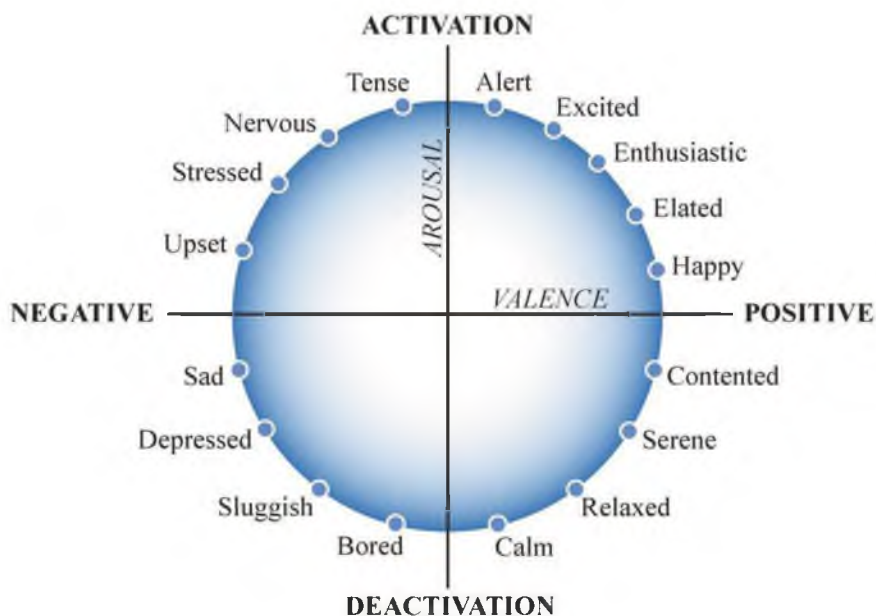


Рис 1.1. Кругова модель класифікації емоцій за їх збудженістю та позитивністю[15]

Підвищене збудження є реакцією автономної нервової системи (АНС), що поділяється на симпатичну (СНС) та парасимпатичну (ПНС), у відповідь на подразники. Вона супроводжується підвищенням кров'яного тиску, ритму серцебиття, частоти дихання, збільшенням потовиділення та ін. Таким чином відбувається пристосування організму до збільшення енергетичних витрат. Коли

подразники зникають, в дію вступає парасимпатична система, що повертає життєві показники до норми. Дональду Хеббу експериментальним шляхом вдалося отримати залежність між рівнем емоційного збудження людини та продуктивністю її праці, яку він зобразив у вигляді кривої:

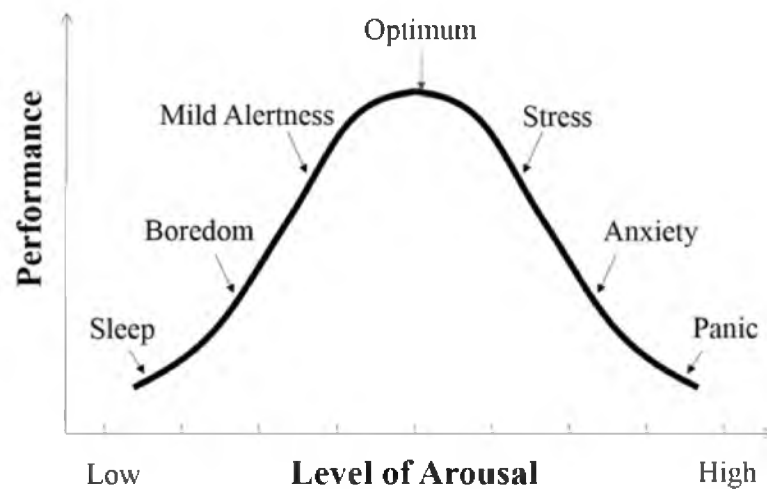


Рис 1.2. Графік залежності продуктивності праці людини від рівня збудженості[15]

А тому можна стверджувати, що пік результативності припадає на середнє значення збудженості.

Натомість варіабельність серцевого ритму вимірює вплив взаємодії між двома складовими АНС на нервову активність серцевого м'яза людини. ВСР –це міра того, як змінюється час між послідовними ударами серця. Навіть під час відпочинку інтервали між його ударами трохи змінюються. У контексті відстежування емоційного стану людини даний показник має ряд переваг над ритмом серцебиття:

- на відміну від показників у частотній області, як-от ритм серцебиття, ВСР дає безпосередню та пряму міру змінності серцевого ритму. Це дозволяє відстежувати відхилення у реальному часі;

- ВСР не є абсолютним значенням, а тому більш стійкий до появи артефактів, а саме поодиноких значень із великим відхиленням, які можуть спотворювати результуючу оцінку.

Існує кілька способів обчислення HRV, їх порівняння наведено у таблиці (див. Додаток А).

Для подальшої роботи було обрано міру SDNN (див Ф. 1.1). Попри те, що значення набуває більшої точності при тривалих вимірюваннях, воно надає більше інформації про загальний баланс АНС.

$$SDNN = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (RR_i - \overline{RR})^2}$$

Ф. 1.1 Формула розрахунку SDNN варіабельності ритму серцебиття де,

- N – кількість вимірювань значень ритму серцебиття
- RR – інтервал між двома ударами,  $\approx \frac{60}{BPM}$

Даний показник набув широкого застосування в різних сферах, включаючи медицину, психологію та спорт. Аналіз ВСР може допомогти оцінити:

- рівень стресу та психофізіологічний стан; хронічний стрес, вигорання та тривога часто асоціюються зі зниженням ВСР;
- когнітивні функції; дослідження показують, що люди з високим ВСР мають кращі когнітивні показники, такі як увага, пам'ять та прийняття рішень;
- серцево-судинну систему; зміни ВСР можуть бути ранніми індикаторами порушень серцево-судинної системи, таких як аритмія та серцева недостатність. Аналіз ВСР допомагає лікарям діагностувати та лікувати різні серцево-судинні захворювання;

- ефективність тренувань та відновлення; в спортивній медицині ВСП використовується для оцінки ефективності тренувань, відновлення та прогнозування ризику травм [17].

### **1.3 Огляд принципу аналізу звукозаписів в контексті емоцій**

Звукозаписи володіють низкою характеристик, які можуть впливати на емоцію, яку вони викликають у слухача. Їх поділяють на спектральні, часові, енергетичні, голосові, мелодійні та ритмічні.

До спектральних належать спектральний центроїд, спектральне згортання та спектральний контраст. Перший вказує на якій частоті зосереджена енергія спектру. Спектральне згортання це частота, нижче якої лежить певний відсоток від загальної спектральної енергії. Спектральний контраст, в свою чергу, характеризує різноманітність спектру.

Часові характеристики включають нульовий перетин, що відображає кількість разів, коли амплітуда сигналу проходить через нуль, та тривалість мовчання – довжина інтервалів без звуку.

Енергія сигналу та динамічний діапазон належать до енергетичних характеристик. Вони означають загальну амплітуду сигналу та різницю між найгучнішими та найтихішими частинами запису відповідно.

Голосові характеристики стосуються безпосередньо записів із вокальною складовою, та включають основну частоту (висоту) голосу, форманти, що є резонансними частотами вокального тракту, а також вірогідність голосу, що є мірою того, наскільки звук наближений до голосового.

До мелодійних належать мелодійний контур – зміни висоти звуку в часі, та хроматичні особливості – інформація про музичні ноти та гармонії.

Ритмічні характеристики включають темп музичного твору та його метричну структуру, що відображає розміщення ударів (акцентів) у музиці [18].

Станом на сьогодні існує кілька інструментів, які дають можливість отримувати перелічені характеристики звукозапису, та жоден не може передбачити, яку емоцію вони викликать у людини без складних алгоритмів машинного навчання та аналізу, які використовуються для розпізнавання емоцій на основі звукових даних. Це пов'язано з тим, що емоційний вплив музики чи мовлення на людину залежить від багатьох факторів, включаючи особистісні особливості слухача, його культурний досвід та контекст прослуховування. Тому, хоча існуючі інструменти можуть точно аналізувати акустичні характеристики, вони потребують додаткових даних та алгоритмів для інтерпретації цих характеристик у термінах специфічних емоційних реакцій.

Таким чином можна сформулювати вимоги для датасету, який може бути використаним для класифікації звукозаписів потрібним чином:

- велика вибірка даних для комплексного аналізу та максимізації точності моделі машинного навчання;
- різноманітність жанрів для забезпечення універсальності класифікації;
- збалансованість даних у датасеті;
- аргументована методика анотацій;
- наявність динамічних анотацій для більш детального аналізу звукозаписів.

## **1.4 Опис датасету для використання**

Для дослідження було використано датасет DEAM (MediaEval Database for Emotional Analysis in Music). Він містить як статичні анотації збудженості та позитивності, так і динамічні з кроком в 0.5 секунди для більш ніж 1800

звукозаписів. Звукозаписи є різними за жанром, походженням, та мовою, що робить його різноманітним та більш придатним для аналізу. Для кожного зі звукозаписів також отримано список із 260 характеристик для кожного кадру з кроком в 45 мілісекунд.

## **Висновки до розділу 1**

У сфері людських емоцій було здійснено багато досліджень, які довели, що будь-яку з них можна класифікувати за мірою їх збудженості та позитивності. Так само доведено, що звукозаписи можуть бути класифіковані за схожим принципом. Оскільки показник ритму серцебиття є абсолютною величиною, він погано підходить для визначення зміни емоції на певному відрізку часу, а тому така величина, як варіабельність серцевого ритму, що розраховується та відображає зміну у часі між двома ударами серця, підходить краще.

## 2 ІМПЛЕМЕНТАЦІЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ЗВУКОЗАПИСІВ ТА ПОСЛІДОВНОЇ ГЕНЕРАЦІЇ МЕЛОДІЙ

### 2.1 Підходи до аналізу звукозаписів

Ми уже розглянули, якими характеристиками можуть володіти звукозаписи. Для їх синтезу належним чином здійснюється перелік кроків для попередньої обробки.

На початку відбувається розбиття запису на частинки тривалістю від 20 до 40 мілісекунд. Наступним кроком застосовується швидке перетворення Фур'є для переведення сигналу із часової площини в частотну, що допомагає краще зрозуміти природу звуку.

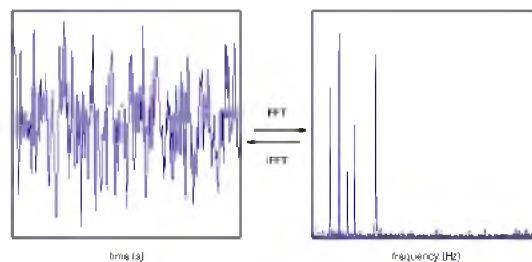


Рис. 2.1. Порівняння спектрограми звукової хвилі в часовому та частотних просторах[16]

Далі не менш важливим етапом можна вважати обробку частот звукозапису за допомогою Мел-шкали. Справа в тому, що людський слух дуже чутливий до змін низьких частот, чого не скажеш про високі, а тому нам майже неможливо розрізнити все, що перевищує 15 kHz, а звукозаписи володіють і такими. Таким чином, аби очистити вихідні дані, застосовується приведення діапазону частот за шкалою таким чином:

$$M(f) = 1125 \ln\left(1 + \frac{f}{700}\right)$$

### Ф. 2.1 Формула для обчислення Мел-показника за частотою

Для того, щоб виконати зворотне перетворення застосовуємо формулу:

$$M^{-1}(m) = 700 \left( \exp\left(\frac{m}{1125}\right) - 1 \right)$$

### Ф. 2.2 Формула для отримання частоти хвилі за Мел-показником

Далі, оскільки людський слух сприймає посилення енергії звуків пропорційно логарифмічній функції, саме за її допомогою ми масштабуємо отримане значення. Зважаючи на те, що кожен елемент банку фільтрів Мела в частотній області має трикутну форму, в результуючій спектрограмі ми отримуємо численні кореляції, які є перешкодою для подальшого аналізу. Саме тому застосовується дискретне косинусне перетворення для отримання Мел-частотних кепстральних коефіцієнтів (MFCC).

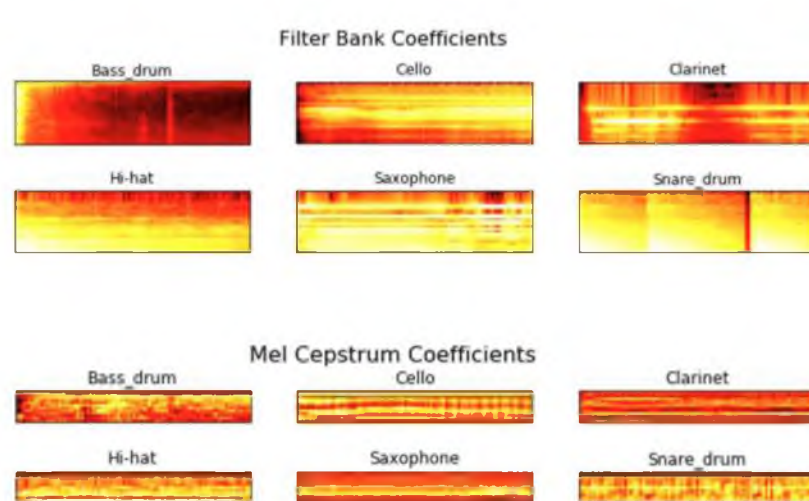


Рис. 2.1 Спектрограма аудіо-хвилі за її Мел-показниками(згори).

Спектрограма аудіо-хвилі за її кепсптральними коефіцієнтами(знизу) [16]



Спектрограми знизу є досить показовими, адже зображені значення є типовими для музичних інструментів незалежно від структури звукозапису.

На щастя, зараз немає потреби виконувати подібні обчислення власноруч, натомість у вільному доступі існують open-source рішення, як-от бібліотека openSMILE. Вона широко використовується не лише для аналізу аудіо-сигналів, а також і відео, надаючи такі характеристики як висота звуку, енергія, тембр та інші [8].

## 2.2 Огляд будови рекурентних нейронних мереж

Найбільш поширеним підходом до класифікації звукозаписів вважається використання машинного навчання, а саме глибокого навчання. Його основною перевагою є можливість ефективно використовувати великі масиви даних для навчання складних моделей, здатних встановлювати складні взаємозв'язки в аудіоданих. Це відбувається за допомогою численної кількості шарів нейронів, пов'язаних між собою.

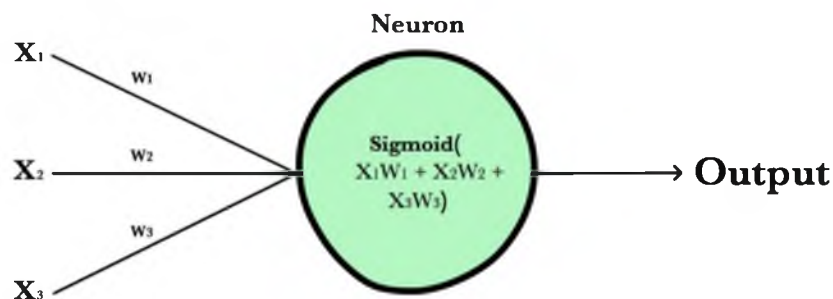


Рис. 2.2 Будова нейрона нейронної мережі[13]

Важливою складовою нейрона є функція активації, яку він застосовує до отриманих значень з метою визначити, чи буде отримане значення переданим далі

по ланцюгу взаємодії. Наприклад, функція ReLU (rectifier linear unit) викликає активацію, якщо вихідне значення нейрона є більшим за нуль. Це призводить до меншої кількості активацій нейронів загалом, чим покращується ефективність та швидкість навчання моделі.

Будова моделі глибокого навчання складається з трьох основних компонентів:

- вхідний шар; кількість нейронів у ньому відповідає розміру вектора вхідних даних;
- приховані шари; їх кількість та кількість нейронів, яку вони містять не є обмеженими. Необхідно підібрати таку комбінацію, за якої точність вихідних даних моделі є найбільшою;
- вихідний шар; кількість нейронів визначена розміром вектору вихідних даних моделі.

Одним з ключових методів глибокого навчання є алгоритм зворотного поширення (backpropagation), який дозволяє нейронним мережам поширювати помилки у зворотному напрямку, оптимізуючи ваги мережі для мінімізації цих помилок. Рекурентні нейронні мережі (RNNs) відрізняються від звичайних нейронних мереж тим, що вони можуть обробляти дані послідовностей та зберігати інформацію про попередній стан у своїх внутрішніх петлях, що дозволяє їм використовувати минулий контекст при обробці нових даних. Така здатність дозволяє подібним мережам демонструвати динамічну поведінку, що робить їх добре придатними для таких завдань, як розпізнавання мови, моделювання мови та прогнозування часових рядів. Однак головний недолік стандартних рекурентних нейронних мереж часто полягає в проблемі зникаючого градієнта при великих наборах даних, коли градієнти зменшуються або збільшуються експоненціально під час зворотного поширення, що робить їх неефективними при вивченні довготривалих залежностей.

Поява мереж із довгою короткостроковою пам'яттю мала на меті виправити проблему зі зникаючим градієнтом. LSTM підтримують більш складну внутрішню структуру в кожному нейроні, аніж стандартні RNN, яка включає три типи воріт: вхідні ворота, ворота забування та вихідні ворота. Такий механізм покликаний регулювати потік інформації в нейрон і з нього, а також контролювати його стан, що є ключовим нововведенням для вирішення проблеми зникаючого градієнта.

Згодом, була запропонована покращена архітектура мереж із довгою короткостроковою пам'яттю, яка отримала назву “Мережі із двонаправленою довгою короткостроковою пам'яттю”, BiLSTM. Їхня особливість полягає у тому, що вони здатні зберігати не лише минулий контекст, а й майбутній, що відчутно покращує їхню точність [19].

### **2.3 Огляд підходів до класифікації звукозаписів**

Існує ціла низка праць спрямованих на розробку концептуально нових способів класифікації звукозаписів чи покращення вже існуючих.

У дослідженні 2017 року “Multi-scale Context Based Attention for Dynamic Music Emotion Prediction:”, модель LSTM показала кращі результати порівняно з моделями MLR (Multiple Linear Regression, Багатофакторна Лінійна Регресія), SVR (Support Vector Regression – Регресія на Опорних Векторах), та GPR (Gaussian Process Regression – Регресія за Процесом Гауса), демонструючи ефективність використання LSTM для точного прогнозування емоцій у музиці. У 2022 році у роботі “DBLSTM-Based Multi-Scale Fusion for Dynamic Emotion Prediction in Music”, комбінація DBLSTM (Deep Bidirectional LSTM – Глибокий Двонаправлений LSTM) з MLR, SVR, ELM (Extreme Learning Machine – Машина Екстремального Навчання), та ANN (Artificial Neural Network – Штучна Нейронна Мережа) забезпечила

виявлення впливу довжини послідовностей на результати, і найкращі результати були отримані завдяки пост-обробці після злиття [4].

Ще одне дослідження 2022 року “Study on Music Emotion Recognition Based on the Machine Learning Model Clustering Algorithm” використовувало SVR та RBRF (Random Ball Cover Radius Based Functions – Функції, Засновані на Радіусі Випадкового Кулевого Покриття), показуючи кращі результати порівняно з традиційними методами SVR та SVM (Support Vector Machine – Машина Опорних Векторів). Важливий прогрес було досягнуто також у дослідженні “Music emotion recognition based on segment-level two-stage learning”, де CNN-based autoencoder (Convolutional Neural Network-based Autoencoder – Автокодер на Основі Згорткової Нейронної Мережі) та BiLSTM (Bidirectional LSTM – Двонаправлений LSTM) модель досягли найкращих результатів валентності на 1-секундних сегментах датасету PMEmo [6].

Останнє дослідження “Music emotion recognition using convolutional long short term memory deep neural networks” 2021 року використовувало LSTM та DNN (Deep Neural Network – Глибока Нейронна Мережа), що забезпечило збільшення продуктивності у порівнянні з моделями SVM, KNN (K-Nearest Neighbors – Метод k-найближчих Сусідів), і RF (Random Forest – Випадковий Ліс) після застосування CFS (Correlation-based Feature Selection – Вибір Ознак на Основі Кореляції). Ці результати підкреслюють значення глибокого навчання та гібридних моделей у вирішенні завдань розпізнавання емоцій у музиці, зокрема у динамічному визначенні емоційних станів [5].

У праці, що була опублікована університетом Шрі-Ланки, було проаналізовано наведені підходи, а також запропоновано альтернативний, що у порівнянні показує більшу точність динамічного розпізнавання емоцій, такого, що засноване на анотаціях величин збудженості та позитивності для кожного складового відрізка, а не цілковитого аудіозапису. Для побудови моделі були

використані BiLSTM шар у поєднанні із кількома глибинними шарами. Експериментально було визначено, що оптимальна кількість шарів дорівнює шести. Точність такої моделі склала 85.32%, що показало приріст у 4% в порівнянні з існуючими рішеннями [3].

## **2.4 Формати зберігання аудіозаписів та особливості формату MIDI**

У міру того, як аудіозаписи різних видів набули широкого застосування серед кінцевих споживачів, з'явилася й велика кількість форматів їх зберігання, що покликані задовольняти їх потреби. Найпопулярніші можна розділити на дві основні категорії: формати зі стисненням з втратами та формати без втрат. Кожна категорія має свої переваги та недоліки:

- **Формати зі стисненням з втратами:**

- MP3 (MPEG Audio Layer 3): Залишається найпопулярнішим форматом завдяки своїй універсальності та широкій підтримці на різних пристроях. MP3 використовує алгоритми стиснення, які зменшують розмір файлу, видаляючи деякі звукові дані, які вважаються менш чутливими для людського вуха. Це дозволяє зберігати велику кількість музики в пам'яті пристрою, але може призвести до помітної втрати якості звуку.

- AAC (Advanced Audio Coding): Розроблений як наступник MP3, AAC зазвичай забезпечує кращу якість звуку при тому ж рівні стиснення. Це досягається завдяки вдосконаленим алгоритмам кодування, які ефективніше зберігають звукову інформацію. AAC широко використовується в Apple Music та інших сервісах потокового аудіо, а також для зберігання музики на мобільних пристроях.

- OGG Vorbis: Відкритий і безкоштовний формат, який пропонує гарний компроміс між якістю звуку та розміром файлу. Він використовує гнучкі алгоритми стиснення, які дозволяють налаштовувати рівень якості звуку залежно від потреб. OGG Vorbis підтримується багатьма програвачами та пристроями, але не настільки широко, як MP3 або AAC [20].

**- Формати без втрат**

- FLAC (Free Lossless Audio Codec): Відкритий формат, який зберігає аудіо без втрати якості. Він використовує алгоритми стиснення, які зменшують розмір файлу без видалення будь-яких звукових даних. Це дозволяє отримати аудіо найвищої якості, але файли FLAC зазвичай займають більше місця, ніж MP3 або AAC. FLAC є популярним вибором серед аудіофілів та музикантів, які цінують якість звуку.

- ALAC (Apple Lossless Audio Codec): Власний формат Apple, який є аналогом FLAC. Він також зберігає аудіо без втрати якості, але використовує власні алгоритми стиснення. Користувачам техніки Apple не слід обирати між цими двома, адже два підтримуються пристроями компанії, проте більшого розповсюдження ALAC не набув.

- WAV (Waveform Audio File Format): Нестиснений формат, який зберігає аудіо в оригінальній якості. Файли WAV зазвичай займають найбільше місця, але вони є стандартом для професійного запису та редагування звуку. WAV також використовується для зберігання звукових ефектів та інших аудіоданих, де важлива максимальна якість [20].

Натомість існує ще один формат – MIDI. Він відрізняється від інших форматів тим, що не зберігає сам звук, а інструкції для відтворення музики, такі як ноти, їх тривалість, гучність та інші параметри. Це дозволяє відтворювати музику на різних пристроях з різними звуками, що робить MIDI універсальним інструментом для

створення та редагування музики. MIDI-файли зазвичай набагато менші за розміром, ніж аудіофайли, але якість звучання залежить від інструментів та синтезаторів, що використовуються. Також цей формат є чудовим вибором для машинного навчання, адже структуровані дані MIDI легко аналізувати та обробляти алгоритмами, дозволяючи генерувати нову музику, класифікувати жанри та виконувати інші завдання музичного аналізу.

## 2.5 Створення моделі послідовної генерації нот

Для створення моделі було використано датасет MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization), що включає двісті годин звукозаписів класичної музики у форматах MIDI та WAV. Для парсингу MIDI-файлів на окремі ноти використано бібліотеку `pretty_midi`. Якщо за її допомогою зобразити аудіозапис на площині, де вертикальна вісь відображає гучність ноти (значення коливається від 0 до 128), а на горизонтальній у вимірі часу зображені її тривалість та час, який пройшов після звучання попередньої ноти, побачимо наступне (див Рис. 2.3):

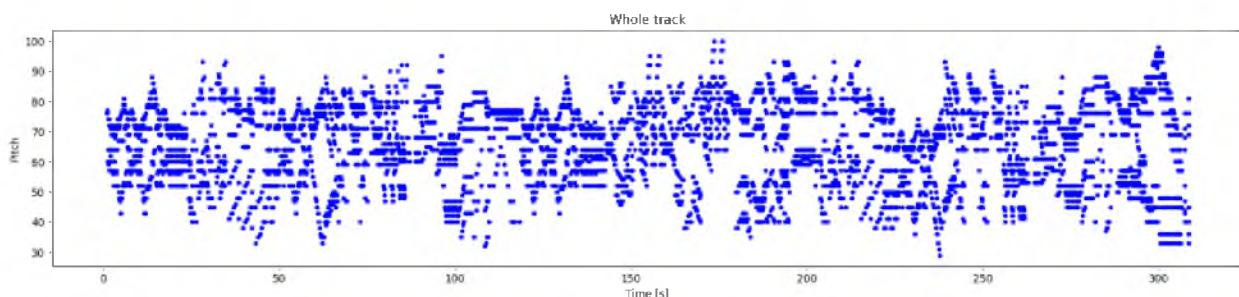


Рис. 2.3 Графік зображення нот аудіозапису

Для перевірки роботи бібліотеки звукозаписи у MIDI форматі були прочитані та представлені у вигляді об'єкту `DataFrame`, а потім конвертовані назад до

початкового формату та відтворені. Звучання не змінилося, що свідчить зворотність перетворень.

Наступним кроком аудіозаписи було поділено на послідовності із 26 нот, 25 з яких слугуватимуть вхідними даними, а 26а – вихідними. Запропонована модель складатиметься із вхідного шару, LSTM-шару розмірністю 128, та трьох вихідних шарів розмірностями 128, 1, 1 для гучності, кроку та тривалості ноти відповідно. Оскільки для пошуку оптимальних ваг використовується метод градієнтного спуску, застосовано оптимізатор Adam, що відзначається більшою ефективністю у порівнянні з альтернативами, такими як AdaGrad, RMSProp та AdaDelta. Модель тренуватиметься впродовж 50 ітерацій, а щоб запобігти її перетренованості, використано функцію переривання на основі метрики loss, яка характеризує наскільки отриманий результат відрізняється від справжнього значення. На графіку продемонстровано зміну метрики впродовж 35 ітерацій, що сигналізує про передчасну зупинку тренування. Значення похибки коливалося від 0.42 до 0.31 (див Рис. 2.4):

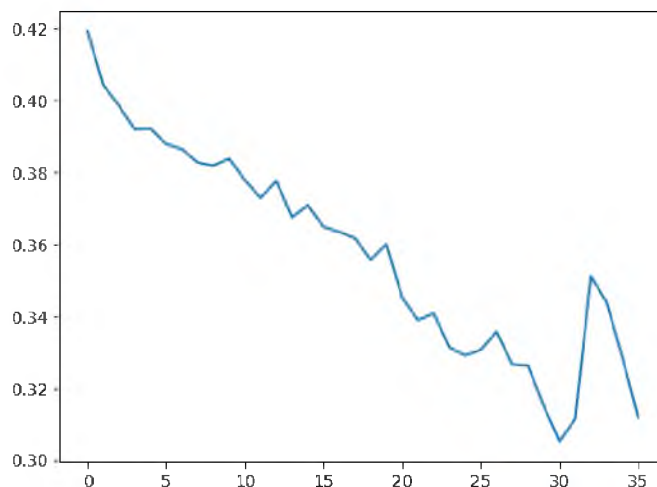


Рис. 2.4 Графік зміни метрики loss моделі впродовж її тренування

## Висновки до розділу 2



Провідним способом аналізу звукозаписів є використання машинного навчання, що підтверджують численні дослідження у даній сфері. Серед них особливо добре для такого завдання підходять рекурентні нейронні мережі, що здатні не лише аналізувати складні зв'язки між даними, а й запам'ятовувати попередні стани метою оптимізації процесу навчання. Продемонстровано, що для класифікації звукозаписів за їх збудженням та позитивністю найкраще підходять мережі із BiLSTM шаром та 5 глибинними шарами нейронів. Серед усіх форматів звукозаписів існує один – MIDI, що не містить в собі безпосередньо звук, а лише інструкції для його виконання. Така структурованість MIDI робить його найкращим вибором для аналізу та машинного навчання. За його допомогою та уже згаданих рекурентних нейронних мереж можливо створити модель машинного навчання, що за послідовністю із нот генеруватиме наступний її елемент.

## 3 СТВОРЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ІМПЛЕМЕНТАЦІЇ МОДЕЛІ ПОСЛІДОВНОЇ ГЕНЕРАЦІЇ МЕЛОДІЙ

### 3.1 Огляд використаних технологій

Попри кількість та різноманіття технологій, що дозволяють створювати мультиплатформенні мобільні застосунки, вирішено розробити нативний за використання мови програмування Swift, що забезпечує такі переваги: оптимальна продуктивність та швидкість роботи додатка, глибока інтеграція з iOS екосистемою, що забезпечує краще використання апаратних можливостей пристроїв та забезпечення кращого користувацького досвіду з використанням останніх фреймворків Apple.

Як середовище розробки обрано Xcode, який володіє багатьма особливостями, що покликані спростити та пришвидшити процес розробки мобільних застосунків. Серед таких особливо корисним є Xcode Previews. Це потужний інструмент, що значно покращує процес розробки інтерфейсів у Xcode, зокрема для SwiftUI. Ця функція дозволяє розробникам швидко переглядати та ітеративно налаштовувати свій UI в реальному часі без необхідності постійно компілювати та запускати додаток на емуляторі чи фізичному пристрої. Previews відображає UI компоненти відповідно до заданих параметрів і станів, що розробник може легко модифікувати. Це не тільки спрощує процес розробки, але й забезпечує більшу гнучкість у дизайні і ефективність в тестуванні змін інтерфейсу.

Для створення користувацького інтерфейсу було використано вже згаданий фреймворк SwiftUI. SwiftUI — це декларативний фреймворк для створення користувацьких інтерфейсів, що інтегрується з екосистемою Apple (iOS, macOS, watchOS, tvOS). Йому передував фреймворк під назвою UIKit, і що важливо розуміти, вони не взаємозаперечують один одного та можуть використовуватися одночасно, а це надає такі переваги:

- легший та швидший перехід з UIKit на SwiftUI; мігрувати старі та масштабні проекти може бути ледь не нездійсненним завданням, проте у даному випадку це можна робити крок за кроком, переписуючи окремі компоненти по черзі;
- можливість розширити набір компонентів SwiftUI за допомогою UIKit; SwiftUI володіє обмеженою кількістю компонентів через свою новизну, а тому може бути легко доповненим компонентами із UIKit за умови використання протоколу `UIViewRepresentable`.

У ролі фреймворку для імплементації машинного навчання було обрано TensorFlow Lite через його високу ефективність у виконанні легких моделей на пристроях з обмеженими обчислювальними ресурсами, таких як мобільні телефони та вбудовані системи. TensorFlow Lite забезпечує швидку реалізацію машинного навчання за допомогою оптимізованого виконання моделей, меншого використання пам'яті та енергії. Це дозволяє застосовувати складні алгоритми навчання безпосередньо на кінцевих пристроях, забезпечуючи кращу інтеграцію та залучення користувачів. CoreML, фреймворк від Apple, володіє схожими характеристиками і навіть здатний використовувати в повній мірі апаратні можливості пристроїв, проте ключовою перевагою TensorFlow Lite стала можливість зберігати претреновані моделі віддалено, що у свою чергу робить можливим їх удосконалення без потреби перезбирати цілий додаток заново. Варто лише оновити файл моделі до віддаленого сховища Firebase, та вона буде завантаженою на пристрій та придатною для користування.

Взагалі, Firebase володіє широким спектром можливостей, які стали невід'ємними в сучасній мобільній розробці. Він надає рішення для аутентифікації користувачів, баз даних у реальному часі, хмарного зберігання, функцій на стороні сервера через Cloud Functions, та пуш-нотифікацій. Особливо корисним вважаю Firebase Crashlytics, який дозволяє відстежувати та аналізувати збої програми,

забезпечуючи детальну інформацію про причини помилок і допомагаючи швидше вирішувати проблеми зі стабільністю програми. Це особливо корисно для підтримки високої якості додатків на етапі їхнього розгортання та експлуатації.

### 3.2 Особливості розробки додатка-компаньйона для watchOS

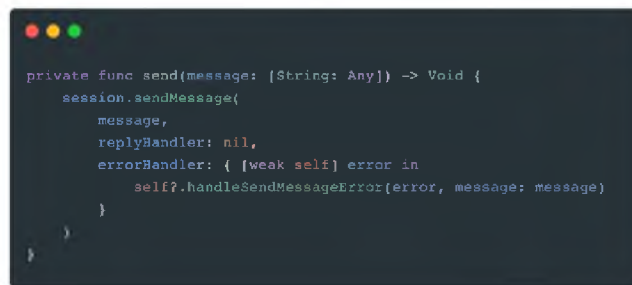
Попри свою компактність, Apple Watch є досить потужним пристроєм, що дозволяє розробникам створювати комплексні застосунки, із порівняно високим набором функціоналу в поєднанні з чудовим користувацьким досвідом. Та їх головний недолік полягає в обмеженості часу акумулятора від одного заряду. Якби смарт-годинник у повній мірі реалізовував задуманий функціонал, то розрядився б за кілька годин. Таким чином було прийнято рішення розробити два додатки-компаньйони, один для Apple Watch, а інший для iPhone. Відповідальність останнього полягатиме в обробці та аналізі отриманих даних, а також використанні моделей машинного навчання для генерації мелодій.

На апаратному рівні комунікація між двома пристроями здійснюється за допомогою Bluetooth Low Energy, Wi-Fi чи стільникової мережі в залежності від віддаленості пристроїв. На програмному рівні вона здійснюється за допомогою WatchConnectivity фреймворка. Для реалізації необхідно:

1. Ініціалізувати WCSSession: Налаштувати WCSSession на обох пристроях (iPhone і Apple Watch). Сесія має бути активована на кожному пристрої для забезпечення двосторонньої комунікації.
2. Імплементувати делегата: Призначити об'єкт-делегат типу WCSSessionDelegate, який буде обробляти вхідні дані та зміни стану сесії.
3. Обмін даними: Використовувати методи WCSSession для передачі даних. Це можуть бути файли, користувацькі дані (userInfo) чи живі повідомлення, залежно від потреби.

4. Обробка помилок та зміни стану: Імплементувати методи делегата для обробки помилок та відслідковування стану сесії та з'єднань.

Після виконання перелічених кроків повідомлення може бути відправленим за допомогою методу `WCSession.sendMessage(_:replyHandler:errorHandler:)`. Першим параметром є саме повідомлення у вигляді словника ключ-значення. Другий – `replyHandler`, що є замиканням, дозволяє обробити відповідь від одержувача на відправлене повідомлення. Третім параметром є замикання `errorHandler`, що відпрацьовує при виникненні помилки при відправці повідомлення (див. Лістинг 3.1).



```
private func send(message: [String: Any]) -> Void {
    session.sendMessage(
        message,
        replyHandler: nil,
        errorHandler: { [weak self] error in
            self?.handleSendMessageError(error, message: message)
        })
}
```

Лістинг 3.1 Реалізація відправки повідомлення з AW на iPhone

Попри те, що імплементувати алгоритм комунікації між пристроями досить легко, в процесі можуть виникати непередбачувані проблеми. Наприклад, можна стикнутися з тим, що не вдається відправити повідомлення іншому пристрою, хоча система просигналізувала, що з'єднання встановлено. У зв'язку з цим було імплементовано механізм, за яким у разі невдалої відправки повідомлення пристрій робить певну кількість повторних спроб, і у разі, якщо всі невдачі – сповіщає користувача про потребу перевірити з'єднання чи перезапустити застосунок.



```
private func handleSendMessageError(_ error: Error, message: [String: Any]) {
    if (error as NSError).code == 7007,
        numberOfRetries > 0
    {
        numberOfRetries -= 1
        let randomDelay = Double.random(in: (0.3...1.0))
        DispatchQueue.main.asyncAfter(deadline: .now() + randomDelay) {
            self.send(message: message)
        }
    } else {
        numberOfRetries = Constants.maxNumberOfRetries
    }
}
```

Лістинг 3.2. Механізм повторної відправки повідомлення при помилці

Дані для відправки будуть отримані за допомогою фреймворка HealthKit. Він надає широкий доступ до здоров'я та фітнес-даних, що зберігаються на пристроях користувачів, дозволяючи додаткам отримувати інформацію про фізичну активність, серцевий ритм, якість сну та інші параметри здоров'я. Він також володіє можливістю співставляти дані про активність з різних пристроїв та об'єднувати їх за потреби, що підвищує точність та повноту вимірювань. Такі дані є чутливими, а отже операційна система вимагає згоди користувача на поширення цих даних для кожного застосунку. Зробити запит можливо, якщо додати відповідне поле до “Info.plist” файлу, вказавши призначення, за яким будуть використовуватися дані. Це також слугує інформацією для App Store, що застосунок відповідає політиці конфіденційності Apple.

Серед усього, HealthKit надає доступ до даних про варіабельність серцевого ритму користувача, який можна було б використати для відстежування його емоцій для генерації мелодій, та не все так просто. Справа у тому, що вимірювання цього показника здійснюється 5-6 разів на добу, що не дозволило б нам отримувати значення у часі, близькому до реального. А тому було прийнято рішення відстежувати показник ритму серцебиття. Можливо, такий спосіб поступається в

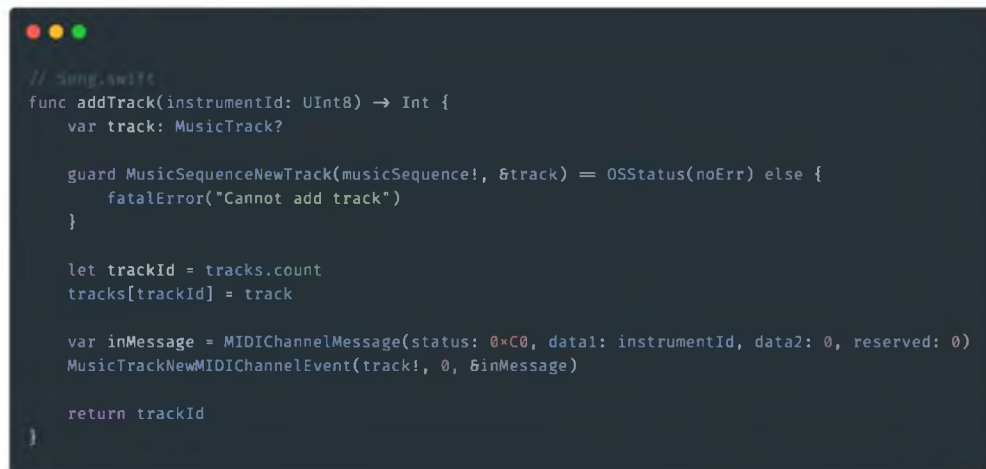
точності тому, що пропонує HealthKit, проте питання наближеності показника до реального часу є критичним для імплементації бажаного функціоналу. Таким чином після того, як ритм серцебиття було виміряно за допомогою Apple Watch та відправлено на iPhone, за допомогою формули є змога обрахувати SDNN варіабельності ритму серцебиття (див Додаток Б). Оскільки дана величина розраховується на основі ряду значень показників ритму, що більше вимірювань ми маємо, то більшу точність обчислення отримуємо. Історія значень зберігається у масиві. Аби запобігти переповненню масиву, а також зберегти високу швидкість обчислення, запам'ятовуються лише останні 200 значень.

### **3.3 Особливості імплементації моделі для генерації мелодії на iOS**

На iPhone для програмування та відтворення музичних нот можна використовувати кілька фреймворків, як-от AVFoundation, AudioToolbox, та CoreAudio. AVFoundation дозволяє завантажувати та відтворювати звукові файли. AudioToolbox призначений для більш детального управління аудіо потоками, включаючи генерацію тонів та відтворення коротких звуків. CoreAudio надає найвищий рівень контролю над аудіо даними, ідеальний для складних музичних додатків. Оскільки згенеровані ноти є такими, які містить в собі MIDI-файл, то необхідно знайти можливість його відтворювати.

Першим кроком є завантаження колекції звуків та інструментів, які можуть бути використані MIDI програвачем для генерації реальних аудіо сигналів з MIDI даних. MIDI файли самі по собі не містять аудіо, вони містять лише інструкції про те, які ноти грати, їх тривалість та інтенсивність. Це також надає можливість відтворювати ноти за допомогою будь-яких інструментів на смак користувача. Для цього додатку було обрано вільний банк звуків “Nokia Tongbao Bank 8-bit version”.

Другим кроком є створення `MusicSequence` та `MusicTrack`. `MusicSequence` – об’єкт, який утримує колекцію `MusicTracks` і керує загальними аспектами музичної композиції, такими як темп. `MusicTrack` – це окрема доріжка в `MusicSequence`, яка може містити різні MIDI події, такі як `MIDINoteMessage`, які вказують на конкретні ноти, їх висоту, силу удару та тривалість. Наприклад, на рисунку нижче продемонстровано створення нового `MusicTrack`. Цікавим є ініціалізація `MIDIChannelMessage`, адже він є свого роду інструкцією для комунікації із MIDI драйвером. Інструкція `0xC0` вказує на те, аби для відтворення було використано інструмент з ідентифікатором `instrumentId`. Важливо зазначити, що не всі банки звуків підтримують всі інструменти, а тому отримати список підтримуваних можна за допомогою функції `CopyInstrumentInfoFromSoundBank` бібліотеки `AudioToolBox`, або на сторінці документації банку.



```
// Song.swift
func addTrack(instrumentId: UInt8) → Int {
    var track: MusicTrack?

    guard MusicSequenceNewTrack(musicSequence!, &track) == OSStatus(noErr) else {
        fatalError("Cannot add track")
    }

    let trackId = tracks.count
    tracks[trackId] = track

    var inMessage = MIDIChannelMessage(status: 0xC0, data1: instrumentId, data2: 0, reserved: 0)
    MusicTrackNewMIDIChannelEvent(track!, 0, &inMessage)

    return trackId
}
```

Лістинг 3.3. Створення запису із ідентифікатором інструменту у вигляді параметру

### 3.4 Опис алгоритму генерації мелодії за HRV



На основі розглянутих концептів можливо описати алгоритм роботи програмного комплексу задля генерації мелодій.

Процес починається з ініціації користувачем мобільного застосунку, який синхронізується з Apple Watch для збору даних про серцевий ритм. Відбувається запит згоди користувача на отримування даних про його серцебиття, у разі відмови – додаток не може продовжувати свою роботи. Варто зазначати, що за існування двох додатків для Apple Watch та iPhone, достатньо зробити запит на одному з них. Далі, дані передаються на iPhone, де відбувається їхній аналіз і перетворення у параметр HRV SDNN, що відображає варіабельність серцевого ритму. На основі значення HRV SDNN система визначає емоційний стан користувача і вибирає відповідну музичну послідовність для відтворення. Вхідна послідовність для генерації мелодії отримана із заздалегідь класифікованих звукозаписів за їх збудженістю та позитивністю. Записи із від'ємною позитивністю не входять до вибірки вхідних даних. Отримана послідовність перетворюється в об'єкт, який може бути відтвореним за допомогою вбудованих можливостей фреймворку Core Audio. Програвана мелодія покликана сприяти нормалізації фізіологічних показників і покращенню самопочуття.

### **Висновки до розділу 3**

Мова програмування Swift та середовище розробки Xcode є інструментами, що забезпечують сучасний, простий та швидкий процес розробки мобільних застосунків. Через свій розмір Apple Watch не володіють достатньою ємністю акумулятора, а тому не підходять для тривалого виконання складних завдань, таких як використання моделей машинного навчання. Вихід в такій ситуації – створення додатку компаньйона для смартфона iPhone, що дозволяє розподілити функціонал

між пристроями, з можливістю легко встановити зв'язок для обміну інформацією. iOS володіє усім необхідним набором інструментів, що дозволять відігравати послідовності нот у вигляді MIDI-файлів.

## ВИСНОВКИ

У даній роботі було розглянуто природу людських емоцій, та який вплив їх зміни мають на фізичний стан та життєві показники людини. Проаналізовано спосіб їх класифікації за мірою збудженості та позитивності, що корелюється з тим, як можуть бути класифіковані звукозаписи. Проведено дослідження методів обробки та аналізу звукозаписів з метою отримання їх характеристик. Проаналізовано існуючі підходи до класифікації звукозаписів, виявлено їх переваги та недоліки, визначено найбільш ефективну архітектуру для даного завдання. На основі датасету, що містить анотації збудженості та позитивності для більш як 1800 звукозаписів, створено рекурентну модель машинного навчання.

Проаналізовано формати зберігання аудіозаписів, а особливо формат MIDI. Він не зберігає сам звук, а лише інструкції для його відтворення, що дозволило застосувати його для створення моделі, що за послідовністю нот здатна генерувати наступну.

Для імплементації практик машинного навчання було розроблено два мобільних застосунки: для iOS та watchOS. Кожен з них володіє окремим функціоналом задля забезпечення якнайкращого користувацького досвіду. Обмін інформацією між ними здійснюється за допомогою фреймворку WatchConnectivity. Моделі машинного навчання впроваджені за допомогою Firebase, що забезпечує можливість їх покращення без потреби створення нової збірки.

Результатом дослідження став програмний комплекс, що за допомогою ритму серцебиття, ряд вимірювань якого перетворюється в показник BCP, здатен генерувати послідовності нот, які відтворюються мобільним застосунком у вигляді мелодії. Таке звучання покликане нормалізувати життєві показники користувача та максимізувати продуктивність його праці.

Можливості для розвитку включають як поширення додатку серед користувачів з метою отримання зворотного зв'язку з приводу ефективності роботи додатку, так і збір та аналіз цих даних для покращення роботи алгоритмів та оптимізації. Також можливо розширити функціонал мобільного застосунку, використовуючи контекст користувача, як-от режими фокусування для підвищення точності рекомендацій.

## СПИСОК ПОКЛИКАНЬ

1. Ming-Chuan C., Li-Wei K. Develop a personalized intelligent music selection system based on heart rate variability and machine learning. 2016.
2. Федорченко С. В. Загальна психологія. Модуль 3 «Емоційна, вольова та мотиваційна сфери / С. В. Федорченко. – 2017.
3. Chandrasena M., Upekha H., Wijesooriya H. Music Emotion Recognition. Department of Computer Engineering University of Peradeniya, Sri Lanka. 2023.
4. Ma Y., XinXing L., Mingxing X. et al. Multi-scale Context Based Attention for Dynamic Music Emotion Prediction. 2017.
5. Hızlısoy S. Music emotion recognition using convolutional long short term memory deep neural networks / S. Hızlısoy, S. Yildirim, Z. Tüfekci. – 2020.
6. Fumei X., Yu T. Study on Music Emotion Recognition Based on the Machine Learning Model Clustering Algorithm. 2022.
7. Jiayu L., Zhiyu H., Yumeng C. et al. Towards Ubiquitous Personalized Music Recommendation with Smart Bracelets. 2022.
8. openSMILE Documentation [Електронний ресурс]. 2022. Режим доступу до ресурсу: <https://audeering.github.io/opensmile/about.html#what-is-opensmile>.
9. DEAM Dataset - Emotional Analysis in Music [Електронний ресурс]. 2021. Режим доступу до ресурсу: <https://www.kaggle.com/datasets/imsparsh/deam-mediaeval-dataset-emotional-analysis-in-music>.
10. Generate Music with an RNN [Електронний ресурс]. 2024. Режим доступу до ресурсу: [https://www.tensorflow.org/tutorials/audio/music\\_generation](https://www.tensorflow.org/tutorials/audio/music_generation).
11. Convert TensorFlow models [Електронний ресурс]. 2024. Режим доступу до ресурсу: [https://www.tensorflow.org/lite/models/convert/convert\\_models](https://www.tensorflow.org/lite/models/convert/convert_models).

12. Mel Frequency Cepstral Coefficient (MFCC) tutorial [Электронный ресурс]. 2013. Режим доступа до ресурсу: <http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/>.
13. Neural Network. ML Glossary [Электронный ресурс]. Режим доступа до ресурсу: [https://ml-cheatsheet.readthedocs.io/en/latest/nn\\_concepts.html](https://ml-cheatsheet.readthedocs.io/en/latest/nn_concepts.html).
14. Music Enhances Productivity: The Science Behind The 10 Best Songs [Электронный ресурс] // Forbes – Режим доступа до ресурсу: <https://www.forbes.com/sites/tracybrower/2023/12/10/music-enhances-productivity-the-science-behind-the-10-best-songs/?sh=1f6c940de604>.
15. García-Martínez B. Entropy and the Emotional Brain: Overview of a Research Field / B. García-Martínez, A. Fernández-Caballero, A. Martínez-Rodrigo. – 2021.
16. DSP Background - Deep Learning for Audio Classification p.1 [Электронный ресурс] – Режим доступа до ресурсу: [https://www.youtube.com/watch?v=Z7YM-HAz-IY&list=PLhA3b2k8R3t2Ng1WW\\_7MiXeh1pfQJOi\\_P](https://www.youtube.com/watch?v=Z7YM-HAz-IY&list=PLhA3b2k8R3t2Ng1WW_7MiXeh1pfQJOi_P).
17. Heart Rate Variability (HRV) [Электронный ресурс] – Режим доступа до ресурсу: <https://my.clevelandclinic.org/health/symptoms/21773-heart-rate-variability-hrv>.
18. Zölzer U. Digital Audio Signal Processing / Udo Zölzer., 2022.
19. Understanding Bidirectional LSTM for Sequential Data Processing [Электронный ресурс] – Режим доступа до ресурсу: <https://medium.com/@anishnama20/understanding-bidirectional-lstm-for-sequential-data-processing-b83d6283befc>.
20. Audio File Formats: The Ultimate Guide [Электронный ресурс] – Режим доступа до ресурсу: <https://emastered.com/blog/audio-file-formats>.

## ДОДАТКИ

### Додаток А. Порівняння способів обчислення HRV

| Назва одиниці виміру<br>HRV                                                                | Одиниці виміру       | Недоліки                                                                                                  |
|--------------------------------------------------------------------------------------------|----------------------|-----------------------------------------------------------------------------------------------------------|
| <b>SDNN</b><br>(Standard Deviation of<br>RR Intervals)                                     | Мілісекунди, ms      | Найбільш наближене до<br>точного значення при<br>тривалих періодах<br>моніторингу (зазвичай<br>24 години) |
| <b>RMSSD</b><br>(Root Mean Square of<br>Successive Differences)                            | Мілісекунди, ms      | Більш чутливий до<br>впливів<br>парасимпатичної<br>системи, аніж<br>симпатичної                           |
| <b>CV</b><br>(Coefficient of Variation)                                                    | Відсотки, %          | Чутливий до різких та<br>короткотривалих змін<br>частоти серцевих<br>скорочень                            |
| <b>NN50</b><br>(Number of Pairs of<br>Successive RRs That<br>Differ by More Than 50<br>ms) | Кількість одиниць, n | Використання<br>відсікання в 50 мс є<br>умовним і може не<br>охоплювати всі значні<br>вегетативні зміни   |

## Додаток Б. Обрахування варіабельності ритму серцебиття

```
func processHeartRate(_ sample: HeartRateSample) {
    heartRateLog.append(sample)

    // Ensure we have more than one heart rate value to calculate intervals
    if heartRateLog.count > 1 {
        // Convert heart rate values in bpm to NN intervals in seconds
        let nnIntervals = heartRateLog.map { 60.0 / Double($0.value) }

        // Calculate the mean NN interval
        let meanNNInterval = nnIntervals.reduce(0.0, +) / Double(nnIntervals.count)

        // Calculate the squared differences from the mean NN interval
        let squaredDifferences = nnIntervals.map { ($0 - meanNNInterval) * ($0 - meanNNInterval) }

        // Calculate the variance of NN intervals
        let variance = squaredDifferences.reduce(0.0, +) / Double(nnIntervals.count - 1) // Using n-1
        for sample: Variance

        // Calculate SDNN as the square root of the variance
        let sdnn = sqrt(variance)

        let sdnnInMilliseconds = sdnn * 1000

        // Append the SDNN value to the log with the current date
        hrvSDNNLog.append(HrvSDNNSample(
            value: sdnnInMilliseconds,
            date: sample.date
        ))
    }
}
```