

Вінницький національний технічний університет  
(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії  
(повне найменування факультету)

Обчислювальної техніки  
(повна назва кафедри)

## **Пояснювальна записка**

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему: Інформаційна технологія для аудіодетекції звукових сигналів

немовленнєвого походження

Виконав: студент 2 курсу, групи 1КІ-19 м  
спеціальності:

123 — комп'ютерна інженерія

(шифр і назва напрямку підготовки)

Тютюнник Ярослав Олександрович

(прізвище та ініціали)

Керівник: Ткаченко Олександр Миколайович

(прізвище та ініціали)

2020 р.

## Індивідуальне завдання

## АНОТАЦІЯ

У даній роботі було розглянуто проблему розпізнавання звукових сигналів немовленнєвого походження. А саме, був проведений огляд основних характеристик звукових хвиль та існуючих на сьогоднішній день методів класифікації. Також у роботі наведений огляд існуючих та популярних сьогодні систем розпізнавання звуку.

Більш детально розглядається метод класифікації заснований на гаусівських змішаних моделях, який буде використовуватись для роботи програми. Було розроблено програмний продукт для розпізнавання звукових сигналів немовленнєвого походження з використанням засобів мови C++.

Проведено тестування продукту на коректність розпізнавання та наявність помилок. Розроблено інструкцію користувача, що дозволить йому отримати інформацію про основи використання сервісу та інструкцію для включення розроблених програмних засобів до інших систем розпізнавання.

Також було проаналізовано економічну доцільність досліджуваної розробки при впровадженні її у виробництво та встановлено, що система має непогані шанси на комерційний успіх.

## ANNOTATION

In this work the problem of recognition of sound signals was considered. Namely, an overview of the main characteristics of sound waves and the existing methods of classification for today. Also in the work is an overview of existing and popular sound recognition systems.

The classification method based on Gaussian mixed models, which will be used to work the program, is considered in more detail. A software product for identifying not linguistic sound signals using a C ++ language was developed.

A product testing has been conducted to verify the correctness and error. A user's manual has been developed, which will allow him to get information about the basics of using the service and instructions for including the developed software in other recognition systems.

Furthermore, analyzed the economic feasibility of the studied development in its implementation in production and found that the system has a good chance of commercial success.

## ЗМІСТ

<b>ВСТУП</b> .....	7
<b>1 ОГЛЯД ОСНОВНИХ ХАРАКТЕРИСТИК ЗВУКОВИХ ХВИЛЬ, МЕТОДІВ КЛАСИФІКАЦІЇ У СИСТЕМАХ РОЗПІЗНАВАННЯ ТА ІСНУЮЧИХ СИСТЕМ ДЕТЕКЦІЇ ЗВУКІВ</b> .....	10
1.1 Основні поняття про звук.....	10
1.2 Розпізнавання звукового сигналу .....	14
1.3 Характеристики звуку, що можуть бути проаналізовані .....	16
1.4 Алгоритми машинного навчання для класифікації звуку.....	18
1.4.1 Метод опорних векторів.....	21
1.4.2 Логістична регресія.....	23
1.4.3 Метод k-найближчих сусідів .....	24
1.4.4 GMM класифікатор.....	27
1.4.5 Наївний баєсів класифікатор .....	28
1.4.6 Дерева рішень.....	30
1.5 Огляд існуючих систем розпізнавання звуків.....	32
1.5.1 ShotSpotter.....	32
1.5.2 Google Assistant .....	34
1.5.3 Siri.....	35
1.5.4 Shazam .....	36
<b>2 МАТЕМАТИЧНЕ ПІДГРУНТЯ РОЗПІЗНАВАННЯ АУДИОСИГНАЛІВ НА ОСНОВІ ГАУСОВСЬКИХ ЗМІШАНИХ МОДЕЛЕЙ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ</b> .....	38
2.1 Огляд основних компонентів інформаційної технології .....	38
2.2 Загальна схема цифрової обробки аудіосигналу для отримання параметрів моделі.....	39

					08-23.МКР.015.00.000 ПЗ					
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Інформаційна технологія для аудіодетекції звукових сигналів немовленнєвого походження.  Пояснювальна записка			<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
Розроб.		Тютюнник Я. О.						5		
Перевір.		Ткаченко О.М.								
Реценз.		Куперштейн Л.М.								
Н. Контр.		Швець С.І.								
Затверд.		Азаров О. Д.			<b>1КІ-19м</b>					

2.3 Обчислення кепстральних коефіцієнтів .....	43
2.4 Обчислення відстані .....	44
2.5 Вибір мови програмування .....	48
2.6 Розробка сутностей .....	51
2.7 Загальний алгоритм роботи програми .....	54
2.8 Опис алгоритму прийняття рішень .....	57
<b>3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СТВОРЕНОЇ</b>	
<b>ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ТА РОЗРОБКА ПРОГРАМНОЇ</b>	
<b>ДОКУМЕНТАЦІЇ.....</b>	
3.1 Основні характеристики системи, що досліджується .....	61
3.2 Результати розпізнавання.....	62
3.3 Інструкція використання програмного комплексу .....	67
3.4 Реалізація розроблених програмних засобів у системі розпізнавання.....	70
<b>4 ЕКОНОМІЧНА ЧАСТИНА .....</b>	
<b>4.1 Оцінювання комерційного потенціалу розробки.....</b>	
<b>4.2 Прогнозування витрат на виконання науково-дослідної та</b>	
<b>конструкторсько-технологічної роботи.....</b>	
<b>4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.....</b>	
<b>4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності. ....</b>	
<b>ВИСНОВКИ .....</b>	
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....</b>	
<b>ДОДАТОК А Технічне завдання .....</b>	
<b>ДОДАТОК Б Лістинг програми .....</b>	
<b>ДОДАТОК В Довідка про друк в журналі .....</b>	
<b>ДОДАТОК Г Схема аудіодетекції звукового сигналу .....</b>	
<b>ДОДАТОК Д Змішана модель Гауса.....</b>	
<b>ДОДАТОК Е Відстані до моделей в базі.....</b>	
<b>ДОДАТОК Ж Алгоритм роботи програми .....</b>	
<b>ДОДАТОК И Успішно розпізнані звуки .....</b>	

## ВСТУП

Протягом останніх років зросла популярність систем для розпізнавання звуків, зображення та музичних композицій. При тому що їх обсяги в глобальній системі інтернету постійно збільшуються виникає потреба у систематизації накопиченої інформації. Проте ручна обробка цих даних є неможливою, враховуючи обсяги. Вирішити цю проблему можуть алгоритми машинного навчання, які на основі попередньо накопичених даних можуть класифікувати вхідні зразки.

Машинне навчання можна охарактеризувати як виділення знань з даних. Це широка галузь науки де перетинаються статистика, штучний інтелект та інформатика. Застосування методів машинного навчання в останні роки стали нормою у повсякденному житті. Автоматичні рекомендації фільмів які нам до вподоби, рекомендації щодо вибору їжі та продуктів, які можна придбати, до персоналізованого онлайн-радіо та навіть визнання друзів по спільним фотографіям. Загалом багато сучасних веб-сайтів і пристроїв мають алгоритми машинного навчання.

Якщо подивитись на такі великі сайти як Facebook, Amazon або Netflix, то дуже ймовірно, що кожна частина сайту містить кілька моделей машинного навчання. Окрім комерційних додатків, машинне навчання мало величезний вплив на те, як сьогодні проводиться наукові дослідження із великою кількістю вхідних даних. Інструменти, машинного навчання можуть бути застосовані до різноманітних наукових проблем, таких як розуміння зірок, знаходження далеких планет, розпізнавання зображень та звуків, аналізуючи послідовності ДНК і забезпечуючи персоналізовані методи лікування раку та багато іншого. Методи машинного навчання фактично можна застосувати усюди де потрібно знаходити якісь закономірності. Варто зазначити, що обсяги даних на яких навчаються алгоритми не дозволяють людині вручну обробити їх.

Однією з проблем, яка може бути вирішена за допомогою методів машинного навчання та класифікації є проблема розпізнавання звуків.

Системи розпізнавання звуків може бути використана як частина більш складного механізму. До прикладу це може бути система розпізнавання пострілів, яка буде визначати тип зброї та вказувати на точне місце походження звуку. Розпізнавати можна не тільки постріли але і скажімо ворожу техніку, авіацію, звуки надзвичайних ситуацій на підприємствах. Крім того системи розпізнавання звуків використовуються для ідентифікації музичних творів на які поширюється дія авторського права. Вони запобігають розповсюдженню піратських копій та допомагають відслідковувати процеси поширення музичних композицій. Загалом використання таких систем є дуже варіативним та обмежується тільки уявою розробників. Тому можна сміливо сказати що тема дослідження є актуальною на сьогоднішній день.

**Метою дослідження** є підвищення ймовірності правильного розпізнавання звуків немовленнєвого походження за рахунок розроблення нової інформаційної технології.

Для досягнення поставленої мети необхідно розв'язати такі **задачі**:

- провести аналіз існуючих на ринку систем розпізнавання звуку;
- дослідити існуючі моделі класифікації та визначити підходящої для розпізнавання звукового сигналу;
- дослідити сучасні інструменти які дозволили б реалізувати на практиці обрану модель класифікації;
- розробити програмне забезпечення;
- провести тестування розробленого програмного забезпечення.

**Об'єктом дослідження** є процес оброблення звукових сигналів у комп'ютерних системах.

**Предметом дослідження** є методи та програмні засоби розпізнавання звукових сигналів немовленнєвого походження.

**Методи дослідження** що використовувались у магістерській роботі є метод виділення кепстральних коефіцієнтів для визначення ознак за якими буде проводитись класифікація фреймів, метод класифікації моделей на



основі Гаусівського змішаного розподілу використовується для знаходження звуків в базі. Також, при розробці програмного продукту, використано принципи об'єктно-орієнтованого програмування для побудови моделей даних.

**Наукова новизна отриманих результатів** магістерської роботи полягає в тому, що удосконалено метод розпізнавання звуків на основі змішаних Гаусівських моделей, в якому, на відміну від існуючого, на стадії прийняття рішення, враховується інформація про тривалість аудіо сигналу, що дало можливість підвищити ймовірність правильного розпізнавання звуків немовленнєвого походження.

**Практичне значення отриманих результатів** магістерської роботи полягає в розробці інформаційної технології розпізнавання звуків немовленнєвого походження, яка містить:

— словник параметрів еталонних звукових сигналів, представлених у вигляді Гаусівських сумішей;

— алгоритми та програмне забезпечення для розпізнавання звукових сигналів, що дає можливість у режимі реального часу розпізнавати звуки наявні в словнику.

**Апробація результатів** магістерської роботи полягає в наступному. Основні результати наукових досліджень за темою роботи доповідалися та обговорювалися на щорічній конференції професорсько-викладацького складу, співробітників та студентів Вінницького національного технічного університету (Вінниця, 2019).

В ході виконання роботи створювались **публікації**. За результатами проведених досліджень було опубліковано дві роботи [1, 2] та подано статтю в Міжнародний науковий журнал «Інформаційні технології та комп'ютерна інженерія» (довідка додається).

# 1 ОГЛЯД ОСНОВНИХ ХАРАКТЕРИСТИК ЗВУКОВИХ ХВИЛЬ, МЕТОДІВ КЛАСИФІКАЦІЇ У СИСТЕМАХ РОЗПІЗНАВАННЯ ТА ІСНУЮЧИХ СИСТЕМ ДЕТЕКЦІЇ ЗВУКІВ

## 1.1 Основні поняття про звук

Звук — це особливий вид механічних коливань які поширюються в пружних середовищах і тілах, та здатний викликати слухові відчуття. Людина своїми слуховими органами може сприймати звук частота коливань яких лежать в межах від 20 Гц до 20 кГц. Коливання на частотах менше 20 Гц називають інфразвук, на частотах більше 20 кГц — ультразвук. Вивченням звуку як фізичного явища займається наука акустика.

Закономірності сприйняття звуку людиною мають нелінійний характер. Іноді ми чуємо те, чого немає насправді, навіть не віддаючи собі в цьому звіту. Психоакустика — це наука, яка займається вивченням законів сприйняття звукових явищ людиною.

Будь-які дослідження в галузі музичної теорії, а також акустики музичних інструментів, можливі тільки за умови об'єднання даних обох наук. Саме цим і займається наука музична акустика. Вивчення музичної акустики є невід'ємною частиною будь-якого курсу композиції електроакустичної музики, головним матеріалом якої є власне природа звуку і його сприйняття людиною (рис. 1.1).

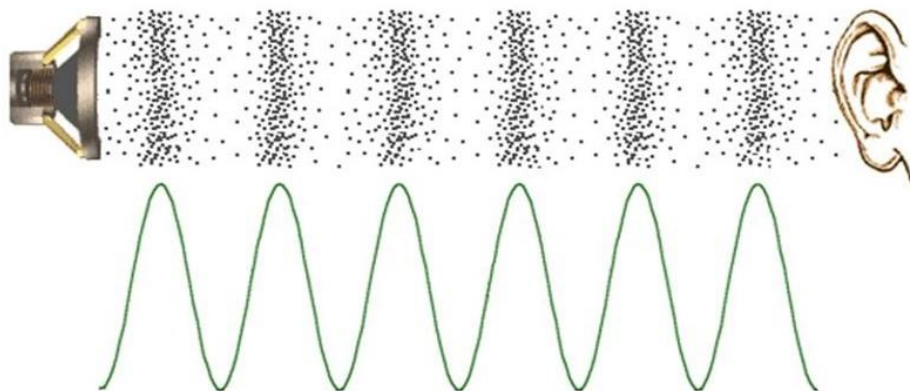


Рисунок 1.1 — Поширення звукових хвиль в просторі

Різні музичні інструменти (коливання мембран барабану, струни від гітари, рояль) здійснюють механічні коливання в повітряному середовищі. Вони створюють області підвищеного та зниженого тиску, що призводить до того створення та поширення в просторі звукових хвиль. Цей процес зображено на рисунку 1.1. Вимірюючи за допомогою приладів (наприклад, мікрофона) тимчасову залежність зміни тиску в певній точці простору, можна побудувати графік, який ілюструє коливальний процес — так зване хвильове уявлення звуку. Тим же способом здійснюється запис звуку на різні носії: магнітні, оптичні, цифрові і т.д.

Гучність звуку визначається амплітудою звукових коливань. Під поняттям висота звуку розуміють частоту цих коливань; тембр звуку, визначається амплітудою коливань обертонів, хоча, строго кажучи, це поняття не має однозначного визначення.

Фазою коливання називають стадію або стан руху тіла, що коливається щодо будь-якого його положення, наприклад, щодо положення рівноваги; це положення можна прийняти за початок відліку фази. Будь-яке інше положення тіла при його коливаннях матиме певну стадію руху або фазу щодо обраного початку відліку[6].

За умов нормальної температури швидкість поширення звукової хвилі (швидкість звуку) складає 340 м / сек. Довжина звукової хвилі на різних частотах також є різною. До прикладу, на частоті 340 Гц довжина хвилі складає 1 м., на частотах 1700 Гц та 10200 Гц довжина хвилі відповідно складає 20 см та 3,3 см.

Інтерференція хвиль представляє собою явище накладання в просторі двох або більше хвиль, при якому в різних точках виходить посилення або ослаблення амплітуди результуючої хвилі. Коли ми чуємо звуки різних, але досить близьких частот відразу від двох джерел, до нас приходять то гребені обох звукових хвиль, то гребінь однієї хвилі і западина інший. В результаті накладання двох хвиль звук то посилюється, у зоні накладання гребенів, то слабшає, у разі накладання гребеня та западини[7].

При розповсюдженні, енергія звукової хвилі поглинається середовищем. Цей ефект отримав назву поглинанням звукових хвиль. Важливо відзначити, що ступінь поглинання звукової енергії залежить як від властивостей середовища (температура, тиск, щільність), так і від частоти звукових коливань: чим вище частота, тим більше розсіювання зазнає на своєму шляху звукова хвиля.

Французький математик Фур'є та його послідовники довели, що будь-яке складне коливання можна представити у вигляді суми найпростіших коливань. Іншими словами, будь-яку періодичну функцію, в разі її відповідності деяким математичним умовам, можна розкласти в ряд (суму) косинусів і синусів. Таке перетворення звукових хвиль до найпростіших коливання називається розкладом в ряд Фур'є.

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} A_k \cos\left(k \frac{2\pi}{\tau} x + \theta_k\right),$$

де  $A_k$  — амплітуда  $k$ -того гармонічного коливання;

$k \frac{2\pi}{\tau} = k\omega$  — кругова частота гармонічного коливання;

$\theta_k$  — початкова фаза  $k$ -того коливання.

Обертон називається частота вище першої, що є найнижчою. Її ще називають основний тон. Ті обертони, частоти яких відносяться від частоти основного тону, називаються гармоніками. Причому перша гармоніка вважається основним тоном.

Якщо звук містить в своєму спектрі тільки гармоніки, то їх сума є періодичним процесом і звук дає чітке відчуття висоти. При цьому суб'єктивно відчувається висота звуку відповідає найменшого спільного кратного частот гармонік.

Сукупність обертонів, що становлять складний звук, називають спектром цього звуку. Розкладання складного звуку на найпростіші складові називають спектральним аналізом. Він здійснюється за допомогою

математичного перетворення видатного французького вченого фізика-математика Фур'є.

Методи спектрального аналізу можуть застосовуватись не тільки до періодичних сигналів, але також і до сигналів, в цифровій формі. Залежно від типу сигналу використовуються різні види спектрального аналізу: ряд Фур'є застосовують для періодичних сигналів, інтеграл Фур'є — для неперіодичних сигналів, дискретне перетворення Фур'є та швидке перетворення Фур'є для цифрових сигналів (рис.1.2).

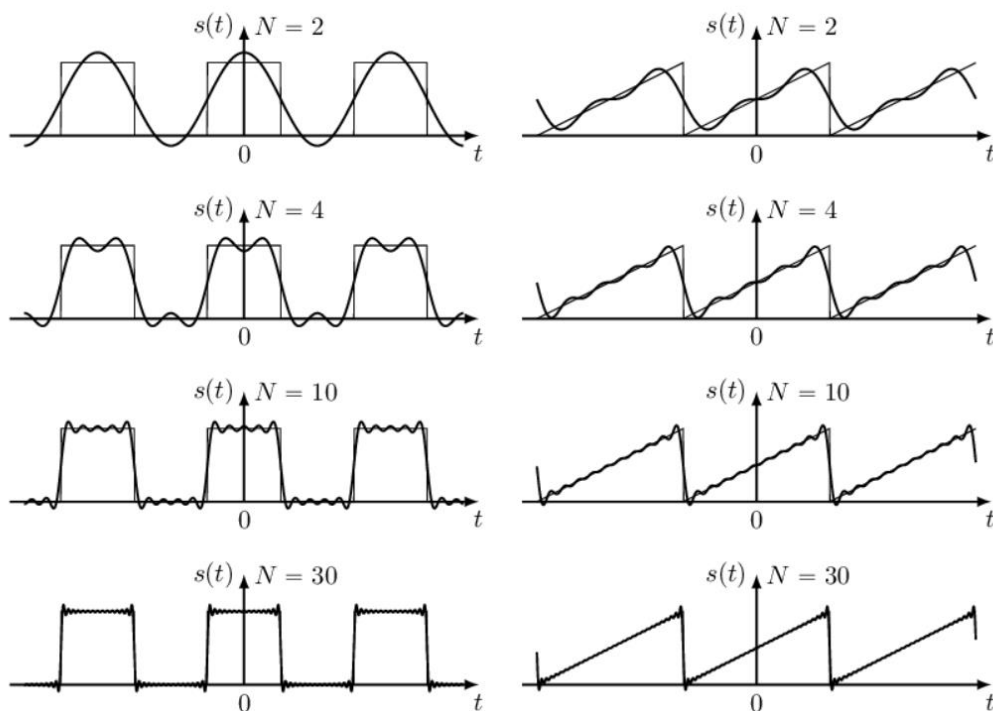


Рисунок 1.2 — Застосування ряду Фур'є для представлення сигналів різної форми

Як видно з рисунку 1.2, будь-який складний періодичний звуковий сигнал може бути представлений у вигляді суми простих гармонійних сигналів з відповідними амплітудами і фазами. Сукупність усіх амплітуд на шкалі частот називається амплітудним спектром, сукупність всіх фаз — фазовим спектром. При цьому, незважаючи на те, що ряд Фур'є може бути нескінченним, запропонована ним форма запису є дуже зручною при проведенні аналізу і обробки.

Ряд Фур'є є дуже зручним для запису періодичних функцій. Однак і на практиці, і в теорії далеко не всі функції періодичні. Щоб отримати можливість розкласти неперіодичну функцію  $f(x)$  в ряд Фур'є можна трохи схитрити. Як правило, при розгляді деякої складної неперіодичної функції нас не цікавлять її значення на всій області визначення; нам досить розглядати функцію лише на певному кінцевому інтервалі  $[x_1, x_2]$  для деяких  $x_1$  і  $x_2$ . У цьому випадку функцію можна розглядати як періодичну, з періодом  $T = x_2 - x_1$ . Для її розкладання в ряд Фур'є на інтервалі  $[x_1, x_2]$  ми можемо штучно уявити  $f(x)$  у вигляді деякої періодичної функції  $f'(x)$ , отриманої шляхом "заиклення" значень функції  $f(x)$  з розглянутого інтервалу. Неперіодична функція  $f(x)$ , після проведених маніпуляцій перетворюється в періодичну  $f'(x)$ , яка може бути розкладена в ряд Фур'є.

Важливою характеристикою звуку є його спектр. В результаті розкладання звуку на прості гармонійні колива ми отримуємо спектр. Він буває суцільний, коли енергія звукових хвиль безперервно рівномірно розподілена в широкій області частот, і лінійчатий — коли енергія звуку зосереджена на окремих частотах, які добре проглядаються. Суцільний спектр сприймається як шум, наприклад шелест дерев під вітром, звуки працюючих механізмів. Лінійчатий спектр характерний, наприклад для музичних інструментів.

## 1.2 Розпізнавання звукового сигналу

Загальну схему розпізнавання звукового сигналу представлено на рисунку 1.3. На першому етапі розпізнавання звукового сигналу відбувається його параметризація, тобто виділення ознак із певного відрізка звукового файлу. Ці ознаки беруться не із усього звуку загалом а із маленьких проміжків на який розбитий звук — фреймів. Довжина кожного фрейму складає 10 мс. Проміжок у даному часовому інтервалі є хорошим вибором, так як форма сигналу не різко змінюється і ми можемо виділити з нього ознаки.

Після того як із фрейму будуть виділені ознаки, ми можемо порівняти їх із заздалегідь підготовленими шаблонними моделями у базі. Відбувається вимірювання відстані до кожної із наявних моделей.

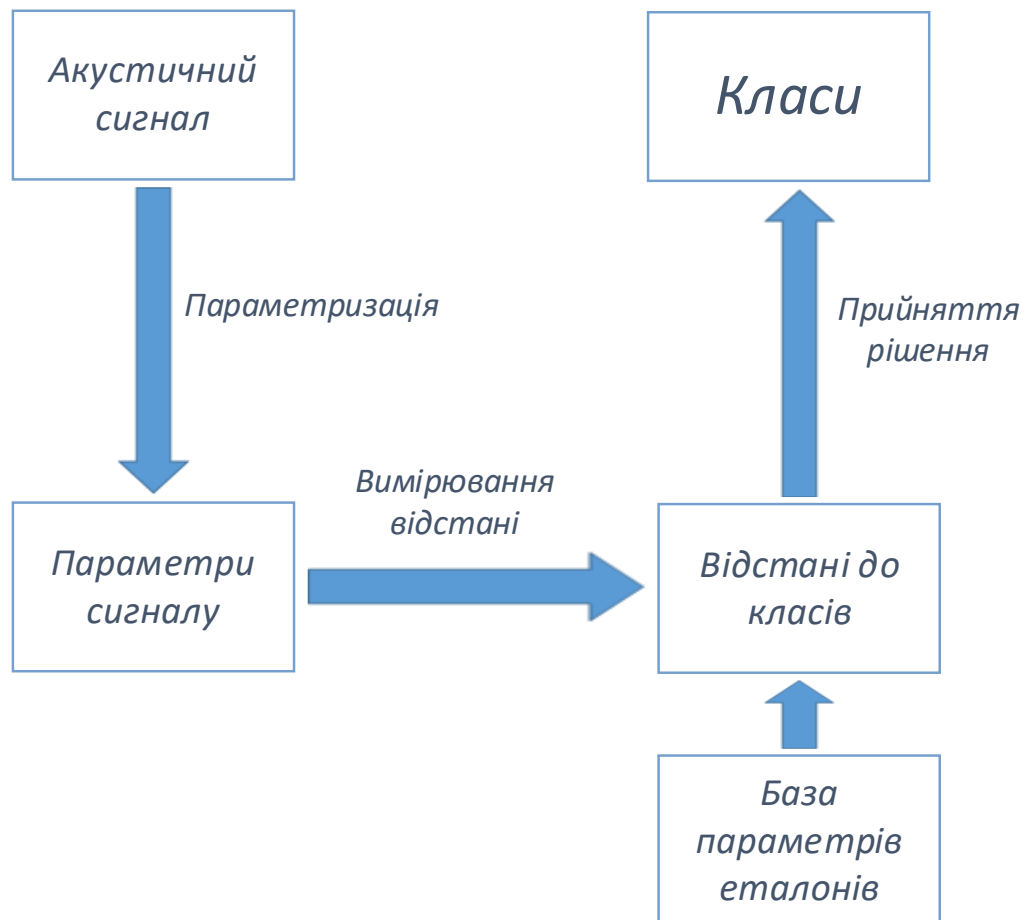


Рисунок 1.3 — Схема розпізнавання звукового сигналу

Далі нам потрібно прийняти рішення про віднесення фрейму до певного класу. Це рішення приймається на основі розробленого алгоритму який буде описаний у розділі 2. У випадку коли відстань є доволі близькою до шаблонної ми можемо віднести даний фрейм до певного класу. Якщо ж відстані є великими до усіх наявних в базі шаблонів, цей фрейм вважається нерозпізнаним. Також, варто зазначити, що для кожного класу ми можемо задавати обмеження на мінімальну та максимальну послідовність фреймів. Це дозволяє нам позбутись прийняття неправильних рішень щодо віднесення окремих фреймів до тих класів, про які ми можемо точно сказати, що вони тут не можуть бути.

### 1.3 Характеристики звуку, що можуть бути проаналізовані

На сьогоднішній день відома велика кількість характеристик сигналів звукового походження, що можуть бути використані для розпізнавання звуків та порівняння з іншими сигналами [3]. Задачі, для розв'язання яких ці характеристики можуть використовуватись, можуть бути різними. Це, наприклад, задачі розпізнавання мови, класифікації звуків, задачі виявлення окремих небезпечних звуків (пострілів, криків та ін.), задачі генерації подібного сигналу, тощо.

Основні поняття, які характеризують параметри звукових коливань можна розділити на чотири групи об'єктивних ознак, що дозволяють розрізняти звукові зразки: спектрально-часові, кепстральні, амплітудно-частотні та ознаки нелінійної динаміки. Детальніше, кожна група ознак:

#### а) спектрально-часові ознаки;

- нормалізовані середні значення спектра;
- середнє значення спектра аналізованого сигналу;
- відносний час перебування сигналу в смугах спектра;
- медіанне значення спектра сигналу в смугах;
- нормалізований час перебування сигналу в смугах спектра;
- відносна потужність спектра сигналу в смугах;
- будь-які інші ознаки сигналу, отримані безпосередньо з

спектру;

- тривалість сегменту, звуку;
- висота сегменту;

#### б) кепстральні ознаки;

- мел-частотні кепстральні коефіцієнти;
- коефіцієнти лінійного передбачення з корекцією на нерівномірність чутливості людського вуха;
- коефіцієнти потужності частоти реєстрації;
- коефіцієнти спектра лінійного передбачення;
- коефіцієнти кепстра лінійного передбачення;



в) амплітудно-частотні ознаки;

- інтенсивність, амплітуда;
- енергія;
- частота основного тону (чот);
- формантні частоти;
- джиттер (jitter), тремтіння частотна модуляція основного тону;
- шиммер (shimmer), амплітудна модуляція на основному тоні;
- радіальна базисна ядерна функція;
- нелінійний оператор Тигер;

г) ознаки нелінійної динаміки;

- відображення Пуанкаре;
- рекурентний графік;
- максимальний характеристичний показник Ляпунова;
- фазовий портрет (атрактор).

Спектрально-часові ознаки описують фізичні та математичні показники сигналу. У них можна побачити періодичність тональних ділянок звукової хвилі а також шумові ділянки, які є неперіодичними.

Амплітудно-частотні ознаки дозволяють отримувати оцінки, значення яких можуть змінюватися в залежності від параметрів дискретного перетворення Фур'є, а також при незначних зрушеннях вікна по вибірці.

Завдання нелінійної динаміки полягає в знаходженні і докладному дослідженні базових математичних моделей і реальних систем, які виходять з найбільш типових пропозицій про властивості окремих елементів, що складають систему, і законах взаємодії між ними. В даний час методи нелінійної динаміки базуються на фундаментальній математичній теорії, в основі якої лежить теорема Такенса

Як ознаки які будуть характеризувати кожен фрейм було вирішено використати спектральні коефіцієнти. Оскільки наш сигнал має розбиватись на фрейми, з відносно сталою формою сигналу, це рішення є найбільш вдалим. Інші ж групи ознак такі як спектрально-часові та амплітудно-

частотні ознаки будуть мало ефективними та не дозволять використати їх як ознаки для класифікації.

#### 1.4 Алгоритми машинного навчання для класифікації звуку

Немає сумнівів, що штучний інтелект і машинне навчання протягом останніх кількох років знайшли широку популярність. Як технології Big Data утримують статус найбільш обговорюваного ІТ-тренда сучасності, так і алгоритми машинного навчання можна вважати дуже потужним інструментом. Один з найбільш глобальних прикладів використання машинного навчання — алгоритми Netflix, які пропонують подивитися кінострічки на підставі тих, що користувач бачив раніше. Сюди ж можна віднести і систему Amazon, що рекомендує придбати книгу, відштовхуючись від куплених раніше електронних або друкованих видань.

Машинне навчання (Machine learning, ML) — збірник методів в області штучного інтелекту, набір алгоритмів, які застосовують, щоб створити систему, яка буде мати змогу вчитися на власному досвіді[4, 8 – 10]. В якості навчання система має обробляти великі масиви вхідних даних і знаходить у них закономірності. Це сукупність алгоритмів, що дозволяють комп'ютеру робити висновки на підставі даних, не слідуючи жорстко заданим правилами. Тобто машина знаходить закономірність у складних і багато-параметричних завданнях (які мозок людини не здатен вирішити), таким чином знаходячи більш точні відповіді. Як результат — вірне прогнозування.

Для якісної роботи машинного навчання ми маємо забезпечити чітке функціонування трьох його складових:

- дані;
- ознаки;
- алгоритм.

Під терміном дані розуміється базова інформація, за допомогою якої машина буде вчитись та покращувати свої характеристики. Сюди входять будь-які вибірки даних, роботі з якими потрібно навчити систему.

До ознак відносяться ті характеристики, від яких залежить вихідний результат. Правильний вибір характеристик безпосередньо впливає на результат, який ви отримаєте. Саме відбір займає багато часу і саме на ньому варто акцентувати увагу. Від правильного вибору ознак залежить те наскільки в кінцевому результаті буде ефективною система. Тут головне – не обмежувати набір характеристик, щоб не спотворити машинне сприйняття.

Алгоритм відповідає за вибір оптимального методу для вирішення поставленого завдання. Метод передбачає покрокове виконання операцій для вирішення, яке в підсумку має дати бажаний результат. Цей вибір має бути продуманим. Із великого числа алгоритмів потрібно обрати саме той, який буде найкращим для розв'язання поставленої нам задачі.

Класифікація — один з розділів машинного навчання, присвячений розв'язанню наступної задачі: в наявності множина об'єктів (ситуацій), розділених деяким чином на класи[5]. Задана певна обмежена множина об'єктів, для яких відомо, до якого класу вони відносяться. Ця множина називається навчальною вибіркою. Класова приналежність інших об'єктів не відома. Вимагається побудувати алгоритм, здатний класифікувати будь-який об'єкт початкової множини.

Задача класифікації відноситься до розділу навчання з учителем. Мають бути зразки кожного класу — об'єкти, про які заздалегідь відомо до якого класу вони належать. Такі завдання називають навчанням з учителем, а відомі дані називаються навчальною вибіркою. В машинному навчанні існує також навчання без вчителя, якщо класи з самого початку не задані, то перед нами завдання кластеризації. У цьому випадку потрібно щоб алгоритм самостійно виявив схожості між об'єктами класифікації і розділив їх класи. Проте ці задачі в деяких випадках досить схожі між собою. Класифікацію розрізняють в залежності від особливостей даних або класів, з якими вона працює.

В залежності від типу даних, на надходять на вхід класифікатора розрізняють класифікацію:

- по ознакам;
- за матрицями відстані між об'єктами;
- часовий ряд, що представляє собою послідовність вимірів у часі;
- зображення або відеоряд.

Класифікація по ознакам, коли кожен об'єкт має сукупність має набір своїх характеристик, що називаються ознаками. Ознаки можуть бути числовими або нечисловими. Ця ознака є найпоширенішою.

Класифікація за матрицями відстані між об'єктами. Кожен об'єкт описується відстанями до інших об'єктів навчальної вибірки. З цим типом вхідних даних працюють небагато методів, такі як: метод найближчих сусідів, метод потенційних функцій.

Часовий ряд чи сигнал представляє собою послідовність вимірів у часі. Кожне вимірювання може бути представленим цифрою, вектором, а в загальному випадку признаковим описом досліджуваного об'єкта в даний момент часу.

Зустрічаються і більш складні випадки, коли вхідні дані представляються у вигляді іншої інформації з більш складною структурою — в вигляді графів, текстів, тощо. Як правило, такі випадки зводяться до першого або другого випадку за допомогою різних методів попередньої обробки даних та виявлення ознак.

Класифікації за класами розділяють на класифікацію двох класів та за багатьма класами. Перший випадок є найпростішим і у технічному плані реалізації, який служить основою для вирішення складних завдань. Другий, коли кількість є більшою за 2 і може досягати великого числа. Багато класів може використовуватись, наприклад, для розшифровки невідомих символів мов. Тоді задача класифікації стає значно складнішою.

Проте існує інший варіант визначення класифікації за класами:

- класифікація з класами, що не перетинаються;
- класифікація з класами, що перетинаються, коли об'єкт може відноситися одночасно до декількох класів;

— класифікація з нечіткими класами, коли вимагається визначити ступінь приналежності об'єкта кожному з класів, зазвичай це дійсне число від 0 до 1.

Задача класифікації звуку є класифікацією по ознакам (глобально кажучи, навіть класифікацію зображень коректніше відносити до класифікації по ознакам, адже значення кожного пікселя можливо розглядати як ознаки), багато класовою (у більшості випадків) та класифікацію з неперетинаючимися класами (коректніше було б працювати з задачею перетинаючихся класів, але вона для даного випадку є значно більш складною, і що важливіше, вимагає відповідних наборів даних)[16 – 21]. Для розпізнавання та класифікації звуку може бути використана більшість існуючих алгоритмів машинного навчання, які в змозі розв'язувати задачу класифікації.

Нижче наведений список основних моделей класифікації.

#### 1.4.1 Метод опорних векторів

Метод опорних векторів представляє із себе різновид класифікатора “із широким зазором”: він належить до методів машинного навчання, які засновані на моделі векторного простору. Їх мета — знайти поділяючі поверхні між класами, максимально віддалені від точок навчальної множини.

Завдання класифікації полягає у визначенні як мінімум з двох класів до якого класу, які є заздалегідь відомі належить цей об'єкт. Зазвичай таким об'єктом є вектор в  $n$ -вимірному матеріальному просторі. Координати вектора описують окремі атрибути об'єкта. Наприклад, колір  $c$ , заданий в моделі RGB, є вектором в тривимірному просторі:  $c = (\text{red}, \text{green}, \text{blue})$ .

Якщо класів всього два («спам / не спам», «давати кредит / не давати кредит», «червоне / чорне»), то завдання відноситься до бінарної класифікації. Якщо класів кілька («червоне / жовте / синє / зелене», «стоїть/ лежить / сидить», «тверда/ рідка / газоподібна») — багатокласова (мультикласова) класифікація.

Метод опорних векторів, що розглядається, відноситься до навчання з учителем. Даний метод відноситься до бінарним класифікаторів, хоча існують способи змусити його працювати і для задач мультикласифікації.

Ідею методу зручно проілюструвати на наступному простому прикладі: дані точки на площині, розбиті на два класи (рис. 1.4). Проведемо лінію, що розділяє ці два класи (червона лінія на рис. 1.4). Усі нові точки (які не входять до навчальної вибірки) автоматично класифікуються наступним чином. Точка вище прямої потрапляє в клас А, точка нижче прямої потрапляє в клас В.

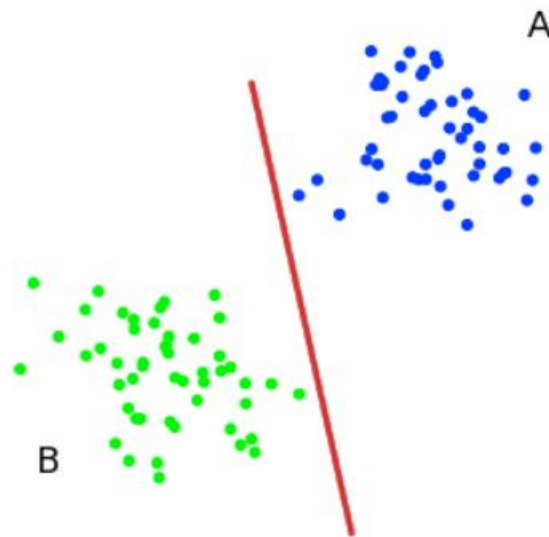


Рисунок 1.4 — Розбиття на класи

Така пряма називається роздільною прямою. Однак, в просторах високих розмірностей пряма вже не буде розділяти наші класи, так як поняття «нижче прямої» або «вище прямої» втрачає будь-який сенс. Тому замість прямих необхідно розглядати гіперплощини — простору, розмірність яких на одиницю менше, ніж розмірність початкового простору. В трьохвимірній площині гіперплощина — це звичайна двовимірна площина. Але існує кілька прямих, які поділяють два класи. Це зображено на рисю 1.5.

З точки зору точності класифікації найкраще вибрати пряму, відстань від якої до кожного класу є максимальною. Іншими словами, обирається та пряма, яка розділяє класи найкращим чином (червона пряма на рис. 1.5). Така

пряма, а в загальному випадку — гіперплощина, називається оптимальною розділяючою гіперплощиною.

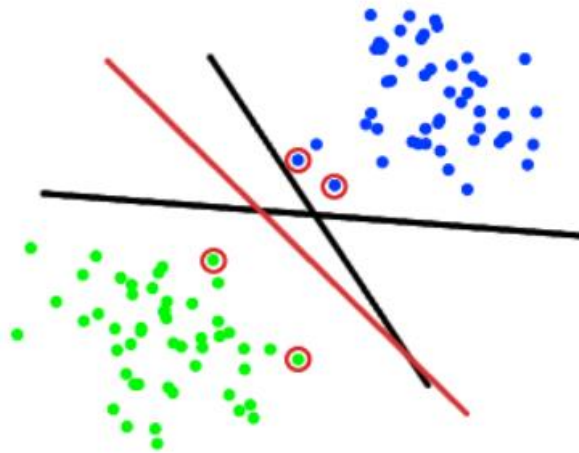


Рисунок 1.5 — Прямі, що розділяють класи

На практиці випадки, коли дані можна розділити гіперплощиною, або, як ще кажуть, лінійно, досить рідкісні. Тому дана модель не дуже добре підходить для розв'язання нашої задачі.

#### 1.4.2 Логістична регресія

Логістична регресія є одним з статистичних методів класифікації з використанням лінійного дискримінанту Фішера. Також вона входить в топ часто використовуваних алгоритмів в науці про дані

На відміну від звичайної регресії, в методі логістичної регресії не проводиться передбачення значення числової змінної виходячи з вибірки вихідних значень. Замість цього, значенням функції є ймовірність того, що дане початкове значення належить до певного класу. Для простоти, давайте припустимо, що у нас є тільки два класи і ймовірність, яку ми будемо визначати,  $P_+$  ймовірності того, що деяке значення належить класу "+". І звичайно  $P_- = 1 - P_+$ . Таким чином, результат логістичної регресії завжди знаходиться в інтервалі  $[0, 1]$ .

Основна ідея логістичної регресії є в тому, що простір вихідних значень має бути розділений на дві відповідають класам області лінійною

площиною. Ситуація дещо схожа з методом опорних векторів, який ми також розглядаємо. Отже, що ж мається на увазі під лінійною кордоном? У разі двох вимірів — це просто пряма лінія без вигинів. У разі трьох — площину, і так далі. Ця межа задається в залежності від наявних вихідних даних і навчального алгоритму. Щоб все працювало, вхідні точки даних повинні бути розподілені так щоб вони могли бути розділені лінійною площиною. Данна площина називається лінійним дискримінантом, так як вона є лінійною з точки зору своєї функції, і дозволяє моделі проводити розподіл, дискримінацію точок на різні класи.

Якщо неможливо провести лінійне поділ точок у вихідному просторі, варто спробувати перетворити вектори ознак в простір з великою кількістю вимірювань, додавши додаткові ефекти взаємодії, члени більш високого ступеня і т.д. Використання лінійного алгоритму в такому просторі дає певні переваги для навчання нелінійної функції, оскільки межа стає нелінійною при поверненні в початкове простір.

#### 1.4.3 Метод k-найближчих сусідів

Людина, стикаючись з новим завданням, використовує свій життєвий досвід, згадує аналогічні ситуації, які колись з ним відбувалися. Про властивості нового об'єкта ми судимо, покладаючись на схожі знайомі спостереження. Наприклад, зустрівши іноземця на вулиці, ми можемо здогадатися про його походження з мови, жестів і зовнішності. Для цього необхідно згадати особу найбільш схожою на нього, походження якого відомо.

Так, подібно до наведеного вище прикладу, схожість об'єктів лежить в основі алгоритму k-найближчих сусідів (k-nearest neighbor algorithm, KNN). Алгоритм здатний виділити серед всіх спостережень k відомих об'єктів (k-найближчих сусідів), схожих на новий невідомий раніше об'єкт. На основі класів найближчих сусідів виноситься рішення щодо нового об'єкта, який проходить процедуру класифікації. Дуже важливим завданням даного



алгоритму є підбір коефіцієнта  $k$  — кількість записів, які будуть вважатися близькими.

Нехай є  $m$  спостережень, кожному з яких відповідає запис в таблиці. Всі записи належать якомусь класу. Необхідно визначити клас для нового запису.

На першому кроці алгоритму слід задати число  $k$  — кількість найближчих сусідів. Якщо прийняти  $k = 1$ , то алгоритм втратить узагальнюючу здатність (тобто здатність видавати правильний результат для даних, що не зустрічалися раніше в алгоритмі) так як нового запису буде присвоєно клас близькому до неї. Якщо встановити занадто велике значення, то багато локальні особливості будуть виявлені.

На другому кроці відбувається пошук сусідів. Знаходяться  $k$  записів з мінімальним відстанню до вектору ознак нового об'єкта. Де  $x, y, z$  — вектори ознак порівнюваних об'єктів. Функція для розрахунку відстані повинна відповідати наступним правилам:

—  $d(x, y) \geq 0$ ,  $d(x, y) = 0$  тоді і тільки тоді, коли  $x = y$ ;

—  $d(x, y) = d(y, x)$ ;

—  $d(x, z) \leq d(x, y) + d(y, z)$ , за умови, що точки  $x, y, z$  чи не лежать на одній прямій.

Для впорядкованих значень атрибутів знаходиться Евклідова відстань:

$$D_E = \sqrt{\sum_i^n (x_i - y_i)^2},$$

де  $n$  — кількість атрибутів.

Для строкових змінних, які не можуть бути впорядковані, може бути застосована функція відмінності, яка задається наступним чином:

$$dd(x, y) = \begin{cases} 0, & x = y \\ 1, & x \neq y \end{cases}$$

Часто перед розрахунком відстані необхідна нормалізація. Нормалізація за допомогою стандартного відхилення виглядає так:

$$X^x = \frac{X - X_{\text{ср}}}{\sigma_x},$$

де  $\sigma_x$  — стандартне відхилення;

$X_{\text{ср}}$  — середнє значення.

При знаходженні відстані іноді враховують значимість атрибутів. Значимість визначається експертом або аналітиком суб'єктивно, покладаючись на власний досвід. У такому випадку при знаходженні відстані кожен  $i$ -тий квадрат різниці в сумі множиться на коефіцієнт  $Z_i$ . Наприклад, якщо атрибут А в три рази важливіше атрибута В ( $Z_A = 3$ ,  $Z_B = 1$ ), то відстань буде перебувати в такий спосіб:

$$D_E = \sqrt{3(x_A - y_A)^2 + (x_B - y_B)^2}.$$

Подібний прийом називають розтягуванням осей (stretching the axes), що дозволяє знизити помилку класифікації.

Слід зазначити, що якщо для запису В найближчому сусідом є А, то це не означає, що В — найближчий сусід А. Дану ситуацію представлено на рис. 1.6. При  $k = 1$  найближчий для точки В буде точка А, а для А — Х. при збільшенні коефіцієнта до  $k = 7$ , точка В так само не буде входити в число сусідів.

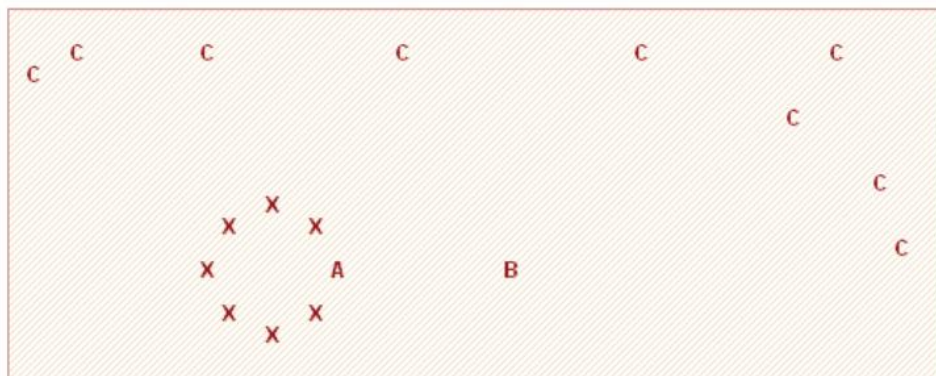


Рисунок 1.6 — Найближчі сусіди А і В

На наступному кроці, коли знайдені записи, найбільш схожі на нову, необхідно вирішити, як вони впливають на клас нового запису. Для цього використовується функція поєднання (combination function). Одним з основних варіантів такої функції є просте незважене голосування (simple unweighted voting).

Спочатку, задавши число  $k$ , ми визначили, скільки записів буде мати право голосу при визначенні класу. Потім ми виявили ці записи, відстань від яких до нової виявилася мінімальним. Тепер можна приступити до простого невиваженого голосування. Відстань від кожного запису при голосуванні тут більше не грає ролі. Всі мають рівні права у визначенні класу. Кожна вільна позиція голосує за клас, до якого належить. Нового запису присвоюється клас, який набрав найбільшу кількість голосів.

До позитивних особливостей алгоритму можна віднести наступне. Він є стійкий до аномальних викидів, тому що ймовірність попадання такого запису в число  $k$ -найближчих сусідів мала. Програмна реалізація алгоритму відносно проста. Результат роботи алгоритму легко піддається інтерпретації. Експертам в різних областях цілком зрозуміла логіка роботи алгоритму, заснована на знаходженні схожих об'єктів.

Алгоритму KNN притаманні і ряд недоліків. По-перше, набір даних, який використовується для алгоритму, повинен бути репрезентативним. По-друге, модель не можна "відокремити" від даних: для класифікації нового прикладу потрібно використовувати всі приклади. Ця особливість сильно обмежує використання алгоритму

#### 1.4.4 GMM класифікатор

Модель Гаусової суміші (англ. Gaussian Mixture Models) є імовірнісною моделлю, яка базується на припущенні, що всі спостереження, що належать до певних класів, утворюються з суміші скінченної кількості гаусових розподілів з невідомими параметрами, тобто суми нормальних розподілів [15, 22, 23]. Модель сумішей можна розглядати як узагальнення кластеризації

k-means з включенням інформації про коваріаційну структуру даних і центри розподілів.

В цілому GMM модель цікава тим, що розглядає дані як результат лінійної комбінації певної кількості гаусових розподілів, в той час, як більшість інших методів статистичної класифікації використовують чи лише один певний розподіл, чи взагалі розглядають лише взаємовідносини між даними.

З урахуванням того, що саме нормальний розподіл є дуже розповсюдженим у реальному світі, вибір саме цього розподілу є досить очевидним при побудові моделі. До того ж, як витікає з центральної граничної теореми, у випадку, коли робота виконується з достатньо складними даними та кількість невідомих факторів достатньо велика, використання суміші нормальних розподілів є досить дескриптивним, тобто таким, що в багатьох випадках надійно описує дані.

Для знаходження центрів розподілів використовується EM-алгоритм (Expectation-maximization алгоритм). Цей алгоритм є ітераційним та використовується для оцінки максимальної правдоподібності при обчисленні параметрів статистичної моделі з схованими змінними. При використанні GMM моделі для класифікації часто використовується припущення, що спостереження різних класів належать до різних розподілів. Після отримання характеристик розподілів за допомогою EM-алгоритму, для кожного спостереження можливо провести класифікацію, класифікувавши його до того класу, в розподілі якого в нього найбільша ймовірність.

#### 1.4.5 Наївний баєсівський класифікатор

Наївний байєсівський алгоритм — це алгоритм класифікації, заснований на теоремі Байєса з припущенням незалежності ознак. Іншими словами, НБА передбачає, що наявність якогось-небудь ознаки в класі НЕ пов'язано з наявністю будь-якого іншого ознаки. Наприклад, фрукт може вважатися яблуком, якщо він червоний, круглий і його діаметр становить близько 8

сантиметрів. Навіть якщо ці ознаки залежать одна від одної або від інших ознак, в будь-якому випадку вони вносять незалежний внесок у ймовірність того, що цей фрукт є яблуком. У зв'язку з таким допущення алгоритм називається «наївним».

Моделі на основі НБА досить прості і вкрай корисні при роботі з дуже великими наборами даних. При своїй простоті НБА здатний перевершити навіть деякі складні алгоритми класифікації.

Теорема Байєса дозволяє розрахувати апостеріорну ймовірність  $P(c | x)$  на основі  $P(B_i)$ ,  $P(A)$  і  $P(A | B_i)$ .

$$P(B_i / A) = \frac{P(B_i) * P(A / B_i)}{P(A)},$$

де  $P(B_i / A)$  — апостеріорна ймовірність даного класу  $c$  (тобто даного значення цільової змінної) при даному значенні ознаки  $x$ ;

$P(B_i)$  — апіорна ймовірність даного класу;

$P(A / B_i)$  — правдоподібність, тобто ймовірність даного значення ознаки при даному класі;

$P(A)$  — апіорна ймовірність даного значення ознаки.

До позитивних сторін даної моделі можна віднести наступне. Класифікація, в тому числі багатокласова, виконується легко і швидко. Коли допущення про незалежність виконується, НБА перевершує інші алгоритми, такі як логістична регресія (logistic regression), і при цьому вимагає менший обсяг навчальних даних. НБА краще працює з категорійними ознаками, ніж з безперервними. Для безперервних ознак передбачається нормальний розподіл, що є досить сильним допущенням.

Але є і негативні сторони. Якщо в тестовому наборі даних є певне значення категорійного ознаки, яке не зустрічалось в навчальному наборі даних, тоді модель присвоїть нульову ймовірність цього значення і не зможе зробити прогноз. Це явище відоме під назвою «нульова частота» (zero frequency). Дану проблему можна вирішити за допомогою згладжування.

Одним з найпростіших методів є згладжування Лапласа (Laplace smoothing). Хоча НБА є хорошим класифікатором, значення прогнозованих ймовірностей не завжди є достатньо точними. Тому не слід занадто покладатися на результати, повернуті методом `predict_proba`. Ще одним обмеженням НБА є припущення про незалежність ознак. В реальності набори повністю незалежних ознак зустрічаються вкрай рідко.

Абстрагуючись від теорії і переходячи до практики, можна виділити наступні сфери застосування теорема баєса:

- «Виллов» спаму в електронній пошті;
- сегментація новинних статей по їх тематиці;
- визначення емоційного забарвлення блоку тексту;
- програмне забезпечення для розпізнавання осіб.

#### 1.4.6 Древа рішень

Дерево прийняття рішень — засіб підтримки прийняття рішень, яке використовує деревовидний граф або модель прийняття рішень, а також можливі наслідки їх роботи, включаючи ймовірність настання події, витрати ресурсів і корисність. На рис. 1.7 подано графічне представлення структури дерева.

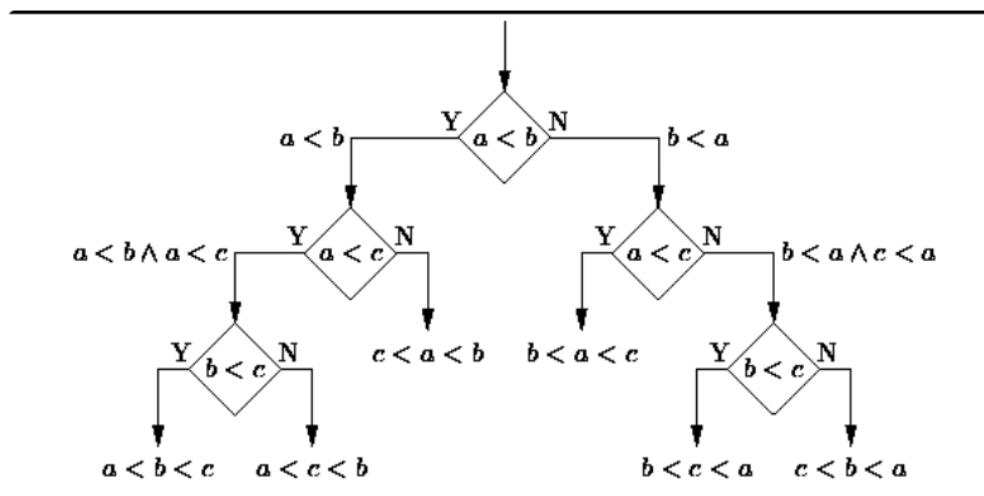


Рисунок 1.7 — Дерево прийняття рішень

З точки зору вирішення проблеми, дерево класифікації є мінімальною кількістю питань «так / ні», відповівши на які, можна зробити вірний вибір. Якщо розглядати дерево як метод, то воно дозволяє підійти до вирішення проблеми зі структурованою і систематичною боку, щоб в результаті прийти до логічного висновку.

Дійсно, побудова дерев рішень дозволяє наочно продемонструвати іншим і розібратися самому в структурі даних, створити працюючу модель класифікації даних, якими б «великими» вони не були.

Не зважаючи на проблеми, що виникають при побудові дерев, було б несправедливо не згадати про їх переваги. Це швидкий процес навчання, генерація правил в областях, де експерту важко формалізувати свої знання, витяг правил на природній мові, інтуїтивно зрозуміла класифікаційна модель, а також висока точність прогнозу, порівнянна з іншими методами (статистика, нейронні мережі).

В силу цих та багатьох інших причин, методологія дерев рішень є важливим інструментом в роботі кожного фахівця, що займається аналізом даних, незалежно від того практик він або теоретик.

Дерева рішень є прекрасним інструментом в системах підтримки прийняття рішень, інтелектуального аналізу даних (data mining). До складу багатьох пакетів, призначених для інтелектуального аналізу даних, вже включені методи побудови дерев рішень. В областях, де висока ціна помилки, вони служать відмінною підмогою аналітика або керівника

Дерева рішень успішно застосовуються для вирішення практичних завдань в наступних областях для:

- оцінки кредитоспроможності клієнтів банку при видачі кредитів у банківській справі;
- контролю за якістю продукції (виявлення дефектів), випробування без руйнувань (наприклад перевірка якості зварювання) у промисловості;
- діагностика різних захворювань у медицині;
- аналізу будови амінокислот у молекулярній біології.

Це далеко не повний список застосувань, де можна використовувати дерева рішень.

### 1.5 Огляд існуючих систем розпізнавання звуків

На сьогоднішній день є досить велика кількість програм, які призначені для розпізнавання звуків, але усі вони є вузькоспеціалізованими у своїй області та мають кожен свої недоліки та переваги. Слід виділити систему яка дає змогу розпізнати постріли — ShotSpotter. Проте більшість подібних програм складають системи розпізнавання мови людини та системи розпізнавання музичних файлів. Це такі системи як Google Assistant, Siri, та Shazam. Звичайно є багато різних подібних систем, але перелічені вище є найбільш відомі. Нижче буде детальніше розказано про кожен із систем.

#### 1.5.1 ShotSpotter

Система виявлення пострілу — це система, яка виявляє і передає розташування стрільби зі зброї або іншої зброї з використанням акустичних, оптичних або потенційно інших типів датчиків, а також комбінації таких датчиків. Ці системи використовуються правоохоронними органами, службами безпеки, військовими і комерційними підприємствами для визначення джерела і, в деяких випадках, напрямки стрільби або типу пострілу. Однією з найвідоміших систем такого типу є ShotSpotter (рис. 1.8).



Рисунок 1.8 — Локатор стрільби



Точне виявлення і точне визначення місця розташування пострілів є складним завданням, враховуючи численні перешкоди такі як будівлі, дерева, ехо-сигнали, вітер, дощі і вуличний шум. Системи, що використовуються в міських умовах, об'єднують географічну інформаційну систему, тому на дисплеї відображається карта і адреса кожного місця події.

Дана система працює таким чином. Програмне забезпечення аналізує звукові сигнали на предмет можливих пострілів. Потім відфільтровує навколишній фоновий шум, такий як рух або вітер, і слухає імпульсивні звуки, характерні для стрільби. Якщо датчик виявляє імпульс, він витягує імпульсні характеристики з форми хвилі, такі як різкість, сила, тривалість і час загасання. І якщо щонайменше три датчика виявляють імпульс, який вважається пострілом, він відправляє невеликий пакет даних на хмарні сервери, де використовується мультилатерація, яка базується на тому що сигнали доходять до датчиків за різний проміжок часу та під різним кутом. Це дозволяє визначити точне місце розташування (рис. 1.9).

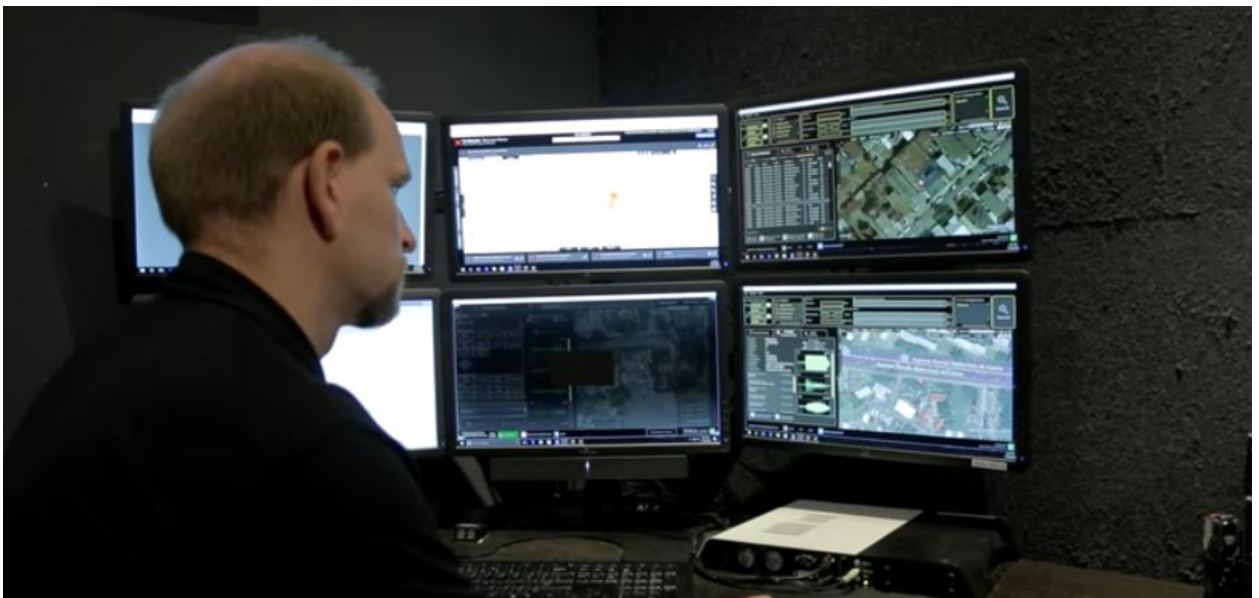


Рисунок 1.9 — Ситуаційний центр виявлення пострілів.

Після того, як програмне забезпечення визначає місце розташування джерела звуку, воно аналізує особливості імпульсу, щоб визначити, чи може звук бути пострілом. Для оцінки і класифікації звуку алгоритми враховують

відстань від джерела звуку, зіставлення зі зразком і інші евристичні методи. Класифікатор машини порівнює звук з великою базою відомих звуків пострілів і інших імпульсивних звуків, щоб визначити, чи є ці звуки пострілами. Після того, як інцидент класифікований як ймовірний постріл, він направляється експертам з акустики в Центрі аналізу інцидентів (IRC) для додаткового аналізу і публікується в поліції або відхиляється.

Система відправляє сповіщення правоохоронним органам і аварійним службам. Повідомлення про інцидент спрацьовують, коли інцидент підтверджується як стрілянина. Сповіщення про стрілянину направляються в мобільні додатки ShotSpotter, а також в програми для настільних комп'ютерів і браузерів. Вся транзакція, від початкової стрільби до оповіщення, відбувається менш ніж за 60 секунд.

### 1.5.2 Google Assistant

Google Assistant представляє з себе хмарний сервіс персонального асистента, розроблений корпорацією Google. До основних його функцій можна віднести розпізнавання мовлення людини та перетворення цієї мови в текст, який в подальшому може бути використана операційною системою та іншими програмами на смартфонах та персональних комп'ютерах. Компанія Google широко використовує даний сервіс в смартфонах, також він включений в Google Allo — додаток для миттєвого обміну повідомленнями, Google Home — розумний голосової Wi-Fi динамік для управління вашим будинком, Android Wear — розумні годинник.

Також розробники сторонніх додатків можуть включати усі функції асистента до своїх розробок. Для цього існує спеціальний пакет SDK Google Assistant. Він дозволяє додавати у ваші розробки визначення гарячих слів, голосове керування, розуміння природної мови і розумні здібності Google. Ваш проект захоплює висловлювання (голосовий аудіо запит, наприклад «Що в моєму календарі?»), Відправляє його в Google Assistant і отримує голосовий звуковий відповідь на додаток до необробленому тексту

висловлювання. SDK в даний час знаходиться в попередньому перегляді для розробників. Продукт є безкоштовно для використання тільки в цілях розробки і має обмеження на кількість запитів, які ви можете зробити в день.

Сьогодні продукт дійшов до рівня на якому він може фактично безпомилково розпізнавати мову людини та вести прямий діалог між машиною та живою людиною. Фактично у реальному часі можна вести діалог з іноземцем, де мобільний телефон буде виступати у якості перекладача. Сервіс є дуже корисним і цікавим, проте він використовується для розпізнавання мови і заточений саме під ці задачі. Він не підходить для розпізнавання будь яких звуків. Крім того Assistant є приватною власністю компанії, і за цей продукт будуть зніматись кошти з користувачів.

### 1.5.3 Siri

Siri — це персональний помічник і система, яка має змогу вести діалог з людиною і є частиною операційній системі iOS, на базі якої працюють такі пристрої, як iPhone або iPad. Додаток використовує обробку мови людини і дає свої рекомендації, відповідає на питання користувача, виконує задані операції. Що цікаво, Siri має можливість підлаштовуватися під кожного користувача, вивчаючи його переваги, в результаті чого відповіді на одні й ті ж питання для декількох користувачів будуть різними.

Компанія SRI International, що є підрозділом DARPA (Агентство з перспективних оборонних науково-дослідних розробок), в 2007 році почала роботу над Siri, а не Apple, як багато хто припускає. Цікавим є той факт, що Siri є результатом роботи досліджень, які накопичувалися більше сорока років. Початковими розробниками Siri стали Даг Кітлаусс, Том Грюбер, Норман Винарський і Адам Чейнер. Уявіть, наскільки масштабна була передісторія створення Siri, а точніше обсяг роботи дослідників в області розпізнавання і обробки голосу. По великому рахунку Siri є аналогом Google Assistant, який описаний у попередньому розділі. Ці дві системи виконують схожі функції, але кожна з них розроблялась повністю окремо різними

командами розробників. Тому можна сказати, що це дві незалежні конкуруючих системи.

Siri використовує удосконалюються технологію розпізнавання мови людини, яку розробила компанія Nuance Communications. Siri індивідуально пристосовується до кожного користувача: слухає і вивчає свого власника, аналізуючи його переваги та особливості мови.

Отже, якщо підбити підсумок, то продукт компанії apple — це продукт для розпізнавання мови людини, який має усі недоліки і переваги подібного рішення від google.

#### 1.5.4 Shazam

Shazam — безкоштовний кросплатформений проект, що дозволяє користувачеві визначити, що за пісня грає в даний момент. Компанія була заснована в 1999 році; штаб-квартира розташована в Лондоні.

Мобільні технології та неймовірний прогрес в області обробки звуку дають розробникам алгоритмів можливість створювати додатки для розпізнавання музичних творів. Одне з найпопулярніших рішень такого роду називається Shazam. Якщо дати йому 10 секунд звучання будь якої частини музичної композиції, неважливо, чи буде це шматок вступу, приспіву або частина основного мотиву, Shazam створить сигнатурний код, звіриться з базою даних і скористається власним алгоритмом розпізнавання музики для того, щоб видати назва твору.

Shazam використовує мікрофон смартфона або комп'ютера, щоб записати короткий зразок відтвореного звуку. Якщо звук записаний нечітко, навколо розмовляють люди або на задньому фоні раптово проїхала вантажівка — не біда. Головне — визначити основну частину.

Використовуючи записаний звук, додаток створює акустичний відбиток та відправляє його на сервер, де він порівнюється із сотнями тисяч збережених відбитків, що знаходяться в центральній базі даних Shazam на предмет відповідності.

Якщо програма знаходить відповідну пісню, то на вікні керування виводиться така інформація, як ім'я виконавця, назва пісні та альбому.

Даний додаток є ганим рішенням для розпізнавання музичних композицій. Але як і попередні розробки, він є вузько спеціалізованим та закритим. Ми не можемо використовувати напрацювання авторів проекту так як це є їх власністю.

## 2 МАТЕМАТИЧНЕ ПІДГРУНТЯ РОЗПІЗНАВАННЯ АУДИОСИГНАЛІВ НА ОСНОВІ ГАУСОВСЬКИХ ЗМІШАНИХ МОДЕЛЕЙ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

### 2.1 Огляд основних компонентів інформаційної технології

На вхід програмного комплексу з розпізнавання звуків має надходити сигнал у форматі .wav файлу, після чого він буде оброблятися різними компонентами системами, що входять до складу інформаційної технології. Вона складається із п'яти частин. Графічне зображення усіх складових компонентів інформаційної технології наведено на рисунку 2.1.



Рисунок 2.1 — Компоненти інформаційної технології аудіодетекції звукових сигналів немовленнєвого походження

На початку сигнал розбивається на фрейми та потрапляє у систему обрахунку кепстральних коефіцієнтів. Кожен фрейм перетворюється у вектор ознак, які можуть бути порівняні із наявною базою збережених моделей звуків. База еталонних моделей є другим компонентом системи. Після цього у справу вступає третій компонент — система вимірювання відстані до еталонних моделей. Тут фрейму встановлюється відстань до кожної моделі, яка наявна в базі. Маючи інформацію від попередньої системи, та оперуючи

додатковою інформацією про звук, це така інформація як мінімальна та максимальна можлива тривалість звуку, система прийняття рішення вирішує до якої моделі можна віднести поточний фрейм. Ну і на сам кінець, система виведення результатів відображає результат користувачеві. Детальніше усі компоненти системи будуть розглянуті в наступних підрозділах.

## 2.2 Загальна схема цифрової обробки аудіосигналу для отримання параметрів моделі

Завдання обробки сигналів схематично представлена на рис. 2.2. Джерелом звукової інформації можуть бути як предмети так і живі істоти. У нашій базі звуків які розпізнаються присутні звуки зброї, трактора, спів пташки та крик людини. На стадії вимірювання або спостереженню зазвичай відбувається вимірювання акустичного коливання. Обробка сигналу передбачає в першу чергу формування опису на основі деякий моделі з подальшим перетворенням отриманого уявлення необхідну форму. Останнім кроком в процесі обробки є виділення необхідної інформації уже з відцифрованого сигналу і використання інформаційного змісту сигналу[13].



Рисунок 2.2 — Схема обробки звукового сигналу

Після того як пройшло вимірювання сигналу, його потрібно перетворити у цифрову форму, щоб у подальшому компютерна техніка розуміла його, і за допомогою програмних засобів ми мали змогу робити з ним різні маніпуляції та вибирати необхідні данні. Для перетворення будь-якого аналогового сигналу в цифрову форму необхідно виконати три основні операції: дискретизацію, квантування і кодування.

Дискретизація — це процес під час якого неперервний в часі аналоговий сигнал представляється представляється відліками у конкретно задані проміжки часу. Відліки сигналу беруться в моменти часу, які відокремлені один від одного інтервалом. Цей інтервал називається інтервалом дискретизації. Величину, зворотню до інтервалу між відліками, називають частотою дискретизації. На рис. 2.3 показаний процес дискретизації.

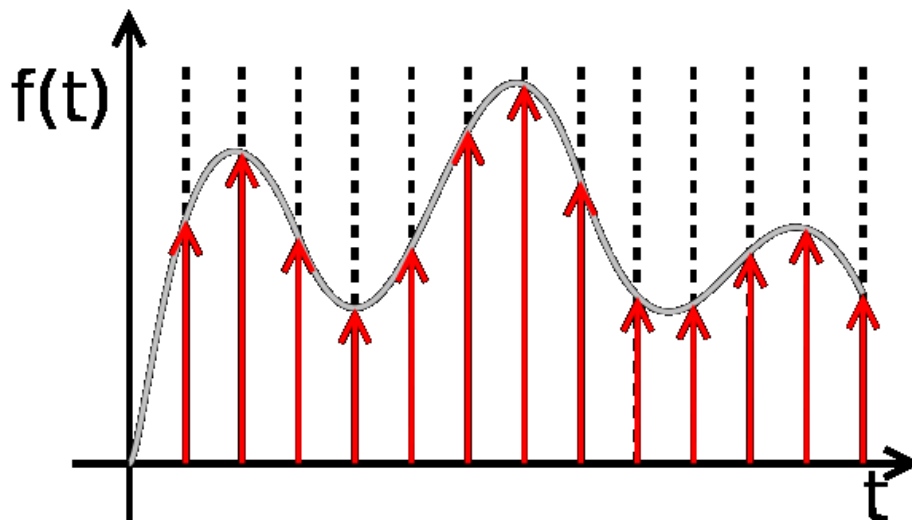


Рисунок 2.3 — Дискретизація сигналу

Зрозумілим є той факт, що чим менше інтервал дискретизації тим менше розходження між вихідним сигналом і його дискретизованої копією. Проте на зберігання такого сигналу потрібно більше місця. Тому це число має бути оптимальним.

Квантування ж являє собою заміну величини відліку сигналу найближчим значенням з набору які є фіксованими — рівнів квантування.



Можна сказати що квантування — це округлення величини відліку. Рівні квантування ділять весь діапазон можливої зміни значень сигналу на кінцеве число інтервалів — кроків квантування. Розташування рівнів квантування обумовлено шкалою квантування. Процедура квантування зображена на рисунку 2.4.

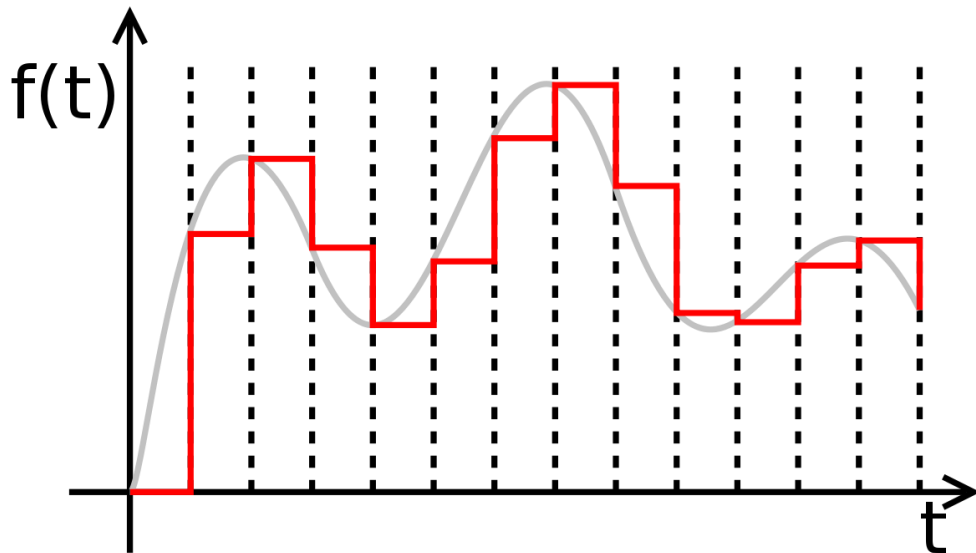


Рисунок 2.4 — Квантування сигналу

Квантований сигнал, на відміну від вихідного аналогового, може приймати тільки кінцеве число значень. Це дозволяє представити його в межах кожного інтервалу дискретизації числом, рівним порядковому номеру рівня квантування. У свою чергу це число можна виразити комбінацією деяких знаків або символів. Сукупність знаків (символів) і система правил, за допомогою яких дані представляються у вигляді набору символів, називають кодом. Кінцева послідовність кодових символів називається кодовим словом. Квантований сигнал можна перетворити в послідовність кодових слів. Ця операція і називається кодуванням. Кожне кодове слово передається в межах одного інтервалу дискретизації. Для кодування сигналів звуку і зображення широко застосовують двійковий код. Якщо квантований сигнал може приймати  $N$  значень, то число двійкових символів в кожному кодовому слові  $n \geq \log_2 N$ . Один розряд, або символ слова, представлений в двійковому

коді, називають бітом. Зазвичай число рівнів квантування одно цілої ступеня числа 2, тобто  $N = 2^n$ .

Після того як ми будемо мати сигналу цифровій формі ми можемо проводити його аналіз. Якщо ми подивимось на сигнали різних звуків то побачимо що параметри цих сигналів не є статичними. Їх аналіз проводити дуже важко. Тому розіб'ємо наші дані на невеликі часові проміжки — фреймам. Причому фрейми повинні йти не строго один за одним, а перекриваючи один одного. Цей процес зображено на рисунку 2.5.

Фрейми є більш придатною одиницею аналізу даних, ніж конкретні значення сигналу, так як аналізувати хвилі набагато зручніше на деякому проміжку, ніж в конкретних точках. Розташування ж фреймів "з перекриттям" дозволяє згладити результати аналізу фреймів, перетворюючи фрейм в так зване "вікно", що рухається уздовж вихідної функції (значень сигналу).

Дослідним шляхом встановлено, що оптимальна довжина фрейму повинна відповідати проміжку в 10мс, «перекриття» має становити 50%.

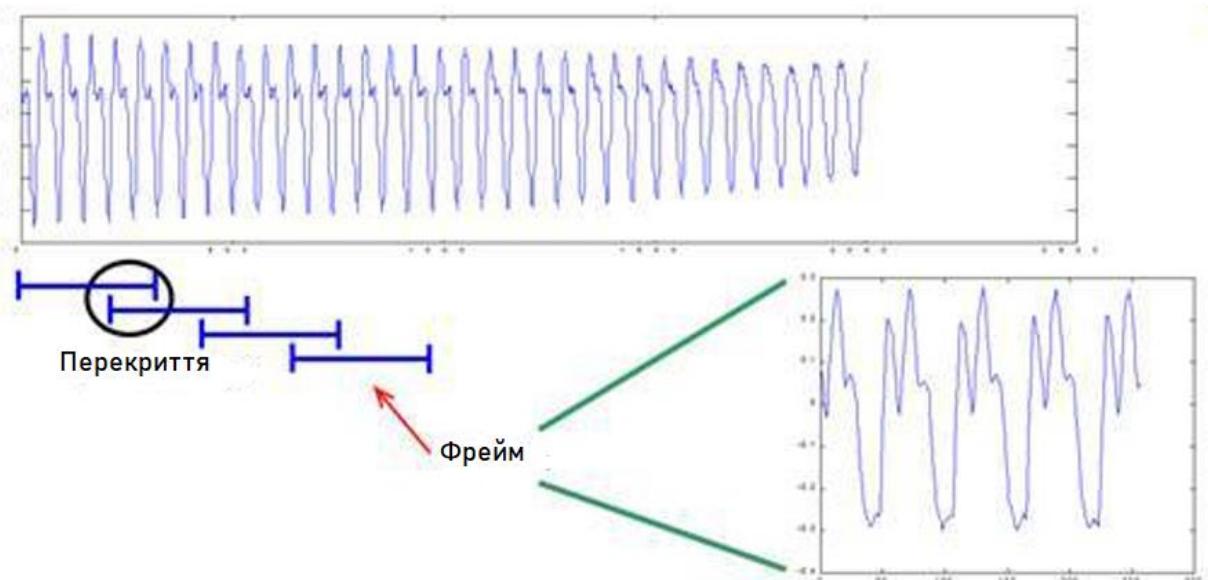


Рисунок 2.5 — Розбиття сигналу на фрейми

Розбивши вхідний сигнал на фрейми можна переходити до виділення необхідної інформації з кожного фрейму.

Існує декілька підходів до отримання дескрипторів (ознак) з аудіо сигналу. Всі вони задаються спільною метою зменшити надлишковість сигналу і виділити найбільш релевантну інформацію, і, в той же час, відкинути нерелевантну. Ознаки, що описують аудіо сигнал з різних боків, комбінуються в один вектор ознак, на основі якого відбувається процес навчання і потім класифікації з використанням обраної навченої моделі.

Є багато ознак які можна виділити з аудіо сигналу. Для розпізнавання фреймів у розробленій програмі було вирішено використовувати кепстральні коефіцієнти.

### 2.3 Обчислення кепстральних коефіцієнтів

Обробка сигналу має на меті виділити в звуковому сигналі інформацію, релевантну для задачі розпізнавання звуків. Виділені ознаки будуть використані для формування шаблону або для порівняння з зареєстрованими шаблонами. Априорі неможливо оцінити, які ознаки більш підходять для розпізнавання. Процес визначення відповідних ознак полягає в переборі можливих варіантів ознак з наступною експериментальною оцінкою.

Сформований нами підхід до процедури обробки мовного сигналу полягає у використанні короткочасного аналізу. Тобто сигнал розбивається на тимчасові вікна фіксованого розміру, на яких, як передбачається, параметри сигналу не змінюються. Для звукових сигналів розмір вікна зазвичай вибирається в межах від 5 мс до 20 мс.

Для більш точного уявлення сигналу між вікнами роблять перекриття, рівну половині довжини вікна. Потім до кожного вікна застосовуються алгоритми вилучення ознак. У нашому випадку це будуть кепстральні коефіцієнти (формула 2.1).

$$c(n) = \text{ifft}(\log |\text{fft}(x(n))|), \quad (2.1)$$

де  $c(n)$  — сигнал представлений спектром;

$(x(n))$  — вхідний сигнал.

Слово «кепстр» — від лат. *capstrum* — означає косинус-перетворення Фур'є логарифма спектра потужності, яке математично позначається формулою вище.

Кепстральні коефіцієнти будуються на основі лінійного передбачення. Суть лінійного передбачення полягає в тому, що лінійною комбінацією певної кількості попередніх відліків можна апроксимувати поточний відлік.

$$x_n \approx \sum_{k=1}^p a_k x_{n-k},$$

Вагові коефіцієнти лінійної комбінації  $a_1, \dots, a_p$  називаються коефіцієнтами лінійного передбачення [10].

Знаходження коефіцієнтів лінійного передбачення здійснюють за допомогою рекурсивного алгоритму Дарбіна [13].

На основі отриманих коефіцієнтів лінійного передбачення розраховуються кепстральні коефіцієнти. Причому таких коефіцієнтів може бути згенеровано більше, ніж самих коефіцієнтів лінійного передбачення.

$$x_n = \begin{cases} a_n + \sum_{k=1}^{n-1} \frac{k}{n} c_k a_{n-k}, & 1 \leq n \leq p; \\ \sum_{k=n-p}^{n-1} \frac{k}{n} c_k a_{n-k}, & n > p. \end{cases}$$

Для сигналу з частотою дискретизації 44100 Гц використовують близько 10 коефіцієнтів лінійного передбачення, з яких генерують 10 кепстральних коефіцієнтів.

#### 2.4 Обчислення відстані

Розпізнавання звуків відрізняється від багатьох біометричних систем тим, що в даному випадку розпізнається динамічний процес, а не, наприклад, статичне зображення відбитків пальців або райдужної оболонки ока. Тому

звук представляється не у вигляді єдиного вектору, а у вигляді послідовності векторів ознак, кожен з яких описує невелику ділянку з певними характеристиками.

Послідовність векторів, отримана після етапу обробки сигналу, використовується для побудови шаблону моделі або для здійснення порівняння з уже побудованими шаблонами. Для розпізнавання має бути визначений спосіб обчислення ступенів подібності пред'явленого зразка з одним або декількома шаблонами. Ступінь подібності обчислюється на основі певної метрики або на основі оцінки ймовірності.

Визначення методу обчислення відстані є основою для шаблонних моделей. У таких моделях об'єкт що розпізнається розглядається як неточна копія одного зі збережених.

Одними з найпоширеніших методів обчислення відстані між векторами є наступні:

— відстань міських кварталів (манхеттенська відстань)

$$d(x, y) = \sum_{i=1}^D |x_i - y_i|;$$

— евклідова відстань

$$d(x, y) = \sum_{i=1}^D (x_i - y_i)^2;$$

— відстань Махалобіса

$$d(x, y) = (x - y)^T * W^{-1} * (x - y),$$

де  $W^{-1}$  — матриця коваріації.

У разі якщо  $W$  є одиничною матрицею, то відстань збігається з евклідовою. Для обрахунку відстані в змішаних гаусових моделях використовується відстань Махалобіса.

Модель гауссових сумішей широко використовується в області розпізнавання звуків. Дана модель являє собою зважену суму Гауссіан.

$$p(x \parallel \lambda) = \sum_{i=1}^M w_i p_i(x),$$

де  $\lambda$  — модель певного класу;

$M$  — кількість компонентів моделі;

$w_i$  — ваги компонентів такі, що

$$\sum_{i=1}^M w_i = 1.$$

Функція щільності ймовірності кожного компонента задається формулою 2.2

$$p_i(x) = \frac{1}{2\pi^{D/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right). \quad (2.2)$$

де  $D$  — розмірність простору ознак;

$\mu_i$  — вектор математичного очікування;

$\Sigma$  — матриця коваріації.

Таким чином, для побудови моделі розпізнавання необхідно визначити вектори середніх, матриці коваріації і ваги компонентів. Дану задачу вирішують за допомогою EM-алгоритму[12]. На вхід подається початкова послідовність векторів  $X = \{x_1, \dots, x_t\}$ . Параметрам моделі присвоюються початкові значення і потім на кожній ітерації алгоритму відбувається переоцінка параметрів.

Для визначення початкових параметрів зазвичай використовують алгоритм кластеризації такий, як алгоритм К-середніх [8]. Побудувавши розбиття множини навчальних векторів на  $M$  кластерів, параметри моделі можуть бути ініційовані в такий спосіб. Початкові значення  $\mu_i$  збігаються з

центрами кластерів, матриці коваріації розраховуються на основі даних векторів що потрапили в даний кластер, ваги компонентів визначаються часткою векторів даного кластера серед загальної кількості навчальних векторів.

На відміну від класичного класифікатора Байєса, де для апроксимації кожного класу виступали параметри гауссівського розподілу, як опція щільності розподілу ймовірності в GMM (Gaussian Mixture Model) моделі виступає «суміш» з декількох Гауссіан [11], параметри яких для кожного класу підбираються на етапі навчання з використанням EM (Expectation Maximization) алгоритму. Як можна бачити на рисунку 2.6, що ілюструє результат класифікації трьох класів, область простору, що відповідає кожному класу, являє собою більш складну форму, ніж еліпс, модель розпізнавання стала більш точною, відповідно, результат розпізнавання має покращитися

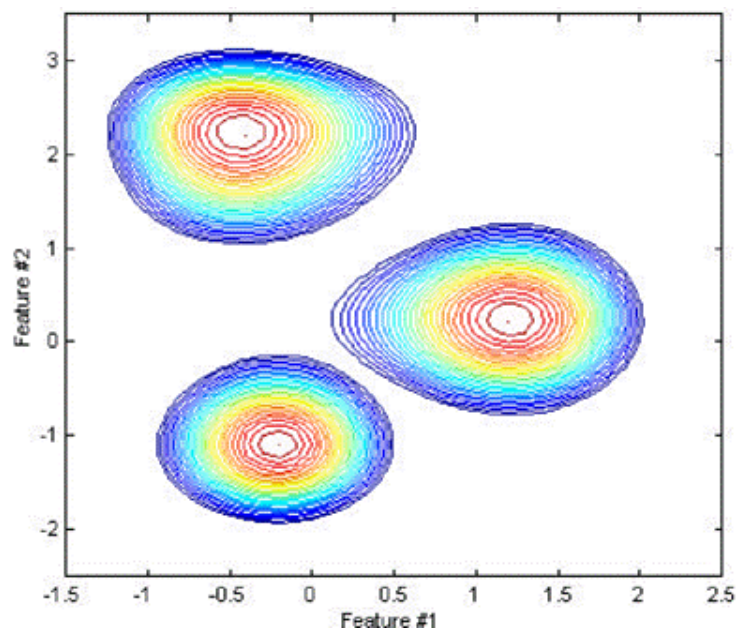


Рисунок 2.6 — Приклад застосування класифікатора на основі GMM

У програмі, яка буде обчислювати наближеність векторів, що проходять перевірку, кожному вектору буде повертатись певне число, яке характеризує його наближеність до певної шаблонної моделі. Чим це число більше тим більша імовірність того, що даний вектор і є шуканим об'єктом.

Для наглядної демонстрації було взято один фрейм звуку для якого наявний шаблонний клас в базі. Ось як буде виглядати графік відстаней при порівнянні з ними фрейму звуку трактора. Як можна побачити із рисунка 2.7, число що характеризує наближеність цього фрейму до класів є найбільшим саме для моделі звуку трактора. Вона складає 8,67081. Відстані до інших моделей є суттєво нижчими. Найближчою є відстань до звуку рушниці, яка приблизно складає — 16.

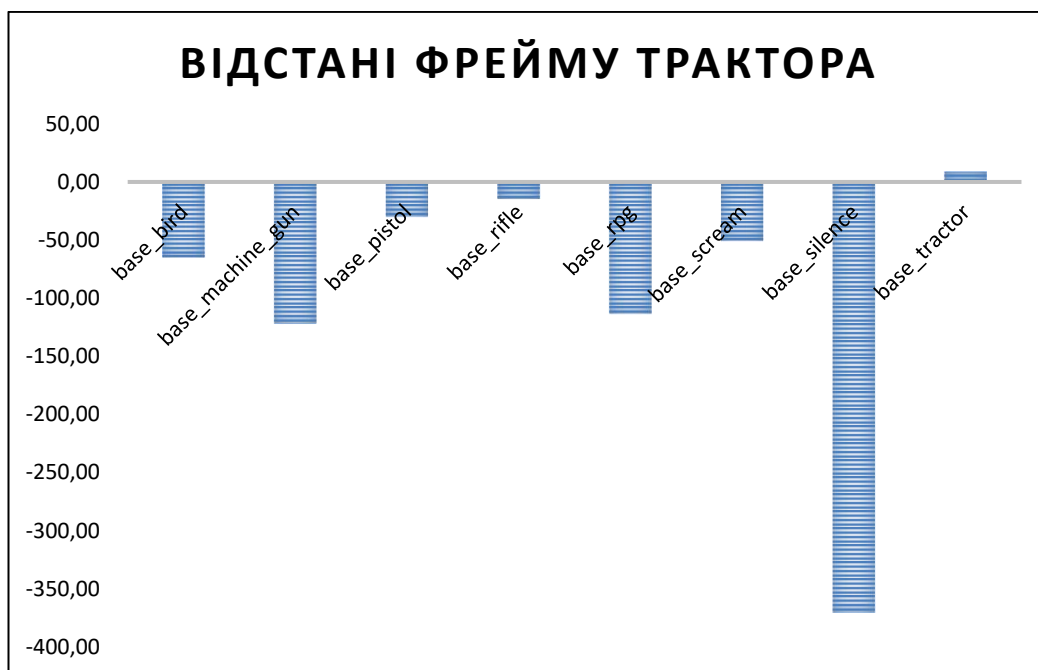


Рисунок 2.7 — Порівняння відстаней з усіма шаблонними класами

Проте не у всіх випадках результати співпадіння з шаблонними моделями є такі хороші. Іноді можуть траплятись помилкові співпадіння. Саме для цього, для вирішення про віднесення певного фрейму до шаблону, ми беремо середнє значення відстані по 30 послідовних фреймах, що дає нам змогу уникнути випадкових співпадінь та покращити результат.

Також, для кожного шаблону нам потрібно знайти певне значення еталонної відстані. Ми це можемо зробити пропустивши деяку кількість векторів які точно належать даній моделі. Після того як буде знайдений еталон можна задати вікно відхилення у яке мають потрапляти шукані вектори.



## 2.5 Вибір мови програмування

C ++ уже протягом багатьох років широко використовується для розробки програмного забезпечення, будучи однією з найпопулярніших мов програмування. Область його застосування включає створення операційних систем, різноманітних прикладних програм, драйверів пристроїв, додатків для вбудованих систем, високопродуктивних серверів, а також ігор. Простіше кажучи, на мові C++ можна написати практично будь-що.

Перш ніж розглядати переваги і недоліки C ++, слід ознайомитися з правилами, які використовував Бйорн Страуструп при проектуванні даної мови [14]:

- розроблено як універсальна мова зі статичними типами даних із ефективністю і підтримкою коду на мові Cі;
- розроблений так, щоб безпосередньо і всебічно підтримувати безліч стилів програмування (процедурне програмування, абстракцію даних, об'єктно-орієнтоване програмування та узагальнена програмування);
- розроблено так, щоб давати програмісту свободу вибору, навіть якщо це дає йому можливість вибирати неправильно;
- розроблено так, щоб максимально зберегти сумісність з Cі, тим самим роблячи можливим легкий перехід від програмування на Cі;
- уникає таких особливостей, які залежать від платформи або не є універсальними;
- не накладає надлишкового навантаження на програму, тобто не забирає додаткові ресурси, а використовує те що безпосередньо потрібно;
- мова розроблена так, щоб не вимагати занадто ускладненою середовища програмування.

Універсальність C ++ це значна перевага даного мови програмування. Оволодівши основами, розробник зможе писати на C ++ прості та складні програми для персональних комп'ютерів, що працюють під управлінням операційної системи Windows та Linux, ігри для різних платформ (включаючи Windows та Android), серверну частину сайтів (back-end) і так

далі. Широке використання даної мови в усіх галузях дає багато можливостей.

Варто також відмітити ефективність використання ресурсів обчислювальної техніки[24, 25]. Адже програми написані на C, C++ одразу компілюються в машині команди процесора. Це дозволяє ефективно використовувати ресурси та проводити обчислення дуже швидко, що важливо для нас, адже система має працювати в реальному часі і повинна швидко опрацьовувати данні.

Якщо ми поглянемо на статистику використання мов програмування за останні 10 років то побачимо, що C та C++ упевнено тримають свої позиції. І поява високорівневих та зручних для програміста мов програмування як java та C# не витіснили її. Це пов'язано в першу чергу з ефективність цієї мови. Адже програми написані на C# та java виконуються в рази повільніше ніж аналогічні на C++. Також для своєї роботи вони вимагають встановлення додаткових програм інтерпретаторів що вимагає додаткових ресурсів. Усе це не дозволяє нам використати ці популярні мови в нашому випадку.

Ще одним великим плюсом на користь вибору даної мови є те що вона присутня на ринку уже дуже довго. За цей період для неї було створено безліч різних корисних бібліотек та проектів. Розв'язання багатьох математичних задач та алгоритмів реалізовано на цій мові і є у вільному доступі в інтернеті.

Варто також враховувати специфіку нашої області. При виборі мови програмування для машинного навчання потрібно враховувати доступні функції і бібліотеки. Сьогодні можна припустити, що найбільш підходящими мовами програмування в професійній області машинного навчання є Python, Java, та C++.

Для розв'язання нашої проблеми потрібна бібліотека яка б давала змогу працювати з змішаними гаусовськими моделями та використовувати їх для класифікації приналежності до шаблонних моделей. Саме такою бібліотекою є mlpack[26].



Рисунок 2.8 — Логотип бібліотеки машинного навчання

Mlpack — це швидка, гнучка бібліотека машинного навчання, написана на C++, метою якої є надання швидких, розширюваних реалізацій передових алгоритмів машинного навчання. Бібліотека надає ці алгоритми у вигляді простих програм командного рядка, прив'язок Python і класів C++, які потім можуть бути інтегровані в більш масштабні рішення для машинного навчання. Дане програмне забезпечення є вільно розповсюджуваним з відкритим вихідним кодом. Це дозволяє нам вільно використовувати його у нашій програмі та в разі необхідності вільно модифікувати.

## 2.6 Розробка сутностей

Отож, у попередніх розділах було вирішено використовувати Гаусовські змішані моделі для створення класів шаблонів та порівняння з ними векторів фреймів які потрібно розпізнати.

Для представлення кожного фрейму була створена структура, яка виглядає наступним чином:

```
struct SampleFrame
{
    double probability = 0;
```

```

std::string nameSound = "NotRecognized";
double distance = 10;
};

struct Frame
{
    unsigned long num = 0;
    double probability = 0;
    std::string nameSound = "NotRecognized";
    vec data;
    vector<SampleFrame> possibleFrame;
    list<SampleFrame> listVallueAllModel;
};

```

Структура Frame містить усі данні які необхідні програмі в процесі розпізнавання. Поле num використовується для нумерації кожного фрейму. Так як послідовність фреймів може бути доволі великою то в процесі роботи програми, під час створення нового фрейму, можна перевіряти чи не перевищує це значення певну межу. І скажімо, якщо система має працювати постійно, в режимі реального часу, це значення кожен годину можна скидати в 0. Обнулення нумерації ніяк не має впливати на роботу програми адже дане поле використовується тільки для виводу інформації на екран.

Поле probability зберігає натуральний логарифм найбільш підходящого значення відстані яке ми отримуємо в процесі порівняння фреймів з шаблонними значеннями. Відповідно поле nameSound зберігає назву моделі, якій відповідає найбільш підходяще значення probability.

У полі data знаходиться вектор кепстральних коефіцієнтів, на основі яких робиться порівняння з шаблоном.

Також Frame складається із параметризованого списку listVallueAllModel де повинні лежати значення перевірки на співпадіння з усіма шаблонами які присутні в базі та список possibleFrame де зберігаються

значення із списку `listVallueAllModel`, але не всі. Тільки ті які задовільняють умови і є претендентами на те щоб бути затвердженими.

Структура `SampleFrame` являє собою урізану версію фрейму. У ній містяться лише поля `probability`, яке виконує функцію аналогічну в попередньо описаній структурі, але ця відстань не обов'язково є найкращою. Поле `distance` містить значення, яке показує наскільки віддалене значення `probability` від еталонного значення моделі.

У програмі присутня структура, яка відповідає кожній шаблонній моделі.

```
struct MyGMMmodel
{
    int first50Frame = 0;
    bool enoughtData = false;
    list<double> listLast50FrameProbab;
    double averageProbability = 0;
    int maxDurationFrame = 0;           //Якщо 0 то обмежень немає
    int minDurationFrame = 0;          //Якщо 0 то обмежень немає
    double idealprobabilities = 10;
    double edge = 10;
    std::string name;
    std::string locateFile;
    GMM* gmm;
    bool activeNow = false;             //зараз у фреймі є співпадіння
    list<double> listGoodFrame;
    bool enoughtListGoodFrame = false;
};
```

Поля `first50Frame` та `enoughtData` необхідні для сигналізації про накопичення достатньої кількості фреймів в списку `listLast50FrameProbab`, щоб розпочинати процедуру порівняння.

Після того як список наповнився значеннями знаходиться середнє значення що записується в поле `averageProbability`.

Ми можемо вказати мінімальну і максимальну послідовність фреймів яка відповідає конкретному шаблону. Ці значення задаються у полях `minDurationFrame` та `maxDurationFrame` Скажімо постріл не може тривати менше 0.5 секунд і не більше 1.5 секунд. Ці значення завантажуються із конфігураційного файлу разом із шаблонними моделями.

Поле `idealprobabilities` містить еталоне значення відстані, яке знаходиться шляхом пропускання капстральних коефіцієнтів які точно присутні в моделі. У кожного еталону є поле `edge`. Це границя допустимих відстаней. Якщо порівнюваний вектор виходить за межі границі він вважається незадовільним і відкидається.

Назва моделі записується у поле `name`. У полі `locateFile` міститься адреса файлу звідки було завантажено модель. Головним полем кожної моделі є `gmm`. Це посилання на шаблону модель, яка проводить перевірку на співпадіння з некласифікованим фреймом.

Список `listGoodFrame` містить набір послідовних фреймів які задовільняють умови перевірки але ще не можуть бути записані у список `Frame.possibleFrame` так як у моделі активне поле `minDurationFrame` та `maxDurationFrame`. У цьому випадку нам потрібно дочекатись закінчення задовольняючих умови послідовності фреймів. І якщо кількість фреймів є такою що входить у вказані межі ми записуємо ці фрейми в список `Frame.possibleFrame`.

## 2.7 Загальний алгоритм роботи програми

Програма розроблена на мові C++ та має вигляд програми для командного рядка. Загальний алгоритм додатку наведено на рисунку 2.9.

Створення сутностей передбачає створення списку у який будуть завантажені шаблони моделі та сам список фреймів. Це робиться наступним чином:

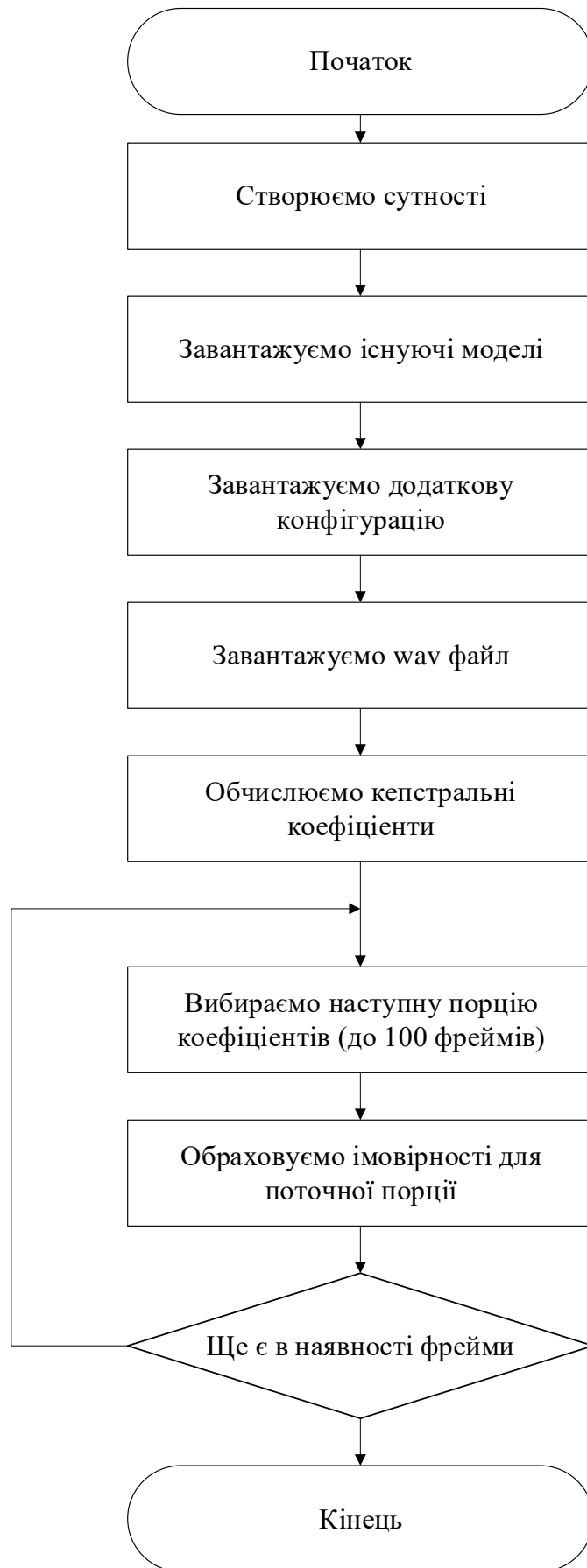


Рисунок 2.9 — Схема загального алгоритму роботи програми

```
vector<MyGMMmodel> listExistGMMModel;
vector<Frame> listFrameSound;
```

Після цього іде завантаження самих моделей. Передбачається, що натреновані моделі мають знаходитись в каталозі `\data\gmm_model`, що є відносним що до виконуваного файлу програми. Усі файли що знаходяться тут вважаються моделями і програма намагається їх завантажити. Якщо ж завантаження невдале — програма сповістить про це.

Файл додаткової конфігурації для моделей знаходиться у папці `\data` у файлі `IdealValue.conf`. Тут міститься інформація про еталону відстань. Ми її отримали пропустивши один із звуків, на якому тренувалась модель і обчислили середнє значення відстані усіх фреймів звуку. Також тут знаходяться обмежуючі параметри моделей (максимальна та мінімальна кількість послідовних фреймів).

Далі іде обрахування кепстральних коефіцієнтів з файлу, який ми маємо розпізнавати. Файл має знаходитись у папці `\data` та мати назву `Sound.wav`. Якщо ж такого файлу немає, програма повідомить, що файл з таким іменем не знайдено. Обчислені коефіцієнти записуються в матрицю. Також паралельно вони виводяться в файл `\data\cps\cpsSoundCoef.txt`. При необхідності ми можемо їх переглянути.

Після того як файл був завантажений ми ділимо його на порції по 100 або менше фреймів та записуємо в матрицю, яка виступає в ролі буфера. Далі для розпізнавання має викликатись функція якій ми маємо передати буфер фреймів список наявних на даний момент фреймів, список шаблонних моделей та шлях до файлу який буде виведено розширений результат. Данна функція має наступний вигляд:

```
recognizeNext(mat          next100Points,          vector<MyGMMmodel>
&listExistGMMModel, vector<Frame> &listFrameSound, string pathToSave =
"data/recognized_sound/recognizeResult.txt"){...}
```

Дана функція виконує розпізнавання отриманих на вхід векторів фреймів та порівнює їх із наявними шаблонами у базі. Якщо ж умови



схожості для якоїсь шаблонної моделі є достатніми фрейм визначається схожим на неї.

Якщо ж ще залишились вектори значень, які мають пройти перевірку ми знову вибираємо порцію у 100 фреймів і розпізнавання продовжується. Коли ми вибрали останню порцію фреймів ми виходимо з циклу і програма завершує свою роботу.

## 2.8 Опис алгоритму прийняття рішень

На вхід функції яка має проводити розпізнавання та приймати рішення про приналежність того чи іншого фрейму до певного класу мають надходити такі значення:

- `mat next100Points` (матриця у якій містяться вектори значень);
- `vector<MyGMMmodel> &listExistGMMModel` (список з усіма шаблонними класами що є в нашій базі);
- `vector<Frame> &listFrameSound` (Список попередніх фреймів);
- `string pathToSave` (шлях до файлу з розширеними результатами розпізнавання. За замовчуванням результати зберігаються в файл `data/recognized_sound/recognizeResult.txt`. Також, вони паралельно виводяться на консоль).

Варто відмітити наступний момент. У програмі розпізнавання нам потрібно враховувати мінімальне та максимальне значення можливої послідовності фреймів для кожного класу шаблонної моделі. Тому для того щоб порахувати послідовність цих фреймів ми будемо виводити результат з деякою затримкою. В основному ці значення потрібні для таких моделей як постріли, звук яких не триває більше 2 секунд. Тому мною було прийняте рішення, що затримка виводу результату буде складати саме такий проміжок часу, що відповідає числу фреймів 200. Це в свою чергу накладає обмеження на встановлення мінімального та максимального порогу послідовності фреймів. Вони не можуть перевищувати даний часовий проміжок затримки! В іншому випадку програма буде видавати некоректні значення.

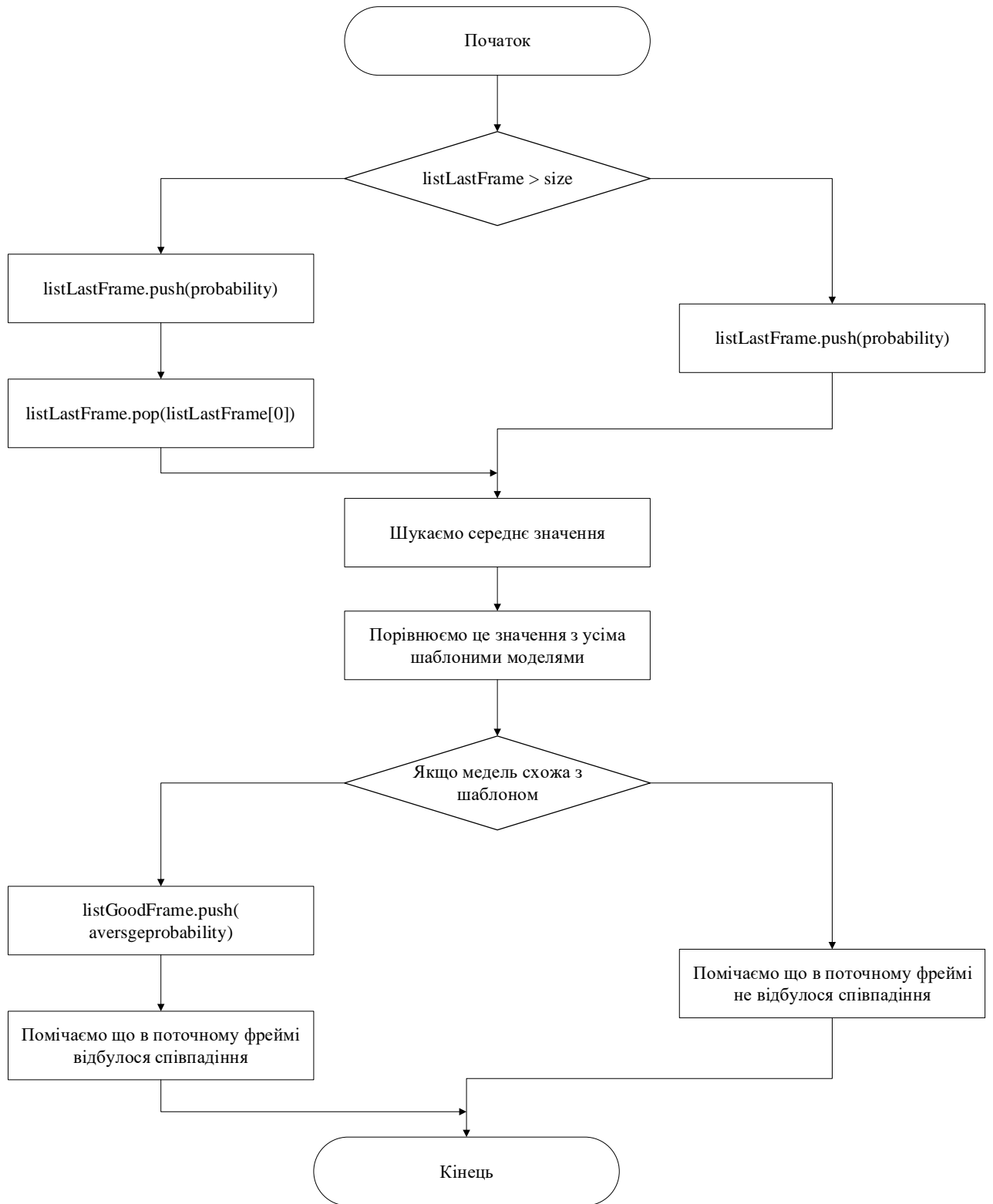


Рисунок 2.10 — Алгоритм обрахування середнього значення

На початковому етапі розпізнавання ми маємо набрати певну послідовність фреймів з якої буде знайдено середнє значення відстані. Це середнє значення ми будемо порівнювати з еталонними величинами і

перевіряти їх віддаленість від них. Якщо відстань не є великою і вкладається в межі зазначені для даної моделі — це означає що фрейм схожий із шаблоном і він є претендентом на те щоб бути віднесеним до даної шаблонної моделі. Він заноситься до масиву `listGoodFrame`, який входить до структури моделі `MyGMMmodel`. Блок схему алгоритму обрахування середнього значення ми можемо побачити на рисунку 2.10.

Але окрім усередненого значення, що дозволяє згладити випадкові коливання відстаней ми враховуємо і можливу тривалість звуку. Скажімо для пострілу можна задати певну тривалість протягом якої мають бути співпадіння з шаблоном. І якщо ця тривалість менша або більша дозволеної ми можемо сказати що звук не належить даній моделі. В програмі ми можемо задати верхню, нижню межу або обидві разом. Якщо ж межі не задані, фрейм одразу записується в список можливих `possibleFrame`, що належить структурі `Frame` обходячи запис в `listGoodFrame`.

Для кожного варіанту обмежень фрейму реалізована своя логіка запису фреймів у `listGoodFrame`. Так якщо в шаблонній моделі задана лише нижня межа ми накопичуємо ці фрейми у списку `listGoodFrame` до моменту коли кількість фреймів у ньому перевищить мінімальне значення. У момент коли це сталося увесь накопичений список записується у загальний `listFrameSound`. Кожен наступний фрейм буде записуватись уже напряму в `listFrameSound` аж до моменту коли співпадіння припиняться. Якщо ж співпадіння припинились раніше — це список анулюється і нікуди не записується.

Коли задана верхня межа ми також накопичуємо послідовність фреймів в `listGoodFrame` і очікуємо моменту коли ці співпадіння припиняться. В цей момент ми рахуємо кількість елементів в списку і якщо вона менша за межову або їй дорівнює ми записуємо елементи масиву в `listFrameSound`. В іншому разі вони просто відкидаються. Але важливо пам'ятати що не можна робити це значення занадто великим, так як вивід результату буде із великою затримкою. Програмою передбачається максимальна затримка 200 фреймів.

Випадок коли задана і верхня і нижня межа більш схожий на обробку моделей у яких наявна верхня межа. Усі дії проведені тут аналогічні, окрім того, що нам потрібно перевіряти щоб у момент припинення співпадінь кількість фреймів у списку `listGoodFrame` була задовільною для верхньої та нижньої межі.

### 3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СТВОРЕНОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ТА РОЗРОБКА ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

#### 3.1 Основні характеристики системи, що досліджується

Для тестування роботи системи була створена база звуків на яких проводилось навчання моделі. Для кожного шаблону було взято по 3 різних звуки. Із цих звуків були отримані капстральні коефіцієнти. Після чого проводилось навчання моделей за допомогою засобів бібліотеки машинного навчання mlpack.

База для розпізнавання нараховує вісім різних звуків:

- спів пташки;
- автомат;
- рушниця;
- пістолет;
- ручний протитанковий гранатомет (RPG);
- крик людини;
- звук роботи трактора;
- тиша.

Усі звуки, присутні у системі розпізнавання, повинні мати однаковий формат. Вирішено взяти аудіосигнали в форматі wav, оскільки він є доволі простим і з ним легко працювати. Це є важливим моментом, так як нам потрібно створювати капстральні коефіцієнти на основі рівнів амплітуд сигналу. На кожен рівень виділяється 16 бітових розрядів. Також, варто зазначити, що звуковий файл має бути у форматі моно.

Частота дискретизації сигналу складає 44 100 Гц. Цієї частоти має вистачити щоб адекватно передати форму сигналу який розпізнається. Тривалість одного фрейму складає приблизно 10 мілісекунд. На один фрейм припадає 441 відліків амплітуд.

Із цих відліків генерується перших 11 капстральних коефіцієнтів. Так як перший коефіцієнт є однаковим для усіх фреймів він не використовується

для побудови вектора. На основі отриманих векторів створюється змішана гаусівська модель.

### 3.2 Результати розпізнавання

Для початку перевіримо результати розпізнавання взявши для тесту одиночні звуки які наявні в базі але не використовувався для навчання моделей. Такими звуками будуть звуки трактора, ручний протитанковий гранатомет та спів пташки.

Усі звуки які проходили перевірку у програмі були однозначно розпізнані. Більш детальна інформація про кількість розпізнаних та не розпізнаних фреймів наводиться нижче.

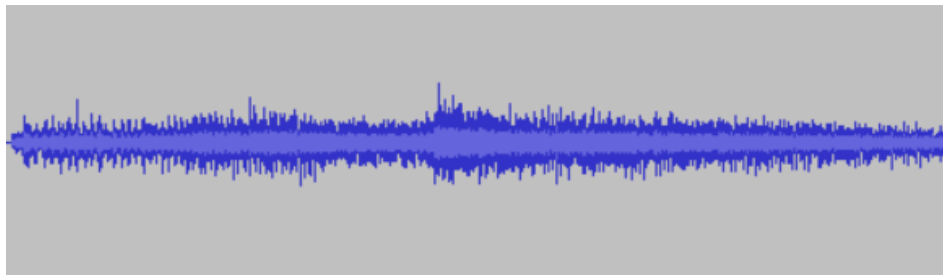


Рисунок 3.1 — Форма сигналу звуку трактора

В результаті, для звуку трактора було розпізнано 615 фреймів із 636 це складає 96,7%. Число фреймів які не були розпізнані складає 21.

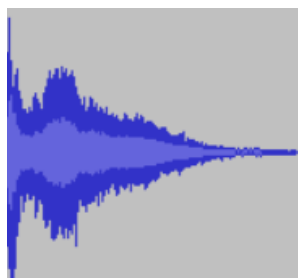


Рисунок 3.2 — Форма сигналу звуку РПГ

Для ручного протитанкового гранатомета показник успішно розпізнаних фреймів складає 73,3%. Було успішно визначено 118 із 161 фрейма. Щоправда цей сигнал був в дечому схожий на інший шаблон,

який розташований у базі сигналів. Також відбувся збіг із сигналом пістолета по 34 фреймах.

І для звуку пташки програма не змогла розпізнати 73 фрейми. Загальна кількість фреймів складала 310 фреймів. Успішно розпізнані фрейми складають 76.5% від загальної кількості.

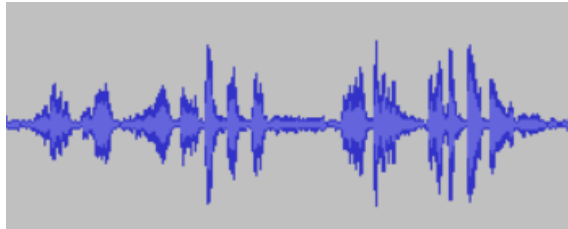


Рисунок 3.3 — Форма сигналу звуку пташки

Отже, перші результати розпізнавання можна вважати вдалим, так як усі звуки були вірно віднесені до своїх моделей.

Далі ми візьмемо багато різних звуків, як наявних в базі так і ні, і змонтуємо їх, щоб вони з'являлись по черзі. Звуки будуть відділені один від іншого невеликими проміжками тиші. В результаті отримаємо один великий звуковий файл. Він зображений на рисунку 3.4 нижче.

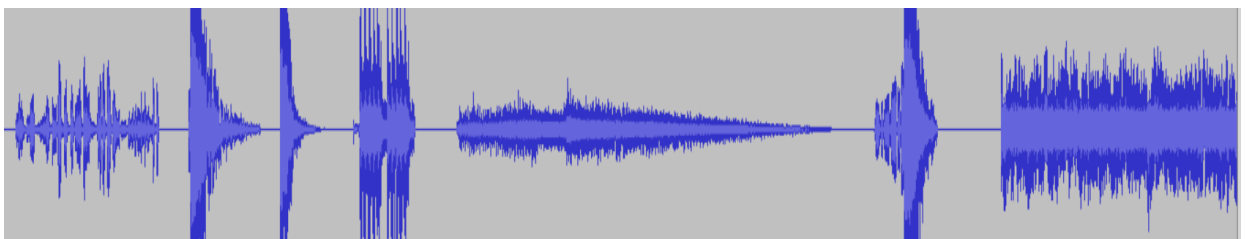


Рисунок 3.4 — Форма сигналу суміші багатьох звуків

Запустивши програму на виконання ми отримали результат який можна вважати доволі вдалим. Усі звуки були успішно віднесені до своїх моделей в базі. Результати розпізнавання які показують які саме фрейми звуків були розпізнані наведено на рисунку 3.5. Не був розпізнаний тільки звук шуму, для якого відсутній шаблонний клас. Більш детальні результати сегментації звуків наведено у таблиці 3.1.

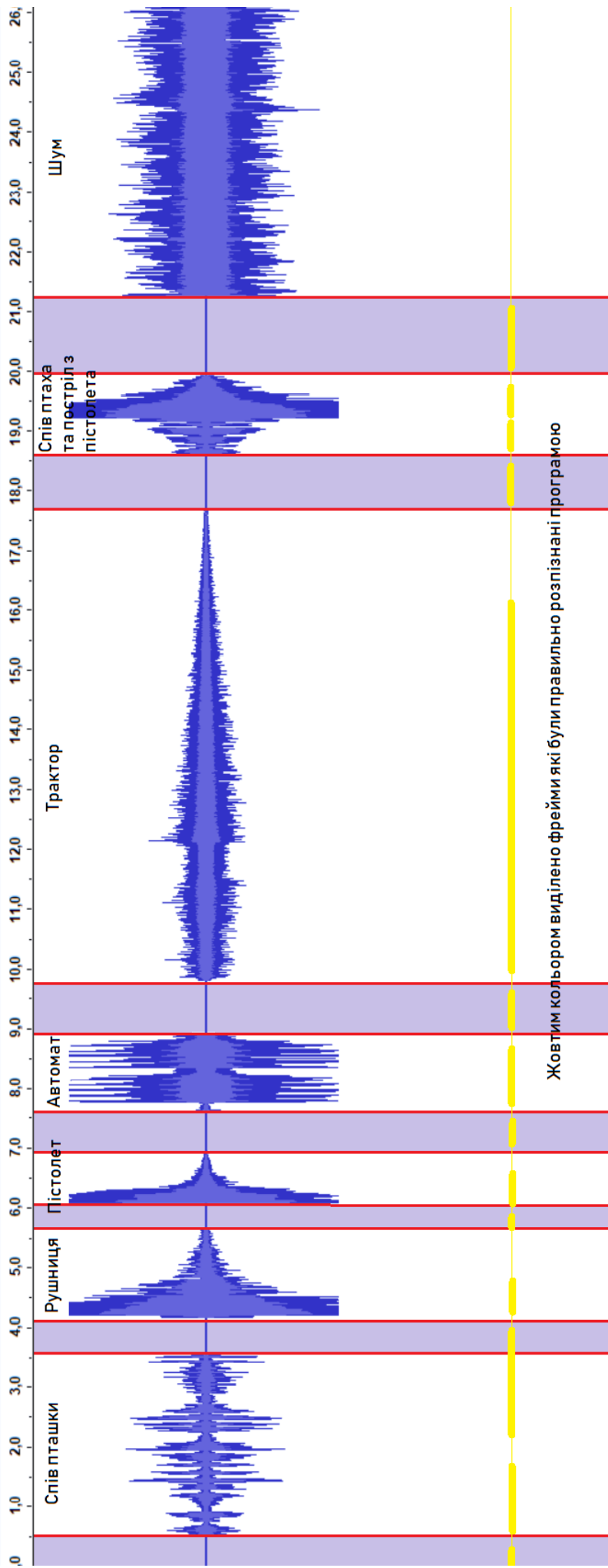


Рисунок 3.5 – Розпізнані за допомогою програми звуку



Таблиця 3.1 — Результати сегментації по кожному фрагменту звуку

№	Назва звуку	Усього фреймів	Розпізнано	Не розпізнано	Відсоток розпізнавання
1	Тиша	52	30	22	58%
2	Спів пташки	310	195	115	63%
3	Тиша	53	34	19	64%
4	Рушниця	157	83	74	53%
5	Тиша	38	19	19	50%
6	Пістолет	90	40	50	44%
7	Тиша	70	45	25	64%
8	Автомат	120	92	28	77%
9	Тиша	90	55	35	61%
10	Трактор	792	639	153	81%
11	Тиша	84	62	22	74%
12	Спів пташки	62	59	3	95%
13	Пістолет	82	45	37	55%
14	Тиша	131	101	30	77%
15	Шум	400	0	400	0%

Середній результат розпізнавання по усіх звуках складає 65%. Цей показник дещо знизився у порівнянні з розпізнаванням окремо взятих звуків. Це пояснюється тим, що іде накопичення середнього значення по тридцяти послідовних фреймах і у місцях де один звук закінчується а інший починається ці значення розмиваються. Також необхідно зазначити, що виникли деякі труднощі при розпізнаванні звуку пістолета та рушниці. Вони виявились доволі схожими. У місцях де є звук пістолета та рушниці програма розпізнала їх обох.

Після цього ми проведемо ще одне розпізнавання звуку але уже із додатковим фоновим шумом. Як фон ми візьмемо звуки природи (рис. 3.5).



Рисунок 3.5 — Форма сигналу фонового звуку природи

Зробимо накладання звуку природи на суміш звуків, які перед цим проходили розпізнавання. Результуючий звук буде мати вигляд як на рисунку 3.6.

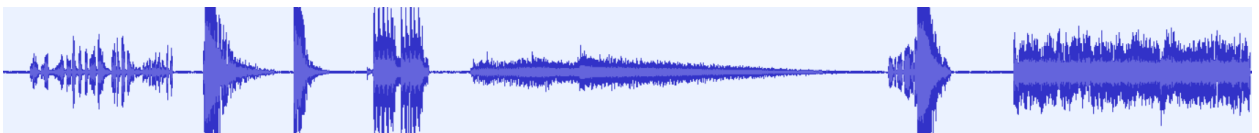


Рисунок 3.6 — Форма сигналу суміші багатьох звуків із шумом

Таблиця з результатами сегментації фреймів при наявності фонового шуму природи приведена нижче.

Таблиця 3.2 — Результати сегментації звуку із шумами

№	Назва звуку	Усього фреймів	Розпізнано	Не розпізнано	Відсоток розпізнавання
1	Тиша	52	29	23	56%
2	Спів пташки	310	182	128	59%
3	Тиша	53	34	19	64%
4	Рушниця	157	81	76	52%
5	Тиша	38	34	4	89%
6	Пістолет	90	26	64	29%
7	Тиша	70	47	23	67%
8	Автомат	120	93	27	78%
9	Тиша	90	37	53	41%
10	Трактор	792	621	171	78%
11	Тиша	84	55	29	65%
12	Спів пташки	62	57	5	92%

## Кінець таблиці 3.2

№	Назва звуку	Усього фреймів	Розпізнано	Не розпізнано	Відсоток розпізнавання
13	Пістолет	82	48	34	59%
14	Тиша	131	104	27	79%
15	Шум	400	0	400	0%

Якщо знову порахувати середній результат розпізнавання сегментованих звуків він складе 65%. Окремі звуки після накладання шуму почали розпізнаватись гірше а деякі (наприклад спів пташки та звук тиші) навпаки краще. Як результат середнє значення залишилось таким же. Щоправда відстані які відділяють шаблоні моделі одна від одної знизилась приблизно на 30%, проте вони усе одно залишились достатньо великими, щоб програма змогла їх відрізнити.

## 3.3 Інструкція використання програмного комплексу

Усі дані які потрібні для роботи програми та результати її роботи знаходяться а папці \data. Вона має знаходитись у тому ж каталозі що і виконуваний exe файл.

Відкривши даний каталог ми побачимо таку структуру папок та файлів:

- cps (каталог);
- gmm\_model (каталог);
- recognized\_sound (каталог);
- IdealValue.conf;
- Sound.wav.

У каталозі cps розташований файл cpsSoundCoef.txt. Цей файл містить набір спектральних коефіцієнтів які отримуються для звуку який потрібно розпізнати. У процесі роботи коефіцієнти з цього файлу не завантажуються. Цей файл має інформативний характер і у разі виникнення збоїв у роботі, ми можемо перевірити правильність їх знаходження.

У папці `gmm_model` розташовуються усі навчені шаблонні моделі GMM. Вони зберігаються у вигляді `txt` файлів. В процесі роботи програми назви моделей формуються з назв файлів. Тому їх потрібно правильно записувати і, в разі необхідності зміни назви шаблону, редагувати потрібно саме назви цих файлів.

Додаткові налаштування моделей, що завантажуються задаються в файлі `IdealValue.conf`. Фрагмент файлу налаштувань наведений нижче:

```
#base_silence4 = 15  
  
base_bird = 11.133245  
minFrame_base_bird = 50  
  
base_pistol = 7  
#minFrame_base_pistol = 30  
maxFrame_base_pistol = 120
```

Для того щоб задати еталоне значення для якоїсь шаблонної моделі потрібно просто прописати назву цієї моделі та через знак дорівнює вказати його. Значення має формат `double`, через це воно не обов'язково має бути цілим числом. Але важливо пам'ятати, що назва моделі має повністю збігатись з назвою шаблонної моделі яка зберігається в папці `gmm_model`.

Якщо потрібно задати обмеження на мінімальну чи максимальну послідовність фреймів, нам потрібно перед назвою моделі вказати префікс `minFrame_` чи `maxFrame_` відповідно. Ну і через знак дорівнює кількість цих фреймів. Тут уже значення має бути цілого типу та не повинно перевищувати 200. Також немає сенсу вказувати значення 1.

В конфігураційний файл можна вносити коментарі. Будь-яка стрічка, що починається з знаку «#» вважається коментарем. Так у прикладі конфігураційного файлу, що наведений вище, ми закоментували обмеження на мінімальну кількість фреймів. Але потрібно уникати введення символів кириличного алфавіту, тому що програма може працювати некоректно.

Каталог `recognized_sound`. Тут знаходиться всього один файл `recognizeResult.txt`. У ньому ми можемо побачити розширену інформацію про результати розпізнавання. Зокрема там представлені відстані вказаного фрейму до усіх моделей що присутні в базі.

Сам файл, який має проходити процедуру розпізнавання, знаходиться у папці `\data` і повинен називатись `Sound.wav`.

Після того як була підготовлена база програми і ми створили файл, який має проходити процедуру розпізнавання нам потрібно просто запустити виконуваний файл програми «`Recognize.exe`».

Результати її роботи відображаються в консолі. Спочатку іде процедура завантаження моделей. У консолі цей процес відображається наступним чином.

```
-----[Start load exist model]-----
Loading model: base_bird.txt
Result load: 1
...
Loading model: base_pistol.txt
Result load: 1
Loading model: base_silence4.txt
Result load: 1
-----[Finish load model. We have 8 model]-----
```

У випадку коли модельне завантажилась у полі `Result load` буде значення 0.

Після цього завантажуються додаткові значення шаблонних моделей із конфігураційного файлу:

```
-----[Start load ConfigValues]-----
(base_bird) = (11.133245)
(minFrame_base_bird) = (50)
(base_pistol) = (7)
```

```
(maxFrame_base_pistol) = (120)
```

```
(base_silence) = (18.143648)
```

```
-----[Finish load ConfigValues]-----
```

Уже після цього на екран виводяться результати розпізнавання:

```
-----[Start recognize]-----
```

```
[0] : {base_silence4(prob:7.93289, dist:2.06711) }
```

```
[1] : {base_silence4(prob:7.62437, dist:2.37563) }
```

```
[2] : {base_silence4(prob:7.78565, dist:2.21435) }
```

```
[3] : {base_silence4(prob:7.84183, dist:2.15817) }
```

```
[4] : {base_silence4(prob:8.17592, dist:1.82408) }
```

```
[5] : {base_silence4(prob:8.22604, dist:1.77396) }
```

```
...
```

```
-----[Finish recognize]-----
```

Як можна здогадатись, у результаті ми отримуємо відповідність кожного фрейма певній шаблонній моделі. Якщо ж фрейм не вдалось розпізнати буде виведено надпис NOT RECOGNIZE.

### 3.4 Реалізація розроблених програмних засобів у системі розпізнавання

При розробці програмного продукту була врахована можливість використання розроблених функцій розпізнавання у інших системах. Для цього система має бути написана на мові C++. Також на комп'ютері має бути встановлена бібліотека машинного навчання mlpack. На головному сайті продукту є інструкція по її встановленню. Для роботи mlpack також потрібно буде встановити додаткові бібліотеки такі як Boost та Armadillo.

Boost — це збірник різних бібліотек класів, які використовують функціональність мови C++ і надають зручний високорівневий інтерфейс для лаконічного кодування різних повсякденних задач програмування. Armadillo — це програмна бібліотека лінійної алгебри для мови програмування C++. Вона покликана забезпечити ефективні і оптимізовані

базові обчислення, водночас маючи простий і зручний інтерфейс, що суттєво пришвидшує процес розробки.

Після того як буде встановлено бібліотеку машинного навчання `mlpack` нам потрібно під'єднати її до програми яка буде використовувати функції розпізнавання, написані нами. Варто зазначити, що при їх написанні була використана бібліотека `mlpack` версії 3.0.4. Під'єднання бібліотеки до нового проекту у середовищі розробки Visual Studio виглядають наступним чином:

1) клацніть правою кнопкою миші по проекту і виберіть «Властивості», виберіть профіль відлагодження `x64`;

2) під `C/C++ > General > Additional Include Directories` додайте:

- `C:\boost\boost_1_66_0`;
- `C:\mlpack\armadillo-8.500.1\include`;
- `C:\mlpack\mlpack-3.1.1\build\include`;

3) під `Linker > Input > Additional Dependencies` додайте:

- `C:\mlpack\mlpack-3.1.1\build\Debug\mlpack.lib`;
- `C:\boost\boost_1_66_0\lib64-msvc-14.1\libboost_serialization-vc141-mt-gd-x64-1_66.lib`;
- `C:\boost\boost_1_66_0\lib64-msvc-14.1\libboost_program_options-vc141-mt-gd-x64-1_66.lib`;

4) у розділі `Build Events > Post-Build Event > Command Line` додайте:

- `xcopy /y "C:\mlpack\mlpack-3.1.1\build\Debug\mlpack.dll" $(OutDir)`;
- `xcopy /y "C:\mlpack\mlpack-3.1.1\packages\OpenBLAS.0.2.14.1\lib\native\bin\x64*.dll" $(OutDir)`.

Після цього потрібно додати код функцій для розпізнавання до свого проекту. А саме файл `sample-ml-app.cpp` у якому знаходяться усі потрібні функції. Код цього файлу знаходиться у додатку Б. Крім того потрібно іще підключити два файли `lrc.h` та `lrcfuncs.h` у яких знаходяться функції виділення з `wav` файлу кепстральних коефіцієнтів. Вони використовуються в головному файлі `sample-ml-app.cpp`.

Для того щоб мати змогу використовувати напрацювання з файлу `sample-ml-app.cpp` потрібно у своїй програмі створити дві колекції. Це список який буде зберігати усі наявні шаблонні моделі та список де будуть зберігатись класифіковані фрейми. Також потрібно створити матрицю у яку будуть завантажуватись кепстральні коефіцієнти. Додаються у програму вони таким чином:

```
using namespace std;
vector<MyGMMmodel> listExistGMMModel; //Вектор усіх моделей
vector<Frame> listFrameSound; //Вектор усіх фреймів
mat dataWavP; //Матриця кепстральних коефіцієнтів
```

Ділі створені колекції будуть передаватись у функції як параметри. У файлі наявні 4 функції необхідні для розпізнавання. Їх список наведений нижче:

```
int LoadModel(vector<MyGMMmodel> *listModel);
bool SetIdealValueToGMM(vector<MyGMMmodel> *listModel);
bool loadCPSfromWAV(string pathToFile, arma::mat& resultMat);
bool recognizeNext(mat next100Points, vector<MyGMMmodel>
&listExistGMMModel, vector<Frame> &listFrameSound, string pathToSave =
"data/recognized_sound/recognizeResult.txt").
```

Функція `LoadModel` використовується для завантаження в список `listModel` усіх наявних в базі моделей. Як параметр їй передається створений перед цим `vector<MyGMMmodel> listExistGMMModel`. Повертає функція кількість моделей які успішно були завантажені. Варто зазначити, що моделі мають знаходитись в каталозі `data/gmm_model`, який є відносним до виконуваного `exe` файлу.

Для завантаження додаткових обмежень на кожну модель, використовується функція `SetIdealValueToGMM`. Обмеження завантажуються з файлу `data/IdealValue.conf`. Формат файлу розглядався



перед цим у розділі 3.3. Як параметр їй передається створений перед цим `vector<MyGMMmodel> listExistGMMModel`.

Для перетворення звукового wav файлу в набір кепстральних коефіцієнтів та завантаження їх в матрицю використовується функція `loadCPSfromWAV`. Їй потрібно вказати назву файлу із якого будуть розраховуватись коефіцієнти та матрицю у яку вони будуть записані. У разі успішного завантаження повертається значення `true` в іншому випадку `false`.

Ну і для проведення розпізнавання самих фреймів потрібно використовувати функцію `recognizeNext`. У вигляді параметрів їй потрібно передати матрицю із кепстральними коефіцієнтами — `mat next100Points`, список із завантаженими шаблонними моделями — `vector<MyGMMmodel> &listExistGMMModel`, список із фреймами — `vector<Frame> &listFrameSound`, та назву файлу у який буде записуватись результат розпізнавання — `string pathToSave`. Останній параметр є необов'язковим. За замовчуванням результати зберігаються у файл `data/recognized_sound/recognizeResult.txt`.

## 4 ЕКОНОМІЧНА ЧАСТИНА

Будь-яка нова розробка потребує вкладення у неї ресурсів, як матеріальних так і часових. Відповідно, ці ресурси мають повернутись інвесторам або принести якість якісне покращення у житті суспільства. Тільки тоді вона може вважатись вдалою. Важливо визначити економічну доцільність досліджуваної розробки при впровадженні її у промислове виробництво. Переглянути існуючий на сьогоднішній момент ринок, та визначити чи зможе витримати інновація конкуренцію на ньому.

Саме цій проблематиці присвячений даний розділ магістерської кваліфікаційної роботи. Пропонується розбити розділ на чотири частини, які зображені на рисунку 4.1. Усі ці етапи дозволять виявити на ранній стадії проекту комерційний потенціал розробки або, навпаки, її комерційну безперспективність, та зробити висновок щодо доцільності або недоцільності подальшого розвитку.

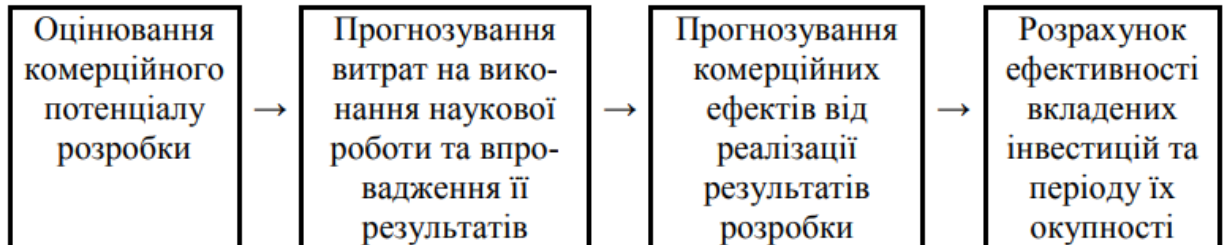


Рисунок 4.1 — Складові економічної частини магістерської кваліфікаційної роботи

### 4.1 Оцінювання комерційного потенціалу розробки

Для того щоб визначити економічну доцільність нашої розробки та переконатись в її якійській користі для суспільства, яка зрештою і буде індикатором успішності нашого продукту, потрібно провести початкове оцінювання комерційного потенціалу розробки.

Для такого оцінювання буде застосована таблиця із критеріями ступеню якості оцінюваного продукту, яка зображена нижче у таблиці 4.1. У

ній технічна розробка оцінюється із різних сторін, таких як: технічна здатність концепції, практична здійсненність, ринкові переваги, недоліки та перспективи. Самі оцінки розробці будуть виставлені спеціально відібраними незалежними експертами, які добре орієнтуються у проблематиці та тематиці досліджуваного явища. Це дозволить доволі об'єктивно та точно оцінити реальні перспективи розробки.

Хотілося б відмітити окремо сильні та слабкі сторони розроблюваного продукту, а саме: інформаційної технології аудіодетекції звукових сигналів немовленнєвого походження. Першим та вагомим плюсом є те, що розроблюваний продукт є програмним забезпеченням. Як наслідок, для його реалізації не потрібно залучення великих інвестицій. Ця особливість сприятиме привабленню потенційних інвесторів або ж і зовсім дозволить реалізувати ідею власними силами. Проте, як і будь-який програмний продукт він потребує захисту. Нажаль саме такий вид розробки може швидко втратити конкурентність на ринку, адже одержавши код програми або розгадавши алгоритм її роботи, не складе великої проблеми зібрати аналог подібної системи.

В умовах бойових дій, які нажалі сьогодні тривають в нашій країні, дуже доречною виглядає розробка систем, які б давали змогу розпізнавати постріли та переміщення воєнної техніки і визначати їх джерело. На сьогодні на ринку таких систем є дуже обмежена кількість і вартують вони дуже дорого. Досліджувана розробка може бути використана при конструюванні таких систем. Вона можливо і є не повною, готовою до використання з коробки, системою, проте вартує значно менших коштів.

Для реалізації розробки на ринку мною пропонується використати механізм ліцензійної угоди. Продукт розроблений таким чином, що він може бути легко інтегрований у будь-які системи розпізнавання (системи розпізнавання пострілів, надзвичайних ситуацій та інші подібні системи). Це дозволить підняти інтерес до нього і отримувати прибуток шляхом ліцензійних відрахувань.

Таблиця 4.1 — Критерії оцінювання комерційного потенціалу розробки

Бали (за 5-ти бальною шкалою)					
Кри- терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи:					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів не має
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фін. ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фін. відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військ. пром. комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використ. у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років.	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років.
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

В якості експертів, які будуть оцінювати розробку, було вирішено залучити керівника магістерської роботи Ткаченка О. М., який має чималий досвід у роботі зі звуками, а також заступника завідувача кафедри ОТ Крупельницького Л.В., який по сумісництву також є завідувача НТЦ "Аналого-цифрові системи" ВНТУ. Результати оцінювання наведено нижче у таблиці 4.2.

Таблиця 4.2 — Результати оцінювання комерційного потенціалу розробки інформаційної системи

Критерії	Прізвище, ініціали, посада експерта	
	Ткаченко О. М. к.т.н., доцент, науковий керівник	Крупельницький Л.В. к.т.н., доцент, заступник завідувача кафедри ОТ
	Бали, виставлені експертами:	
1	3	3
2	3	2
3	4	3
4	1	1
5	2	3
6	2	1
7	2	3
8	3	3
9	1	3
10	4	3
11	4	3
12	4	3
Сумма балів	33	31
Середньоарифметична сума балів СБ	$\overline{СБ} = \frac{\sum_1^2 СБ_i}{2} = \frac{33 + 31}{2} = 32$	

Як можна побачити із результатів таблиці 4.2 середній бал виставлений експертами складає 32. Відповідно, скориставшись таблицею 4.3, яка визначає рівень комерційного потенціалу розробки, можна зробити висновок що він є вище середнього.

Таблиця 4.3 — Рівні комерційного потенціалу розробки інформаційної системи

Середньоарифметична сума балів $\overline{СБ}$ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Отже, перспективи розроблюваного додатку є доволі хорошими. Серед основних переваг, які були виділені експертами можна відмітити доволі низьку ціну реалізації, відсутність великої конкуренції та можливість швидкого втілення проекту в життя.

#### 4.2 Прогнозування витрат на виконання науково-дослідної та конструкторсько-технологічної роботи

Для розрахунку прогнозованих витрат на виконання дослідно-конструкторської та конструкторсько-технологічної роботи потрібно пройти 3 основних етапи:

- 1-й етап є розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;
- 2-й етап є розрахунок загальних витрат на виконання даної роботи;
- 3-й етап є прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

#### 4.2.1 Розрахунок витрат, які безпосередньо стосуються виконавців даного розділу НДДКР

Для того щоб розрахувати основну заробітну плату кожного з розробників, за умови що вони працюють в наукових установах бюджетної сфери скористаємося формулою 4.1:

$$Z_o = \frac{M}{T_p} \cdot t \text{ (грн.)}, \quad (4.1)$$

де  $M$  — місячний посадовий оклад конкретного розробника, грн;

$T_p$  — число робочих днів в місяці;

$t$  — число днів роботи розробника.

Так як, над створенням даної розробки працювало безпосередньо дві людини: керівник роботи та студент (інженер), потрібно для кожного з них розрахувати основну заробітну плату  $Z_o$ .

Заробітна плата керівника

$$Z_o = \frac{9000,00}{22} \cdot 20 = 8181,82 \text{ (грн.)},$$

заробітна плата інженера

$$Z_o = \frac{6000,00}{22} \cdot 50 = 13636,36 \text{ (грн.)}.$$

Результати розрахунків занесемо в таблицю 4.4.

Таблиця 4.4 — Заробітна плата

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн
1 Керівник	9000	409,09	20	8181,82
2 Інженер	6000	272,72	50	13636,36
Всього:				21818,18

Зробивши підрахунки ми в кінцевому вигляді отримали суму 21818,18 грн.

Так як досліджувана розробка є програмним продуктом то вона не вимагає додаткової допомоги робітників. Тому можна задати  $Z_p = 0$ .

Додаткова заробітна плата  $Z_d$  всіх розробників та робітників, які брали участь у виконанні даної НДДКР, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = Z_o \cdot 0,11 = 21818,18 \cdot 0,11 = 2399,98 \text{ (грн)}.$$

Нарахування на заробітну плату  $H_{зп}$  розробників та робітників, які брали участь у виконанні даної НДДКР, розраховується формулою 4.2:

$$H_{зп} = (Z_o + Z_p + Z_d) \cdot \frac{\beta}{100} \text{ (грн)}, \quad (4.2)$$

де,  $Z_o$  — основна заробітна плата розробників, грн;

$Z_p$  — основна заробітна плата робітників, грн;

$Z_d$  — додаткова заробітна плата всіх розробників та робітників, грн.;

$\beta$  — ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, з 01.01.2016 року ставка ЄСВ знижена до 22% (для всіх категорій платників).

Тоді отримуємо наступне:

$$H_{зп} = (21818,18 + 0 + 2399,98) \cdot 0,22 = 5327,99 \text{ (грн)}.$$

Амортизація обладнання, комп'ютера, пристроїв та приміщень  $A$ , які використовувались під час виконання НДДКР розраховується формулою 4.3:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12} \text{ (грн)}, \quad (4.3)$$

де,  $Ц$  — загальна балансова вартість всього обладнання, комп'ютера, пристроїв та приміщення, тощо, що використовувались для виконання НДДКР, грн.;



$H_a$  — річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що  $H_a=(10\dots25)\%$ ;

$T$  — термін, використання обладнання, приміщень тощо, місяці.

Розраховані значення амортизаційних відрахувань наведено у нижче.

Таблиця 4.5 — Амортизація основного обладнання

Найменування обладнання, приміщень	Балансова вартість, грн	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
Офіс	200000,00	15	2	5000,00
Комп'ютери	24000,00	15	2	600,00
Всього:				5600,00

Так як кінцевим продуктом який ми розробляємо є програмний засіб, він не несе у собі вагомих витрат на комплектуючі  $K$ . Проте сюди можна віднести канцтовари, які розробники використовують у процесі своєї роботи. Розраховуються витрати на комплектуючі за формулою 4.4:

$$K = \sum_i^n H_i \cdot C_i \cdot K_i \cdot (\text{грн}), \quad (4.4)$$

де,  $H_i$  — кількість комплектуючих  $i$ -го виду, шт.;

$C_i$  — ціна комплектуючих  $i$ -го виду, грн;

$K_i$  — коефіцієнт транспортних витрат,  $K_i = (1,1\dots1,15)$ ;

$n$  — кількість видів комплектуючих.

Результати розрахунків занесено до таблиці 4.6.

Витрати на силову електроенергію  $V_e$  розраховуються за формулою 4.5:

$$V_e = V \cdot P \cdot \Phi \cdot K_p \quad (\text{грн}), \quad (4.5)$$

де,  $V$  — вартість 1 кВт-години електроенергії (3 грн/кВт);

$P$  — установлена потужність обладнання, кВт;

$\Phi$  — фактична кількість годин роботи обладнання;

$K_p$ — коефіцієнт використання потужності 0,8.

Таблиця 4.6 — Комплектуючі та їх вартість

№ п/п	Найменування товару	Кількість шт.	Ціна за шт., грн	Сума,грн.
1	Зошит	2	10,00	20,00
2	Ручка	1	25,00	25,00
3	Олівець	2	7,50	15,00
4	Гумка	1	5,00	5,00
Всього:				65,00
Разом з транспортними витратами		$65 \cdot 1,15 = 74,75$		

Підставивши усі потрібні значення у формулу 4.5 отримуємо ціну за електроенергію:

$$V_e = 3 \cdot 2 \cdot 240 \cdot 0,8 = 1152 \text{ (грн)}$$

Інші витрати  $V_{ін}$  охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Як правило ця сума має знаходитись в межах від 100% до 300% від суми заробітної плати розробників та робітників. Ми візьмемо її за 150%.

$$V_{ін} = 1,5 \cdot (21818,18 + 0) = 32727,27 \text{ (грн)}$$

Сума всіх попередніх витрат дає змогу оцінити витрати, які безпосередньо стосуються виконавців даного розділу роботи В.

$$V = 21818,18 + 2399,98 + 5327,99 + 5600 + 74,75 + 1152 + 32727,27 = 69100,17 \text{ (грн)}$$

#### 4.2.2 Розрахунок загальних витрат на виконання роботи

Розрахунок загальних витрат на виконання роботи здійснюється у тому випадку якщо ми робимо лише частину НДДКР. Проте, робота була зроблена у повному обсязі. Тому  $V_{\text{заг}}$  у нашому випадку не зміниться і буде рівний  $V$ . Тому можна зробити висновок що  $V_{\text{заг}}$  (загальна вартість виконання НДДКР) буде складати 69100,17 грн.

#### 4.2.3 Прогнозування загальних витрат на виконання та впровадження результатів роботи

Прогнозування загальних витрат  $ZB$  на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою 4.6:

$$ZB = \frac{V_{\text{заг}}}{\beta} \text{ (грн)}, \quad (4.6)$$

де,  $\beta$  — коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Створена розробка уже працює, тому вона не має суттєво змінюватись перед випуском на ринок. Проте можливо буде потреба у проведенні додаткових досліджень, експериментів та удосконалень. Тому можна вважати що продукт готовий на 60% . Підставивши ці данні в формулу отримаємо наступне:

$$ZB = \frac{69100,17}{0,6} = 115166,95 \text{ (грн)}$$

#### 4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Кожний інвестор прагне якомога об'єктивніше оцінити майбутні результати від реалізації інвестиційного проекту, оскільки проекти, які не дадуть відповідного ефекту, безперспективні, і ніхто не вкладатиме в них кошти. Звісно, провести точні розрахунки в умовах ринку який динамічно змінюється дуже важкою. Також потрібно враховувати непередбачувані труднощі, які можуть виникнути у процесі впровадження виробу у

виробництво або його експлуатації. Проте ми можемо приблизно спрогнозувати, яку вигоду можна отримати від впровадження нашої розробки у виробництво.

Так, приблизно, для розробки технології аудіодетекції звукових сигналів немовленнєвого походження нам знадобиться 6 місяців роботи. Після цього прогнозується основний прибуток протягом трьох років продажу ліцензійних копій.

Для того щоб розрахувати збільшення чистого прибутку підприємства  $\Delta\Pi_i$  для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, скористаємось формулою 4.7:

$$\Delta\Pi_i = \sum_1^n (\Delta C_0 \cdot N + C_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right) \text{ (грн)}, \quad (4.7)$$

де,  $\Delta C_0$  — покращення основного оціночного показника від впровадження результатів розробки у даному році. Зазвичай таким показником може бути ціна одиниці нової розробки;

$N$  — основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

$C_0$  — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

$\Delta N$  — покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$n$  — кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

$\lambda$  — коефіцієнт, який враховує сплату податку на додану вартість. У 2020 році ставка податку на додану вартість дорівнює 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  — коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати  $\rho = 0,2 \dots 0,3$ ;

$v$  — ставка податку на прибуток, в 2020 році  $v = 18\%..$

Нехай в результаті впровадження якісних покращень в роботу програмного забезпечення вдалось підвищити ціну його реалізації на 1000 грн. Початкова ціна продукту подібного типу може складати 3000 грн за шт. До впровадження наукової розробки реалізація продукції складала 0 шт. Прогнозується, що за перший рік буде продано 100 екземплярів подібного програмного продукту, другого року кількість збільшиться на 400 шт. і третього — на 500 шт.

Для розрахунку прогнозованого збільшення чистого прибутку підприємства від впровадження результатів наукової розробки упродовж наступних 3 років відносно базового скористаємося наведеною вище формулою.

Так для першого року реалізації збільшення чистого прибутку  $\Delta\Pi_1$  складе:

$$\Delta\Pi_1 = [1000 \cdot 0 + (3000 + 1000) \cdot 100] \cdot 0,8333 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 68330,60 \text{ грн.}$$

Для другого року  $\Delta\Pi_2$ :

$$\Delta\Pi_2 = [1000 \cdot 0 + (3000 + 1000) \cdot (100 + 400)] \cdot 0,8333 \cdot 0,25 \cdot 0,82 = 341653 \text{ грн.}$$

Для третього року  $\Delta\Pi_3$ :

$$\Delta\Pi_3 = [0 + (3000 + 1000) \cdot (100 + 400 + 500)] \cdot 0,8333 \cdot 0,25 \cdot 0,82 = 683306 \text{ грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Якщо ми прогнозуємо позитивний комерційний ефект від впровадження досліджуваної розробки, це ще не є гарантією того, що інвестор на 100 відсотків зацікавлений у його реалізації. Він погодиться вкладати кошти у реалізацію даної наукової розробки тільки за певних умов.

Так, основними індикаторами успішності інвестиційного проекту є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності. У даному розділі ми розрахуємо усі ці показники.

Для початку нам потрібно розрахувати теперішню вартість інвестицій  $PV$ . Будемо вважати що  $PV = ЗВ$ , прогнозована величина загальних витрат, розрахована нами раніше. Вона складає 115166,95 грн.

Очікуване збільшення прибутку нами також було попередньо розраховано. Так прогнозовані прибутки за перший рік  $\Delta\Pi_1 = 68330,60$  грн.,  $\Delta\Pi_2 = 341653$  грн. та  $\Delta\Pi_3 = 683306$  грн..

Для спрощення подальших розрахунків побудуємо вісь часу, на яку нанесемо всі платежі, що мають місце під час виконання науково-дослідної роботи та впровадження її результатів. Дана схема відображена на рисунку 4.2.

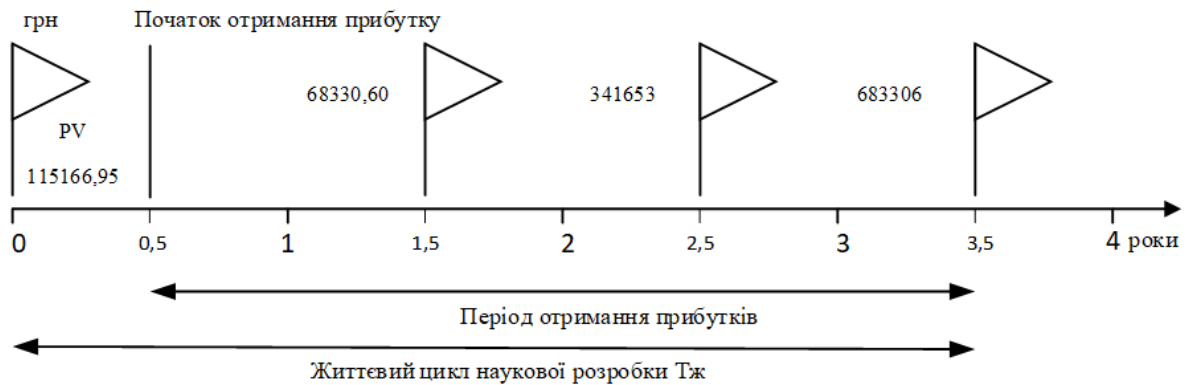


Рисунок 4.2 — Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Щоб розрахувати абсолютну ефективність вкладених інвестицій  $E_{абс}$  скористаємося формулою 4.8.

$$E_{абс} = (ПП - PV) \text{ (грн)}, \quad (4.8)$$

де, ПП — приведена вартість всіх чистих прибутків, що їх отримає організація від реалізації результатів наукової розробки, грн;

$PV$  — теперішня вартість інвестицій.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за наведеною нижче формулою 4.9:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t} \text{ (грн)}, \quad (4.9)$$

де,  $\Delta\Pi_i$  — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

$T$  — період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

$\tau$  — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

$t$  — період часу (в роках) від моменту отримання чистого прибутку до точки «0».

$$\begin{aligned} ПП &= \frac{68330,60}{(1+0,1)^{1,5}} + \frac{341653}{(1+0,1)^{2,5}} + \frac{683306}{(1+0,1)^{3,5}} = \frac{68330,60}{1,15} + \frac{341653}{1,27} + \frac{683306}{1,396} = \\ &= 59417,91 + 269018,11 + 489474,21 = 817910,23 \text{ грн.} \end{aligned}$$

Далі потрібно обрахувати  $E_{abc}$ :

$$E_{abc} = (817\,910,23 - 115\,166,95) = 702743,28 \text{ (грн)},$$

Оскільки  $E_{abc} > 0$ , це означає що результат від проведення наукових досліджень та їх впровадження принесе прибуток інвестору.

Далі розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_B$ . Скористаємося формулою 4.10:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.10)$$

де,  $E_{abc}$  — абсолютна ефективність вкладених інвестицій, грн;

$T_{ж}$  — життєвий цикл наукової розробки;

$PV$  — теперішня вартість інвестицій, грн.

$$E_B = \sqrt[3,5]{1 + \frac{702743,28}{115\,166,95}} - 1 \approx \sqrt[3,5]{7,102} - 1 \approx 0,75,$$

Потрібно розрахувати мінімальну ставку дисконтування, формула 4.11:

$$\tau = d + f, \quad (4.11)$$

де,  $d$  — середньозважена ставка за депозитними операціями в комерційних банках; в 2020 році в Україні  $d=0.16$ ;

$f$  — показник, що характеризує ризикованість вкладень; зазвичай величина  $f=(0,05\dots 0,1)$ .

$$\tau = 0,16 + 0,1 = 0,26 .$$

Порівнюючи ці значення між собою, можна дійти висновку що  $E_B > \tau$ , а саме  $0,75 > 0,26$ . Звідси слідує, що інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

І на сам кінець, розрахуємо термін окупності вкладених в реалізацію наукового проекту інвестицій. Він розраховується формулою 4.12:

$$T_{ок} = \frac{1}{E_B}, \quad (4.12)$$

У нашому випадку розрахунки матимуть наступний вигляд:

$$T_{ок} = \frac{1}{0,75} = 1,3(3) \text{ роки}$$

Термін окупності в 1 рік і 4 місяці є дуже хорошим, тому фінансування даної наукової розробки є доцільним.



## ВИСНОВКИ

1 У першому розділі було розглянуто фізичну природу звукового сигналу. Наведені основні поняття про звук та його основні характеристики. Адже для того щоб проводити розпізнавання потрібно добре орієнтуватись у природі цього явища. Також був проведений огляд основних характеристик акустичних сигналів, які можуть бути використані для їх класифікації.

На сьогоднішній день існує досить багато алгоритмів класифікації які упродовж багатьох років доводять свою ефективність. Деякі з них були детальніше розглянуті в підрозділі 1.4. Для класифікації звукового сигналу було обрано модель змішаних Гаусових розподілень, оскільки вона найбільш вдало підходить для поставленої задачі. Також був проведений огляд уже існуючих технологій розпізнавання звуків, та описані їх основні переваги та недоліки.

2 В другому розділі було розглянуто принципи перетворення аналогового сигналу, яким описуються звукові коливання в цифрову форму. Так як, сигнали які мають розпізнаватись, є занадто складними на усьому проміжку нами було вирішено розбивати сигнали на окремі фрейми. Цей спосіб є більш зручним для подальшої роботи з ними. Так як на маленьких ділянках сигнали є стаціонарними процесом це дозволяє нам отримати їх кепстральні коефіцієнти. На основі кепстральних коефіцієнтів будуються шаблонні моделі. Для класифікації моделей було вирішено використовувати змішані гаусівські моделі.

Для розробки програми була взята мова C++. Вона є ефективною в плані швидкодії та зручною для написання програм, що підтверджується уже протягом багатьох років її популярністю. Важливим фактором при обранні мови також став той факт, що на ній написана бібліотека машинного навчання `mlpack`. Ця бібліотека містить класи для роботи зі змішаними гаусівськими моделями, які активно використовуються в програмі. Були розроблені основні сутності які описують об'єкти які розпізнаються та

засоби за допомогою яких відбувається розпізнавання. Для аналізу звуків був розроблений алгоритм розпізнавання, який реалізований засобами мови C++.

3 В третьому розділі були розглянуті основні характеристики розробленої інформаційної технології для аудіодетекції звукових сигналів. Система нараховує вісім тестових шаблонних моделі для розпізнавання звуків. Це число може бути збільшене. Для цього потрібно провести навчання додаткових моделей та помістити їх у базу звуків (відповідний каталог прогами). Звуковий сингал, що проходить процедуру розпізнавання, має бути у форматі wav. Основні вимоги до нього розглянуті у першому підрозділі.

Також було проведено тестове розпізнавання звуків та проаналізовано результати. Для перевірки правильності роботи системи ми взяли звуки для яких наявні моделі у нашій базі. На першому етапі ці звуки розпізнавались поодинці. Результати можна вважати успішними, так як кожен звук був вірно віднесена до свого навченого шаблону. Кількість фреймів які були розпізнані також була доволі високою. На другому етапі проводилось розпізнавання звуків які були об'єднані у один великий файл. Результати дещо погіршились, проте усі звуки були правильно віднесені до своїх моделей. Ну і наостанці, ми перевірили можливість розпізнавання звуків при наявності шумів. На файл суміші багатьох звуків був накладений звук природи. Після чого він знову пройшов процедуру розпізнавання. Кількість розпізнаних фреймів по окремим звукам дещо зменшилась. Проте усі звуки були вірно розпізнані.

Була створена інструкція користувача, яка допоможе людині, незнайомій з програмою, швидко розібратись з усіма її особливостями. Там описана структура каталогів та усі файли необхідні для роботи програми. Також наводиться приклад налаштування конфігурації шаблонних моделей та приклад виводу результатів розпізнавання. Окрім того була описана можливість інтеграції технології аудіодетекції звукових сигналів до інших програмних засобів.

4 В четвертому розділі була розрахована економічна доцільність досліджуваної розробки при впровадженні її у промислове виробництво. Була відібрана група експертів, якою був оцінений комерційний потенціал розробки. Вона позитивно оцінила перспективи досліджуваної новачки та вказала, що її комерційний потенціал є вище середнього. Це свідчить про доволі хороші перспективи та можливість отримання прибутку. Серед основних переваг, які були виділені експертами можна відмітити доволі низьку ціну реалізації, відсутність великої конкуренції та можливість швидкого втілення проекту в життя.

У другій частині, мною були проведені розрахунки витрат на виконання наукової роботи та впровадження її результатів. Як виявилось загальні витрати на розробку програмного продукту складатимуть 115166,95 грн.

Для виконання даної наукової роботи та впровадження її результатів необхідно затратити 6 місяців. Перший комерційні ефекти від реалізації результатів розробки проявиться через півтора роки і буде складати суму 68330,60 грн . Збільшення чистого прибутку для 2-го року складе 341653 грн, а для 3-го 683306 грн.

В завершення, була визначена абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності. Так, абсолютна ефективність від реалізації проекту складе 702743,28 грн. Це показує що розробка у будь-якому разі буде прибутковою. Відносна ефективність інвестицій складає 75%. Даний фактор має привабити потенційних інвесторів. Період окупності складає 1 рік і 4 місяці, що є дуже непоганим показником. Усі ці фактори свідчать про економічну доцільність розробки програмного продукту.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Тютюнник Я. О., Чорний Д. С., Ткаченко О. М. Програмні засоби для аудіодетекції звукових сигналів немовленнєвого походження. Тези: 2019 рік : матеріали Всеукр. наук.-практ. конф. Молодь в науці, 11–30 травня. 2019 р. Вінниця : ВНТУ, 2019. С. 1–2.
2. Чорний Д. С., Тютюнник Я. О., Ткаченко О. М. Програмні засоби формування словника параметрів звукових сигналів для аудіодетекції. Тези: 2019 рік : матеріали Всеукр. наук.-практ. конф. Молодь в науці, 11–30 травня. 2019 р. Вінниця : ВНТУ, 2019. С. 1–2.
3. Тютюнник Я. О., Ткаченко О. М. Інформаційна технологія для аудіодетекції звукових сигналів немовленнєвого походження. Тези: 2021 рік : матеріали Всеукр. наук.-практ. конф. Молодь в науці, 1–14 травня. 2021 р. Вінниця : ВНТУ, 2021. С. 1–2.
4. Грийо Тукало О. Ф. Быстрый поиск векторов при сжатии речевых сигналов : монография / О. Ф. Грийо Тукало, А. Н. Ткаченко // LAP LAMBERT Academic Publishing. – 2014. – 108 с. – ISBN 978-3-659-50369-6.
5. Hastie T. The elements of statistical learning / T. Hastie, R. Tibshirani, F. Jerome. – Stanford, California: Springer, 2008. – 764 с. – (Second Edition).
6. Метод дельта-ущільнення мовних сигналів / О.М. Ткаченко, О.Д. Феферман, С.В. Хрущак // Інформаційні технології та комп'ютерна інженерія. Вінниця: УНІВЕРСУМ-Вінниця. - 2008.- № 1(11). – С. 8-13.
7. Ткаченко О.М. Ідентифікація музичного твору за фрагментом / О.М.Ткаченко, О.Ф. Грийо Тукало, Ю. Л. Далекий // Міжнародна науково–практична конференція "Інформаційні технології та комп'ютерна інженерія". – С.256–257.– Вінниця. – 2014.– 2 с.
8. Müller A. Introduction to Machine Learning with Python / A. Müller, S. Guido. – Boston: O'Reilly Media, Inc, 2017. – 369 с. – (Jupyter).
9. Murphy K. Machine Learning: A Probabilistic Perspective / Kevin Murphy. – Massachusetts: Massachusetts Institute of Technology, 2012. – 1098 с.
10. Кузюрин Н. Н. Эффективные алгоритмы и сложность вычислений / Н. Н.

Кузюрин, С. А. Фомин. – Москва, 2019. – 345 с.

11. Моделі гаусових сумішей (GMM) [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/gaussian-mixture-833c88587c7f>.

12. EM — масштабований алгоритм кластеризації [Електронний ресурс] – Режим доступу до ресурсу: <https://basegroup.ru/community/articles/em>.

13. Рабинер Л. Р. Цифровая обработка речевых сигналов / Л. Р. Рабинер, Р. В. Шафер. – Москва: Радио и связь, 1981. – 496 с.

14. Страуструп Б. Дизайн и эволюция C++ / Берн Страуструп. – Москва: ДМК Пресс, 2006. – 448 с.

15. Gaussian Mixture Model [Електронний ресурс] – Режим доступу до ресурсу: <https://brilliant.org/wiki/gaussian-mixture-model/>.

16. Николенко С. И. Глубокое обучение. Погружение в мир нейронных сетей. / С. И. Николенко, А. А. Кадурин, Е. О. Питер: ДМК Пресс, 2018. – 480 с

17. Supervised Machine Learning: Classification [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/supervised-machine-learning-classification-5e685fe18a6d>.

18. Ben-David S. Understanding Machine Learning: From Theory to Algorithms / Ben-David. – London: Cambridge University, 2014. – 410 с.

19. Duda R. O. Pattern Classification / Richard Duda. – New York: Wiley-Interscience, 2004. – 816 с.

20. Grus J. Data Science from Scratch / Joel Grus. – New York: O'Reilly Media, 2015. – 330 с.

21. Classification Algorithms in Machine Learning... [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/datadriveninvestor/classification-algorithms-in-machine-learning-85c0ab65ff4>.

22. Gaussian mixture models¶ [Електронний ресурс] – Режим доступу до ресурсу: <https://scikit-learn.org/stable/modules/mixture.html>.

23. In Depth: Gaussian Mixture Models [Електронний ресурс] – Режим доступу до ресурсу: <https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html>.

24. Разработка классических приложений для Windows [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.microsoft.com/ru-ru/cpp/windows/desktop-applications-visual-cpp?view=vs-2019>.
25. Солтер Н. А. Язык программирования C++ для профессионалов / Н. А. Солтер, С. Д. Клепер. – Москва: ДМК Пресс, 2006. – 912 с.
26. Mlpack. flexible C++ machine learning library [Электронный ресурс] – Режим доступа до ресурсу: <https://www.mlpack.org/>.
27. Система виявлення пострілів [Электронный ресурс] – Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Gunfire\\_locator](https://en.wikipedia.org/wiki/Gunfire_locator).
28. Поннятя про звукові хвилі [Электронный ресурс] – Режим доступа до ресурсу: <https://lektsii.com/1-103106.html>.
29. Лекції з фізики для бакалаврів. Коливання і хвилі [Электронный ресурс] – Режим доступа до ресурсу: <http://physics.zfftt.kpi.ua/mod/book/tool/print/index.php?id=299>.
30. Гудфеллоу Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль. – Москва: ДМК Пресс, 2018. – 652 с.