Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи розпізнавання голосу для керування системою розумний будинок

(тема)

Виконав:

студент II курсу, групи СПм-22-6

Міхайлов І. О.
(прізвище, ініціали)

Спеціальність

123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

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

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: доц. Ляшенко О. С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

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

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Міхайлову Іллі Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи розпізнавання голосу для керування системою розумний будинок

затверджена наказом по університету від " 01 " квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024 р.

3. Вхідні дані до роботи 1)онлайн та оффлайн методи; 2)порівнювальні характеристики – точність, затримка, надійність; 3)програмний застосунок мовою Python

4. Перелік питань, що потрібно опрацювати у роботі

1) огляд популярних систем розумний будинок з голосовим управлінням;
2) вибір та обґрунтування методики та засобів дослідження;
3) програмна реалізація методів;
4) проведення експериментальних досліджень;
5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд популярних систем розумний будинок	02.04.24-08.04.24	
	з голосовим управлінням		
2	Вибір та обґрунтування методики та засобів	09.04.24-12.04.24	
	дослідження		
3	Вибір та обґрунтування методів розпізнавання	13.04.24-19.04.24	
4	Програмна реалізація методів	20.04.24-09.05.24	
5	Проведення експериментальних досліджень	10.05.24-23.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.24-03.06.24	
7	Подання кваліфікаційної роботи керівникові та	04.06.24-07.06.24	
	Її попередній захист		
8	Подання кваліфікаційної роботи на	08.06.24-12.06.24	
	рецензування		

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Ляшенко О. С.

(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 78 с., 19 рис., 12 табл., 5 дод., 12 джерел.

РОЗПІЗНАВАННЯ ГОЛОСУ, РОЗУМНИЙ БУДИНОК, МЕТОД, МОДЕЛЬ, ПОРІВНЯННЯ, OFFLINE, ONLINE.

Метою кваліфікаційної роботи є аналіз методів розпізнавання голосу, наочне порівняння, визначення найбільш підходящих для розумного будинку та розробка рекомендацій щодо їх реалізації.

У ході виконання кваліфікаційної роботи було розглянуто сучасні популярні системи розумного будинку, що використовують голосове управління, їх позитвині та негативні сторони під час використання.

Розглянуто існуючі методи розпізнавання голосу, досліджено алгоритми роботи та досліджено статті відомих видань для кращого розуміння суті методов. У процесі аналізу було відмічено два найперспективніших методи та обрано для детального дослідження.

У результаті було вирішено програмно реалізувати використання цих методів. Розроблені застосунки мають зрозумілий, зручний для користувача інтерфейс, надають можливість розпізнання голосу. Використовуючи розроблені застосунки було проведено експериментальні дослідження, в ході якого за технічними характеристиками було порівняно два методи.

Результати експериментального дослідження було записано у вигляді таблиць, а відповідні формули та розрахунки з результатами записано у форматі тексту. Завдяки отриманним результатам було підведено підсумки щодо кожного з методів та надано рекомендації щодо їх доцільного використання.

ABSTRACT

Master’s thesis:: 78 pages, 19 figures, 12 tables, 5 appendices, 12 sources.

VOICE RECOGNITION, SMART HOME, METHOD, MODEL, COMPARISON, OFFLINE, ONLINE.

The major goal of this thesis is the analysis of voice recognition methods, a visual comparison, the determination of the most suitable for a smart home and the development of recommendations for their implementation.

In the course of the qualification work, modern popular smart home systems that use voice control, their positive and negative sides during use were considered.

Existing methods of voice recognition were considered, work algorithms were studied, and articles from well-known publications were studied for a better understanding of the essence of the methods. In the process of analysis, the two most promising methods were noted and selected for detailed study.

As a result, it was decided to programmatically implement the use of these methods. The developed applications have a clear, user-friendly interface, provide voice recognition. Using the developed applications, experimental studies were conducted, during which two methods were compared according to technical characteristics.

The results of the experimental study were recorded in the form of tables, and the corresponding formulas and calculations with the results were recorded in text format. Thanks to the obtained results, conclusions were drawn about each of the methods and recommendations were given regarding their appropriate use.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ГУ – голосове управління

РБ – розумний будинок

E2E-ASR – наскрізна обробка мовлення (англ., End-to-End Speech Processing)

SR – speech recognition

ВСТУП

У цій кваліфікаційній роботі розглядаються методи та методологія розпізнавання голосу для управління системою «Розумний дім». Ця тема була обрана через недостатню увагу до потреб користувачів з обмеженими можливостями в сучасних системах. Хоча компанії, такі як Google, Apple та Amazon, надають рішення для розумного будинку, їх системи не завжди зручні для людей з вадами мови або координації рухів.

Поточні системи розумного будинку можуть розпізнавати різні ситуації в будинку і реагувати на них завдяки вбудованим алгоритмам. Наприклад, при несанкціонованому відкритті вікна спрацьовує датчик і система активує тривогу. Основними джерелами даних для таких систем є датчики, розміщені по всьому будинку.

Крім датчиків, у систему можна інтегрувати мікрофон для керування функціями будинку голосом. Наприклад, користувач може сказати "Увімкни режим користувач у кімнаті", і система виконає заздалегідь налаштовані дії. Однак це зручно лише для користувачів без проблем із мовою чи координацією рухів.

Більшість досліджень технологій розумного будинку зосереджені на інженерних аспектах, дизайні та охороні здоров'я, у той час як зручність користувачів часто залишається осторонь. Дослідження показують, що технології розумного будинку можуть використовуватися для безпеки, енергозбереження та інших цілей, а також для розваг. Наприклад, пристрої Amazon Alexa реагують на команди, від дзвінків та повідомлень до відтворення музики. Дверні дзвінки підвищують безпеку, а розумні лампочки заощаджують енергію завдяки своїй ефективності.

На підставі досліджень існуючих систем, їх переваг та мінусів, було вирішено роззглянути методи розпізнавання голосу, порівняти їх технічні параметри та аспекти роботи та (за можливістю) розробити систему, яка може

інтегруватися з існуючими системами розумного будинку та надавати допомогу користувачам з обмеженими можливостями, що робить запропоноване рішення актуальним. Реалізувати це планується з використанням розглянутих методів голосового управління та програмного забезпечення мовою Python, а також за допомогою зовнішніх пристроїв, таких як мікрофон, камера та динамік.

1 АНАЛІЗ ЕВОЛЮЦІЇ ГОЛОСОВОГО УПРАВЛІННЯ У КОНТЕКСТІ РОЗУМНОГО БУДИНКУ

Метою роботи є аналіз методів розпізнавання голосу, наочне порівняння, визначення найбільш підходящих для розумного будинку та розробка рекомендацій щодо їх реалізації. Очікується, що результати будуть цінними як для теоретиків, так і для практиків.

1.1 Огляд побудови розумного будинку

Далі будуть розглянуті принципи роботи систем розпізнавання голосу, існуючі моделі та методи, особливості застосування в розумному будинку, а також проведено порівняльний аналіз та надано рекомендації щодо практичної реалізації.

Почати планується з огляду побудови розумного будинку (рисунок 1.1).

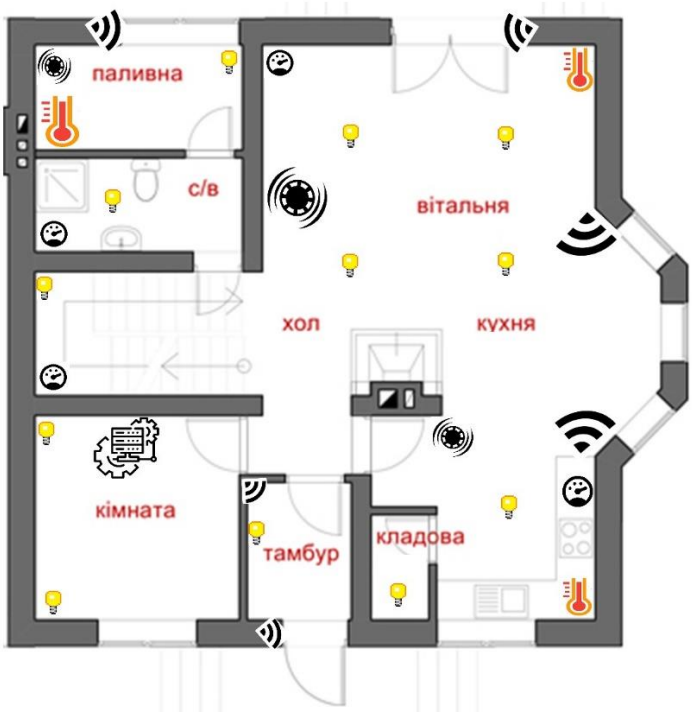

Рисунок 1.1 – Приклад розташування системи у будинку

Пояснення до рисунку:

1.  – хаб;
2.  – датчик диму;
3.  – датчик руху;
4.  – датчик температури;
5.  – датчик вологості;
6.  – лампа освітлення.

Ефективний вибір та розподіл датчиків по розумному будинку – одна з частин сучасної технологічної еволюції. Цей процес допомагає під ще одним кутом подивитись на потреби користувача, а особливо користувача з обмеженими можливостями а відповідно й на актуальність та затребуваність системи.

Зробити ефективний крок у дослідженні систем розумний будинок звичайно допоможе аналіз існуючих аналогів. Нижче представлена порівняльна таблиця існуючих аналогів систем крупних компаній (таблиця 1.1).

Таблиця 1.1 – Порівняльна таблиця різних існуючих аналогів систем «Розумний будинок»

Назва	Зображення	Плюси	Мінуси
Google Nest		Інтеграція з екосистемою Google: Google Assistant, Google Home та ін.	Залежність від Інтернету
		Простота використання	Велика вартість
		Гідна енергоефективність	Погана сумісність з девайсами інших виробників
Apple HomeKit		Інтеграція з екосистемою Apple: Siri, Apple Home та ін.	Обмежений вибір пристроїв для розумного будинку
		Приватність та безпека даних	Велика вартість
		Apple регулярно надає оновлення програмного забезпечення	Залежність від Інтернету
Amazon Echo		За допомогою пристроїв Echo можна створювати розумні рутини, автоматизуючи завдання та сценарії.	Приватність та дані: пов'язано зі збором аналітичних даних, що може викликати недовіру у користувача
		Гарна інтеграція зі сторонніми онлайн сервісами	Залежність від Інтернету
		Широкий вибір пристроїв	Можливі проблеми із розпізнаванням голосових команд.

Затребуваність системи базується на подоланні бар'єрів зручності та забезпечення належного способу життя. Ці системи надають:

- комфорт: регулює умови середовища для максимального комфорту;
- безпеку: моніторинг та повідомлення для забезпечення безпеки користувача;
- свободу руху: голосове керування та безконтактні інтерфейси для свободи руху;

- турботу про здоров'я: інтегровані системи медичного моніторингу для піклування про здоров'я;

- оптимізацію ресурсів: передбачення потреб та ефективне використання ресурсів.

Тож виходячи з аналізу потреб користувачів – для підтримки актуальності система має включати такі пункти (рисунок 1.2):

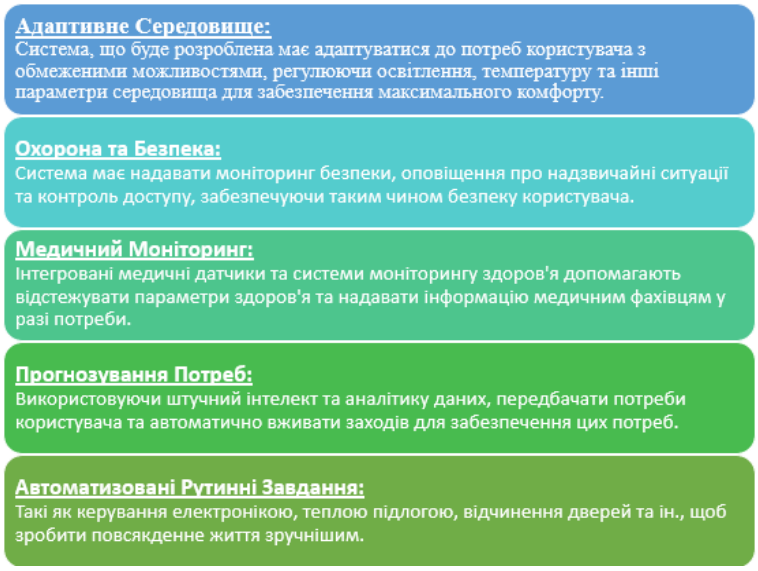

Рисунок 1.2 – Пункти, що забезпечують актуальність системи

Автоматизованість є важливим аспектом сучасних систем розумного будинку. Наявність голосового керування дозволяє покращити рівень автоматизації в домашньому середовищі, роблячи його більш інтелектуальним та ефективним. Голосові команди можуть бути пов'язані з різними сценаріями автоматизації, такими як керування освітленням, опаленням, кондиціюванням повітря, аудіо-відео системами та безпекою. Це дозволяє користувачам створювати персоналізовані налаштування та режими роботи свого будинку, враховуючи їх переваги та повсякденні потреби. Таким чином, голосове управління не тільки забезпечує зручність і доступність, але й сприяє підвищенню рівня автоматизації в розумному будинку, роблячи його розумнішим, ефективнішим та адаптивнішим до потреб користувачів.

Для системи розумного будинку, що розглядається, наявність саме голосового управління є важливою, зважаючи на те, що воно забезпечує доступність та зручність управління домашнім середовищем для всіх членів сім'ї, включаючи людей з обмеженими можливостями. Голосове керування дозволяє контролювати різні функції на місці без необхідності фізичної взаємодії з пристроями, що робить систему більш інклюзивною та доступною для всіх.

Крім того, голосове управління сприяє збільшенню безпеки та комфорту в будинку. Воно дозволяє швидко активувати системи безпеки та реагувати на надзвичайні ситуації, такі як пожежа або вторгнення з мінімальними зусиллями з боку користувача. Це особливо важливо для людей з обмеженими фізичними можливостями, яким важко використовувати традиційні методи управління. Тому в цій кваліфікаційній роботі буде звернена особлива увага на розробку та оптимізацію голосового управління у системах розумного будинку.

1.2 Історія розвитку методів голосового управління

Голосове управління, від перших експериментальних кроків до сучасних діалогових інтерфейсів, пройшло значний шлях із невпевненості до широкого впровадження. Починаючи з 1952 року, коли Bell Laboratories розробили першого експериментального голосового робота Audrey, і до наших днів, голосове керування еволюціонувало разом з технологічними досягненнями.

У 1960-х роках з'явилися системи розпізнавання ізольованих слів, які дозволяли керувати комп'ютерами, що стало проривом у розвитку голосового інтерфейсу. Протягом 1970-х років почалося впровадження систем голосового управління в промисловості, що відкрило нові можливості в автоматизації.

У 2000-х роках з'явилися мобільні девайси з голосовим керуванням, а 2007 рік приніс випуск першого iPhone з Siri, віртуального асистента, що відкрив нові горизонти для інтерактивності з пристроями.

Сучасний етап розвитку голосового управління характеризується впровадженням штучного інтелекту та створенням розумних асистентів, які можуть вести розмову. Релізи продуктів, таких як Amazon Echo з Alexa та Google Home, показують напрямки, в яких розвивається ця технологія, роблячи її доступною для широкого загалу [2].

Основний помітний вплив технологічних досягнень у сфері голосового управління:

1. розвиток мікрофонів:

- підвищення чутливості: мікрофони стали більш чутливими, що дозволяє їм уловлювати тихий голос та команди з дальньої відстані. Це робить голосове керування зручнішим і практичнішим, оскільки вам не потрібно кричати або підходити близько до пристрою;

- шумопридушення: мікрофони тепер краще справляються із шумом, що дозволяє їм розпізнавати команди навіть у голосливій обстановці. Це робить розпізнавання голосу більш надійним та функціональним, тому що вам не потрібно турбуватися про те, що ваш голос буде заглушений фоновим шумом;

2. поліпшення алгоритмів розпізнавання мови:

- підвищення точності: алгоритми розпізнавання мови стали більш точними, що дозволяє розпізнавати команди з меншою кількістю помилок. Це робить голосове управління зручнішим і практичнішим, тому що вам не потрібно повторювати команди кілька разів;

- підвищення швидкості: алгоритми розпізнавання промови почали працювати швидше, що дозволяє їм розпізнавати команди практично миттєво. Це робить голосове керування більш чуйним та інтуїтивним, оскільки вам не потрібно чекати, поки пристрій розпізнає вашу команду;

3. розвиток штучного інтелекту:

- підвищення природності: штучний інтелект дозволяє зробити голосове управління природнішим. Тепер є можливість розмовляти з

пристроями так, як користувач розмовляє з людьми, використовуючи природну мову та інтонації;

- підвищення функціональності: ШІ дозволяє розширити функціональність голосового керування. Тепер можливо не тільки керувати пристроями, але й ставити запитання, отримувати інформацію та виконувати складні завдання.

Технологічні досягнення значно підвищили точність, швидкість та функціональність голосового управління. Штучний інтеллект грає ключову роль розвитку сучасних систем голосового управління.

1.3 Класифікація методів голосового управління. Сегментація та каталогізація

Методи голосового управління можна класифікувати за кількома критеріями:

1. спосіб розпізнавання мови:
- дискретне розпізнавання: цей метод розпізнає окремі слова чи фрази;
- безперервне розпізнавання: цей метод розпізнає потокове мовлення;
2. розмір словника:
- спеціалізований: словник обмежений набором команд, відповідних конкретному завданню;
- універсальний: словник містить широкий спектр слів та фраз;
3. підхід до моделювання мови:
- статичний: моделі мови ґрунтуються на статичних даних;
- динамічний: моделі мови ґрунтуються на штучних нейронних мережах;
4. спосіб навчання:
- спрямоване навчання: моделі навчаються на labeled data;
- ненаправлене навчання: моделі навчаються на unlabeled data;
5. доступ до сервера:

- клієнт-сервер: розпізнавання мовлення відбувається на сервері;
- вбудований: розпізнавання мовлення відбувається на пристрої;

6. модальність:

- голосове керування: управління здійснюється лише голосом;

мультимодальне управління: управління здійснюється голосом та іншими способами, так як жести або дотики;

7. мова:

- мономовний: система працює лише з однією мовою;
- багатомовний: система працює з кількома мовами;

8. активність користувача:

- на запит: користувач повинен активувати систему перед використанням;
- постійне: система завжди активна та готова до роботи;

9. доступність:

- відкритий вихідний код: код системи доступний для всіх;
- пропрієтарний: код системи недоступний;

10. ціна:

- безкоштовний: система безкоштовна для використання;
- платна: система вимагає оплати;

11. продуктивність:

- точність: відсоток правильно розпізнаних команд;
- затримка: час, необхідний для розпізнавання команди;
- надійність: стійкість системи до помилок;

12. безпека:

- конфіденційність: захист персональних даних користувача;
- аутентифікація: підтвердження особи користувача;

13. доступність:

- легкість використання системи для користувачів з різним рівнем підготовки;
- універсальність: підтримка різних пристроїв та мов.

Важливо, що дана класифікація є умовною, і деякі методи голосового управління можуть одночасно належати до кількох категорій [3].

В даний час найбільш поширені методи голосового управління, засновані на безперервному розпізнаванні мови, статичних моделях мови та спрямованому навчанні. Проте, нейронні методи голосового управління стають дедалі популярнішими, оскільки можуть забезпечити вищу точність і швидкість розпізнавання промови. У майбутньому методи голосового управління розвиватимуться далі, і вони стануть ще зручнішими, практичнішими, надійнішими, функціональнішими та природнішими.

2 АНАЛІЗ МОДЕЛЕЙ ТА МЕТОДІВ

2.1 Аналіз сучасних підходів до голосового управління

Для того, щоб бути впевненим у правильноми виборі методів, що будуть використані під час створення застосунків було вирішено зробити досконалий аналіз літературних джерел. Нижче буде опис декількох статей, що розглядають методи, моделі та загальний підхід до голосового управління.

2.1.1 Дослідження розпізнавання мовлення, перекладу та розуміння за допомогою дискретних мовних одиниць: порівняльне дослідження

У наступній роботі досліджується застосування дискретних мовних одиниць у моделях наскрізної обробки мовлення (End-to-End Speech Processing, E2E-ASR).

Автори статті [4] пропонують використовувати дискретні мовні одиниці, отримані з самоконтрольованого навчання замість багатовимірних мовних характеристик, таких як спектрограми. Це дозволяє значно стиснути розмір мовних даних і, отже, скоротити час навчання моделі.

Для подальшого зменшення довжини мовної послідовності автори пропонують використовувати методи дедуплікації та моделювання підслів.

Експерименти, проведені на 16 корпусах даних, показали, що моделі, засновані на дискретних мовних одиницях, досягають порівнянних з традиційними моделями результатів значно менших обчислювальних витратах.

Як методи і моделі розпізнавання мови, розглянутих у цій роботі, можна назвати:

- моделі самоконтрольованого навчання: використовуються для одержання дискретних мовних одиниць;

- методи дедуплікації: використовуються зменшення довжини мовної послідовності;
- моделювання підслів: використовується зменшення довжини мовної послідовності;
- моделі наскрізної обробки мовлення (e2e-asr): моделі, у яких всі етапи розпізнавання мовлення виконуються у єдиній нейронної мережі [4].

2.1.2 Комерційна дискретна система розпізнавання мовлення з великим словниковим запасом: DragonDictate.

У роботі [5] розглядається система розпізнавання промови DragonDictate.

DragonDictate – це комерційно доступна система розпізнавання мовлення загального призначення, яка використовує дискретну мову і залежить від того, хто говорить. Ця система використовує дискретне мовлення та залежить від мовця, адаптуючись до голосу мовця та мовної моделі з кожним словом. Її акустична адаптивність базується на трирівневій фонології та стохастичній моделі виробництва. Фонологічними рівнями є фонеми, доповнені трифони (фонеми в контексті або PIC) і спектральні зрізи в стаціонарному стані, які об’єднані для наближення спектрів цих PIC (фонетичних елементів або PEL) і, таким чином, слів.

Акустична адаптивність DragonDictate заснована на трирівневій фонології:

- фонеми: базові одиниці звуку;
- розширені трифони: фонеми у тих, які враховують вплив сусідніх звуків;
- стійкі спектральні зрізи: апроксимація спектрів розширених трифонів [5].

2.1.3 Міжмовне самонавчання для вивчення багатомовного представлення для розпізнавання мовлення з низькими ресурсами

У статті [6] демонструються значні покращення продуктивності, досягнуті з використанням запропонованого методу. Наприклад, було показано покращення точності розпізнавання мовами з обмеженими ресурсами. Застосування крос-лінгвального самонавчання дозволило суттєво підвищити точність розпізнавання команд у розумних будинках, де можуть використовуватися різні мови, та знизити залежність від великих обсягів розмічених даних, що критично для мов з обмеженими ресурсами.

Використання методу крос-лінгвального самонавчання та мультилінгвальних уявлень виправдане, оскільки дозволяє покращити взаємодію з користувачами в системах розумного будинку, забезпечуючи зручність та доступність для користувачів, а також знизити витрати на розробку, заощаджуючи час та ресурси при зборі даних для кожної мови окремо [6].

2.1.4 Розумний домашній помічник для незрячих з розпізнаванням голосу

Ще одна стаття [7] пропонує важливі технологічні рішення для покращення систем розпізнавання голосу, які можуть бути адаптовані до розумного будинку.

Основний внесок цієї статті полягає у розробці прототипу розумного будинку для людей з порушеннями зору, який використовує голосове управління. Система включає мікрофон, підключений до Android-пристрою, і NodeMCU, з'єднані через Wi-Fi. Система розпізнавання голосу на основі Android обробляє голосові команди та надсилає відповідні інструкції на мікроконтролер для виконання завдань.

Запропоновані рішення включають управління електронними пристроями (включення та вимкнення), надання інформації новин, розваги (відтворення потокового радіо) та надання загальної інформації (час і дата). Ці функції значно покращують взаємодію з користувачами, особливо з тими, хто має обмеження зору. Застосування подібних технологій у розумних будинках робить їх доступнішими та зручнішими для всіх категорій користувачів, включаючи людей з обмеженими можливостями. Методи та рішення, представлені у статті, сприяють підвищенню автономності та покращенню якості життя користувачів розумних будинків, знижуючи залежність від фізичних взаємодій із пристроями.

Таким чином, це дослідження надає цінні практичні методи та технології, які можуть бути адаптовані та впроваджені в систему розумного будинку для покращення функціональності та доступності голосового управління [7].

2.2 Дослідження розглянутих методів та моделей розпізнавання голосу, їх переваг та недоліків

Після проведення аналізу методів голосового управління в попередніх розділах можемо порівняти такі підходи, як:

Методи:

- дискретне розпізнавання;
- безперервне розпізнавання.

Моделі:

- статичні моделі;
- динамічні моделі.

При дослідженні методів розпізнавання голосу важливо розуміти, що методи мають такі порівнювальні характеристики, як важкість реалізації, затребувана обчислювальна потужність, точність, швидкість та надійність.

Досліджуючи різні методи розпізнавання голосу було неодноразово помічено, що вони дають відповідно різні результати та вочевидь різні аспекти використання. Це було відображено у вигляді таблиці (таблиця 2.1):

Таблиця 2.1 – Порівняння різних методів розпізнавання голосу

Параметр	Дискретне розпізнавання	Безперервне розпізнавання
Плюси	Низькі обчислювальні ресурси	Природна промова
	Проста реалізація	Розпізнавання пауз та інтонацій
	Висока точність розпізнавання слів	Широка функціональність
Мінуси	Необхідність чіткої артикуляції	Високі обчислювальні ресурси
	Неможливість розпізнавання пауз та інтонацій	Складна реалізація
	Обмежена функціональність	Низька точність розпізнавання слів

Ці два методи є найбільш поширеними та добре вивченими. Вони є принципово різними підходами до розпізнавання мови, що дозволяє порівняти їх переваги та недоліки. Дискретне розпізнавання мови вимагає від користувачів чіткої артикуляції та може бути обмеженим у функціональності. З іншого боку, безперервне розпізнавання мови дозволяє природніше спілкування, але вимагає більше обчислювальних ресурсів.

Мовна модель – це ймовірнісний розподіл на безлічі словникових послідовностей. Для речень таких як «Будинок, відкрий двері» або «Бот, зачини вікно», мовна модель дасть нам ймовірність того, що ми зустрінемо цю пропозицію. Оцінкою якості мовної моделі служить перплексія (perplexity) — міра того, наскільки добре розподіл імовірності або статична модель прогнозує вибірку. Її можна використовувати для порівняння ймовірнісних моделей.

Низька перплексивність означає, що розподіл ймовірності добре передбачає вибірку.

Якщо казати про моделі, то основні два протиставлення будуть виглядати наступним чином (таблиця 2.2):

Таблиця 2.2 – Порівняння різних моделей розпізнавання голосу

Параметр	Статичні моделі	Динамічні моделі
Плюси	Проста реалізація	Висока точність
	Низькі обчислювальні ресурси	Самонавчання
	Добре вивчені	Стійкість до шуму
Мінуси	Обмежена точність	Складна реалізація
	Нездатність до самонавчання	Високі обчислювальні ресурси
	Чутливість до шуму	Вимагають великих масивів даних

Після проведеного аналізу та порівняння є зформоване розуміння які методи будуть розглядатися надалі.

2.3 Визначення та обґрунтування методів

В результаті дослідження було визначено два методи розпізнавання голосу, які будуть розглянуті надалі. Ці методи будуть докладно вивчені для оцінки їх ефективності та застосування у різних завданнях розпізнавання мови.

2.3.1 Перший метод

Перший метод полягає у хешуванні вхідних даних для розпізнавання у оффлайн режимі. Цей підхід було обрано через універсальність методу по відношенню до пристроїв а також до фактору наявності інтернету.

Якщо розглядати метод хешування, то відбуваються така послідовність:

- аудіофайл розбивається на короткі часові відрізки (фрейми), та для кожного кадру виконується перетворення Фур'є. Це дозволяє представити аудіосигнал у частотній ділянці;

- у спектрограмі для кожного кадру виділяються найбільш значущі частотні піки (крапки з максимальною амплітудою). Ці піки є ключовими характеристиками аудіосигналу;

- використовуючи координати вибраних піків (частоту та час), створюються хеші. Хеш являє собою компактне представлення аудіофрагменту і включає інформацію про частоти та їх відносне розташування в часі.

Тепер, коли є хеш, що являє собою вхідні дані відбувається оффлайн розпізнавання:

- відбувається порівняння вхідного хешу з тими хешами, що були збережені у моделі;

- якщо знайдено значні збіги, то система повертає інформацію про розпізнане слово/фразу, що найбільш ймовірно відповідає фрагменту запиту. Якщо збігів недостатньо, може бути повернено повідомлення про неможливість розпізнавання.

2.3.2 Другий метод

Другий метод полягає у онлайн розпізнаванні голосу шляхом перетворення аудіодоріжки на масив з частотою дискретизації. Під час роботи такого методи зазвичай відбуваються такі кроки:

- записана аудіодоріжка перетворюється на цифровий формат: це перетворення включає дискретизацію аудіосигналу, тобто перетворення безперервного звукового сигналу на масив цифрових значень з певною частотою дискретизації (зазвичай вимірюється в Герцах, Гц);

- обробка аудіо даних: програма обробляє отриманий масив цифрових значень. Цей етап включає приглушення фонового шуму підвищення якості розпізнавання промови. Також можуть бути застосовані інші методи попередньої обробки, такі як нормалізація рівня гучності;

- передача даних моделі розпізнавання: оброблений масив даних передається в модель розпізнавання мови. У разі онлайн-методу, модель розпізнавання, наприклад Google Speech-to-Text, знаходиться на віддаленому сервері;

- порівняння та розпізнавання: модель на сервері порівнює отриманий масив із навченими зразками мови. На основі цього порівняння модель генерує текст, що відповідає розпізнаній мові;

- повернення розпізнаного тексту: результат, тобто розпізнаний текст, повертається до програми. Цей текст може бути використаний для виконання різних команд або завдань.

Розглянувши та обравши два методи розпізнавання голосу їх було обґрунтовано та обрано як основи для написання двох застосунків.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ РОЗПІЗНАВАННЯ ГОЛОСУ

Під час написання 1 розділу, «Аналіз еволюції голосового управління у контексті розумного будинку», було розглянуто побудову розумного будинку, для якої було аргументовано необхідність додання голосового управління. Виходячи з цього було проведено аналіз з метою вибору моделі розпізнавання голосу та подальшого розвитку проєкту.

У ході аналізу роботи системи розумний будинок для наочності було побудовано наступну блок-схему (рисунок 3.1).

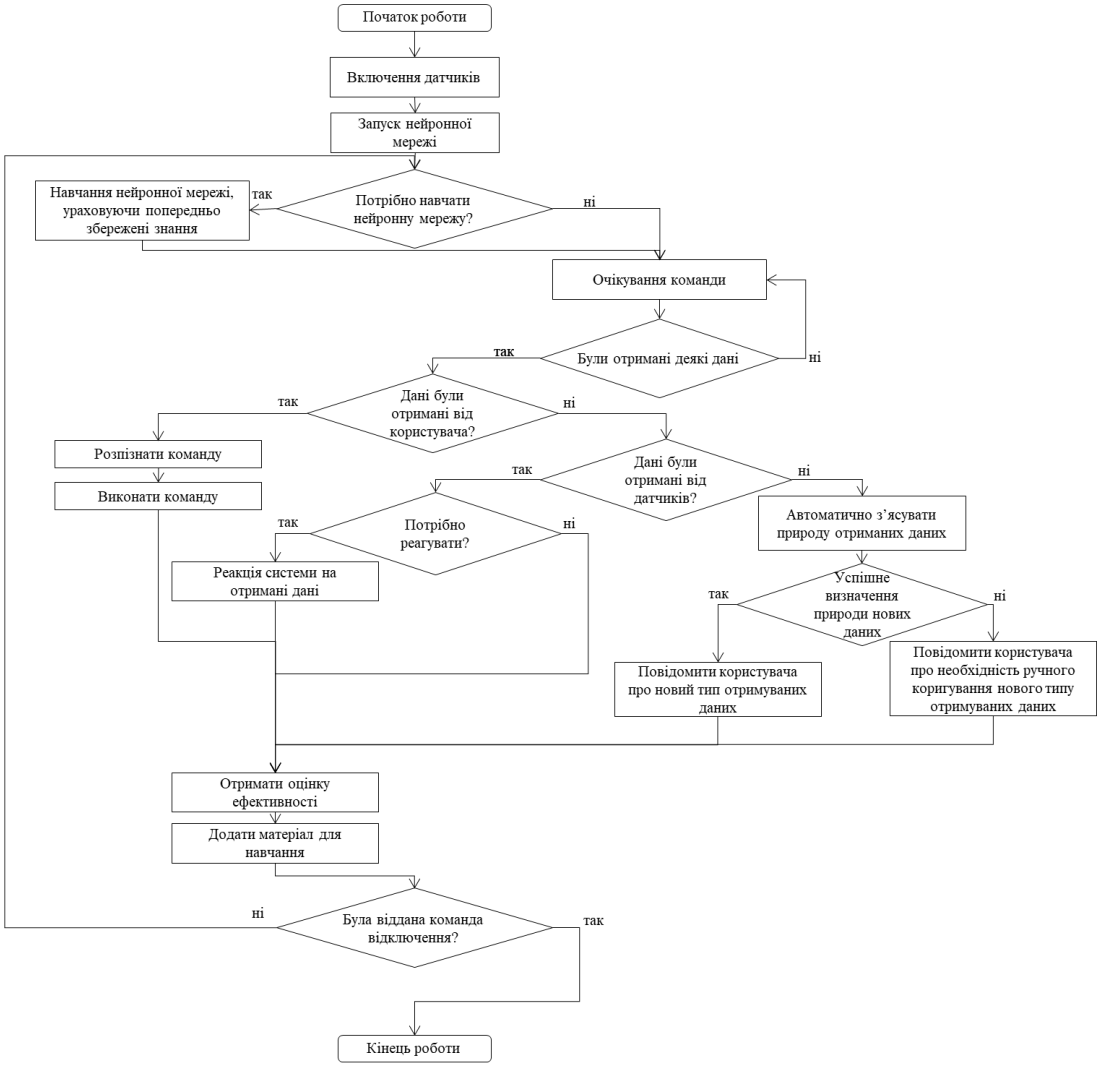

Рисунок 3.1 – Блок-схема, що наочно відображає поведінку розумного будинку

На базі проведеного аналізу було вирішено розробити два застосунки, що використовують різні, майже протилежні, методи розпізнавання голосу. А також порівняти їх за ключовими технічними параметрами. Ці застосунки будуть засновані на обґрунтованих у розділі 2 методах:

- Offline розпізнавання голосу, реалізоване за допомогою хешування та бібліотеки vosk;
- Online метод розпізнавання голосу шляхом перетворення аудіодоріжки на масив з частотою діскретизації.

Надалі буде детальніше розглянуто створення та принцип роботи цих застосунків.

3.1 Offline розпізнавання голосу, реалізоване за допомогою хешування та бібліотеки vosk

3.1.1 Огляд використаних бібліотек. Створення інтерфейсу

Дослідивши та обравши методи розпізнавання голосу було вирішено зупинитися на статичній одномовній моделі. Реалізувати це було вирішено за допомогою мови програмування Python та бібліотек:

- tkinter;
- customtkinter;
- vosk;
- CountVectorizer;
- LogisticRegression.

tkinter та customtkinter були використані для створення інтерфейсу продукту, якщо користувач буде використовувати його за допомогою комп’ютеру.

Розроблений інтерфейс виглядає таким чином (рисунок 3.2):

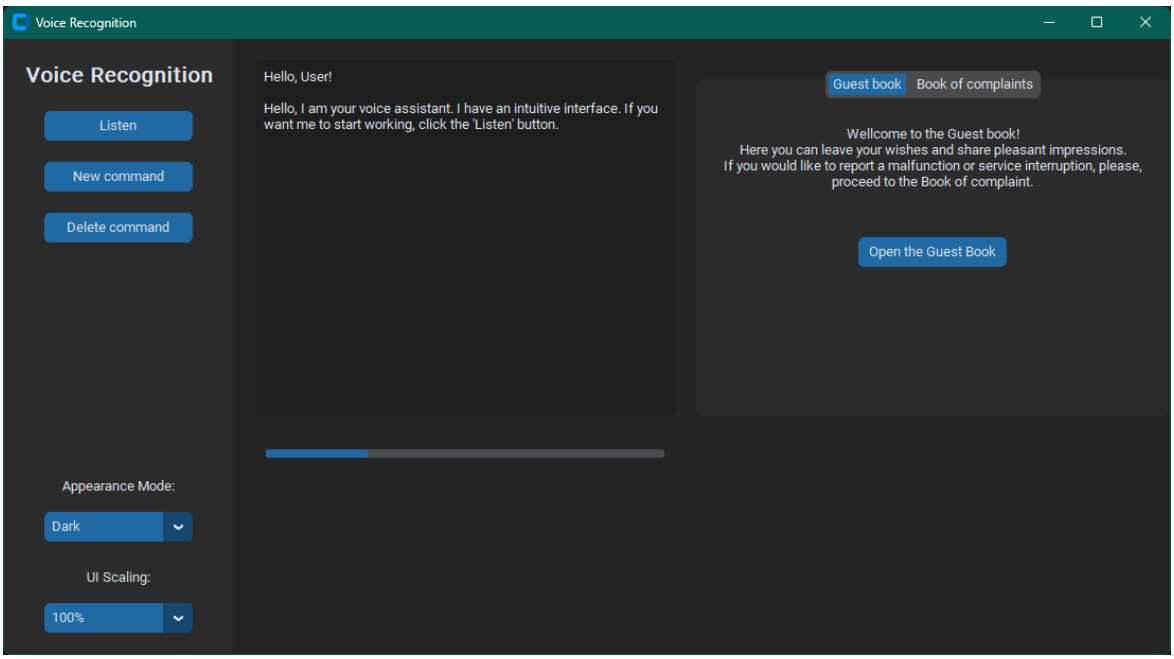

Рисунок 3.2 – Домашня сторінка продукту

Якщо користувач бажає залишити відгук або скаргу, то він може це зробити, натиснувши відповідну кнопку (рисунок 3.3):

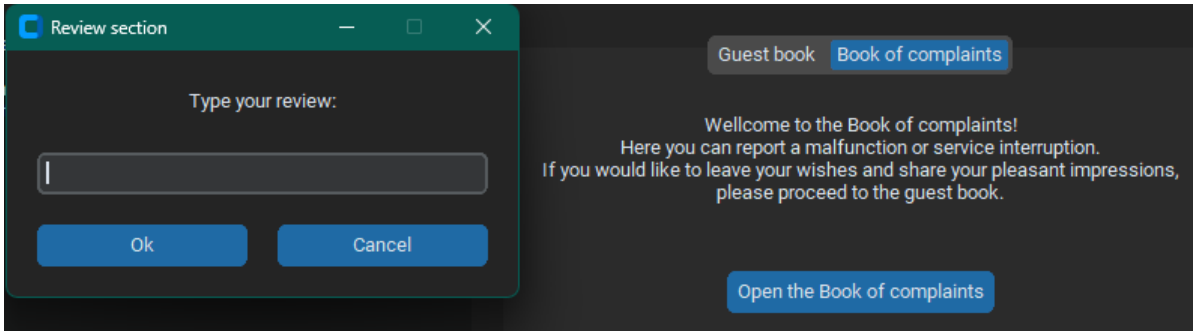

Рисунок 3.3 – Вікно для того щоб залишити скаргу або відгук

Якщо потрібно додати нову розпізнавану команду – вистачить лише написати її у текстовому форматі, натиснувши на відповідну кнопку «New command» (рисунок 3.4):

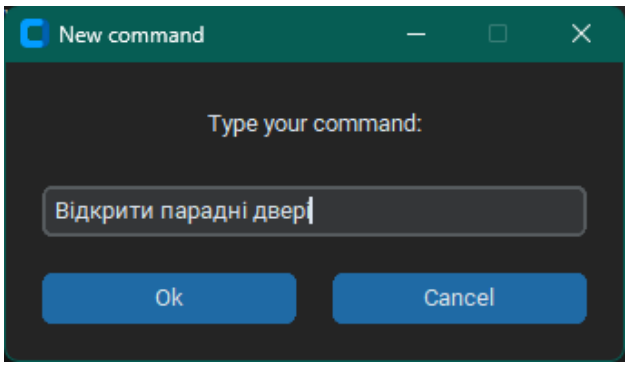

Рисунок 3.4 – Створення нової команди

Для видалення команди потрібно зробити аналогічні дії, натиснувши відповідну кнопку «Delete command».

Окрім того у користувача за потреби є можливість змінити масштаб або кольорову схему.

Але якщо казати про основний функціонал, то потрібно натиснути на кнопку «Listen». Після цього основний функціонал починає працювати:

- прослуховується уся промова користувача;
- якщо було звернення до бота(за заздалегідь визначеним кодовим словом), та у звернені була сказана записана команда – бот віддає наказ (датчикам/пристроям/ін.) виконати команду;
- прослуховування продовжується доки не буде віддана команда «вимкнути систему».

3.1.2 Реалізація методу розпізнавання голосу

Під час розпізнавання задіяні бібліотеки vosk, CountVectorizer та LogisticRegression.

vosk:

Бібліотека для розпізнавання промови "vosk". Переваги бібліотеки:

- підтримує 20+ мов та діалектів – англійська, індійська англійська, німецька, французька, португальська, іспанська, китайська, турецька, в'єтнамська, італійська, голландська, валенсійська, арабська, грецька, перська,

філіппінська, українська, каз. есперанто, хінді, чеська, польська, узбецька, корейська. незабаром будуть додані й інші;

- працює без доступу до мережі навіть на мобільних пристроях – raspberry pi, android, ios;

- встановлюється за допомогою простої команди pip3 install vosk без додаткових кроків;

- моделі для кожної мови займають всього 50мб, але є і більш точні великі моделі для більш точного розпізнавання;

- зроблено для потокової обробки звуку, що дозволяє реалізувати миттєву реакцію на команди;

- підтримує кілька популярних мов програмування – java, c#, javascript…;

- дозволяє швидко налаштовувати словник розпізнавання для покращення точності розпізнавання;

- дозволяє ідентифікувати того, хто говорить [8].

Найголовніше, що робить ця бібліотека – надає змогу створювати базу слів, що розпізнаються та доповнювати її, що й було використано.

Використано для тестування дві готові навчені моделі для української мови, а саме (таблиця 3.1):

Таблиця 3.1 – Використані під час тестів моделі

Модель	vosk-model-small-uk-v3-nano	vosk-model-uk-v3-lgraph
Опис	Невелика статична модель	Велика динамічна модель
Розмір (Мб)	73	325
Пам’ять (Мб)	300	1600
Точність	Низька	Висока
Швидкість	Висока	Середня
Застосування	Мобільні пристрої, системи, що вбудовуються	Сервери, високопродуктивні пристрої
Додаткові характеристики	Низьке споживання енергії	Середня латентність

Велика модель вже мала у собі 4820246 слів, які може розпізнавати. У той час як маленька модель має лише 1690714 слів. Під час тестування було відмічено, що велика модель має набагато довший час включення, але при цьому більші значення показників точності, швидкості та надійності.

CountVectorizer перетворює текст, що було отримано (роспізнавши голос з мікрофону за допомогою vosk) на матрицю підрахунків токенів. Тобто робить хешування тексту. Ця реалізація створює розріджене представлення підрахунків. Якщо не надається апріорний словник і не використовується аналізатор, який виконує певний вибір функцій, тоді кількість функцій дорівнюватиме розміру словника, знайденого в результаті аналізу даних [9].

LogisticRegression – Цей клас реалізує регуляризовану логістичну регресію з використанням «liblinear» бібліотеки, «newton-cg», «sag», «saga» та «lbfgs». У випадку з кількома класами алгоритм навчання використовує схему «один проти решти» (OvR), якщо для параметра «multi_class» встановлено значення «ovr», і використовує втрату крос-ентропії, якщо для параметра «multi_class» встановлено значення «multinomial» [10]. Цей клас(бібліотека) використовується для порівняння (отриманих за допомогою CountVectorizer) хеш-функцій. Порівнюються ті, що надходять з мікрофону з тими, що збережені у моделі.

У додатку А «Лістинг файлу «speech_recognition_1.py»» можна побачити, що файл «voice_recognition_1.py» містить основну логіку для розпізнавання голосу та виконання відповідних команд. Використовуються бібліотеки sounddevice для доступу до аудіо даних з мікрофону, vosk для розпізнавання голосу, CountVectorizer для хешування тексту, та LogisticRegression для порівняння хешів та визначення відповідних дій.

У додатку Б «Лістинг файлу «app.py»» можна помітити файл «app.py» – цей файл є центральною інтерфейсною частиною програми, яка призначена для активної взаємодії з користувачем і координації процесу розпізнавання мови. Цей файл відіграє ключову роль у забезпеченні досвіду користувача,

надаючи користувачеві зручний і інтуїтивно зрозумілий інтерфейс для використання функцій розпізнавання голосу.

3.1.3 Створення інтелектуальної бази

Створено файл «words.py» (повний зміст якого міститься у додатку Г), який використовується як клас з набором команд та тригерів. Виглядає таким чином (рисунок 3.5):

words.py > ...

```
TRIGGERS = {'бот', 'буд', 'бут', 'дім', 'будинок', 'команда'}



#382693 бот big
#128762 бот small
data_set = {

#Base
'виключай систему': 'offBot Turning off',
'виключає систему': 'offBot Turning off',
'виключаю систему': 'offBot Turning off',

#Smart home
'відчини двері': 'passive Opening doors',
'відчинить двері': 'passive Opening doors',
'відчинять двері': 'passive Opening doors',
'відчиняй двері': 'passive Opening doors',
'відчиню двері': 'passive Opening doors',

'виключи воду': 'passive Turning off the taps',
'виключай воду': 'passive Turning off the taps',
'виключу воду': 'passive Turning off the taps',
'виключиш воду': 'passive Turning off the taps',
'закрий воду': 'passive Turning off the taps',
'закривай воду': 'passive Turning off the taps',
'закриваю воду': 'passive Turning off the taps',
'перекрий воду': 'passive Turning off the taps',

'включи воду': 'passive Turning on the taps',
'включай воду': 'passive Turning on the taps',
'включу воду': 'passive Turning on the taps',
'включиш воду': 'passive Turning on the taps',
'відкрий воду': 'passive Turning on the taps',
'відкривай воду': 'passive Turning on the taps',
```

Рисунок 3.5 – Файл «words.py»

Як можна помітити – файл містить:

- TRIGGERS – тригери, які є звертанням до програми;

- data_set – Набір команд, що при розпізнанні та успішному порівнянні хеш-функцій будуть запускати ‘x’ набор інструкцій та говорити ‘y’ фразу:

-

```
‘команда’: ‘x y’
```

Набори інструкцій також було винесено в окремий файл «skills.py» (повний зміст якого міститься у додатку В) та використано як клас. Виглядає наступним чином (рисунок 3.6):

Рисунок 3.6 – Файл «skills.py»

У цьому файлі містяться набори інструкцій/функції, що будуть виконуватися у випадку успішного порівняння хеш-функцій. Деякі набори ще не до кінця продумані, але будуть дороблятися.

Загалом папка проєкту містить файли «app.py», «voice_recognition_1.py», «words.py» та «skills.py». Виглядає проєкт наступним чином (рисунок 3.7):

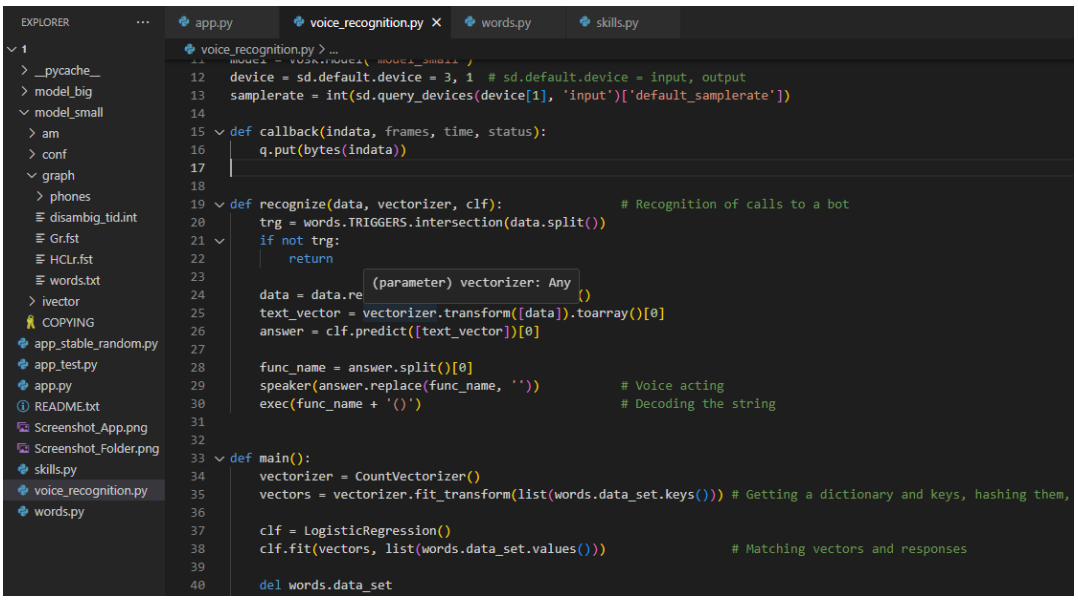

Рисунок 3.7 – Вигляд папки проєкту вцілому

Окрім того для розробників було розроблено окремий режим роботи, який передбачає відображення тимчасових результатів розпізнавання голосу. Використовується цей режим для дослідження процесу розпізнавання, доцільності обраного методу та ін. Виглядає наступним чином (рисунок 3.8):

Рисунок 3.8 – Режим «Для розробника»

У результаті отримано застосунок з Offline розпізнаванням голосу. Реалізоване це було за допомогою технології хешування та бібліотеки vosk.

Наступним кроком є створення іншого, майже протилежного застосунку, що використовує інший метод розпізнавання.

3.2 Online метод розпізнавання голосу шляхом перетворення аудіодоріжки на масив з частотою дискретизації

3.2.1 Огляд використаних бібліотек

Speech_reconition це коренева бібліотека для цього застосунку. Ця бібліотека використовує метод online розпізнавання голосу шляхом перетворення аудіодоріжки на масив з частотою дискретизації та поверненням результату у вигляді вже розпізнаного тексту. Ця бібліотека була започаткована як демо-проєкт у 2022 році.

Сама по собі ця бібліотека має змогу лише використовувати online метод розпізнавання, що й буде розглянуто у цій роботі. Але бібліотека також може «під’єднати» до себе сторонню навчену модель для автономного offline розпізнавання. Окрім того speech_recognition має такі можливості:

- підтримка різних движків і api: speech recognition може працювати з кількома движками розпізнавання мови, як онлайн, так і офлайн, включаючи google speech-to-text, cmu sphinx та microsoft azure speech services;
- простота використання: бібліотека надає звичайний та зрозумілий інтерфейс для роботи з розпізнаванням мови. Можливо легко записувати аудіо з мікрофона, перетворювати його на текст і виконувати інші завдання;
- підтримка кількох мов: бібліотека підтримує широкий спектр мов, що робить її придатною для використання у різних проектах;
- підтримка різних форматів аудіо: бібліотека може працювати з різними форматами аудіофайлів, такими як WAV, FLAC, OGG та MP3;

- розпізнавання ключових слів: Speech Recognition може бути налаштована на розпізнавання певних ключових слів або фраз, що може бути корисним для створення систем голосового керування або активації команд;

- фільтрування фонового шуму: Speech Recognition може використовуватися для фільтрації фонового шуму з аудіозаписів, що покращує якість розпізнавання [11].

Під час роботи з методом та бібліотекою була використана така модель (таблиця 3.2):

Таблиця 3.2 – Таблиця властивостей моделі Google Cloud Speech-to-Text

Модель	Google Cloud Speech-to-Text
Опис	Універсальна модель для різних сценаріїв.
Розмір (Мб)	Cloud (50 Мб)
Пам’ять (Мб)	Cloud (256 Мб)
Точність	Висока
Швидкість	Низька
Застосування	Підходить для більшості випадків, де потрібне розпізнавання мови.
Додаткові характеристики	Доступна адаптація до акустики каналу. Можна використовувати для розпізнавання мовлення відео.

3.2.2 Розробка застосунку

У цього застосунку лише один файл, лістинг якого було відображено у додатку Ґ «Лістинг файлу «speech_recognition_2.py»».

Цей застосунок дозволяє ефективно перетворювати промову на текст у реальному часі. Переваги такого підходу включають високу точність

розпізнавання завдяки потужним моделям, що працюють на віддалених серверах

У результаті проведеного аналізу коду було сформовано алгоритм його роботи покроково:

1. підключення бібліотек;
2. початок роботи, включення мікрофону;
3. встановлення тривалості прослуховування (якщо користувач мовчить);
4. приглушення шумів;
5. передача результатів (у вигляді масиву з частотою діскретизації) до моделі для розпізнання;
6. підключення до моделі google для порівняння отриманного масиву з набором масивів з навченої моделі
7. у разі отимання позитивних результатів – повернення їх у форматі:

```
you just said: «результат»
```

8. порівняння отриманих результатів та встановлених тригерів (слов-активаторів) заздалегіть виділенних наборів інструкцій/команд;
9. у випадку істино-позитивного порівняння – виконати набір інструкцій/команду;
10. повтор/кінець роботи.

У результаті отримано застосунок, що у форматі Online розпізнає голос та виконує команди, якщо був такий запит. Приклад роботи застосунку зображено нижче (рисунок 3.9):

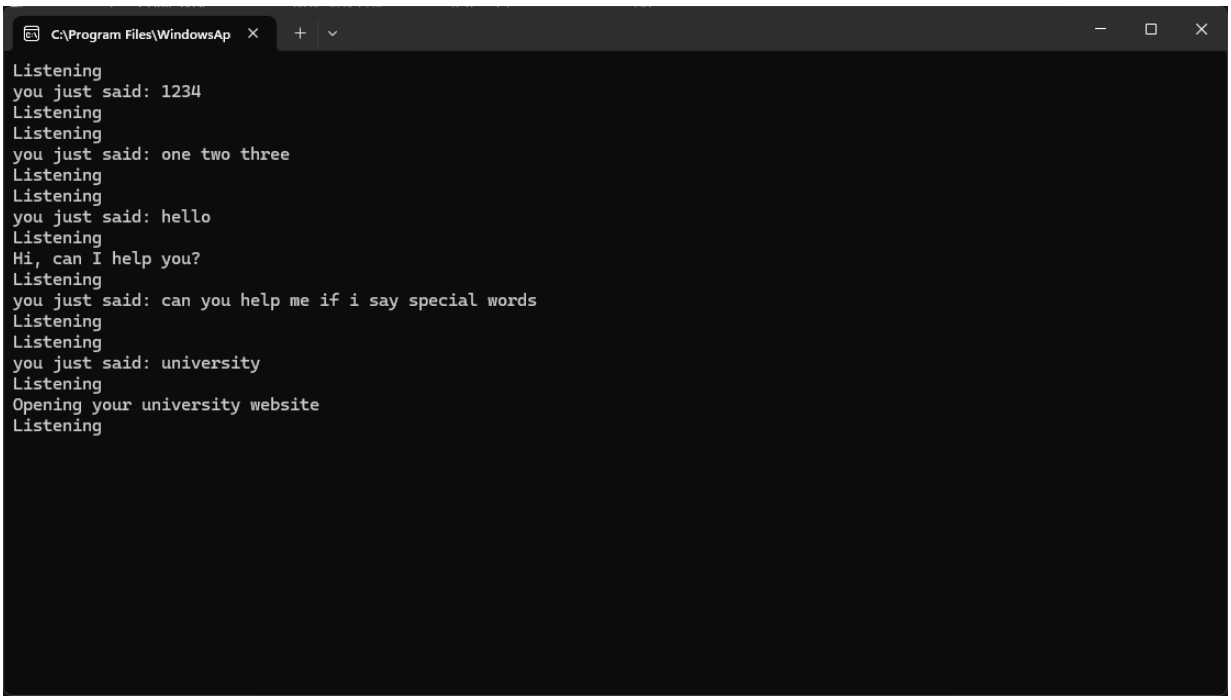

Рисунок 3.9 – Приклад роботи застосунку 2

3.3 Підсумки

У цьому розділі було розглянуто два методи розпізнавання мови: оффлайн метод з використанням хешування та онлайн метод на основі перетворення аудіодоріжки в масив із частотою дискретизації.

Перший метод продемонстрував можливість розпізнавання мови без підключення до Інтернету. Цей метод спирається на заздалегідь навчені моделі та локальні обчислювальні ресурси. Він має високий рівень конфіденційності та незалежності від зовнішніх сервісів, що особливо важливо в умовах, коли доступ до інтернету обмежений або потрібна обробка чутливих даних. Однак такий підхід потребує значних обчислювальних ресурсів для зберігання та обробки великих обсягів даних.

Другий метод показав переваги онлайн розпізнавання мови. Він відрізняється високою точністю і гнучкістю завдяки можливості використання потужних серверних ресурсів і моделей, що постійно оновлюються.

Обидва підходи мають свої переваги та недоліки, що дозволяє обрати найбільш підходящий метод залежно від конкретних умов та вимог. У наступному розділі будуть проведені розрахунки та порівняння ефективності цих методів, що дозволить зробити остаточні висновки щодо доцільності їх використання у різних сценаріях.

4 РОЗРАХУНКИ

4.1 Методи розрахунків, що будуть використані

Як було вказано у меті роботи – у цій роботі буде порівнюватись два методи розпізнавання мови. На даний момент реаалізовано два застосунка, що використовують різні методи розпізнавання голосу, а саме:

- Offline розпізнавання голосу, реалізоване за допомогою хешування та бібліотеки vosk.
- Online метод розпізнавання голосу шляхом перетворення аудіодоріжки на масив з частотою діскретизації

Робота цих застосунків, а відповідно й методів буде порівняна за такими технічними параметрами:

- точність = Accuracy $= \frac{W}{C} * 100\%$, де W – кількість правильно розпізнаних слів, C – загальна кількість слів;
- затримка = Delay $= \frac{T}{C}$, де T – час розпізнавання всіх слів C – загальна кількість слів;
- надійність = Reliability $= \frac{A_1+A_2+A_3}{3}$, де A_1 – точність при відсутності шуму, A_2 – точність при 1% шумі, A_3 – точність при 10% шумі.

Для дослідження вигляду волноформи та додання шуму на аудіозапис був використаний застосунок під назвою Audacity. Audacity це безкоштовний багатоплатформенний аудіоредактор звукових файлів, орієнтований на роботу з декількома доріжками. Цей застосунок забезпечує:

1. Кросплатформенність: працює на всіх основних операційних системах – Windows, MacOS і Linux.
2. Імпорт, експорт, конвертацію: Audacity підтримує всі основні аудіоформати, що дозволяє конвертувати WAV у MP3, FLAC, Ogg та багато іншого.

3. Підтримку плагінів: є можливість покращити свою продуктивність за допомогою широкого вибору плагінів сторонніх розробників, включаючи VST3, Nyquist тощо.

4. Глибокий аналіз аудіо: можливо візуалізувати частоти в представленні спектрограми Audacity або використовувати наукові аналізатори Vamp, щоб робити відкриття [12].

А значення, необхідні для підставлення у формули будуть отримані таким чином:

Робиться запис голосу довільної команди або набору слів. Наприклад: «Будинок два», що означатиме для будинку виконувати список дій під номером 2. Волноформа цього запису виглядає таким чином (рисунок 4.1):

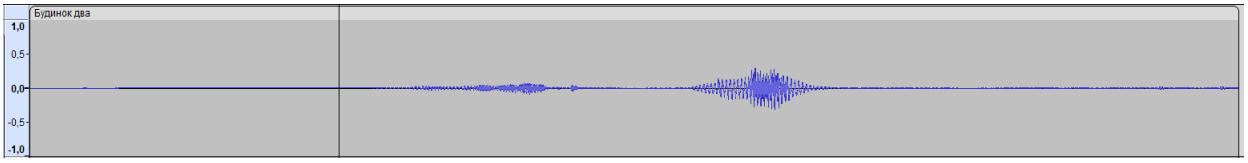

Рисунок 4.1 – Волноформа запису «Будинок два»

Цей запис тестується на обраному методі розпізнавання голосу визначену кількість разів та отримуються набір результатів, що усереднюється та вноситься до таблиці для порівняння.

Після чого накладається на запис шум, наприклад 1% (рисунок 4.2):

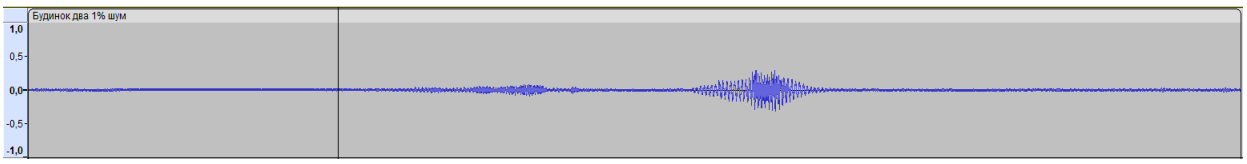

Рисунок 4.2 – Волноформа запису «Будинок два» з шумом 1%

Також само тестується запис та отримуються усереднені результати, а комбінуючи їх з результатом попереднього дослідження отримуються більш наближені до реального досвіду використання результати.

Також можна накласти шум 10%(або більше) для тестування точності, швидкості та надійності системи у екстремальних умовах (рисунок 4.3).

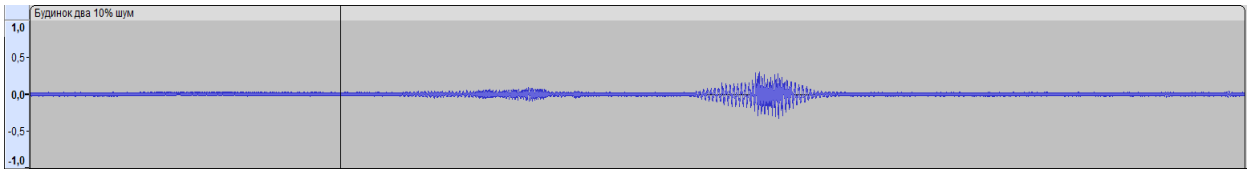

Рисунок 4.3 – Волноформа запису «Будинок два» з шумом 10%

Доцільніше буде використовувати команди з більшою кількістю слів для відповідності реальним умовам використання.

4.2 Розрахунки за умови відсутності шумів

Перед початком тестів зроблено аудіозапис на фізичний мікрофон. Цей запис містить промову таких слів-англіцизмів:

- бот;
- хот-дог;
- гугл;
- чекін;
- боулінг;
- вебсайт;
- мітинг;
- футбол;
- кабінет;
- магазин.

Волноформа цього запису виглядає таким чином (рисунок 4.4):

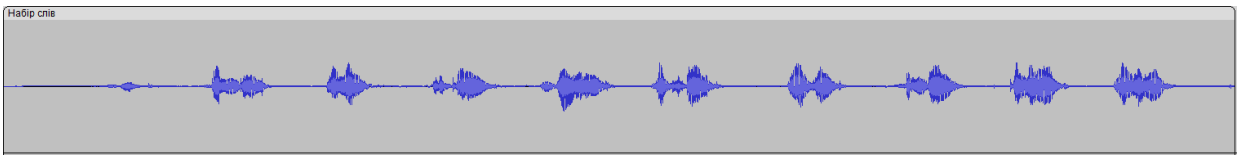

Рисунок 4.4 – Волноформа запису з набором слів для тесту

Результати першого тестування були записані у вигляді таблиці (таблиця 4.1):

Таблиця 4.1 – Результати тестування запису з набором слів-англіцизмів на застосунку 1

слово \ спроба №	1	2	3	4	5	6	7	8	9	10
бот	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗
хот-дог	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
гугл	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗
чекін	✓	✗	✗	✗	✗	✓	✗	✓	✓	✗
боулінг	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓
веб-сайт	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
мітинг	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
футбол	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
кабінет	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
магазин	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Під час тестування були зафіксовані такі тимчасові результати (рисунок 4.5):

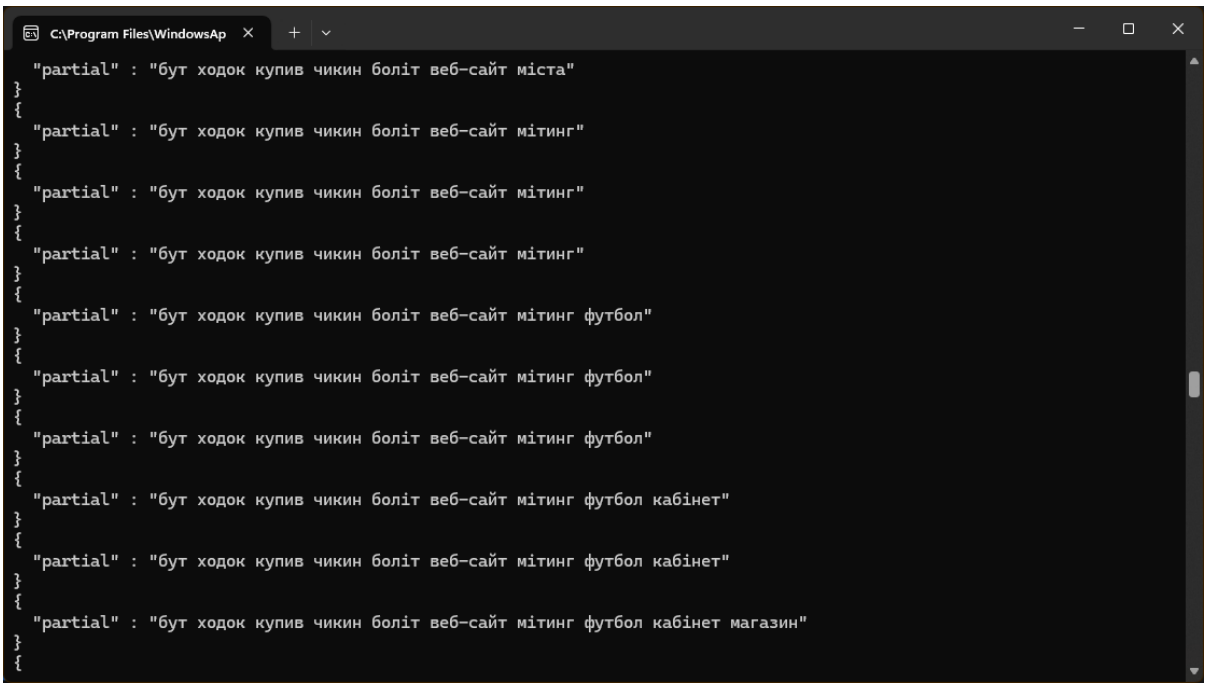

Рисунок 4.5 – Тимчасові результати першого тестування запису

А час (у секундах) розпізнавання всіх слів вийшов таким при різних спробах: 1 – 10.5; 2 – 10.7; 3 – 10.0; 4 – 10.3; 5 – 10.4; 6 – 11.0;7 – 10.6; 8 – 10.1; 9 – 11.1; 10 – 10.2.

За допомогою формул вираховується точність та затримка методу:

$$\text{Accuracy} = \frac{W}{C} * 100\% = 70, 60, 60, 60, 50, 70, 60, 90, 70, 60\ \%.$$

Середньо-статистична точність становить 65% .

$$\text{Delay} = \frac{T}{C} = 1.05, 1.07, 1.00, 1.03, 1.04, 1.10, 1.06, 1.01, 1.11, 1.02.$$

Середньо-статистична затримка становить 1.04.

Для застосунку 2 результати виявилися такими (таблиця 4.2):

Таблиця 4.2 – Результати тестування запису на застосунку 2

слово \ спроба №	1	2	3	4	5	6	7	8	9	10
бот	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
хот-дог	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
гугл	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗
чекін	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
боулінг	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
веб-сайт	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
мітинг	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
футбол	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
кабінет	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
магазин	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Тимчасові результати для цього методу виглядають наступним чином(рисунок 4.6):

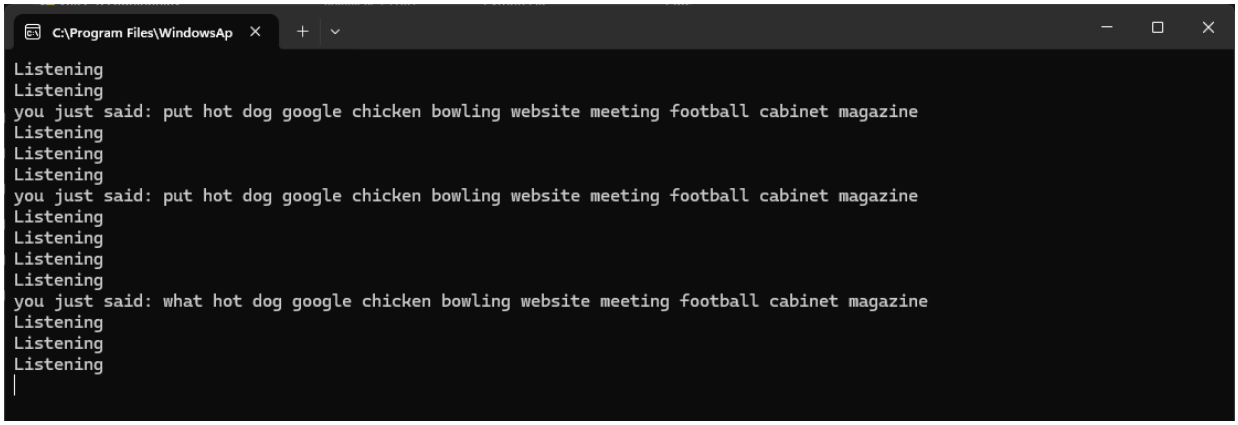

Рисунок 4.6 – Тимчасові результати

Час (у секундах) розпізнавання всіх слів вийшов таким при різних спробах: 1 – 12.1; 2 – 14.4; 3 – 16.0; 4 – 14.6; 5 – 14.6; 6 – 15.2;7 – 14.6; 8 – 15.1; 9 – 15.1; 10 – 14.5.

За допомогою формул вираховується точність та затримка методу:

$$\text{Accuracy} = \frac{W}{C} * 100\% = 80, 80, 70, 80, 80, 80, 80, 80, 90, 80\ \%.$$

Середньо-статистична точність становить 80%.

$$\text{Delay} = \frac{T}{C} = 1.21, 1.44, 1.60, 1.46, 1.46, 1.52, 1.46, 1.51, 1.51, 1.45.$$

Середньо-статистична затримка становить 1.46.

4.3 Розрахунки за умови шуму 1%

На той самий запис було додано монотонний білий шум у кількості 1%. Волноформа запису з шумом 1% виглядає наступним чином (рисунок 4.7):

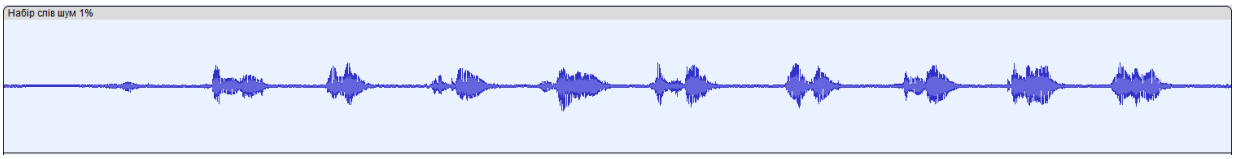

Рисунок 4.7 Волноформа запису з набором слів з шумом 1%

Цей запис було подано на вхід мікрофону, використовуючи застосунки. Отримані результати для застосунку 1(таблиця 4.3).

Час (у секундах) розпізнавання всіх слів вийшов таким при різних спробах: 1 – 11.3; 2 – 11.0; 3 – 11.3; 4 – 10.7; 5 – 11.1; 6 – 11.2; 7 – 11.6; 8 – 12.0; 9 – 11.6; 10 – 11.3.

Таблиця 4.3 – Результати тестування запису на методі 1 в умовах шуму 1%

слово \ спроба №	1	2	3	4	5	6	7	8	9	10
бот	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗
хот-дог	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
гугл	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
чекін	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
боулінг	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
веб-сайт	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
мітинг	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
футбол	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓
кабінет	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
магазин	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

За допомогою формул вираховується точність та затримка методу:

$$\text{Accuracy} = \frac{W}{C} * 100\% = 50, 50, 40, 50, 50, 50, 40, 60, 60, 40.$$

Середньо-статистична точність становить 49%.

$$\text{Delay} = \frac{T}{C} = 1.13, 1.10, 1.13, 1.07, 1.11, 1.12, 1.16, 1.20, 1.16, 1.13.$$

Середньо-статистична затримка становить 1.13.

Результати тестування застосунку (методу) 2 в умовах шуму 1% (таблиця 4.4):

Таблиця 4.4 – Результати тестування запису на методі 2 в умовах шуму 1%

слово \ спроба №	1	2	3	4	5	6	7	8	9	10
бот	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
хот-дог	✓	✓	✓	✗	✓	✗	✓	✗	✗	✓
гугл	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
чекін	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗
боулінг	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗
веб-сайт	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
мітинг	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
футбол	✓	✓	✗	✓	✗	✓	✗	✓	✓	✓
кабінет	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
магазин	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Час розпізнавання: 1 – 14.8; 2 – 15.0; 3 – 14.2; 4 – 15.3; 5 – 15.0; 6 – 15.6; 7 – 14.8; 8 – 16.1; 9 – 17.0; 10 – 15.1.

За допомогою формул вираховується точність та затримка методу:

$$\text{Accuracy} = \frac{W}{C} * 100\% = 70, 80, 70, 60, 60, 60, 70, 60, 60, 70.$$

Середньо-статистична точність становить 66%.

$$\text{Delay} = \frac{T}{C} = 1.38, 1.50, 1.42, 1.53, 1.50, 1.56, 1.48, 1.61, 1.70, 1.51.$$

Середньо-статистична затримка становить 1.51.

4.4 Розрахунки за умови шуму 10%

Для цього тесту на запис було накладено білий монотонний шум у кількості 10%. Волноформа запису виглядає наступним чином (рисунок 4.8):

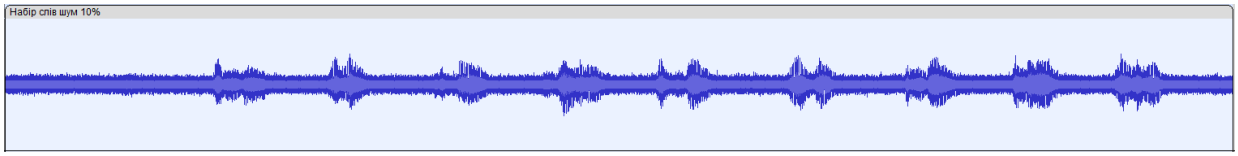

Рисунок 4.8 – Волноформа запису набору слів з шумом 10%

Також само цей запис подається на вхід застосунку 1 та отримані результати сформовані у вигляді таблиці (таблиця 4.5):

Таблиця 4.5 – Результати тестування запису на методі 1 в умовах 10% шуму

слово \ спроба №	1	2	3	4	5	6	7	8	9	10
бот	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
хот-дог	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
гугл	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
чекін	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
боулінг	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
веб-сайт	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
мітинг	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
футбол	✓	✓	✗	✓	✗	✓	✗	✓	✓	✓
кабінет	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓
магазин	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗

Час розпізнавання окремих спроб: 1 – 11.3; 2 – 11.5; 3 – 12.0; 4 – 11.4; 5 – 11.8; 6 – 12.2; 7 – 11.5; 8 – 11.9; 9 – 12.1; 10 – 11.0.

За допомогою формул вираховується точність та затримка методу:

$$\text{Accuracy} = \frac{W}{c} * 100\% = 40, 30, 30, 40, 20, 30, 40, 40, 20, 30\ \%.$$

Середньо-статистична точність становить 32%.

$$\text{Delay} = \frac{T}{C} = 1.13, 1.15, 1.20, 1.14, 1.18, 1.22, 1.15, 1.19, 1.21, 1.10.$$

Середньо-статистична затримка становить 1.16.

Маючи результати усіх трьох тестувань можливо зробити розрахунок надійності для застосунку 1:

$$\text{Reliability} = \frac{A_1+A_2+A_3}{3} = \frac{65+49+32}{3} = 48.6\ \%.$$

Після цього також само запис подається на вхід застосунку 2. Результати у вигляді таблиці (таблиця 4.6):

Таблиця 4.6 – Результати тестування запису на методі 2 в умовах 10% шуму

слово \ спроба №	1	2	3	4	5	6	7	8	9	10
бот	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
хот-дог	✓	✓	✗	✗	✓	✗	✓	✗	✗	✓
гугл	✓	✓	✓	✗	✗	✓	✓	✓	✗	✓
чекін	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
боулінг	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗
веб-сайт	✓	✗	✓	✗	✓	✓	✗	✓	✓	✗
мітинг	✓	✓	✗	✓	✓	✗	✓	✗	✗	✓
футбол	✓	✓	✗	✓	✗	✓	✗	✓	✓	✓
кабінет	✓	✓	✗	✗	✓	✗	✗	✓	✗	✓
магазин	✓	✓	✓	✗	✓	✗	✓	✗	✗	✗

Час розпізнавання окремих спроб: 1 – 16.1; 2 – 15.7; 3 – 17.1; 4 – 17.2; 5 – 16.5; 6 – 15.7; 7 – 16.7; 8 – 16.4; 9 – 15.8; 10 – 17.0.

За допомогою формул вираховується точність та затримка методу:

$$\text{Accuracy} = \frac{W}{C} * 100\% = 80, 70, 40, 30, 50, 40, 50, 40, 30, 40\ \%.$$

Середньо-статистична точність становить 47%.

$$\text{Delay} = \frac{T}{C} = 1.61, 1.57, 1.71, 1.72, 1.65, 1.57, 1.67, 1.64, 1.58, 1.70.$$

Середньо-статистична затримка становить 1.64.

Маючи результати усіх трьох тестувань можливо зробити розрахунок надійності для застосунку 2:

$$\text{Reliability} = \frac{A_1+A_2+A_3}{3} = \frac{80+66+47}{3} = 64{,}3\ \%.$$

4.5 Підсумки

У результаті проведених розрахунків отримані дані, що допомагають побачити різницю між двома розглянутими методами розпізнавання голосу. Ці дані було структуровано у таблицю (таблиця 4.7) та проаналізовано:

Таблиця 4.7 – Підсумкові результати

		1	2
Без шуму	Точність	65%	80%
	Затримка	1.04	1.46
Шум 1%	Точність	49%	66%
	Затримка	1.13	1.51
Шум 10%	Точність	32%	47%
	Затримка	1.16	1.64
Надійність		48.6%	64.3%

Як можна побачити з підсумкової таблиці – Offline метод розпізнавання голосу, реалізований за допомогою хешування та бібліотеки vosk показував нижчу точність, а відповідно й надійність, ніж Online метод, реалізований

шляхом перетворення аудіодоріжки на масив з частотою діскретизації. Якщо буди точним, то на 15.7% нижче.

Однак в той же час він мав набагато меншу затримку. А саме на 38% у середньому результат був кращим у 1 методу.

Виходячи з підсумків – обидва розглянутих методів мають позитивні та негативні результати використання, що говорить про те, що універсального методу, що підходить для всіх випадків, не існує. Тож необхідність ретельного вибору методу залежно від конкретних цілей та завдань є невід'ємною частиною будь-якого проекту, принаймі в рамках розпізнавання голосу.

На основі отриманих результатів було написано рекомендації по використанню кожного методу:

Offline метод:

1. розпізнавання голосу в автономних пристроях:

- у смарт-динаміках, таких як Amazon Echo, у девайсах, що можуть використовуватися автономно, таких як Apple Watch – використовується offline метод розпізнавання голосу для активації пристрою та виконання команд без підключення до інтернету;

- у бортових системах керування автомобілями offline метод розпізнавання голосу може використовуватися для керування навігацією, клімат-контролем та іншими функціями без необхідності підключення до мобільної мережі.

2. розпізнавання голосу в умовах обмеженого доступу до інтернету:

- у віддалених районах з обмеженим доступом до інтернету offline метод розпізнавання голосу може використовуватися для збору даних та спілкування з людьми, які не мають можливості підключитися до мережі;

- у застосунках для військових та правоохоронних органів offline метод розпізнавання голосу може використовуватися для забезпечення безпеки та конфіденційності інформації.

Online метод:

1. розпізнавання мовлення в даний час:

- у системах розпізнавання мовлення для телефонних дзвінків, таких як Google Assistant або Siri, використовується online метод розпізнавання мовлення для перетворення мови на текст у режимі реального часу;

- у системах субтитрування метод розпізнавання мови може використовуватися до створення субтитрів як реального часу;

2. розпізнавання мови у складних акустичних умовах:

- у системах розпізнавання мовлення для шумних середовищ, таких як колл-центри або відкриті вулиці, використовується online метод розпізнавання мовлення, який може адаптуватися до фонового шуму та інших акустичних перешкод;

- у системах розпізнавання мови для багатомовних користувачів online метод розпізнавання мови можна використовувати для розпізнавання мови різними мовами.

Важливо, що вибір методу розпізнавання голосу залежить від конкретних цілей та завдань:

- Offline метод наприклад підходить для випадків, коли потрібна не висока точність та надійність, а коли доступ до інтернету обмежений а також потрібна нижча затримка;

- Online спосіб підходить для випадків, коли потрібна більша точність розпізнавання мовлення а також в режимі сьогодення або у складних акустичних умовах.

ВИСНОВКИ

Було проведено аналіз методів розпізнавання голосу, наочне порівняння, визначення найбільш підходящих для розумного будинку та розробка рекомендацій щодо їх реалізації. Результати можуть бути цінними як для теоретиків, так і для практиків.

Результати, що були отримані під час роботи можна зформувати у такий список:

- спроєктовано та протестовано два різних застосунки, що використовують різні методи розпізнавання голосу;
- проаналізовано та протестовано два методи розпізнавання голосу, а саме:
  - Offline метод розпізнавання голосу, реалізований за допомогою хешування та бібліотеки vosk;
  - Online метод, реалізований шляхом перетворення аудіодоріжки на масив з частотою діскретизації;
- на основі розглянутих методів та розроблених застосунків проведено теоретичні та практичні розрахунки, що надали змогу порівнювати технічні аспекти користання методами;
- на основі отриманих під час розрахунків результатів проведено порівняння методів та зроблено висновок щодо того, що жоден з методів не є універсальним, а підходят кожен для свого спектру використання;
- Надано рекомендації щодо можливих спектрів використання кожного з методів.

Ця кваліфікаційна робота робить внесок у область розпізнавання голосу для систем розумного будинку шляхом порівняльного аналізу різних методів та розробки прототипів застосунків.

Подальші напрями досліджень:

- у майбутньому можна провести більш глибоке вивчення offline- та online- методів розпізнавання, спрямоване на підвищення точності та швидкості роботи;

- планується можливість додання нейронної мережі для навчання розпізнаванню голоса замість вже навчених моделей;

- перспективним напрямом є дослідження методів адаптації онлайн-розпізнавання до індивідуальних особливостей голосу користувача;

- планується розглянути можливість розробки системи на базі придбаних знань, що буде доцільно допомагати користувачам з обмеженими можливостями.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. O. Barkovska, D.Rosinskiy, I.Mikhailov, M.Vintonovych Ensuring Safe Resource Utilization Of Living Space Through Control Of The Microclimate Of A Smart Home //Scientific Bulletin Of Kherson State University. Series «Economic Sciences». – 2023. – №. 49. – С. 5-13. DOI: https://doi.org/10.32999/ksu2307-8030/2023-49-1

2. https://k-call.com/ua/blog/istoriya-razvitiya-i-budushchee-tekhnologii-raspoznavaniya-rechi

3. V.I. Stetsiuk, V.V. Kovalenko Khmelnytskyi National University Methods for processing speech commands of voice control systems – DOI 10.31891/2307-5732-2019-279-6-125-130

4. Xuankai Chang; Brian Yan; Kwanghee Choi; Jee-Weon Jung; Yichen Lu; Soumi Maiti – Exploring Speech Recognition, Translation, and Understanding with Discrete Speech Units: A Comparative Study https://ieeexplore.ieee.org/abstract/document/10447929

5. Mark A. Mandel – A Commercial Large-Vocabulary Discrete Speech Recognition System: DragonDictate https://journals.sagepub.com/doi/abs/10.1177/002383099203500218

6. Zhang, Z.-Q.; Song, Y.; Wu, M.-H.; Fang, X.; McLoughlin, I.; Dai, L.-R. – Cross-Lingual Self-training to Learn Multilingual Representation for Low-Resource Speech Recognition https://doi.org/10.1007/s00034-022-02075-7

7. L Triyono, TR Yudantoro, S Sukamto – Smart home assistant for blind with voice recognition https://doi.org/10.1088/1757-899X/1108/1/012016

8. https://alphacephei.com/vosk/index – Vosk, a speech recognition toolkit

9. CountVectorizer, Python library – https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

10. LogisticRegression, Python library – https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

11. Anthony Zhang, Arvind Chembarpu, Kevin Smith – SpeechRecognition 3.10.4 https://github.com/Uberi/speech_recognition/blob/master/README.rst

12. https://www.audacityteam.org/