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 голосової ідентифікації
в комп’ютерних системах

(тема)

Виконав:

студент II курсу, групи СПм-20-1

Коваль Д.І.
(прізвище, ініціали)

Спеціальність

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

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

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: доц. Іващенко Г.С.
(посада, прізвище, ініціали)

Допускається до захисту

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

2021 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-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Ковалю Дмитру Ігоровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Методи голосової ідентифікації в комп’ютерних системах

затверджена наказом по університету від “ 5 ” листопада 2021 р. № 1657 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2021 р.

3. Вхідні дані до роботи

1) документація мови програмування Python;
2) документація бібліотеки машинного навчання TensorFlow;
3) редактор коду Visual Studio Code.

4. Перелік питань, що потрібно опрацювати у роботі

1) аналіз предметної області;
2) вибір програмного забезпечення;
3) аналіз методів голосової ідентифікації;
4) огляд існуючих досліджень;
5) аналіз результатів дослідження;
6) висновки.

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

Слайд-презентація – 13 слайдів

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	Аналіз способів ідентифікації	09.11.21-12.11.21	
2	Вибір інструментальних засобів	13.11.21-16.11.21	
3	Аналіз методів голосової ідентифікації	17.11.21-22.11.21	
4	Розробка програмного забезпечення	23.11.21-28.11.21	
5	Оформлення результатів кваліфікаційної роботи	29.11.21-07.12.21	
6	Подання кваліфікаційної роботи керівникові та	08.12.21-09.12.21	
	її попередній захист		
7	Подання кваліфікаційної роботи на рецензу-	10.12.21-11.12.21	
	вання		

Дата видачі завдання 8 листопада 2021 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Іващенко Г.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 80 с., 12 рис., 8 табл., 3 дод., 31 джерело.

VISUAL STUDIO CODE, PYTHON, TENSORFLOW, НЕЙРОННІ МЕРЕЖІ, ГОЛОСОВА ІДЕНТИФІКАЦІЯ.

Метою кваліфікаційної роботи є дослідження методів ідентифікації людини за голосом.

У ході виконання кваліфікаційної роботи був проведений аналіз існуючих методів, способи оцінки їх надійності та якості, розглянуті поточні обмеження та проблеми. Було використано мелкепстральні коефіцієнти для опису голосового сигналу для його подальшої обробки. За допомогою Python та TensorFlow був створений програмний продукт, який надає можливість працювати з wav-файлами та проводити аналіз методів голосової ідентифікації. Проведено експериментальне дослідження реалізованого алгоритму ідентифікації за специфічними ознаками голосу.

ABSTRACT

Master’s thesis: 80 pages, 12 figures, 8 table, 3 appendice, 31 sources.

VISUAL STUDIO CODE, PYTHON, TENSORFLOW, NEURAL NETWORKS, VOICE IDENTIFICATION.

The purpose of the qualification work is to study the methods of human identification by voice.

In the course of the qualification work, an analysis of existing methods, ways to assess their reliability and quality, current limitations and problems were considered. Mel-cepstral coefficients were used to describe the voice signal for its further processing. With the help of Python and TensorFlow, a software product was created that provides the ability to work with wav-files and analyze voice identification methods. An experimental study of the implemented identification algorithm based on specific features of the voice was performed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АЧХ – амплітудно-частотна характеристика

ДПФ – дискретне перетворення Фур’є

КС – комп’ютерна система

ПММ – прихована марковська модель

ШНМ – штучна нейронна мережа

CLR – common language runtime

CTC – connectionist temporal classification

DCT – дискретне косинусне перетворення

DL – deep learning

DTW – алгоритм динамічної трансформації часової шкали

MFCC – мелчастотні кепстральні коефіцієнти

ML – machine learning

RMS – root mean square, середнє квадратичне

WER – word error rate, показник помилкового розпізнавання

ВСТУП

Актуальною технічною задачею в галузі телекомунікаційних систем, інтернет-технологій тощо, є забезпечення функцій контролю доступу, що полягають у формуванні дозволу або забороні доступу до певних визначених даних. Такий контроль ґрунтується на ідентифікації суб’єктів, яким потрібен доступ, і об’єкта даних, що є метою доступу. В галузі інформаційної безпеки під ідентифікацією розуміється процедура розпізнавання користувача в системі шляхом сприйняття системою ідентифікаторів користувача, які формуються на основі апріорної інформації про нього.

Особливо поширеним сьогодні є розроблення для задачі ідентифікації особи автоматизованих методів і засобів, що ґрунтуються на оцінюванні її фізіологічних або поведінкових характеристик – засобів біометрії. Це пояснюється їхньою винятковістю та низькою ймовірністю помилки ідентифікації. При цьому, всі методи біометричної ідентифікації можна розділити на статичну і динамічну. До першої групи належать методи ідентифікації за відбитком пальця, формою долоні, розташуванням вен на тильній стороні долоні, сітківкою ока, райдужною оболонкою ока, формою обличчя, термограмою особи тощо. Методи динамічної ідентифікації ґрунтуються на поведінковій характеристиці людини, зокрема ідентифікація проводиться за рукописним чи клавіатурним почерком, голосом, рухом губ тощо. Із усіх зазначених методів біометричної ідентифікації найбільш перспективним при ймовірності відмови у доступі чи помилкової ідентифікації 0,5–5% є метод голосової ідентифікації, якому властива простота технічної реалізації та низька собівартість порівняно з іншими методами отримання біометричних параметрів. Важливою при цьому є задача обґрунтування методу опрацювання голосових сигналів та виділення інформативних ознак, оцінки яких носили б індивідуальний характер та давали б можливість проведення ідентифікації особи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Проблема ідентифікації

Однією з актуальних задач розвитку інформаційних технологій на поточному етапі є забезпечення надійного захисту інформації. Існуючі методи захисту інформації поділяють на апаратні, програмні та змішані. Задача захисту інформації є особливо актуальною в умовах активного розвитку систем банківських операцій та електронної торгівлі, систем дистанційного навчання та великих корпоративних мереж, в яких циркулює конфіденційна інформація.

Одним з головних завдань забезпечення інформаційної безпеки є ідентифікація користувачів інформаційних систем. Основною метою ідентифікації користувача інформаційної системи є зниження певних загроз безпеки інформаційної системи, а власне загрози порушення конфіденційності інформації, загрози порушення цілісності інформації та загрози порушення працездатності системи. Подібні загрози можуть бути викликані різними способами порушень, найбільш поширеним і різноманітним видом порушень є несанкціонований доступ. В даний час загроза несанкціонованого доступу несе найбільшу небезпеку для безпеки інформаційних систем [1], оскільки зловмисник може здійснити незаконне проникнення в інформаційну систему і отримати можливість до реалізації вищеописаних загроз [2].

Процедура ідентифікації користувача являє собою перевірку приналежності суб’єкту доступу пред’явленого їм ідентифікатора і підтвердження його автентичності, іншими словами, пройти перевірку, чи є даний суб’єкт тим, за кого він себе видає.

Способи автентифікації користувачів можна розділити на три типи (за типами автентифікаційної інформації): парольна автентифікація (перевіряється користувач знає якусь унікальну інформацію), автентифікація

з використанням унікального предмета (користувач володіє деяким предметом з унікальними характеристиками), біометрична автентифікація (автентифікаційна інформація є невід'ємною частиною користувача).

Широко розповсюджений парольний захист має певний ряд очевидних недоліків. У разі порушення конфіденційності пароля, що може залишитися непоміченим його власником, відразу порушується захист всієї інформації, до якої власник має доступ. В якості альтернативи парольній системі чи для її доповнення може розглядатися ідентифікація користувачів за біометричними характеристиками [2]. Біометричні технології ідентифікації, автентифікації мають низку переваг перед традиційними і знаходять все більше застосування в комп'ютерних системах. Біометричне підтвердження, а не проста перевірка пароля, який може бути вкрадений, перехоплений або вгаданий, є ключовим при розповсюдженні інтернет-торгівлі [3], створенні нових систем безпеки інформації в корпоративних мережах та системах дистанційного навчання і тестування.

1.2 Основні методи біометричної ідентифікації

Ідентифікувати людину можна за ознаками, пов'язаними з її фізіологічними особливостями, які можуть однозначно ідентифікувати особу. До таких ознак можна віднести: геометричну будову руки, відбитки пальців, райдужну оболонку ока, особливості малюнка сітківки ока, портрет (інфрачервона карта людини), характеристики і особливості мови, клавіатурний та комп'ютерний почерк, рукописний почерк, інші фізіологічні особливості людини, що робить її певною мірою «особливою» [2].

Своєрідність ідентифікації за біометричними параметрами базується на їх винятковості та унікальності [3]. Ймовірність того, що знайдуться дві людини з однаковими ознаками, достатньо мала (ймовірність того, що в двох різних людей на однакових пальцях однієї руки співпадатимуть відбитки пальців, рівна 1 на 24 млн).

Методи біометричної ідентифікації діляться на дві групи:

- статичні методи, які ґрунтуються на фізіологічних характеристиках людини;

- динамічні методи, які ґрунтуються на особливостях поведінки людини – підсвідомих рухах в процесі виконання дії.

Статичні та динамічні методи біометричної ідентифікації являються взаємопов’язаними та взаємодоповнюючими напрямами. Головною перевагою статичних методів біометричної ідентифікації є їх відносна незалежність від психологічного стану користувача, малих затрат зусиль користувача [3, 5] і можливість організації біометричної ідентифікації великих потоків людей. Біометрична ідентифікація на основі динамічних характеристик, зазвичай, простіша в реалізації, оскільки, не вимагає дорогого устаткування і може обмежуватися тільки програмним забезпеченням, яке вимагає мінімальну підтримку спеціаліста в процесі експлуатації.

1.2.1 Статичні методи біометричної ідентифікації

Виділяються такі статичні методи біометричної ідентифікації як розпізнавання за відбитком пальців, розпізнання за формою кисті руки, розпізнавання за портретом, розпізнавання за сітківкою ока.

Розпізнання за формою кисті руки побудовано на розпізнаванні геометрії кисті руки (яка також є унікальною біометричною характеристикою людини) за допомогою спеціальних пристроїв, що дозволяють отримувати тривимірний образ кисті руки (деякі виробники сканують форму декількох пальців). Отримані дані використовують для отримання унікального набору даних, що однозначно ідентифікує людину [6]. Є два основних підходи до використання геометричних характеристик кисті руки. Перший ґрунтується на геометричних характеристиках руки. Другий використовує ще і образні характеристики руки (образи на стиках між фалангами пальців і візерунки кровоносних судин).

Розпізнавання за відбитком пальців. На отриманому зі сканера зображенні відбитків пальців можливо виділити певні характерні ознаки, які надалі можна використовувати в цілях ідентифікації.

На найпростішому технічному рівні, якщо роздільна здатність отриманого зі сканеру зображення складає від 300 dpi до 500 dpi, на поверхні зображення пальця можна виділити досить велику кількість дрібних деталей, за допомогою яких їх можна класифікувати. Зазвичай, в системах ідентифікації використовують всього два типи деталей візерунку:

- кінцеві точки – точки, в яких «виразно» закінчуються папілярні лінії;
- точки розгалуження – точки в яких папілярні лінії роздвоюються.

Розпізнавання за сітківкою ока. Сканування сітківки відбувається з використанням інфрачервоного світла низької інтенсивності, направленого через зіницю до кровоносних судин на задній стінці ока. Ймовірність пропуску незареєстрованого користувача [7] (ймовірність помилки першого роду) при скануванні сітківки ока складає 0,0001%. Ймовірність помилки другого роду достатньо висока – приблизно 0,1%. Пояснюється це тим, що спочатку дані системи були розроблені на замовлення військових [6], де до помилок першого роду пред’являють обмеження. При цьому передбачається, що користувачі можуть повторити процедуру автентифікації кілька разів.

Розпізнавання за портретом. У даному статичному методі ідентифікації будується двовимірний чи тривимірний образ обличчя людини. За допомогою камери і спеціалізованого програмного забезпечення на зображенні або наборі зображень особи виділяються контури губ, носа, брів, очей, обчислюються відстані між ними та інші параметри, залежно від алгоритму, який використовується. За цими даними будується образ, що перетворюється в цифрову форму для порівняння [8]. Кількість, якість і різноманітність образів (різні кути повороту голови, зміни нижньої частини обличчя при вимові ключового слова) варіюється залежно від алгоритмів і функцій системи, що реалізує даний метод.

1.2.2 Динамічні методи біометричної ідентифікації

Серед динамічних методів біометричної ідентифікації виділяють ідентифікацію особи за особливостями голосу, ідентифікацію за клавіатурним почерком, ідентифікацію за почерком миші та інші.

Ідентифікація особи за особливостями голосу має певний ряд переваг. Існує високорозвинена телефонна мережа, звукові карти стали стандартним устаткуванням сучасних персональних комп’ютерів [7]. В якості недоліку біометричних систем ідентифікації особи за голосом необхідно відзначати те, що парольну фразу важко зберегти в таємниці.

Сучасні засоби акустичного прослуховування надають змогу досить успішно здійснювати несанкціоноване копіювання парольної фрази [8]. Як потенційна протидія прослуховуванню використовується комбінування з іншими методами біометричної автентифікації.

Для того, щоб ідентифікувати абонента за голосом, необхідно мати голосовий шаблон, з яким порівнюватиметься голосовий ключ, що вводиться в систему. Порівняння ключа і шаблону може проводитися в цілому або за декількома характеристиками мовленевого сигналу (цифровий голосовий сигнал, що пройшов обробку і адаптований до поставленого завдання): амплітуда і потужність (гучність), частотні (тембр), часові, фазові, енергетичні, характеристики [6].

Для забезпечення простоти аналізу голосовий сигнал попередньо піддають дискретизації з використанням Вейвлет або частотного перетворення.

Ідентифікація користувача комп’ютерної системи може виконуватися за такими ознаками [9]:

- короткочасна енергія сигналу (визначається функцією короткочасної енергії з використанням вікон Хеммінга);
- автокореляційна функція (дозволяє визначити енергію і періодичні властивості сигналу);

- число переходів сигналу через нуль (високі частоти приводять до великого числа переходів через нуль, а низькі – до малого, отже існує зв’язок між числом нульових переходів і розподілом енергії по частотах);
- спектр сигналу;
- коефіцієнти лінійного передбачення;
- кепстральні коефіцієнти;
- кепстральні коефіцієнти, обчислені на основі лінійного передбачення.

Ідентифікація за клавіатурним почерком. У розповсюджених системах захисту інформації доступ здійснюється за допомогою паролів [3]. Якщо пароль збільшити, то при введенні пароля з’являється можливість спостерігати характерний для користувача клавіатурний почерк. Наприклад, як пароль використовується наступна фраза: «звичайний пароль для перевірки». При введенні такої парольної фрази біометрична система фіксує час натиснення кожної клавіші і інтервал часу між натисненням чергової клавіші і відпуском попередньої клавіші.

Важливою характеристикою технології біометричної ідентифікації за клавіатурним почерком є довжина парольної фрази [6]. Зазвичай парольна фраза має бути такою, що легко запам’ятовується і містить від 20 до 40 натискань на клавіші. При створенні парольної фрази допустиме використання слів з сенсом з певного словника. На відміну від класичних паролів, при наборі довгої парольної фрази допустимі помилки в декілька символів, що може погіршувати стійкість парольної фрази до статичного підбору, але значно знижує ймовірність помилок першого роду [9].

Ідентифікація за почерком миші. Існуючі дослідження моніторингу маніпулятора при роботі користувача в системі показують надійність розпізнавання від 80% до 90%. Екран розбивається на зони, в яких курсор миші знаходиться найчастіше і кожні декілька хвилин аналізуються характеристики руху миші між зонами.

При використанні ідентифікації за динамічними характеристиками насамперед необхідно визначити спосіб представлення набору числових

значень [6]. Під час аналізупідпису можна виділити координати характерних точок (екстремуми точки розриву підписів і так далі) та інші параметри траєкторії. Після вибору ключових значень можна почати накопичення бази зразків характеристик користувачів, на підставі порівняння з якими здійснюватиметься ідентифікація (еталонні зразки) [10].

Також необхідно враховувати важливе обмеження: всі траєкторії вважаються осмисленими, тобто користувач їх продукує в процесі повсякденної діяльності і з певною метою – маніпулювання певними елементами управління програмного забезпечення. В такому випадку генеровані траєкторії зумовлені такими факторами: антропологічними, фізіологічними, психологічним [12].

Антропологічні дані людини впливають на таку характеристику, як радіус кривизни траєкторії. Фізіологічні дані людини, наприклад, як структура м’язів плечового і ліктьового суглобів, впливають на швидкість і прискорення руху курсору, тобто динаміку руху. А з іншого боку, психологічні фактори також впливають на зазначені характеристики, додатково вводячи елементи звички при виконанні робочих операцій. Таким чином, приведені фактори вступають у взаємозв’язок між собою і постійно впливають на процес генерування траєкторії [9].

Задача аналізу вказаних траєкторій має спільні сторони із задачею аналізу рукописного тексту чи рукописних підписів. Але комп’ютерна система дає змогу розглянути цей процес в динаміці і скористатися додатковою інформацією про динаміку руху курсору [11].

Однією з важливих переваг динамічних біометричних методів ідентифікації і ідентифікації за динамікою інформаційного почерку, є їх дешевизна і простота реалізації, оскільки в даному випадку не потрібне додаткове дороге устаткування [10], наприклад, у при використанні сканування сітківки ока.

Реалізація такої системи дозволить здійснювати постійний контроль за доступом до певної конфіденційної інформації та ефективно протидіяти

витоку інформації та інформаційному шпигунству. Але, при дешевизні і простоті, ідентифікація за інформаційним підписом має низьку точність розпізнавання.

Особливо необхідно відзначити, що найбільшу ефективність захисту забезпечують такі системи, які поєднують біометричні методи з іншими апаратними засобами автентифікації або декількома різними типами біометричної ідентифікації [11, 13]. Комбінуючи різні способи біометричної і апаратної ідентифікації, можна отримати надійну систему захисту [12].

1.3 Ідентифікація користувачів за голосовим сигналом

Перші напрацювання в галузі голосової ідентифікації були створені у вигляді вирішення задачі розпізнавання мовлення. Вже наприкінці 80-х років XX століття створені перші системи розпізнавання на базі прихованих Марківських моделей (ПММ, Hidden Markov Model – HMM) [2, 3].

Можливість побудови моделей алгоритмів розпізнавання з’явилася відносно недавно, через апаратні обмеження, що були подолані появою нових архітектур у XX-XXI ст.

Вхідними даними виступають голосові команди, представлені у вигляді акустичного аудіо сигналу, що в подальшому має бути трансформований в цифрову форму для забезпечення можливості аналізу за допомогою штучних нейронних мереж (ШНМ) [4].

Задача розпізнавання за класами постала 50 років тому на прикладі системи розпізнавання мовлення «Watermelon», що була здатна до розпізнавання одного слова – watermelon [5]. Дана можливість була надана за рахунок спеціального розміщення голосних у слові, програма мала лише один клас і два випадки, що дозволяло побудувати рішення на повністю аналоговій базі. З цього випливає, що збільшення кількості класів розпізнавання веде до потреби збільшення апаратних потужностей, задіяних в процесі аналізу, а також призводить до збільшення часу, необхідного для аналізу [7].

Наприкінці 90-х років XX століття, існуючі алгоритми та моделі надавали можливість класифікації до 100 слів, що було недостатнім показником для безперешкодного розпізнавання людського мовлення. Задача була непридатною для розв’язання з кількістю класів більше тисячі, і в умовах, коли слова були сказані без пауз. Кожна нова умова – наявність шуму, зміна швидкості мовлення, емоційного забарвлення голосу породжувала створення нових класів розпізнавання і унеможливлювала аналіз в даних апаратних умовах [12].

Нові апаратні засоби надали можливість швидшого розпізнавання та забезпечили системи підтримкою великих словників. Але якість розпізнавання залишалася сталою [13].

Оскільки розпізнавання мовлення не є ключовою задачею роботи, аналіз голосу можна звести до дослідження індивідуальних голосових показників людського індивіда – голосових формант. Форманти – це тембральні показники голосового забарвлення, що є індивідуальними для кожної людини.

1.4 Способи розпізнавання голосового сигналу в комп’ютерних системах

Звук – це фізична вібрація, хвиля, яку треба захопити для подальшого аналізу. Аналіз може бути виконаний після перетворення, що дозволяє розділити сигнал на різні частотні діапазони. Людське вухо здатне виконувати це перетворення за своєю природою. Різні частини внутрішнього вуха реагують на різні частотні діапазони, виконуючи нелінійне перетворення з вхідними звуками [16].

Спираючись на перетворення Фур’є, кожен звук може бути представлений у вигляді комбінації синусоїдальних коливань різної частоти. Кожна синусоїдна хвиля має різну амплітуду і частоту, що змушує звучати її в окремому тоні. Після комбінації хвилі будуть резонувати одна з одною, створюючи піки або ями на частотному діапазоні. Ця характеристика дозволяє

сприймати інформацію шляхом відокремлення різних частотних діапазонів, тому модуль трансформації розробленої комп’ютерної системи повинен мати такі можливості, як аналіз частотного діапазону від 40Гц до 600Гц та розкладання вхідного сигналу на окремі гармоніки.

Отже, розроблена комп’ютерна система вимагає створення функцій для поділу звуків на букви, або послідовності букв, готових до об’єднання в слова. Голосні звуки легше піддаються розпізнаванню, тому перші механізми розпізнавання мови були побудовані на цьому принципі, аналізуючи послідовності голосних [5].

Наступним кроком є розробка динамічного алгоритму, здатного аналізувати частотний спектр у кожний момент часу. Як людина здатна розібрати аналоговий сигнал, комп’ютерна система повинна мати аналогічний спосіб інтерпретації інформації, розділяючи сигнал на дискретні частини для виконання аналізу. Існує загальний конвеєр для систем розпізнавання голосу, описаний на рисунку 1.1.

Мовний сигнал, який надсилається на вхід, може описувати тільки амплітуду [10].

Короткочасні мовні сегменти формуються шляхом дискретизації аналогового сигналу – поділ неперервного звуку на окремі частини малої довжини (10 мс). Всі дискретизовані частини множаться на функцію, яка прибирає частоти, що містять вторинні гармоніки і шум.

Виконання DFT (Discrete Fourier Transform, Дискретне перетворення Фур’є) дає спектральну залежність між амплітудою і частотою для кожної аудіо-частини.

Складання спектральних залежностей дискретних частинок в динамічну картину забезпечує співвідношення між кожною амплітудою частоти і часом, що називається спектрограмою [12].

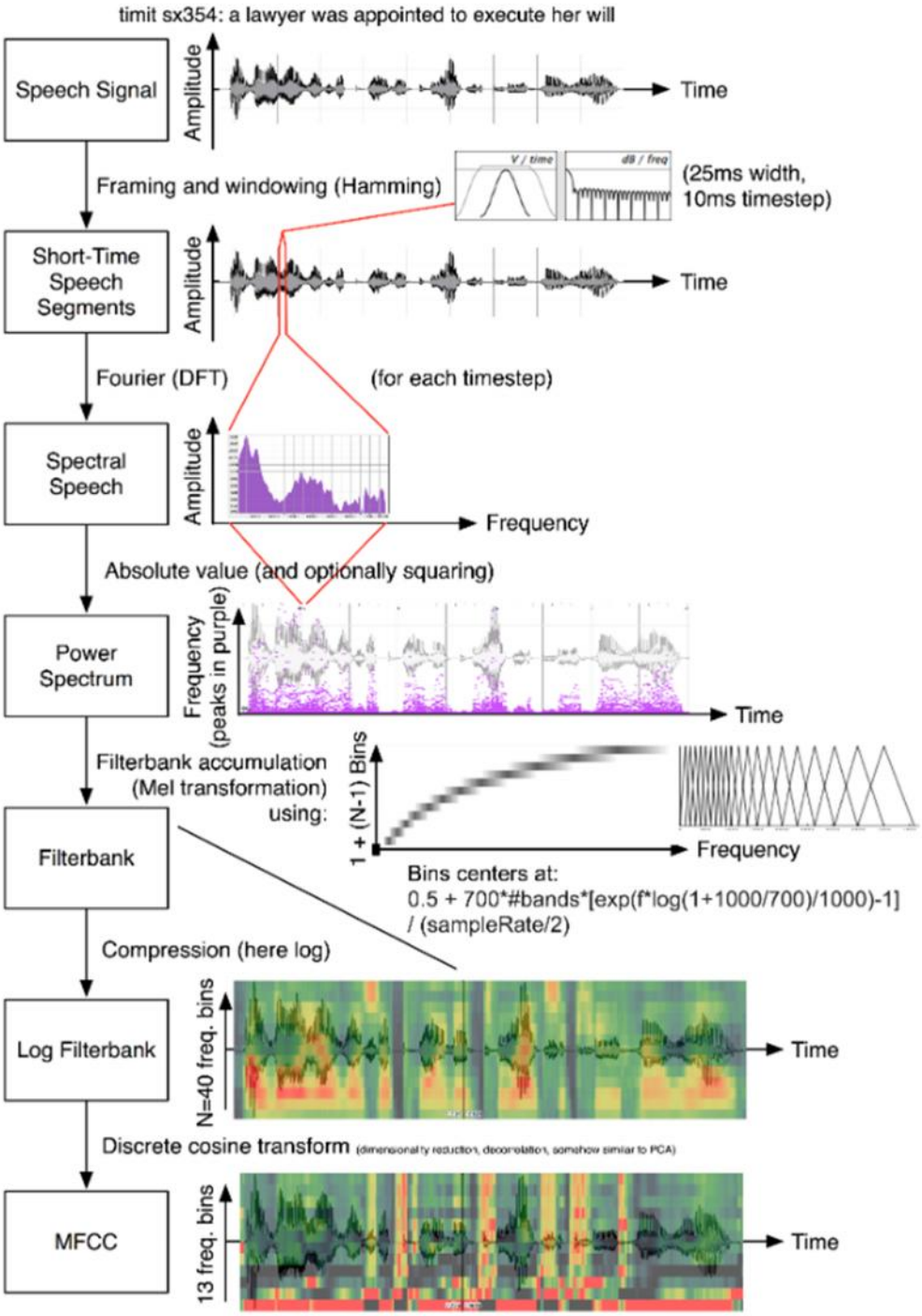

Рисунок 1.1 – Алгоритм трансформації звуку

Вихідний сигнал необхідно відфільтрувати, щоб зробити його впізнаваним ЕОМ, відрізаючи частоти. 3, приблизно, 500 Гц, залишається не більше 50 Гц, стискаючи його необхідним фільтром. Цей етап називається фільтровим накопиченням, або Мел-трансформацією.

Мел-трансформація дозволяє алгоритму стискати вихідний сигнал, зменшуючи величину формуючих частот.

Класичний спосіб побудови алгоритму вимагає застосування MFCC (Mel-frequency cepstral coefficients, Мелчастотні кепстральні коефіцієнти), який стискає звук до 13 формуючих частот, за DCT (Discrete Cosine Transform, дискретне косинусне перетворення), що шукає повторювані гармоніки і видаляє їх, щоб досягти результату в 13 частот.

Отримана інформація готова до аналізу. Але тоді є два способи вирішення завдання, шляхом перетворення звуку в фонеми чи букви [13].

Перший метод передбачує розпізнавання фонем, для побудови потрібного слова. Процес конструкції сам по собі означає порівняння фонем з існуючими словниками, які називаються датасетами. Таким чином, цінність розпізнавання залежить від розміру набору даних, оскільки вона стає більшою, зі збільшенням слів, які готова визначити.

Одними з перших датасетів були:

- TIMIT Database (англ.) – 3 години, 1993;
- 199CMUDict (англ.) – 100 тис. слів, 1993.

Другий метод вимагає розпізнавання букв з аудіо сигналу. Цей метод збільшує розмір словника, що вимагає більшої апаратної потужності для обробки даних. Перші словники без фонетичного аналізу:

- WSJ (англ.) – 81 годин, 1993;
- Switchboard (англ.) – 240 годин, 1993;
- VoxForge (рос.) – 17 годин, 2009;
- Fisher (англ.) – 2000 годин, 2004;
- LibriSpeech (англ.) – 960 годин, 2015;
- Open_TTS (рос.) – більше 3000 годин, 2019.

Перші три датасети складаються з фраз, розділених на слова. Останні три складаються з неподіленої розмовної мови, яка, в першу чергу, має бути обробленою у потрібній формі за допомогою процесу вирівнювання. Воно забезпечує перетворення аудіо сигналу на окремі букви.

Існує декілька алгоритмів вирівнювання (alignment) – алгоритм прямого зворотного зв’язку (Forward-Backward) та алгоритм Вітербі [14]. Їх зміст полягає у створенні ймовірнісної таблиці, що складається з різних букв, і їх відповідності до ймовірності появи в звуковому сигналі (рисунок 1.2).

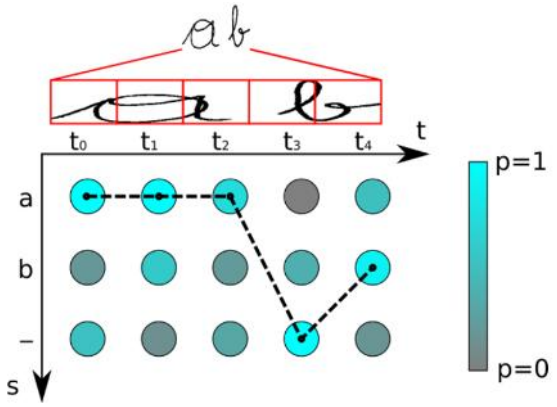

Рисунок 1.2 – Процес вирівнювання

Крім того, існує алгоритм HMM. Але його ефективність нижче, ніж в алгоритму прямого зворотного зв’язку, тому що HMM не може працювати з послідовностями більше 4 звуків, букв, фонем. У зв’язку з недоліком HMM програма більш сумісна з алгоритмом Forward-Backward, що складає матриці ймовірності кожного звуку до кожної букви. Потім алгоритм знаходить найкоротший шлях через матрицю ймовірностей, мінімізуючи необхідні метрики.

Forward-Backward використовується в алгоритмі CTC, який порівнює вихідне слово, передбачене ШНМ, з найбільш схожим словом, закріпленим у наборі даних. Потім алгоритм знаходить найменшу відстань редагування – цифру в клітинках таблиці, яка відповідає кількості перетворень, необхідних для завершення вихідного слова [5]. Щоразу, коли мережа робить помилку у визначенні відстані редагування, система отримує штраф, таким чином забезпечується можливість її самонавчання.

1.5 Постановка задачі

Метою роботи є розробка програмного забезпечення для ідентифікації по голосу. Основною задачею системи є здійснення звукових перетворень для подальшого порівняння отриманих зразків з наборами даних.

Передбачається використання, аналіз та порівняння алгоритмічних методів і нейронних мереж в вирішенні поставленої задачі [17].

Також слід приділити увагу обробці голосу і його інтерпретації, оскільки окрім зовнішніх факторів (шум) існують і персональні фактори людини. Досліджуваний сигнал може бути дуже різноманітним.

Різноманітність виникає з багатьох причин, включаючи з наступних причин:

- відмінності людських голосів;
- рівень мови диктора;
- варіації у вимові;
- варіювання руху артикуляторів (мови, губ, щелепи, піднебіння).

Будь-який сигнал можна представити як вектор в будь-якому параметричної просторі, потім цей вектор може бути використаний в штучних нейронних мережах. Однією з моделей ШНМ, яка навчається без вчителя є карта ознак Кохонена – це система, що самоорганізується. У ній для множини вхідних сигналів формується нейронні ансамблі, що представляють ці сигнали [6]. Цей алгоритм має здатність до статистичного усереднення, тобто вирішується проблема з варіативністю мови. Як і багато інших нейромережевих алгоритмів, він здійснює паралельну обробку інформації, тобто одночасно працюють всі нейрони. Тим самим вирішується проблема зі швидкістю розпізнавання – зазвичай час роботи нейронної мережі складає декілька ітерацій.

2 АЛГОРИТМИ ТА МЕТОДИ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Застосування розподілів мелчастотних кепстральних коефіцієнтів для голосової ідентифікації особистості

Як і будь-який інший біометричний підхід, голосова ідентифікація не є абсолютно надійною. На її якість впливають розташування диктора щодо мікрофона, стан його здоров’я (наявність або відсутність хрипу в голосі), особливості реалізації алгоритмів попередньої обробки сигналу і отримання вектору ознак, що його характеризує, застосовуваний алгоритм ідентифікації. Таким чином, незважаючи на активний розвиток систем голосової ідентифікації, є необхідність в їх поступовому вдосконаленні [8].

Пропонується новий підхід до формування вектору ознак, що описує індивідуальні характеристики голосу диктора [21]. Він заснований на застосуванні виділення корисної інформації про акустичний сигнал, заснованому на обчисленні мелчастотних кепстральних коефіцієнтів (MFCC – Mel-Frequiency Cepstral Coefficients) [1] і побудові їх розподілів для фраз довільної довжини. Відмінною особливістю запропонованого підходу є незалежність отриманого вектору від довжини вихідного голосового зразка, його відносно малий розмір і врахування в ньому індивідуальних характеристик голосового тракту ідентифікованого суб’єкта.

2.2 Отримання вектору ознак на основі MFCC

Система ідентифікації особистості на основі голосових даних реалізується за допомогою таких етапів, як рівень обробки сигналу, рівень моделі, рівень прийняття рішень.

Рівень обробки сигналу. Виділення ознак, істотних для завдання розпізнавання і формування так званого вектору ознак [11].

Рівень моделі. Дозволяє шляхом побудови математичної моделі проводити зіставлення векторів ознак один з одним і обчислювати ступеня подібності між зареєстрованими ознаками і збереженою моделлю.

Рівень прийняття рішень. Забезпечує прийняття кінцевих рішень на основі отриманих ступенів подоб і, якщо необхідно, заданих порогових значень [16].

До теперішнього часу в галузі склався типовий алгоритм попередньої обробки акустичного сигналу після його записи. Цифрований сигнал розбивається на блоки тривалістю 25-30мс. До кожного подібного блоку застосовується вагова функція і потім дискретне перетворення Фур'є. Прикладом ваговій функції може служити вікно Хеммінга:

$$w_n = 0{,}54 - 0{,}46 * \cos\left(2\pi \frac{n}{N-1}\right), \text{ n=0,…,N-1,} \quad (2.1)$$

де N – довжина вікна, виражена у відліках.

Вагова функція використовується для зменшення спотворень в Фур'є аналізі, викликаних кінцівкою вибірки. Тоді дискретне перетворення Фур'є зваженого сигналу можна записати у вигляді:

$$X_k = \sum_{n=0}^{N-1} x_n w_n \exp\left(-\frac{2\pi}{N} k\right), \quad (2.2)$$

де значення індексів k відповідають частотам:

$$f_k = \frac{F_s}{N} k, \quad (2.3)$$

де F_s – частота дискретизації сигналу. Отримане представлення сигналу в частотній області розбивають на діапазони за допомогою банку (гребінки) трикутних фільтрів. Межі фільтрів розраховують в шкалі мел. Переклад в мел-частотну область здійснюється за формулою:

$$B(f) = 1127 * ln\left(1 + \frac{f}{7}\right), \quad (2.4)$$

де N_f – кількість фільтрів (зазвичай використовують близько 24 фільтрів); (f_{lo} , f_{hi_h}) – досліджуваний діапазон частот. Тоді даний діапазон переводиться в шкалу мел, розбивається на N_f рівномірно розподілених діапазонів, що перекриваються, і обчислюють відповідні межі в області лінійних частот [22]. Позначаються через $H_{m,k}$ вагові коефіцієнти отриманих фільтрів. Фільтри застосовуються до квадратів модулів коефіцієнтів перетворення Фур'є. Отримані значення логарифмуються:

$$e_m = ln\left(\sum_{k=0}^{N} |X_k|^2 H_{m,k}\right), m = 0, \dots, N_F - 1. \quad (2.5)$$

Заключним етапом в обчисленні MFCC коефіцієнтів є дискретне косинусне перетворення:

$$c_i = \sum_{m=0}^{N_F - 1} e_m c_{\left(\frac{\pi\,(m+0{,}5)}{N_F}\right)}, i = 1, \dots, N_M \quad , \quad (2.6)$$

де коефіцієнт c_0 не використовується, тому що являє енергію сигналу. Кількість коефіцієнтів N_M на практиці вибирають від 12 до 30. На рисунку 2.1 наведено приклад графіка мел-кепстральних коефіцієнтів.

Для типових акустичних сигналів, що застосовуються при ідентифікації (коротких парольних фраз, наприклад «один, два, три»), кількість блоків розбиття, для яких вираховуються коефіцієнти MFCC, є плаваючою величиною, яка залежить від тривалості проголошення фрази. Відповідні вектори ознак мають також різну довжину і містять близько кількох тисяч кепстральних коефіцієнтів. У деяких випадках до цих даних можуть додаватися також ще і розраховані на основі MFCC значення перших і других похідних по часу, що ще більше збільшує довжини векторів [4].

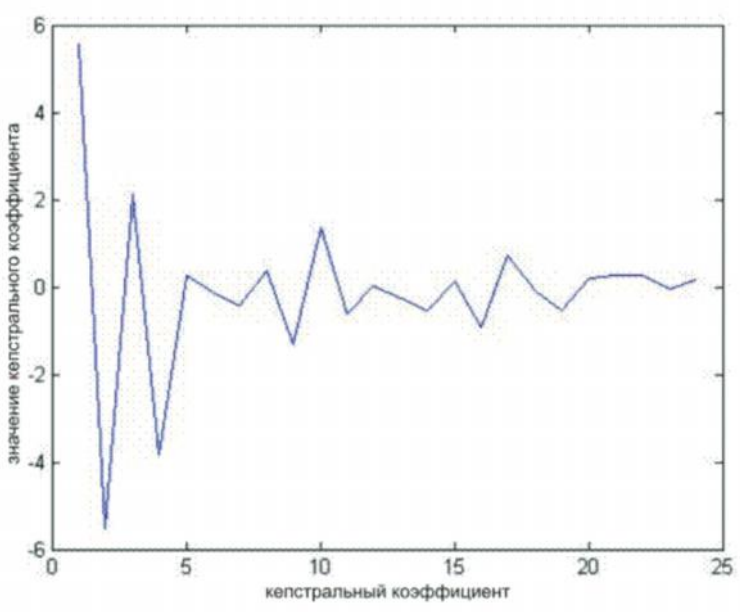

Рисунок 2.1 – Приклад мел-кепстральних коефіцієнтів для фрази «один-два-три»

Існує кілька підходів до фіксації і зменшення довжини результуючого вектору ознак:

- розмір вікна для розбиття сигналу використовувати не фіксованої довжини для всіх зразків, а розбивати їх на фіксовану кількість вікон, довжини, що розраховується для кожного зразка;
- не розбивати сигнал на вікна, а отримувати вектор ознак значень мел-частотних кепстральних коефіцієнтів для всього сигналу. Для всіх зразків довжина вектору ознак буде дорівнює заданій кількості кепстральних коефіцієнтів.
- приводити все зразки на етапі попередньої обробки до однієї довжині. Цей метод є неприйнятним в задачі розпізнавання диктора, так як вносить викривлення у вихідний сигнал [13].

Такий підхід до попередньої обробки акустичних даних має задовільні характеристики і може використовуватися при побудові ефективних систем голосової ідентифікації.

2.3 Використання штучних нейронних мереж

Штучна нейронна мережа – це математична програмна модель, побудована за принципом функціонування біологічних нейронних мереж – мереж нервових клітин живого організму [17]. Працює штучна нейронна мережа наступним чином: на входи нейронів надходять сигнали, які сумуються. При цьому враховується синаптична маса, тобто значимість кожного з входів. Далі, вхідні сигнали одних нейронів надходять на входи інших нейронів. Можливість навчання – одна з головних переваг нейронних мереж перед традиційними алгоритмами [2, 4]. Навчання полягає в знаходженні коефіцієнтів зв’язків між нейронами. В процесі навчання ШНМ здатна виявляти складні залежності між вхідними даними й вихідними, а також здійснювати узагальнення.

Для процесу навчання необхідно мати модель зовнішнього середовища, у якій функціонує нейронна мережа – потрібну для вирішення задачі інформацію [20]. Необхідно визначити, як модифікувати вагові параметри мережі.

Існують три загальні парадигми навчання: «з вчителем», «без вчителя» (самонавчання) та змішана.

У першому випадку нейромережа має у своєму розпорядженні правильні відповіді (виходи мережі) на кожен вхідний приклад. Ваги налаштовуються так, щоб мережа виробляла відповіді найбільш близькі до відомих правильних відповідей.

Навчання без вчителя не вимагає знання правильних відповідей на кожен приклад навчальної вибірки. У цьому випадку розкривається внутрішня структура даних та кореляція між зразками в навчальній множині, що дозволяє розподілити зразки по категоріях.

При змішаному навчанні частина ваг визначається за допомогою навчання зі вчителем, у той час як інша визначається за допомогою самонавчання [26].

Існує декілька видів нейронного зв’язку. Найчастіше використовуються синапсоідальний та ReLU. В першому випадку нейронна мережа використовує дані в діапазоні від -1 до 1 (що фактично відповідає -100% до 100%). В другому вхідні дані передаються через значення 0 та inf (інформація любого характеру).

Алгоритм обчислення з використанням синапсоідальної функції:

- дані поступають на нейрон;
- обчислюється їх вага;
- результати обчислень передаються на наступний нейрон;
- процес повторюється.

Кількість обчислень визначається шляхом встановлення кількості шарів.

Для реалізації процесу порівняння голосових відбитків пропонується використовувати нейромережеве порівняння за допомогою мережі Кохонена, що самоорганізується, так як дана нейронна мережа навчається без учителя, із застосуванням модифікації, що дозволяє прискорити її роботу з мінімальною втратою точності [9].

Навчальна множина для використовуваної штучної нейронної мережі складається лише з вхідних векторів, а навчальний алгоритм підлаштовує ваги мережі так, щоб виходили узгоджені вихідні вектори, тобто щоб пред’явлення досить близьких вхідних векторів давало однакові виходи. Процес навчання, отже, виділяє статистичні властивості навчальної множини і групує подібні вектори в класи. Пред’явлення на вхід вектору з даного класу дасть певний вихідний вектор [3].

Методика роботи нейронної мережі в стандартному варіанті складається з трьох дій – навчання, ідентифікація, верифікація.

Навчання складається з наступних етапів:

- нумерація вхідного вектору;
- подача вектору на вхід мережі;
- обчислення сум в вузлах шару Кохонена;
- активація вузла з сумою 1, якщо такого немає – повтор попереднього пункту;

- корекція ваги вузла за формулою: $w_n = w_o + \alpha(x - w_o)$, де w_n – нове значення ваги; w_o – старе значення; α – швидкість навчання; x – величина входу.

Етап ідентифікації складається з наступних елементів:

- нумерація вхідного вектору;
- подача вектору на вхід мережі;
- обчислення сум в вузлах шару Кохонена;
- активація вузла з сумою 1, якщо такого немає – повтор пункту «с»;
- отримання класифікованого вектору.

Верифікація складається з наступних елементів:

- знаходження середньоквадратичного відхилення між точками введеного вектору і наявного в пам'яті (класифікованого);
- порівняння отриманих значень відхилення з граничним значенням.

Зважаючи на велику кількість ітерацій, обчислення займають тривалий час. У методиці «SOM: Leading Neurons» пропонується скоротити кількість ітерацій шляхом вибору лідируючого нейрона. Таким чином, спрацьовує нейрон, для якого вектор входу найближче до вектору ваг зв'язків $\Delta\ (x) = \sum_{j=1}^{n} w_j x^j - w_o - \alpha\left(x^j - w_o\right)$, де $\Delta\ (x)$ – відстань між векторами входу і вектором ваг зв'язків; w_j – нове значення ваги; $x = (x^1, \dots, x^n)$ – n-мірний вектор входу; w_o – початковий поріг активації; α – швидкість навчання.

В якості активаційної функції використовується сигмоїда, яка має вигляд $f(x) = \frac{1}{1+e^{-\alpha * x}}$, де α – параметр нахилу.

Таким чином, під час дослідження латеральний підхід був замінений на активацію лідируючих нейронів.

2.4 Мова програмування Python

Python – це високорівнева, інтерпретована й універсальна динамічна мова програмування, яка зосереджена на читабельності коду. Синтаксис в

Python допомагає програмістам писати код за меншу кількість кроків у порівнянні з Java або C++ [21]. Досить велике поширення Python отримав у галузі машинного навчання та досліджень штучного інтелекту.

Основні особливості мови програмування Python:

- легко писати код. Це мова програмування високого рівня, вона дуже легко вивчається в порівнянні з іншими мовами, такими як C, C#, Java тощо;
- підтримка об’єктно-орієнтованої та функціональної парадигм;
- інтерпретація програм. Для роботи зі скриптами необхідний інтерпретатор, який запускає та виконує скрипт (рисунок 2.2);
- портативність і платформонезалежність;
- автоматичне керування пам’яті;
- підтримка програмування GUI;
- велика стандартна бібліотека
- динамічна типізація.

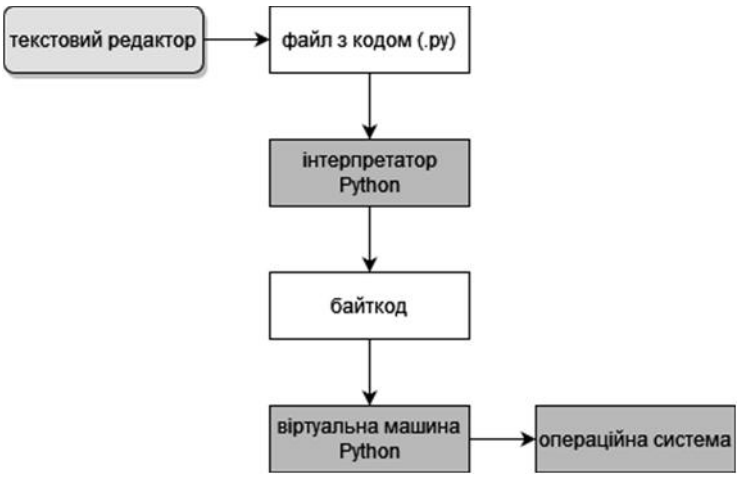

Рисунок 2.2 – Процес виконання програми

Хоча формально трансляція інтерпретатором вихідного коду в байткод і переведення байткоду віртуальною машиною в набір машинних команд це два різні процеси, але вони об’єднані в самому інтерпретаторі.

2.5 Бібліотека TensorFlow

TensorFlow – це бібліотека програмного забезпечення з відкритим вихідним кодом. Спочатку вона була розроблена дослідниками та інженерами, які працювали в команді Google Brain Team для цілей проведення машинного навчання та досліджень глибоких нейронних мереж, але система є достатньо загальною, для застосування в багатьох інших областях.

TensorFlow використовується для збору наборів даних, навчальних систем за допомогою моделей і надання результатів на їх основі.

Основні особливості бібліотеки TensorFlow:

- легка розробка моделей. TensorFlow підтримує високорівневі API, за допомогою яких можна легко створювати моделі машинного навчання за допомогою нейронних мереж
- можна виконувати складні числові обчислення. оскільки набір вхідних даних величезний, математичні обчислення/розрахунки можна зробити легко;
- містить API машинного навчання: TensorFlow багатий на API машинного навчання як низького, так і високого рівня;
- легке розгортання та обчислення за допомогою CPU або GPU. Обчислення можна виконувати як на ЦП, так і на графічному процесорі;
- містить попередньо навчені моделі та набори даних. Google включив багато наборів даних і попередньо навчених моделей у TensorFlow. Набори даних включають mnist, vgg_face2, ImageNet, coco тощо;
- попередньо підготовлені моделі для мобільних пристроїв та вбудованих пристроїв. Моделі машинного навчання можна розгорнути на мобільних і вбудованих пристроях за допомогою TensorFlow. Попередньо підготовлені моделі можна безпосередньо використовувати для виробництва;
- tensorboard, набір із використанням набору інструментів візуалізації. Tensorboard – це набір інструментів візуалізації TensorFlow, який використовується для відображення зображень, графіків тощо;

- підтримка Keras. Keras – це високорівневий API TensorFlow, який побудований на основі TensorFlow і Theano. Зараз Keras став популярним як широко використовуваний API TensorFlow;

- відкритий код. TensorFlow – це платформа з відкритим вихідним кодом, безкоштовна у використанні та дозволяє розробникам і дослідникам створювати та розгортати моделі машинного навчання.

2.6 Формат голосових файлів WAV

Людське мовлення – це послідовність звуків. Звук – це набір окремих синусоїдних хвиль різних частот, які резонують між собою в один момент часу [28], згідно з перетворенням Фур'є. Хвиля – це механічне коливання з певними атрибутами, як амплітуда і частота. Комп'ютерні системи потребують перетворення звуків у спеціальний, доступний тип даних (рисунок 2.3).

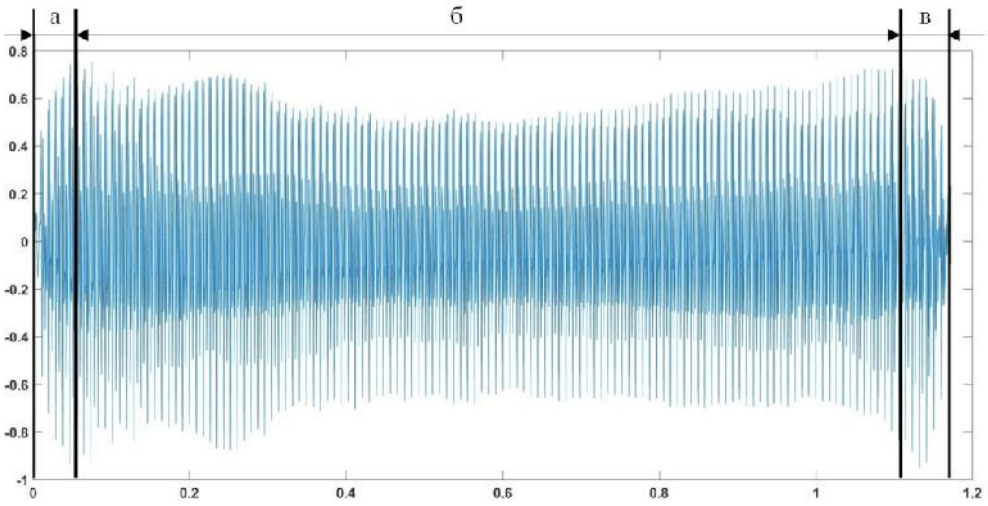

Рисунок 2.3 – Амплітудно-часова характеристика людського голосу [28]

Людський голос в контексті комп'ютерної системи виражений як спеціальний звуковий тип даних WAV, що є стандартним при роботі з аудіо інформацією (рисунок 2.4).

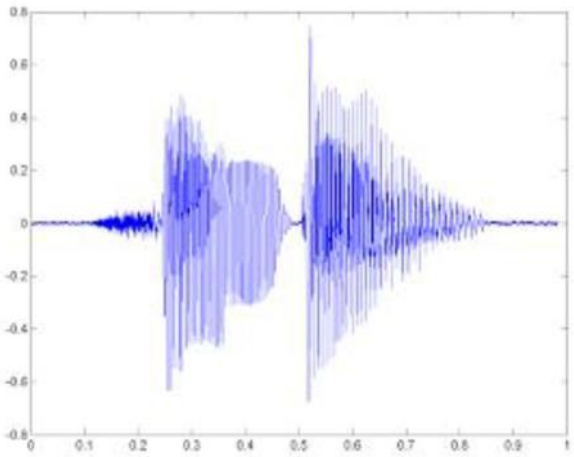

Рисунок 2.4 – Амплітудно-частотна характеристика людського голосу

WAV – формат аудіофайлу, розроблений компаніями Microsoft та IBM (додаток Б). WAVE базується на форматі RIFF, поширюючи його на інформацію про такі параметри аудіо, як застосований кодек, частота дискретизації та кількість каналів (рисунок 2.5). WAV як і RIFF передбачався для комп'ютерів IBM PC, тому всі змінні записані у форматі little endian.

Формат відображує, що у файлі є два блоки. Перший блок являє собою заголовок з інформацією про аудіопотоки: бітрейт, частота, кількість каналів, довжина файлу і т.д. Другий блок складається з даних цифрового сигналу, набору значень амплітуд [22].

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Попередня обробка даних

Загальна логіка попередньої обробки вихідних даних:

- зчитування заголовку;
- перевірка обмежень (стиснення, розбиття та інше);
- збереження даних у виділеному масиві (лістинг 3.1).

Лістинг 3.1 – Читання файлу WAV

```
def wavefile_to_waveform(wav_file, features_type):
    data, sr = sf.read(wav_file)
    if features_type == 'vggish':
        tmp_name = str(int(np.random.rand(1)*1000000)) + '.wav'
        sf.write(tmp_name, data, sr, subtype='PCM_16')
        sr, wav_data = wavfile.read(tmp_name)
        os.remove(tmp_name)
        data = wav_data / 32768.0
        src_repeat = data
    return data, sr
```

Далі, до голосового сигналу застосовується процедура дискретизації (рисунок 3.1).

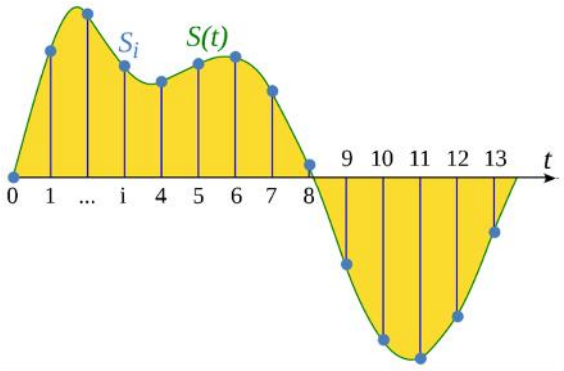

Рисунок 3.1 – Дискретизація сигналу

Для зчитування даних з файлу використовується модуль scipy.io.

Дискретизація необхідна для забезпечення подальшого аналітичного процесу (лістинг 3.2).

Лістинг 3.2 – Фрагмент реалізації етапу дискретизації з бінарним пошуком

```
def binm(rr,ra):
  ih=len(ra)-1
  il=0
  if rr<ra[il]: return il
  while (ih-il>1):
    ie=(ih+il)/2
    if rr<ra[ie]:
      ih=ie
    else:
      il=ie
  return ih

bins = [20,30,40]
results = [0,0,0,0]

for _ in range(iterations):
  x = somefunction()
  ib=binm(x,bins)
  results[ib]+=1
```

Дані поділяються на короткі інтервали часу – фрейми. Фрейми не повинні слідувати один за одним, вони повинні «перекриватися». Тобто кінець одного кадру повинен накладатися на початок першого (лістинг 3.3).

Лістинг 3.3 – Розбиття на фрейми

```
from pydub import AudioSegment
import math

class SplitWavAudioMubin():
    def __init__(self, folder, filename):
        self.folder = folder
        self.filename = filename
        self.filepath = folder + '\\' + filename

        self.audio = AudioSegment.from_wav(self.filepath)
```

```
    def get_duration(self):
        return self.audio.duration_seconds

    def single_split(self, from_min, to_min, split_filename):
        t1 = from_min * 60 * 1000
        t2 = to_min * 60 * 1000
        split_audio = self.audio[t1:t2]
        split_audio.export(self.folder + '\\' + split_filename,
format="wav")
    def multiple_split(self, min_per_split):
        total_mins = math.ceil(self.get_duration() / 60)
        for i in range(0, total_mins, min_per_split):
            split_fn = str(i) + '_' + self.filename
            self.single_split(i, i+min_per_split, split_fn)
            print(str(i) + ' Done')
            if i == total_mins - min_per_split:
                print('All splited successfully')
```

Застосування даної процедури надає можливість аналізу сигналу на інтервалах, замість його дослідження в конкретній точці.

Наступною проблемою є поділ сигналу на слова, задля виключення із подальшого аналізу фрагментів тиші між словами (лістинг 3.4). Заради спрощення, інтервали мовчання вважатимуться роздільниками слів [23]. В даному випадку, поріг між роздільником і словом задається величиною амплітуди сигналу. Математичне подання порогу – ентропія, що є величиною коливання конкретного кадру.

Лістинг 3.4 – Поділ сигналу на слова

```
from pydub import AudioSegment
from pydub.silence import split_on_silence

sound_file = AudioSegment.from_wav("a-z.wav")
audio_chunks = split_on_silence(sound_file,
    # must be silent for at least half a second
    min_silence_len=500,
    # consider it silent if quieter than -16 dBFS
    silence_thresh=-16
)
for i, chunk in enumerate(audio_chunks):
    out_file = ".//splitAudio//chunk{0}.wav".format(i)
    print "exporting", out_file
    chunk.export(out_file, format="wav")
```

Для розрахунку порогу, в кожному кадрі виконуються наступні дії:

- нормування сигналу – інтерпретація сигналу в діапазоні від -1 до 1;
- побудова гістограми сигналу;
- розрахунок ентропії за наступною формулою:

$$E = \sum_{i=0}^{N-1} P[i] * \log_2(P[i]), \quad (3.1)$$

де N – кількість фреймів. Задля відокремлення інформативної складової сигналу від тиші, обчислюється поріг ентропії (лістинг 3.5), як середнього між максимальними і мінімальними значеннями всіх кадрів. Числовою характеристикою якості ідентифікації обрано RMS – середній квадрат всіх значень.

Лістинг 3.5 – Розрахунок ентропії

```
def entropy1(labels, base=None):
  value,counts = np.unique(labels, return_counts=True)
  return entropy(counts, base=base)
def entropy2(labels, base=None):
  n_labels = len(labels)
  if n_labels <= 1:    return 0
  value,counts = np.unique(labels, return_counts=True)
  probs = counts / n_labels
  n_classes = np.count_nonzero(probs)
  if n_classes <= 1:    return 0
  ent = 0.
  base = e if base is None else base
  for i in probs:
    ent -= i * log(i, base)
  return ent
def entropy3(labels, base=None):
  vc = pd.Series(labels).value_counts(normalize=True,
sort=False)
  base = e if base is None else base
  return -(vc * np.log(vc)/np.log(base)).sum()
def entropy4(labels, base=None):
  value,counts = np.unique(labels, return_counts=True)
  norm_counts = counts / counts.sum()
  base = e if base is None else base
  return -(norm_counts * np.log(norm_counts)/np.log(base)).sum()
```

3.2 Використання мелчастотних коефіцієнтів в штучних нейронних мережах

Наступною складовою обробки вихідних даних є перетворення сигналу обрано MFCC перетворення, що відображає інтенсивність сигналу.

Переваги такого методу:

- використання спектру сигналу (розкладання по базису ортогональних синусоїдальних функцій), що дозволяє проводити подальший спектральний аналіз;
- відображення спектру на спеціальну мел-шкалу (лістинг 3.6), задля виділення та підсилення головних гармоній, потрібних для сприйняття людського мовлення [24];
- здатність до регуляції обчислюваних коефіцієнтів, кількість яких може бути обмежена до будь якого значення, що дозволяє «стиснути» кадр, і як наслідок відсіяти побічні гармоніки.

Лістинг 3.6 – Отримання спектрограми MFCC

```
def extract_features(file_name):    audio, sample_rate = 
librosa.load(file_name, res_type='kaiser_fast')
    mfccs = librosa.feature.mfcc(y=audio, sr=sample_rate, 
n_mfcc=40)
    mfccs_processed = np.mean(mfccs.T,axis=0)

    return mfccs_processed
features = []
for index, row in metadata.iterrows():

    file_name = 
os.path.join(os.path.abspath(fulldatasetpath),'fold'+str(row["fo
ld"])+'/',str(row["slice_file_name"]))

    class_label = row["class"]
    data = extract_features(file_name)
    features.append([data, class_label])
featuresdf = pd.DataFrame(features, columns=['feature', 'fold'
,'class_label'])
featuresdf.head()
```

Процес обчислення коефіцієнтів MFCC для певного кадру (фрейму). Фрейм зображується у вигляді вектору $x[k],0 \leq k < N$, де N – розмір фрейму.

За допомогою дискретного перетворення Фур'є (ДПФ) відбувається розрахунок спектру сигналу:

$$X[k] = \sum_{n=0}^{N-1} x[n] * e^{\frac{-2*\pi*i*k*n}{N}} = 0, 0 \leq k < N. \quad (3.2)$$

Застосовується функція Хеммінга (лістинг 3.7) задля «згладжування» результатів по краях кадрів:

$$H[k] = 0.54 - 0.46 * cos\left(\frac{2*\pi*k}{N-1}\right). \quad (3.3)$$

Лістинг 3.7 – Реалізація функції Хеммінга

```
def _mfcc_and_labels(audio, labels):
  mfcc_sample_rate = 100.0
  winfunc = lambda x: np.hamming(x)
  mfcc_features = python_speech_features.mfcc(audio,
samplerate=timit.SAMPLE_RATE, winlen=0.025,

winstep=1.0/mfcc_sample_rate, lowfreq=85.0,

highfreq=timit.SAMPLE_RATE/2, winfunc=winfunc)
  t_audio = np.linspace(0.0, audio.shape[0] * 1.0 /
timit.SAMPLE_RATE, audio.size, endpoint=False)
  t_mfcc = np.linspace(0.0, mfcc_features.shape[0] * 1.0 /
mfcc_sample_rate, mfcc_features.shape[0], endpoint=False)
  interp_func = scipy.interpolate.interp1d(t_audio, labels,
kind='nearest')
  mfcc_labels = interp_func(t_mfcc)
  return mfcc_features, mfcc_labels
```

За формулою 3.4, розраховується результуючий вектор, з якого формується спектрограма вхідного сигналу (рисунок 3.2).

$$X[k] = X[k] * H[k], 0 \leq k < N. \quad (3.4)$$

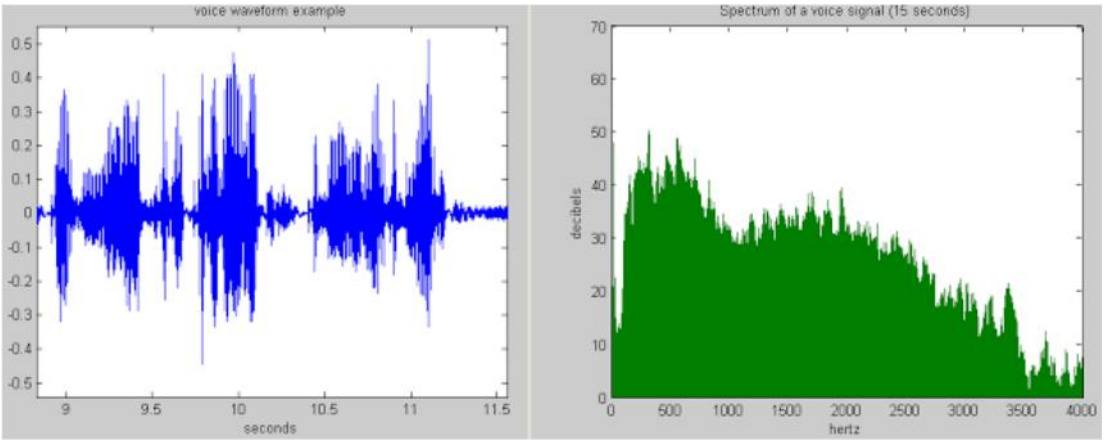

Рисунок 3.2 – Трансформація вхідного сигналу в амплітудно-частотну характеристику

Наступним кроком є підрахунок мел-фільтрів. Мел є «психофізичною одиницею висоти звуку» [25], заснована на суб'єктивному сприйнятті середньостатистичної людини. Мел залежить від частоти (рисунок 3.3), гучності і тембру звуку. Таким чином, це значення, що відображає цінність окремого звуку в певному частотному діапазон.

Мел-трансформація виконується наступним чином:

$$M = 1127 * ln\ (1 + \frac{F}{7}), \tag{3.5}$$

де F – частота. Для зворотного перетворення можна використати:

$$F\ = 700 * \left(e^{\frac{M}{1}} - 1\right), \tag{3.6}$$

де M – значення мел-коефіцієнту. За наявності сигналу довжиною 256 фреймів, рекомендована кількість мел-коефіцієнтів становить 10, так як частота дискретизації сигналу має стандарт 16000 Гц, а людська мова лежить в діапазоні [300; 8000] Гц.

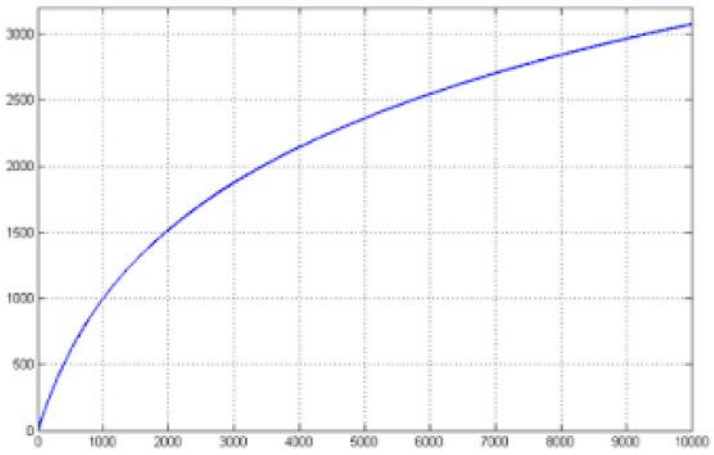

Рисунок 3.3 – Графік залежності мел-шкали від частоти

Для того, щоб розкласти діапазон від 300 до 8000 Гц у мел-масштабі, необхідно створити «гребінь» фільтрів (лістинг 3.8).

Лістинг 3.8 – Розрахунок мел-фільтрів

```
low_freq_mel = 0
high_freq_mel = (2595 * numpy.log10(1 + (sample_rate / 2) /
700))  # Convert Hz to Mel
mel_points = numpy.linspace(low_freq_mel, high_freq_mel, nfilt +
2)  # Equally spaced in Mel scale
hz_points = (700 * (10**(mel_points / 2595) - 1))  # Convert Mel
to Hz
bin = numpy.floor((NFFT + 1) * hz_points / sample_rate)

fbank = numpy.zeros((nfilt, int(numpy.floor(NFFT / 2 + 1))))
for m in range(1, nfilt + 1):
    f_m_minus = int(bin[m - 1])   # left
    f_m = int(bin[m])             # center
    f_m_plus = int(bin[m + 1])    # right
    for k in range(f_m_minus, f_m):
        fbank[m - 1, k] = (k - bin[m - 1]) / (bin[m] - bin[m -
1])
    for k in range(f_m, f_m_plus):
        fbank[m - 1, k] = (bin[m + 1] - k) / (bin[m + 1] -
bin[m])
filter_banks = numpy.dot(pow_frames, fbank.T)
filter_banks = numpy.where(filter_banks == 0,
numpy.finfo(float).eps, filter_banks)  # Numerical Stability
filter_banks = 20 * numpy.log10(filter_banks)
```

Кожен мел-фільтр є трикутною віконною функцією, що дозволяє підсумувати величину інтенсивності в певному діапазоні частот [26] і таким чином отримати мел-коефіцієнт. На основі кількості мел-коефіцієнтів і проаналізованого діапазону частот, будується набір фільтрів (рисунок 3.4).

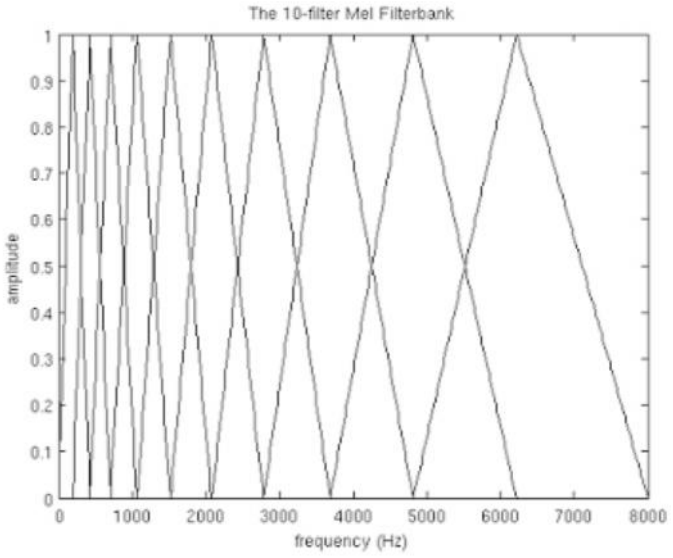

Рисунок 3.4 – Каскад мел-фільтрів

Порядковий номер мел-коефіцієнта визначає ширину базису його фільтра: зі збільшенням номера серії збільшується ширина базису. Інформативний частотний діапазон знаходиться в межах [300; 8000] Гц. Застосовуючи формулу 3.5, на мел-шкалі даний діапазон перетворюється на [401,25; 2834.99].

Далі відбувається побудова фільтрів за опорними точками $m[i] = [401.25, 621.50, 842.75, 1063.00, 1226.25, 1507.50, 1728.74, 1950.09, 2160.74, 2302.39, 2623.54, 2834.99]$. Зворотне перетворення за формулою 3.6 є відображенням даної шкали в частотному діапазоні $[300, 516.33, 761.90, 1122.96, 1495.04, 1983.32, 2574.53, 3231.52, 4131.13, 5160.56, 6445.8, 800]$.

Як видно з рисунку 3.4, збільшення ширини базису фільтра відбувається задля вирівнювання динаміки зростання цінності на різних частотних діапазонах.

Знаючи довжину спектра, його частоту дискретизації, вираховується опорні точки за наступною формулою:

$$f(i) = f \quad ((f \quad + 1) * sa \quad), \tag{3.7}$$

де frameSize – розмір фрейму. Визначено набір опорних точок 4,8,12,17,23,31,40,52,66,82,103,128. Отримавши опорні точки, з'являється можливість побудови необхідних фільтрів:

$$H_m(k) = \begin{cases} 0 & k < f(m-1), k > f(m+1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)}, & f(m-1) \le k \le f(m) \\ \frac{f(m+k) - k}{f(m+k) - f(m)} & f(m) \le k \le f(m+1) \end{cases}. \tag{3.8}$$

Застосування фільтра передбачає попарне множення його значення з значеннями діапазону (лістинг 3.9).

Лістинг 3.9 – Розрахунок мел-коефіцієнтів

```
(nframes, ncoeff) = mfcc.shape
n = numpy.arange(ncoeff)
lift = 1 + (cep_lifter / 2) * numpy.sin(numpy.pi * n /
cep_lifter)
mfcc *= lift
```

Результатом цієї операції є мел-коефіцієнт. Кількість коефіцієнтів дорівнює кількості фільтрів: наступною формулою:

$$S[m] = lo \quad (\sum_{k=0}^{N-1} |X[k]|^2 * H_m[k]),\ 0 \le m < M. \tag{3.9}$$

Однак, необхідно застосовувати мел-фільтри не до значень спектра, а до його інтенсивності, після чого проводиться логарифмування отриманих результатів. Таким чином зменшується чутливість мел-коефіцієнтів до шуму.

Лістинг 3.10 – Застосування дискретного косинусного перетворення

```
def amplitudes_at_frequencies(freqInds, timeseries, times=None,
transform='dct'):
    amplitudes = {}
    for o in timeseries.keys():
        if transform == 'dct':
            temp = _dct(timeseries[o], norm='ortho')[freqInds] /
_np.sqrt(len(timeseries[o]) / 2)
            if 0. in freqInds:
                temp[0] = temp[0] / _np.sqrt(2)
            amplitudes[o] = list(temp)
        else:
            raise NotImplementedError("This function only
currently works for the DCT!")
    return amplitudes
```

Дискретне косинусне перетворення (DCT) (лістинг 3.10) використовується для отримання тих же «кепстральних» коефіцієнтів. Його значення полягає в «стисканні» отриманих результатів [27], а отже, збільшенні цінності перших коефіцієнтів і зменшення цінності останніх.

$$C[l] = \sum_{m=0}^{M-1} S[m] * c \quad (\pi * l * \frac{m+\frac{1}{2}}{M})),\ 0 \leq l\ < M. \qquad (3.10)$$

Тепер, відповідність між кожним кадром і значенням MFCC досягнута для проведення подальшого аналізу. На даному етапі досягнута відповідність між динамічною послідовністю кадрів, та мел-значенням кожного з них.

Отримані мелкепстральні коефіцієнти можливо використовувати для ідентифікації диктора, наприклад, з використання штучних нейронніх мереж, що дає можливість ШНМ аналізувати вхідний сигнал.

3.3 Порівняння за допомогою штучних нейронних мереж

Для реалізації процесу порівняння голосових відбитків пропонується використовувати нейромережеве порівняння, із застосуванням модифікації, що дозволяє прискорити роботу ШНМ з мінімальною втратою точності.

Навчальна множина для нейронної мережі складається лише з вхідних векторів, а навчальний алгоритм підлаштовує ваги нейронів мережі так, щоб виходили узгоджені вихідні вектори, тобто, щоб передача досить близьких вхідних векторів призводила до однакових виходів.

Процес навчання, таким чином, виділяє статистичні властивості навчальної множини та групує подібні вектори до класів. Передача на вхід вектору даного класу дасть певний вихідний вектор (лістинг 3.11).

Лістинг 3.11 – Попередня обробка даних датасетів

```
data = pd.read_csv('dataset.csv')
data.head()# Dropping unneccesary columns
data = data.drop(['filename'],axis=1)#Encoding the Labels
genre_list = data.iloc[:, -1]
encoder = LabelEncoder()
y = encoder.fit_transform(genre_list)#Scaling the Feature
columns
scaler = StandardScaler()
X = scaler.fit_transform(np.array(data.iloc[:, :-1], dtype =
float))#Dividing data into training and Testing set
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)
```

Зважаючи на велику кількість ітерацій, обчислення займають тривалий час. У методиці «SOM: Leading Neurons» [13] пропонується скоротити кількість ітерацій шляхом вибору лідируючого нейрона.

Таким чином, спрацьовує нейрон, для якого вектор входу найближче до вектору ваг зв’язків:

$$\Delta \ (x) = \sum_{j=1}^{n} w_j x^j - w_o - \alpha\left(x^j - w_o\right), \tag{3.11}$$

де $\Delta \ (x)$ – відстань між векторами входу і вектором ваг зв’язків; w_j – нове значення ваги; $x = (x^1, \dots, x^n)$ – n-мірний вектор входу; w_o – початковий поріг активації; α – швидкість навчання (лістинг 3.12).

Лістинг 3.12 – Побудова штучної нейронної мережі

```
model = Sequential()
model.add(layers.Dense(256, activation='relu',
input_shape=(X_train.shape[1],)))
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10,
activation='softmax'))model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

В якості активаційної функції використовується сигмоїда:

$$f(x) = \frac{1}{1+e^{-\alpha * x}}, \tag{3.12}$$

де α – параметр нахилу.

Процес налаштування штучної нейронної мережі виконується за допомогою функції fit (лістинг 3.13).

Лістинг 3.13 – Підгонка моделі

```
classifier = model.fit(X_train,
                   y_train,
                   epochs=100,
                   batch_size=128)
```

3.4 Прихована марковська модель

Прихована марковська модель – це імовірнісна модель, яка використовується для пояснення чи виведення ймовірнісної характеристики будь-якого випадкового процесу. В основному це говорить про те, що спостережувана подія не буде відповідати її покроковому статусу, а буде пов’язана з набором розподілів ймовірностей. Система, яка моделюється, вважається ланцюгом Маркова, і в цьому процесі є деякі приховані стани. У

цьому випадку можна сказати, що приховані стани – це процес, який залежить від основного марковського процесу (рисунок 3.5).

Основна мета ПММ – отримати ланцюг Маркова, спостерігаючи за його прихованими станами. Розглядаючи марковський процес X із прихованими станами Y, ПММ затверджує, що для кожної позначки часу розподіл ймовірності Y не повинен залежати від історії X відповідно до цього часу.

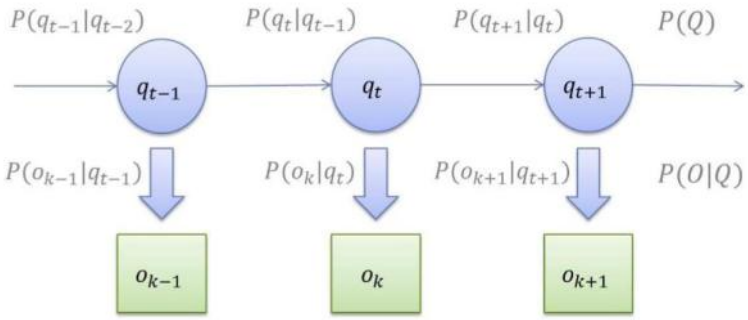

Рисунок 3.5 – Прихована марковська модель

Прихована марковська модель визначається наступними параметрами:

- множиною прихованих станів $Q = \{q0, \ldots, q\ \}$, де $q0$ – це початковий стан, а q – кінцевий стан;

- множиною спостережень $O = \{o1, \ldots, oM\}$;

- вихідним розподіленням станів $\pi = \{\pi i\}$, $1 \leq i \leq N$, який визначає ймовірність початку роботи у стані i;

- матрицею ймовірностей переходів між прихованими станами $AN \times N$ згідно $A(i,j) = a = P(q\ , q\ ), 1 \leq i,j \leq N$;

- матрицею ймовірностей спостережень $B \times M: B(i,j) = b = P(o\ |q\ ), 1 \leq i \leq N, 1 \leq j \leq M$.

Існують три основні задачі, які можно вирішити з використанням ПММ.

Обчислення ймовірності послідовності спостережень (лістинг 3.14). Треба визначити ймовірність появи послідовності спостережень $O = \{o1, \ldots, o\ \}$. Для цього використовується такі вирази:

$$a0(i) = a0i,\ \ 1 \le i \le N, \tag{3.13}$$

$$aj(r) = \sum_{i=1}^{N} ai(r-1)aij * bjt,\ \ 1 \le j \le N, 1 \le r \le R, \tag{3.14}$$

де $\alpha\ (r)$ – ймовірність того, що на кроці r модель буде у стані j.

Тоді шукана ймовірність визначається таким чином (лістинг 3.14):

$$P(O|A,B) = \sum_{i=1}^{N} a\ (R). \tag{3.15}$$

Лістинг 3.14 – Розрахунок ймовірності

```
for i in range(8):
    result = round()
    if result > x:
        x = result
if x == 5:
return True
```

Знаходження найбільш правдоподібної послідовності прихованих станів для послідовності. Потрібно знайти найбільш правдоподібну послідовність прихованих станів $Q = \{q1,\ldots,q\ \}$ для заданої послідовності спостережень $O = \{o1,\ldots,o\ \}$, при якій досягається $m\quad P\ (O|Q)$. Ця задача вирішується з використанням алгоритму, подібному алгоритму з попереднього пункту але з тієї різницею, що на кожному кроці запам’ятовується стан і, в якому αi(r) приймає найбільше значення. Далі обирається послідовність станів, для якої αi(R) приймає найбільше значення. Цей алгоритм має назву алгоритм Вітербі (лістинг 3.15).

Лістинг 3.15 – Реалізація алгоритму Вітербі

```
def viterbi(y, A, B, Pi=None):
    K = A.shape[0]
    Pi = Pi if Pi is not None else np.full(K, 1 / K)
    T = len(y)
```

```
    T1 = np.empty((K, T), 'd')
    T2 = np.empty((K, T), 'B')
    T1[:, 0] = Pi * B[:, y[0]]
    T2[:, 0] = 0
    for i in range(1, T):
        T1[:, i] = np.max(T1[:, i - 1] * A.T
        T2[:, i] = np.argmax(T1[:, i - 1] * A.T, 1)
    x = np.empty(T, 'B')    x[-1] = np.argmax(T1[:, T - 1])
    for i in reversed(range(1, T)):
        x[i - 1] = T2[x[i], i]
    return x, T1, T2
```

Навчання параметрів моделі по вказаній послідовності спостережень та множини прихованих станів. Треба обчислити для заданої моделі матриці A і B. Для вирішення цієї задачі застосовується алгоритм Баума-Велша (лістинг 3.16).

Лістинг 3.16 – Реалізація алгоритму Баума-Велша

```
def BaumWelch(T, S, H, A, B, pi, X):
    alpha = cal_alpha(T, S, H, A, B, pi)
    beta = cal_beta(T, S, H, A, B, pi)
    gamma = cal_gamma(T, S, H, A, B, pi, alpha, beta)
    xi = cal_xi(T, S, H, A, B, pi, alpha, beta)
    N = len(H)
    M = len(S)
    for j in range(N):
        for k in range(M):
            c = 0           d = 0
            for t in range(T):
                if X[t] == S[k]: c += gamma[t][j]
                d += gamma[t][j]
            B[j][k] = c / d
```

Для задачі ідентифікації диктора прихованими станами використовуються вектори ознак голосового сигналу з навчальної вибірки, а в якості спостережень – вектори ознак мовленевого сигналу з тестової вибірки. В якості моделі збереження тут виступають матриці A і B. ПММ досить прості в розумінні, мають досить високу точність розпізнавання, але, в основному застосовуються для завдань тексто-залежної ідентифікації диктора.

4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

В результаті аналізу описаних методів, для подальшої роботи було обрано метод мел-частотних кепстральних коефіцієнтів для аналізу голосового сигналу, а для розв’язання задачі ідентифікації диктора – використання штучних нейронних мереж.

Головною перевагою методу кепстральних коефіцієнтів є простота реалізації при високій якості розпізнавання, метод не поступається іншим популярним алгоритмам. Також даний метод передбачає менший обсяг аналізу даних, що значно скорочує час навчання нейронної мережі.

Перевагою ідентифікації диктора за допомогою нейронних мереж є здатність до навчання системи, що підвищує точність розпізнавання при більшій кількості циклів навчання.

4.1 Тестові дані

На основі відкритого датасету зразків голосу (Open Speech To Text (STT/ASR) Dataset) були сформовані навчальні і тестові зразки. Датасет містить зразки мовлення [19], які можна використати для ідентифікації диктора. У даному випадку він складається з аудіо зразків мовлення.

Переваги використання існуючого датасету полягають у наступних властивостях:

- після побудови, датасет може бути використаний багаторазово;
- датасет відображає мовні дані в реальному середовищі, що спрощує аналіз лексично-граматичної структури мови, шляхом наглядного відображення мовленнєвого процесу в динамічну структуру;
- широка сфера застосування датасетів, що використовуються у тестуванні пошукових систем та машинних морфологій, систем перекладу а також використовуються у лінгвістичних дослідження [28].

Наявність датасету розширює можливості аналізу мовного матеріалу, та надає можливість автоматизування аналітичного процесу. В процесі формування навчальної вибірки, кількість матеріалу визначає значимість, змістовність отриманих даних, а також рівень їх надійності.

Розмітка є ключовим процесом аналізу датасету, що відрізняє навчальний набір від наявних в інтернеті аудіо колекцій, енциклопедій, бібліотек. Текстова розмітка – це механізм відокремлення тексту для більш зручного аналізу.

Існують різні види розмітки:

- мета-текстова розмітка (автор, назва, дата створення, обсяг, тема тексту тощо), що характеризує текст у цілому;

- структурна розмітка – це інформація про структуру тексту, яка дозволяє відокремити одне слово від іншого, вибрати межі фрази, речення, тексту;

- лінгвістична розмітка полягає у присвоєнні одиницям тексту певної мовної інформації (заперечне речення або питання, тощо.

Повнота і різноманітність розмітки визначає її наукову та освітню цінність [30]. В Україні навчальна вибірка української мови була сформована співробітниками Інституту філології Київського національного університету імені Тараса Шевченка під керівництвом Н. П. Дарчук.

Датасет складається з текстів, що пропущені крізь лінгвістичний аналізатор, що присвоює кожній структурній одиниці тексту супутню інформацію: сенс, лексичне значення, синтаксична функція в реченні, граматична форма, тощо. Датасет надає два види інформації:

- конкорданс – контекст використання потрібної одиниці. Конкордансі дають можливість проаналізувати особливості використання слів у текстах різних стилів, у літературному аналізі;

- кількісні характеристики використання в текстах мовних одиниць.

Частотна інформація розкриває закономірності лексико-статистичної структури тексту, функціонування мовлення в мові, стилістичні особливості,

формальні особливості одиниць і граматичні категорії [29]. Для навчання нейронної мережі було обрано набір WAV-файлів. Кожен файл має структуровану розмітку.

Після прийому навчальних прикладів програма приймає кожен випадок один за одним і, згідно з алгоритмом формує вхідні дані в числовому вигляді. Після цього запускається цикл тренування, поки допустима помилка не буде в межах допустимої норми.

4.2 Результати та аналіз дослідження

У даному розділі проводиться аналіз результатів роботи реалізованого тестового програмного середовища. Завдання полягало в тому, щоб простежити залежність точності розпізнавання дикторів від різних параметрів: кількість епох навчання ШНМ, кількість окремих дикторів, тексто-незалежність розпізнавання.

В даному розділі буде представлений результат роботи тестового програмного середовища.

Експеримент розбитий на кілька етапів – тест системи на тексто-незалежність з трьома дикторами та дослідження роботи у тексто-залежному випадку в залежності від кількості дикторів:

- тест системи з трьома дикторами;
- тест системи з п’ятьма дикторами;
- тест системи на тексто-незалежність з трьома дикторами.

Умови експерименту (таблиця 4.1):

- кількість дикторів – 3;
- кількість епох навчання – 500;
- кількість експериментів – 10.

Як можна побачити з таблиці 4.1, частка помилкового розпізнавання становить 26,6%.

Таблиця 4.1 – Результати ідентифікації для системи з трьома дикторами (500 епох навчання)

Диктор	Результат експерименту									
	1	2	3	4	5	6	7	8	9	10
1	+	+	-	+	+	+	-	+	+	-
2	-	+	+	+	-	-	-	+	+	+
3	+	+	+	-	+	+	+	+	+	+

В процесі навчання кращий результат був досягнутий на останній епосі навчання. Час, витрачений на процес навчання, склав 4 секунди.

У наступному випадку кількість епох навчання складає 1000 ітерацій (таблиця 4.2). Умови експерименту:

- кількість дикторів – 3;
- кількість епох навчання – 1000;
- кількість експериментів – 10.

Таблиця 4.2 – Результати ідентифікації для системи з трьома дикторами

Диктор	Результат експерименту									
	1	2	3	4	5	6	7	8	9	10
1	+	+	+	+	-	+	+	+	+	+
2	-	+	-	+	-	+	+	+	-	+
3	+	+	+	+	+	+	+	+	+	+

Як можна побачити з результатів, зазначених у таблиці 4.2, вірогідність помилкового розпізнавання диктора становить 16,6%. Даний результат на 10% кращий за попередній. В процесі навчання кращий результат був досягнутий на останній епосі навчання. Час, витрачений на процес навчання склав 8 секунд.

Зараз розглядається випадок коли кількість епох навчання складає 5000. Умови експерименту:

- кількість дикторів – 3;
- кількість епох навчання – 5000;
- кількість експериментів – 10.

Таблиця 4.3 – Результати ідентифікації для системи з трьома дикторами (5000 епох навчання)

Диктор	Результат експерименту									
	1	2	3	4	5	6	7	8	9	10
1	+	+	+	+	+	+	+	+	+	+
2	+	+	+	+	+	+	+	+	-	+
3	+	+	+	+	+	-	+	+	+	+

Як можна побачити з таблиці 4.3, частка помилкового розпізнавання становить 6%. Даний результат на 10,6% краще, ніж при навчанні в 1000 епох. В процесі навчання кращий результат був досягнутий на 1598 епосі навчання. 23 секунди було витрачено на процес навчання.

Тепер будуть розглядатися експерименти за умови п’яти дикторів.

Умови експерименту:

- кількість дикторів – 5;
- кількість епох навчання – 500;
- кількість експериментів – 10.

Як видно з таблиці 4.4, частка невірного розпізнавання становить 38%. Кращий результат був досягнутий на останній епосі навчання. Час, витрачений на процес навчання, склав 5 секунд.

Проводиться наступний експеримент при збільшенні кількості епох навчання до 1000.

Умови експерименту:

- кількість дикторів – 5;
- кількість епох навчання – 1000;
- кількість експериментів – 10.

Таблиця 4.4 – Результати ідентифікації для системи з п’ятьма дикторами (500 епох)

Диктор	Результат експерименту									
	1	2	3	4	5	6	7	8	9	10
1	-	-	+	+	+	-	-	-	+	+
2	+	-	-	-	+	-	-	+	-	+
3	+	+	-	+	+	+	+	-	-	+
4	+	+	+	+	-	+	+	+	-	-
5	-	+	+	+	+	-	+	+	+	+

Як видно з таблиці 4.5, частка невірного розпізнавання становить 24%, що на 14% краще результату попереднього експерименту. В процесі навчання кращий результат був досягнутий на останній епосі навчання. На процес навчання витрачено 8 секунд.

Таблиця 4.5 – Результати ідентифікації для системи з п’ятьма дикторами (1000 епох)

Диктор	Результат експерименту									
	1	2	3	4	5	6	7	8	9	10
1	-	-	+	-	+	+	-	+	+	+
2	+	+	-	+	+	+	+	-	-	+
3	+	+	+	+	+	+	-	+	+	+
4	+	+	-	+	-	-	+	+	+	-
5	-	+	+	+	+	+	+	+	+	+

Для наступного експерименту кількість епох збільшується до 5000.

Умови експерименту:

- кількість дикторів – 5;
- кількість епох навчання – 5000;
- кількість експериментів – 10.

Таблиця 4.6 – Результати ідентифікації диктора для системи з п’ятьма дикторами (5000 епох)

Диктор	Результат експерименту									
	1	2	3	4	5	6	7	8	9	10
1	+	+	+	+	+	+	+	-	+	+
2	-	+	-	+	+	+	+	+	-	+
3	+	+	+	+	+	+	+	+	-	+
4	+	+	+	+	-	+	+	+	+	-
5	+	+	+	+	+	+	-	+	+	+

Як видно з таблиці 4.6, частка невірного розпізнавання становить 16%, що на 8% краще результату попереднього експерименту. В процесі навчання кращий результат був досягнутий на 3787 епосі навчання. Час, витрачений на процес навчання склав 27 секунд.

Для перевірки системи на тексто-незалежність було обрано варіант з трьома дикторами. Навчання нейронної мережі відбувається по одній фразі, а ідентифікація за іншою (таблиця 4.7). Кількість епох навчання дорівнюватиме 5000.

Таблиця 4.7 – Результати ідентифікації для системи з трьома дикторами (5000 епох навчання)

Диктор	Результат експерименту									
	1	2	3	4	5	6	7	8	9	10
1	+	+	+	+	-	+	+	+	+	+
2	+	-	+	+	+	+	-	+	+	+
3	+	+	+	+	+	+	+	+	-	+

Умови проведеного експерименту:

- кількість дикторів – 3;
- кількість епох навчання– 5000;

- кількість експериментів – 10;
- навчання та ідентифікація засновані на різних фразах.

Як можна побачити з таблиці 4.7, частка помилкового розпізнавання становить 13,3%. Даний результат на 7.3% гірше, ніж тексто-залежна ідентифікація при тих же умовах. В процесі навчання кращий результат був досягнутий на 4835 епосі навчання. Час, витрачений на процес навчання, склав 22 секунди. Узагальнені результати дослідження вказані в таблиці 4.8, а графік зміни середньоквадратичної помилки показаний на рисунку 4.4.

Таблиця 4.8 – Узагальнені результати досліджень

		Кількість епох навчання ШНМ		
		500	1000	5000
Частка хибного розпізнавання	Три диктора	26,6%	16,6%	6,0%
	П’ять дикторів	38,0%	24,0%	16,0%
	Десять дикторів	47,0%	34,0%	22,0%

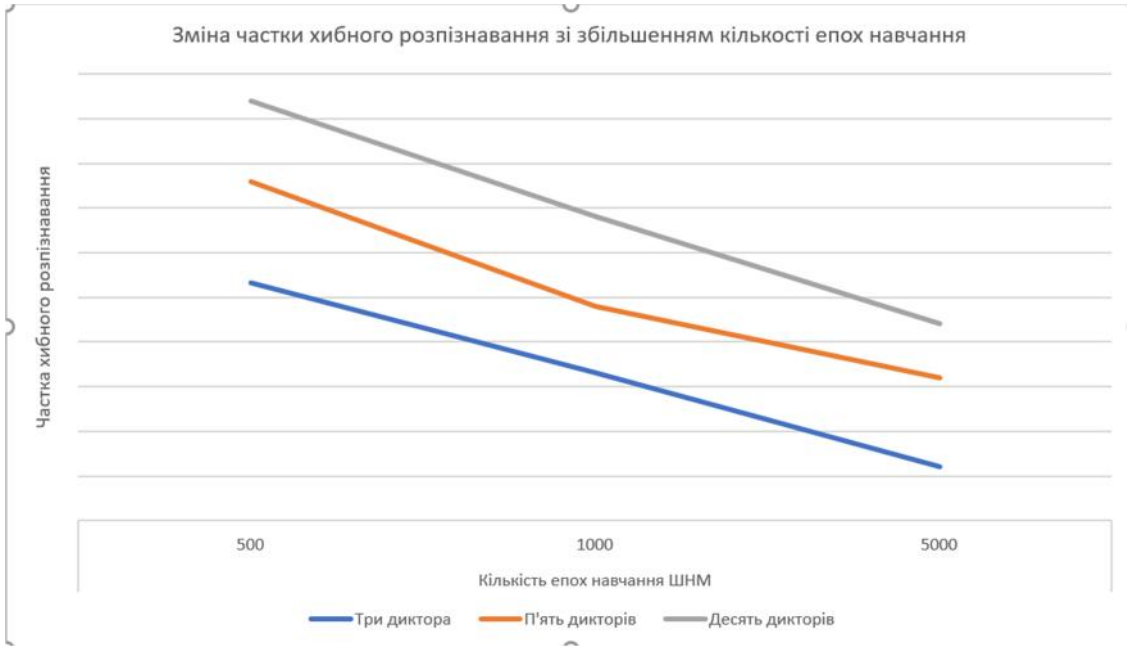

Рисунок 4.4 – Залежність точності розпізнавання від епох навчання для систем з різною кількістю дикторів

За результатами проведених експериментів був побудований графік, на якому наочно видно, як змінюється точність розпізнавання дикторів системою при зміні параметрів: кількості дикторів для розпізнавання і кількості епох навчання. На підставі цього можна зробити висновок, що при збільшенні кількості дикторів в системі точність ідентифікації зменшується, а при збільшенні кількості епох навчання зростає.

Окремо варто зауважити, що при тексто-незалежній ідентифікації дикторів точність розпізнавання зменшується. Пов’язано це з тим, що для такої ідентифікації необхідно більше епох навчання нейронної мережі, ніж для тексто-залежної, з таким же набором дикторів.

Даний експеримент доводить, що мел-частотні кепстральні коефіцієнти не прив’язуються до слів, які вимовляє диктор, а використовують при розпізнаванні «значущі» для людського вуха частоти, які характеризують людський голос.

Для збільшення точності розпізнавання дикторів нейронна мережа вимагає більш тонкого налаштування, більшої кількості епох навчання. Для забезпеченя поточної тривалості навчання буде потрібно збільшити вимоги щодо обчислювальних можливостей комп’ютера.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи проведений аналіз способів біометричної ідентифікації. Були досліджені існуючі методи голосової ідентифікації диктора. Для роботи був обраний метод мелчастотних кепстральних коефіцієнтів для отримання певних ознак голосу, а для ідентифікації були використані штучні нейронні мережі.

Головними перевагами методу мелкепстральних коефіцієнтів є проста реалізація, висока якість розпізнавання, яка не гірше інших подібних алгоритмів. Цей метод також передбачає невеликий етап аналізу даних, що збільшує швидкість навчання нейронної мережі. Перевагою ідентифікації диктора з використанням нейронних мереж є здібність до навчання системи, що може підвищувати точність розпізнавання при збільшенні кількості циклів навчання ШНМ. Але серед недоліків слід зазначити можливість перенавчити штучну нейронну мережу, що погіршить якість ідентифікації.

Була описана програмна реалізація функцій, що використані при розробці програми з розрахунку мел-частотних кепстральних коефіцієнтів і для побудови і використання штучної нейронної мережі.

При перевірці програми були проведені певні експерименти з різними умовами роботи програми та при різних вхідних даних. Проаналізувавши результати експериментів, які були проведені, можна дійти висновку, що при збільшенні кількості дикторів вірогідність вдалої ідентифікації зменшується, а при збільшенні кількості циклів навчання ШНМ точність ідентифікації зростає. При тексто-незалежній перевірці точність ідентифікації зменшується. Це явище спричинене тим, що для такого варіанту ідентифікації необхідно більше циклів навчання штучної нейронної мережі, аніж для тексто-залежної з ідентичними умовами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Заковряшин, А. С. Применение распределений мел-частотных кепстральных коэффициентов для голосовой идентификации личности [Текст] / А. С. Заковряшин, П. В. Малинин, А. А. Лепендин // Известия Алтайского государственного университета. – 2014. – №1 (81). – С. 156–160.

2. Меньшаков, П. А. Методика голосовой идентификации на основе нейронных сетей [Текст] / П. А. Меньшаков, И. А. Мурашко // Доклады Белорусского государственного университета информатики и радиоэлектроники. – 2014. – №4 (106). – С. 12–18.

3. Бідюк, П. Сучасні методи біометричної ідентифікації [Текст] / П. Бідюк, В. Бондарчук // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні : науково-технічний збірник. – 2009. – №1 (18). – С. 137–146.

4. Терейковський, І. Нейронні мережі в засобах захисту комп’ютерної інформації [Текст] / І. Терейковський. − К. : ПоліграфКонсалтинг. 2007. – 209 с.

5. Васильев, Р. А. Исследование особенностей идентификации дикторов по голосу [Текст] / Р. А. Васильев // Материалы Семнадцатой научной конференции по радиофизике, посвященной 100-летию со дня рождения В. С. Троицкого. – Новгород, 2013. – С. 45–48.

6. Матвеев, Ю. Н. Технологии биометрической идентификации личности по голосу и другим модальностям [Текст] / Ю. Н. Матвеев // Вестн. МГТУ им. Н. Э. Баумана. Сер. Приборостроение. Специальный выпуск. Биометрические технологии. – 2012. – № 3(3). – С. 46–61.

7. Davis, S. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences [Текст] / S. Davis, P. Mermelstein // EEE transactions on acoustics, speech, and signal processing. – 1980. – №28(4). – С. 357–366.

8. Mallat, S. Group invariant scattering [Текст] / S. Mallat // Communications on Pure and Applied Mathematics. – 2012. – №65(10). – С. 1331–1398.

9. Anden, J. Multiscale Scattering for Audio Classification [Текст] / J. Anden, S. Mallat // ISMIR. – 2011. – С. 657–662.

10. Первушин, Е. А. Обзор основных методов распознавания дикторов [Текст] / Е. А. Первушин // Математические структуры и моделирование. – 2011. – №3(24). – С. 41–54.

11. Ganchev, T. Comparative evaluation of various MFCC implementations on the speaker verification task [Текст] / T. Ganchev, N. Fakotakis, G. Kokkinakis // Proceedings of the SPECOM. – 2005. – №2005. – С. 191–194.

12. Малинин, П. В. Иерархический подход в задаче идентификации личности по голосу с помощью проекционных методов классификации многомерных данных [Текст] / П. В. Малинин, В. В. Поляков // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2010. – №1 – 1(21).

13. Сорокин, В. Н. Распознавание личности по голосу : аналитический обзор [Текст] / В. Н. Сорокин, В. В. Вьюгин, А. А. Тананыкин // Информационные процессы. – 2012. – №12(1). – С. 1–30.

14. Adeyemo, Z. K. Development of hybrid radio frequency identification and biometric security attendance system / Z. K. Adeyemo, O. J. Oyeyemi, I. A. Akanbi // International Journal of Applied. – 2014. – №4. – С. 190–197.

15. Petrovsky, A. Instantaneous harmonic analysis : techniques and applications to speech signal processing [Текст] / A. Petrovsky, E. Azarov // International Conference on Speech and Computer. – 2014. – №8773. – С. 24–33.

16. You, Y. Audio Coding : Theory and Applications [Текст] / Y. You. – MA : Springer Science Business Media, 2010. – 349 с.

17. Zheng, F. Comparison of different implementations of MFCC [Текст] / F. Zheng, G. Zhang, Z. Song // Journal of Computer science and Technology. – 2001. – №16(6). – С. 582–589.

18. Коваль, С. Л. Идентификация дикторов на основе сравнения статистик основного тона голоса [Текст] / С. Л. Коваль, П. В. Лабутин, Е. В. Малая, Е. А. Прощина // Матеріли XV Міжнародної научн. конф. – М., 2006. – С. 324–327.

19. Симончик, К. К. Алгоритм обнаружения речевой активности на основе статистик основного тона в задаче распознавания диктора [Текст] / К. К. Симончик, О. С. Галинина, А. И. Капустин // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика, телекоммуникации и управление. – 2010. – №4. – С. 18–23.

20. Дедів, І. Ю. Обгрунтування методу голосової ідентифікації особи [Текст] / І. Ю. Дедів, Л. Є. Дедів, С. Макар // Матеріали Міжнародної науково-технічної конференції «Теоретичні та прикладні аспекти радіотехніки, приладобудування і комп’ютерних технологій» присвячена 80-ти річчю з дня народження професора ЯІ Проця. – 2019. – С. 90–91.

21. Zue, V. Speech database development at MIT : TIMIT and beyond [Текст] / V. Zue, S. Seneff, J. Glass // Speech Communication. – 1990. – №9. – С. 351–356.

22. Ахмад, Х. М. Сравнительное исследование эффективности различных методов кепстрального описания речевых сигналов в задачах распознавания [Текст] / Х. М. Ахмад // Вестник ТГТУ. – 2007. – Том 13, №4. – С. 887–891.

23. Хайкин, С. Нейронные сети : полный курс [Текст] / С. Хайкин. – М. : Издательский дом «Вильямс», 2006. – 1104 с.

24. Nakagawa, S. Text-Independent/Text-Prompted Speaker Recognition by Combining Speaker-Specific GMM with Speaker Adapted Syllable-Based HMM [Текст] / S. Nakagawa W. Zhang, M. Takahashi // IEICE Transactions on Information and Systems. – 2006. – №89 (3). – С. 1058–1065.

25. Osman, R. Development of a speaker recognition system using wavelets and artificial neural networks [Текст] / R. Osman, C.P. Lim, S.C. Woo //

International Sym- posium on Intelligent Multimedia, Video and Speech Processing. – Hong Kong, 2001. – С. 413–416.

26. Scheffer, N. Towards noise-robust speaker recognition using probabilistic linear discriminant analysis [Текст] / N. Scheffer, M. Graciarena, L. Ferrer, L. Burget, Y. Lei // International Conference on Acoustics, Speech, and Signal Processing (ICASSP). – Kyoto, 2012. – С. 4253–4256.

27. Козлов, А. В. Алгоритм детектирования музыкальных фрагментов в задачах речевой обработки [Текст] / А. В. Козлов, А. И. Лоханова, К. К. Симончик // Научно-технические ведомости СПбГПУ. – 2010. – №4 (103). – С. 7–11.

28. Галяшина, Е. И. Идентификация дикторов по цифровым фонограммам [Текст] / Е. И. Галяшина // Речевые технологии. – 2010. – №3. – С. 23–28.

29. Іващенко, Г. С. Методи голосової ідентифікації [Текст] / Г. С. Іващенко, Д. І. Коваль // Проблеми інформатизації. – 2021. – Том 3. – С. 94.

30. Pastushenko, M. Analysis of voice signal phase data informativity of authentication system [Текст] / M. Pastushenko, Ya. Krasnozheniuk, O. Lemeshko // Proceedings of The Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020). – 2020. – С. 1040–1053.

31. Сорокин, В. Н. Фундаментальные исследования речи и прикладные за- дачи речевых технологий [Текст] / В. Н. Сорокин // Речевые технологии. – 2008. – №1. – С. 18–48.