

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Міністерство освіти і науки України

Кваліфікаційна наукова  
праця на правах рукопису

**Борисов Гліб Олександрович**

УДК 621.391.83

**ДИСЕРТАЦІЯ**  
**АДАПТИВНІ СИСТЕМИ ОБРОБЛЕННЯ АКУСТИЧНОЇ ІНФОРМАЦІЇ**  
**ДЛЯ СТВОРЕННЯ ПЕРСОНАЛІЗОВАНОГО МЕДІАКОНТЕНТУ**

17 – Електроніка та телекомунікації

171 – Електроніка

Подається на здобуття наукового ступеня доктора філософії.

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_ / Борисов Г. О.

Науковий керівник Трапезон Кирил Олександрович, кандидат технічних наук,  
доцент

## АНОТАЦІЯ

*Борисов Г.О.* Адаптивні системи оброблення акустичної інформації для створення персоналізованого медіаконтенту. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії у галузі знань 17 – Електроніка та телекомунікації за спеціальністю 171 «Електроніка». – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», МОН України, Київ, 2025.

Дисертаційна робота присвячена дослідженню адаптивних систем оброблення акустичної інформації для створення персоналізованого медіаконтенту.

Зміст дисертаційного дослідження викладено в трьох розділах, де представлено та обґрунтовано основні результати роботи.

Актуальність дисертаційної роботи обґрунтовано у вступі, де сформульовано мету та задачі дослідження, описано методи дослідження, надано інформацію про наукову новизну та практичне значення одержаних результатів.

Об'єктом дослідження є різноманітний аудіо контент з записом живого або синтетично створеного голосового повідомлення на українській та англійській мовах.

Застосування технологій оброблення акустичної інформації може бути спрямовано на забезпечення алгоритмів створення якісного персоналізованого медіаконтенту, наприклад для систем клонування голосу. У контексті цифрової трансформації суспільства важливість таких технологій останнім часом невпинно зростає, адже вони знаходять своє застосування у багатьох галузях, включаючи медицину, освіту, інформаційні системи, розваги та засоби комунікації.

Одним із ключових аспектів роботи є застосування нейромережових алгоритмів для оброблення акустичних сигналів. Використання нейронних мереж, як альтернативний спосіб, дозволяє отримати точну ідентифікацію

голосу, реалізувати синтез природного мовлення та ефективно зменшення шуму і реверберації сигналів. Особливо актуальним це стає для систем, які працюють у складних акустичних умовах.

Значний інтерес викликає завдання створення персоналізованого контенту, яке базується на здатності нейронних мереж адаптуватися до індивідуальних характеристик мовця. Це включає можливість збереження унікальних інтонацій, тембру та інших специфічних особливостей голосу. Крім того, задача підвищення розбірливості мовлення є важливою для поліпшення комунікації між користувачами у різних акустичних середовищах, серед яких це лекційні зали, офіси або відкриті простори.

Зокрема, використання нейронних мереж дозволяє автоматизувати та покращувати процес обробки звукових сигналів, що є основою медіасистем. Такий підхід забезпечує можливість створювати персоналізований контент, який враховує, у тому числі, специфічні вподобання користувачів.

У першому розділі розглянуто сучасний стан досліджень у галузі обробки акустичної інформації та створення персоналізованого медіаконтенту. Представлено загальні відомості про основні типи акустичних сигналів, які включають широке різноманіття звукових хвиль — від природних шумів до мовлення, музики та техногенних сигналів. Розкрито їх ключові характеристики, такі як амплітуда, частота, тривалість і спектральний склад, які формують базу для їх подальшого аналізу та обробки. Описано ключові технології, такі як згорткові нейронні мережі, рекурентні архітектури та їх застосування у задачах розпізнавання мовлення, синтезу голосу та зменшення шумів. Наведено приклади використання часово-частотного представлення сигналів (спектрограм, мел-спектрограм) для вилучення інформативних ознак з аудіоданих. Також наведено актуальні підходи до адаптації моделей до оброблення сигналів у складних акустичних умовах. Розглянуто методи оцінювання ефективності нейромережових моделей, а також перспективи їхнього використання для персоналізації голосу в різних прикладних задачах.

У другому розділі, присвяченому огляду загальних засад адаптивних систем оброблення акустичної інформації, розглянуто основні принципи

побудови таких систем. Наведено загальні концепції адаптивності, що забезпечують ефективну роботу систем у змінних акустичних умовах. Описано ключові компоненти адаптивних систем, включаючи модулі вилучення ознак, класифікації та синтезу мовлення. Проведено аналіз сучасних архітектур нейронних мереж, таких як згорткові та рекурентні моделі, які є основою для створення адаптивних рішень. Представлено приклади використання систем із застосуванням часово-частотного представлення сигналів, що дозволяє досягти високої точності вилучення інформативних ознак. Також наведено актуальні підходи до інтеграції методів адаптації, таких як нормалізація даних, компенсація шумів і реверберації. Розглянуто перспективи використання адаптивних систем для вирішення прикладних задач, таких як синтез персоналізованого голосу, автоматичне розпізнавання мовлення та аудіообробка в реальному часі. Особливий акцент зроблено на значенні цих систем для інноваційних сфер, таких як голосові асистенти, медичні пристрої, системи безпеки та мультимедійні додатки. Це підкреслює їхній внесок у підвищення комфорту, інтерактивності та персоналізації сучасних технологій.

У третьому розділі детально описано проведення серії експериментальних досліджень, спрямованих на перевірку ефективності розроблених методів оброблення акустичної інформації. Наведено опис експериментальної бази, включаючи використане програмне забезпечення, набори даних та параметри середовищ. Представлено результати перевірки розроблених алгоритмів для задач вилучення ознак, синтезу мовлення та адаптації аудіосигналів у різних акустичних умовах. Зокрема, розглянуто методи зменшення впливу шумів та реверберації, а також забезпечення персоналізації голосу. Описано проведення експериментів на різних наборах аудіоданих, що дозволило оцінити стабільність і точність запропонованих підходів. Висвітлено практичну цінність отриманих результатів у реальних сценаріях, таких як створення персоналізованого медіаконтенту, ідентифікація за голосом та обробка аудіо у складних умовах.

В дисертаційній роботі отримано наступні наукові результати:

1. Вперше розроблено систему ідентифікації за голосом, яка є стійкою до штучно підробленого голосу і показує високу точність схожості відразу за 4 критеріями.

2. Вперше побудовано акустичну модель розпізнавання мовних сигналів з підтримкою нейронної мережі, яка дозволяє в якості вхідної інформації використовувати українські словосполучення. Для її реалізації розроблено змінену рекурентну нейронну мережу, яка вирізняється тим, що за рахунок вбудованої пам'яті в структурі етап навчання та тестування нейронної мережі моделі можна проводити одночасно.

3. Удосконалено програмний алгоритм дереверберації записаних аудіо сигналів з адитивним додаванням шуму, де використано згорткову нейронну мережу за архітектурою U-Net і яка адаптована до запису не тільки тестових сигналів типу 'сплеск' або "постріл", але й словосполучень українською мовою.

4. Набуло подальшого розвитку створення систем клонування голосу за рахунок введення послідовно трьох попередньо навчених нейронних мереж. Такий підхід дозволив зберегти акцент, інтонаційні та інші фонетичні особливості у синтезованих фразах як англійською, так і українською мов.

Практичне значення одержаних в дисертаційній роботі результатів полягає в тому, що отримані результати можуть бути використані для широкого спектру завдань у галузі обробки аудіосигналів. Практичне значення отриманих результатів полягає у розробці та впровадженні інноваційних методів обробки акустичної інформації, що базуються на принципах функціонування сучасних нейронних мереж. Отримані результати можуть бути використані для створення систем автоматичного розпізнавання мовлення, синтезу персоналізованого голосу, адаптації аудіосигналів до різних акустичних умов та зменшення впливу шумів і реверберації. Запропоновані алгоритми та підходи є універсальними та можуть бути інтегровані у широкий спектр застосувань, таких як голосові помічники, системи безпеки, слухові апарати, медичне обладнання, інтерфейси "розумного будинку" та мультимедійні платформи. Практична значущість роботи підтверджується можливістю

використання її результатів для підвищення точності, стійкості та адаптивності сучасних технологій персоналізованого медіаконтенту.

*Ключові слова: розбірливість мовлення, оцінка, якість мовлення, тестовий сигнал, реверберація, шуми, моделювання, процес, звук, Інтернет речей, IoT, комп'ютерна система, рівень сигналу, розповсюдження сигналу.*

## SUMMARY

Borisov G.O. Adaptive acoustic information processing systems for creating personalized media content.

Dissertation for the degree of Doctor of Philosophy in Knowledge Area 17 - Electronics and Telecommunications, specialty 171 "Electronics." - National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ministry of Education and Science of Ukraine, Kyiv, 2025.

The dissertation is devoted to the study of adaptive acoustic information processing systems for creating personalized media content.

The content of the dissertation research is presented in three chapters, where the main results of the work are presented and substantiated.

The relevance of the thesis is substantiated in the introduction, which formulates the purpose and objectives of the study, describes the research methods, and provides information on the scientific novelty and practical significance of the results.

The object of the study is a variety of audio content with live or synthetically generated voice messages in Ukrainian and English.

The development of modern acoustic information processing technologies is inextricably linked to ensuring the creation of high-quality personalized media content, especially for voice cloning systems. In the context of the digital transformation of society, the importance of such technologies is growing, as they are used in many fields, including medicine, education, information systems, entertainment, and communication.

One of the key aspects of the work is the study of neural network algorithms for processing acoustic signals. The use of neural networks is likely to enable accurate voice identification, natural speech synthesis, and effective noise and reverberation reduction. This is especially relevant for systems operating in difficult acoustic conditions.

The task of creating personalized content based on the ability of neural networks to adapt to the individual characteristics of the speaker is of great interest. This includes the ability to preserve unique intonations, timbre, and other specific voice

features. In addition, the task of enhancing speech intelligibility is important for improving communication between users in different acoustic environments, such as lecture halls, offices, or open spaces.

In particular, the use of neural networks allows automating and improving the processing of audio signals, which is the basis of media systems. This approach makes it possible to create personalized content that takes into account the specific preferences of users.

The first chapter discusses the current state of research in the field of acoustic information processing and personalized media content creation. General information about the main types of acoustic signals is presented, which include a wide variety of sound waves - from natural noise to speech, music, and man-made signals. Their key characteristics, such as amplitude, frequency, duration, and spectral composition, which form the basis for their further analysis and processing, are revealed. Key technologies such as convolutional neural networks, recurrent architectures and their application in speech recognition, voice synthesis and noise reduction are described. Examples of the use of time-frequency representation of signals (spectrograms, mel-spectrograms) to extract informative features from audio data are given. The current approaches to the adaptation of models to signal processing in complex acoustic conditions are also presented. The methods for evaluating the effectiveness of neural network models, as well as the prospects for their use for voice personalization in various applied tasks, are considered.

In the second section, devoted to the review of the general principles of adaptive acoustic information processing systems, the basic principles of building such systems are considered. General concepts of adaptability are presented to ensure efficient operation of systems under variable acoustic conditions. The key components of adaptive systems are described, including modules for feature extraction, classification, and speech synthesis. An analysis of modern neural network architectures, such as convolutional and recurrent models, which are the basis for creating adaptive solutions, is presented. Examples of the use of systems with the use of time-frequency representation of signals are presented, which allows to achieve high accuracy of informative features extraction. The article also presents



current approaches to the integration of adaptation methods, such as data normalization, noise and reverberation compensation. The prospects of using adaptive systems to solve applied problems, such as personalized voice synthesis, automatic speech recognition, and real-time audio processing, are considered. Particular emphasis is placed on the importance of these systems for innovative areas such as voice assistants, medical devices, security systems, and multimedia applications. This emphasizes their contribution to increasing the comfort, interactivity and personalization of modern technology.

Chapter 3 describes in detail the series of experimental studies aimed at verifying the effectiveness of the developed methods for processing acoustic information. We describe the experimental setup, including the software used, data sets, and environmental parameters. The results of testing the developed algorithms for the tasks of feature extraction, speech synthesis, and adaptation of audio signals in different acoustic conditions are presented. In particular, the methods for reducing the impact of noise and reverberation, as well as for ensuring voice personalization are considered. Experiments on different sets of audio data are described, which allowed us to evaluate the stability and accuracy of the proposed approaches. The practical value of the obtained results in real-life scenarios, such as the creation of personalized media content, voice identification, and audio processing in complex environments, is highlighted.

The following scientific results were obtained in the dissertation:

1. For the first time, a voice identification system has been developed that is resistant to artificially faked voices and shows high accuracy of similarity by 4 criteria at once.
2. For the first time, an acoustic model of speech signal recognition with the support of a neural network was built, which allows using Ukrainian word combinations as input. For its implementation, a modified recurrent neural network was developed, which is distinguished by the fact that due to the built-in memory in the structure, the training and testing of the model's neural network can be carried out simultaneously.

3. A software algorithm for deregulation of recorded audio signals with additive noise addition has been improved, using a convolutional neural network based on the U-Net architecture and adapted to record not only test signals such as 'burst' or 'shot', but also word combinations in Ukrainian.

4. The development of voice cloning systems was further developed by introducing three pre-trained neural networks in sequence. This approach made it possible to preserve the accent, intonation, and other phonetic features in the synthesized phrases of both English and Ukrainian.

The dissertation is devoted to the investigation of adaptive systems for acoustic information processing aimed at creating personalized media content.

The practical significance of the results obtained in this thesis is that the findings can be used for a wide range of tasks in the field of audio signal processing. The practical significance of the results obtained is the development and implementation of innovative methods of acoustic information processing based on the principles of functioning of modern neural networks. The obtained results can be used to create systems for automatic speech recognition, personalized voice synthesis, adaptation of audio signals to different acoustic conditions, and reduction of noise and reverberation. The proposed algorithms and approaches are versatile and can be integrated into a wide range of applications, such as voice assistants, security systems, hearing aids, medical equipment, smart home interfaces, and multimedia platforms. The practical significance of the work is confirmed by the possibility of using its results to improve the accuracy, stability, and adaptability of modern technologies for personalized media content.

*Key words: speech intelligibility, evaluation, speech quality, test signal, reverberation, noise, modeling, process, sound, Internet of Things, IoT, computer system, signal strength, signal propagation.*

### *Список публікацій здобувача*

1. Борисов Г., Трапезон К., Дослідження особливостей створення електронних систем розпізнавання мови на основі нейронних мереж, Вчені записки Таврійського національного університету імені В.І.Вернадського. Серія: Технічні науки. Том 33(72), №5, 2022, <https://doi.org/10.32782/2663-5941/2022.5/57>
2. Борисов Г., Трапезон К., Особливості дереверберації мовних сигналів за допомогою нейронних мереж, Вісник Кременчуцького національного університету імені Михайла Остроградського. 2023. Випуск 3 (140), <https://doi.org/10.32782/1995-0519.2023.3.18>
3. Борисов Г., Трапезон К., Дослідження особливостей створення текстонезалежних голосових систем доступу з захистом від спуфінг-атак, Вісник Кременчуцького національного університету імені Михайла Остроградського. 2024. Випуск 1 (143), <https://doi.org/10.32782/1995-0519.2024.1.34>
4. Борисов Г., Трапезон К., Підходи та принципи створення системи клонування голосу, Вісник Кременчуцького національного університету імені Михайла Остроградського. 2024. Випуск 4 (147), <https://doi.org/10.32782/1995-0519.2024.4.8>
5. Борисов Г., Трапезон К., Особливості створення електронних систем розпізнавання мови на основі нейронних мереж, XI Міжнародна науково-технічна конференція “Радіотехнічні поля, сигнали, апарати та системи”, КПІ ім. Ігоря Сікорського, 22-24 листопада 2022 р. Київ, 2022. с. 64-66

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	14
ВСТУП.....	15
1. ЛІТЕРАТУРНИЙ ОГЛЯД.....	21
1.1 Інноваційні підходи до оброблення аудіоконтенту.....	21
1.2 Аналіз сигналів мовлення .....	22
1.3 Оцінка розбірливості мовлення.....	25
1.4 Деревверберації мовних сигналів.....	27
1.5 Адаптивні системи оброблення акустичної інформації.....	30
1.5.1 Розпізнавання та синтез мови .....	33
1.6 Типи нейронних мереж для оброблення акустичної інформації .....	36
1.6.1 Рекурентні нейронні мережі .....	37
1.6.2 Згорткові нейронні мережі.....	39
1.6.3 Алгоритм зворотного поширення помилки .....	39
2 ОГЛЯД ЗАГАЛЬНИХ ЗАСАД СТВОРЕННЯ АДАПТИВНИХ СИСТЕМ ОБРОБЛЕННЯ АКУСТИЧНОЇ ІНФОРМАЦІЇ .....	43
2.1 Теоретичні підходи побудови систем ідентифікації за голосом.....	43
2.1.1 MEL-кепстральні коефіцієнти .....	44
2.1.2 Регуляризація.....	47
2.1.3 Структура систем ідентифікації за голосом.....	48
2.2 Теоретичні підходи побудови систем підвищення розбірливості мови ....	49
2.2.1 Реверберація .....	50
2.2.2 Структура системи підвищення розбірливості мови .....	51
2.3 Теоретичні основи побудови систем клонування голосу .....	53
3 ПРАКТИЧНА ЧАСТИНА ДОСЛІДЖЕННЯ.....	56

3.1 Система ідентифікації за голосом .....	59
3.1.1 Навчання системи ідентифікації за голосом .....	60
3.1.2 Перевірка розробленої системи.....	63
3.2 Система розпізнавання мови.....	65
3.2.1 Навчання системи розпізнавання мови .....	67
3.2.1 Практична перевірка системи розпізнавання мови .....	68
3.3 Система покращення розбірливості мови .....	73
3.3.1 Етап навчання нейронної мережі для системи покращення розбірливості мови.....	75
3.3.2 Вихідні дані до експерименту .....	78
3.3.3 Експериментальна перевірка 1 (209 навчальна аудиторія) .....	78
3.3.4 Експериментальна перевірка 2 (438 навчальна аудиторія) .....	82
3.3.5 Експериментальна перевірка 3 (житлова кімната).....	87
3.3.6 Експеримент з додаванням фонового шуму .....	91
3.4 Система клонування голосу .....	99
3.4.1 Етап навчання елементів системи клонування голосу.....	99
3.4.2 Практичний експеримент .....	102
3.5 Оцінка якості системи ідентифікації голосу .....	112
ВИСНОВКИ.....	115
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....	118
ДОДАТОК А Програмний код системи ідентифікації за голосом .....	129
ДОДАТОК Б Програмний код системи покращення розбірливості мови .....	130
ДОДАТОК В Програмний код системи клонування голосу .....	132

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

ПММ — Приховані марківські моделі;

ADM — Adaptive Dual Microphone;

Adagrad — Adaptive Gradient Algorithm (Адаптивний алгоритм градієнта);

Adam — Adaptive Moment Estimation (Адаптивна оцінка моментів);

AUC – Area Under the Curve;

CNN — Convolutional Neural Network (Згортова нейронна мережа);

FPR – False Positive ratio;

ISTFT — Inverse Short-Time Fourier Transform (Зворотне короткочасне перетворення Фур'є);

LSTM — Long Short-Term Memory;

MFCC — Mel-Frequency Cepstral Coefficients (Мел-кепстральні коефіцієнти);

MSE — Mean Square Error;

RNN — Recurrent Neural Network (Рекурентна нейронна мережа);

ROC – Receiver Operating Characteristic curves;

SGD — Stochastic Gradient Descent (Стохастичний градієнтний спуск);

STFT — Short-Time Fourier Transform (Короткочасне перетворення Фур'є);

TPR – True Positive Ratio.

## ВСТУП

**Актуальність роботи.** Розвиток сучасних технологій обробки акустичної інформації нерозривно пов'язаний з забезпеченням створення якісного персоналізованого медіаконтенту, особливо для систем клонування голосу. У контексті цифрової трансформації суспільства важливість таких технологій зростає, адже вони знаходять застосування у багатьох галузях, включаючи медицину, освіту, інформаційні системи, розваги та засоби комунікації.

Одним із ключових аспектів роботи є дослідження нейромережових алгоритмів для обробки акустичних сигналів. Використання нейронних мереж напевно дозволяє отримати точну ідентифікацію голосу, синтез природного мовлення та ефективно зменшення шуму й реверберації. Особливо актуальним це стає для систем, які працюють у складних акустичних умовах.

Значний інтерес викликає завдання створення персоналізованого контенту, яке базується на здатності нейронних мереж адаптуватися до індивідуальних характеристик мовця. Це включає можливість збереження унікальних інтонацій, тембру та інших специфічних особливостей голосу. Крім того, задача підвищення розбірливості мовлення є важливою для поліпшення комунікації між користувачами у різних акустичних середовищах, таких як лекційні зали, офіси або відкриті простори.

Зокрема, використання нейронних мереж дозволяє автоматизувати та покращувати процес обробки звукових сигналів, що є основою медіасистем. Такий підхід забезпечує можливість створювати персоналізований контент, який враховує специфічні вподобання користувачів.

Також одним з напрямків обробки акустичної інформації є безпосереднє покращення звукового потоку знешумлення або дереверберації яке широко використовується в слухових апаратах. Питаннями оброблення акустичної інформації займаються такі вітчизняні та іноземні вчені як, Продеус А.М, Наконечний Р.А., Бабак В.П., Міхаель Ферлендер, Девід Хейвелок, Джон Картер, Майкл Джонсон та інші.

Актуальність дослідження обґрунтовується також тим, що такі технології знаходять широке застосування в системах безпеки, інтерфейсах "розумного будинку", слухових апаратах, а також у медіа-індустрії для автоматичного дубляжу, створення віртуальних асистентів та інтерактивних систем. Ці розробки забезпечують новий рівень інтерактивності та персоналізації, який раніше був недосяжним.

**Мета і завдання дослідження.** *Метою дисертаційної роботи є дослідження нейромережових алгоритмів для створення персоналізованого медіаконтенту на основі підходів оброблення акустичної інформації. Дослідження спрямоване на отримання покращення якості мовних сигналів та врахування індивідуальності медіа контенту, шляхом застосування методів навчання нейронних мереж, які в свою чергу, здатні адаптуватись в процесі навчання до голосових особливостей мовця, і які також дозволяють ефективно здійснювати дереверберацію, синтез і розпізнавання мовних сигналів.*

*Об'єктом дослідження є різноманітний аудіо контент з записом живого або синтетично створеного голосового повідомлення на українській та англійській мовах.*

*Предметом дослідження є методи та алгоритми використання нейронних мереж для обробки мовних сигналів, включаючи розпізнавання, зменшення шуму, дереверберацію і клонування голосу, які спрямовані на покращення якості та забезпечення персоналізації медіаконтенту.*

Для досягнення поставленої мети були сформульовані наступні завдання:

1. Розробити систему ідентифікації за голосом, яка характеризується задовільною точністю спрацювання при п'яти різних варіантах вимовлення контрольної фрази-перевірки.

2. Розробити модель системи розпізнавання мовних сигналів з досягненням мінімального значення функції втрат.

3. Розробити алгоритм застосування нейронної мережі для розв'язання задач дереверберації та зменшення шуму.



4. Розробити схему клонування голосу з використанням нейронних мереж для збереження інтонацій, тембру та інших специфічних фонетичних характеристик мовлення, що забезпечить у свою чергу високий рівень персоналізації контенту.

Ці завдання спрямовані на всебічне дослідження та впровадження адаптивних систем для обробки аудіосигналів, що дозволить покращити якість та персоналізацію медіаконтенту в умовах існування реальних звукових середовищ та оточення.

**Методи дослідження.** Для досягнення поставленої мети використано методи аналітичного та комп'ютерного моделювання з використанням інструментарію Matlab, а також методи експериментального дослідження та метод експертних оцінок. При аналітичному оцінюванні застосовувались математичні моделі часово-частотного представлення сигналів, такі як спектрограми та мел-спектрограми, що дозволило формалізувати процеси вилучення ознак та адаптації аудіосигналів. Комп'ютерне моделювання здійснювалося із використанням відкритого програмного забезпечення на основі сучасних нейронних мереж, зокрема згорткових та рекурентних архітектур, що дали змогу ефективно реалізувати різні аспекти обробки акустичної інформації. Експериментальні дослідження включали кілька етапів, зокрема тестування розроблених систем на різних наборах даних, оцінювання точності та адаптації сигналів у складних акустичних умовах.

**Наукова новизна** одержаних результатів полягає у тому, що:

1. Вперше розроблено систему ідентифікації за голосом, яка є стійкою до штучно підробленого голосу і показує високу точність схожості з еталонним записом відразу за 4 критеріями.

2. Вперше побудовано акустичну модель розпізнавання мовних сигналів з підтримкою нейронної мережі, яка дозволяє в якості вхідної інформації використовувати українські словосполучення. Для її реалізації розроблено змінену рекурентну нейронну мережу, яка вирізняється тим, що за

рахунок вбудованої пам'яті в структурі етап навчання та тестування нейронної мережі моделі можна проводити одночасно.

3. Удосконалено програмний алгоритм дереверберації записаних аудіо сигналів з адитивним додаванням шуму, де використано згорткову нейронну мережу за архітектурою U-Net і яка адаптована до запису не тільки тестових сигналів типу 'сплеск' або "постріл", але й словосполучень українською мовою.

4. Набуло подальшого розвитку створення систем клонування голосу за рахунок введення послідовно трьох попередньо навчених нейронних мереж. Такий підхід дозволив зберегти акцент, інтонаційні та інші фонетичні особливості у синтезованих фразах як англійської, так і української мов.

**Особистий внесок здобувача.** Усі результати, наведені у дисертаційній роботі і винесені на захист, отримано за активної участі здобувача та опубліковано у спеціалізованих фахових виданнях.

У роботі «Дослідження особливостей створення електронних систем розпізнавання мови на основі нейронних мереж», Борисов Г., Трапезон К., Вчені записки Таврійського національного університету імені В.І.Вернадського. Серія: Технічні науки. Том 33(72), №5, 2022, опублікованій у співавторстві, здобувач був співавтором програмного коду.

У роботі «Особливості дереверберації мовних сигналів за допомогою нейронних мереж», Борисов Г., Трапезон К., Вісник Кременчуцького національного університету імені Михайла Остроградського. 2023. Випуск 3 (140), опублікованій у співавторстві, здобувач особисто проводив експериментальні дослідження, частина яких використовується у дисертаційній роботі.

У роботі «Дослідження особливостей створення текстонезалежних голосових систем доступу з захистом від спуффінг-атак», Борисов Г., Трапезон К., Вісник Кременчуцького національного університету імені Михайла Остроградського. 2024. Випуск 1 (144) опублікованій у співавторстві, здобувач особисто проводив експериментальні дослідження, частина яких використовується у дисертаційній роботі.

У роботі «Підходи та принципи створення системи клонування голосу», Борисов Г., Трапезон К., Вісник Кременчуцького національного університету імені Михайла Остроградського. 2024. Випуск 4 (147), опублікованій у співавторстві, здобувач особисто проводив експериментальні дослідження, частина яких використовується у дисертаційній роботі.

**Практичне значення отриманих результатів.** Практичне значення отриманих результатів полягає у розробці та впровадженні ефективних підходів обробки акустичної інформації, які базуються на принципах функціонування сучасних нейронних мереж. Отримані результати можуть бути використані для створення систем автоматичного розпізнавання мовлення, синтезу персоналізованого голосу, адаптації аудіосигналів до різних акустичних умов та зменшення впливу шумів і реверберації. Запропоновані алгоритми та підходи є універсальними та можуть бути інтегровані у широкий спектр застосувань, таких як голосові помічники, системи безпеки, слухові апарати, медичне обладнання, інтерфейси "розумного будинку" та мультимедійні платформи. Практична значущість роботи підтверджується можливістю використання її результатів для підвищення точності, стійкості та адаптивності сучасних технологій персоналізованого медіаконтенту.

#### **Зв'язок роботи з науковими програмами, планами, темами.**

Викладені у дисертації нові теоретичні та практичні результати досліджень можуть використовуватися при розробці систем оброблення акустичної інформації та створення аудіо контенту з застосування технологій Інтернету речей, а також у освітньому процесі при підготовці нових навчальних дисциплін за спеціальністю 171 Електроніка.

**Апробація результатів дисертації.** Основні положення та результати дисертаційного дослідження доповідались на 1 міжнародній науково-практичній конференції:

1. Особливості створення електронних систем розпізнавання мови на основі нейронних мереж. Борисов Г. О., Трапезон К. О., XI Міжнародна

науково-технічна конференція «Радіотехнічні поля, сигнали, апарати та системи», КПІ ім. Ігоря Сікорського, 22-24 листопада 2022 р.

**Публікації.** За результатами досліджень опубліковано 5 наукових публікацій (з них 4 статті у наукових фахових виданнях України за спеціальністю 171 Електроніка.

**Структура та обсяг дисертаційної роботи.** Робота складається зі вступу, трьох розділів, списку використаних джерел із 92 найменування та 3 додатки. Робота містить 29 рисунків та 21 таблиця. Загальний обсяг дисертаційної роботи складає 137 сторінка.

## 1. ЛІТЕРАТУРНИЙ ОГЛЯД

### 1.1 Інноваційні підходи до оброблення аудіоконтенту

Оброблення аудіоінформації, наприклад, з метою фільтрації її від небажаних шумів або для виявлення аномалій при її прослуховуванні можна проводити на основі різних підходів та алгоритмів. Так, деякі розробники пропонують використовувати приховані марківські моделі, як дуже зручний засіб для аналізу фонем у словесних конструкціях. Але в даному випадку, необхідно мати великий набір навчальних даних, вміти проводити фонетичний аналіз та, власне, знати як саме створюються статистичні моделі для оцінки вірогідностей переходів між різними фонемами та словами. Інші пропонують методи фільтрації сигналів, на основі яких можливе очищення сигналів від шуму. Абсолютно іншим перспективним підходом для проведення оброблення аудіоданих може стати використання нейронних мереж, завдяки чому для користувача стає доступним простий і ефективний інструментарій з налаштування та адаптації цих мереж у випадку, коли результат оброблення сигналу потребує забезпечення певної якості відтворення, як от у випадку роботи з мовними сигналами. Такі вимоги щодо якості аудіосигналів дуже часто виникають у випадках створення систем з розпізнавання образів та мови, при проведенні процедур дереверберації записаних сигналів в приміщенні акустично не налаштованим для такого виду запису, при записі аудіокниг з текстового джерела (системи озвучування розмовного тексту), при налаштуванні та створенні голосових асистентів, які аналогічні світовим розробкам від компаній Google чи Amazon [1]. Нейронні мережі дозволяють при аудіоаналізі виділити важливі ознаки, які характерні, наприклад, для мовного сигналу – акцент, інтонація, артикуляція тощо. Натомість, застосування нейронних мереж дозволяє реалізувати системи розпізнавання мови, синтезу мови, і навіть допомагає виявляти аномалії в аудіоданих. До числа останніх відносять наприклад раннє виявлення звуків, пов'язаних з

хворобами у людині, насамперед, легень (стетоскопічні звуки), серця (аритмія). Також нейронні мережі можна використати при дослідженні аномалій у випадках функціонування різних машин та механізмів у промисловості. Їх виявлення дозволить уникнути відмов при роботі відповідного обладнання. Іншим аспектом застосування мереж може стати їх впровадження в системах відеонагляду, причому не лише в якості основи голосових асистентів, а й коли необхідно, задля підвищення рівня захисту території, швидко отримати аналіз і реакцію на незвичні звукові сигнали (вторгнення у приміщення, розбиття скла, тощо). Створення синтетичного мовного сигналу також стає можливим на основі залучення нейронної мережі і цей напрям успішно сьогодні реалізовано при створенні аудіовізуальних ефектів для комп'ютерних ігор та відеопродукту.

## **1.2 Аналіз сигналів мовлення**

Аналіз мовлення має багато різних форм і використовує різноманітні методи обробки сигналів. Класифікація мовлення на звучне та незвучне, оцінка основної частоти та ідентифікація формантів є типовими параметрами, які можна отримати за допомогою аналізу мовлення. Також можна відокремити специфічні характеристики мовлення, які стосуються конкретного завдання обробки мовлення. Наприклад, існує безліч алгоритмів виділення ознак, які можуть бути використані на початкових етапах систем розпізнавання мовлення. Ці методи виділення використовують різні форми аналізу мовлення для отримання характеристик, які дозволяють розрізняти різні звуки мовлення. Одним з важливих методів аналізу мовлення є аналіз лінійного передбачення (LPC)[2], частотний аналіз[3], та кепстральний аналіз[4].

Аналіз лінійного передбачення (LPC) — це метод обробки сигналів, який використовується для моделювання мовленнєвих сигналів. LPC аналізує сигнал, визначаючи його частотні характеристики, засновані на припущенні, що кожен зразок сигналу є лінійною комбінацією попередніх зразків. Він застосовується для виділення параметрів голосового тракту, таких як форманти,

і для кодування мовлення. LPC широко використовується у синтезі мовлення, розпізнаванні, а також у низькошвидкісних кодах завдяки ефективності у стисканні мовленнєвих даних. Аналіз лінійного передбачення використовується для стиснення даних, аналізу сигналів, його також використовують для забезпечення захищеного бездротового зв'язку, де голос необхідно оцифрувати, зашифрувати та передати через вузький голосовий канал.

Частотний аналіз — це метод дослідження сигналів, що визначає частотні компоненти, присутні в сигналі. Він перетворює сигнал з часової області в частотну, щоб показати, які частоти складають сигнал і з якою інтенсивністю. Зазвичай використовується для аналізу аудіо- та мовленнєвих сигналів, вібрацій, радіохвиль тощо. Найпоширенішим інструментом частотного аналізу є перетворення Фур'є, яке дозволяє виявити періодичні складові сигналу і зрозуміти його спектральні властивості. Формула перетворення Фур'є для частотного аналізу сигналу визначається як [5]:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt \quad (1.1)$$

де:

$X(f)$  — це комплексний спектр сигналу  $x(t)$  на частоті  $f$ ,  $x(t)$  — сигнал у часовій області,  $j$  — уявна одиниця (тобто  $j^2 = -1$ ),  $e^{-j2\pi ft}$  — комплексна експоненціальна функція, яка здійснює перетворення у частотну область. Частотний аналіз використовується в системах зв'язку при обробці сигналів тощо.

Аналіз амплітудного спектра мовленнєвого сигналу показав, що він включає як інформацію про спектральну огинаючу, пов'язану з частотною характеристикою голосового тракту, так і деталі від основної частоти та її гармонік. Формантний аналіз ефективно застосовується до спектральної огинаючої, тоді як оцінка основної частоти потребує тонких спектральних деталей.

Кепстральний аналіз — це метод гомоморфної обробки, що дозволяє розділити ці компоненти. Кепстральне перетворення визначається як обернене

дискретне перетворення Фур'є логарифмічного спектра потужності короткочасного фрагмента сигналу, як показано на рисунку 1.1 [5].

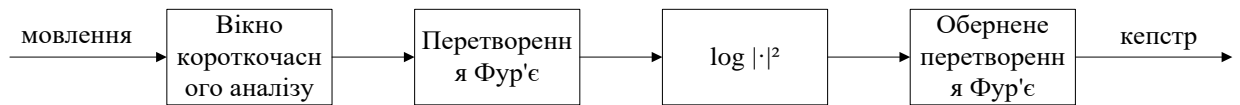


Рисунок 1.1 – Кепстральне перетворення мовленнєвого сигналу [5]

Вплив кепстрального перетворення на мовленнєвий сигнал можна пояснити таким чином. Вхідний мовленнєвий сигнал  $x(n)$  складається зі згортки двох компонент: імпульсної характеристики голосового тракту  $h(n)$  і сигналу збудження  $u(n)$ :

$$x(n) = h(n) * u(n)$$

Перетворення сигналу  $x(n)$  у логарифмічний спектр потужності,  $\log|X(f)|^2$ , перетворює операцію згортки на операцію додавання [5]:

$$\log|X(f)| = \log|H(f)| + \log|U(f)|.$$

Логарифмічний спектр потужності компонента голосового тракту,  $\log|H(f)|$  виглядає як спектральна огибаюча, що повільно змінюється вздовж осі частот. Компонент  $\log|U(f)|$  відображає тонкі спектральні деталі основної частоти та її гармонік і змінюється набагато швидше, ніж компонент голосового тракту. Для завершення кепстрального перетворення виконується обернене дискретне перетворення Фур'є логарифмічного спектра амплітуд,  $\log|X(f)|$ . Це дозволяє розділити компоненти голосового тракту та збудження сигналу. Повільно змінювана інформація про голосовий тракт розташована в області низьких кепстренцій кепстра (кепстренція — це область, яка утворюється після кепстрального перетворення і вимірюється в секундах). Компонент збудження, який змінюється швидше, знаходиться в області високих кепстренцій. Застосування низькочастотного віконного фільтра до кепстра зберігає інформацію про голосовий тракт, тоді як високочастотний



віконний фільтр зберігає інформацію про збудження [5]. На рисунку 1.2 зображено приклад кепстра фрагмента мовлення.

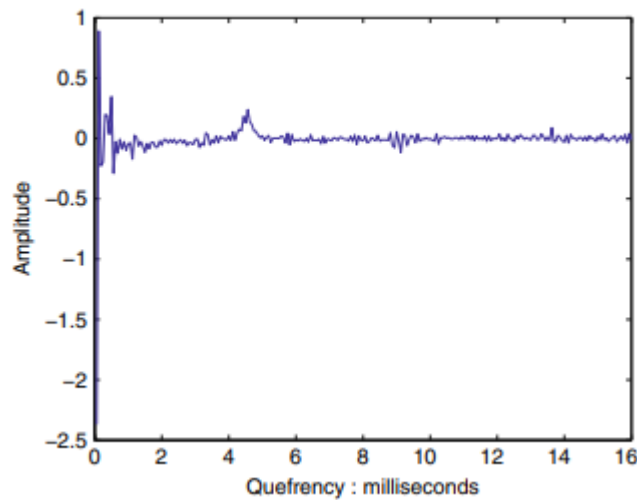


Рисунок 1.2 – Кепстр фрагмента (кадру) мовлення тривалістю 16 мс [5]

Кепстр є послідовністю кепстральних коефіцієнтів, які використовуються для аналізу спектральних характеристик, зокрема спектральної огинаючої, яка відображає особливості голосового тракту людини.

Більшість систем розпізнавання мовлення використовують варіант кепстральних ознак, відомий як MEL-частотні кепстральні коефіцієнти (MFCC)[6, 7]. Аналіз MFCC спрямований на виділення лише компоненти голосового тракту, використовуючи частотну шкалу MEL, що імітує чутливість людського вуха: висока роздільна здатність на низьких частотах і низька на високих. Алгоритм створення цих коефіцієнтів наведено у 2-му розділі роботи.

### 1.3 Оцінка розбірливості мовлення

При будь-якому перетворенні або генерації медіаконтенту одним з важливих факторів являється кількісна оцінка якості медіаконтенту. Розбірливість мовлення ( $SI$ ) є важливою для різних галузей досліджень, інженерії та діагностики, оскільки вона дозволяє кількісно оцінити різноманітні явища, такі як якість записів, пристроїв для комунікації та відтворення,

реверберацію у залах, характеристики порушення слуху, користь від використання слухових апаратів або поєднання цих факторів [5].

Функція розбірливості описує розбірливість мовлення слухача як функцію рівня мовлення  $L$  (у дБ), який може відноситися або до рівня звукового тиску мовленнєвого сигналу, або до співвідношення сигнал-шум, якщо тест проводиться в умовах завад.

У більшості випадків функцію  $SI(L)$  можна підігнати до емпіричних даних, використовуючи [5]:

$$SI(L) = \frac{1}{A} \left( 1 + SI_{max} \frac{A - 1}{1 + \exp\left(-\frac{L - L_{mid}}{s}\right)} \right)$$

де:

$L_{mid}$  – рівень мовлення в серединній точці функції розбірливості.

$s$ : параметр нахилу. Нахил в точці  $L_{mid}$  дорівнює  $\frac{SI_{max}(A-1)}{4As}$ .

$SI_{max}$ : параметр для максимальної розбірливості. У деяких випадках це значення може бути менше ніж 1 (наприклад, у випадках спотворених мовних сигналів або для слухачів з порушенням слуху). Асимптотичний максимум  $SI$  дорівнює  $SI_{max} + (1 - SI_{max})/A$

$A$ : кількість варіантів відповіді. Наприклад,  $A = 10$  коли слухач має відповісти у закритому форматі відповіді, використовуючи цифри від «0» до «9». У тестах  $SI$  з «відкритим форматом відповіді», таких як тести на слова без обмеження кількості варіантів відповіді, вважається, що  $A$  дорівнює нескінченності, що означає [5]:

$$SI = SI_{max} \frac{1}{1 + \exp\left(-\frac{L - L_{mid}}{s}\right)}.$$

### 1.4 Деревербації мовних сигналів

Деревербація — це процес усунення або зменшення ефекту реверберації в аудіосигналі, який виникає через багаторазові відбиття звуку від поверхонь приміщення, в якому проводиться запис цього аудіосигналу.

Рисунок 1.3 показує поєднання відомих схем обробки, а саме адаптивного диференційного мікрофона (ADM) [8] та бінауральної схеми зниження реверберації [9], що, можливо, представляє поточний оптимум для використання в умовах дифузного та ревербераційного шуму.

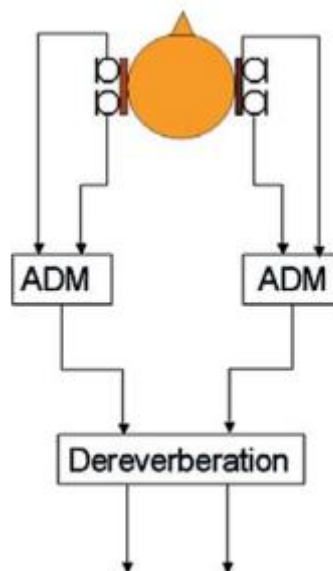


Рисунок 1.3 – Схема слухового апарату з функцією деревербації [5]

Загальна схема бінауральної обробки, що поєднує два адаптивні диференційні мікрофони (ADM), по одному на кожному вусі, та схему бінаурального зниження реверберації, яка забезпечує придушення одного джерела направленого шуму, а також дифузного фонового шуму та реверберації [5].

Важливим аспектом при розробці слухових апаратів є забезпечення чистого звуку та високої чіткості вимови [10]. Також нейромережі активно використовуються для передачі голосових команд через мікрофони, що відіграє

ключову роль в управлінні основними системами розумного будинку, де необхідна висока точність і чіткість сигналу [11, 12]. Одним з аспектів покращення якості звуку є дереверберація. Існує багато різноманітних методів дереверберації класичні та на основі нейронних мереж. Один з класичних методів дереверберації це інверсний фільтр [13].

Задача інверсної фільтрації для дереверберації полягає в застосуванні лінійного, зазвичай багатоканального, фільтра до сигналів мікрофонів для відновлення вихідного сигналу за допомогою процесу, відомого як (багатоканальна) еквалізація. Спочатку припускається, що імпульсні характеристики кімнати  $h$  відомі або можуть бути оцінені з достатньою точністю, наприклад, за допомогою сліпої ідентифікації системи. Потім багатоканальний інверсний фільтр  $g$  проектується за допомогою алгоритму еквалізації та застосовується в багатоканальному еквалайзері таким чином, що сума виходів фільтрів  $g_1, g_2, \dots, g_m$  формує  $\hat{s}(n)$ , що є оцінкою вихідного сигналу  $s(n)$  [14].

Враховуючи  $h$  та мікрофонні сигнали  $x_m(n)$  для  $m=1,2,\dots,M$ , потенційно можливо виконати точну дереверберацію за допомогою інверсної системи  $g$ , яка побудована у відповідній формі до  $h$  у рівнянні і складається з фільтрів  $g_m$ , що задовольняють рівняння [14]:

$$k_m^T g_m = k \delta(n - \tau)$$

де  $k$  та  $\tau$  — це довільний коефіцієнт масштабу та затримка. Зазначимо, що будь-які часові зміни  $h$  тимчасово не враховуються для ясності [14].

У літературі описані різноманітні методи та підходи для оцінки якості записаної аудіоінформації. Наприклад, у [15] автори пропонують застосовувати дифузійні генеративні або ймовірнісні моделі, засновані на статистичних фреймворках, для покращення розбірливості мовних записів. Але в роботі не в повному обсязі наведені параметри мовного сигналу та не описані обмеження до застосування на практиці цих моделей. В дослідженні [16] для оцінки якості дереверберації мовних сигналів пропонується практичний метод на основі статистичної моделі.

У роботі [17] запропоновано цікавий підхід до зменшення шуму в записаних мовних сигналах. Дослідження акцентує увагу на паузах між словами під час розмови, коли на записі присутній лише фоновий шум. Ці інтервали тиші використовуються для аналізу шумової динаміки, що дозволяє на основі методу спектрального віднімання та машинного навчання знижувати рівень шуму. Однак, основним недоліком цього підходу є його низька ефективність при наявності фонові музики під час розмови.

Для вирішення задачі зниження шуму та деверберації мовних сигналів у роботі [18] запропоновано метод, який комбінує локальну зважену регресію зі зваженими помилками передбачення. Однак цей метод не був протестований у випадках, коли при записі мовних сигналів присутні сторонні джерела шуму різної природи.

Нейронні мережі знайшли широке застосування в обробці аудіосигналів, особливо для підвищення їхньої якості. У статті [19] аналізується застосування методів машинного навчання для відокремлення мовних компонентів сигналу від шумів, а також розглядаються стандартні підходи до дереверберації. Чимало наукових робіт присвячено використанню нейронних мереж [20] як ключового інструменту для вирішення задач зниження шумів і реверберації. Наприклад, у роботі [21] запропоновано застосування глибокої нейронної мережі для усунення спотворень, спричинених впливом фонів шумів і реверберації в приміщенні. Використання цієї мережі планується в два етапи: спочатку для шумозаглушення, а потім для дереверберації. Дослідження [22] показує, що нейронні мережі активно застосовуються для автоматичного розпізнавання мовних сигналів, значно перевершуючи традиційні системи на базі прихованих марковських моделей за основними показниками ефективності.

Під час запису різноманітного аудіоконтенту, особливо живого мовлення, не завжди можна обрати приміщення з відповідними акустичними характеристиками. Це означає, що в умовах звичайних приміщень на якість мовних сигналів впливають реверберація простору [20] і зовнішній шум, який проникає або знаходиться в самому приміщенні. З акустичної теорії відомо, що

реверберація, яка виникає внаслідок багаторазового відбиття звукових хвиль від стін і предметів, погіршує чіткість мовлення, особливо у людей з вадами слуху. Основним завданням мовної дереверберації є збереження глобальної часової та спектральної інформації сигналу [23], що можна досягти за допомогою нейронних мереж [21].

Існує багато методів усунення реверберації на основі одно- та багатомікрофонних систем [23], таких як інверсна фільтрація [22, 24], багатокрокові лінійні прогнози [25], автоматичні мовні кодери та декодери [17], а також обробка спектральної області мовного сигналу з використанням методу найменших квадратів [17]. Проте, основним недоліком цих підходів є те, що вони не повною мірою враховують спектральну структуру мовлення, через що можуть втрачатися чіткі часово-частотні закономірності, характерні для мовного сигналу.

Ідея застосування нейронних мереж для обробки мовних сигналів виникла завдяки їх здатності успішно вирішувати складні завдання, такі як керування безпілотними транспортними засобами, розпізнавання об'єктів на зображеннях, голосове розпізнавання та машинний переклад. У нашому випадку, якщо перетворити записаний мовний сигнал з реверберацією та шумом у частотну форму, спектрограма може бути представлена як зображення. Завдання нейронної мережі полягає в аналізі та корекції цього зображення, щоб зберегти характерні особливості сигналу, такі як форманти, клацання, висота тону, а також вплив реверберації та адитивного шуму [26].

### **1.5 Адаптивні системи оброблення акустичної інформації**

Здатність нейронних мереж до адаптивності реалізується через використання методів машинного навчання, таких як, зворотне поширення помилки та алгоритми оптимізації, наприклад, метод стохастичного градієнтного спуску. Це дозволяє системам навчатися на основі отриманих акустичних даних і покращувати свої результати з кожною ітерацією (циклом).

У контексті персоналізації це означає, що системи можуть автоматично налаштовуватися під конкретні умови користувача, наприклад, адаптуватися до акценту, тембру голосу чи навколишнього шуму. Тим самим змінюються параметри системи, перерозподіляються ваги складових елементів всередині шару нейронної мережі.

Як вже було зазначено у вступі, нейронні мережі стали невід'ємною складовою сучасних технологій і знаходять застосування у багатьох галузях, таких як квантова хімія [27], фінанси [28], медицина [29], геоморфологія [30], гідрологія [31, 32] та прогнозування траєкторій [33]. Вони також активно використовуються для розпізнавання образів: облич [34], тексту [35], голосу [36] тощо.

На сьогоднішньому етапі розвитку технологій нейронні мережі все глибше інтегруються у повсякденне життя, і часто люди не усвідомлюють, що взаємодіють з ними. Приклади використання включають перетворення голосових команд у текст, розпізнавання тексту на зображеннях і перетворення його у цифрові файли, а також автоматичне розпізнавання номерних знаків [37] при порушеннях правил дорожнього руху. Згідно зі статтею [38], нейронні мережі також знаходять застосування в розпізнаванні домінуючих інструментів у поліфонічних музичних творах.

Перший успішний випадок застосування нейронних мереж для розпізнавання зображень відбувся у 1989 році, коли компанія Bell Labs створила систему для ідентифікації рукописних цифр. Ця нейронна мережа, відома як LeNet, почала використовуватись поштовою службою США у 1990 році для автоматичного розпізнавання поштових індексів на конвертах [39]. Проте розвиток нейронних мереж на той час був повільним через обмежені можливості обчислювальної техніки. Ситуація змінилась із появою більш потужних обчислювальних технологій, зокрема завдяки впровадженню CUDA, що стало ключовим моментом для швидкого прогресу нейронних мереж у 2010-х роках. Деякі дослідники почали створювати CUDA-реалізації нейронних мереж, серед перших були Ден Кайєсан [40] та Алекс Крижевський [41].

Сьогодні нейронні мережі досягли значних проривів у складних сферах машинного навчання [42], зокрема:

- класифікація зображень на рівні, порівнянному з людськими можливостями;
- успішне розпізнавання рукописного тексту з високою точністю;
- значне поліпшення якості машинного перекладу між різними мовами;
- вдосконалення технологій перетворення тексту на голос із природнішим звучанням.
- розпізнавання мови на рівні точності, еквівалентному людині;

Одним з класичних методів вирішення задачі розпізнавання мовлення є приховані марківські моделі [42]. Приховані марківські моделі (ПММ) є математичним інструментом для моделювання часових послідовностей даних, зокрема в галузі розпізнавання мовлення. Розпізнавання мовлення — це процес автоматичного перетворення мовленнєвих сигналів у текст або команди, які може розуміти комп'ютер. ПММ дозволяють ефективно моделювати послідовності звуків у мовленні, враховуючи їхню імовірнісну природу та часові залежності.

ПММ складаються з двох основних компонентів набір станів та набір спостережень. Набір станів це кінцевий набір можливих станів системи, які не спостерігаються безпосередньо (приховані). У контексті мовлення стани можуть відповідати фонемам або частинам фонем. Набір спостережень це вихідні дані, які можна виміряти. У розпізнаванні мовлення це акустичні ознаки, наприклад, мел-кепстральні коефіцієнти (MFCC), які отримуються з мовленнєвого сигналу. Крім того, прихована марківська модель має два типи параметрів: ймовірності переходів та ймовірності виходів. Під час застосування ПММ генерує певну послідовність спостережень  $T = T_1, T_1 \dots T_n$ , де кожне спостереження  $T_n$  є символом алфавіту, а  $n$  — це кількість символів у спостережуваній послідовності.

Навчання параметрів у прихованих марківських моделях полягає у визначенні оптимального набору ймовірностей переходів між станами та



ймовірностей виходів для заданої послідовності спостережень або набору таких послідовностей. Зазвичай це завдання зводиться до оцінювання цих параметрів ПММ методом максимальної правдоподібності на основі даного набору спостережуваних послідовностей. З іншого боку, ПММ має ряд особливостей які необхідно враховувати при їх використанні:

- наявність великого набору навчальних даних, особливо коли ці моделі використовуються для мовного контенту (набір фраз та словосполучень);
- необхідність проведення фонетичного аналізу зі створенням гіпотез, які визначають той факт, що дана фраза з послідовності фонем пояснює фрагмент звукового сигналу;
- необхідність проведення оцінки вірогідності для кожної гіпотези того, що дана послідовність фонем відповідає заданому фрагменту аудіосигналу.

### **1.5.1 Розпізнавання та синтез мови**

Ідентифікація мовлення та синтез голосу є одними з важливих завдань у сфері машинного навчання [43]. На сьогоднішній день технології розпізнавання мовлення інтегровані у смарт-пристрої [44], системи "розумний будинок" [45] та використовуються для керування транспортними засобами [46]. Окрім цього, технології ідентифікації голосу широко застосовуються в системах безпеки, що займаються аутентифікацією та верифікацією користувачів при доступі до захищених персональних даних [47-53].

Синтез мовлення ґрунтується на автоматичному перетворенні текстової інформації в усне мовлення [54]. Крім того, його використовують для клонування голосу та створення підроблених голосових біометричних даних людини [55-58].

Клонування голосу — це завдання, яке полягає у навчанні синтезувати людський голос за допомогою комп'ютерних технологій на основі попередньо записаних зразків мовлення. Основна мета цієї процедури полягає в тому, щоб при відтворенні мовних сигналів максимально зберегти природність і унікальні

особливості оригінального голосу. Основною метою клонування є збереження природності мовлення, його якості та максимальної схожості з оригінальним голосом [59]. Можливість збереження природності мови надає використання MEL-спектрограм для навчання нейронної мережі. Приклад MEL-спектрограми наведено на рисунку 1.4.

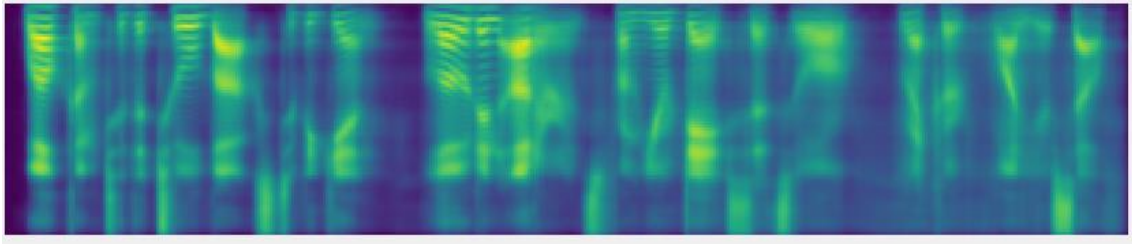


Рисунок 1.4 – Приклад MEL-спектрограми [60]

Такі можливості спричинили зростання використання аудіопідробок, які становлять загрозу як засіб обману [61, 62], що в свою чергу дало поштовх до створення систем ідентифікації голосу на основі використання нейронних мереж як інструменту для захисту від суффінг-атак [45, 50, 56].

Варто зазначити, що найбільш вразливим елементом в системах голосової ідентифікації є ситуації, коли мовний сигнал генерується на основі оригіналу, з урахуванням індивідуальних біометричних особливостей користувача через спеціальні налаштування [55].

Більшість систем голосового доступу базуються на унікальних біометричних характеристиках користувача, зокрема голосу. Процес ідентифікації в таких системах складається з кількох етапів: запис мовного зразка, початкова обробка сигналу, а також аналіз ознак. Доступ може здійснюватися як на основі текстозалежного, так і текстонезалежного висловлювання [47, 49]. Подальші етапи включають сегментацію сигналу, виділення мовних (акустичних) характеристик, їхній аналіз за допомогою класифікатора, що визначає, чи належить сигнал до оригіналу, чи є підробкою (спуфінг).

Деякі розробники пропонують використовувати складні архітектури глибоких нейронних мереж, поєднуючи згорткові та рекурентні нейронні мережі, для більш точного розпізнавання звукових сигналів. Такий підхід, як зазначено, ефективний для ідентифікації коротких фраз [45]. В іншому дослідженні, представленою в роботі [51], розглядається метод верифікації користувачів у випадках, коли є короткі фрази або фоновий шум. На етапі виділення мовних ознак запропоновано використовувати перетворення ентропії Фур'є як нову характеристику для опису ознак. Проте автори не обговорюють питання надмірності даних, яке може виникати при цьому методі.

Технологія синтезу голосу використовується не тільки в клонуванні голосу для створення аудіопідробок але також знайшла своє застосування в таких сферах, як автоматичний дубляж для кіно та телебачення, а також у створенні голосових чат-ботів і інших інформаційних систем [63].

Проблема клонування голосу [54, 58] тісно пов'язана з активним впровадженням нейронних мереж, популярність яких значно зросла в останні роки. До того ж, сучасні дослідження в сфері штучного інтелекту все частіше базуються на методах глибокого навчання, зокрема на алгоритмах, що моделюють біологічні нейронні мережі та включають функцію навчання [64, 65].

Варто зазначити, що перші дослідження у сфері створення синтезованого звучання були тісно пов'язані з концепцією "конкатенації" мовлення [63]. Цей метод полягає в поділі записаного звукового сигналу на невеликі сегменти, які згодом об'єднуються в нові мовні конструкції. При цьому враховуються характерні риси, визначені в еталонному шаблоні записаного зразка звуку.

У дослідженні [65] було проаналізовано використання згорткової нейронної мережі глибокого навчання для класифікації аудіозаписів. Цей метод дозволяє обробляти вхідні дані у вигляді спектрограм — візуальних зображень звуку. В роботі [65] зазначено, що така технологія може використовуватися для виявлення різноманітних характеристик аудіозаписів, зокрема для ідентифікації диктора, визначення його статі, класифікації музичного жанру висловлювання,

а також для розпізнавання різних музичних інструментів, що супроводжують мовний сигнал. [60]

Системи для клонування голосу мають складну архітектуру і включають три окремі нейронні мережі [66-68], що спільно вирішують завдання відтворення голосу.

Аналіз статей показує, що нейронні мережі дедалі ширше використовується в повсякденні. І використання нейронних мереж для задач, пов'язаних з мовними сигналами є досить широким і наразі активно розвивається.

## **1.6 Типи нейронних мереж для оброблення акустичної інформації**

У сучасних технологіях обробки акустичної інформації нейронні мережі відіграють ключову роль, завдяки своїй здатності адаптуватися до різних умов і вирішувати складні завдання. Ці технології дозволяють здійснювати високоточний аналіз мовленнєвих сигналів, ефективно шумозаглушення, розпізнавання мовлення, синтез мовлення та навіть клонування голосу. Нейронні мережі здатні виявляти специфічні особливості акустичних сигналів і адаптуватися до їх складних структур.

Сучасні мови програмування, такі як Python та MATLAB, надають потужні інструменти для створення та впровадження нейронних мереж. MATLAB, наприклад, оснащений Deep Learning Toolbox, що забезпечує всі необхідні функції для розробки та застосування моделей. Python, у свою чергу, підтримує широке коло популярних бібліотек, таких як Keras, TensorFlow та PyTorch, які дозволяють застосовувати, покращувати та розробляти нейронні мережі.

Нейронні мережі являють собою обчислювальні моделі, в яких велика кількість простих елементів (нейронів) працюють паралельно, не маючи єдиного централізованого управління [69]. Кожен нейрон володіє власними ваговими коефіцієнтами, і саме через ретельний підбір цих коефіцієнтів можна отримати правильні результати від нейронної мережі. Цей процес

налаштування вагових параметрів здійснюється під час навчання мережі, коли вона адаптується до вхідних даних і коригує свої параметри для досягнення бажаних результатів [70]. Одним із популярних інструментів для створення нейронних мереж є фреймворк Keras, який спрощує розробку складних моделей, зокрема систем для розпізнавання мовлення [71].

### 1.6.1 Рекурентні нейронні мережі

Рекурентні нейронні мережі (RNN) — це спеціалізовані архітектури штучних нейронних мереж, розроблені для роботи з послідовними даними, такими як аудіо, текст або інші дані, що мають часову залежність. Головною особливістю RNN є здатність запам'ятовувати інформацію з попередніх елементів послідовності через використання прихованих станів, що передаються від одного кроку до наступного. Це дозволяє аналізувати поточні елементи даних у контексті всієї послідовності.

Сучасні розширення RNN, такі як LSTM (Long Short-Term Memory), дозволяють долати проблему зникнення градієнта, забезпечуючи обробку навіть довготривалих залежностей у послідовностях. Це робить RNN важливим інструментом у сфері штучного інтелекту

LSTM мережі широко використовуються для обробки послідовних даних, таких як аудіосигнали. Основним елементом архітектури є рекурентні нейронні мережі, які дозволяють зберігати довготривалі залежності в часових рядах, що є важливим для аналізу голосу та розпізнавання мови.

Модель використовує LSTM шари для обробки послідовностей MFCC. У другій половині 90-х років XX століття Сепп Хохрайтер разом із колегами розробили вдосконалені варіанти штучних нейронів, які отримали назву LSTM (Long Short-Term Memory). LSTM-мережі стали найпоширенішою формою рекурентних нейронних мереж. Основними елементами таких мереж є комірки пам'яті та "вентилі", зокрема, вентиль входу та вентиль забуття. Коли ці вентилі зачинені, вміст блоку (нейрона) залишається незмінним при переході між

часовими кроками. Завдяки вентиляційним механізмам модель може зберігати інформацію в комірці пам'яті протягом великої кількості часових інтервалів. Моделі на основі LSTM здатні зберігати інформацію протягом певного часу і використовувати її для прогнозування майбутніх подій. LSTM шари обробляють послідовні дані, зберігаючи контекст попередніх кроків через приховані стани, виявляють часові залежності в даних та формують приховані представлення для кожного кроку в послідовності. LSTM-мережі використовують власну внутрішню пам'ять для обробки послідовностей різної довжини, до яких можна зарахувати аудіосигнали.

Вихідні значення рекурентних нейронних мереж обробляють повнозв'язні шари, ще їх називають пов'язані, для остаточного узагальнення. Вони виконують трансформації для отримання прогнозів або класифікації та забезпечують перетворення багатовимірних даних у вихідні формати (наприклад, ймовірності, класи).

Для додавання нелінійності до шарів нейронної мережі, обмеження значень прихованих станів, стабілізації градієнтів і забезпечення можливості моделювання короткотривалих і довготривалих залежностей використовуються функції активації. Вони грають критичну роль у тому, щоб модель залишалася стабільною і точно передавала контекст у часі. Одними з можливих функцій активації є ReLU, Leaky ReLU та Softmax.

Функція ReLU обраховується наступним чином [72]:

$$f(x) = \max(0, x) \quad (1.2)$$

Функція Leaky ReLU визначається наступним чином [72]:

$$f(x) = \begin{cases} x, & \text{якщо } x \geq 0 \\ \alpha x, & \text{якщо } x < 0 \end{cases} \quad (1.3)$$

де  $\alpha$  — невеликий коефіцієнт (використовується 0.2), який визначає нахил для від'ємних значень.

Функція Softmax обраховується наступним чином [72]:

$$P_j = \frac{e^{z_j}}{\sum_{k=1}^N e^{z_k}} \quad (1.4)$$

де  $P_j$  це ймовірність, що зразок належить класу  $j$ ,  $z_j$  це не нормоване передбачення моделі для класу  $j$ ,  $N$  це загальна кількість класів,  $e$  це основа натурального логарифма 2.718.

### 1.6.2 Згорткові нейронні мережі

Згорткові нейронні мережі (CNN) — це спеціалізований тип глибоких нейронних мереж, призначений для обробки просторових даних, таких як зображення, аудіосигнали та інші. Головна особливість CNN полягає у здатності автоматично витягувати ознаки з вхідних даних. Використовуючи операцію згортки, мережа аналізує невеликі частини вхідних даних (так звані фільтри), що дозволяє виявляти важливі деталі, наприклад, контури, текстури або патерни. Вони ефективно працюють завдяки своїй здатності зменшувати розмірність вхідних даних завдяки операції згортки, зберігаючи їхню суттєву інформацію для виділення ознак різної складності.

Згорткова операція є центральною складовою згорткових нейронних мереж. Формально, згортка в контексті обробки зображень полягає у “ковзанні” фільтра (ядра згортки) над вхідним зображенням та обчисленні скалярних добутків між фільтром та локальним підфрагментом зображення. Кожен фільтр навчається вилучати певний тип ознак, наприклад, контури, текстури чи складніші патерни. Після згортки застосовуються нелінійні активаційні функції, найчастіше ReLU, яка дозволяє моделі краще виражати складні залежності.

### 1.6.3 Алгоритм зворотного поширення помилки

Сучасні нейронні мережі змінюють свої ваги під час навчання за допомогою алгоритму зворотного поширення помилки. Цей метод був розроблений Румельхартом, Хінтоном і Вільямсом у 1986 році, спеціально для навчання багатoshарових мереж [73]. Уявімо, що ми маємо мережу з  $M$  шарами

та навчальну вибірку  $E = \{x_1, \dots, x_N\}$ , а також набір правильних відповідей мережі  $y_1, \dots, y_N$ . Тут  $y_i$  представляє вектор значень, які мають бути отримані на виході останнього шару мережі, коли на вхід першого шару подається вектор  $x_i$  (рисунок 1.5).

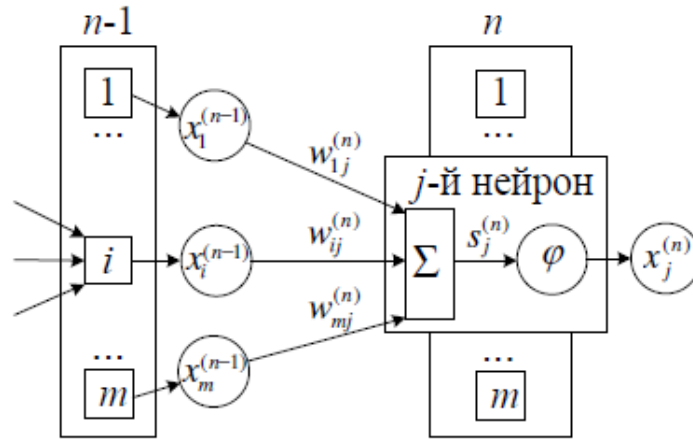


Рисунок 1.5 – Багатошарова нейронна мережа [73]

Нехай між  $i$ -м нейроном шару  $n-1$  і  $j$ -м нейроном шару  $n$  існують випадкові зв'язки, позначені як  $w_{ij}^{(n)}, x^{(n)}$ , де  $x^{(n)}$  — це вектор вихідних значень для  $n$ -го шару, і  $n$  варіюється від 1 до  $M$  (де  $x^{(M)}$  — це вектор виходу останнього шару, тобто всієї мережі). Завдання навчання багатошарової мережі полягає в тому, щоб знайти набір вагових матриць  $W^{(n)} = w_{ij}^{(n)}$  для кожного  $n$  від 2 до  $M$ , таким чином, щоб середньоквадратична помилка класифікації була мінімальною. Тобто, потрібно мінімізувати функцію помилки [73]:

$$F(W^2, \dots, W^{(M)}) = \sum_{x \in E} \|x^{(M)} - y\|^2 = \sum_{x \in E} \sum_j (x_j^{(M)} - y_j)^2 \rightarrow \min.$$

Мінімізація цього функціоналу помилки здійснюється за допомогою методу градієнтного спуску, шляхом поступового коригування ваг кожного шару. В  $j$ -му нейроні  $n$ -го шару перетворення вхідного сигналу відбувається за формулою [73]:



$x_j^{(n)} = \varphi(s_j^{(n)})$ , де  $x_j^{(n)} = \varphi(s_j^{(n)})$ , а  $\varphi$  є вибраною активаційною функцією. На  $n$ -му шарі зміна ваги  $w_{ij}^{(n)}$  виконується на величину:

$$\Delta w_{ij}^{(n)} = -h \frac{\partial F}{\partial w_{ij}^{(n)}} = -h \frac{\partial F}{\partial x_j^{(n)}} \frac{dx_j^{(n)}}{ds_j^{(n)}} \frac{\partial s_j^{(n)}}{\partial w_{ij}^{(n)}} = -h \delta_j^{(n)} x_i^{(n)}, \quad (1.5)$$

де  $\delta_j^{(n)} = \frac{\partial F}{\partial x_j^{(n)}} \frac{dx_j^{(n)}}{ds_j^{(n)}}$ . Величина  $\delta_j^{(n)}$  виражає помилку на  $n$ -му шарі в  $j$ -му нейроні і визначає величину помилки для корекції ваг.

Розглянемо, як змінюється величина помилки  $\delta_j^{(n)}$  при переході з  $n$ -го шару на  $(n+1)$ -ий шар. Вираз для  $\delta_j^{(n)}$  буде наступним [73]:

$$\begin{aligned} \delta_j^{(n)} &= \frac{\partial F}{\partial x_j^{(n)}} \frac{dx_j^{(n)}}{ds_j^{(n)}} = \left( \sum_k \frac{\partial F}{\partial x_k^{(n+1)}} \frac{dx_k^{(n+1)}}{ds_k^{(n+1)}} \frac{\partial s_k^{(n+1)}}{\partial x_j^{(n+1)}} \right) \frac{\partial x_j^{(n)}}{\partial s_j^{(n)}} \\ &= \left( \sum_k \delta_k^{(n+1)} w_{jk}^{(n+1)} \right) \frac{\partial x_j^{(n)}}{\partial s_j^{(n)}}, \end{aligned}$$

оскільки  $\frac{\partial s_k^{(n+1)}}{\partial x_j^{(n+1)}} = w_{jk}^{(n+1)}$ . Отже, отримуємо ітераційну формулу для розрахунку коефіцієнтів  $\delta_j^{(n)}$ :

$$\delta_j^{(n)} = \left( \sum_k \delta_k^{(n+1)} w_{jk}^{(n+1)} \right) \frac{\partial x_j^{(n)}}{\partial s_j^{(n)}}, \quad n = M - 1, M - 2, \dots, 1. \quad (1.6)$$

На останньому шарі формула виглядає так [73]:

$$\delta_j^{(M)} = (x_j^{(n)} - y_j) \frac{\partial x_j^{(M)}}{\partial s_j^{(M)}}. \quad (1.7)$$

Зазначимо, що похідні  $\frac{dx_j^{(n)}}{ds_j^{(n)}} = \varphi'(s_j^{(n)})$  – це похідні від активаційної функції  $\varphi$ . Процес обчислення значень  $\delta_j^{(n)}$  за формулами (1.6) і (1.7) можна розглядати як зворотне поширення помилки, яке проходить від вихідного шару до вхідного [73].

Алгоритм навчання нейронної мережі методом зворотного поширення помилки виконується за такими етапами [73]:

1. Ініціалізуються початкові значення вагових матриць:  $W^2, \dots, W^M$ .
2. На вхід першого шару мережі подається навчальний вектор  $x$ . В звичайному режимі роботи обчислюються всі значення  $s_j^{(n)} = \sum_i w_i^{(n-1)} w_{ij}^{(n)}$ , де  $n = 1, \dots, M, j = 1, \dots, N$ .
3. За формулою (1.6) (або (1.7) для вихідного шару) обчислюються значення помилок  $\delta_j^{(n)}$ .
4. Ваги на поточному шарі коригуються відповідно до формули (1.5).
5. Аналогічно, виконавши зворотне поширення помилки за формулою (1.6), проводиться корекція ваг на інших шарах.
6. Перевіряється умова зупинки, яка полягає в стабілізації критерію мінімізації  $F$  (тобто  $F^{k+1} = F^k$ ): якщо  $F$  стабілізувався, алгоритм завершується, інакше — повертаємось до пункту 2.

Концепція оновлення вагів в зворотному напрямку на даний момент є основним для навчання нейронних мереж. Всі сучасні алгоритми використовують цей підхід з використанням різних оптимізацій для зменшення похибки, один з найрозповсюдженіших алгоритмів оптимізації є метод Адама (метод стохастичної оптимізації) [74].

## **2 ОГЛЯД ЗАГАЛЬНИХ ЗАСАД СТВОРЕННЯ АДАПТИВНИХ СИСТЕМ ОБРОБЛЕННЯ АКУСТИЧНОЇ ІНФОРМАЦІЇ**

Системи оброблення акустичної інформації використовуються в різних галузях техніки. Серед можливих напрямків є такі галузі, як стиснення акустичної інформації, музична обробка та інші. Зокрема, найбільшу зацікавленість викликають, системи ідентифікації особи за голосовими біометричними даними, розпізнавання мови, підвищення розбірливості та клонування голосу.

### **2.1 Теоретичні підходи побудови систем ідентифікації за голосом**

Системи ідентифікації за голосом останнім часом набувають все більшої популярності, оскільки сфера їх застосування визначається не лише окремими областями побутової діяльності людей (додатки та прилади в межах технології “Розумний будинок”), але й поступово зачеплює сфери функціонування державних органів країни, які пов’язані насамперед з налагодженням систем пропуску в приміщення державних установ, банків, офісів. Окремим можливим застосуванням системи ідентифікації за голосом можуть бути пункти контролю за переміщеннями людей на вокзалах, аеропортах та в пунктах перетину митних кордонів. Одним з підходів реалізації таких систем є використання інтелектуальних підходів, на основі впровадження можливостей нейронних мереж.

Розпізнавання особи за голосом реалізується через процес адаптації системи ідентифікації, тобто на основі формування бази даних осіб і для цього має бути підготовлена вибірка для навчання нейронної мережі. У цьому підході ключовим етапом є попередня обробка аудіосигналів, які перетворюються в MEL-кепстральні коефіцієнти — числові представлення звукових даних. Ці коефіцієнти є компактними й інформативними ознаками, що дозволяють

системі ефективно аналізувати голосові характеристики для точного ідентифікування користувача.

### **2.1.1 MEL-кепстральні коефіцієнти**

MEL-кепстральні коефіцієнти (MFCC) це набір коефіцієнтів, що широко використовуються для представлення звукових сигналів в задачах розпізнавання мови, класифікації музики та інших акустичних аналізах. Цей метод полягає в тому, щоб перетворити складний звуковий сигнал на набір характеристик, які можуть бути легко оброблені алгоритмами машинного навчання. MFCC дозволяють стиснути важливу інформацію про звук, зберігаючи при цьому ключові акустичні особливості, що важливі для розпізнавання.

Основний принцип роботи MFCC полягає в тому, щоб аналізувати звук з урахуванням того, як людське вухо сприймає різні частоти. Людина має нелінійну здатність до сприйняття частот, ми більш чутливі до низьких частот і менш до високих. Крім цього, є залежність слухового сприйняття до змін частот в окремих діапазонах. Ці закономірності в 1964 відобразив у своїй науковій праці Мелвін Джонсон [1]. Також він запропонував використовувати MEL-шкалу частот, яка імітує сприйняття людського слуху. Мел-шкала це спеціальна шкала, яка відображає те, як людина сприймає різницю між звуками різних частот. Частоти в нижньому діапазоні (до 1000 Гц) розрізняються більш точно, тому мел-шкала розташовує їх ближче одна до одної. Натомість для високих частот (понад 1000 Гц) людське вухо менш чутливе, і мел-шкала робить їх більш розрідженими.

Варто зазначити, що процедурі отримання MFCC передуює передобробка аудіосигналу. Тобто, проводиться процедура так званого “поділу на фрейми”, а точніше поділу неперервного аудіо потоку на маленькі часові сегменти по 15-30 мс. За таких умов при подальшому аналізі, коли коефіцієнти розраховуються для кожного такого фрейму, можна врахувати локальність і зміни в сигналі за

чом (переходи між різними звуками, амплітудні варіації, тощо). При цьому рекомендується поділ з перекриттям цих сегментів, щоб врахувати кореляцію між сусідніми сегментами. Далі, в межах попередньої обробки, як правило застосовується перетворення Фур'є за формулою (1.1) з розкладом сигналу на набір синусоїдальних компонент різних частот та процедура логарифмування до амплітуди кожної компоненти [1].

Перехід від частотного спектру до мел-шкали яка враховує особливості слухового сприйняття людиною, виконується через спеціальне перетворення, де звичайні герци перетворюються на мели за допомогою формули [75]:

$$Mel(f) = 2595 \log_{10}(1 + \frac{f}{700}).$$

Ця формула показує, що частотна шкала після перетворення стає нелінійною, що допомагає краще моделювати наше сприйняття звуків. Як результат, низькі частоти, на які ми звертаємо більше уваги, розподіляються з більшою точністю, а високі частоти, що сприймаються менш чутливо, згруповані в ширші діапазони. Це перетворення дозволяє більш точно захопити властивості звукового сигналу, наближаючи його аналіз до особливостей людського слуху.

Кепстр обраховується через логарифмічний спектр потужності, а обрахування MFCC на цьому етапі відбувається інакшим чином. Для обрахунку MFCC відбувається накладання банку фільтрів мел на спектральну потужність [75]  $M$  трикутних фільтрів  $H_m(k)$ , рівномірно розташованих на мел-шкалі. Кожен фільтр перекривається з сусідніми, охоплюючи певний діапазон частот рисунку 2.1. Такий підхід застосування фільтрів запроваджується, щоб точно врахувати особливості сприйняття слуху людиною. Ці фільтри “імітують” те, як людське вухо сприймає звуки.

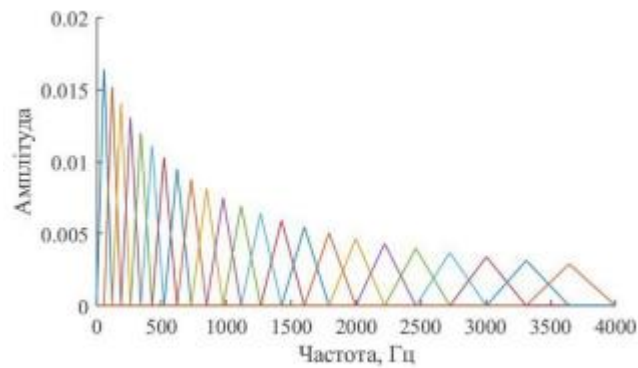


Рисунок 2.1 – Банк фільтрів [75]

Застосування мел-фільтрів до спектральної потужності відбувається наступним чином [75]:

$$S_m = \sum_{k=1}^N P(k)H_m(k), \quad m = 1, 2, \dots, M$$

На цьому етапі ми отримуємо мел-коефіцієнти  $S_m$ , що відображають характеристики звукового сигналу на мел-шкалі частот. Далі треба обрахувати логарифм мел-коефіцієнтів для стиску динамічного діапазону та перетворення мультиплікативних ефектів в адитивні

$$\log S_m, \quad m = 1, 2, \dots, M$$

і після цього кроку маємо логарифмовані мел-коефіцієнти, готові до перетворення в мел-кепстральні коефіцієнти. Для цього застосовуємо дискретне косинусне перетворення до логарифмованих мел-коефіцієнтів.

$$MFCC_n = \sum_{m=1}^M \log S_m \cdot \cos\left(\frac{\pi n(m - 0.5)}{M}\right), \quad n = 1, 2, \dots, L$$

де  $L$  – кількість коефіцієнтів.

Отримані мел-кепстральні коефіцієнти формують мел-кепстр, приклад мел-кепстру наведений на рисунку 2.2.

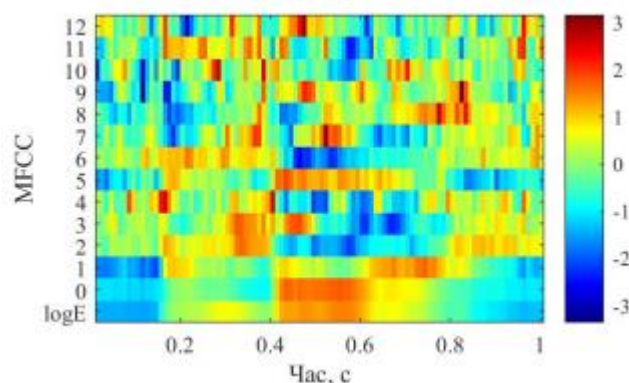


Рисунок 2.2 – Мел-кепстральні коефіцієнти представлені у вигляді бітової карти [75]

### 2.1.2 Регуляризація

При побудові систем ідентифікації за голосом, систем покращення якості розбірливості мови або клонування голосу з використанням нейронних мереж часто трапляються перенавчання системи під час відповідно навчання і це є причиною гірших результатів роботи системи на нових даних, а саме даних які нейронна мережа не бачила під час навчання. Для запобігання перенавчання використовується регуляризація.

Регуляризація — це техніка, яка використовується в машинному навчанні для запобігання перенавчанню моделі. Перенавчання виникає, коли модель занадто добре підлаштовується під тренувальні дані, запам'ятовує їх деталі та шум, але погано узагальнює інформацію на нових даних. Одними з технік регуляризації є L2 регуляризація та дропаут [72].

L2 регуляризація (також відома як Ridge-регуляризація) — це техніка регуляризації, яка додає штраф до функції втрат у нейронній мережі або іншій моделі машинного навчання, щоб контролювати величини ваг моделі. L2 регуляризація додає штраф до функції втрат, який пропорційний сумі квадратів ваг усіх параметрів моделі. Це означає, що модель намагається мінімізувати не тільки помилку передбачення, але й ваги нейронів. Оскільки ваги моделі будуть меншими, модель буде менш схильною до перенавчання. Обрахунок функції втрат з L2 регуляризацією відбувається наступним чином[72]:

$$L = \text{Loss Function} + \lambda \sum_{i=1}^n w_i^2$$

де:

$L$  — загальна функція втрат.

*Loss Function* — стандартна функція втрат моделі.

$\lambda$  — гіперпараметр регуляризації, який контролює інтенсивність штрафу за великі ваги.

$w_i$  — ваги моделі.

Дропаут (Dropout) — це популярна техніка регуляризації, яка використовується для запобігання перенавчанню в нейронних мережах. Вона була запропонована для того, щоб зробити модель більш узагальнюючою і стійкою до шуму у тренувальних даних. Dropout випадково вимикає (деактивує) частину нейронів під час навчання моделі, змушуючи модель навчатися більш узагальнених ознак [72].

### 2.1.3 Структура систем ідентифікації за голосом

Структура системи ідентифікації за голосом складається з процедур запису голосу, його відтворення, обрахунку MFCC на основі завантаженого голосу, обробкою послідовності мел-кепстральних коефіцієнтів нейронною мережею, яка побудована на основі згаданих раніше рекурентних шарів та повнозв'язних шарів (див.п.1.6.1).

На рисунку 2.3 зображена спрощена структура системи ідентифікації за голосом.

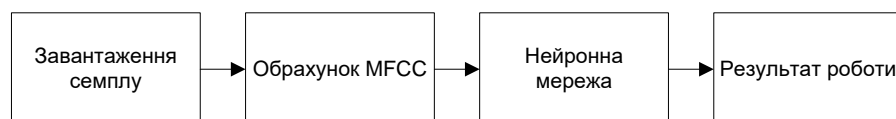


Рисунок 2.3 – Структура системи ідентифікації за голосом



Нейронна мережа рекурентного типу для розробленої системи ідентифікації за голосом складається з 4 внутрішніх шарів з LSTM та 4 повнозв'язних шарів. Всього модель містить 8 прихованих шарів кількість яких можна змінювати в експерименті, що наведено буде далі в роботі. Рекурентні шари використовуються для обробки послідовних мовних даних, де важливо враховувати залежності між елементами послідовності. Повнозв'язні шари використовуються для узагальнення ознак після їх обробки в інших шарах та класифікації голосу до певної особи за голосовими характеристиками.

Систему перед використанням слід навчити ідентифікувати мовців за голосом. Для ідентифікації особи система попередньо має бути навчена на персоналізованих семплах. Для цього, семпли з голосом мовців групуються по групам (класифікація за мітками) де кожна група відповідає мовцю і система під час навчання вчиться ідентифікувати мовців. При попередній обробці семплів обраховуються MFCC. MFCC дозволяють виділити найважливіші акустичні характеристики сигналу, які відображають його спектральні властивості, перетворені у компактну і зрозумілу форму для моделі.

Також наведена структура підходить для задач розпізнавання мови тільки при навчанні слід групувати навчальні дані не за приналежністю записаного семплу до мовця, а за записаними словами в семплі.

## **2.2 Теоретичні підходи побудови систем підвищення розбірливості мови**

На розбірливість мовлення впливають багато різних чинників, і один з них може виникнути, коли запис мовних сигналів проводиться в приміщенні, яке акустично не адаптовано під цей запис. Наприклад, при цьому може вплинути на якість запису не характерні значення часу реверберації, які можна оцінити в цьому приміщенні шляхом вимірювання імпульсної характеристики.

Таким чином, підвищення розбірливості вже записаних аудіосигналів можливо через проведення процедури дереверберації. Реалізацію такої процедури використаємо в рамках дослідження нейронної мережі.

### 2.2.1 Реверберація

Реверберація — це явище багатократного відбиття звукових хвиль від поверхонь у приміщенні, що призводить до тривалого затухання звуку після припинення його випромінювання. Цей процес є важливим фактором при обробці мовних сигналів, оскільки реверберація може суттєво вплинути на розбірливість мовлення, а також ускладнити роботу систем розпізнавання мовлення та клонування голосу.

Коли звуковий сигнал генерується у замкнутому середовищі, наприклад, у кімнаті, він поширюється у всіх напрямках. Частина цього сигналу досягає слухача (або мікрофона) безпосередньо — це прямий сигнал. Однак значна частина хвиль відбивається від стін, підлоги, стелі та інших поверхонь, створюючи відбиті сигнали. Через відбиття сигнал приходить до слухача із затримкою, що викликає ефект реверберації.

Тривалість цього затухання, тобто час, за який рівень звуку зменшується до нечутного стану, називається часом реверберації. Час реверберації — це час, за який рівень звукового тиску зменшується на 60 дБ після припинення звукового випромінювання. Цей показник є важливою характеристикою акустичного середовища і визначає, наскільки довго звук «відлунюється» у приміщенні після припинення його джерела. Для кількісного опису цього явища застосовується формула Сабіна, яка дозволяє розрахувати час реверберації на основі об'єму приміщення та акустичних властивостей його поверхонь. Формула Сабіна визначається наступним чином [5]:

$$T = \frac{0,16 \cdot V}{S\bar{\alpha} + 4mV}$$

де:

- $T$  – це час затухання у секундах для зменшення рівня на 60 дБ;
- $V$  – об'єм приміщення (в  $\text{м}^3$ );
- $\bar{\alpha}$  – коефіцієнт звукопоглинання;
- $S$  – Сумарна площа всіх поверхонь приміщення;
- $m$  – це коефіцієнт затухання повітря, який відповідає за ослаблення звуку під час його поширення через повітря.

Рекомендовані значення часу реверберації для різних приміщень наведено в таблиці 2.1.

Таблиця 2.1 – Рекомендовані значення часу реверберації [5]:

Приміщення	Об'єм	Час реверберації
Студія звукозапису	$< 50 \text{ м}^3$	0,3 с
Класна кімната	$< 200 \text{ м}^3$	0,4 - 0,6 с
Офіс	$< 1000 \text{ м}^3$	0,5 - 1,1 с
Лекційна зала	$< 5000 \text{ м}^3$	1,0 - 1,5 с
Концертний зал, опера	$< 20000 \text{ м}^3$	1,4 - 2,0 с

### 2.2.2 Структура системи підвищення розбірливості мови

Система підвищення розбірливості мови побудована з використанням нейронних мереж, як програмного інструменту для досягнення деревербації сигналів. Нейронна мережа побудована за принципом U-Net. U-Net — це архітектура згорткової нейронної мережі, яка, використовуючи зображення, автоматично виділяє важливі ознаки або патерни, а потім розширює їх до повної карти сегментації з високою точністю деталізації. Вона складається з енкодера, що поступово згортає дані, та декодера, який відновлює їхній просторовий розмір, зберігаючи всю цінну інформацію [76]. В нашому ж випадку, якщо записаний мовний сигнал перевести в частотну форму, то на

основі часово-частотного представлення маємо спектрограму, яку можна представити у формі зображення.

Структура нейронної мережі наведена на рисунку 2.4 та складається з двох основних частин: стискаюча (encoder) та розширююча частина (decoder).

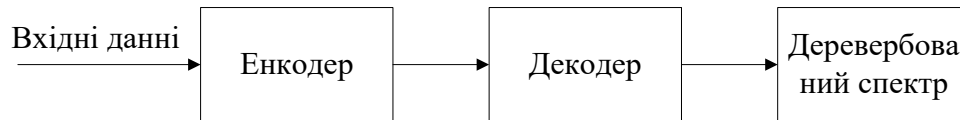


Рисунок 2.4 – Спрощена структурна схема нейронної мережі

Стиснення виконується за допомогою групи згорткових шарів із поступовим зменшенням розмірності вхідних даних, де кожен шар використовує функцію активації Leaky ReLU згідно формули (1.3), щоб зберегти деталі у низькочастотних компонентах.

Розширення відновлює просторові розміри даних, одночасно поєднуючи інформацію, витягнуту на етапі стискання. Цей етап забезпечує мережі можливість детально реконструювати початкове зображення та виділити важливі особливості. Після кожного рівня розширення розмір даних збільшується, і мережа поступово додає деталі, намагаючись відновити їх до вихідного розміру [76]. Розширення виконується за допомогою транспонованих згорткових шарів. В якості функції активації для шарів розширення використовується ReLU згідно формули (1.2).

Використання функції ReLU забезпечує чітке і стабільне відновлення позитивних ознак, що сприяє точнішій реконструкції та допомагає уникати артефактів, які можуть виникати при передачі від'ємних значень на етапі реконструкції.

Шари стиснення та розширення з'єднуються за допомогою так званих скіп-з'єднань. Скіп-з'єднання з'єднують відповідні шари стиснення та розширення, забезпечуючи перенесення інформації з ранніх шарів стиснення до відповідних шарів розширення, щоб відновити важливі деталі, втрачені під час

стискання. Навчена нейронна мережа за результатами своєї роботи дозволяє на виході отримати відновлені сигнали з спектральними ознаками дереверберації.

### **2.3 Теоретичні основи побудови систем клонування голосу**

Клонування голосу – задача навчитись синтезувати голос людини на основі комп'ютерних засобів через використання заздалегідь зроблених зразків записаної мови.

На відміну від існуючих технічних рішень, які було розглянуто в першому розділі роботи, наприклад в оглядовій статті [63] де відмічено, що система клонування мовних сигналів може бути створена як шляхом перетворення тексту в мовний сигнал (перший спосіб), так і на основі перетворення голосу з можливістю його наближення до певного еталонного варіанту (другий спосіб). В даному випадку в нашому дослідженні обрано саме перший спосіб, оскільки за другим способом наявна процедура модифікації форми звукової хвилі по голосу, який записано на оригінальному аудіо зразку. І цю процедуру звісно можна реалізувати на основі відомих алгоритмів оброблення сигналів, проте можуть бути втрачені важливі лінгвістичні та фонетичні характеристики мовлення.

Фактично, така процедурна спрямована на те, щоб при синтезі мовних сигналів зберегти в копії природність та індивідуальність оригінального звучання. Синтез мовлення базується на автоматичному перетворенні написаного довільного тексту в усне мовлення. При цьому, лінгвістично-акустичні характеристики обираються на основі записаних оригінальних мовних зразків. При синтезі мовлення букви чи група букв виражаються через фонеми. Далі відбувається перетворення тексту в еквівалентні слова (процедура токенизації). Після цього, кожне слово з відповідною фонетичною транскрипцією об'єднується у фонетичні групи, які далі є основою для створення мовних звуків [60].

Для клонування голосу та збереження лінгвістичних і фонетичних характеристик мовлення запропоновано використати нейронні мережі глибокого навчання. Структурна схема системи клонування голосу наведена на рисунку 2.5.

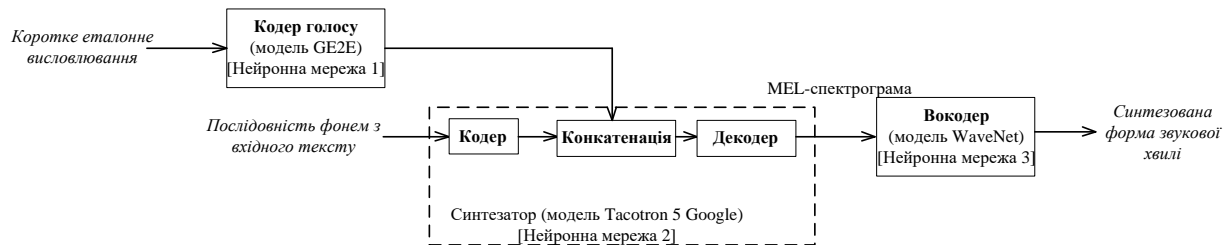


Рисунок 2.5 – Структурна схема системи клонування голосу

Система клонування голосу включає в себе три нейронні мережі, які проходять навчання незалежно одна від одної в послідовному порядку. У схемі, представлений на рисунку 2.5, реалізовано метод перетворення тексту на мовленнєве висловлювання із заданими акустичними характеристиками за допомогою технології SV2TTS [77]. Ця структурна схема є прикладом статистичної параметричної моделі синтезу мовних сигналів, яка демонструє чіткий зв'язок між текстовими ознаками на вході та синтезованими акустичними характеристиками на виході.

У кодер голосу на вхід подається короткий еталонний зразок мовлення, на основі якого формується репрезентація голосу для роботи синтезатора. При цьому кодер генерує одновимірний вектор із фіксованою розмірністю, відомий як  $d$ -вектор [68].

Синтезатор виконує ключову роль у процесі генерації Mel-спектрограм, які потім передаються вокодеру для відтворення кінцевого аудіосигналу. Синтезатор приймає на вхід текстовий опис (текст мовлення) та репрезентацію голосу  $d$ -вектор (генерований кодером) і створює Mel-спектрограму — проміжне представлення звукового сигналу. Ця Mel-спектрограма описує спектральні характеристики аудіо, зокрема, які частоти містяться в сигналі та їхню інтенсивність. Синтезатор використовує  $d$ -вектор для створення Mel-

спектрограми, що відповідають тембру, інтонації та стилю конкретного мовця. Mel-спектрограма, створена синтезатором, є лише проміжним результатом. Вона містить важливу акустичну інформацію, але не є повноцінним звуковим сигналом. Для отримання звуку потрібен вокодер.

Вокодер виконує головну роль у процесі відтворення кінцевого аудіосигналу з проміжного представлення — Mel-спектрограм, що генеруються синтезатором. Mel-спектрограма є лише спектральним представленням аудіосигналу, яке описує амплітуди частот у часі, а вокодер в свою чергу отримує Mel-спектрограму на вхід і відтворює неперервний аудіосигнал, що можна прослухати. Вокодер враховує часові та спектральні особливості Mel-спектрограм і відтворює мовлення з відповідними інтонаціями, тембром та гучністю.

### 3 ПРАКТИЧНА ЧАСТИНА ДОСЛІДЖЕННЯ

В рамках практичної частини дослідження розроблено три системи оброблення акустичної інформації, які дозволяють вирішити ряд ключових задач, а саме:

- забезпечення надійної перевірки та контролю доступу до приміщення особи за голосом (система ідентифікації за голосом);
- розроблення мовного автомату з розпізнавання слів (іменник, дієслово, прикметник) як українською так і англійською мовами з різною швидкістю та інтонацією проголошення слів (система розпізнавання мови);
- відновлення записаного мовного сигналу в приміщенні, яке неналежно акустично оброблено (система підвищення розбірливості мови з функцією дереверберації);
- формування синтезованого голосу для створення аудіо додатків, наприклад, для створення голосових асистентів чи для створення системи переведення текстів у голосові файли (система клонування голосу).

Зауважимо, що усі розроблені системи мають працювати з аудіо контентом, який створено або записано не лише англійською мовою, але й українською мовою, враховуючи місцевий діалект. Крім цього, необхідно забезпечити переваги розроблених систем у порівнянні з існуючими аналогами, насамперед з точки зору швидкодії навчання та роботи, точності та головне, розроблені системи мають вирізнятись властивістю адаптації роботи до зміни умов функціонування.

Процедура ідентифікації за голосом є першим важливим і ключовим етапом, який забезпечує можливість персоналізованої взаємодії з системою і де фактично перевіряються персоналізовані характеристики людини. Використовуючи нейронні мережі для аналізу унікальних акустичних характеристик голосу, цей модуль дозволяє точно визначати особу користувача.



Така персоналізація створює основу для наступного етапу — розпізнавання мови, де на базі очищених та сегментованих даних, система переводить мовлення у текстову форму. Це особливо важливо для забезпечення точного розуміння контексту та змісту мовлення, що, в свою чергу, часом потребує покращення якості сигналу в умовах реверберації. Деревверберація значно покращує якість сигналу, у випадку коли запис цього сигналу відбувався у приміщенні, яке не адаптовано за акустичними умовами. Якщо запис проводиться саме мовних сигналів, то реверберація може вплинути негативно і на точність розпізнавання мови. В рамках задачі створення системи ідентифікації за голосом також був проведений експеримент щодо перевірки цієї системи для випадку, коли мовний сигнал перевірки є штучно синтезованим голосом на основі комп'ютерних засобів, що не завжди є доступним, якісним та зручним засобом. Тому додатково, було вирішено реалізувати систему, яка навчена швидко робити якісний клон голосу, так щоб клон був схожий на оригінал краще, ніж при використанні комп'ютерних засобів.

Для ідентифікації за голосом існують такі рішення як Amazon's Alexa та Apple's Siri, але вони, на жаль, не доступні для української мови. Більш того система ідентифікації мовців, яка навчена безпосередньо на україномовних даних, матиме низку переваг порівняно з тією, що проходила тренування на англійських прикладах. Модель, навчена саме на українській вибірці, краще розрізняє типові звуки й поєднання української мови. Система, адаптована до англійської, орієнтується на інший набір акустичних патернів і погано інтерпретує характерні для української мови спектральні особливості.

Система розпізнавання мови широко представлені на ринку. Компанії такі як Microsoft, Google та Amazon пропонують рішення для розпізнавання мови, але не всі вони на жаль підтримують українську мову. Також присутня проблема в тому, що часто спотворені слова за рахунок акценту або швидкості вимови можуть розпізнаватись не коректно, і це додатково викликає інтерес до дослідження щодо перевірки поведінки точності розпізнавання.

Як зазначено в пункті 1.4, існує багато методів усунення реверберації на основі одно- та багатомікрофонних систем [23], таких як, інверсна фільтрація [22, 24], багатокрокові лінійні прогнози [25], автоматичні мовні кодери та декодери [17], а також обробка спектральної області мовного сигналу з використанням методу найменших квадратів [17]. Проте, основним недоліком цих підходів є те, що вони не повною мірою враховують спектральну структуру мовлення, через що, можуть втрачатися чіткі часово-частотні закономірності, характерні для мовного сигналу. Іншими словами, внаслідок використання зазначених підходів ці закономірності можуть бути просто втрачені. Задачею дереверберації мовлення є збереження глобальної часової та спектральної інформації сигналу [23], що можливо забезпечити на основі використання нейронних мереж [16]. Ідея використання нейронних мереж для оброблення записаних мовних сигналів виникла через те, що ці мережі успішно дозволяють розв'язувати різноманітні прикладні задачі пов'язані з обробленням зображень і в нашому випадку, цікаво перевірити як ці мережі можна використати саме для мовної інформації. В нашому ж випадку, якщо записаний мовний сигнал з реверберацією та шумом перевести в частотну форму, то на основі часово-частотного представлення маємо спектрограму, яку можна представити у формі візуального зображення. І задачею нейронної мережі вже є аналіз та корекція цього зображення. Спектрограма дозволяє показати типові паттерни (форманти, області клацання, висоту тону сигналу, тощо) а також характерні особливості, які привносить реверберація та адитивний шум [26].

Розв'язок задачі дереверберації з використанням нейронних мереж показано в роботі [21], проте в цій статті не вказані особливості системи та її будову, що унеможлиблює її практичне використання. Тому, для реалізації аналогічної системи, підхід який наведено, ми не можемо використати. В роботі [19] наведено приклад нейронної мережі для дереверберації, яка побудована на рекурентних нейронних мережах, але згорткові нейронні мережі які наведені в пункті 2.2 дисертаційної роботи, на відміну від рекурентних, можуть одночасно обробляти всі частини спектрограми, що дозволяє їм швидше та точніше

виділяти ключові закономірності, пов'язані з реверберацією. У той час як рекурентні моделюють послідовні залежності, вони обмежені у здатності фільтрувати просторові патерни, що можуть бути ключовими для визначення ревербераційних компонентів. Крім того, згорткові нейронні мережі виконують безпосереднє перетворення спектра, що дозволяє не просто приглушувати реверберацію, а точно відновлювати вихідний мовний сигнал, уникаючи надмірного придушення або втрати корисних ознак. У роботі [78] розглянуто глибоку нейронну мережу, яка обробляє кожен кадр спектрограми незалежно, і мережа CNN обробляє всі частини спектрограми одночасно, що дозволяє приглушувати ревербераційні компоненти та адаптивно зберігати структуру сигналу. Це дозволяє отримати більш природне звучання без втрати важливих фонетичних деталей. Крім того, CNN є більш стійкою до шумів, оскільки може ефективно розпізнавати загальні закономірності реверберації незалежно від конкретних акустичних умов.

Для клонування голосу на ринку існує достатньо рішень такі як Descript, Microsoft Azure Custom Neural Voice, ElevenLabs та інші. Наведені системи офіційно на жаль не працюють з українською мовою. Також для клонування голосу ці системи вимагають від 5 до 30 хвилин тривалості запису з голосу мовця, щоб просто донавчити систему на записаних семплах, а не намагаються відтворити голос “на льоту”, і це обмеження, а точніше її подолання також викликає інтерес для дослідження.

### **3.1 Система ідентифікації за голосом**

На відміну від наведених технічних рішень в статтях [79, 80], запропонована система ґрунтується на рекурентній нейронній мережі, а не на згортковій. Такий підхід дозволяє враховувати часові залежності, послідовність, глобальний контекст і дозволяє працювати зі змінною довжиною сигналів. Запропонований підхід в статті [81] також не здатен аналізувати

часові послідовності, та має такий самий недолік, як і у випадку використання згорткових нейронних мереж.

Для реалізації системи ідентифікації за голосом за основу взято рекурентну нейронну мережу, характеристики якої описано в п.2.1.3 роботи. При цьому, як було зазначено в архітектурі використано 8 шарів, з яких відразу 7 прихованих шарів, і чотири з цих прихованих шарів мають вбудовану внутрішню пам'ять, яка дозволяє аналізувати сусідні елементи словесної послідовності. Крім цього, необхідно відмітити що сам аудіо сигнал пройшов крізь процедуру фреймінгу, тобто неперервний сигнал поділяється на сегменти, або фрейми тривалістю 20 мс. Перевірку ж розробленої системи проведено за умови, що фрази будуть відтворені за різних умов – помилкова фраза, фраза з емоційним змістом, фразу створено штучними засобами (синтез). Лістинг програми розробленої системи ідентифікації за голосом з використанням нейронної мережі наведено у додатку А роботи.

### **3.1.1 Навчання системи ідентифікації за голосом**

На етапі навчання нейронної мережі, яка є складовою системи ідентифікації за голосом, задля забезпечення процедури регуляризації використаємо техніку дропауту та L2 регуляризацію, і приймемо, що кількість відключених для оновлення ваг нейронів складає 30% і L2 регуляризація має значення 0.001. Три повнозв'язних шари використовують функцію активації *relu* згідно формули (1.2), а останній шар використовує функцію активації *softmax* згідно формули (1.4). Тобто, визначено, що структура нейронної мережі для реалізації функції ідентифікації за голосом, відповідно до п.2.1.3, характеризується тим, що додано захист від перенавчання, а в самій архітектурі використано відразу дві функції активації.

Вибірка для навчання нейронної мережі представлено у форматі аудіофайлів з частотою дискретизації 22050 Гц які містять записи голосу. Аудіосигнали завантажуються з файлів та обробляються для подальшого

аналізу. Аудіосигнали перетворюються в набір MEL-кепстральних коефіцієнтів. Для обрахунку MEL-кепстральних коефіцієнтів використовувались наступні параметри:

- частота дискретизації – 22050 Гц;
- розмір вікна – 2048;
- крок між вікнами 512;
- кількість MEL-кепстральних коефіцієнтів – 20.

Для навчання нейронної мережі використовувався оптимізатор Адам та функція втрат з назвою “категоріальна кросентропія”.

Функція втрат — це математичний інструмент, який вимірює різницю між цільовими та прогнозованими значеннями, оцінюючи, наскільки точно нейронна мережа відтворює залежності у даних. Під час навчання нейронної мережі метою є мінімізувати цю втрату, що дозволяє мережі наближатися до правильних результатів. Оптимізатор відіграє ключову роль у цьому процесі: він оновлює параметри нейронної мережі (ваги та зміщення) відповідно до градієнта функції втрат, використовуючи алгоритм, як-от градієнтний спуск. Таким чином, функція втрат визначає, наскільки мережа відхиляється від цільового значення, а оптимізатор коригує параметри, щоб мінімізувати цю похибку. Існують такі функції втрат, як категоріальна крос ентропія, середня квадратична помилка, середня абсолютна похибка та інші.

Категоріальна кросентропія обраховується наступним чином (3.1) [82]:

$$E_{CC} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C p_{ic} \cdot \log(y_{ic}) \quad (3.1)$$

де  $N$  це кількість даних при навчанні,  $C$  – кількість класів,  $p_{ic}$  це значення яке показує чи належить  $i$  –й приклад до класу  $c$ ,  $y_{ic}$  це прогнозована вірогідність для  $i$  –го прикладу, що належить класу  $c$ .

Оптимізатори Адам є алгоритмом на основі методу модифікованого градієнтного спуску. Формула градієнтного спуску [83]:

$$\theta_{t+1} = \theta_t - a_t \Delta f(\theta_t)$$

де  $\theta_t$  це параметри моделі на ітерації  $t$ ,  $a_t$  це швидкість навчання на ітерації  $t$ ,  $\Delta f(\theta_t)$  градієнт функції втрат  $f(\theta_t)$  на ітерації  $t$ .

Окрім оптимізатора Адам є і інші похідні оптимізатори, такі як простий стохастичний градієнтний спуск, адаград та інші.

Для навчання нейронної мережі встановлено 50 епох. При тренуванні нейронної мережі 90% набору даних йшло на навчання, а 10% на перевірку роботи. Набір даних містить 90 семплів. З них 30 складаються зі слів “іменник”, “прикметник” та “дієслово” по 10 записів на кожен, які є еталонами, тобто належать користувачу по якому має відбуватись ідентифікація. Інші 60 записів містять слова “один”, “два”, “три” англійською по 10 записів на кожен та слова “іменник”, “прикметник”, “дієслово” також по 10 записів на кожен і належать іншим людям. Тривалість кожного семпла 2 секунди. Загалом семпли в датасеті містять наступні слова: “one”, “two”, “three”, “іменник”, “прикметник”, “дієслово”.

Значення втрат навченої моделі на повному обсязі навчальних даних, в які входять перевірочні дані становить 0.91 а точність 0.98. Значення втрат на перевірочних даних які не брали участь у навчанні становить 0.97 а точність 0.88 (рис.3.1).

```
Навчальні дані
3/3 [=====] - 2s 28ms/step - loss: 0.9158 - accuracy: 0.9889
Loss: 0.9158126711845398, Accuracy: 0.9888888597488403
Перевірочні дані
1/1 [=====] - 0s 105ms/step - loss: 0.9758 - accuracy: 0.8889
Loss: 0.9758468270301819, Accuracy: 0.8888888955116272
```

Рисунок 3.1 – Точність навченої моделі на навчальних даних

Для використання нейронної мережі використовується аудіофайл тривалістю в 2 секунди (валідаційні дані). З аудіофайлу відбувається обчислення мел-кепстральних коефіцієнтів аналогічно тому, як це робилось для навчання. Мел-кепстральні коефіцієнти подаються на вхід нейронної мережі, яка була попередньо навчена на подібних даних. Завдяки здатності розпізнавати індивідуальні особливості голосу, нейронна мережа аналізує ці дані і визначає, до якої саме особи належить представлений голосовий зразок.

### 3.1.2 Перевірка розробленої системи

Для проведення експерименту зроблена перевірка системи зі стандартною вимовою та з різними змінами (інтонаціями) вимови. Для перевірки системи були створені семпли з наступними відмінностями:

1. неправильний виголос фрази;
2. емоційний стан (імітовано вимову фразу під час стресу);
3. складне акустичне середовище (накладений білий шум);
4. природні зміни голосу (зміна тембру голосу);
5. штучно синтезований мовний сигнал.

Перевірка системи ідентифікації здійснювалась через запис власного голосу, який в подальшому буде визначатись для системи в якості еталонного. В таблиці 3.1 зазначена вірогідність приналежності голосу в семплі до мого власного голосу який використовувався при навчанні системи. З отриманих даних в таблиці 3.1 слід відмітити низьку вірогідність схожості еталонного голосу з штучно синтезованим. Тобто, в цьому випадку система заблокує вірну відповідь при створенні комп'ютерного запису. До того ж, якщо людина при ідентифікації є хворою на застуду, то це теж може бути причиною відмови з боку системи ідентифікації. Для всіх інших випадках з 100 спроб приблизно 85-88 будуть такими, які характеризують правильність спрацювання системи ідентифікації.

Таблиця 3.1 – Вірогідність приналежності до еталонного зразка голосу

Відмінності	Неправильний виголос фрази	Емоційний стан	Складне акустичне середовище	Природні зміни голосу	Стандартна вимова	Штучно синтезований запис
Вірогідність схожості з еталонним голосом	0,865	0,857	0,8802	0,7541	0,8855	0,0112

З метою розширення даних експерименту для перевірки стійкості роботи системи, яка має спрацьовувати лише у випадку еталонного голосу, проведемо перевірку на її реакцію коли відтворюються сторонні абсолютно незнайомі її голоса під час навчання.

Голоси були взяті з набору даних ([84]) Edinburgh Datashare, яке являє собою репозиторій голосових сигналів з університету Единбурга. Так сформовано тестову вибірку з голосів 23 різних людей та перевірено з якою вірогідністю голоси осіб з нового набору даних схожі на еталонний. В таблиці 3.2 наведено результати експерименту.

Таблиця 3.2 – Результати експерименту

Особа	Вірогідність схожості з еталонним голосом
Людина 1	0,0144
Людина 2	0,0154
Людина 3	0,0128
Людина 4	0,8711
Людина 5	0,0119
Людина 6	0,0116
Людина 7	0,0175
Людина 8	0,0149
Людина 9	0,0363
Людина 10	0,0122
Людина 11	0,0121
Людина 12	0,7454
Людина 13	0,0209
Людина 14	0,0344
Людина 15	0,0437



Продовження таблиці 3.2 – Результати експерименту

Особа	Вірогідність схожості з еталонним голосом
Людина 16	0,0183
Людина 17	0,0123
Людина 18	0,8428
Людина 19	0,0134
Людина 20	0,0173
Людина 21	0,0223
Людина 22	0,0296
Людина 23	0,0236

В цілому, за виключенням зразків записів під номерами 4, 12, та 18, система ідентифікації за голосом показує високу точність спрацювання на схожість (вірогідність схожості чужого голосу з вибірки з еталонним не більше 0,04). Для записів під номерами 4, 12, та 18 виявлено, що модель нейронної мережі показала хибний результат стосовно вірогідності схожості з еталонним голосом. Для покращення моделі необхідно зібрати більшу вибірку навчальних даних для нейронної мережі.

### 3.2 Система розпізнавання мови

У статті [85] представлено систему розпізнавання голосу, яка базується на штучній нейронній мережі, що обробляє вхідний сигнал як статичний набір ознак, не враховуючи при цьому його часову структуру. Такий підхід може бути ефективним у випадку чітких, стандартизованих зразків мовлення, але в реальних умовах, голосові сигнали мають варіації в інтонації та швидкості вимови. Натомість, запропонована модель в дисертаційній роботі, заснована на рекурентній нейронній мережі (RNN), та забезпечує краще врахування часових зв'язків у мовленні, що дозволяє точніше розпізнавати слова навіть при зміні

темпу мовлення чи незначних коливаннях у вимові. Це особливо важливо, оскільки, вхідний сигнал є динамічним процесом, а не статичним набором характеристик, і його коректне оброблення вимагає збереження контексту між послідовними акустичними ознаками. Завдяки цьому, підхід на основі RNN є більш гнучким і надійним, оскільки адаптується до природних особливостей мовлення та забезпечує стабільніші результати розпізнавання в умовах реального використання.

Подібна проблема спостерігається і в підході, запропонованому у статті [86], де використовується глибока нейронна мережа (DNN) без рекурентних компонентів. Основним методом покращення точності є алгоритм постеріорної ймовірності, який дозволяє оптимізувати вихідні оцінки моделі, мінімізуючи помилки класифікації. Однак, попри цю оптимізацію, відсутність механізму пам'яті та неврахування послідовності вхідного сигналу може призвести у підсумку до втрати важливої інформації при роботі з довгими фразами або змінами тембру мовлення. Це знову ж таки свідчить про те, що для коректного розпізнавання мовлення критично важливо зберігати часовий контекст, що є ключовою перевагою рекурентних нейронних мереж. Завдяки їхній здатності обробляти послідовні дані, RNN значно ефективніше адаптуються до природних змін мовлення, що робить їх кращим вибором для завдань розпізнавання голосу у складних акустичних умовах.

Для створення системи розпізнавання мовних сигналів використаємо ту ж саму структуру нейронної мережі рекурентного типу, яка була визначена в п.3.1 роботи (додаток А). Але відмінність буде полягати у тому, що дана система та її навчання і функціонування буде використано для задачі розпізнавання мовних сигналів, які виголошуються за різних умов і різними мовами – українською та англійською. Крім цього, будуть використовуватись різні оптимізатори, а також буде перевірена ефективність системи, коли кількість прихованих шарів в архітектурі нейронної мережі може змінюватись.

### 3.2.1 Навчання системи розпізнавання мови

При навчанні нейронної мережі, яка є основою системи розпізнавання мови, для регуляризації використовується техніка дропауту, і кількість відключених для оновлення ваг нейронів складає 30%. Три повнозв'язних шари використовують функцію активації *relu* згідно формули (1.2), а останній шар використовує функцію активації *softmax* згідно формули (1.4).

Дані для навчання представлені у вигляді аудіофайлів з тривалістю 2 секунди кожен. Аудіосигнали завантажуються і конвертуються у набір ознак за допомогою коефіцієнтів MFCC. Навчальні дані представлені у форматі аудіофайлів з частотою дискретизації 22050 Гц, які містять записи голосу. Для обрахунку MEL-кепстральних коефіцієнтів використовувались такі самі параметри як і в п.3.1.1.

Для навчання нейронної мережі використовується функція втрат категоріальна кросентропія (4).

При навчанні системи 90% тренувального набору даних йшло на навчання, а 10% на перевірку роботи. Для навчання системи сформовано 3 набори даних:

1. Складається зі слів “один”, “два” та “три” англійською по 10 записів на кожен.
2. Складається зі слів “іменник”, “дієслово”, “прикметник” по 10 записів на кожен.
3. Складається з записів слів “іменник”, “дієслово”, “прикметник” з доданими записами де слова вимовлялись швидше, повільніше та з акцентом по 16 записів на кожне слово.

Перший набір даних використовувався для навчання системи розпізнавати англійські слова, другий і третій використовувався для навчання системи розпізнавати саме українські слова.

З аудіофайлу відбувається обчислення мел-кепстральних коефіцієнтів аналогічно тому як це робилось для навчання. Мел-кепстральні коефіцієнти

подаються на вхід нейронної мережі. На виході система розпізнавання визначає яке слово було сказане в семплі.

### 3.2.1 Практична перевірка системи розпізнавання мови

Для експерименту було навчено певну кількість нейронних мереж. Нейронні мережі які навчались мали різну кількість прихованих шарів, і використовували при навчанні різні оптимізатори (adam, sgd, adagrad), також використовувались різні набори даних при навчанні.

Для навчання системи розпізнавати англійські слова використовувався 1 набір даних, а для розпізнавання українських слів використовувався 2 та 3 набір даних. Для більш точного порівняння експерименти ставились при однаковій початковій ініціалізації вагів. Також ставились експерименти і при різній початковій ініціалізації вагів. Ваги можуть ініціалізуватись так, що відразу опиняться в локальному мінімумі що може ускладнити роботу оптимізатору, а в іншому експерименті навпаки буде ситуація з більш сприятливою ініціалізацією. Для кожного експерименту кількість епох для навчання обрана на рівні 10000.

В таблиці 3.3 наведено результати навчання систем для розпізнавання англійських слів. Для цього використовувався 1 набір даних.

Таблиця 3.3 – Результати навчання систем для розпізнавання англійських слів

Кількість прихованих шарів	Ініціалізація вагів	Оптимізатор	Епоха (цикл), на якій досягалось мінімальне значення функції втрат	Значення функції втрат на навчальному наборі даних	Точність на навчальному наборі даних	Значення функції втрат на тестовому (валідаційному) наборі даних	Точність на тестовому наборі даних
7	Випадкова	adam	25	0.0324	100%	0.0606	100%

Продовження таблиці 3.3 – Результати навчання систем для розпізнавання англійських слів

Кількість прихованих шарів	Ініціалізація вагів	Оптимізатор	Епоха (цикл), на якій досягалось мінімальне значення функції втрат	Значення функції втрат на навчальному наборі даних	Точність на навчальному наборі даних	Значення функції втрат на тестовому (валідаційному) наборі даних	Точність на тестовому наборі даних
7	Не випадкова	adam	5145	$3.53 \cdot 10^{-8}$	100%	0.002	100%
7	Випадкова	sgd	9994	$1.52 \cdot 10^{-4}$	100%	0.0656	100%
7	Не випадкова	sgd	9970	$1.56 \cdot 10^{-4}$	100%	0.0311	100%
7	Випадкова	Adagrad	9998	0.0015	100%	0.0672	100%
7	Не випадкова	Adagrad	9994	0.0012	100%	0.0444	100%

Отримані результати перевірки навчання системи розпізнавання англійської мови з таблиці 3.3 свідчать про те, що сама нейронна мережа з 7 прихованими шарами, для якої при навчанні використано різні оптимізаційні моделі, демонструє високу точність, але при цьому використовується різна кількість епох для отримання мінімального значення функції втрат, що є показником того, що мережа навчана. Так, при перевірці на основі тестового набору даних, мінімальне значення функції втрат для різних оптимізаційних моделей дорівнює не більше порогового значення 0,067. Таке значення підтверджує, що нейронна мережа якісно навчана на основі даних з набору англійських слів. При навчанні кращим оптимізатором серед трьох обраних за критерієм швидкодії та мінімальним значенням функції втрат, виявився оптимізатор adam. Далі перевіримо, як нейронна мережа навчається, коли набір навчальних даних складається з українських слів.

В таблиці 3.4 наведено результати навчання систем для розпізнавання українських слів. Системи з трьома різними видами архітектур за кількістю

прихованих шарів навчались на другому наборі даних. З аналізу даних таблиці 3.4 випливає, що при навчанні кількість прихованих шарів, їх збільшення або зменшення не суттєво впливає на абсолютне мінімальне значення функції втрат. Але при цьому, при досягненні мінімального значення функції втрат за умови збільшення прихованих шарів збільшується і кількість циклів, крізь які мережа проходить при навчанні на обраному наборі даних. Якщо порівняти ефективність навчання мережі у випадку 7 прихованих шарів (дані таблиці 3.3 та 3.4) при випадковій ініціалізації вагів і одному і тому самому оптимізаторі adam, то кращі результати навчання, за практично однакової кількості епох, при досягненні мінімального значення функції втрат, мережа показує саме при використанні українських слів. В даному випадку це значення дорівнює 0,0066, тоді як для англійського набору даних, значення функції дорівнює вже 0,0324.

Таблиця 3.4 – Результати навчання систем для розпізнавання українських слів

Кількість прихованих шарів	Ініціалізація вагів	Оптимізатор	Епоха на якій досягалось мінімальне значення	Значення функції втрат на навчальному наборі даних	Точність на навчальному наборі даних	Значення функції втрат на тестовому наборі даних	Точність на тестовому наборі даних
7	Випадкова	adam	29	0.0066	100%	0.0695	100%
7	Не випадкова	adam	35	$6.2 \cdot 10^{-4}$	100%	0.061	100%
7	Випадкова	sgd	9981	$1.59 \cdot 10^{-4}$	100%	0.2472	100%
7	Не випадкова	sgd	9811	$1.62 \cdot 10^{-4}$	100%	0.0049	100%
7	Випадкова	Adagrad	9976	0.0013	100%	0.0248	100%

Продовження таблиці 3.4 – Результати навчання систем для розпізнавання українських слів

Кількість прихованих шарів	Ініціалізація вагів	Оптимізатор	Епоха на якій досягалось мінімальне значення	Значення функції втрат на навчальному наборі даних	Точність на навчальному наборі даних	Значення функції втрат на тестовому наборі даних	Точність на тестовому наборі даних
7	Не випадкова	Adagrad	9974	0.0013	100%	0.0191	100%
9	Не випадкова	adam	52	$1.22 \cdot 10^{-5}$	100%	$1.6 \cdot 10^{-5}$	100%
9	Не випадкова	sgd	9960	$1.14 \cdot 10^{-5}$	100%	0.0277	100%
5	Не випадкова	adam	4016	$1.32 \cdot 10^{-7}$	100%	$8.33 \cdot 10^{-4}$	100%
5	Не випадкова	sgd	9750	$1.69 \cdot 10^{-4}$	100%	0.0024	100%

Для нейронної мережі з 9 шарами та обраним оптимізатором adam вдалось знайти найкраще за мінімумом значення функції втрат на тестовому наборі даних, і також для цього прикладу різниця функції втрат для тестового набору даних та для тренувального набору є найменшою. Також оптимізатор adam за даними таблиці 3.4, швидше знаходить оптимальне рішення ніж інші обрані два оптимізатори.

Додатково проведено експеримент, де було додано нові аудіозаписи в яких, слова вимовлялись швидше, повільніше і з акцентом (2 навчальний набір з українських слів). При цьому на етапі навчання ці видозмінені записи не використовувались, щоб мережа не змогла б адаптуватись до них. Експерименти з додатковими аудіозаписами проведено не з випадковою початковою ініціалізацією вагів.

В таблиці 3.5 представлено результати роботи навчання систем для розпізнавання українських слів, які навчались на 2 наборі даних, і які не містили семпли з видозміненою вимовою українських слів.

Таблиця 3.5 – Результати навчання мережі без використання семплів з видозмінено вимовою українських слів

Кількість прихованих шарів	Оптимізатор	Значення функції втрат	Точність
5	adam	2.45	77%
7	adam	1.15	81%
9	adam	2.81	72%

З даних таблиці 3.5. виходить, що всі нейронні мережі з різною кількістю прихованих шарів, не змогли з максимальною точністю розпізнати нові семпли з видозміненою вимовою.

В таблиці 3.6 наведено результати навчання систем для розпізнавання українських слів, які навчались на 3 наборі даних, і цей набір містить зразки семплів з видозміненою вимовою слів. При цьому, при навчанні був використаний лише один оптимізатор adam, який виявився найкращим у попередніх етапах експерименту.

Таблиця 3.6 – Результати навчання систем для розпізнавання українських слів

Кількість прихованих шарів	Регуляризація	Оптимізатор	Епоха на якій досягалось мінімальне значення	Значення функції втрат на навчальному наборі даних	Точність на навчальному наборі даних	Значення функції втрат на тестовому наборі даних	Точність на тестовому наборі даних
5	Ні	adam	6	0.198	100%	0.7525	80%
5	Так	adam	6	0.028	100%	0.0274	100%
7	Ні	adam	15	0.0079	100%	0.0607	100%
9	Ні	adam	7	0.439	100%	0.8849	60%
9	Так	adam	5892	0.0083	100%	0.0073	100%

При деяких експериментах нейронна мережа не змогла розпізнати всі слова в тестовому наборі даних. Це сталося через фіксацію факту перенавчання, тобто мережа завчила дані, які призначені для навчання і не змогла сформувати



загальні патерни, по яким змогла б розпізнати слова в семплах призначених для тесту. Для протидії перенавчанню була додана при експерименті регуляризація.

З даних таблиці 3.6 можна відмітити, що нейронна мережа з 9 прихованими шарами та включеною регуляризацією, змогла знайти найкраще значення функції втрат на тестовому наборі даних, і також для цього прикладу різниця функції втрат для тестового набору даних та для тренувального виявилась найменшою.

### **3.3 Система покращення розбірливості мови**

Далі розглянемо та власне перевіримо можливості розробленої нейронної мережі, теоретичний опис якої наведено у п.2.2.2 роботи, а програмний код реалізація цієї системи з застосуванням нейронної мережі розміщено у додатку Б роботи. При цьому основною задачею розробленої архітектури нейронної мережі буде якраз реалізації дереверберації аудіо сигналів. Тобто, за умовами практичного експерименту взято побудовані імпульсні характеристики двох навчальних лабораторій 209 та 438 12 корпусу факультету електроніки. Методика отримання цих характеристики описано в роботі [87]. Зазначимо, що в якості тестового сигналу обрано 16 записаних послідовних сплесків (тестовий сигнал *mls* з частотою дискретизації 44,1 кГц), які ретранслявались через активну акустичну систему Genius SP-HF 2.0 (джерело сигналу), а приймачами сигналу в цих навчальних аудиторіях обрано вимірювальні конденсаторні мікрофони Superlux ECM-999 (20 Гц - 20 кГц, динамічний діапазон 106 дБ, сигнал-шум 70 дБ) які у свою чергу були підключені до звукової карти Steinberg UR242.

Крім цього, зазначимо, що сам мікрофон в аудиторії 209 розміщено на відстані 7 м від джерела, тоді як в аудиторії 438 ця відстань дорівнює 5 м. Об'єм аудиторії 209 складає 270 м<sup>3</sup>, аудиторії 438 – 170 м<sup>3</sup>. При записі імпульсної характеристики приміщення самі аудиторії були не заповнено студентами.

Сама процедура деревереберації записаних сигналів полягала у тому, що нейронну мережу попередньо необхідно навчити, далі для моделювання реверберації на мовному сигналі використовується імпульсна характеристика аудиторії [87]. Після чого імпульсна характеристика кімнати об'єднується з семплом шляхом згортки щоб отримати ревербований сигнал. І цей сигнал вже подається на вхід нейронної мережі, структурна схема якої показана на рисунку 2.4. Слід зазначити, що згортка накладається після 1.5 с, навіть якщо тривалість імпульсної характеристики становить 1.5 с, через ефект "хвоста" імпульсної характеристики. І навіть, якщо основний сигнал починається лише після 1.5 с, згортка враховує, що імпульсна характеристика накладається на всі точки сигналу, включаючи ті, що знаходяться впродовж її "хвоста".

Додатково, для перевірки можливостей створеної нейронної мережі був проведений експеримент з запису сигналу в побутовій кімнаті, план якої показано на рисунку 3.2.

Враховуючи, що отримання імпульсної характеристики навчальних аудиторій та кімнати проходило за умови відсутності нейронної мережі, то саму нейронну мережу перед перевіркою необхідно навчити, і навчальні дані мають бути підібрано за умови, що ревербераційний ефект при отриманні імпульсних характеристик скоріш за все буде різним, враховуючи розміри аудиторій та кімнати, та відповідну акустичну підготовку цих приміщень, а точніше її відсутність.

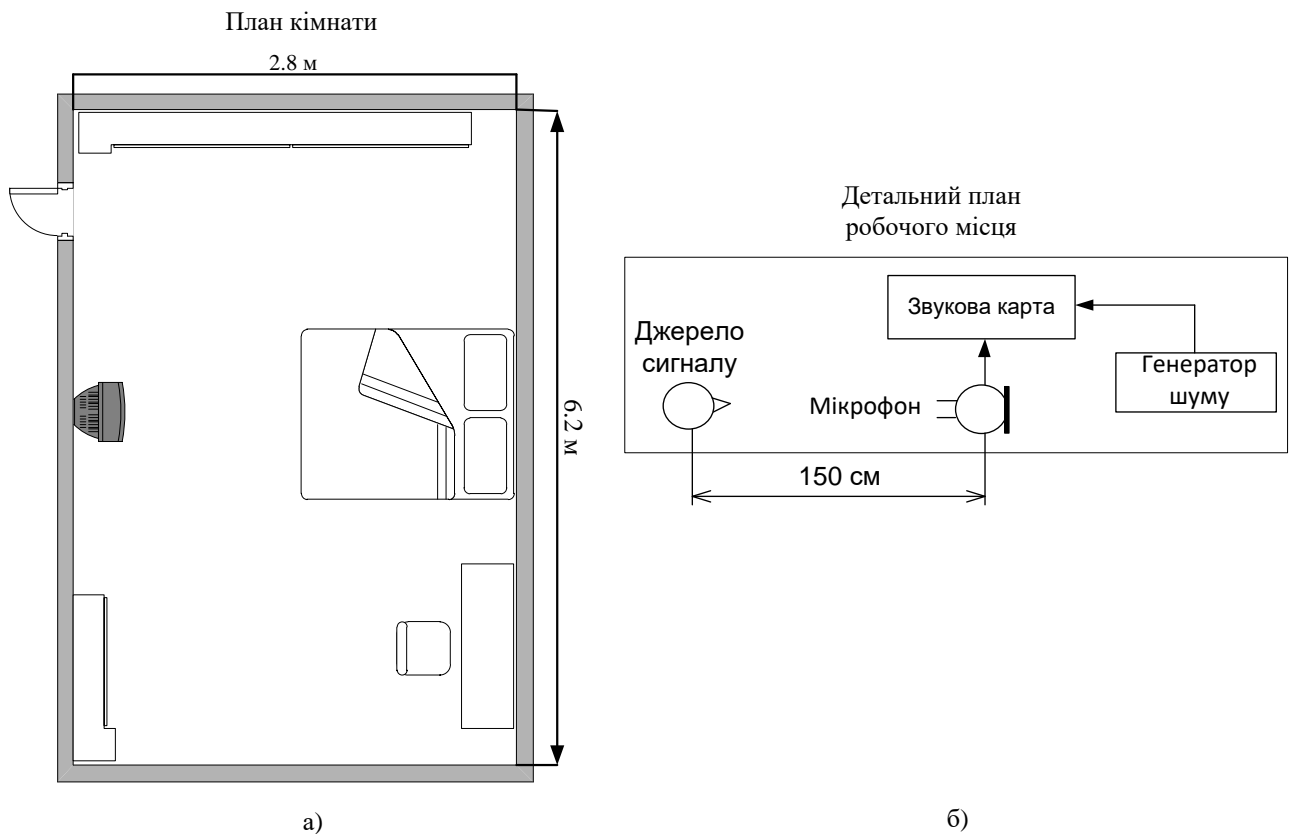


Рисунок 3.2 – План кімнати (а) та схема фіксації мовного сигналу (б)

### 3.3.1 Етап навчання нейронної мережі для системи покращення розбірливості мови

Для навчання нейронної мережі були використані набори звукових даних з Edinburgh Datashare (публічне електронне сховище даних Університету Единбурга). В якості даних на яких мережа навчалася деревербації використовувався датасет [88], а в якості перевіірочних даних використовувався датасет [84]. Лістинг програми розробленої системи покращення розбірливості мови з використанням нейронної мережі наведено у додатку Б роботи.

Набори даних було розроблено для навчання та тестування методів покращення мови, які працюють на частоті 44,1 кГц. Кількість семплів в обох датасетах налічує більше 10000 аудіо різної тривалості.

Для навчання було взято перші 10000 семплів з кожного датасету. Для збільшення варіативності деревербації в навчальному наборі даних [88] була

зроблена аугментація даних. Ця процедура використовується для збільшення кількості тренувальних даних та покращення здатності моделі до узагальнення, шляхом введення варіативності в тренувальні зразки. Набір даних було збільшено на 1572 семпла. Для цього було взято 1572 семпла з перевірного набору даних і накладена на них реверберація.

Реверберація додавалась випадковим чином з наступними характеристиками:

- час затримки між оригінальним сигналом та початком реверберації в діапазоні 0.15 - 0.25 с;
- швидкість затлумлення на 60 дБ в діапазоні 0.2 - 0.5 с
- співвідношення між "мокрим" (обробленим) та "сухим" (необробленим) сигналом в межах 0.3 - 0.45.

Відповідно до розмірів входних ознак для мережі, семпли розбиваються на фрагменти тривалістю 20,7 мс з перекриттям 50%.

Враховуючи, що наявність занадто великої кількості безмовних сегментів може негативно вплинути на навчання нейронної мережі, то були видалені безмовні сегменти, щоб виключити їх з навчання мережі. Кожен фрагмент аналізується на наявність ділянок з низькою енергією. Якщо тиша займає більше ніж 50% тривалості сегмента, цей сегмент відкидається і не використовується у подальшому для навчання.

Далі було обчислено спектральні ознаки за допомогою короткочасного перетворення Фур'є STFT. В процесі попередньої обробки даних спектральні ознаки (коефіцієнти, які отримані з короткочасного перетворення) зазнають наступних перетворень:

- логарифмічне представлення амплітуди (натуральний логарифм);
- нормалізація даних в діапазоні -1 та 1;
- перетворення до 4-вимірному формату (вектор ознак).

Для навчання нейронної мережі були використані наступні параметри:

- оптимізаційна модель Adam (метод стохастичного градієнтного спуску);

- початкова швидкість навчання  $8 \cdot 10^{-4}$ ;
- кількість епох для навчання дорівнює 50;
- зменшення швидкості навчання на 0.1 за кожні 15 епох;
- функція втрат при навчанні – середньоквадратична помилка (MSE), точніше середнє значення всіх квадратів помилок (середньоквадратичне абсолютна різниця між передбачувальними (прогнозованими) та фактичними значеннями).

Навчання проводилось з використанням CPU і тривало 206 хвилин.

Вхідними даними для попередньо навченої нейронної мережі є спектральні ознаки які обраховуються за допомогою короткочасного перетворення Фур'є (STFT) ревербераційного аудіо. Під час використання системи покращення розбірливості мови вхідні дані розбиваються на сегменти та обраховуються їх спектральні ознаки на основі використання нейронної мережі. Їх слід попередньо обробити таким самим чином як описано в пункті 3.1. Далі обраховані значення на виході перетворення Фур'є переводять у логарифмічну форму з нормалізацією і подаються на нейронну мережу для деревербації. Мережа прогнозує логарифмічну величину STFT деревербованого вхідного сигналу.

Перш ніж застосовувати функцію зворотного короткочасного перетворення Фур'є (ISTFT), слід попередньо обробити спектральні ознаки наступним чином:

- перетворення до 3-вимірної форми;
- денормалізація даних;
- експоненціальне масштабування;
- об'єднання всіх фрагментів;

Далі застосовується функція ISTFT для відновлення деревербераційного мовного сигналу в часовій області, використовуючи прогнозовані значення спектральних ознак та фазу ревербераційного мовного сигналу.

### 3.3.2 Вихідні дані до експерименту

Для тестування нейронної мережі записувались тестові семпли мікрофоном USB BOYA BY-M100UA на відстані 150 см від джерела сигналу. Мікрофон має наступні характеристики:

- робочий частотний діапазон 50 Гц-18 кГц;
- частота дискретизації 44,1 кГц;
- чутливість -36 дБ;
- сигнал/шум: 78 дБ.

Для моделювання ревербації на мовному сигналі використовується імпульсна характеристика приміщення (аудиторії 209, 438 та побутова кімната з рис.3.2). Імпульсна характеристика кімнати об'єднується з семплом шляхом згортки щоб отримати реверберований сигнал.

### 3.3.3 Експериментальна перевірка 1 (209 навчальна аудиторія)

Враховуючи алгоритм, який описано в п.3.3 роботи для перевірки нейронної мережі та в цілому системи покращення розбірливості на рисунку 3.3 наведено імпульсну характеристику 209 навчальної лабораторії, яку отримано на основі наведеної методики в роботі [87]. Слід зазначити що імпульсна характеристика наведена лише для одного каналу – лівого, оскільки для правого вона буде така сама.

Після етапу навчання нейронної мережі на навчальній вибірці звукових даних з Edinburgh Datashare з доданими налаштуваннями змін реверберації перевіримо як ефективно система реалізовує дереверберацію записаного сигналу. Для цього використаємо аудіосемпл з фразою: “Believe me we have happy times in America this is Chico's Seth”. Тривалість сигналу складає 4.9 секунди.

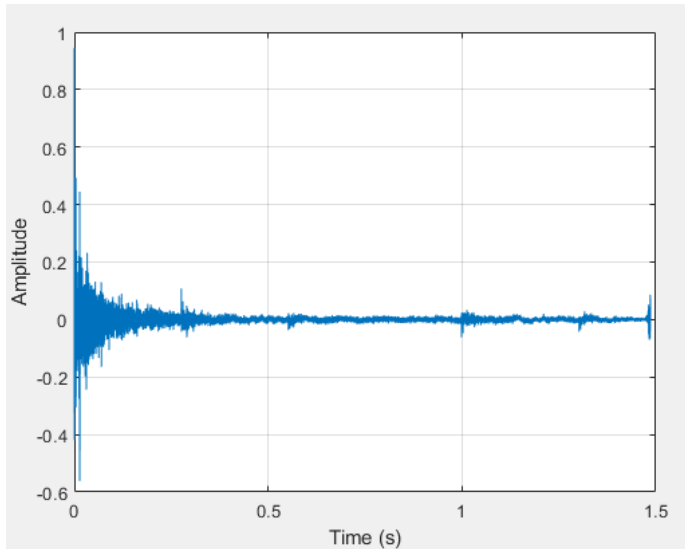
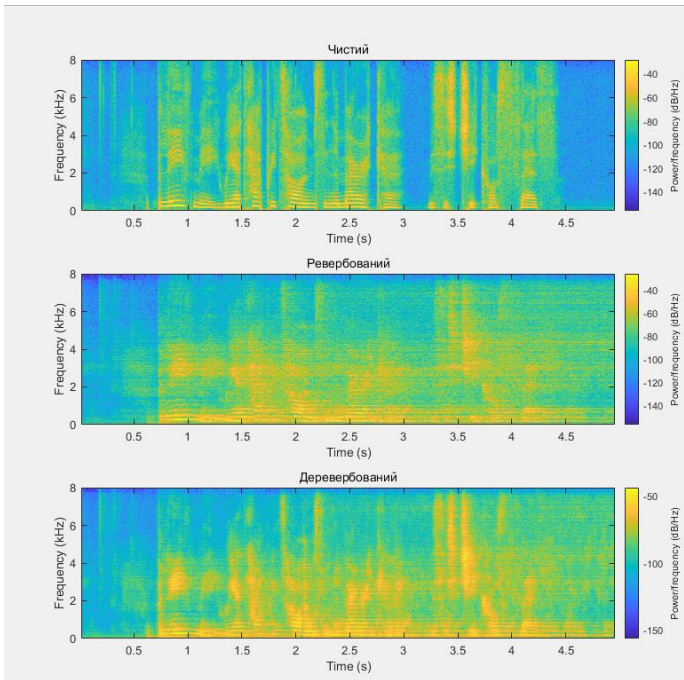
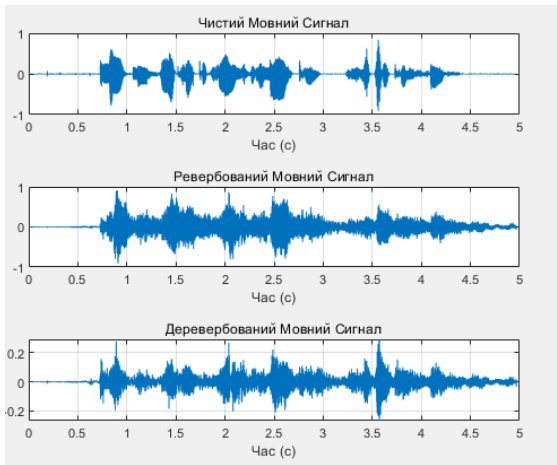


Рисунок 3.3 – Імпульсна характеристика ревербераційної кімнати 209

Спектрограми, які дозволяють оцінити ефективність системи покращення розбірливості для даного семплу наведено на рисунку 3.4 (а). На рисунку 3.4 (б) наведено сигналограми “чистого”, “ревербованого” та як результат роботи мережі – деревербованого сигналів. Дані графічні залежності отримано на основі створення лістингу програму в ПЗ Matlab (додаток Б).



а)



б)

Рисунок 3.4 – Спектрограми (а) та сигналограми (б) для тестового семплу англійською мовою

Аналізуючи отримані результати на рисунку 3.4 (а) можна відмітити, що результуюча спектрограма дереверберованого сигналу у порівнянні з реверберованим більш наближена за розподілом потужності до частоти в одиницях вимірювання дБ/Гц до спектрограми “чистого” семплу. Така сама тенденція прослідковується і якщо порівняти сигналограми на рисунку 3.4 (б). З іншого боку, враховуючи що нейронна мережа навчалась на англomовному датасеті, то перевіримо як її ефективність, коли запис тестового семплу зроблено українською мовою. Для цього текст в семплі наступний: “Перший тестовий сигнал”. Тривалість сигналу для цього випадку складає 4,4 сек. Отримані результати графічно наведено на рисунку 3.5 (а) – спектрограми та рисунку 3.5 (б) – сигналограми.

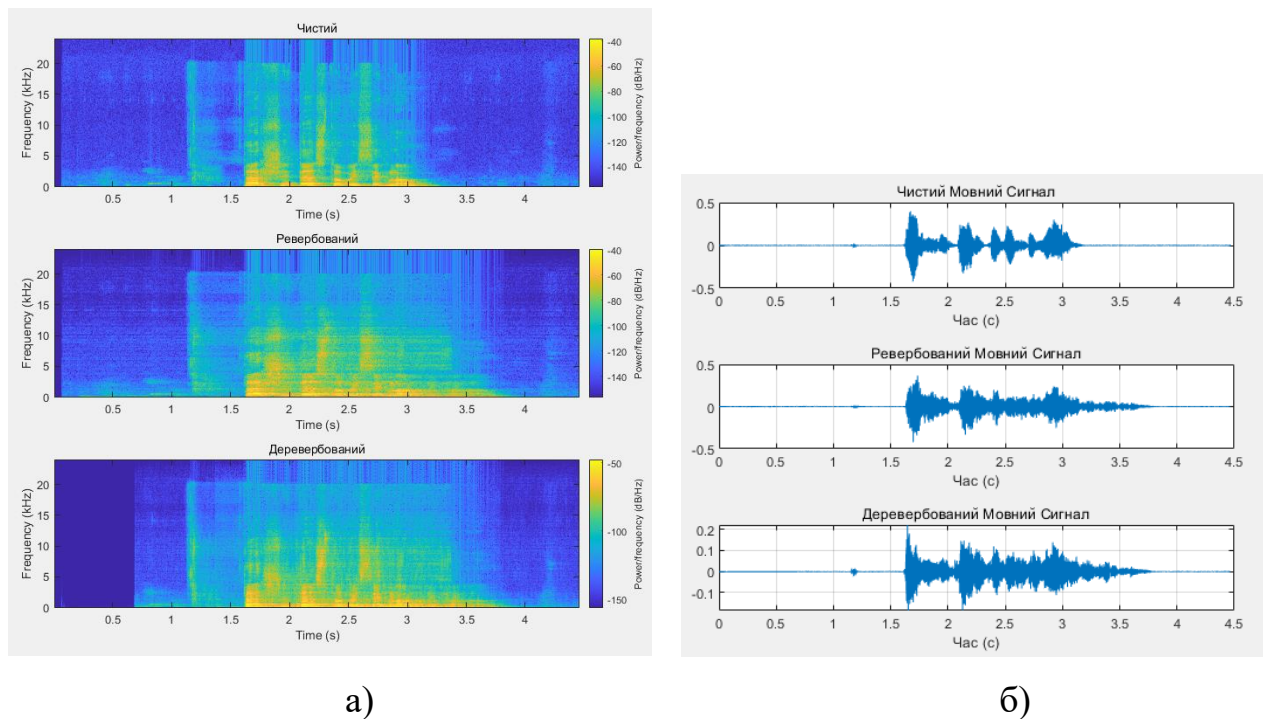


Рисунок 3.5 – Спектрограми (а) та сигналограми (б) для тестового семплу українською мовою

Для цього випадку, можна відмітити, що деревереберація показує кращі результати у порівнянні з картиною ревербераційного сигналу на частоті близько 4 кГц. Що стосується аналізу сигналограм, то тут результати краще з інтервалу 1,7 с, але можна відмітити при цьому що амплітуда сигналу (за



огиноючою) буде в результаті трохи збільшена внаслідок процедури дереверберації.

Для перевірки якості проведеної дереверберації сигналу було використано метод експертних оцінок, а саме була сформована група експертів, включаючи і мене з 10 людей:

- 6 людей чоловіки віком від 25 до 34 років без вад слуху;
- 1 людина жінка віком 22 роки без вад слуху;
- 1 людина жінка віком 50 років без вад слуху;
- 2 людини чоловіки віком від 50 до 55 років з помірними вадами слуху.

Експертам з групи було запропоновано дати оцінку якості ревербованого та деревербованого сигналів для двох мов за шкалою оцінок від 0 до 10. При цьому було запропоновано враховувати наступні критерії при оцінюванні: розбірливість слів, точність вимовлення окремих слів (тобто, чи не налязять слова один на одне), чи достатньо гучність звучання слів у фразі.

Результати опитування наведено в таблиці 3.7. Ці результати свідчать що в середньому якість сигналу до деревербації становить 4.3 для англійської мови та 4.1 для української, а після деревербації 7.4 та 6.4 відповідно. Тобто, на основі прослуховування мовних сигналів групою експертів, можна зробити висновок, що запропонований алгоритм на основі використання нейронних мереж з навчанням, зменшує рівень шуму, пов'язаного з реверберацією.

В наступному етапі експерименту, з'ясуємо чи характерні ці закономірності нейронної мережі, коли використовується імпульсна характеристика іншого приміщення, в даному випадку для аудиторії 438, яка має інший об'єм. Для чистоти експерименту самі тестові сигнали як англійською так і українською мовами залишимо такими самими, які і для аудиторії 209.

Таблиця 3.7 – Результати опитування

Експерт	Якість ревербованого сигналу для англійської мови	Якість результуючого сигналу для англійської мови	Якість ревербованого сигналу для української мови	Якість результуючого сигналу для української мови
Людина 1	4	9	4	7
Людина 2	3	6	3	5
Людина 3	5	8	5	6
Людина 4	5	7	4	7
Людина 5	4	7	4	6
Людина 6	4	8	4	7
Людина 7	3	8	4	5
Людина 8	6	7	5	8
Людина 9	5	6	4	7
Людина 10	4	8	4	6

### 3.3.4 Експериментальна перевірка 2 (438 навчальна аудиторія)

На рисунку 3.6 наведено імпульсну характеристику 438 навчальної лабораторії, яку отримано на основі наведеної методики в роботі [87]. Тут так само при записі імпульсної характеристики використано тестовий сигнал у формі 16 записаних послідовних сплесків (тестовий сигнал *mls* з частотою дискретизації 44,1 кГц). Обладнання для запису, яке наведено у п.3.3 таке саме, як і в аудиторії 209 з єдиною відмінністю, що сам мікрофон в аудиторії 438 розташовано на відстані приблизно 5 м від джерела сигналу (активна акустична система, яка ретранслює записаний сигнал сплесків). Об'єм аудиторії 438 – 170 м<sup>3</sup>. Порівнюючи імпульсні характеристики для аудиторій 209 і 438

(рисунки 3.3 та 3.6) можна відмітити, що для аудиторії 209 є виразне раннє відбиття в області часового інтервалу приблизно 0,29 с (рис.3.3) та на початку фіксації характеристики – приблизно в часовому околі 0,03 сек та 0,05 сек. Подібні особливості також були відмічені і в роботі [87]. Що стосується імпульсної характеристики для навчальної аудиторії 438, то тут важко визначити якісь переважаючі сплески на інтервалі від 0 с до 0,5 сек. Час реверберації для обох аудиторій дорівнює приблизно 0,95 сек.

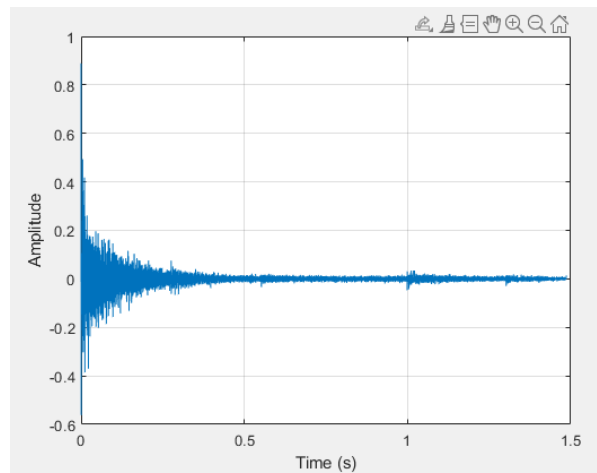


Рисунок 3.6 –Імпульсна характеристика ревербераційної кімнати 438

Як і для аудиторії 209, перевіримо ефективність функціонування розробленої нейронної мережі у випадку запису в навчальній аудиторії 438, причому в якості тестового семплу обрано такі самі сигнали, як і для аудиторії 209. Зазначимо, що аудиторія 438 12 навчального корпусу була досліджена на відповідність акустичним нормам у статті [89].

Спектрограми, які дозволяють оцінити ефективність системи покращення розбірливості для даного семплу англійською мовою наведено на рисунку 3.7 (а). На рисунку 3.7 (б) наведено сигналами “чистого”, “реверберованого” та як результат роботи мережі – дереверберованого сигналів.

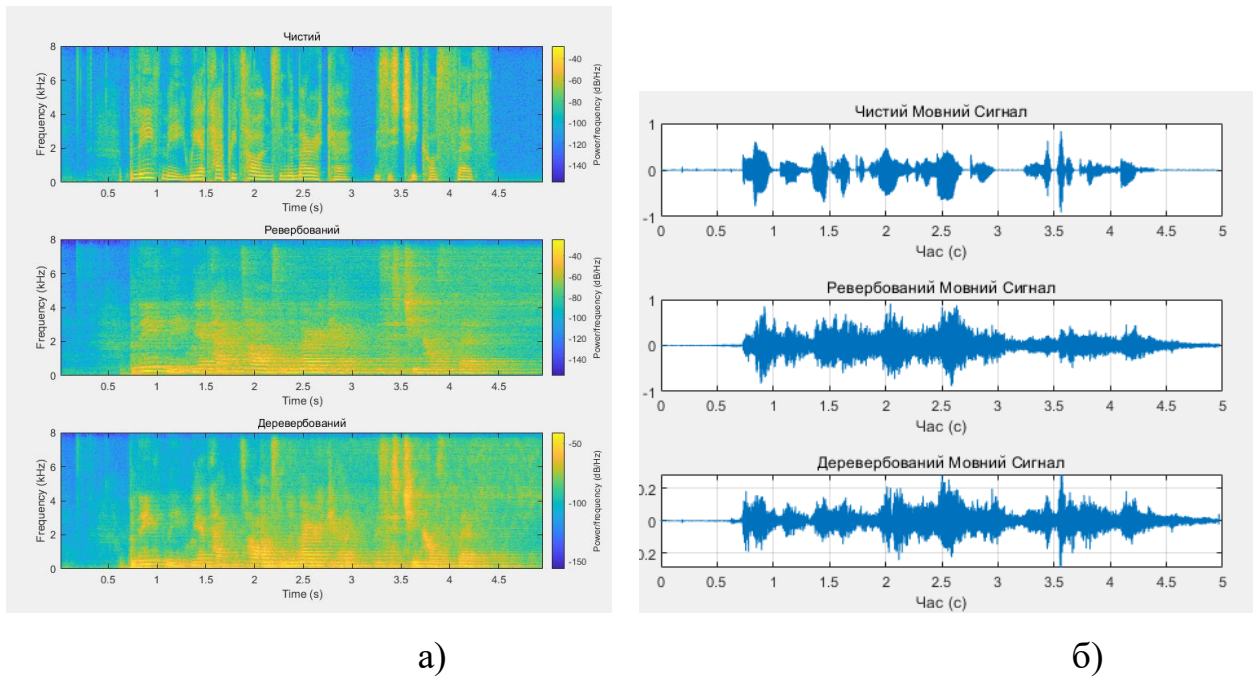
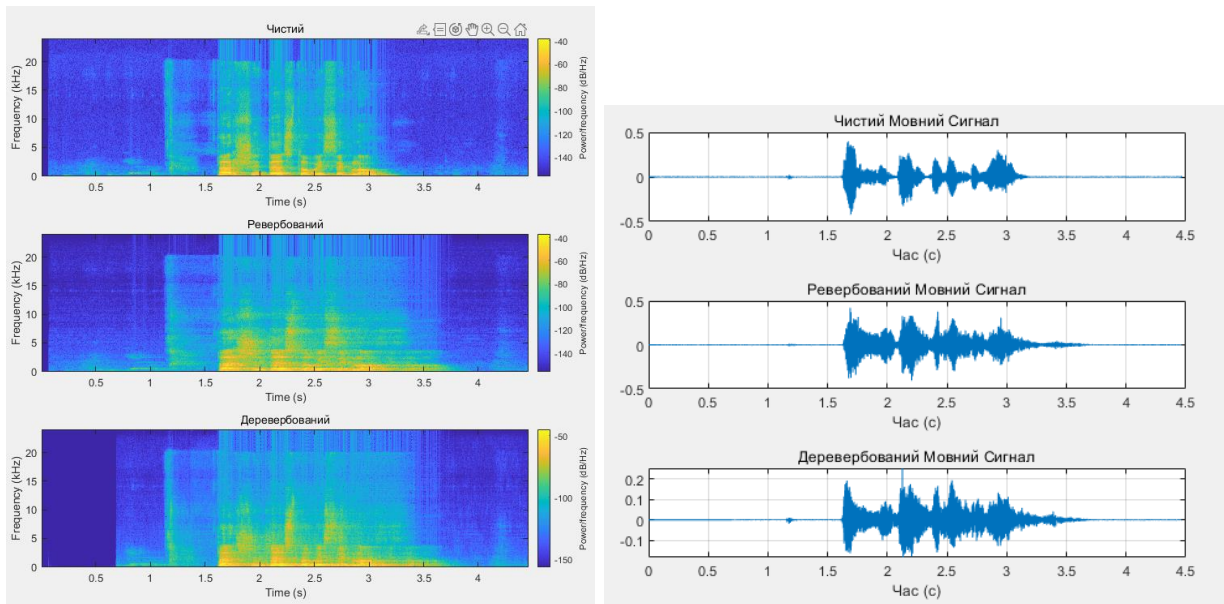


Рисунок 3.7 – Спектрограми (а) та сигналограми (б) для тестового семплу англійською мовою

Так само, як і для першого експерименту, перевіримо якість роботи нейронної мережі з дереверберації, коли записано тестовий семпл української мови. Для цього, обраний текст в семплі наступний: “Перший тестовий сигнал”. Тривалість сигналу для цього випадку складає 4,4 сек. Отримані результати графічно наведено на рисунку 3.8 (а) – спектрограми та рисунку 3.8 (б) – сигналограми.

Аналізуючи отримані результати на рисунку 3.7 можна відмітити, що нейронна мережа показує свою ефективність з дереверберації сигналу і особливо, це помітно на спектрограмі в області 2,2 сек та 3,5 сек (рис.3.7(а)). Найбільш це помітно саме для області 3,5 сек і підтвердження цьому можна отримати аналізуючи сигналограми на рис.3.7 (б).



а)

б)

Рисунок 3.8 – Спектрограми (а) та сигналограми (б) для тестового семплу українською мовою

Що стосується ревербованого мовного сигналу, то тут варто відмітити з аналізу сигналогами, що реверберація призводить і до збільшення енергетичної складової сигналу. З іншого боку, така сама тенденція, хоч і не так яскраво прослідковується і для результуючого сигналу. Особливо це помітно в діапазоні від 4,5 сек як на спектрограмі, так і на сигналограмі. Таку особливість, напевно можна пояснити особливостями самого алгоритму створення ревербованого сигналу та роботу спроектованої нейронної мережі.

Рисунок 3.8 (а) та (б) показує результати роботи нейронної мережі для випадку 438 аудиторії, коли тестовий семпл записано українською мовою. Відмітимо, що на початку роботи в інтервалі від 1,7 сек до 2 сек нейронна мережа показує свою ефективність, якщо аналізувати сигналогами з рис.3.8 (б). Разом з тим, якщо проаналізувати спектрограми то в діапазоні від 10 кГц помітно зменшення значення параметру за шкалою потужність/частота (дБ/Гц). Тобто, для використання запису, наприклад, музичних композицій в приміщенні, саму нейронну мережу необхідно донавчати. Крім цього, як і для випадку англійського семплу на інтервалі від 3,2 сек є певний “хвіст” за

нормованим значенням інтенсивності сигналу як ревербованого, так і результуючого. Для покращення цієї ситуації, напевно слід при навчанні нейронної мережі збільшити кількість циклів (епох) та додатково донавчити базою українських слів [90].

Також для перевірки якості результуючого сигналів як і в п.3.3.3 роботи, була використана група з тих самих експертів для забезпечення чистоти експерименту.

Аналогічно було запропоновано дати оцінку якості ревербованого та деревербованого сигналів для двох мов від 0 до 10. Результати опитування наведено в таблиці 3.9.

Таблиця 3.9 – Результати опитування

Експерт	Якість ревербованого сигналу для англійської мови	Якість результуючого сигналу для англійської мови	Якість ревербованого сигналу для української мови	Якість результуючого сигналу для української мови
Людина 1	4	8	5	7
Людина 2	5	7	4	6
Людина 3	3	6	3	6
Людина 4	5	7	5	7
Людина 5	6	8	6	6
Людина 6	4	7	4	6
Людина 7	5	7	5	7
Людина 8	3	8	3	7
Людина 9	4	7	4	5
Людина 10	5	8	5	6

Результати отриманих експертних оцінок свідчать, що в середньому якість сигналу до деревербації становить 4.4 для англійської мови та 4.4 для

української, а після деревербації 7.3 та 6.3 відповідно. Тобто і тут отримано підтвердження, що застосування нейронних мереж для вирішення задач деревербації сигналів є правильним кроком в рамках дослідження.

### 3.3.5 Експериментальна перевірка 3 (житлова кімната)

Для перевірки ефективності роботи розробленої нейронної мережі з деревербації сигналів, зробимо ще один експеримент, а саме перевіримо її роботу для житлового приміщення. Імпульсна характеристика кімнати наведена на рисунку 3.9 і тут на відміну від характеристик на рисунках 3,3 та 3,6 наявний сплеск не на початку фіксації в околі 0 сек, а в області 0,4-0,5 сек. При цьому тестові семпли, їх тривалість будуть такі самі як і в п.3.3.3 та п.3.3.4.

Отримані результати на рисунку 3.10 (а) показують, що в результаті створення реверберованого мовного сигналу та результуючого сигналу відбулось певне зміщення, як спектрограм так і сигналограм.

Це обумовлено тим, що імпульсна характеристика має зсув (пік основної енергії зсунутий від початку). Зсув на результуючому сигналі внаслідок математичної операції може бути різним і це залежить від різних факторів. По-перше, при операції згортки сигналу семплу з імпульсною характеристикою, може виникнути зсув у результуючому сигналі який визначається тим, де енергія семплу "накладається" на пік характеристики.

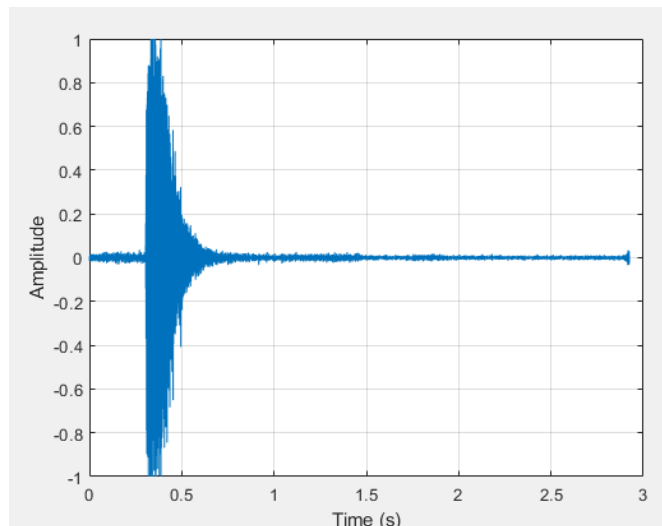


Рисунок 3.9 – Імпульсна характеристика житлової кімнати

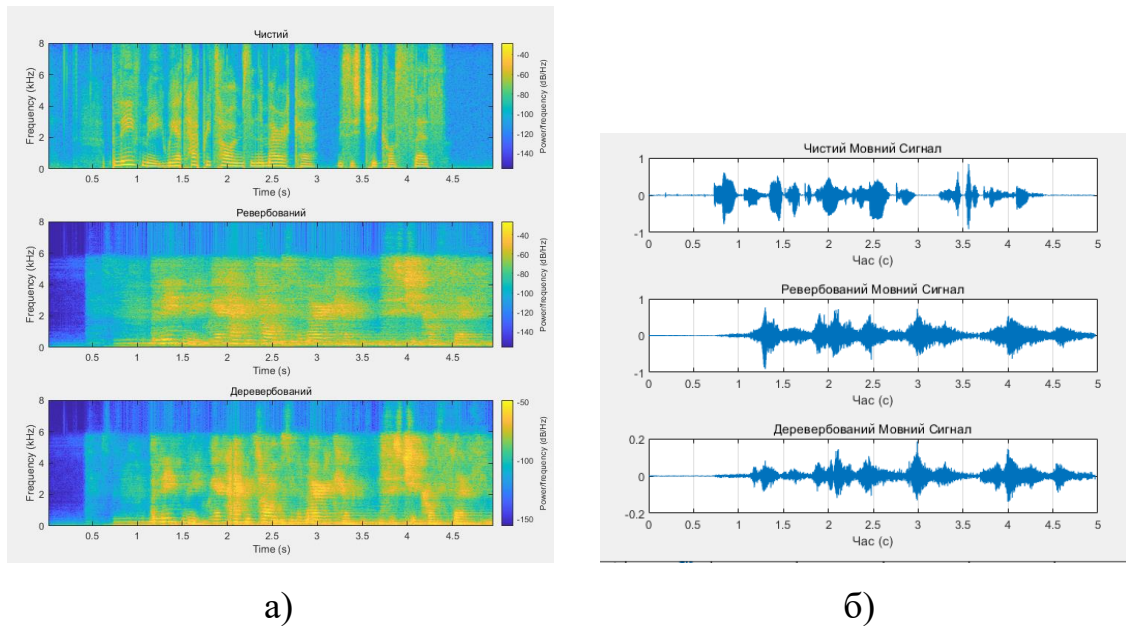


Рисунок 3.10 – Спектрограми (а) та сигналограми (б) для тестового семплу англійською мовою

Так, якщо пік основної енергії імпульсної характеристики знаходиться на певній затримці, то для кожного семплу ця затримка буде діяти по-різному залежно від того, де зосереджена енергія самого семплу. Наприклад, семпл із коротким "ударним" сигналом (імпульс) чітко взаємодіє з піком характеристики, і результуючий зсув буде майже рівним затримці цієї характеристики. По-друге, для створеної частотної характеристики кімнати, семпл має свій власний частотний спектр. І якщо семпл містить більше енергії в тих частотах, які сильно підсилюються чи заглушуються імпульсною характеристикою, то це змінює сприйняття "центру ваги" результуючого сигналу. В результаті маємо, що семпл із переважно високими частотами може створити зсув, який ближчий до піку імпульсної характеристики, оскільки високі частоти мають коротші часові затримки. Семпл із низькими частотами може "здаватися" більш зміщеним, бо низькі частоти зазвичай мають довші часові складові.

Тобто, аналіз графіків варто проводити, враховуючи що отримані спектрограми та сигналограми зміщено праворуч на  $\Delta_1=0,5$  сек. Крім цього, аналіз спектрограми показує, що нейронна мережа внаслідок введенного



обмеження для реверберованого мовного сигналу, а саме зниження рівню за частотою більше 6 кГц, відображає приблизно таку саму тенденцію для цього діапазону частот. Тобто, тут вірогідно за все варто змінити параметри до створенні варіацій ефекту реверберації. Додатково, порівнюючи результат дереверберації з “чистим” семплом враховуючи зміщення за часом на рівні  $\Delta_1=0,5$  сек праворуч, за сигналами як і у попередніх експериментах, що нейронна мережа трохи підсилює за рівнем огинаючої результуючий мовний сигнал.

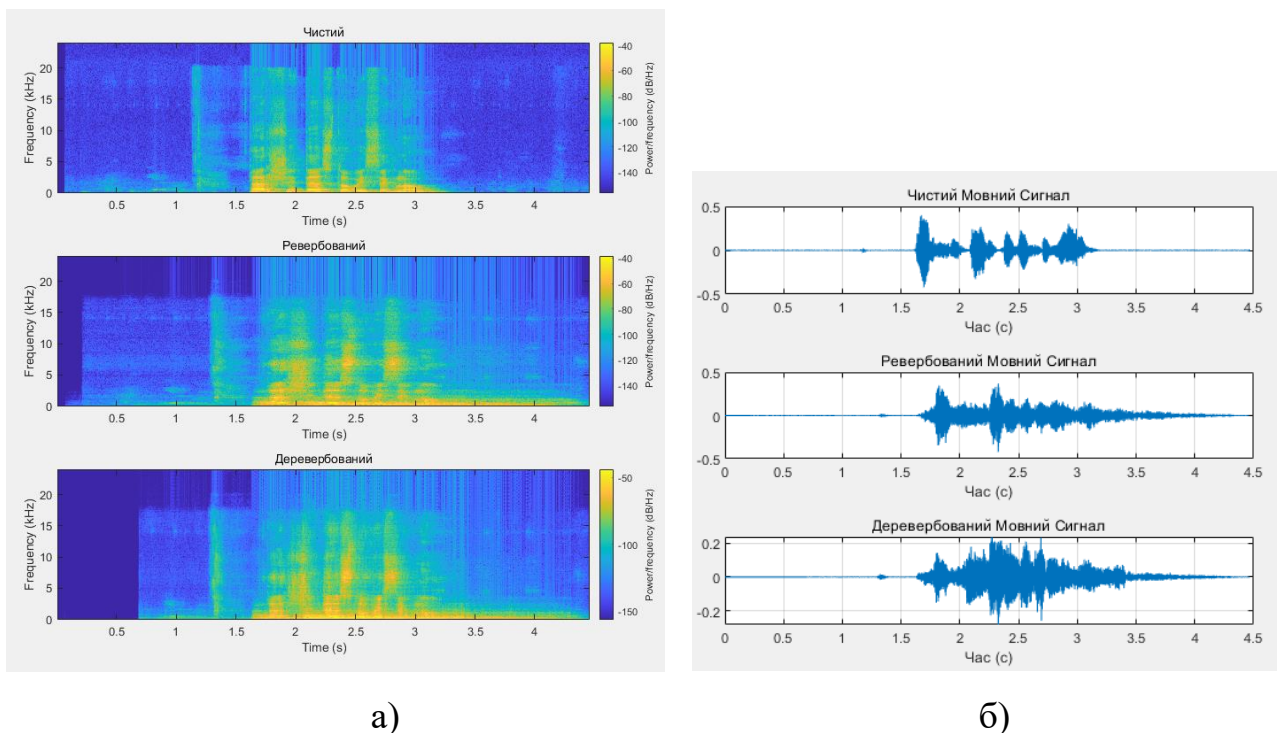


Рисунок 3.11 – Спектрограми (а) та сигналами (б) для тестового семплу українською мовою

У випадку аналізу запису українською мовою можна переконатись, враховуючи зміщення праворуч  $\Delta_1=0,2$  сек, що отриманий дереверберований сигнал отримав додаткові ознаки. Пояснення цьому є те, що алгоритм отримання реверберованого сигналу сильно спотворив оригінальний сигнал і мережа не повністю відпрацьовує відновлення. Крім цього, слід враховувати що навчання цієї мережі переважно проводилось на англійському датасеті.

Також для перевірки якості результуючого сигналів була використана група експертів, склад якої наведено у пункті 3.3.3 роботи.

Так само, як і раніше, було запропоновано дати оцінку якості ревербованого та деревербованого сигналів для двох мов від 0 до 10. Результати опитування наведені в таблиці 3.9.

Таблиця 3.9 – Результати опитування

Експерт	Якість ревербованого сигналу для англійської мови	Якість результуючого сигналу для англійської мови	Якість ревербованого сигналу для української мови	Якість результуючого сигналу для української мови
Людина 1	4	6	4	6
Людина 2	4	7	4	6
Людина 3	3	6	3	6
Людина 4	4	7	4	5
Людина 5	5	6	4	6
Людина 6	3	6	3	6
Людина 7	4	5	4	5
Людина 8	4	7	3	6
Людина 9	3	6	4	7
Людина 10	4	6	3	6

Результати опитування експертів свідчать, що в середньому якість прослуханого сигналу до деревербації становить 3.8 для англійської мови та 3.6 для української, а після деревербації 6.2 та 5.9 відповідно. Хоча за результатами і не отримано серед групи експертів найвищого експертного балу, але все одно якість запроваджених процедур деревербації призвело до підвищення якості сигналу за критеріями розбірливості слів, точності вимовлення окремих слів (тобто, чи не налазять слова один на одне), чи достатньо їх гучність звучання.

Перевіримо далі, як нейронна мережа буде функціонувати, якщо у структуру основного сигналу, в якому записано українську фразу буде додано

адитивний шум різної конфігурації. Цю перевірку виконаємо для дослідження характеристик навчальної аудиторії 209 (п.3.3.3).

### 3.3.6 Експеримент з додаванням фонового шуму

Ускладнимо трохи ситуацію і для перевірки нейронної мережі розглянемо запис тестового семплу українською мовою “Перший тестовий сигнал” тривалістю 4,4 секунди і додамо адитивно шум (використовуючи інструменти програми Adobe Audition Trial version) тривалістю 2,5 сек (тобто, шум накладається лише на частину семплу), в якому для експерименту використаємо чотири види найбільш поширених види шуму: коричневий (brown), рожевий (pink), сірий (grey) та білий (white). Так як і вище всі дії будемо виконувати для одного з двох каналів – лівого, оскільки для правого каналу результати будуть ідентичні. Параметри шуму були визначені за умови наступних налаштувань (рис.3.12):

- інтенсивність шуму 30 дБ;
- offset=0;
- тривалість впливу 2,5 сек;
- режим накладання Overlap(mix).

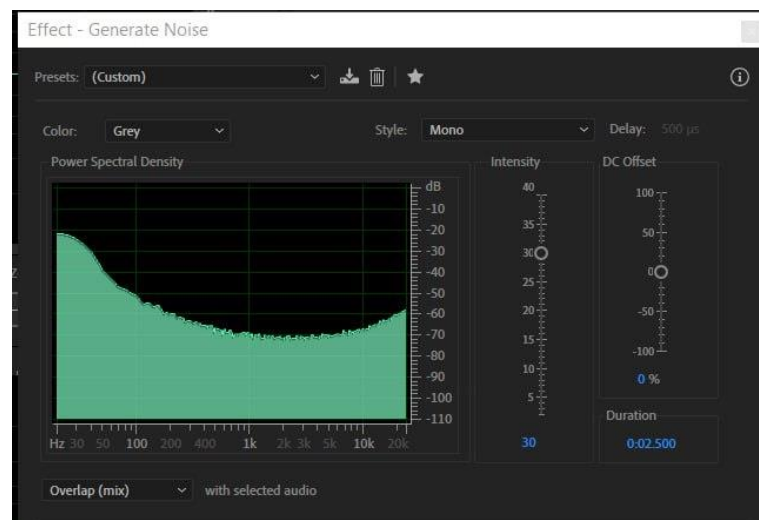
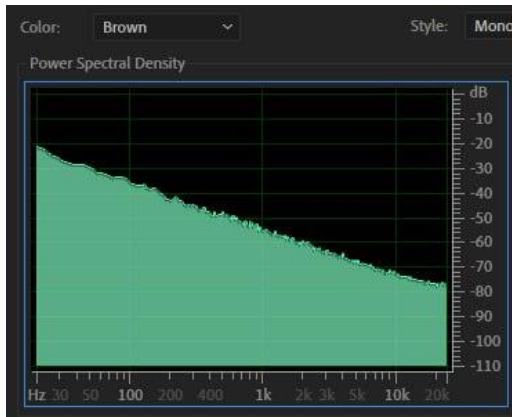


Рисунок 3.12 – Налаштування параметрів адитивного шуму

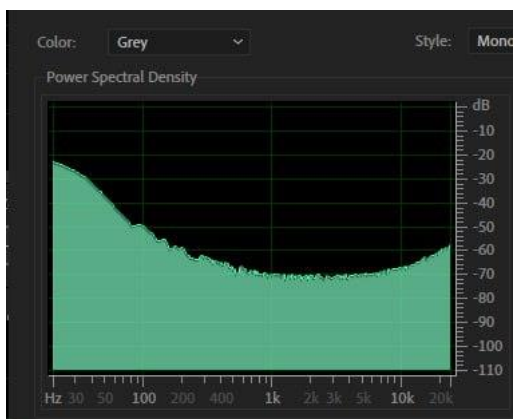
Розподіл щільності спектральної потужності обраних типів шуму наведено на рисунку 3.13 (а-г).



а)



б)



в)

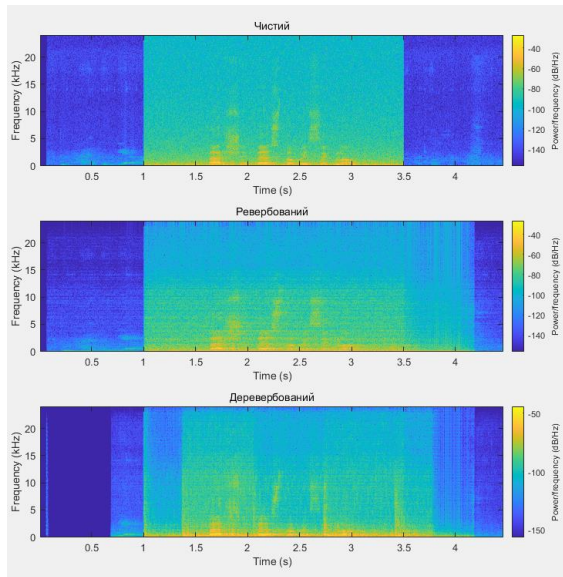


г)

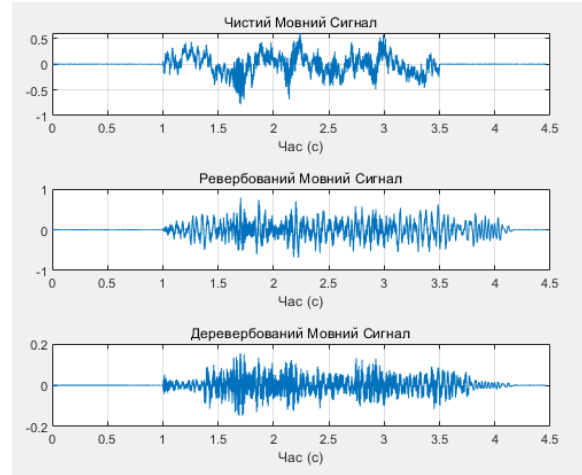
Рисунок 3.13 – Візуальне зображення видів шуму для експерименту: а) коричневий; б) рожевий; в) сірий; г) білий

Аналізуючи отримані результати на рисунках 3.14-3.17, можна виявити наступні характерні особливості. По-перше, нейронна мережа реагує при відключенні шуму, так це яскраво можна помітити при значенні 3,5 сек і особливо яскраво це помітно на спектрограмі при застосуванні білого шуму. По-друге, саме у випадку білого шуму, як це впливає з рисунку 3.17 а) та б) нейронна мережа не може якісно реалізувати процедуру дереверберації. Кращі результати можна отримати у випадку застосування коричневого шуму, за умови накладання шуму повністю на весь семпл. Для випадку рожевого шуму, судячи з спектрограм характерна наявність певного паттерна, який відсутній у “чистому” сигналі. За аналізом сигналограм менше за все впливає на процедуру

дереверберації накладений сірий шум. Що не можна сказати так само при використанні рожевого та білого шумів.

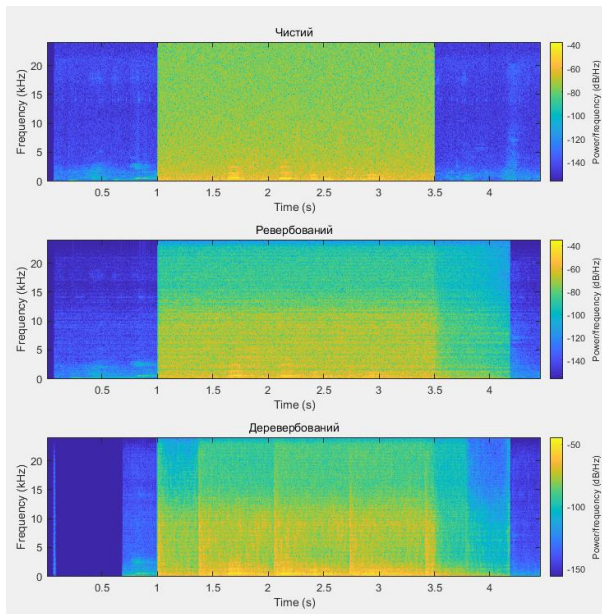


а)

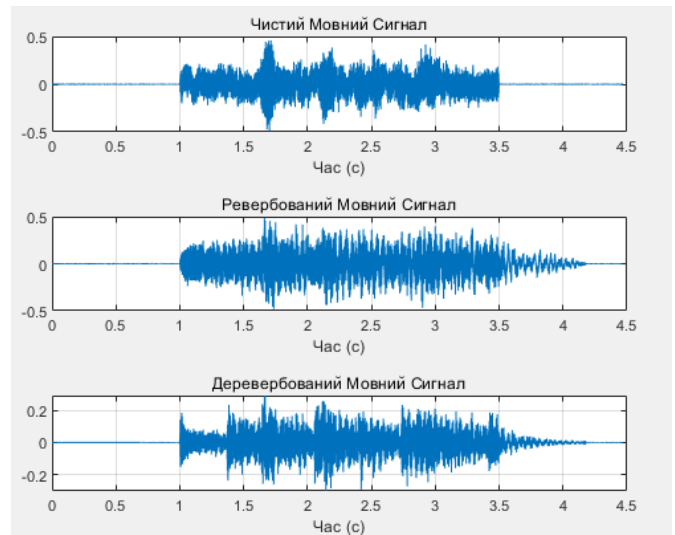


б)

Рисунок 3.14 – Спектрограми (а) та сигналограми (б) для тестового семплу українською мовою з доданим коричневим шумом



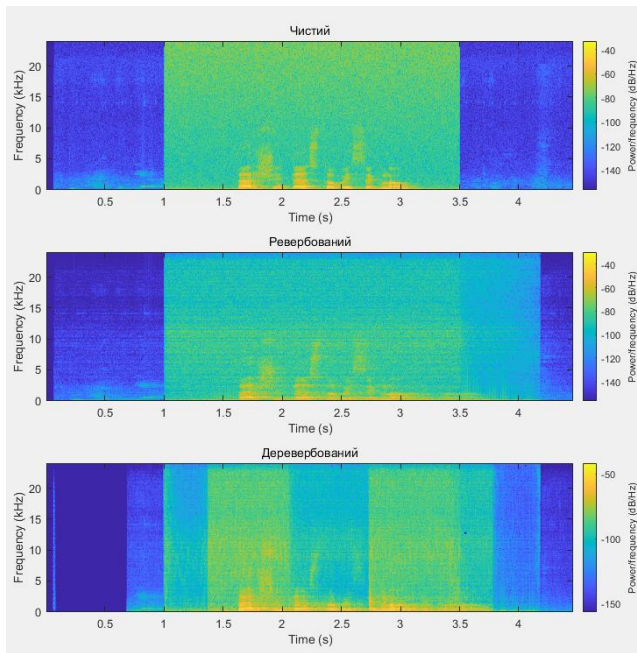
а)



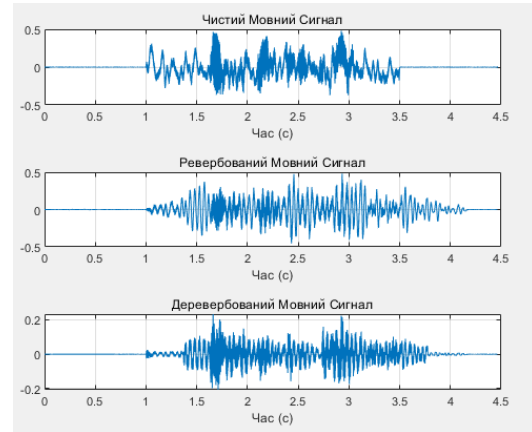
б)

Рисунок 3.15 – Спектрограми (а) та сигналограми (б) для тестового семплу українською мовою з доданим рожевим шумом



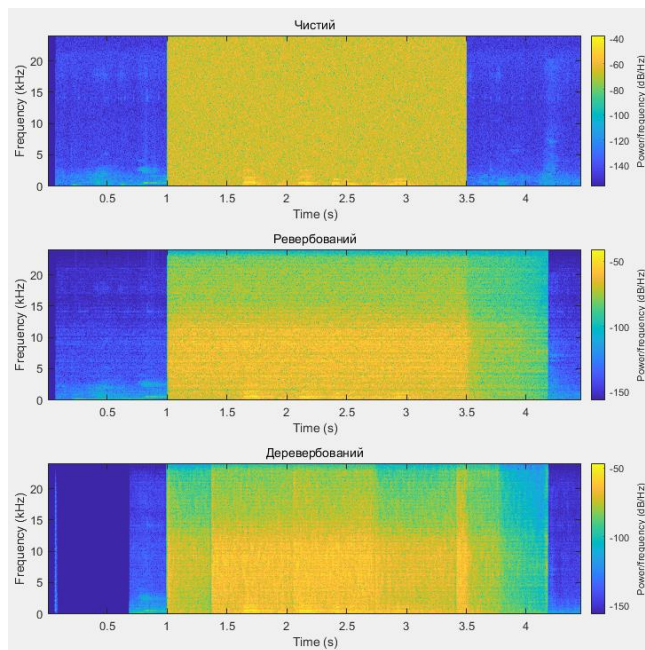


а)

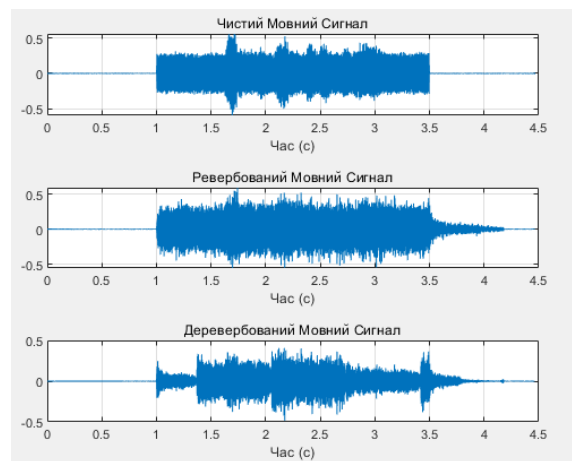


б)

Рисунок 3.16 – Спектрограми (а) та сигналограми (б) для тестового семплу українською мовою з доданим сірим шумом



а)



б)

Рисунок 3.17 – Спектрограми (а) та сигналограми (б) для тестового семплу українською мовою з доданим білим шумом

Для перевірки якості результуючих сигналів, також була використана група експертів, яка сформована в пункті 3.3.3 роботи.

Було запропоновано дати оцінку якості мовних сигналів з накладеним шумом, ревербованих та деревербованих для яких ставився експеримент вище від 0 до 10. Результати опитування наведено в таблиці 3.10.

Таблиця 3.10 – Результати опитування

Експерт	Якість сигналу з накладеним шумом	Якість ревербованого сигналу	Якість результуючого сигналу
Коричневий шум			
Людина 1	4	2	5
Людина 2	2	1	3
Людина 3	1	0	3
Людина 4	1	0	2
Людина 5	2	0	4
Людина 6	3	2	5
Людина 7	1	0	4
Людина 8	2	1	3
Людина 9	3	0	4
Людина 10	2	1	4

Продовження таблиці 3.10 – Результати опитування

Експерт	Якість сигналу з накладеним шумом	Якість ревербованого сигналу	Якість результуючого сигналу
Сірий шум			
Людина 1	3	1	4
Людина 2	1	0	3
Людина 3	4	1	3
Людина 4	2	1	3
Людина 5	2	1	2
Людина 6	4	1	4
Людина 7	4	0	3
Людина 8	3	2	4
Людина 9	2	0	3
Людина 10	2	1	2
Рожевий шум			
Людина 1	1	0	1



Продовження таблиці 3.10 – Результати опитування

Експерт	Якість сигналу з накладеним шумом	Якість ревербованого сигналу	Якість результуючого сигналу
Рожевий шум			
Людина 2	1	0	1
Людина 3	2	0	1
Людина 4	1	1	1
Людина 5	1	0	1
Людина 6	1	0	2
Людина 7	0	0	2
Людина 8	1	0	1
Людина 9	2	1	1
Людина 10	0	0	3
Білий шум			
Людина 1	0	0	1
Людина 2	1	0	1

Продовження таблиці 3.10 – Результати опитування

Експерт	Якість сигналу з накладеним шумом	Якість ревербованого сигналу	Якість результуючого сигналу
Білий шум			
Людина 3	1	0	0
Людина 4	1	0	1
Людина 5	1	0	1
Людина 6	1	0	2
Людина 7	0	0	1
Людина 8	1	0	1
Людина 9	1	0	1
Людина 10	0	0	2

В таблиці 3.11 наведені середні значення якості сигналу для 4 видів шумів.

Аналізуючи отримані значення на основі експертних оцінок з вибірки людей групи можна зазначити, що розроблена система краще справляється щодо покращення сигналу для коричневого і сірого шуму, в порівнянні з рожевим та білим, де якість на жаль все ж залишається низькою.

Таблиця 3.11 – Середнє значення результатів опитування

Тип доданого адитивного шуму	Якість сигналу з накладеним шумом	Якість ревербованого сигналу	Якість результуючого сигналу
Коричневий	2.1	0.7	3.7
Сірий	2.7	0.8	3.1
Рожевий	1	0.2	1.4
Білий	0.7	0	1.1

Таким чином, при застосуванні запису з шумом, необхідно враховувати шум і у випадку навчання нейронної мережі. Таке твердження пояснюється тим, що сама природа шуму, як наприклад для випадку білого шуму, сильно впливає в режимі Overlap на саму структуру мовного сигналу і тут для отримання кращих результатів з дереверберації варто використовувати попередню обробку сигналу, яка спрямована на зменшення рівня шуму при запису.

### 3.4 Система клонування голосу

#### 3.4.1 Етап навчання елементів системи клонування голосу

Як було відмічено у п. 2.3, відповідно до структурної схеми з рисунку 2.5, система клонування голосу включає в себе три нейронні мережі, які проходять навчання незалежно одна від одної в послідовному порядку. Причому, метод перетворення тексту на мовлення є висловлювання із заданими акустичними характеристиками реалізовано на основі технології SV2TTS. Виходячи з розробленої структурної схеми система складається з трьох елементів – кодер голосу, синтезатор та вокодер.

При навчанні нейронної мережі, яка визначає роботу кодера голосу, щоб уникнути пауз у вхідних даних, підключено бібліотеку webrtcvad8 середовища

pyCharm. Навчання нейронної мережі кодера голосу проводиться на основі міксованого набору даних LibriSpeech (англ. фрази) та бази даних Mozilla Common Voice [91]. Семпли в наборі даних LibriSpeech є монофонічними та мають частоту дискретизації 16 кГц. Для навчання нейронної мережі кодера використовувалися логарифмічні MEL-спектрограми. Такий підхід і використання MFCC коефіцієнтів обрано з тим, щоб врахувати особливості сприйняття звуку людським слухом, про що було зазначено в п.2.1.1. Побудова Mel-спектрограм проводилась з врахуванням наступних особливостей:

- кількість Mel-каналів: 40;
- розмір часового вікна для фреймінгу: 25 мс;
- крок між вікнами: 10 мс;
- частота дискретизації сигналу: 16 кГц.

Синтезатор послідовності, який є ланкою системи клонування голосу (рис.2.5) реалізовано на основі модуля TensorFlow і бібліотеки Tacotron5, генерує MEL-спектрограму з послідовності фонем, яка залежить від значень вектора вбудовування. Важливо зазначити, що цей вектор об'єднується (конкатенується) з виходом кодера синтезатора на кожному часовому кроці. Навчання нейронної мережі синтезатора відбувається на основі пар текстових транскрипцій та відповідних аудіозаписів.

На вході система перетворює текст на послідовність фонем, а окремі символи тексту визначаються як одновимірні вектори та подаються на згорткові шари LSTM-мережі (нейронна мережа 2 з рис.2.5) у вигляді кадрів. Після обробки ці кадри передаються на вхід декодера, де кожен кадр об'єднується з попереднім вихідним кадром, створюючи авторегресійну модель.

MEL-спектрограма генерується за допомогою згорткової функції з вікном Хаана шириною 50 мс і кроком 12,5 мс. Вхідний сигнал обробляється через банк Mel-фільтрів із 80 каналами. Тривалість висловлювань для навчання нейронної мережі кодера синтезатора не повинна перевищувати 11 сек, при тривалості сегмента з паузою не більше 0,4 сек [60].

Процес навчання нейронної мережі синтезатора триває 150 тисяч ітерацій, при цьому розмір пакету даних (batch size) складає 144 кадри.

Для моделі вокодера з рисунку 2.5 визначена нейронна мережа глибокого навчання WaveNet. Сама архітектура нейронної мережі складається з 3 стеків одновимірних розширених згорток з коефіцієнтами, які за експонентою зростають по мірі заглиблення шару в архітектурі нейронної мережі. Кожен стек містить 10 залишкових блоків, і в кожному з них розташовано по два згорткових шари. Тобто, в даній моделі нейронної мережі використано 60 згорткових шарів, або 30 шарів розширеної згортки. В основі роботи вокодера реалізована авторегресійна модель, основне призначення якої полягає у інвертуванні синтезованих спектрів в часові форми сигналів. Тобто, для вокодера WaveNet – згорткова нейронна мережа глибокого навчання, яка створює на виході прогнозовані відліки мовного сигналу, використовуючи при цьому авторегресійний спосіб. Крім цього, на етапі роботи вокодера передбачено етап пакетної дискретизації з довжиною сегмента 8000 відліків і довжиною перекриття 400 відліків. Базова частота дискретизації дорівнює 16 кГц. Перекриття сегментів використовується для того, щоб зберегти певний контекст між завершенням попереднього сегменту і початком наступного сегменту в послідовності даних. Тобто, невеликий фрагмент кінця попереднього сегменту повторюється на початку наступного сегменту. Далі відрізки, що перекриваються, об'єднуються на основі перехресного згладжування. Слід відмітити, що на кожному кроці навчання нейронної мережі моделі вокодера MEL-спектрограма розрізається на однакову кількість сегментів і пропускається через ResNet-модуль. У підсумку створюється вектор умов, який далі поділяється порівну на чотири частини. Перша з цих частин об'єднується з дискретизованою MEL-спектрограмою і з відрізком форми синтезованого сигналу з попереднього часового кроку. Результуючий вектор проходить через кілька перетворень і перед кожним кроком вектор умов об'єднується з проміжною формою синтезованого сигналу. Таким чином, два щільні шари моделі утворюють розподіл звукових значень синтезованого

сигналу за дискретними значеннями, які відповідають дев'яти бітному кодуванню аудіо [60]. Слід відмітити, що програмний лістинг команд розробленої системи клонування голосу наведено у додатку В.

### 3.4.2 Практичний експеримент

Для тестування нейронних мереж в складі системи клонування голосу (рис.2.5) записувались тестові семпли мікрофоном USB BOYA BY-M100UA на відстані 50 см від джерела сигналу. Мікрофон має наступні характеристики:

- робочий частотний діапазон 50 Гц-18 кГц;
- частота дискретизації 48 кГц;
- чутливість -36дБ;
- сигнал/шум: 78дБ.

Перед використанням частота дискретизації змінювалась до рівня того значення, на якій навчалась система, а саме до 16 кГц.

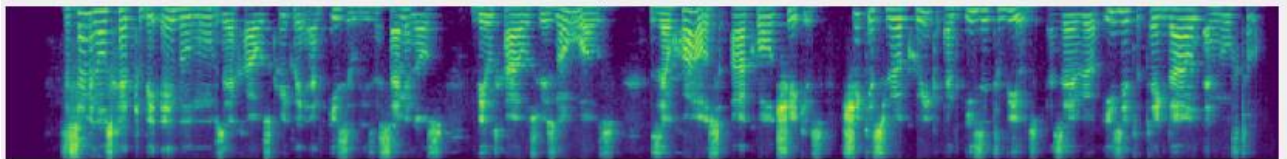

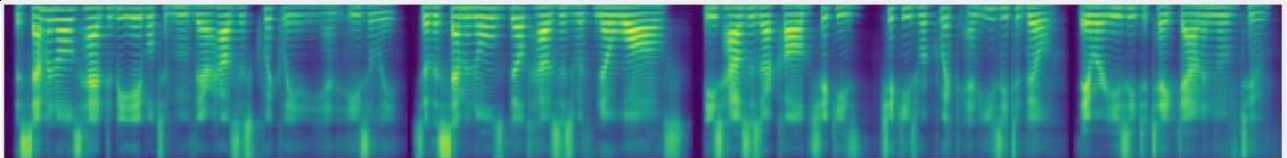
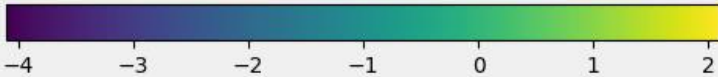
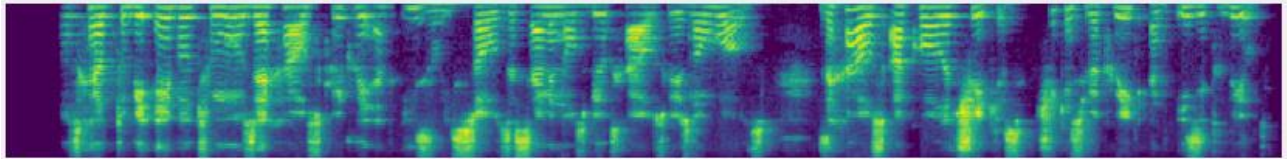

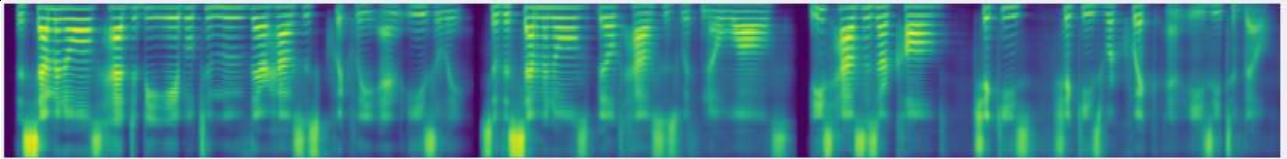

Основна ідея практичного експерименту полягає у тому, щоб визначити як система клонування голосу створює голосове повідомлення, коли на її вхід подається голосові фрази, як утворено спочатку з англійських слів, потім з українських слів. При цьому, в даній ситуації враховується чи є автор фрази на вході системи, носієм цієї мови. Для додаткової перевірки повідомлення, які подаються на вхід системи клонування записано з різною інтонацією та швидкістю вимови окремих слів фрази.

В таблицях 3.12 та 3.13 наведено результати першої перевірки системи, коли на вхід послідовно подаються звукові семпли з різною тривалістю. Різниця в даному випадку у тому, вимова фрази відбувається носієм чи не носієм мови. Слід зазначити, що створення синтезованого сигналу і відповідно спектрограми, відбувається з певною часовою затримкою. Проте з аналізу спектрограм на таб. 3.12 (семпл 1) з урахуванням цього зміщення, прослідковуються чотири переходи, які визначають окремі речення у фразі. Тобто, аналіз фрази та його клону можна умовно для даного випадку поділити

на чотири фрагменти. Порівняння в межах, наприклад першого речення: “The Seminoles had to be able to move their houses quickly and easily” дозволяє стверджувати, що синтезований сигнал за виглядом MEL-спектрограми має певне розмиття (своєрідне погіршення “чіткості”) зображення спектрограми. Така сама тенденція наявна і в наступних етапах проведеної перевірки системи клонування голосу. Ця особливість, вірогідно, пов’язана з внутрішніми налаштуваннями нейронних мереж системи. В цілому, з врахуванням певної затримки роботи нейронної мережі та зміни енергетичної складової синтезованого сигналу (синтезований сигнал вимовляється трохи гучніше), аналіз показав високу схожість з окремими словами оригінального тексту фрази та синтезованого. Крім цього, можна відмітити, що три нейронні мережі в результаті роботи системи клонування голосу додають слабке “рипіння” при синтезі. Відмітимо, що вимова семплів в таблиці 3.12 створено не носієм мови, а в таблиці 3.13 носієм мови.

Таблиця 3.12 – MEL-спектрограми для семплів 1 та 2

Дані на вході	Синтез	Номер семплу та тривалість (особливості)	Фраза семплу, тривалість	Фраза на виході НМ, тривалість
The Seminoles had to be able to move their houses quickly and easily. So the Seminoles built houses with no walls. These houses are called chickees. Chickees were quick and easy to build. They were easy to take down and move, too.	The Seminoles had to be able to move their houses quickly and easily. So the Seminoles built houses with no walls. These houses are called chickees. Chickees were quick and easy to build. They were easy to take down and move, too.	1 (особистий голос; не носій мови)	13.3 сек	12.8 сек

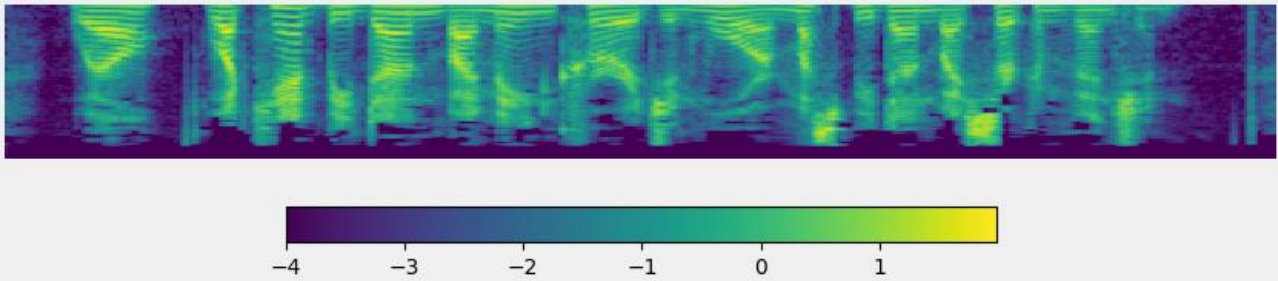
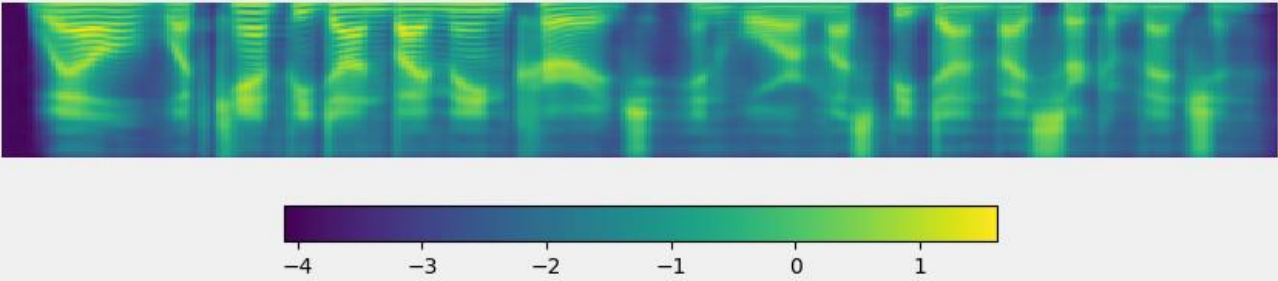
MEL-спектрограма семплу:				
				
				
MEL-спектрограма створена системою:				
				
				
The Seminoles had to be able to move their houses quickly and easily. So the Seminoles built houses with no walls. These houses are called chickees. Chickees were quick and easy to build.	The Seminoles had to be able to move their houses quickly and easily. So the Seminoles built houses with no walls. These houses are called chickees. Chickees were quick and easy to build.	2 (особистий голос; не носій мови)	9.4 сек	10.6 сек
MEL-спектрограма семплу:				
				
				
MEL-спектрограма створена системою:				
				
				

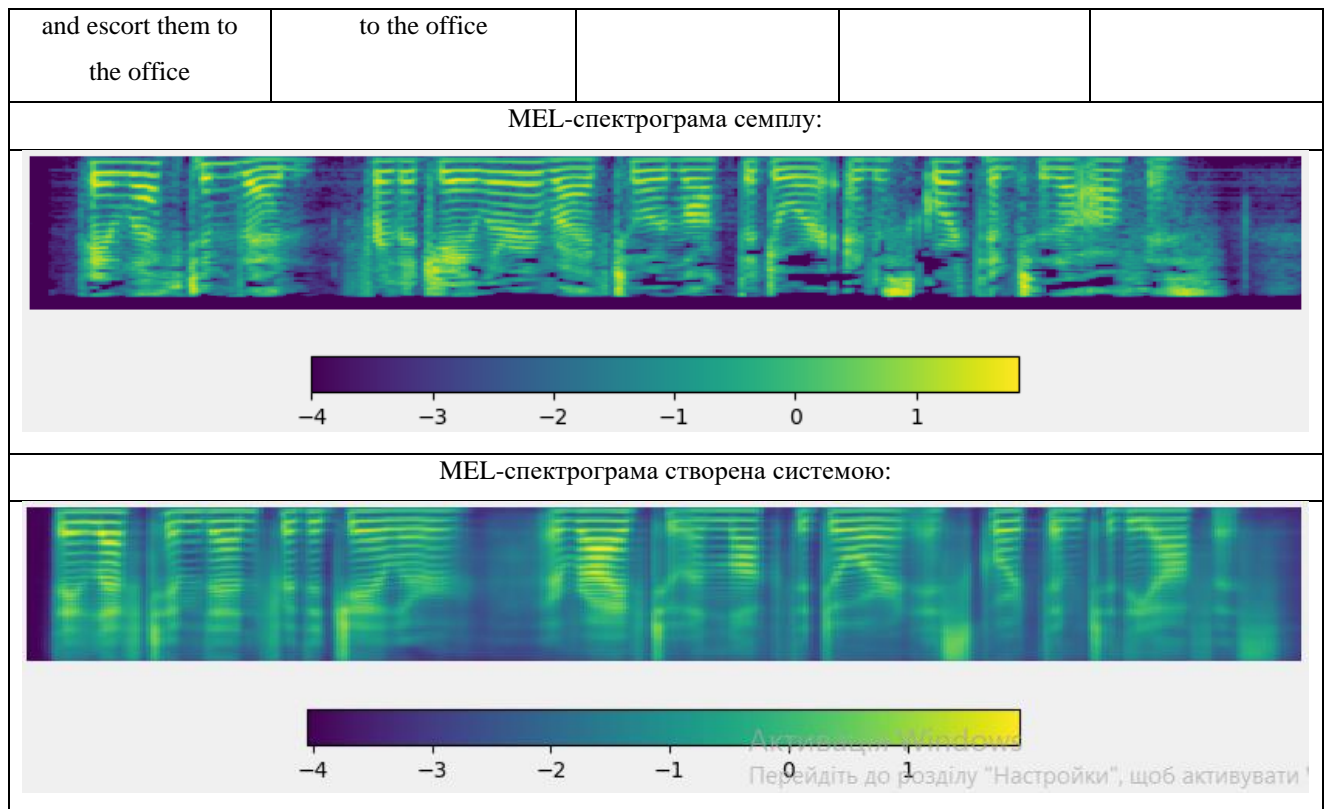


В таблиці 3.12 для семплу 2 вимова фрази скорочена на одне речення у порівнянні з семплом 1. І для семплу 2 (табл.3.12) можна відмітити ще одну особливість - для трьох речень фрази за виглядом MEL-спектрограми система клонування голосу починає синтезувати голос навіть трохи раніше.

В таблиці 3.13 наведено спектрограми, коли вхідна фраза англійською мовою вимовляється носієм мови. Аналізуючи отримані спектрограми, у випадку коли оригінальна фраза вимовляється носієм мови (семпли 3 та 4), можна відмітити, що крім певного розмиття, та невеликої затримки картина розподілу виявилась при порівнянні кращою ніж у випадку, коли диктор не є носієм мови.

Таблиця 3.13 – MEL-спектрограми для семплів 3 та 4

Дані на вході	Синтез	Номер семплу та тривалість (особливості)	Фраза семплу, тривалість	Фраза на виході НМ, тривалість
Well, what can't be done by main courage in war must be done by circumvention?	Well, what can't be done by main courage in war must be done by circumvention?	3 (чоловік, носій мови)	4.95 сек	4.25 сек
MEL-спектрограма семплу:				
				
MEL-спектрограма створена системою:				
				
They left him then for the jailer arrived to unlock the door	They left him then for the jailer arrived to unlock the door and escort them	4 (жінка, носій мови)	5 сек	4.6 сек



Так само як і у попередніх етапах експериментального дослідження, для перевірки схожості клонованого голосу з оригіналом використано групу експертів, склад якої визначено у пункті 3.3.3 роботи. Було запропоновано експертам дати оцінку схожості клонованого голосу з оригіналом для носія і не носія мови по шкалі від 0 до 10. Результати опитування наведено в таблиці 3.14.

Таблиця 3.14 – Результати опитування

Експерт	Оцінка схожості для носія мови	Оцінка схожості для не носія мови
Людина 1	8	7
Людина 2	8	8
Людина 3	7	6
Людина 4	7	7
Людина 5	7	6
Людина 6	9	7

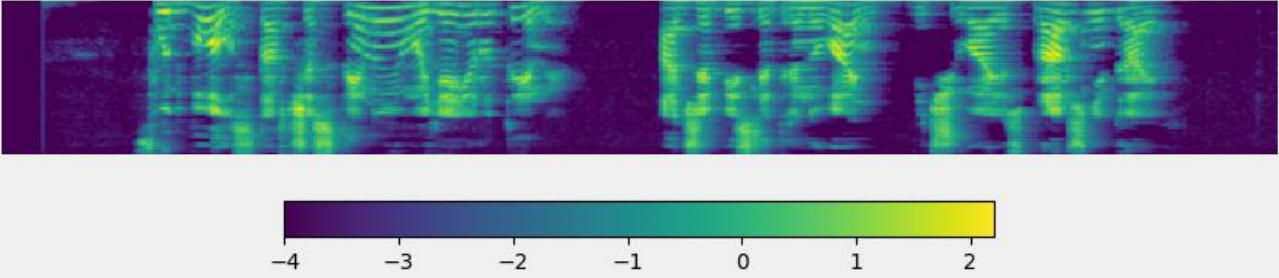
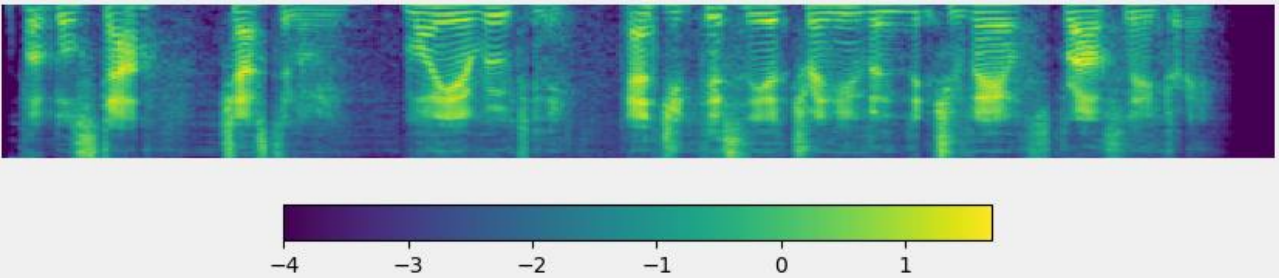
Продовження таблиці 3.14 – Результати опитування

Експерт	Оцінка схожості для носія мови	Оцінка схожості для не носія мови
Людина 7	8	7
Людина 8	7	6
Людина 9	7	6
Людина 10	8	7

Результати отриманих експертних оцінок з таблиці 3.14 свідчать, що в середньому схожість клонованого сигналу з оригінальним становить 7.6 для носія мови та 6.7 для не носія мови. Тобто, отримано підтвердження в рамках наукового експерименту, що існує вплив на роботу розробленої системи клонування стосовно того, що саме вимовлення оригінальної фрази виконується носієм мови або ні. В даному випадку перевірялась саме англійська мова.

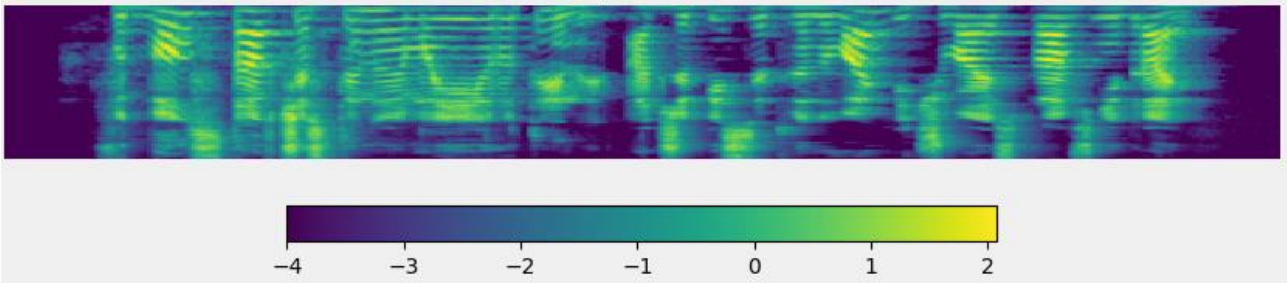
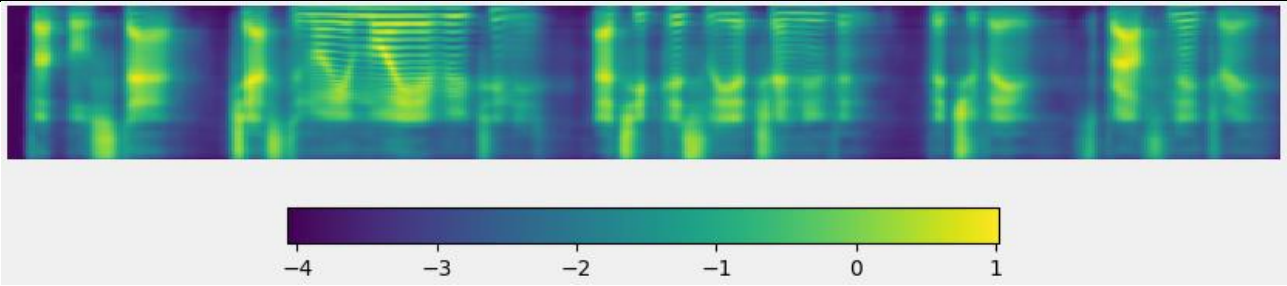
Далі перевіримо як працює запропонована система клонування голосу, коли на вхід подається фраза українською мовою. В таблиці 3.15 наведено перший приклад перевірки системи клонування голосу. В даному випадку вимовлення фрази “ Футбол став частиною моєї рутини, адже після тренувань відчуваю справжній драйв” зроблено зі звичайною інтонацією та швидкістю вимовлення окремих слів фрази. Тут слід відмітити, що сама система клонування голосу формує фразу трохи швидше (5,35 сек проти 7 сек) і це пов’язано з тим, що при синтезі голосу аналізуються і інтервали тиші та пауз і якщо подивитись на MEL-спектрограму семплу 5, то посередині є інтервал (у фразі це відповідає розділовий знак “кома”) з тишою, який прибирається при синтезі системою.

Таблиця 3.15 – MEL-спектрограми для семплу 5

Дані на вході	Синтез	Номер семплу та тривалість (особливості)	Фраза семплу, тривалість	Фраза на виході НМ, тривалість
Футбол став частиною моєї рутини, адже після тренувань відчуваю справжній драйв	Футбол став частиною моєї рутини, адже після тренувань відчуваю справжній драйв	5 (особистий голос; носій мови)	7 сек	5.35 сек
MEL-спектрограма семплу:				
				
MEL-спектрограма створена системою:				
				

В таблиці 3.16 використана та сама фраза українською мовою, що і для таблиці 3.15, але в цьому випадку вимовлення фраз у оригінальному семплі виконано з підвищеною швидкістю проголошення літер. При пришвидшенні вимови чутно погіршення синтезованого семлу в порівнянні з семлом згенерованим на звичайному вхідному семлі. Синтезований сигнал вимовляється трохи тихіше.

Таблиця 3.16 – Отримані результати для семплу 6

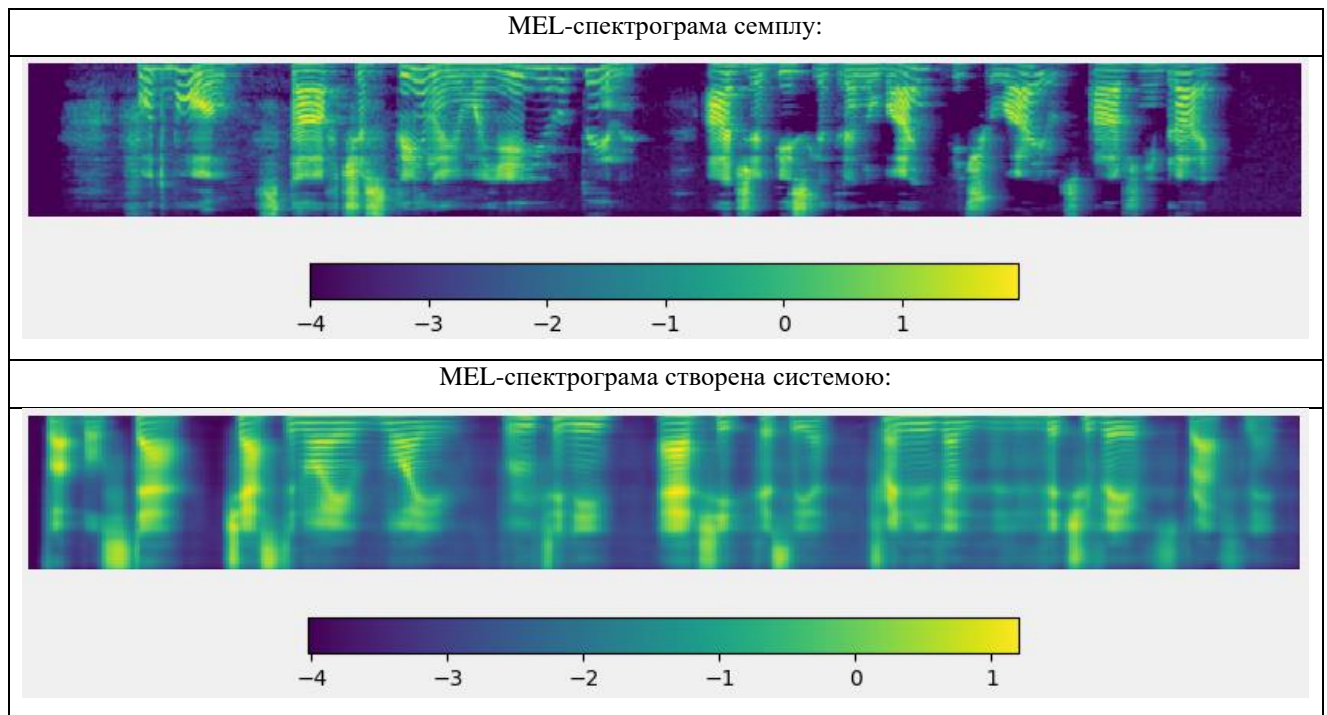
Дані на вході	Синтез	Номер семплу та тривалість (особливості)	Фраза семплу, тривалість	Фраза на виході НМ, тривалість
Футбол став частиною моєї рутини, адже після тренувань відчуваю справжній драйв	Футбол став частиною моєї рутини, адже після тренувань відчуваю справжній драйв	6 (особистий голос; носії мови)	5 сек	5.2 сек
MEL-спектрограма семплу:				
				
MEL-спектрограма створена системою:				
				

В таблиці 3.17 використана та сама фраза українською мовою, що і для таблиці 3.15, але в цьому випадку вимовлення фраз у оригінальному семплі виконано з певним сповільненням вимову слів фрази.

Таблиця 3.17 – Отримані результати для семплу 7

Дані на вході	Синтез	Номер семплу та тривалість (особливості)	Фраза семплу, тривалість	Фраза на виході НМ, тривалість
Футбол став частиною моєї рутини, адже після тренувань відчуваю справжній драйв	Футбол став частиною моєї рутини, адже після тренувань відчуваю справжній драйв	7 (особистий голос; носії мови)	9 сек	6.7 сек



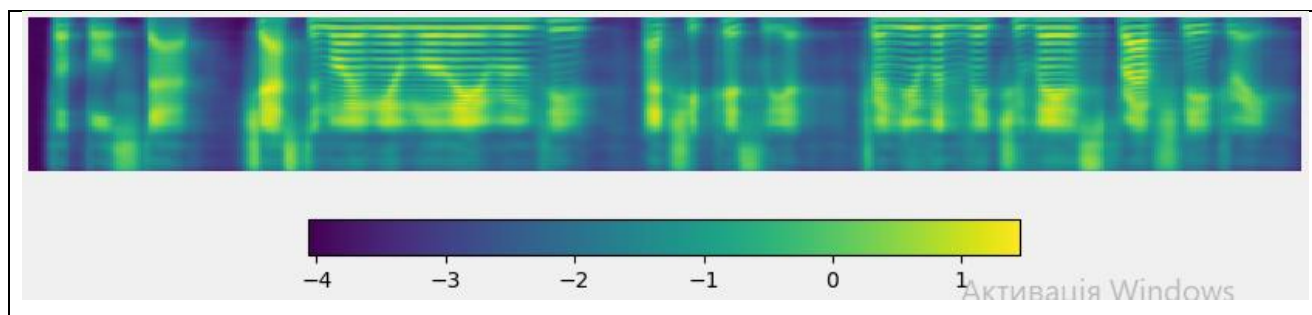


При сповільненні вимови також сповільнюється вимова і в генерованому мовному семплі як результат роботи системи клонування голосу.

Таблиця 3.18 містить результати роботи системи клонування голосу. В даному випадку використано ту саму фразу українською мовою, що і для даних таблиці 3.15, але самі слова вимовляються в оригіналі зі зміненою інтонацією.

Таблиця 3.18 – Отримані результати для семплу 8

Дані на вході	Синтез	Номер семплу та тривалість (особливості)	Фраза семплу, тривалість	Фраза на виході НМ, тривалість
Футбол став частиною моєї рутини, адже після тренувань відчуваю справжній драйв	Футбол став частиною моєї рутини, адже після тренувань відчуваю справжній драйв	8 (особистий голос; носій мови)	7.1 сек	5.35 сек
MEL-спектрограма семплу:				
MEL-спектрограма створена системою:				



Аналіз отриманих спектрограм з таблиці 3.18 свідчить, що система клонування голосу генерує синтезований голос з певним пришвидшенням на початку і внаслідок зміненої інтонації незважаючи, що тривалість клону менше та нібито розтягує трохи розподіл за інтенсивністю окремих слів фрази. Синтезований семпл 8 вимовляється трохи тихіше ніж оригінал. Для перевірки схожості клонованого голосу з оригіналом була залучена група експертів, яка сформована в пункті 3.3.3 роботи. Було запропоновано дати оцінку схожості клонованого голосу з оригіналом для згенерованих сигналів які наведені в таблицях 3.15, 3.16, 3.17 та 3.18 за шкалою від 0 до 10. Результати опитування наведено в таблиці 3.19.

Таблиця 3.19 – Результати опитування

Експерт	Оцінка схожості для семплу 5	Оцінка схожості для семплу 6	Оцінка схожості для семплу 7	Оцінка схожості для семплу 8
Людина 1	6	6	6	5
Людина 2	6	5	5	6
Людина 3	7	6	6	8
Людина 4	5	6	6	5
Людина 5	6	7	6	6
Людина 6	5	5	5	5
Людина 7	7	6	6	7
Людина 8	7	7	5	7
Людина 9	6	6	6	6
Людина 10	8	7	7	7

В таблиці 3.20 наведені середні значення схожості семплів з таблиці 3.19.

Таблиця 3.20 – Середні значення результатів опитування

Семпл	Середнє значення схожості
5	6.3
6	6.1
7	5.8
8	6.2

На основі даних таблиць 3.19 та 3.20 можна стверджувати, що при сповільненні вимови спостерігається погіршення схожості синтезованого голосу з оригіналом. Також результати експертів показали схожість тембру та окремих слів оригінального тексту фрази та синтезованого. Отримані менші значення схожості української мови у порівнянні з англійською мовою, свідчить про необхідність навчання системи на більш якісному наборі даних української мови.

### 3.5 Оцінка якості системи ідентифікації голосу

Як показано вище, якість розробленої системи дереверберації та системи клонування голосу перевірено на основі застосування методу експертних оцінок, про що свідчать отримані результати опитувань вибірки групи людей з 10 осіб (п.3.3.3). Натомість, для оцінки якості системи ідентифікації за голосом (п.3.1.2) використаємо інший спосіб, а саме, на основі побудови метрики якості моделі нейронної мережі [92]. Для цього, проаналізуємо отримані дані таблиці 3.2, де наведено вірогідність схожості чужих записів голосів людей (23 зразки) з набору датасету з еталонним записаним голосом. Показано, що для трьох випадків нейронна мережа спрацювала з помилкою, визначивши хибну належність до еталонного голосу (зразки під номерами 4,12 та 18). В якості метрики якості системи використано в роботі дві метрики оцінки прогнозованої сили моделі - ROC (Receiver Operating Characteristic curves) та AUC (Area Under



the Curve) [92]. Крива ROC являє собою графічну залежність двох параметрів зміни швидкості значень – True Positive Ratio (TPR) та False Positive ratio (FPR). Ці параметри визначаються на основі таких співвідношень:

$$TPR = \frac{a}{a + b};$$

$$FPR = \frac{c}{c + d},$$

де  $a$ - істинно додатній результат (тобто, нейронна мережа визначає не приналежність зразку до еталонного голосу і в реальності так і є);

$b$ - хибно від'ємний результат (нейронна мережа показує, що голос зразку належить еталонному, але насправді це не так);

$c$ - хибно додатній результат (нейронна мережа стверджує, що голос зразку не належить еталонному, а в реальності це не так);

$d$  – істинно від'ємний результат (нейронна мережа показує що голос належить еталонному і в реальності це правда).

Крива ROC за умови ідеальної роботи нейронної мережі, коли не було б трьох хибних визначень має вигляд:

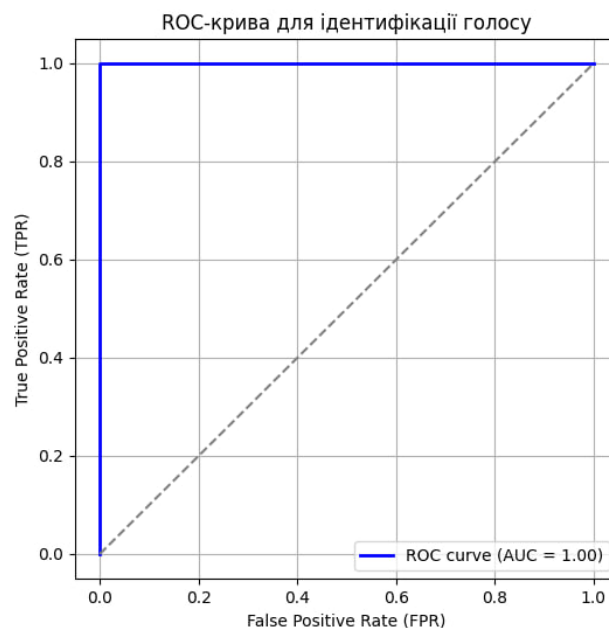


Рисунок 3.18 – Метрика ROC оцінки прогнозованої сили моделі при ідеальній роботі нейронної мережі системи ідентифікації за голосом

Для отриманих даних експерименту крива ROC матиме вигляд, як показано на рисунку 3.19.

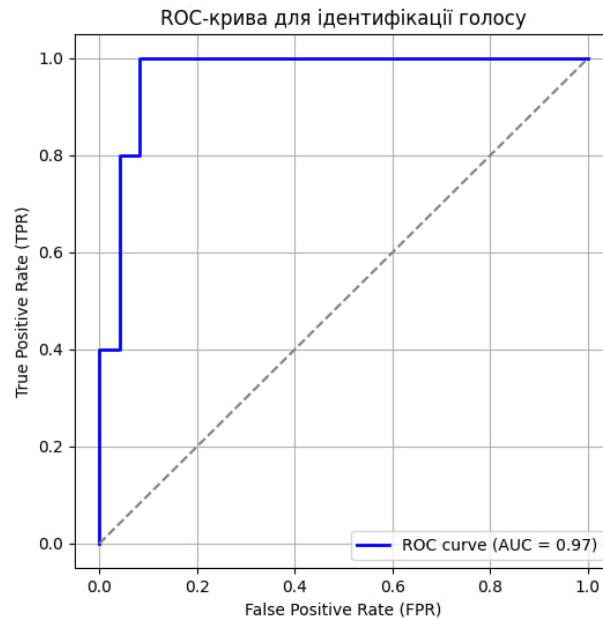


Рисунок 3.19 – Метрика ROC з експерименту

Метрика AUC являє собою площу покриття простору під ROC-кривою і у відносних значень, цей параметр для даних експерименту виявився на рівні 0,97. Тобто, можна стверджувати, що оцінка якості моделі складає 97% від 100% для ідеального варіанту, коли всі зразки були б точно визначені як такі, що не належать до еталонного голосу. Тобто можна передбачити, що напевно результати під номерами 4,12 та 18 є випадковими і їх поява мало ймовірна при збільшенні часу навчання нейронної мережі за рахунок збільшення кількості записів еталонного голосу на вході цієї мережі.

## ВИСНОВКИ

За підсумками проведеного дослідження можна відмітити наступне.

1. Розроблено нову систему ідентифікації за голосом на основі використання нейронної мережі і проведена її перевірка за 5 окремими критеріями. Отримано за результатами порівняння з еталонним голосом, якому навчалась нейронна мережа, що при впровадженні певних модифікацій голосу, лише для штучно синтезованого комп'ютерного голосу вірогідність схожості з еталонним зразком склала всього лише 0,0112. В інших ситуаціях, отримано що неправильний виголос фрази, чи фраза вимовлена емоційно, чи виголос зроблено при оточуючому шумі, чи сам голос є змінений, внаслідок хвороби, не стали причиною помилкової роботи системи ідентифікації за голосом. Тобто, для даних модифікацій голосу вірогідність схожості з еталонним голосом в середньому виявилась на високому рівні в межах 0,75-0,88.

2. Удосконалено систему розпізнавання фраз, яка може працювати з мовними сигналами, записаними англійською чи українською мовами. Проведена перевірка системи для трьох різних навчальних наборів і з різною архітектурою нейронних мереж (різна кількість прихованих шарів та різні моделі оптимізаторів). Знайдено, що найкращою системою розпізнавання українських слів (для третього набору навчальних слів) серед трьох представлених модифікацій виявилась система з 9 прихованими шарами, яка при навчанні використовувала оптимізатор adam. В цьому випадку, отримано значення функції втрат на рівні  $1,22 \times 10^{-5}$ .

3. Запропоновано модифікований алгоритм дереверберації акустичних сигналів з використанням нейронної мережі. Самі акустичні сигнали попередньо було записано у трьох різних приміщеннях – дві навчальні аудиторії 12 навчального корпусу КПІ ім. Ігоря Сікорського та побутова кімната. Для збільшення варіативності деревербації в навчальних даних для системи проведена аугментація даних. Для навчання цієї мережі обрано набір з 10000 аудіозаписів та збільшено їх кількість за рахунок аугументації даних на

1572 аудіозаписи. Отримано, що при записі тестового семплу українською мовою нейронна мережа, не дивлячись, що навчалась на англomовному датасеті, показала свою ефективність з дереверберації сигналу, особливо це помітно для часових фрагментів в області 2,2 сек та 3,5 сек. Що стосується реверберованого мовного сигналу, то варто відмітити з аналізу, що реверберація призводить до збільшення енергетичної складової сигналу. З іншого боку, така сама тенденція, хоч і не так яскраво, прослідковується і для результуючого сигналу. Особливо, це помітно в діапазоні від 4,5 сек як на спектрограмі, так і на сигналограмі. За результатами експертних опитувань якість отриманих деревербованих сигналів покращується на 1.9 – 2.9 пункти за обраною шкалою оцінок. Також для зашумлених сигналів, на які було накладено реверберацію, спостерігається гірша якість покращення розбірливості для рожевого і білого шуму на 1.1 та 1.2 пункти відповідно за шкалою експертних оцінок. Натомість, отримано трохи кращу якість покращення розбірливості мовного сигналу при адитивному додаванні сірого та коричневого шуму, і в даному експерименті, це покращення за даними експертів склало в середньому на 2.3 та 3 пункти за обраною шкалою експертних оцінок.

4. Розроблена нова система клонування мовних сигналів на основі створення та підключення відразу трьох нейронних мереж. Для навчання цих мереж обрано вибірки з двох відкритих словесних баз даних – LibriSpeech (англійській набір) та Mozilla Common Voice (український набір). Знайдено, що синтезований мовний сигнал для оригінальної англійської фрази залежить від того чи оригінал був записаний носієм або не носієм цієї мови. Схожість синтезованого семплу з оригіналом (варіант англійської мови) серед опитаних становить в середньому 7.6 для носія мови англійської мови та 6.7 для не носія мови, за шкалою оцінок від 0 до 10 на думку групи експертів. Аналіз показав високу схожість з окремими словами оригінального тексту фрази та синтезованого за виключенням того, що нейронні мережі вносять певні зміни в енергетичний спектр сигналу і сам сигнал, в ході отриманих результатів

відтворюється або гучніше або тихіше, ніж оригінал. Знайдено, що система клонування голосу проводить аналіз сигналу на наявність тиші та пауз і за їх наявності вилучає при створенні синтезованого сигналу. Додатково, для перевірки системи сформовано оригінальні семпли українською мовою зі зміненою інтонацією, швидкістю вимовлення слів і знайдено, що при сповільненні вимови також сповільнюється вимова і в генерованому мовному сигналі. За результатами опитувань отримано меншу величину схожості для української мови (середня експертна оцінка дорівнює 6.3) в порівнянні з англійською мовою (середня експертна оцінка дорівнює 7.6). Такі дані для випадку української мови у порівнянні з англійською мовою, вірогідно за все, свідчать, про необхідність навчання системи на більш якісному наборі даних української мови.

5. Проведена оцінка якості розроблених систем шляхом використання методу експертних оцінок – для системи дереверберації сигналів та системи клонування голосу. Для системи ідентифікації за голосом використано дві метрики оцінки якості моделі - ROC та AUC. В останньому випадку метрика AUC склала 0,97 що є високою оцінкою прогнозованої сили моделі системи для розв'язання задачі класифікації за принципом “правильне визначення-хибне визначення”.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Carter J. Neural Networks Beginnings. *Self-published*, 2023. ISBN 979-8392131624.
2. Saito S., Itakura F. Theoretical consideration of the statistical optimum recognition of the spectral density of speech, *J. Acoust. Soc. Japan*. 1967
3. Proakis J. G., Manolakis D. G. Digital Signal Processing: Principles, Algorithms and Applications. *Upper Saddle River, New Jersey*, 1992. 1033 p. ISBN: 0-13-3TM33fl-cl
4. Randall R. B., Peeters B., Antoni J., Manzato S. New cepstral methods of signal pre-processing for operational modal analysis // *Proceedings of ISMA2012*, Leuven, Belgium, September 2012. P. 755–764.
5. Havelock D., Kuwano S., Vorländer M. Handbook of Signal Processing in Acoustics. Vol. 1. *New York: Springer Science & Business Media*, 2008. ISBN 978-0-387-77698-9
6. Sahidullah M., Saha G. Design, analysis and experimental evaluation of block-based transformation in MFCC computation for speaker recognition // *Speech Communication*. 2012. Vol. 54, No. 4. P. 543–565. DOI: 10.1016/j.specom.2011.11.004
7. Борисов Г., Трапезон К. Дослідження особливостей створення електронних систем розпізнавання мови на основі нейронних мереж // *Вчені записки Таврійського національного університету імені В. І. Вернадського*. Серія: Технічні науки. 2022. Т. 33(72), № 5. DOI: <https://doi.org/10.32782/2663-5941/2022.5/57>
8. Elko G. W., Pong A.-T. N. A simple adaptive first-order differential microphone // *Proceedings of IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*. 1995. DOI: 10.1109/ASPAA.1995.482983
9. Wittkop T., Hohmann V. Strategy-selective noise reduction for binaural digital hearing aids // *Speech Communication*. 2003. Vol. 39. DOI: 10.1016/S0167-6393(02)00062-6

10. Kinoshita K., Delcroix M., Gannot S. et al. A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research // *EURASIP Journal on Advances in Signal Processing*. 2016. No. 7. DOI: <https://doi.org/10.1186/s13634-016-0306-6>
11. Lemercier J.-M., Thiemann J., Koning R. et al. A neural network-supported two-stage algorithm for lightweight dereverberation on hearing devices // *EURASIP Journal on Audio, Speech, and Music Processing*. 2023. No. 18. P. 1–12. DOI: <https://doi.org/10.1186/s13636-023-00285-8>
12. Han Z., Ke Y., Li X. et al. Parallel processing of distributed beamforming and multichannel linear prediction for speech denoising and dereverberation in wireless acoustic sensor networks // *Journal of Audio, Speech, and Music Processing*. 2023. No. 25. DOI: <https://doi.org/10.1186/s13636-023-00287-6>.
13. Fritzell B. Inverse filtering // *Journal of Voice*. 1992. Vol. 6, No. 2. P. XX–XX. DOI: [https://doi.org/10.1016/S0892-1997\(05\)80124-9](https://doi.org/10.1016/S0892-1997(05)80124-9)
14. Naylor P. A., Gaubitch N. D. (Eds.). *Speech Dereverberation*. Springer Science & Business Media, 2010
15. Richter J., Welker S., Lemercier L.-M. et al. Speech Enhancement and Dereverberation with Diffusion-based Generative Models // *arXiv:2208.05830*. 2023. P. 1–12. DOI: <https://doi.org/10.48550/arXiv.2208.05830>
16. Berkun R., Cohen I. Microphone array power ratio for quality assessment of reverberated speech // *EURASIP Journal on Advances in Signal Processing*. 2015. No. 49. DOI: <https://doi.org/10.1186/s13634-015-0233-y>.
17. Xu R., Wu R., Ishiwaka Y. et al. Listening to sounds of silence for speech denoising // *arXiv:2010.12013*. 2020. P. 1–7. DOI: <https://doi.org/10.48550/arXiv.2010.12013>
18. Sheeja J. J. C., Sankaragomathi B. Speech dereverberation and source separation using DNN-WPE and LWPR-PCA // *Neural Computing and Applications*. 2023. Vol. 35. P. 7339–7356. DOI: <https://doi.org/10.1007/s00521-022-07884-0>

19. Nercessian S. et al. Speech dereverberation using recurrent neural networks // *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19)*. Birmingham, UK, September 2–6, 2019
20. Oo Z., Wang L., Phapatanaburi K. et al. Phase and reverberation aware DNN for distant-talking speech enhancement // *Multimedia Tools and Applications*. 2018. Vol. 77. P. 18865–18880. DOI: <https://doi.org/10.1007/s11042-018-5686-1>
21. Routray S., Mao Q. A context-aware deep neural network approach for simultaneous speech denoising and dereverberation // *Neural Computing and Applications*. 2022. Vol. 34. P. 9831–9845. DOI: <https://doi.org/10.1007/s00521-022-06968-1>
22. Zheng N., Shi Y., Rong W. et al. Effects of skip connections in CNN-based architectures for speech enhancement // *Journal of Signal Processing Systems*. 2020. Vol. 92. P. 875–884. DOI: <https://doi.org/10.1007/s11265-020-01518-1>
23. Naylor P. A. Speech Dereverberation. *Springer*, London, 2010. 388 p. DOI: <https://doi.org/10.1007/978-1-84996-056-4>
24. Dong H. Y., Lee C. M. Speech intelligibility improvement in noisy reverberant environments based on speech enhancement and inverse filtering // *Journal of Audio, Speech, and Music Processing*. 2018. No. 3. DOI: <https://doi.org/10.1186/s13636-018-0126-8>
25. Ren B., Wang L., Lu L. et al. Combination of bottleneck feature extraction and dereverberation for distant-talking speech recognition // *Multimedia Tools and Applications*. 2016. Vol. 75. P. 5093–5108. DOI: <https://doi.org/10.1007/s11042-015-2849-1>
26. Ernst O., Shlomo E., Gannot S. et al. Speech dereverberation using fully convolutional networks // *arXiv:1803.08243*. 2019. P. 1–5. DOI: <https://doi.org/10.48550/arXiv.1803.08243>
27. Balabin R. M., Lomakina E. I. Neural network approach to quantum-chemistry data: accurate prediction of density functional theory energies // *Journal of Chemical Physics*. 2009. Vol. 131, No. 7. DOI: 10.1063/1.3206326



28. French J. The time traveller's CAPM // *Investment Analysts Journal*. 2017. Vol. 46, No. 2. P. 81–96. DOI: 10.1080/10293523.2016.1255469
29. Ganesan N. Application of neural networks in diagnosing cancer disease using demographic data // *International Journal of Computer Applications*. 2010. Vol. 1, No. 26. P. 81–97. DOI: 10.5120/476-783
30. Ermini L., Catani F., Casagli N. Artificial neural networks applied to landslide susceptibility assessment // *Geomorphology*. 2005. Vol. 66, No. 1–4. P. 327–343. DOI: <https://doi.org/10.1016/j.geomorph.2004.09.025>
31. Govindaraj R. S. Artificial neural networks in hydrology. I: preliminary concepts // *Journal of Hydrologic Engineering*. 2000. Vol. 5, No. 2. P. 115–123. DOI: [https://doi.org/10.1061/\(ASCE\)1084-0699\(2000\)5:2\(115\)](https://doi.org/10.1061/(ASCE)1084-0699(2000)5:2(115))
32. Govindaraj R. S. Artificial neural networks in hydrology. II: hydrologic applications // *Journal of Hydrologic Engineering*. 2000. Vol. 5, No. 2. P. 124–137. DOI: [https://doi.org/10.1061/\(ASCE\)1084-0699\(2000\)5:2\(124\)](https://doi.org/10.1061/(ASCE)1084-0699(2000)5:2(124))
33. Zissis D., Xidias E. K., Lekkas D. A cloud-based architecture capable of perceiving and predicting multiple vessel behaviour // *Applied Soft Computing*. 2015. Vol. 35. P. 652–661. DOI: <https://doi.org/10.1016/j.asoc.2015.07.002>
34. Wang Y., Wu Q. Research on face recognition technology based on PCA and SVM // *Proceedings of the 7th International Conference on Big Data Analytics (ICBDA)*. 2022. P. 248–252. DOI: 10.1109/ICBDA55095.2022.9760320
35. Su Y.-M., Peng H.-W., Huang K.-W., Yang C.-S. Image processing technology for text recognition // *Proceedings of the International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. 2019. P. 1–5. DOI: 10.1109/TAAI48200.2019.8959877
36. Zhang J., Wang B. A novel voice recognition model based on HMM and fuzzy PPM // *Proceedings of the IEEE 10th International Conference on Signal Processing (ICOSP)*. 2010. P. 637–640. DOI: 10.1109/ICOSP.2010.5656855
37. Ali F., Rathor H., Akram W. License plate recognition system // *Proceedings of the International Conference on Advance Computing and Innovative*

- Technologies in Engineering (ICACITE)*. 2021. P. 1053–1055. DOI: 10.1109/ICACITE51222.2021.9404706
38. solanki A., Pandey S. Music instrument recognition using deep convolutional neural networks // *International Journal of Information Technology*. 2022. Vol. 14. P. 1659–1668. DOI: <https://doi.org/10.1007/s41870-019-00285-y>
  39. Chollet F. Deep Learning with Python. Manning Publications Co, 2018. 373 p. ISBN 978-1-61729-443-3
  40. Cireşan D. C., Meier U., Masci J., Gambardella L. M., Schmidhuber J. Flexible, high-performance convolutional neural networks for image classification // *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*. 2011. P. 1237–1242. DOI: 10.5591/978-1-57735-516-8/IJCAI11-210
  41. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks // *Advances in Neural Information Processing Systems*. 2012. No. 25. DOI: 10.1145/3065386
  42. Starner T., Pentland A. Real-time American Sign Language visual recognition from video using hidden Markov models // *Proceedings of the International Symposium on Computer Vision (ISCV)*. Feb. 1995. DOI: 10.1109/ISCV.1995.477012
  43. Борисов Г., Трапезон К. Особливості дереверберації мовних сигналів за допомогою нейронних мереж // *Вісник Кременчуцького національного університету імені Михайла Остроградського*. 2023. Вип. 3 (140). DOI: <https://doi.org/10.32782/1995-0519.2023.3.18>.
  44. Reyes-Diaz F., Hernandez-Sierra G., Calvo de Lara J. DNN and i-vector combined method for speaker recognition on multi-variability environments // *International Journal of Speech Technology*. 2021. Vol. 24. P. 409–418. DOI: <https://doi.org/10.1007/s10772-021-09796-1>
  45. Chakroun R., Frikha M. A deep learning approach for text-independent speaker recognition with short utterances // *Multimedia Tools and*

- Applications*. 2023. Vol. 82. P. 33111–33133. DOI: <https://doi.org/10.1007/s11042-023-14942-9>
46. Борисов Г., Трапезон К. Дослідження особливостей створення текстонезалежних голосових систем доступу із захистом від спуффінг-атак // *Вісник Кременчуцького національного університету імені Михайла Остроградського*. 2024. Вип. 1 (143). DOI: <https://doi.org/10.32782/1995-0519.2024.1.34>
47. Hourri S., Kharroubi J. A deep learning approach for speaker recognition // *International Journal of Speech Technology*. 2020. Vol. 23. P. 123–131. DOI: <https://doi.org/10.1007/s10772-019-09665-y>
48. Mittal A., Dua M. Automatic speaker verification systems and spoof detection techniques: review and analysis // *International Journal of Speech Technology*. 2022. Vol. 25. P. 105–134. DOI: <https://doi.org/10.1007/s10772-021-09876-2>
49. Sun L., Gu T., Xie K. et al. Text-independent speaker identification based on deep Gaussian correlation supervector // *International Journal of Speech Technology*. 2019. Vol. 22. P. 449–457. DOI: <https://doi.org/10.1007/s10772-019-09618-5>
50. Liu H., Zhao L. A speaker verification method based on TDNN–LSTMP // *Circuits, Systems, and Signal Processing*. 2019. Vol. 38. P. 4840–4854. DOI: <https://doi.org/10.1007/s00034-019-01092-3>
51. Al-Karawi K., Mohammed D. Improving short utterance speaker verification by combining MFCC and entropy in noisy conditions // *Multimedia Tools and Applications*. 2021. Vol. 80. P. 22231–22249. DOI: <https://doi.org/10.1007/s11042-021-10767-6>.
52. Reynolds D., Quatieri T. Speaker verification using adapted Gaussian mixture models // *Digital Signal Processing*. 2000. Vol. 10. P. 19–41. DOI: <https://doi.org/10.1006/dspr.1999.0361>
53. Neekhara P., Hussain S., Koushanfar F. Expressive neural voice cloning // *arXiv:2102.00151*. 2021. P. 1–12. DOI: <https://doi.org/10.48550/arXiv.2102.00151>

54. Milewski K., Zaporowski S., Czyzewski A. Comparison of the ability of neural network model and humans to detect a cloned voice // *Electronics*. 2023. Vol. 12, No. 21. Article 4458. DOI: <https://doi.org/10.3390/electronics12214458>
55. Zhang X., Zhang X., Sun M. et al. Imperceptible black-box waveform-level adversarial attack towards automatic speaker recognition // *Complex & Intelligent Systems*. 2023. Vol. 9. P. 65–79. DOI: <https://doi.org/10.1007/s40747-022-00782-x>
56. Xie Y., Li Z., Shi C. et al. Real-time, robust and adaptive universal adversarial attacks against speaker recognition systems // *Journal of Signal Processing Systems*. 2021. Vol. 93. P. 1187–1200. DOI: <https://doi.org/10.1007/s11265-020-01629-9>
57. Mahum R., Irtaza A., Javed A. DeepDet: YAMNet with BottleNeck Attention Module (BAM) for TTS synthesis detection // *Journal of Audio, Speech, and Music Processing*. 2024. No. 18. DOI: <https://doi.org/10.1186/s13636-024-00335-9>
58. Kumar Y., Koul A., Singh C. A deep learning approach in text-to-speech system: a systematic review and recent research perspective // *Multimedia Tools and Applications*. 2023. Vol. 82. P. 15171–15197. DOI: <https://doi.org/10.1007/s11042-022-13943-4>
59. Chen Z., Ai Z., Ma Y. Optimizing feature fusion for improved zero-shot adaptation in text-to-speech synthesis // *Journal of Audio, Speech, and Music Processing*. 2024. No. 28. DOI: <https://doi.org/10.1186/s13636-024-00351-9>
60. Борисов Г., Трапезон К. Підходи та принципи створення системи клонування голосу // *Вісник Кременчуцького національного університету імені Михайла Остроградського*. 2024. Вип. 4 (147). DOI: <https://doi.org/10.32782/1995-0519.2024.4.8>
61. Khochare J., Joshi C., Yenarkar B. A deep learning framework for audio deepfake detection // *Arabian Journal for Science and Engineering*. 2022. Vol. 47. P. 3447–3458. DOI: <https://doi.org/10.1007/s13369-021-06297-w>

62. Dagar D., Vishwakarma D. A literature review and perspectives in deepfakes: generation, detection, and applications // *International Journal of Multimedia Information Retrieval*. 2022. Vol. 11. P. 219–289. DOI: <https://doi.org/10.1007/s13735-022-00241-w>
63. Masood M., Nawaz M., Malik K. Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward // *Applied Intelligence*. 2023. Vol. 53. P. 3974–4026. DOI: <https://doi.org/10.1007/s10489-022-03766-z>
64. Kaur N., Singh P. Conventional and contemporary approaches used in text-to-speech synthesis: a review // *Artificial Intelligence Review*. 2023. Vol. 56. P. 5837–5880. DOI: <https://doi.org/10.1007/s10462-022-10315-0>
65. Zeng Y., Mao H., Peng D. Spectrogram-based multi-task audio classification // *Multimedia Tools and Applications*. 2019. Vol. 78. P. 3705–3722. DOI: <https://doi.org/10.1007/s11042-017-5539-3>
66. Jia Y., Zhang Y., Weiss R. Transfer learning from speaker verification to multispeaker text-to-speech synthesis // *Advances in Neural Information Processing Systems*. 2018. Vol. 31. P. 4485–4495. DOI: <https://doi.org/10.48550/arXiv.1806.04558>
67. Aylett M., Yamagishi J. Combining statistical parametric speech synthesis and unit-selection for automatic voice cloning // *Proc. LangTech 2008*. 2008
68. Variani E., Lei X. Deep neural networks for small footprint text-dependent speaker verification // *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014. P. 4080–4084. DOI: <https://doi.org/10.1109/ICASSP.2014.6854363>
69. Patterson J., Gibson A. Deep learning: a practitioner's approach. Sebastopol, *O'Reilly Media, Inc.*, 2017. 530 p
70. Shahin I., Nassif A. B., Hindawi N. Speaker identification in stressful talking environments based on convolutional neural network // *International Journal of Speech Technology*. 2021. Vol. 24. P. 1055–1066. DOI: <https://doi.org/10.1007/s10772-021-09869-1>

71. Gulli A., Pal S. Deep learning with Keras. Birmingham, *Packt Publishing*, 2017. 318 p
72. Di W., Bhardwaj A., Wei J. Deep learning essentials. *Packt Publishing Ltd*, 2018. ISBN 978-1-78588-036-0.
73. Rumelhart D. E., Hinton G. E., Williams R. J. Learning representations by back-propagating errors // *Nature*. 1986. Vol. 323, No. 6088. P. 533–536. DOI: <https://doi.org/10.1038/323533a0>
74. Lan R., Zou H., Pang C. et al. Image denoising via deep residual convolutional neural networks // *Signal, Image and Video Processing*. 2021. Vol. 15. P. 1–8. DOI: <https://doi.org/10.1007/s11760-019-01537-x>
75. Лавриненко О., Бахтіяров Д., Конахович Г., Курушкін В. Аналіз ефективності системи голосової ідентифікації на основі MFCC та GMM-SVM за умов впливу завад у каналі зв'язку // *Наукоємні технології*. 2023. Т. 59, № 3. DOI: <https://doi.org/10.18372/2310-5461.59.17950>
76. Ernst O., Chazan S. E., Gannot S., Goldberger J. Speech dereverberation using fully convolutional networks // *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*. 2018. P. 390–394. DOI: [10.23919/EUSIPCO.2018.8553141](https://doi.org/10.23919/EUSIPCO.2018.8553141)
77. Jia Y., Zhang Y., Weiss R. J., Wang Q., Shen J., Ren F., Chen Z., Nguyen P., Pang R., Lopez-Moreno I., Wu Y. Transfer learning from speaker verification to multispeaker text-to-speech synthesis // *arXiv:1806.04558*. 2018. DOI: <https://doi.org/10.48550/arXiv.1806.04558>
78. Xiao X., Zhao S., Nguyen D. H. H., Zhong X., Jones D. L., Chng E. S. Speech dereverberation for enhancement and recognition using dynamic features constrained deep neural networks and feature adaptation // *EURASIP Journal on Advances in Signal Processing*. 2016. DOI: <https://doi.org/10.1186/s13634-015-0300-4>
79. Tang J. Deep learning-based speaker identification system // *Proceedings of the 2024 Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence (DEAI '24)*. New

- York: Association for Computing Machinery, 2024. P. 744–748. DOI: <https://doi.org/10.1145/3675417.367554>
80. Lukic Y., Vogt C., Dürr O., Stadelmann T. Speaker identification and clustering using convolutional neural networks // *Proceedings of the 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. 2016. DOI: 10.1109/MLSP.2016.7738816
  81. Venayagamoorthy G. K., Sandrasegaran K. Voice recognition using neural networks // *Proceedings of the 1998 South African Symposium on Communications and Signal Processing (COMSIG '98)*. 1998. DOI: 10.1109/COMSIG.1998.736916
  82. Rusiecki A. Trimmed categorical cross-entropy for deep learning with label noise // *Electronics Letters*. 2019. Vol. 55, No. 6. DOI: <https://doi.org/10.1049/el.2018.7980>
  83. Andrychowicz M., Denil M., Gomez S., Hoffman M. W., Pfau D., Schaul T., Shillingford B., de Freitas N. Learning to learn by gradient descent by gradient descent // *Advances in Neural Information Processing Systems*. 2016. DOI: 10.48550/arXiv.1606.04474
  84. Noisy speech database for training speech enhancement algorithms and TTS models, URL: <https://datashare.ed.ac.uk/handle/10283/2791> (дата звернення: 10.02.2023)
  85. Daniel M. O., Olajide I. A. Design of a voice recognition system using artificial neural network // *International Journal of Electrical and Computer Engineering Research*. 2024. DOI: <https://doi.org/10.53375/ijecer.2024.371>
  86. Zhang N. Oral voice recognition system based on deep neural network posteriori probability algorithm // *Procedia Computer Science*. 2024. Vol. 243. P. 213–223. DOI: <https://doi.org/10.1016/j.procs.2024.09.028>
  87. Дворник О., Продеус А., Дідковська М., Моторнюк Д. Апаратно-програмний комплекс «Штучна голова». Частина 2. Оцінювання розбірливості мови в аудиторіях // *Microsystems, Electronics and Acoustics*.

2020. Т. 22, № 3. С. 48–55. DOI: <https://doi.org/10.20535/2523-4455.me.209928>
88. Reverberant speech database for training speech dereverberation algorithms and TTS models, URL: <https://datashare.ed.ac.uk/handle/10283/2031> (дата звернення: 10.02.2023)
89. Костючок Ю. С., Мартинович Л. С., Моторнюк Д. Є., Нечитайло В. О., Храпачевський О. В., Продеус А. М. Акустична паспортизація навчальних приміщень // *Електроніка та зв'язок*. 2016. Т. 21, № 2(91). С. 63–70.
90. Ющук І. П. Практичний довідник з українського правопису. 2-ге вид., доопрац. 2020. 128 с. ISBN 978-966-983-115-6
91. Mozilla Common Voice, URL: <https://commonvoice.mozilla.org/uk/datasets> (дата звернення: 17.08.2024)
92. Corbacioglu S., Akcel G. Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value // *Turk J Emerg Med*. 2023. Vol. 23, № 4. P. 195–198. DOI: [https://doi.org/10.4103/tjem.tjem\\_182\\_23](https://doi.org/10.4103/tjem.tjem_182_23)



## ДОДАТОК А

### Програмний код системи ідентифікації за голосом

```
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout
from keras.regularizers import l2

model = Sequential()
model.add(LSTM(units=256, return_sequences=True, kernel_regularizer=l2(0.001)))
model.add(Dropout(0.3))
model.add(LSTM(units=128, return_sequences=True, kernel_regularizer=l2(0.001)))
model.add(Dropout(0.3))
model.add(LSTM(units=64, return_sequences=True, kernel_regularizer=l2(0.001)))
model.add(Dropout(0.3))
model.add(LSTM(units=32))

model.add(Dense(units=128, activation='relu', kernel_regularizer=l2(0.001)))
model.add(Dropout(0.3))
model.add(Dense(units=64, activation='relu', kernel_regularizer=l2(0.001)))
model.add(Dropout(0.3))
model.add(Dense(units=32, activation='relu', kernel_regularizer=l2(0.001)))
model.add(Dropout(0.3))
model.add(Dense(units=2, activation='softmax'))
```

## ДОДАТОК Б

### Програмний код системи покращення розбірливості мови

```

inputLayer = [
    imageInputLayer([256, 256, 1], "Name", "input", "Normalization", "none")
    convolution2dLayer([6 6], 64, "Name", "conv1", "Padding", "same", "Stride", [2 2])
    leakyReluLayer(0.2, "Name", "leaky-relu1")];

layers = inputLayer;

layers = [layers;
    convolution2dLayer([6, 6], 128, "Name", "conv2", "Padding", "same", "Stride", [2 2])
    batchNormalizationLayer("Name", "batchnorm2")
    leakyReluLayer(0.2, "Name", "leaky-relu2")];

layers = [layers;
    convolution2dLayer([6, 6], 256, "Name", "conv3", "Padding", "same", "Stride", [2 2])
    batchNormalizationLayer("Name", "batchnorm3")
    leakyReluLayer(0.2, "Name", "leaky-relu3")];

layers = [layers;
    convolution2dLayer([6, 6], 512, "Name", "conv4", "Padding", "same", "Stride", [2 2])
    batchNormalizationLayer("Name", "batchnorm4")
    leakyReluLayer(0.2, "Name", "leaky-relu4")];

layers = [layers;
    convolution2dLayer([6, 6], 512, "Name", "conv5", "Padding", "same", "Stride", [2 2])
    batchNormalizationLayer("Name", "batchnorm5")
    leakyReluLayer(0.2, "Name", "leaky-relu5")];

layers = [layers;
    convolution2dLayer([6, 6], 512, "Name", "conv6", "Padding", "same", "Stride", [2 2])
    batchNormalizationLayer("Name", "batchnorm6")
    leakyReluLayer(0.2, "Name", "leaky-relu6")];

layers = [layers;
    convolution2dLayer([6, 6], 512, "Name", "conv7", "Padding", "same", "Stride", [2 2])
    batchNormalizationLayer("Name", "batchnorm7")
    leakyReluLayer(0.2, "Name", "leaky-relu7")];

layers = [layers;
    convolution2dLayer([6, 6], 512, "Name", "conv8", "Padding", "same", "Stride", [2 2])
    batchNormalizationLayer("Name", "batchnorm8")
    reluLayer("Name", "relu8")
    transposedConv2dLayer([6, 6], 512, "Name", "deconv7", "Cropping", "same", "Stride", [2 2])
    batchNormalizationLayer("Name", "de-batchnorm7")
    dropoutLayer(0.5, "Name", "de-dropout7")
    reluLayer("Name", "de-relu7")];

```

```

layers = [layers;
    concatenationLayer(3,2,"Name","concat7")
    transposedConv2dLayer([6, 6],512,"Name","deconv6","Cropping","same","Stride",[2 2])
    batchNormalizationLayer("Name","de-batchnorm6")
    dropoutLayer(0.5,"Name","de-dropout6")
    reluLayer("Name","de-relu6")];

layers = [layers;
    concatenationLayer(3,2,"Name","concat6")
    transposedConv2dLayer([6, 6],512,"Name","deconv5","Cropping","same","Stride",[2 2])
    batchNormalizationLayer("Name","de-batchnorm5")
    dropoutLayer(0.5,"Name","de-dropout5")
    reluLayer("Name","de-relu5")];

layers = [layers;
    concatenationLayer(3,2,"Name","concat5")
    transposedConv2dLayer([6, 6],512,"Name","deconv4","Cropping","same","Stride",[2 2])
    batchNormalizationLayer("Name","de-batchnorm4")
    reluLayer("Name","de-relu4")];

layers = [layers;
    concatenationLayer(3,2,"Name","concat4")
    transposedConv2dLayer([6, 6],256,"Name","deconv3","Cropping","same","Stride",[2 2])
    batchNormalizationLayer("Name","de-batchnorm3")
    reluLayer("Name","de-relu3")];

layers = [layers;
    concatenationLayer(3,2,"Name","concat3")
    transposedConv2dLayer([6, 6],128,"Name","deconv2","Cropping","same","Stride",[2 2])
    batchNormalizationLayer("Name","de-batchnorm2")
    reluLayer("Name","de-relu2")];
layers = [layers;
    concatenationLayer(3,2,"Name","concat2")
    transposedConv2dLayer([6, 6],64,"Name","deconv1","Cropping","same","Stride",[2 2])
    batchNormalizationLayer("Name","de-batchnorm1")
    reluLayer("Name","de-relu1")];

layers = [layers;
    concatenationLayer(3,2,"Name","concat1")
    transposedConv2dLayer([6, 6],1,"Name","deconv0","Cropping","same","Stride",[2 2])
    tanhLayer("Name","de-tanh0")];

layers = [layers;
    regressionLayer('Name','output')];

unetLayerGraph = layerGraph(layers);

unetLayerGraph = connectLayers(unetLayerGraph,"leaky-relu1","concat1/in2");
unetLayerGraph = connectLayers(unetLayerGraph,"leaky-relu2","concat2/in2");
unetLayerGraph = connectLayers(unetLayerGraph,"leaky-relu3","concat3/in2");
unetLayerGraph = connectLayers(unetLayerGraph,"leaky-relu4","concat4/in2");
unetLayerGraph = connectLayers(unetLayerGraph,"leaky-relu5","concat5/in2");
unetLayerGraph = connectLayers(unetLayerGraph,"leaky-relu6","concat6/in2");
unetLayerGraph = connectLayers(unetLayerGraph,"leaky-relu7","concat7/in2");

```

## ДОДАТОК В

### Програмний код системи клонування голосу

#### Енкодер:

```
import torch
import torch.nn as nn
import numpy as np
from torch.nn.utils import clip_grad_norm_
from scipy.interpolate import interp1d
from scipy.optimize import brentq
from sklearn.metrics import roc_curve

class Encoder(nn.Module):
    def __init__(self, compute_device, loss_compute_device):
        super().__init__()
        self.loss_compute_device = loss_compute_device

        self.recurrent_block = nn.ModuleList([
            nn.LSTM(input_size=40, hidden_size=256, batch_first=True).to(compute_device),
            nn.LSTM(input_size=256, hidden_size=256, batch_first=True).to(compute_device),
            nn.LSTM(input_size=256, hidden_size=256, batch_first=True).to(compute_device)
        ])

        self.dense_layer = nn.Linear(in_features=256, out_features=256).to(compute_device)
        self.activation = nn.ReLU().to(compute_device)

        self.similarity_scale = nn.Parameter(torch.tensor([10.])).to(loss_compute_device)
        self.similarity_offset = nn.Parameter(torch.tensor([-5.])).to(loss_compute_device)

        self.loss_function = nn.CrossEntropyLoss().to(loss_compute_device)

    def adjust_gradients(self):
        self.similarity_scale.grad *= 0.01
        self.similarity_offset.grad *= 0.01
        clip_grad_norm_(self.parameters(), max_norm=3, norm_type=2)

    def forward(self, input_sequences, init_hidden_state=None):
        output = input_sequences
        for lstm_layer in self.recurrent_block:
            output, (hidden_state, cell_state) = lstm_layer(output, init_hidden_state)

        feature_vector = self.activation(self.dense_layer(hidden_state[-1]))
        normalized_vector = feature_vector / (torch.norm(feature_vector, dim=1, keepdim=True) + 1e-5)
        return normalized_vector
```

```

def compute_similarity(self, embeddings):
    num_speakers, num_samples = embeddings.shape[:2]

    avg_centroids = torch.mean(embeddings, dim=1, keepdim=True)
    avg_centroids = avg_centroids.clone() / (torch.norm(avg_centroids, dim=2, keepdim=True) + 1e-5)

    excl_centroids = (torch.sum(embeddings, dim=1, keepdim=True) - embeddings) / (num_samples - 1)
    excl_centroids = excl_centroids.clone() / (torch.norm(excl_centroids, dim=2, keepdim=True) + 1e-5)

    similarity_mat = torch.zeros(num_speakers, num_samples, num_speakers).to(self.loss_compute_device)
    exclusion_mask = 1 - np.eye(num_speakers, dtype=np.int32)

    for i in range(num_speakers):
        excluded_indices = np.where(exclusion_mask[i])[0]
        similarity_mat[excluded_indices, :, i] = (embeddings[excluded_indices] * avg_centroids[i]).sum(dim=2)
        similarity_mat[i, :, i] = (embeddings[i] * excl_centroids[i]).sum(dim=1)

    similarity_mat = similarity_mat * self.similarity_scale + self.similarity_offset
    return similarity_mat

def compute_loss(self, embeddings):
    num_speakers, num_samples = embeddings.shape[:2]
    similarity_scores = self.compute_similarity(embeddings)
    similarity_scores = similarity_scores.view((num_speakers * num_samples, num_speakers))

    expected_labels = np.repeat(np.arange(num_speakers), num_samples)
    target_labels = torch.from_numpy(expected_labels).long().to(self.loss_compute_device)
    loss_value = self.loss_function(similarity_scores, target_labels)

    with torch.no_grad():
        convert_labels = lambda idx: np.eye(1, num_speakers, idx, dtype=np.int32)[0]
        label_vectors = np.array([convert_labels(idx) for idx in expected_labels])
        predictions = similarity_scores.detach().cpu().numpy()

        false_pos_rate, true_pos_rate, _ = roc_curve(label_vectors.flatten(), predictions.flatten())
        eer_value = brentq(lambda x: 1. - x - interp1d(false_pos_rate, true_pos_rate)(x), 0., 1.)

    return loss_value, eer_value

```

## Вокодер:

```

class Vocoder(nn.Module):
    def __init__(self, hidden_dim=896, quant_levels=256):
        super(Vocoder, self).__init__()

        self.hidden_dim = hidden_dim
        self.half_hidden = hidden_dim // 2

        self.hidden_transform = nn.Linear(self.hidden_dim, 3 * self.hidden_dim, bias=False)

        self.fc1 = nn.Linear(self.half_hidden, self.half_hidden)
        self.fc2 = nn.Linear(self.half_hidden, quant_levels)
        self.fc3 = nn.Linear(self.half_hidden, self.half_hidden)
        self.fc4 = nn.Linear(self.half_hidden, quant_levels)

        self.input_proj_coarse = nn.Linear(2, 3 * self.half_hidden, bias=False)
        self.input_proj_fine = nn.Linear(3, 3 * self.half_hidden, bias=False)

        self.bias_gate = nn.Parameter(torch.zeros(self.hidden_dim))
        self.bias_reset = nn.Parameter(torch.zeros(self.hidden_dim))
        self.bias_candidate = nn.Parameter(torch.zeros(self.hidden_dim))

    def forward(self, previous_sample, previous_state, coarse_sample):
        transformed_hidden = self.hidden_transform(previous_state)
        gate_hidden, reset_hidden, candidate_hidden = torch.split(transformed_hidden, self.hidden_dim, dim=1)

        coarse_transformed = self.input_proj_coarse(previous_sample)
        gate_coarse, reset_coarse, candidate_coarse = torch.split(coarse_transformed, self.half_hidden, dim=1)

        fine_input = torch.cat([previous_sample, coarse_sample], dim=1)
        fine_transformed = self.input_proj_fine(fine_input)
        gate_fine, reset_fine, candidate_fine = torch.split(fine_transformed, self.half_hidden, dim=1)

        gate_combined = torch.cat([gate_coarse, gate_fine], dim=1)
        reset_combined = torch.cat([reset_coarse, reset_fine], dim=1)
        candidate_combined = torch.cat([candidate_coarse, candidate_fine], dim=1)

        update_gate = torch.sigmoid(gate_hidden + gate_combined + self.bias_gate)
        reset_gate = torch.sigmoid(reset_hidden + reset_combined + self.bias_reset)
        candidate_state = torch.tanh(reset_gate * candidate_hidden + candidate_combined + self.bias_candidate)

        new_hidden_state = update_gate * previous_state + (1. - update_gate) * candidate_state
        coarse_part, fine_part = torch.split(new_hidden_state, self.half_hidden, dim=1)

        coarse_output = self.fc2(F.relu(self.fc1(coarse_part)))
        fine_output = self.fc4(F.relu(self.fc3(fine_part)))

        return coarse_output, fine_output, new_hidden_state

    def initialize_hidden(self, batch_size=1):
        return torch.zeros(batch_size, self.hidden_dim).cuda()

```

## Синтезатор:

```

import numpy as np
import torch
import torch.nn as nn
import torch.nn.functional as F

class Highway(nn.Module):
    def __init__(self, size):
        super().__init__()
        self.gate_layer = nn.Linear(size, size)
        self.transformation_layer = nn.Linear(size, size)
        self.transformation_layer.bias.data.fill_(0.)

    def forward(self, x):
        transformed = self.transformation_layer(x)
        gate = torch.sigmoid(self.gate_layer(x))
        output = gate * F.relu(transformed) + (1 - gate) * x
        return output

class Encoder(nn.Module):
    def __init__(self, embedding_dim, vocab_size, hidden_dim, kernel_size, num_highways, dropout_rate):
        super().__init__()
        self.embedding_layer = nn.Embedding(vocab_size, embedding_dim)
        self.preprocessor = PreProcessor(embedding_dim, hidden_dim, dropout_rate)
        self.feature_extractor = FeatureProcessingUnit(kernel_size, hidden_dim, num_highways)

    def forward(self, x, speaker_embed=None):
        x = self.embedding_layer(x)
        x = self.preprocessor(x)
        x = x.transpose(1, 2)
        x = self.feature_extractor(x)
        if speaker_embed is not None:
            x = self.combine_with_speaker_embedding(x, speaker_embed)
        return x

    def combine_with_speaker_embedding(self, x, speaker_embed):
        batch_size, seq_length = x.shape[:2]
        speaker_embed = speaker_embed.repeat_interleave(seq_length, dim=1).view(batch_size, seq_length, -1)
        x = torch.cat((x, speaker_embed), dim=2)
        return x

class NormConv(nn.Module):
    def __init__(self, in_channels, out_channels, kernel_size, activation=True):
        super().__init__()
        self.conv_layer = nn.Conv1d(in_channels, out_channels, kernel_size, padding=kernel_size // 2, bias=False)
        self.norm_layer = nn.BatchNorm1d(out_channels)
        self.activation = activation

    def forward(self, x):
        x = self.conv_layer(x)
        x = F.relu(x) if self.activation else x
        return self.norm_layer(x)

```



```

class FeatureProcessingUnit(nn.Module):
    def __init__(self, kernel_size, channels, num_highways):
        super().__init__()
        self.conv_layers = nn.ModuleList([NormConv(channels, channels, k) for k in range(1, kernel_size + 1)])
        self.maxpool = nn.MaxPool1d(2, stride=1, padding=1)
        self.highways = nn.ModuleList([Highway(channels) for _ in range(num_highways)])
        self.recurrent_layer = nn.GRU(channels, channels // 2, batch_first=True, bidirectional=True)

    def forward(self, x):
        residual = x
        conv_results = [conv(x) for conv in self.conv_layers]
        x = torch.cat(conv_results, dim=1)
        x = self.maxpool(x)
        x += residual
        x = x.transpose(1, 2)
        for highway in self.highways:
            x = highway(x)
        x, _ = self.recurrent_layer(x)
        return x

class PreProcessor(nn.Module):
    def __init__(self, input_dim, hidden_dim, dropout_rate):
        super().__init__()
        self.fc1 = nn.Linear(input_dim, hidden_dim)
        self.fc2 = nn.Linear(hidden_dim, hidden_dim // 2)
        self.dropout_rate = dropout_rate

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = F.dropout(x, self.dropout_rate, training=True)
        x = F.relu(self.fc2(x))
        x = F.dropout(x, self.dropout_rate, training=True)
        return x

class NeuralAttention(nn.Module):
    def __init__(self, attn_dim):
        super().__init__()
        self.key_transform = nn.Linear(attn_dim, attn_dim, bias=False)
        self.score_layer = nn.Linear(attn_dim, 1, bias=False)

    def forward(self, encoder_outputs, query):
        transformed_query = self.key_transform(query).unsqueeze(1)
        scores = self.score_layer(torch.tanh(encoder_outputs + transformed_query))
        return F.softmax(scores, dim=1).transpose(1, 2)

class Decoder(nn.Module):
    def __init__(self, mel_dim, encoder_dim, decoder_dim, rnn_dim, dropout_rate, speaker_embed_size):
        super().__init__()
        self.mel_dim = mel_dim
        self.prenet = PreProcessor(mel_dim, decoder_dim * 2, dropout_rate)
        self.attention_layer = NeuralAttention(decoder_dim)
        self.decoder_rnn = nn.GRUCell(encoder_dim + decoder_dim * 2 + speaker_embed_size, decoder_dim)
        self.rnn_input = nn.Linear(encoder_dim + decoder_dim + speaker_embed_size, rnn_dim)
        self.rnn_layers = [nn.LSTMCell(rnn_dim, rnn_dim) for _ in range(2)]
        self.mel_projection = nn.Linear(rnn_dim, mel_dim, bias=False)

```



```

def forward(self, encoder_outputs, prenet_input, hidden_states, context, t):
    prenet_out = self.prenet(prenet_input)
    attn_rnn_input = torch.cat([context, prenet_out], dim=-1)
    hidden_states[0] = self.decoder_rnn(attn_rnn_input.squeeze(1), hidden_states[0])
    attn_scores = self.attention_layer(encoder_outputs, hidden_states[0])
    context = attn_scores @ encoder_outputs
    context = context.squeeze(1)
    rnn_input = torch.cat([context, hidden_states[0]], dim=1)
    rnn_input = self.rnn_input(rnn_input)
    for i in range(len(self.rnn_layers)):
        hidden_states[i + 1], _ = self.rnn_layers[i](rnn_input, hidden_states[i + 1])
    mel_output = self.mel_projection(hidden_states[-1])
    return mel_output, attn_scores, hidden_states, context

class Tacotron(nn.Module):
    def __init__(self, embed_dims, vocab_size, encoder_dims, decoder_dims, mel_dim,
                  fft_bins, num_highways, dropout, speaker_embed_size):
        super().__init__()
        self.encoder = Encoder(embed_dims, vocab_size, encoder_dims, 16, num_highways, dropout)
        self.decoder = Decoder(mel_dim, encoder_dims, decoder_dims, decoder_dims, dropout, speaker_embed_size)

    def forward(self, x, mels, speaker_embedding):
        encoder_outputs = self.encoder(x, speaker_embedding)
        prenet_input = torch.zeros(mels.shape[0], mels.shape[1], device=mels.device)
        hidden_states = [torch.zeros_like(prenet_input) for _ in range(3)]
        context = torch.zeros_like(prenet_input)
        mel_outputs = []
        for t in range(mels.shape[2]):
            mel_out, attn_scores, hidden_states, context = self.decoder(encoder_outputs, prenet_input,
                                                                        hidden_states, context, t)
            mel_outputs.append(mel_out)
        return torch.cat(mel_outputs, dim=2)

```