

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації та управління

До захисту допущено:

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ _____ ” _____ 2020 р.

Дипломний проєкт
на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: *Комплекс задач з аналізу фонокардіографічного сигналу*

Виконав : студент IV курсу, групи ІС-61

_____ Бусов Ілля Олександрович _____
(прізвище, ім'я, по батькові) (підпис)

Керівник доц. к.ф.-м.н. Гавриленко Олена Валеріївна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

**Консультант з
графічної
документації**

_____ доц., к.т.н., доц. Телишева Тамара Олексіївна _____
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент

_____ ст. викл. каф. ТК, к.т.н. Солдатова М.О. _____
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____
(підпис)

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки

(повна назва)

Кафедра автоматизованих систем обробки інформації та управління

(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ

(підпис) (вл.ім'я, прізвище)

“ ” _____ 2020 р.

**ЗАВДАННЯ
на дипломний проєкт студенту**

Бусову Іллі Олександровичу

(прізвище, ім'я, по батькові)

1. Тема проєкту *«Комплекс задач з аналізу фонокардіографічного сигналу»*

керівник проєкту Гавриленко Олена Валеріївна, доцент, к.т.н

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7” травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01” червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. *Схема структурна класів програмного забезпечення*

2. *Схема структурна діаграми послідовності*

3. *Схема структурна діаграми розгортання*

4. *Рішення з математичного забезпечення*

5. *Креслення вигляду екранних форм*

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>01.03.2020</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>15.03.2020</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>17.03.2020</i>	
4.	<i>Розробка інформаційного забезпечення</i>	<i>15.04.2020</i>	
5.	<i>Алгоритмізація задачі</i>	<i>31.03.2020</i>	
6.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>15.04.2020</i>	
7.	<i>Розробка програмного забезпечення</i>	<i>30.04.2020</i>	
8.	<i>Налагодження програми</i>	<i>05.05.2020</i>	
9.	<i>Виконання графічних документів</i>	<i>10.05.2020</i>	
10.	<i>Оформлення пояснювальної записки</i>	<i>30.05.2020</i>	
11.	<i>Подання ДП на попередній захист</i>	<i>15.05.2020</i>	
12.	<i>Подання ДП на основний захист</i>	<i>01.06.2020</i>	
13.	<i>Подання ДП рецензенту</i>	<i>02.06.2020</i>	

Студент

Ілля БУСОВ

Керівник

Олена ГАВРИЛЕНКО

Пояснювальна записка до дипломного проєкту

на тему: Комплекс задач з аналізу фонокардіографічного сигналу

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з шести розділів, містить 17 рисунків, 5 таблиць, 1 додаток, 12 джерел.

Дипломний проект присвячений розробці комплексу задач з аналізу фонокардіографічного сигналу. Метою роботи є дослідження та покращення існуючих алгоритмів сегментації та класифікації фонокардіограми для виявлення аномалій у роботі серця.

У розділі інформаційного забезпечення було описано вхідні та вихідні дані програми. Описані набори даних, які використовувалися для навчання моделей

Розділ математичного забезпечення присвячений постановці задачі, опису методів, які були використані для розв'язку задачі, а також обґрунтуванню доцільності їх використання.

В розділі програмне забезпечення описані засоби для розробки програмного продукту, вимоги до технічного забезпечення та архітектура програми.

У технологічному розділі описано керівництво користувача та наведено результати проведених тестів з програмним забезпеченням

КЛЮЧОВІ СЛОВА: обробка сигналів, фонокардіограма, машинне навчання, нейронні мережі, приховані марковські моделі

					ДП ІС-6105.00.1081-с.ПЗ		
		<i>Прізвище</i>	<i>Підпис</i>				
<i>Розроб.</i>	Бусов І.О.			Комплекс задач з аналізу фонокардіографічного сигналу	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Перевірив.</i>	Гавриленко О.В.					2	
<i>Н. кон.</i>	Телишева Т.О.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-		
<i>Затв.</i>	Павлов О.А.				61		

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of six sections, contains 17 drawings, 5 tables, 1 application, 12 sources.

The diploma project is devoted to the development of a set of tasks for the analysis of phonocardiographic signal. The aim of the work is to study and improve the existing algorithms for segmentation and classification of the phonocardiogram to detect abnormalities in the heart.

The information and input data of the program were described in the information support section. Describes the data sets used to train the models

The section of mathematical support is devoted to the formulation of the problem, the description of the methods that were used to solve the problem, as well as the justification of their feasibility.

The software section describes software development tools, hardware requirements, and program architecture.

The technological section describes the user manual and presents the results of tests performed with the software

KEY WORDS: signal processing, phonocardiogram, machine learning, neural networks, hidden Markov models

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП.....	5
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	6
1.1 Опис предметного середовища	6
1.1.1 Опис процесу діяльності	8
1.1.2 Опис функціональної моделі	8
1.2 Огляд наявних аналогів	10
1.3 Постановка задачі	13
1.3.1 Призначення розробки	13
1.3.2 Цілі та задачі розробки	13
Висновки до розділу.....	13
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	15
2.1 Вхідні дані.....	15
2.2 Вихідні дані.....	15
2.3 Структура масивів інформації	16
2.4 Висновки до розділу	16
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	18
3.1 Змістовна постановка задачі	18
3.2 Математична постановка задачі.....	18
3.3 Опис методів розв'язання.....	19
3.4 Обґрунтування методу розв'язання	26
Висновки до розділу.....	28
4 Програмне та технічне забезпечення.....	30
4.1 Засоби розробки.....	30

4.2	Вимоги до технічного забезпечення	31
4.3	Архітектура програмного забезпечення	32
4.3.1	Діаграма послідовності	32
4.3.2	Діаграма класів	32
4.3.3	Діаграма розгортання	32
4.3.4	Специфікація функцій	33
	Висновки до розділу	36
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ	37
5.1	Керівництво користувача	37
5.2	Випробування програмного продукту	39
5.2.1	Мета випробувань	39
5.2.2	Загальні положення	39
5.2.3	Результати випробувань	39
	Висновки до розділу	41
	ЗАГАЛЬНІ ВИСНОВКИ	42
	ПЕРЕЛІК ПОСИЛАНЬ	44
	ДОДАТОК А	46

ВСТУП

Дипломний проект присвячений розробці комплексу задач з аналізу фонокардіографічного сигналу. Фонокардіографічний сигнал - це запис звуків та шумів, які виникають внаслідок серцевої активності.

Хвороби серця є найрозповсюдженішою причиною смертності у світі на даний момент, а їх рання діагностика є найважливішим фактором у подальшому лікуванні хвороби. Фонокардіограма є простим та ефективним способом діагностики хвороб серця, бо порушення тонів серця та шуми можуть бути наслідком відхилень у роботі цього органу. Зазвичай лікарі отримують фонокардіограму прослуховуючи серце за допомогою стетоскопа або можуть записати сигнал цифровим стетоскопом для подальшого візуального та звукового аналізу. Поява цифрових стетоскопів та мобільних девайсів дає можливість для високоякісного запису звуків серця. Вже розроблені стетоскопи, які можуть приєднуватися до смартфона або іншого портативного електронного девайсу та записувати сигнал фонокардіограми. Тому запис фонокардіограми став доступний звичайній людині, що є безперечною перевагою перед іншими способами моніторингу серцевої діяльності. Таким чином постає задача аналізу зібраних даних. Для того щоб користувач міг отримати швидкий висновок з аналізу фонокардіограми одразу після її запису не очікуючи прийому лікаря, необхідно розробити алгоритм, який би автоматично визначав наявність аномалій у роботі серця лише за вхідним сигналом фонокардіограми.

Використання комп'ютерних алгоритмів для аналізу сигналу фонокардіограми може максимально пришвидшити процес виявлень патологій серця в людини. Проте ніякий автоматизований аналіз медичних даних не може давати надійні висновки про стан здоров'я користувача, а лише є причиною для звернення до спеціаліста та підґрунтям для подальшого аналізу експертом предметної області. Так і у випадку з розробленим алгоритмом аналізу фонокардіограми, він не дає чіткого визначення діагнозу для пацієнта, проте за наявності аномалій у роботі серця, алгоритм фіксує їх та сповіщає про це

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

користувачу, що є підставою для звернення до лікаря для більш детального аналізу.

Практичне значення одержаних результатів. Вдосконалено алгоритми сегментації та класифікації сигналу фонокардіограми

Публікації. Результати роботи були опубліковані у тезах доповідей на Всеукраїнських науково-практичних конференціях ІСТУ-2019-осінь та ІСТУ-2020-весна [1-2].

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Під час проведення фонокардіографії проводиться дослідження серця, а саме звуків (тони та шуми), які виникають у процесі роботи серця. Основними перевагами цього фонокардіографії є її неінвазивність, легкість проведення дослідження та можливість неодноразового застосування впродовж будь-якого часу. Цей метод дослідження важливий для виявлення вад у роботі клапанів. Фонокардіограма знімається за допомогою стетоскопу, а сам процес прослуховування називається аускультациєю.

Отож фонокардіографічний сигнал можна аналізувати чи під час прослуховування пацієнта, чи після прослуховування, використовуючи записаний сигнал. Сигнал фонокардіограми відображає активність серця, тому у цьому сигналі можна виділити особливі звуки, які асоціюються з певними подіями серцевого циклу.

Серце - це орган тіла людини, який за допомогою ритмічних скорочень забезпечує ток крові по судинах. Серце складається з чотирьох камер, а саме з двох передсердь та двох шлуночків. Передсердя - це відділ серця, куди поступає кров з судин, а шлуночки - відділи серця, куди потрапляє з передсердь потрапляє кров, а після цього проштовхується до артерій. Між передсердями і шлуночками та між шлуночками і артеріями розташовані клапани, які запобігають

					ДП ІС-6105.00.1081-с.ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

оберненому току крові. Серцевий цикл це послідовність процесів, які відбуваються у роботі серця впродовж одного скорочення та розслаблення. Кожен цикл поділяється на три фази: систола передсердь, систола шлуночків та діастола. Під час систоли серця скорочується для того щоб проштовхнути кров, а під час діастоли серце розслабляється. Цикл починається з систоли шлуночків, м'язи серця скорочуються та проштовхують кров у артеріальне русло. Під час цієї фази відкриваються аортальний та легеневий клапани для випускання крові з серця. Після цього йде фаза діастоли, м'язи серця розслабляються, легеневий та аортальний клапани закриваються, натомість відкриваються передсердно-шлуночкові клапани і через різницю тиску кров потрапляє з передсердь до шлуночків. Остання та найкоротша фаза циклу це систола передсердь, яка завершує перекачку крові до шлуночків. В результаті руху крові під час вищеприписаного процесу і відкриття та закриття клапанів, виникають характерні звуки, які і є об'єктом дослідження фонокардіографії.

Виділяються два основні тони(звуки) серця, які називаються S1 та S2. Звук S1 виникає на початку фази систоли. Він викликаний закриттям атріовентрикулярних клапанів, на початку скорочення шлуночків та виникає через раптове блокування току крові клапанами. Цей тон є найдовшим та найгучнішим у разі нормальної роботи серця. Звук S2 викликаний закриттям напівмісячних клапанів (аортальний клапан і легеневий клапан) в кінці шлуночкової систоли та на початку діастоли. Так само як і тон S1, цей звук викликаний раптовим блокуванням току крові клапанами. Має меншу амплітуду та протяжність у порівнянні з S1. Але часто виникає таке явище, що відоме як розщеплення, що робить цей звук довшим. Відбувається у випадку, коли через вдихання тиск у грудній клітці падає і тому більше часу уходить на проштовхування крові. Проте це явище може бути ознакою порушень у роботі серця, якщо розщеплення значне. Також іноді виділяються рідкі та менш помітні звуки S3, S4, які викликані коротким оберненим током крові і частіше за все також є ознаками порушень у роботі серця.

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином, після виділення основних тонів серця на фонокардіограмі проводиться аналіз їх характеристик і на основі цього робиться висновок про наявність відхилень у роботі серця. Як можна побачити, то основними задачами під час аналізу фонокардіограми стають сегментація (виділення звуків S1, S2, фази систоли та діастоли) та класифікація (визначення чи фонокардіограма є нормальною, чи є ознаки аномалій у роботі серця).

1.1.1 Опис процесу діяльності

У даній роботі автоматизовано процес аналізу фонокардіограми, а саме сегментацію сигналу за періодами систоли, діастоли, тонів S1, S2 та класифікацію, а саме визначення чи сигнал має ознаки порушень роботи серця чи ні. Як лікар прослухаючи фонокардіограму оцінює тривалість, гучність, частотні характеристики звуків серця, так і розроблена модель має спочатку виділяти основні тони серця на записі, потім аналізувати характеристики виділених звуків та на їх основі визначати чи є ознаки серцевих патологій у пацієнта, тобто відносити сигнал до одного з двох класів: нормальний сигнал, аномальний сигнал. В результаті пацієнт матиме змогу, записавши фонокардіограму з допомогою електронного стетоскопу, завантажити звуковий файл на сервер веб-застосунку та отримати візуалізацію сегментованого сигналу та попередній діагноз.

1.1.2 Опис функціональної моделі

У системі передбачені такі варіанти використання

Таблиця 1.1 – Варіант використання UC01

Назва	Завантаження звукового файлу
Опис	Користувач завантажує звуковий файл на сервер
Учасники	Користувач

Передумова

Відкрив головну сторінку веб-додатку

Продовження таблиці 1.1

Постумова	Надано сегментований сигнал та клас до якого віднесено сигнал(нормальний/аномальний)
Сценарій	<ol style="list-style-type: none">1. Користувач відкриває головну сторінку2. Користувач завантажу звуковий файл на сервер3. Сервер повертає веб-сторінку із зображенням сегментованого сигналу, а також клас до якого віднесений сигнал

Виходячи з описаного варіанта використання можна скласти такі функціональні вимоги

Таблиця 1.2 – Функціональна вимога REQ01

Номер	REQ01
Назва	Завантаження звукового сигналу на сервер
Опис	Веб-застосунок має давати змогу користувачу завантажити звуковий файл формату .wav на сервер

Таблиця 1.3 – Функціональна вимога REQ02

Номер	REQ02
Назва	Обробка звукового файлу
Опис	Сервер має обробити звуковий файл з фонокардіограмою для його сегментації та класифікації

Змн.	Арк.	№ докум.	Підпис	Дата

ДП ІС-6105.00.1081-с.ПЗ

Арк.

9

Таблиця 1.4 – Функціональна вимога REQ03

Номер	REQ03
Назва	Вивід результату
Опис	Сервер має повернути користувачу веб-сторінку, де зображено сегментований сигнал фонокардіограми та виведено назву класу, до якого віднесено сигнал

1.2 Огляд наявних аналогів

В останні роки із розвитком технологій запису фонокардіограми за допомогою електронних стетоскопів з’явилася велика кількість наборів даних, які включаються в себе записаний фонокардіографічний сигнал та деяку супутню інформацію, таку як записаний паралельно сигнал електрокардіограми та мітку про наявність у пацієнта порушень у роботі серця. Також дослідження у сфері аналізу даних, особливо машинного навчання та глибокого навчання і поява зручних інструментів для реалізації розроблених алгоритмів призвело до появи значної кількості робіт у сфері автоматичного аналізу роботи серця та аналізу фонокардіограми зокрема.

Як приклад робіт, у якій робота серця аналізується шляхом дослідження іншого сигналу, а саме електрокардіограми, можна привести роботи [3] та [4], де автори застосували лінгвістичне моделювання для виявлення аномалій у роботі серця.

Зазвичай у роботах дослідники зосереджуються лише на одній з вище описаних задач: сегментації або класифікації, тому роботи описані нижче присвячені кожній задач розглянути окремо.

Майже всі знайдені роботи з сегментації сигналу фонокардіограми використовуються один з трьох наступних методів: сегментація сигналу на основі аналізу огинаючої, методи машинного навчання чи приховані моделі Маркова. Оптимальним на даний момент вважається алгоритм сегментації, представлений у роботі [5]. У даній роботі використовується набір даних, який представлений паралельно записаними сигналами фонокардіограми та електрокардіограми, базуючись на якому отримується модель сегментації сигналу фонокардіограми, базуючись лише на одному цьому сигналі. Використовуючи інформацію про фізіологічні причини появи тонів S1 та S2 та їх зв'язок з елементами електрокардіограми, із сигналу електрокардіограми отримується сегментація фонокардіограми. Сигнал фонокардіограми моделюється як прихований марковський процес, де значення, що спостерігаються це значення амплітуди сигналу, що подається на вхід, а приховані стани це шукані сегменти періоди систоли, діастоли, тонів S1, S2. Особливістю роботи є процес знаходження параметрів моделі а також введення у модель параметру часу перебування системи певному стані. Матриця ймовірностей переходів та ймовірність початкових станів знаходяться як середнє значення цих параметрів по всіх сигналам з набору даних. Ймовірність перебування моделі у поточному стані впродовж наступних d одиниць часу моделюється як Гауссівський розподіл параметри якого знаходяться як середній тривалість та середньоквадратичне відхилення тривалості кожного з чотирьох станів. Нововведенням даної роботи є процес знаходження розподілу ймовірності отримання певного значення спостереження при певному стані системи. Шукана ймовірність виражається наступним чином

$$b_j(O_t) = P(O_t | q_t = j) = \frac{P(O_t) * P(q_t=j|O_t)}{P(q_t=j)} \quad (1.1)$$

Розподіл $P(O_t)$ моделюється як нормальний розподіл. А розподіл $P(q_t = j | O_t)$ знаходиться за допомогою тренування моделі логістичної регресії, на виході якої і отримується шукана ймовірність, якщо на вході подане значення спостереження.

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Модель розроблена у даній роботі була протестована на наборі даних, що складається з 405 записів сигналів фонокардіограми та електрокардіограми довжиною 30-40 секунд кожна. В результаті отримано точність 92.5%(частка коректно знайдених значень станів) та метрика F1 95.6%.

Розглядаючи існуючі рішення задачі класифікації фонокардіографічного сигналу, необхідно загадати проведене у 2016 змагання Classification of Heart Sound Recordings - The PhysioNet Computing in Cardiology Challenge 2016. Організатори змагання надали набір даних, який складається з 3126 записів фонокардіограм пацієнтів, та учасники мали розробити модель класифікації сигналів на класи: нормальний сигнал (немає ознак порушень у роботі серця), аномальний сигнал (є ознаки порушень у роботі серця). Найкращий результат (точність правильно класифікованих сигналів) показала модель, розроблена у роботі [6]. Автори роботи використали ансамбль моделей з нейронної мережі та алгоритму AdaBoost для класифікації сигналу фонокардіограми. Ансамбль моделей це техніка при якій для отримання загального результату класифікації аналізується результати окремих різних моделей (наприклад обирається клас, який видали більшість моделей). Для навчання алгоритму AdaBoost з сигналу було отримано 36 часових та 52 частотних характеристик сигналу. Конволюційна нейронна мережа, в свою чергу, приймає на вхід сигнал у 4 різних частотних діапазонах (25-45 Гц, 45-80 Гц, 80-200 Гц, 200-400 Гц) та видає на виході мітку класу. Сигнал визначається як аномальний, якщо обидві моделі визначили даний сигнал, як аномальний та як нормальний у іншому випадку. В результаті частка правильно визначених міток класів даною моделлю складає 86%

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3 Постановка задачі

1.3.1 Призначення розробки

У зв'язку з появою електронних стетоскопів та мобільних застосунків, які роблять запис фонокардіограми доступним не тільки лікарям, з'явилася потреба у автоматичному аналізі отриманої фонокардіограми без залучення лікаря, що значно пришвидшує діагностування хвороб серця. Розроблений алгоритм дозволяє швидко отримати попередній діагноз, щодо якості роботи серця.

1.3.2 Цілі та задачі розробки

Метою розробки є покращення існуючих алгоритмів виявлення аномалій у роботі серця за сигналом фонокардіограми. Це дозволить більш якісно автоматизувати діагностування серцево-судинних хвороб людини. Для цього потрібно розв'язати наступні задачі

1. Сегментація

Розбиття сигналу за фази систоли, діастоли та періодів тонів S1, S2

2. Класифікація

Визначення чи є сигнал фонокардіограми нормальним, чи в пацієнта є ознаки порушень у роботі серця

Висновки до розділу

У даному розділі описано природу появи основних тонів серця, їх зв'язок з серцевим циклом, описано процес аналізу фонокардіограми та основні кроки цього аналізу. В основному при аналізі фонокардіограми досліджують тони серця, які називаються S1 та S2, викликані закриттям клапанів. Ці тони завжди присутні у фонокардіограмі та дослідження їх характеристик може дати надійну інформацію про функціонування серця. Розглянуто основні роботи у даній області та методи які у них використані, найкращі результати в області

					ДП ІС-6105.00.1081-с.ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

сегментації сигналу отримані за допомогою прихованих марковських моделей, а в області класифікації сигналу використовуються комбіновані моделі машинного навчання. Встановлено основні функціональні вимоги до системи відповідно до описаних задач, які необхідно розв'язати аналізуючи сигнал фонокардіограми.

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						14
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Розглядаючи вхідні дані для створення системи аналіз фонокардіограми можна розглянути дві різні категорії

- Дані для навчання моделей аналізу сигналу
- Дані, які має надати користувач для аналізу сигналу

Так як аналіз фонокардіограми складався з задач сегментації та класифікації, то при створення моделі для кожної з задач використовувався свій набір даних. У випадку задачі сегментації було використано набір даних з 792 фонокардіограм, записаних у 135 пацієнтів. Дані представлені у вигляді бінарного файлу формату .mat, який містить у собі:

- сигнал фонокардіограми у вигляді значень амплітуди звукового сигналу, записаний з частотою 1000 Гц;
- позиції R піків та кінців T хвиль, отриманих з електрокардіограми, яка записана з частотою 50 Гц;
- діагноз пацієнта, 0 - здоровий, 1 - наявність серцевих патологій

Для створення моделі класифікації був взятий інший набір даних, який містить більше записів, бо великий об'єм вхідних даних це одна з ключових вимог для створення якісної моделі. Набір даних для класифікації включає в себе 3126 сигналів фонокардіограми, кожен довжиною від 5 до 120 секунд. Серед записів налічуються 665 аномільних сигналів та 2461 нормальних сигналів.

Для того щоб система змогла коректно проаналізувати сигнал фонокардіограми користувач має завантажити звуковий файл формату .wav, який містить якісний сигнал без сторонніх шумів та частоту дискретизації сигналу.

2.2 Вихідні дані

Вихідні дані для моделі сегментації сигналу це вектор тієї ж довжини, що і сигнал, де кожен елемент вектору дорівнює 0, 1, 2 або 3, що відповідає

					ДП ІС-6105.00.1081-с.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

належності до сегменту тону S1, систоли, тону S2, діастоли відповідно. Модель класифікації на виході повертає мітку класу 0 - робота серця нормальна, 1 - ознаки порушень у роботі серця. При користуванні застосунком, користувач отримує візуалізацію

2.3 Структура масивів інформації

Вхідний набір даних для сегментації зберігається у вигляді одного бінарного файлу формату .mat. Цей формат призначений для зберігання наборів даних у системі Matlab, проте бібліотека scipy мови Python дозволяє читати такі файли у звичайні масиви. Набір даних для тренування моделі класифікації складається з файли, кожен сигнал описують два файли .wav та .hea, де звуковий сигнал зберігається у файлі .wav, а метадані, такі як номер пацієнта та діагноз, зберігаються у .hea файлі

d0001.he	64 B	2016-02-29
d0001.w	25.9 KB	2016-02-29
d0002.he	66 B	2016-02-29
d0002.w	42.4 KB	2016-02-29

Рисунок 2.1 – Структура файлів

Вихідні дані приходять користувачу у вигляді файлу формату json наступної структури

```
{
  "segmentation": [
  ],
  "signal": [
  ],
  "result": "Abnormal"
}
```

Рисунок 2.2 – Файл з результатами сегментації та класифікації

2.4 Висновки до розділу

У даному розділі розглянуто які дані необхідні для розв'язання задач сегментації та класифікації, дані які має надати користувач для аналізу фонокардіограми. Було описано структуру даних для тренування моделей сегментації та класифікації та структура вихідного файлу після аналізу сигналу. Дані для сегментації зберігаються у бінарному файлі, який створений у програмі Matlab, а дані для класифікації зберігається у файлах .wav та .hea. Вказано вимоги

до вхідних даних користувача та структуру вихідних даних, які він отримує після аналізу фонокардіограми системою.

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						17
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Маємо сигнал фонокардіограми за яким необхідно визначити чи є в пацієнта порушення роботи серця та частоту дискретизації цього сигналу.

Фонокардіограма - крива, що зображує амплітуду та частоту звукових коливань, які виникають внаслідок роботи серця.

Частота дискретизації - кількість вимірів сигналу за секунду при перетворенні неперервного сигналу в дискретний.

Для розв'язку поставленої задачі необхідно провести сегментацію, а після цього класифікацію сигналу. Сегментація полягає в розділенні сигналу фонокардіограми на інтервали, що відповідають тонам S1, S2, періодам систоли та діастолі, ці інтервали не перетинаються та їх об'єднання покриває весь сигнал. Класифікація сигналу полягає у віднесенні сигналу до однієї з категорій: “нормальний” (серце працює нормально) або “аномальний” (порушення у роботі серця). Отже розв'язком задачі буде вектор з довжиною рівній довжині сигналу, де кожен елемент це число, що відповідає одному з чотирьох періодів, що необхідно визначити та мітка одного з двох класів

3.2 Математична постановка задачі

Нехай дано вектор дійсних чисел $X[n] = \{x_1, x_2, \dots, x_n\}$ та частоту дискретизації F . Необхідно сегментувати вхідний сигнал, а саме, вектору $X[n]$ поставити у відповідність вектор $Y[n] = \{y_1, y_2, \dots, y_n\}$, де $y_i \in \{0, 1, 2, 3\}$. Кожен елемент вектору $Y[n]$ відповідає одному з періодів: систолі, діастолі, тонам S1 або S2.

Для того щоб класифікувати сигнал необхідно за вхідним вектором $X[n]$ поставити у відповідність число з множини $\{0, 1\}$, де 0 відповідає належності до класу “нормальний сигнал”, 1 - до класу “аномальний сигнал”.

3.3 Опис методів розв'язання

Першим етапом роботи з даними після їх зчитування є передобробка даних. На етапі передобробки сигнал фільтрується за допомогою фільтра Баттерворта з верхньою пропускнуою частотою 400 Гц та нижньою частотою в 25 Гц. Після цього перетворення даних залежить від подальшого алгоритму.

Спочатку розглянемо задачу кластеризації, яка може бути розв'язана двома методами, які будуть описані нижче.

Один з розглянутих підходів це застосування конволюційних нейронних мереж. Конволюційні нейронні мережі це тип архітектури нейронних мереж, які включають в собі шари, де відбувається операція згортки матриці. Зазвичай конволюційні нейронні мережі застосовуються для обробки зображень, проте у даній роботі вони застосовані для аналізу сигналу, тому застосовується одновимірна згортка, таку конволюційну нейронну мережу називають одновимірною. Шар згортки включає в себе ядро згортки, вагові коефіцієнти якого знаходяться в процесі навчання, ці вагові коефіцієнти і є вагами даного шару. На кожному шарі може застосовуватися декілька згорток, тоді вихідний шар буде мати декілька каналів. Розглянемо l -й шар мережі на якому застосовується фільтр розміру 3, вхідна матриця A^l , яка має N_l каналів, вихідна матриця матиме k_l каналів, ваги мережі $W^l \in R^{N^l \times k^l - 1}$, тоді вихідні значення даного шару

$$Z_{i,j}^l = \sum_{i'}^3 \sum_{j'}^{N_l} A^l_{i'j'} W^l_{i+i',j} \quad (2.1)$$

Приклад одновимірної згортки зображений на рисунку нижче.

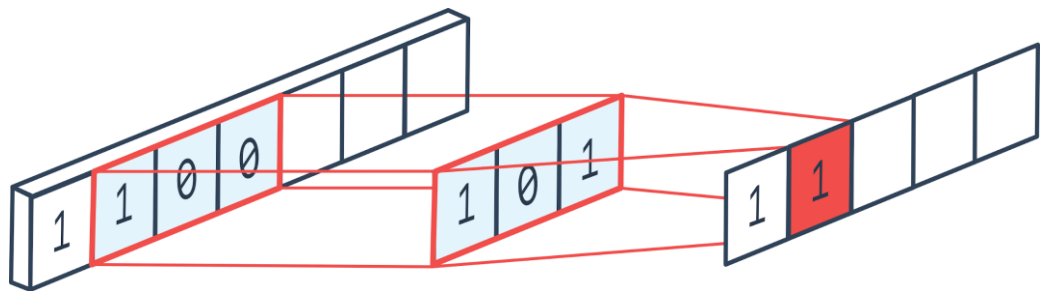


Рисунок 3.1 - Одновимірна згортка

згортку, а червоні пунктирні лінії позначають конкатенацію шарів. Конкатенація шарів дозволяє скопіювати інформацію про ще не спотворені дані на ранніх шарах і передати їх до останніх шарів, де застосовується операція оберненої згортки. На всіх шарах, крім останнього застосовується функція активації $ReLU(x) = \max(x, 0)$, а на вихідному шарі для отримання ймовірностей класів (належності відповідного елементу вектора до кожного з чотирьох можливих інтервалів, у даній конкретній задачі), використовується функція $Softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$. Використані функції активації дозволяють забезпечити нелінійність перетворень у мережі, а також запобігають проблемі затухаючого градієнту. На виході даної нейронної мережі отримується матриця $Y^{N \times 4}$, для отримання номеру відповідного періоду необхідно взяти argmax по кожному рядку матриці і тоді отримаємо шуканий вектор $Y[n]$.

У даному випадку мережа тренується не на фільтрованому сигналі, а на 4 типах огибаючих: огибаюча Гілберта, гомоморфна огибаюча, вейвлет огибаюча, огибаюча спектральної щільності потужності. Огибаюча - це така крива, що побудована по характерним точкам сигналу, наприклад по екстремумам.

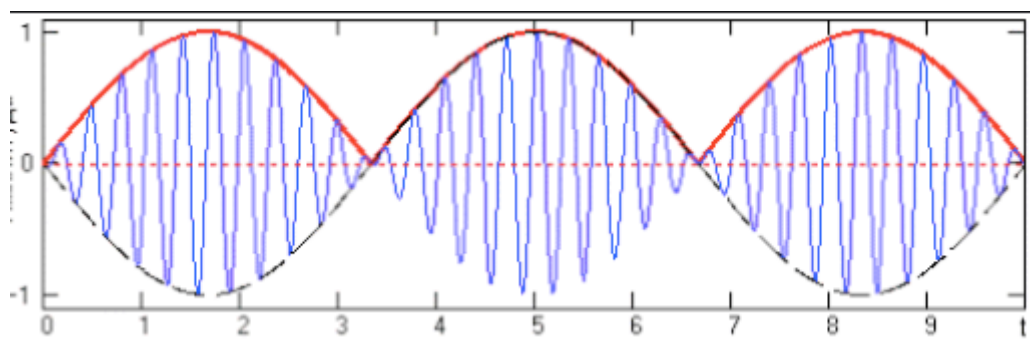


Рисунок 3.4 – Огибаюча синусоїди

Огибаюча Гілберта це модуль з перетворення Гілберта. Перетворення Гілберта це згортка сигналу з ядром перетворення $h(t) = \frac{1}{\pi t}$, отже саме перетворення виглядає як $s(t) = \int \frac{s(\tau)}{\pi(t-\tau)} d\tau$.

Гомоморфна огибаюча отримана наступним чином

$$HEnv(t) = e^{L(|x(t)|)} \quad (2.2)$$

де L - фільтр Баттерворта із пропусканням частот менше 8 КГц

									Арк.
									21
Змн.	Арк.	№ докум.	Підпис	Дата					

Вейвлет огибаюча отримана після розрахунку модуля коефіцієнтів неперервного вейвлет перетворення, був використаний вейвлет Морле. Вейвлет перетворення дозволяє перетворити сигнал у частотно-часове представлення, використовуючи згортку сигналу із вейвлет функцією [8]. Вейвлет функція це функція, що локалізована у часі та частоті та має середнє значення 0. Вейвлет Морле має наступний вигляд

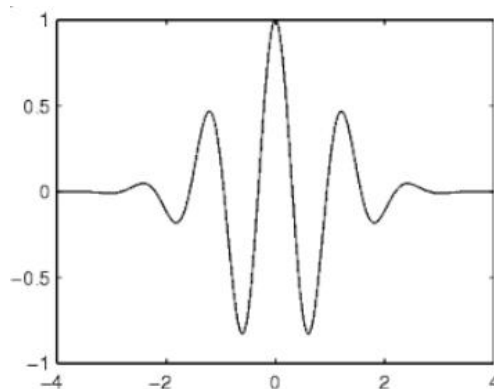


Рисунок 3.5 – Графік функції вейвлету Морле

Загальний вигляд неперервного вейвлет перетворення

$$W(a, b) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{a}} \frac{t-b}{a} dt \quad (2.3)$$

Де a та b це коефіцієнти розтяження за зміщення вейвлет функції.

Результатом перетворення є коефіцієнти $W(a, b)$

Огибаюча спектральної щільності потужності отримана внаслідок діставання спектральної щільності потужності у частотах від 40 Гц до 60 Гц, саме у цих частотах переважно лежать тони S1 та S2. Спектральна щільність потужності це функція залежності сигналу від частоти, вона допомагає дослідити енергію компонент сигналу різної частоти. Отримати можна, розрахувавши функцію автокореляції коефіцієнтів перетворення Фур'є

Цей набір огибаючих було обрано як той, що найкраще виділяють інформацію з сигналу у потрібних частотах та при цьому зберігають амплітуду звуків.

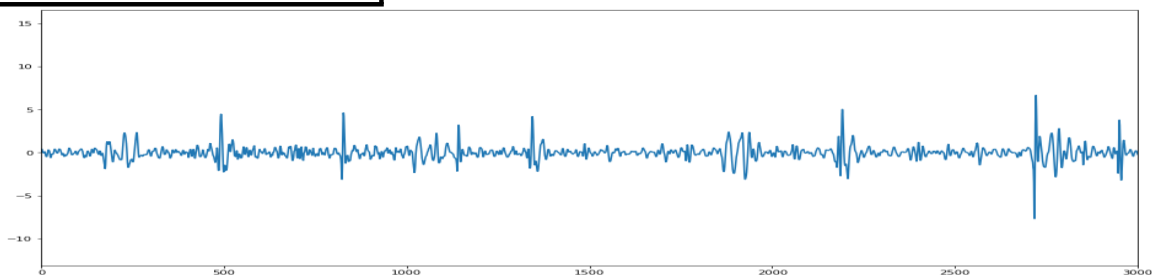


Рисунок 3.6 – Уривок сигналу фонокардіограми

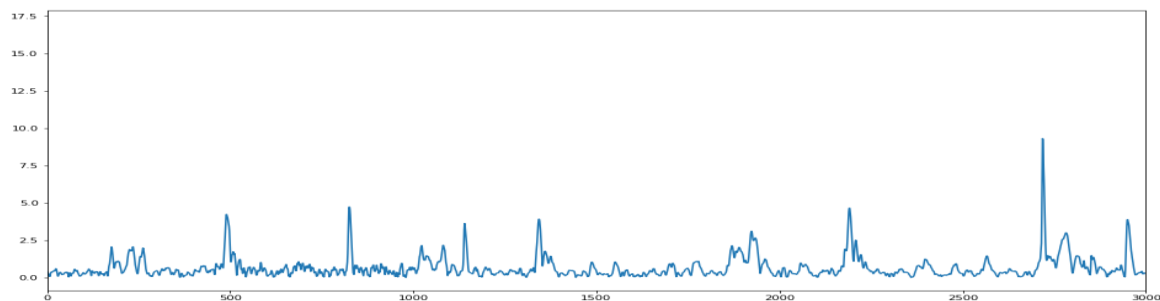


Рисунок 3.7 – Огинаюча Гілберта

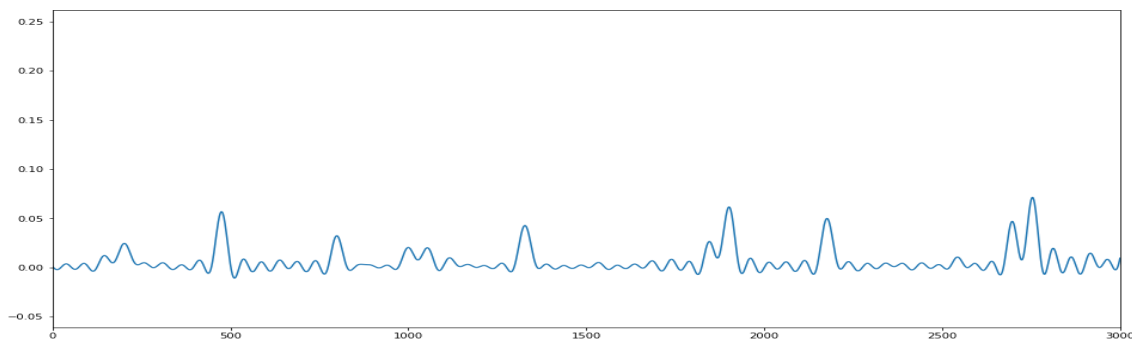


Рисунок 3.8 – Огинаюча спектральної щільності потужності

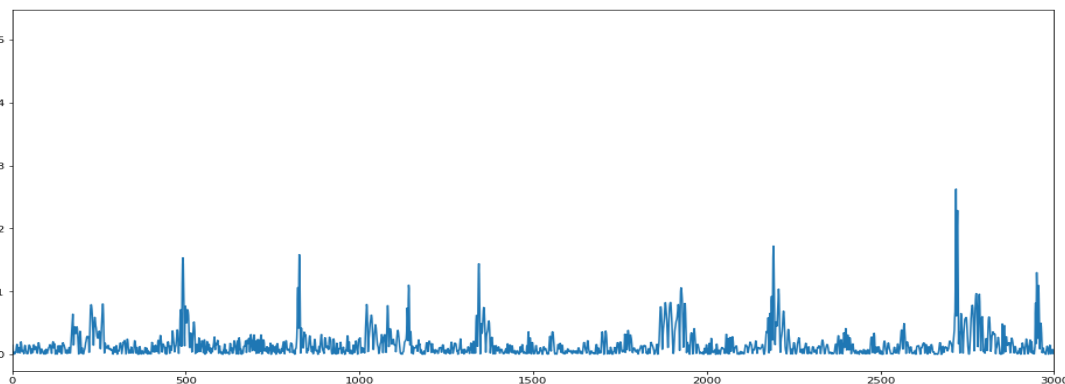


Рисунок 3.9 – Вейвлет огинаюча

Змн.	Арк.	№ докум.	Підпис	Дата

$$P_j(O_t) = P(O_t | q_t = j) = \frac{P(q_t=j|O_t)P(O_t)}{P(q_t=j)} \quad (2.4)$$

Розподіли ймовірності $P(O_t)$ та $P(q_t = j)$ моделюються за нормальним законом розподілу, параметри якого легко знаходяться за тренувальним набором даних, як середнє значення мат очікування та середньоквадратичного відхилення значень сигналу чи частоті станів по всіх сигналах. Ймовірність $P(q_t = j | O_t)$ знаходиться за допомогою логістичної регресії, бо фактично вхідним параметром є значення сигналу, а вихідним є ймовірність стану або класу в загальному випадку. Аналогічний підхід до розрахунку даного розподілу ймовірності було використано у роботі [9]

Отже знайшовши відомі параметри ПММ необхідно знайти послідовність прихованих станів, для цього використано алгоритм Вітербі. Введемо ймовірність, того, що у момент часу t система знаходилася в стані j , а в момент часу $t - d$ у стані j

$$\delta_t(j, i, d) = \delta_{t-d}(i) a_{ij} p_j(d) \prod_{s=0}^{d-1} b_j(O_{t-s}) \quad (2.5)$$

Тоді можна знайти ймовірність того, що попередній стан був j та момент часу t це момент переходу у наступний стан

$$\delta_t(j) = \max_d \max_{i \neq j} \delta_{t-d}(i) a_{ij} p_j(d) \prod_{s=0}^{d-1} b_j(O_{t-s}) \quad (2.6)$$

Обчислюється найбільш ймовірна тривалість поточного стану

$$D_t(j) = \operatorname{argmax}_d \max_{i \neq j} \delta_{t-d}(i) a_{ij} p_j(d) \prod_{s=0}^{d-1} b_j(O_{t-s}) \quad (2.7)$$

Звідси знаходиться найбільш ймовірний попередній стан

$$\psi_t(j) = \operatorname{argmax}_i [\delta_{t-D_t(j)} a_{ij}] \quad (2.8)$$

Далі за допомогою динамічного програмування знаходиться послідовність станів.

Далі розглянемо алгоритми використані для класифікації сигналу. У роботі було використано алгоритм градієнтного бустинга. Цей алгоритм є одним з найефективніших алгоритмів розпізнавання, що досягається завдяки використанню техніки композиції базових алгоритмів.

Градієнтний бустинг це алгоритм машинного навчання, який складається з множини простих моделей і кожна з моделей навчається на помилці

виділяти певні патерни на сигналах за допомогою операції згортки, а також враховувати розташування виділених патернів у сигналі. До того ж розглянута архітектура нейронної мережі широко використовується у сегментуванні зображень, отже можна спробувати застосувати її і до сегментування сигналів.

Інший підхід - моделювання сигналу як марковського процесу. Так як під час сегментації сигналу стоїть задача визначення стану процесу роботи серця для кожного виміру сигналу, то застосовується прихована марковська модель. Цей метод був реалізований, бо ПММ добре себе зарекомендувала у задачі розпізнавання мовлення, а задача сегментації фонокардіограми є схожою задачею, бо необхідно виділяти певні звуки у сигналі. На даний момент, відповідно до опублікованих робіт з сегментації сигналу фонокардіограми [9], ПММ показують найкращий результат у розв'язку цієї задачі.

Для задачі класифікації було використано градієнтний бустинг, що є одним з найпотужніших алгоритмів класифікації на даний момент. Градієнтний бустинг забезпечує високу точність класифікації за рахунок того, що він є композицією великої кількості простих базових моделей класифікації, кожна з яких навчається на помилці попередньої моделі, фактично виправляючи її помилку прогнозування. Так як у якості базових моделей були використані дерева рішень, то процес навчання та прогнозування є швидким.

В результаті реалізації даних алгоритмів були отримані наступні результати. Для оцінки задачі сегментації оцінювалася середня доля правильно визначених станів для кожного заміру сигналу. Дана метрика розраховувалася як на даних, на яких навчалася модель, так і на залишку даних (валідаційний набір даних), при тестуванні валідаційний набір складав 10 % від розміру всього набору даних. Репрезентативним є результат отриманий на валідаційному наборі даних. Для конволюційної мережі отримано результат у 93 %, а для ПММ - 73 %. Можна побачити, що конволюційна нейронна мережа має велику перевагу у точності, в порівнянні з іншим методом.

Для задачі класифікації використовувалася така сама методика тестування, було відібрано 10 % даних від усього набору, а на інших 90 % була навчена

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

модель. Проте у даному випадку для оцінки якості було обрано метрику F1. Це комбінована метрика, що є середнім гармонійним точності та повноти. Точність - частка правильно визначених позитивних класів, тобто тих, що мають умовну мітку “1”, у даній задачі це наявність аномалій у роботі серця. Повнота - ймовірність того, що позитивний прогноз моделі (умовна мітка “1” на виході) є вірним. Дана метрика була обрана через незбалансованість набору даних, відношення розміру класів складає приблизно 4 до 1 з більшою кількістю нормальних сигналів. Матрицю помилок можна побачити нижче

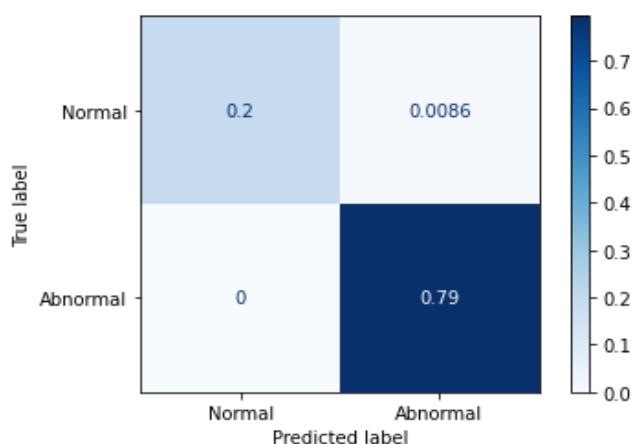


Рисунок 3.11 – Матриця помилок

На матриці помилок можна побачити, що жодного разу не було спрогнозовано клас “Нормальний сигнал”, при тому що справжня мітка була “Аномальний сигнал”. Це важливий показник при роботі з медичними даними, бо цей результат говорить про те, що модель схильна до помилки у бік виявлення сигналу як аномального. Це зменшує ймовірність того, що хвороба не буде виявлена у пацієнта, але допускає можливість хибно позитивного діагнозу. В результаті значення F1 метрики

Висновки до розділу

У даному розділі сформульовано змістовну та математичну постановку задачі. В ході опису методів розв’язання описано принципи роботи використаних у роботі алгоритмів, а саме конволюційних нейронних мереж та прихованих марковських моделей для сегментації та градієнтного бустингу для класифікації

сигналу. Описані переваги цих моделей та обґрунтоване їх використання при розв'язку поставлених задач. Розглянуті результати тестування обраних моделей. Конволюційні нейронні мережі виявилися набагато кращими за приховані марковські моделі, із точністю у 93 %. Для оцінки класифікації було використано F1 метрику і алгоритм градієнтного бустингу продемонстрував результат

					ДП ІС-6105.00.1081-с.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

4 Програмне та технічне забезпечення

4.1 Засоби розробки

Для розробки програмного продукту були використані наступні інструменти. Розробка велась на платформі OS Linux Ubuntu 18.04 - операційній системі з відкритим кодом, що є дистрибутивом ядра Linux. Для написання коду було використано середу розробки Visual Studio Code. Код написаний на мові програмування Python версії 3.8. Python є популярною мовою програмування загального призначення, що поєднує в собі високу швидкість розробки та читаємість коду. Перевагою мови Python є велика кількість сторонніх бібліотек, пакетів та фреймворків, які дозволяють реалізувати програмні застосунки будь-якого спрямування.

Зокрема, для зчитування даних з бінарних файлів та звукових файлів була використана бібліотека Scipy, яка має відповідні функції для читання файлів формату .mat та звукових файлів у форматі .wav. Для перетворення та фільтрації сигналів було також використано бібліотеку Scipy, що має модуль scipy.signal, який включає в себе функції для роботи з сигналами. Також при роботі з сигналами було використано бібліотеку NumPy, це бібліотека оптимізована для роботи з векторами, матрицями та високорівневий математичними операціями.

Для тренування нейронної мережі використано Tensorflow - бібліотеку для машинного навчання. Ця бібліотека дозволяє описувати архітектуру моделей нейронних мереж та контролювати процес їх навчання, також є можливість тренувати мережі на GPU. При користуванні Tensorflow було використано бібліотеку Keras, що надає доступ до високорівневого API для користування функціями Tensorflow.

При реалізації прихованої марковської моделі було використано бібліотку hsmmlearn. Це бібліотека з відкритим вихідним кодом, що надає функціонал для роботи з марковськими моделями.

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Для тренування градієнтного бустингу використано бібліотеку Scikit-learn. Це бібліотека машинного навчання, яка включає в себе реалізацію великої кількості моделей для задач класифікації, регресії та кластеризації.

Для демонстрації роботи моделей було створено веб-застосунок, який був написаний за допомогою фреймворку Flask, що дає змогу обробляти запити користувача.

4.2 Вимоги до технічного забезпечення

Для користування веб-застосунком необхідно мати встановлений браузер Google Chrome на будь-якій операційній системі. Для запуску серверу веб-застосунку необхідно мати комп'ютер з наступними мінімальними конфігураціями:

- Операційна система Linux Ubuntu 18.04
- Об'єм оперативної пам'яті 4 Гб
- Об'єм вільного місця на жорсткому диску 10 Гб
- Інтерпритатор Python версії 3.8
- Наступні бібліотеки мови Python: Flask, Tensorflow, Keras, Numpy, Scipy, Scikit-learn
- файли з моделями сегментації та класифікації

					ДП ІС-6105.00.1081-с.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма послідовності

При роботі з веб-застосунком для аналізу можна виділити єдиний процес аналізу завантаженого файлу з сигналом фонокардіограми. В цей процес залучені користувач, веб-сервер та моделі сегментації та класифікації сигналу. Процес складається з таких дій. Спочатку користувач завантажує файл з сигналом на сервер, сервер приймає цей файл, зчитує з нього сигнал та передає сигнал до моделі сегментації сигналу. Модель сегментує сигнал та повертає результат до веб-сервера, який посилає сегментований сигнал на обробку до моделі класифікації. Модель класифікації повертає свій результат до веб-сервера. Після цього веб-сервер формує результати класифікації та сегментації та повертає сторінку з результатами користувачу. Діаграма застосунків знаходиться у графічних матеріалах

4.3.2 Діаграма класів

На діаграмі класів, що знаходиться у графічних матеріалах, зображено 3 класи. Клас Model містить в собі об'єкт математичної моделі, методи fit та predict призначені для тренування моделі та для повернення прогноза за вхідними даними відповідно. Клас Signal містить в собі поля data та fs. Поле data є вектором значень амплітуди сигналу, а fs - параметр дискретизації сигналу. Клас SignalProcessor має поля типів Signal та Model, містить функції, що реалізують сегментацію та класифікацію сигналу.

4.3.3 Діаграма розгортання

На діаграмі розгортання, що знаходиться у графічних матеріалах зображено такі елементи системи, як сервер (Server) та робоча машина користувача (User workstation). На робочій машині користувача працює веб-браузер, який

відправляє запити на сервер застосунку. Запити на сервері обробляє програма, що написана за допомогою веб-фреймворку Flask.

4.3.4 Специфікація функцій

Таблиця 4.1 – Специфікація функцій

Функція	Опис
<pre>@app.route('/segmentation', methods=['POST']) def segmentation():</pre>	<p>Функція повертає представлення, що є json документом із результатом сегментації. Приймає HTTP запит методу POST, отримує дані про сигнал</p> <ul style="list-style-type: none"> - повертає код статусу 200 та результати сегментації, якщо алгоритм успішно виконано - повертає код статусу 500, якщо в ході роботи алгоритму виявлено помилку виконання
<pre>@app.route('/classification', methods=['POST']) def classification():</pre>	<p>Функція повертає представлення, що є json документом із результатом класифікації. Приймає HTTP запит методу POST, отримує дані про сигнал</p> <ul style="list-style-type: none"> - повертає код статусу 200 та результати класифікації, якщо алгоритм успішно виконано - повертає код статусу 500, якщо в ході роботи алгоритму виявлено помилку виконання

Продовження таблиці 4.1

<pre>@app.route('/', methods=["GET", "POST"]) def index():</pre>	<p>Приймає HTTP записи методами GET або POST. Якщо надіслано GET запит, то повертається представлення головної сторінки застосунку</p> <p>Якщо надіслано POST запит, то повертається представлення сторінки з результатами аналізу</p> <ul style="list-style-type: none"> - повертає код статусу 200 та результати аналізу, якщо аналіз успішно виконано - повертає код статусу 500, якщо в ході аналізу виявлено помилку виконання
<pre>def get_envelope(signal):</pre>	<p>За наданим сигналом функція повертає огинаючу Гілберта</p>
<pre>def butterworth_filterig(signal, cutoff_freq, order, sample_rate, mode='low'):</pre>	<p>За вхідним сигналом, частотою дискретизації та за частотною полосою пропускання повертає фільтрований сигнал</p>
<pre>def wavelet_denoising(signal, threshold=0.4):</pre>	<p>Фільтрує сигнал за допомогою вейвлет перетворення та повертає фільтрований сигнал</p>
<pre>def average_smoothing(signal, lag=1):</pre>	<p>Згладжування сигналу рухомих середнім</p>

Продовження таблиці 4.1

<code>def homomorphic_envelope(input_signal, sampling_frequency, lpf_frequency=8):</code>	Функція повертає гомоморфну огибаючу сигналу
<code>def downsampling(signal, fs_old, fs_new):</code>	Функція повертає частоту дискретизації сигналу та повертає перетворений сигнал
<code>def normalization(signal):</code>	Нормалізація сигналу до середнього значення 0 та середньоквадратичного відхилення 1
<code>def psd_envelope(signal, pcg_fs):</code>	Функція повертає огибаючу спектральної щільності потужності
<code>def wavelet_envelope(signal):</code>	Функція повертає вейвлет огибаючу сигналу
<code>def spike_removal(signal, pcg_fs):</code>	Функція видаляє викиди на сигналі та повертає перетворений сигнал
<code>def extract_features(signal, fs):</code>	Функція повертає характеристики сигналу, що описані у математичному розділі
<code>def slice_signal(pcg, slice_length, step):</code>	Нарізає сигналу на уривки заданої довжини та повертає список уривків
<code>def get_segmentation(self, signal, fs):</code>	Функція повертає вектор із сегментованим сигналом
<code>def get_classification(self, signal, segm, fs):</code>	Функція повертає результат класифікації сигналу
<code>def get_heart_rate(self, signal, fs):</code>	Функція повертає значення пульсу, що отримане з вхідного сигналу

Змн.	Арк.	№ докум.	Підпис	Дата

Висновки до розділу

У даному розділі описано технології та мову, що використані при розробці програмного забезпечення. Складено вимоги до технічного забезпечення для успішного запуску розробленої програми. Описано діаграму послідовності, що відображає процеси, що відбуваються при роботі програми. За допомогою діаграми класів описано основні сутності програмного коду. Наприкінці наведено специфікацію функцій програми, описано дії даних функцій, їх вхідні та вихідні дані.

					ДП ІС-6105.00.1081-с.ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Згідно з описаним функціоналом веб-застосунку, користувач має змогу завантажити файл з фонокардіограмою на сервер та отримати результати аналізу.

На головній сторінці застосунку є форма завантаження файлу. Для завантаження файлу необхідно натиснути кнопку Choose File



Рисунок 5.1 – Головна сторінка веб-застосунку

Після цього з'явиться діалогове вікно з можливістю обрання файлу. У ньому необхідно обрати потрібний файл формату .wav

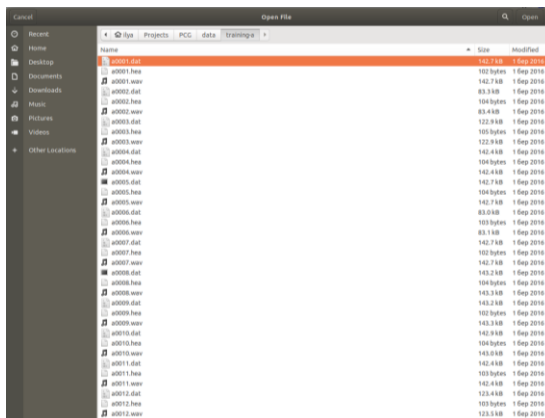


Рисунок 5.2 – Вікно вибору файлу

Після вибору файлу, для його відправлення на сервер для обробки, необхідно натиснути кнопку “Відправити”. Після відправлення файлу на сервер користувач потрапляє на сторінку результатів, де до отримання цих результатів буде наявний напис “Обробка сигналу”.

Змн.	Арк.	№ докум.	Підпис	Дата

Рисунок 5.3 – Вікно очікування результатів

Після того як результати аналізу будуть отримані, вони з'являться на сторінці. У першому рядку отримано результат класифікації сигналу, в другому - пульс розрахований за сигналом, далі - графік з сегментованим сигналом



Рисунок 5.4 – Сторінка з результатом аналізу

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

Метою випробувань є перевірка відповідності функцій комплексу задач з аналізу фонокардіографічного сигналу вимогам технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.60392. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Для перевірки виконання функціональних вимог були проведені тести розробленого програмного забезпечення. Опис тестів та їх результати наведені нижче

Таблиця 5.1 – Тест 1

Назва	Тестування відправлення даних без вибраного файлу
Початковий стан	Відкрита головна сторінка застосунку і файл з сигналом не обрано
Виконані дії	Було натиснуто кнопку “Відправити”
Результат	Відкрита головна сторінка з повідомлення про те що необхідно обрати файл
Відповідність вимогам	Відповідає

Таблиця 5.2 – Тест 2

Назва	Тестування відправлення файл невірною формату
Початковий стан	Відкрита головна сторінка застосунку
Виконані дії	Завантажено файл з розширенням не .wav та натиснуто кнопку “Відправити”
Результат	Відкрита головна сторінка з повідомлення про те що необхідно обрати файл з розширенням .wav
Відповідність вимогам	Відповідає

Таблиця 5.3 – Тест 3

Назва	Тестування відправлення файл формату .wav
Початковий стан	Відкрита головна сторінка застосунку
Виконані дії	Завантажено файл з розширенням .wav та натиснуто кнопку “Відправити”
Результат	Відкрита сторінка результатів, де після проведення аналізу відображається результат класифікації та сегментації
Відповідність вимогам	Відповідає

Висновки до розділу

У даному розділі було розроблено керівництво користувача, наведено зображення основних сторінок веб-застосунку. Сформовано мету тестування застосунку. Складено та проведено тести із застосунком, в результаті виявлено, що робота застосунку відповідає функціональним вимогам, що були описані вище

					ДП ІС-6105.00.1081-с.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗАГАЛЬНІ ВИСНОВКИ

Поява великої кількості медичних даних, що зумовлена ростом доступності електронного медичного обладнання, відкриває можливості для розробки математичних моделей, що могли б автоматизувати певні аспекти роботи лікарів у напрямку діагностування хвороб. Тому у даній роботі представлено дослідження комплексу задач з аналізу фонокардіографічного сигналу. В ході роботи було досліджено алгоритми сегментації та класифікації сигналу фонокардіограми для виявлення аномалій у роботі серця.

На початку роботи було описано предметну область, дано визначення основним медичним термінам, що використовувалися далі, в описано процес роботи серця. Оглянуто основні підходи до розв'язку поставлених задач та наведено наявні роботи з цієї області та результати, отримані у них. Сформульовано мету роботи та задачі, що необхідно розв'язати.

У наступному розділі описано структуру даних, що використовувалися при розробці. Наведено структуру наборів даних на яких було натреновано розроблені моделі.

Після цього було сформульовано змістовну та математичну постановку задачі. Описано алгоритми та моделі, що були використані у роботі, а саме конволюційні нейронні мережі, приховані марковські моделі та градієнтний бустинг. Обґрунтовано вибір цих методів та наведено результати тестування реалізованих моделей. Для задачі сегментації було порівняно якість прогнозування конволюційних нейронних мереж та прихованих марковських моделей. З отриманих результатів зроблено висновок, що конволюційні мережі є більш ефективними для розв'язку задачі сегментації сигналу фонокардіограми.

Також було описано розроблене програмне забезпечення. Наведено основні засоби для розробки програми та архітектуру програмного забезпечення, що була описана за допомогою діаграми класів та послідовності, а також специфікації функцій, де докладно описано призначення кожної з функцій програми.

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Наприкінці було продемонстровано роботу програми, проведено тести, що мали на меті перевірити відповідність розробленого продукту поставленим вимогам. В результаті, всі тести були успішно пройдені, тому зроблено висновок про те, що розроблена програма відповідає поставленим вимогам до функціональності.

Отже, у даній роботі було досліджено комплекс задач з аналізу фонокардіографічного сигналу та розроблено програмний продукт для автоматизації аналізу фонокардіограми за допомогою реалізованих алгоритмів.

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						43
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ПЕРЕЛІК ПОСИЛАНЬ

1. Бусов І.О., Гавриленко О.В. Сегментація та класифікація сигналу фонокардіограми. Інформаційні системи та технології управління, 2019. №3, С. 183-186.
2. Бусов І.О., Гавриленко О.В. Комплекс задач з аналізу фонокардіографічного сигналу. Інформаційні системи та технології управління, 2020. №4, С. 22-25.
3. Baklan I, Mukha I, Oliinyk Y, Lishchuk K, Nedashkivsky E, Gavrilenko O. ANOMALIES DETECTION APPROACH IN ELECTROCARDIOGRAM ANALYSIS USING LINGUISTIC MODELING. Springer, 2019. 513 с
4. Baklan I, Oliinyk Y, Mukha I, Lishchuk K, Gavrilenko O, Ocheretianyi O, Tsytsyliuk A. Adaptive Multistage Method of Anomalies Detection in ECG Time Series. Computational Linguistics and Intelligent Systems. Proc. 4 th Int. Conf. COLINS, 2020. 670 с.
5. Springer D, Tarassenko L, Clifford G. Logistic Regression-HSMM-based Heart Sound Segmentatio. IEEE Transactions on Biomedical Engineering, 2016. 11 с.
6. Potes C, Parvaneh S, Rahman A, Conroy B. Ensemble of Feature-based and Deep learning-based Classifiers for Detection of Abnormal Heart Sounds. 2016 Computing in Cardiology Conference, 2016. 4 с.
7. Renna F, Oliveira J, Coimbra M. Deep Convolutional Neural Networks for Heart Sound Segmentation. IEEE Journal of Biomedical and Health Informatics, 2019. 12 с.
8. Chun-Lin L. A Tutorial of the Wavelet Transform [Електронний ресурс]. – Режим доступу до ресурсу: <http://people.duke.edu/~hpgavin/SystemID/References/LiuWaveletTransform-2010.pdf>.
9. S E Schmidt, C Holst-Hansen, C Graff, E Toft and J J Struijk. Segmentation of heart sound recordings by a duration-dependent hidden Markov model. IEEE, 2010. 18 с.

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

10. F Meziani, S M Debbal, A Atbi. Analysis of Phonocardiogram Signals Using Wavelet Transform. Journal of Medical Engineering & Technology, 2012. 20 c.
11. Grzegorzczuk I, Stępień K, Gierałtowski J, Rymko J. PCG Classification Using a Neural Network Approach. Computing in Cardiology, 2016. 4 c.
12. Ping Wang, Chu Sing Lim, Sunita Chauhan, Jong Yong Abdiel. Phonocardiographic Signal Analysis Method Using a Modified Hidden Markov Model. Annals of Biomedical Engineering, 2007. 8 c.

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						45
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ДОДАТОК А

Тексти програмного коду

Система аналізу фонокардіографічного сигналу

(Найменування програми (документа))

DVD-R

(Вид носія даних)

12.7 кб, 8 арк.

(Обсяг програми (документа) , арк.,)

Київ - 2020

					ДП ІС-6105.00.1081-с.ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import os

import base64
from io import BytesIO
import requests
import json
import pickle

from flask import Flask, request, jsonify, url_for
from flask import render_template
import scipy
from scipy.io import wavfile
from matplotlib.figure import Figure

from processing import SignalProcessor

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'data/'
ALLOWED_EXTENSIONS = {'wav'}

def allowed_file(filename):
    return '.' in filename and filename.split('.')[-1].lower() in ALLOWED_EXTENSIONS

@app.route('/segmentation', methods=['POST'])
def segmentation():
    data = request.get_json()
    processor = SignalProcessor()
    sign, segm = processor.get_segmentation(data['signal'], data['fs'])
    return jsonify({
        'signal': sign.tolist(),
        'fs': data['fs'],
        'segmentation': segm.argmax(1).tolist()
    })

@app.route('/classification', methods=['POST'])
def classification():
    data = request.get_json()
    processor = SignalProcessor()
    result = processor.get_classification(data['signal'], data['segmentation'], data['fs'])
    heart_rate = processor.get_heart_rate(data['signal'], data['fs'])
    return jsonify({
        'result': result,
        'heart_rate': heart_rate
    })

@app.route('/', methods=["GET", "POST"])

```

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

def index():
    if request.method == 'GET':
        return render_template('index.html')
    elif request.method == 'POST':
        if 'file' in request.files:
            if request.files['file'].filename == "":
                return render_template('index.html', error_msg='Файл не обрано!')
            if not allowed_file(request.files['file'].filename):
                return render_template('index.html', error_msg='Недопустиме розширення файлу! '
                    'Доступні такі розширення {}'.format(', '.join(ALLOWED_EXTE
NSIONS)))
            file = request.files['file']
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], file.filename))

            fs, data = wavfile.read(os.path.join(app.config['UPLOAD_FOLDER'], file.filename))
            send_data = {
                'signal': data.tolist(),
                'fs': fs
            }

            return render_template('result.html', data=send_data)

if __name__ == '__main__':
    app.run('0.0.0.0', port=8080, debug=True)

import pickle

from keras import models
import numpy as np
import scipy
import scipy.signal
import scipy.fftpack
import pywt
import sklearn

def get_envelope(signal):
    return np.abs(scipy.signal.hilbert(signal))

def butterworth_filterig(signal, cutoff_freq, order, sample_rate, mode='low'):
    b, a = scipy.signal.butter(order, cutoff_freq, fs=sample_rate, btype=mode)
    signal = scipy.signal.filtfilt(b, a, signal)
    return signal

def wavelet_denoising(signal, threshold=0.4):
    w = pywt.Wavelet('sym4')
    maxlev = pywt.dwt_max_level(len(signal), w.dec_len)

```

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

```

coeffs = pywt.wavedec(signal, 'sym4', level=maxlev)
for i in range(1, len(coeffs)):
    coeffs[i] = pywt.threshold(coeffs[i], threshold * max(coeffs[i]))

return pywt.waverec(coeffs, 'sym4')

def average_smoothing(signal, lag=1):
    result_signal = np.zeros(len(signal))
    for i in range(len(signal)-1):
        result_signal[i] = np.mean(signal[max(0, i - lag):i+1])
    return result_signal

def homomorphic_envelope(input_signal, sampling_frequency,lpf_frequency=8):
    B_low, A_low = scipy.signal.butter(1,2*lpf_frequency/float(sampling_frequency),'low')
    homomorphic_envelope = np.exp(scipy.signal.filtfilt(B_low,A_low,np.log(np.abs(scipy.signal.hilbert(input_signal))))))
    return homomorphic_envelope

def downsampling(signal, fs_old, fs_new):
    return scipy.signal.resample_poly(signal, 1, fs_old // fs_new)

def normalization(signal):
    return (signal - np.mean(signal)) / np.std(signal)

def psd_envelope(signal, pcg_fs):
    f, t, Sxx = scipy.signal.spectrogram(signal.flatten(),
        fs=pcg_fs,
        window='hamming',
        nperseg=pcg_fs // 20,
        noverlap=pcg_fs // 40)
    mean_Sxx = np.mean(Sxx[2:4, :], axis=0)
    return scipy.signal.resample(mean_Sxx, len(signal))

def wavelet_envelope(signal):
    return abs(pywt.cwt(signal, [1, 2, 3], 'morl')[0][-1])

def spike_removal(signal, pcg_fs):
    frames = []
    signal = signal.flatten()
    for i in np.arange(0, len(signal), 0.5 * pcg_fs):
        frames.append(signal[int(i):min(len(signal), int(i+0.5*pcg_fs))])

    MAAs = np.zeros(len(frames))
    for index, frame in enumerate(frames):
        MAAs[index] = np.max(np.abs(frame))

    while np.sum(MAAs > 2 * np.median(MAAs)) > 0:

```

```

max_index = np.argmax(MAAs)
MAAs_index = np.argmax(np.abs(frames[max_index]))

left_zero_crossing = MAAs_index
for i in range(MAAs_index - 1, -1, -1):
    if frames[max_index][i] * frames[max_index][i + 1] < 0:
        left_zero_crossing = i
        break

right_zero_crossing = MAAs_index
for i in range(MAAs_index + 1, len(frames[max_index])):
    if frames[max_index][i] * frames[max_index][i - 1] < 0:
        right_zero_crossing = i
        break

if right_zero_crossing == MAAs_index:
    right_zero_crossing = len(frames[max_index]) - 1
frames[max_index][left_zero_crossing:right_zero_crossing+1] = 0
for index, frame in enumerate(frames):
    MAAs[index] = np.max(np.abs(frame))

new_signal = np.concatenate(frames)
return new_signal

def extract_features(signal, fs):
    new_fs = 50

    signal = signal.flatten()
    signal = butterworth_filterig(signal, (25, 400), 2, fs, mode='bandpass')

    homomorphic_env = homomorphic_envelope(signal, fs)
    homomorphic_env = downsampling(homomorphic_env, fs, new_fs)
    homomorphic_env = normalization(homomorphic_env)

    hilbert_env = get_envelope(signal)
    hilbert_env = downsampling(hilbert_env, fs, new_fs)
    hilbert_env = normalization(hilbert_env)

    psd_env = psd_envelope(signal, fs)
    psd_env = downsampling(psd_env, fs, new_fs)
    psd_env = normalization(psd_env)

    wavelet = wavelet_envelope(signal)
    wavelet = downsampling(wavelet, fs, new_fs)
    wavelet = normalization(wavelet)

    return (homomorphic_env, hilbert_env, psd_env, wavelet)

def reshape(ar):

```

Змн.	Арк.	№ докум.	Підпис	Дата

```

temp = np.zeros((ar.shape[1], ar.shape[2], ar.shape[0]))
for i in range(temp.shape[0]):
    temp[i] = ar[:, i, :].T
return temp

```

```

def slice_signal(pcg, slice_length, step):
    slices = []

    for start_index in np.arange(0, len(pcg), step):
        slices.append(pcg[start_index:min(len(pcg), start_index+slice_length)])
    return slices

```

```

def get_features(signal, segm, fs):
    signal = np.asarray(signal)
    indx = np.where(np.abs(np.diff(segm)) > 0)[0]

```

```

    K = None
    if segm[indx[0] + 1] == 3:
        K = 0
    elif segm[indx[0] + 1] == 2:
        K = 1
    elif segm[indx[0] + 1] == 1:
        K = 2
    elif segm[indx[0] + 1] == 0:
        K = 3
    indx = indx[K:]
    indx = indx[:len(indx) - (len(indx) % 4)]
    indx = indx.reshape(len(indx) // 4, 4)

```

```

    m_RR = np.mean(np.diff(indx[:, 0]))
    std_RR = np.std(np.diff(indx[:, 0]))
    m_S1 = np.mean(indx[:, 1] - indx[:, 0])
    std_S1 = np.std(indx[:, 1] - indx[:, 0])
    m_S2 = np.mean(indx[:, 3] - indx[:, 2])
    std_S2 = np.std(indx[:, 3] - indx[:, 2])
    m_Sys = np.mean(indx[:, 2] - indx[:, 1])
    std_Sys = np.std(indx[:, 2] - indx[:, 1])
    m_Dia = np.mean(indx[1:, 3] - indx[:-1, 0])
    std_Dia = np.std(indx[1:, 3] - indx[:-1, 0])

```

```

    R_sysRR = np.divide(indx[:-1, 2] - indx[:-1, 1], np.diff(indx[:, 0]))
    R_diaRR = np.divide(indx[1:, 3] - indx[:-1, 0], np.diff(indx[:, 0]))
    R_sysDia = np.divide(indx[:-1, 2] - indx[:-1, 1], indx[1:, 3] - indx[:-1, 0])

```

```

    m_Ration_sysRR = np.mean(R_sysRR)
    std_Ration_sysRR = np.std(R_sysRR)
    m_Ration_diaRR = np.mean(R_diaRR)
    std_Ration_diaRR = np.std(R_diaRR)

```

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

```

m_Ration_sysDia = np.mean(R_sysDia)
std_Ration_sysDia = np.std(R_sysDia)

P_S1 = np.zeros(indx.shape[0] - 1)
P_Sys = np.zeros(indx.shape[0] - 1)
P_S2 = np.zeros(indx.shape[0] - 1)
P_Dia = np.zeros(indx.shape[0] - 1)
for i in range(indx.shape[0] - 1):
    P_S1[i] = signal[indx[i, 0]:indx[i, 1]].sum() / (indx[i, 1] - indx[i, 0])
    P_Sys[i] = signal[indx[i, 1]:indx[i, 2]].sum() / (indx[i, 2] - indx[i, 1])
    P_S2[i] = signal[indx[i, 2]:indx[i, 3]].sum() / (indx[i, 3] - indx[i, 2])
    P_Dia[i] = signal[indx[i, 3]:indx[i+1, 0]].sum() / (indx[i, 3] - indx[i+1, 0])
P_SysS1 = np.divide(P_Sys, P_S1)
P_DiaS2 = np.divide(P_Dia, P_S2)

if not any(np.isnan(P_SysS1)) and not any(np.isinf(P_SysS1)) and len(P_SysS1) > 1:
    m_Amp_SysS1 = np.mean(P_SysS1)
    std_Amp_SysS1 = np.std(P_SysS1)
else:
    m_Amp_SysS1 = 0
    std_Amp_SysS1 = 0
if not any(np.isnan(P_DiaS2)) and not any(np.isinf(P_DiaS2)) and len(P_DiaS2) > 1:
    m_Amp_DiaS2 = np.mean(P_DiaS2)
    std_Amp_DiaS2 = np.std(P_DiaS2)
else:
    m_Amp_DiaS2 = 0
    std_Amp_DiaS2 = 0

freq_bands = ((25,45), (45,65), (65,85), (85,105), (105,125), (125,150), (150,200), (200,300), (300,500))
NFFT = 512
f = NFFT / fs
P_S1 = np.ndarray((indx.shape[0], NFFT // 2))
P_sys = np.ndarray((indx.shape[0], NFFT // 2))
P_S2 = np.ndarray((indx.shape[0], NFFT // 2))
P_dia = np.ndarray((indx.shape[0], NFFT // 2))

for i in range(len(indx) - 1):
    P_S1[i] = np.abs(scipy.fftpack.fft(signal[indx[i, 0] * 40: indx[i, 1] * 40], NFFT))[:NFFT // 2]
    P_sys[i] = np.abs(scipy.fftpack.fft(signal[indx[i, 1] * 40: indx[i, 2] * 40], NFFT))[:NFFT // 2]
    P_S2[i] = np.abs(scipy.fftpack.fft(signal[indx[i, 2] * 40: indx[i, 3] * 40], NFFT))[:NFFT // 2]
    P_dia[i] = np.abs(scipy.fftpack.fft(signal[indx[i, 3] * 40: indx[i+1, 0] * 40], NFFT))[:NFFT // 2]

p_S1 = np.ndarray(len(freq_bands))
p_sys = np.ndarray(len(freq_bands))
p_S2 = np.ndarray(len(freq_bands))
p_dia = np.ndarray(len(freq_bands))
for i, band in enumerate(freq_bands):
    p_S1[i] = np.median(P_S1[:, int(band[0] * f):int(band[1] * f)])
    p_sys[i] = np.median(P_sys[:, int(band[0] * f):int(band[1] * f)])

```

					ДП ІС-6105.00.1081-с.ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

```

p_S2[i] = np.median(P_S2[:, int(band[0] * f):int(band[1] * f)])
p_dia[i] = np.median(P_dia[:, int(band[0] * f):int(band[1] * f)])

p_S1[np.isnan(p_S1)] = 0
p_sys[np.isnan(p_sys)] = 0
p_S2[np.isnan(p_S2)] = 0
p_dia[np.isnan(p_dia)] = 0

return (m_RR, std_RR, m_S1, std_S1, m_S2, std_S2, m_Sys, std_Sys, m_Dia, std_Dia, m_Ration_sysR
R, std_Ration_sysRR, \
        m_Ration_diaRR, std_Ration_diaRR, m_Ration_sysDia, std_Ration_sysDia, m_Amp_SysS1, std_A
mp_SysS1, m_Amp_DiaS2, \
        std_Amp_DiaS2, *p_S1.tolist(), *p_sys.tolist(), *p_S2.tolist(), *p_dia.tolist())

class SignalProcessor:
    def __init__(self):
        self._model = models.load_model('../models/best_model.h5')

    def get_segmentation(self, signal, fs):
        signal = np.asarray(signal)
        envs = extract_features(signal, fs)
        sliced_envs = [[] for _ in range(len(envs))]
        pad_len = 1

        for i, env in enumerate(envs):
            sliced_envs[i] = slice_signal(env, 512, 512)
            for j in range(len(sliced_envs[i])):
                if len(sliced_envs[i][j]) != 512:
                    ar = np.zeros(512)
                    pad_len = ar.shape[0] - len(sliced_envs[i][j])
                    ar[:len(sliced_envs[i][j])] = sliced_envs[i][j]
                    sliced_envs[i][j] = ar
            inp = reshape(np.asarray(sliced_envs))
            predictions = self._model.predict(inp)
            return signal[:, fs // 50], np.concatenate(predictions, axis=0)[-pad_len:]

    def get_classification(self, signal, segm, fs):
        features = get_features(signal, segm, fs)
        with open('../models/boosting.pkl', 'rb') as file:
            model = pickle.load(file)

        pred = model.predict(np.asarray(features).reshape(1, -1))[0]
        if pred == "Abnormal":
            pred = "Наявні аномалії"
        elif pred == "Normal":
            pred = "Норма"
        return pred

    def get_heart_rate(self, signal, fs):

```

```

signal = np.asarray(signal)
new_fs = 50
env = homomorphic_envelope(signal.flatten(), fs)
env = downsampling(env, fs, new_fs)
xcorr = np.asarray([np.correlate(env, np.roll(env, i)) for i in range(len(env))])
index = np.argmax(xcorr[int(0.5*new_fs):int(2*new_fs)]) + 0.5 * new_fs
hr = 60 / (index / new_fs)

max_sys_duration = (60 / hr)* new_fs / 2
min_sys_duration = 0.2 * new_fs

pos = np.argmax(xcorr[int(min_sys_duration):int(max_sys_duration)])
sys_time_interval = (min_sys_duration + pos) / new_fs
return hr

```

					<i>ДП ІС-6105.00.1081-с.ПЗ</i>	Арк.
						54
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

ЗАТВЕРДЖУЮ

Керівник проєкту

Завідувач кафедри

_____ Гавриленко О.В.
(підпис) (ініціали, прізвище)

_____ О.А. Павлов
(підпис) (ініціали, прізвище)

“___” _____2020 р.

“___” _____2020 р.

« Комплекс задач з аналізу фонокардіографічного сигналу »

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр ДП ІС-6105.1153-с.ТЗ

на 8 сторінках

Київ – 2020 року

1 Загальні положення

1.1 Повне найменування системи та умовне позначення

Повна назва системи: Комплекс задач з аналізу фонокардіографічного сигналу

1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником є кафедра Автоматизованих систем обробки інформації та управління НТУУ "КПІ". Представником замовника є Гавриленко Олена Валеріївна

Розробниками системи є студенти групи ІС-61 факультету інформатики та обчислювальної техніки НТУУ «КПІ ім. Ігоря Сікорського» Бусов Ілля

1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку роботи – 1 березня 2020 року

					ДП ІС-6105.01.1081-с.ТЗ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Плановий термін закінчення роботи за створення системи – 1 червня 2020 року

2 Призначення і цілі створення комплексу задач

2.2 Призначення комплексу задач

Даний комплекс задач присвячений автоматизації аналізу фонокардіографічного сигналу

2.3 Цілі створення системи

Цілі розробки

- Пришвидшити процес аналізу фонокардіограми
- Зробити процес діагностування хвороби серця доступним без участі лікарі

Для досягнення цих цілей необхідно розв'язати такі задачі

- Задачу сегментації сигналу
- Задачу класифікації сигналу

					ДП ІС-6105.01.1081-с.ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

3 Характеристика об'єкту автоматизації

Об'єктом автоматизації є аналіз фонокардіограми, що включає в себе виділення основних відомих звуків серця, дослідження їх характеристик та висування діагнозу на основі цих характеристик.

					ДП ІС-6105.01.1081-с.ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

4 Вимоги до програмного забезпечення

4.2 Вимоги до функціональних характеристик

Система має підтримувати наступні функції

- Завантаження звукового файлу формату .wav на сервер
- Проводити сегментацію та класифікацію сигналу та повертати результат формату json файлу
- Виводити зображення із візуалізацією сегментації сигналу та сповіщати користувача про результат класифікації

4.3 Вимоги до надійності

У разі наступних варіантів використання програма має продовжувати функціонування або виводити повідомлення про помилку

- Завантаження файлу з розширення відмінного від .wav
- Завантаження .wav файлу, який не містить сигналу фонокардіограми
- Завантаження сигналу недостатньої якості, неможливо провести коректний аналіз
- Помилки пов'язані з не підтримкою браузером використаних технологій

У цих випадках користувач має отримати повідомлення про характер помилки та інструкції щодо подальших дій.

4.4 Вимоги до складу і параметрів технічних засобів

Для коректної роботи розробленого програмного продукту необхідна наявність комп'ютера з наступними мінімальними характеристикам

					ДП ІС-6105.01.1081-с.ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

- Оперативна пам'ять 2 Гб
- Тактова частота процесора 1 ГГц
- Наявність клавіатури та миші
- Встановлений браузер Google Chrome або Mozilla Firefox останньою версії

					<i>ДП ІС-6105.01.1081-с.ТЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

5 Стадії та етапи розробки

№ п/п	Назва етапу роботи	Термін виконання етапу	Результат виконання
1.	Підготовка технічного завдання на розробку програмного продукту	15.03.2020	
2.	Розробка сценарію роботи	05.04.2020	
3.	Технічне проектування – функціональність, модулі, задачі, цілі тощо	10.04.2020	
5.	Розробка інформаційного забезпечення	15.04.2020	
6.	Розробка програмного забезпечення	30.04.2020	
7.	Налагодження програми	5.05.2020	
8.	Тестування програми	7.05.2020	
9.	Здача готового програмного продукту замовнику	01.06.2020	

6 Порядок контролю та приймання

6.2 Види випробувань

Тестування програмного забезпечення передбачає випробування різних варіантів використання та отримання результату, що відповідає описаним вимогам

Тестування завантаження звукового файлу. Завантаження файлу .wav із сигналом фонокардіограми на сервер

Тестування сегментації сигналу. Відправка на сервер json файлу зі значеннями амплітуди сигналу та частотою дискретизації

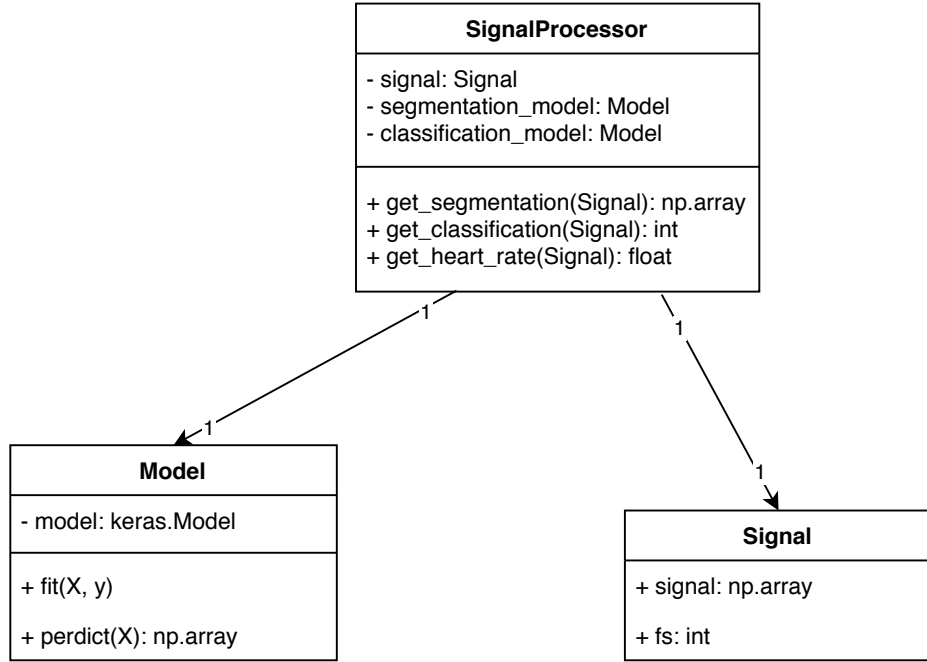
Тестування класифікації сигналу. Відправка на сервер json файлу зі значеннями амплітуди сигналу, сегментацією сигналу та частотою дискретизації

					ДП ІС-6105.01.1081-с.ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

Графічний матеріал до дипломного проєкту

на тему: Комплекс задач з аналізу фонокардіографічного сигналу

Київ – 2020 року



Зм.	Арк.	№ документа	Підпис	Дата
Розробив		Бусов І.О.		
Перевірив		Гавриленко О.В		
Т. кон.				
Н. кон.		Тєлишева Т.О		
Затвердив		Гавриленко О.В		

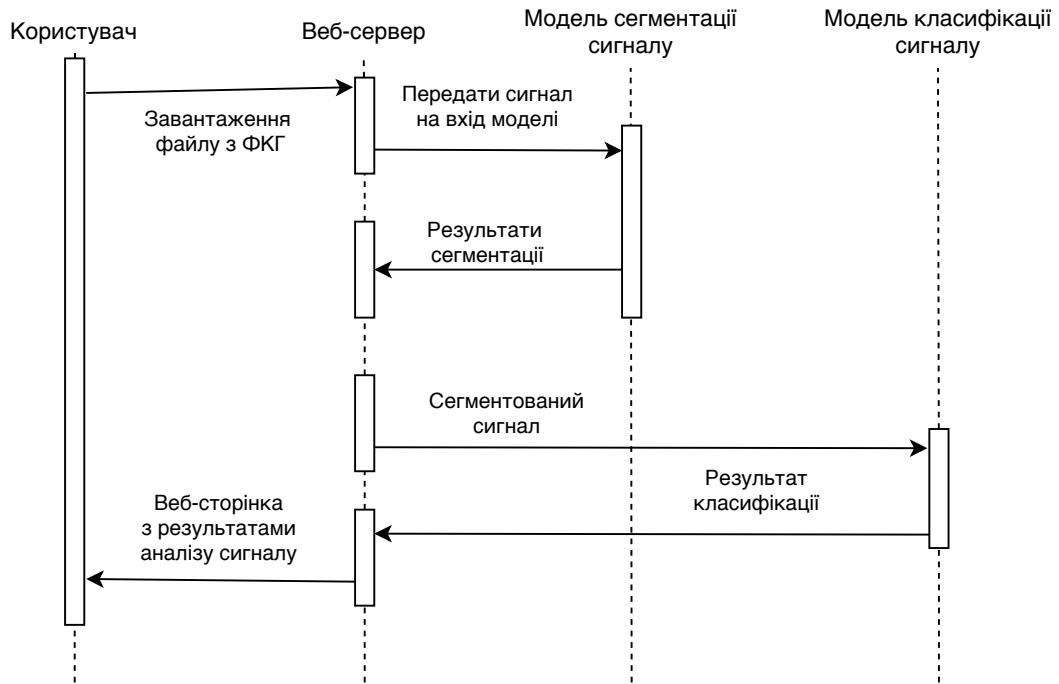
ДП ІС-6105.02.1081-с.ССК

Структурна схема класів програмного забезпечення

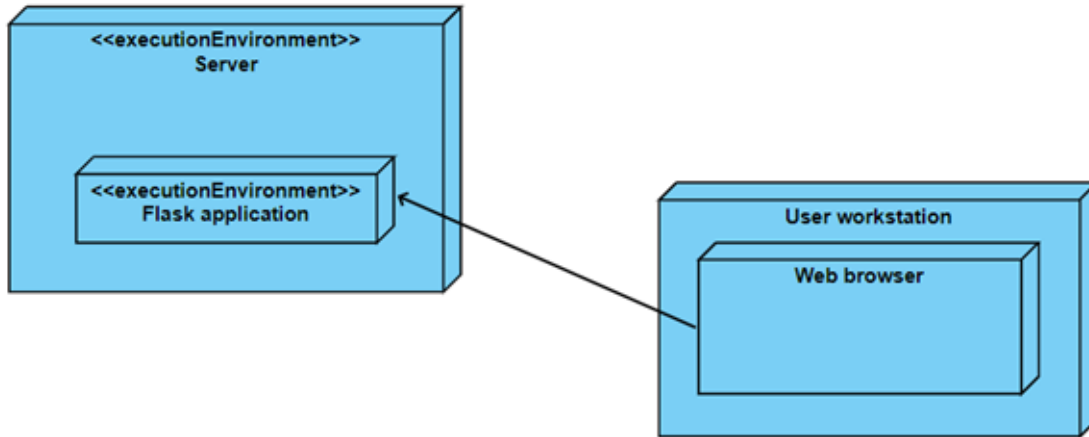
Комплекс задач з аналізу фонокардіографічного сигналу

Літера	Маса	Масштаб
Аркуш 1	Аркушів 1	

КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61



					<i>ДП ІС-6105.03.1081-с.ССП</i>		
					<i>Структурна схема послідовності</i>		
Зм.	Арк.	№ документа	Підпис	Дата	Літера	Маса	Масштаб
Розробив		Бусов І.О					
Перевірив		Гавриленко О.В			Аркуш 1		Аркушів 1
Т. кон.					Назва диплому КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61		
Н. кон.		Телишева Т.О.					
Затвердив		Гавриленко О.В					



					ДП ІС-6105.04.1081-с.ССР			
						Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата	Структурна схема розгортання			
Розробив	Бусов І.О.							
Перевірив	Гавриленко О.В.					Аркуш 1	Аркушів 1	
Т. кон.								
Н. кон.	Телишева Т.О.							
Затвердив	Гавриленко О.В.				Комплекс задач з аналізу фонокардіографічного сигналу	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61		

