

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДІВ ДІАГНОСТИКИ ТА ЛІКУВАННЯ
ЗАХВОРЮВАННЯ ЗА ДОПОМОГОЮ ГЛИБОКОГО НАВЧАННЯ
ДЛЯ СИСТЕМИ ОХОРОНИ ЗДОРОВ'Я

(тема)

Виконав:

студент 2 курсу, групи ІНФМ-21-1

Берковський Д.С.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Кузьомін О.Я.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

Харків 2022 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Берковському Денису Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів діагностики та лікування захворювання за допомогою глибокого навчання для системи охорони здоров'я

затверджена наказом по університету від 9 листопада 2022 року № 1469Ст

2. Термін подання студентом роботи до екзаменаційної комісії 2 грудня 2022 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, математичні моделі розпізнавання захворювань, перелік використовуваних програмних засобів, теоретичні відомості про методи глибокого навчання.

Вихідні дані до роботи науково-методична та науково-технічна література, математичні моделі розпізнавання захворювань, перелік використовуваних програмних засобів, теоретичні відомості про методи глибокого навчання

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Проведення аналізу існуючих методів діагностування захворювань. _____

2. Проведення аналізу існуючих моделей розпізнавання захворювань. _____

3. Розробка програмного застосунок розпізнавання захворювань. _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність дослідження, мета дослідження, постановка задачі, визначення методів, створення системи, перспективи, висновки, апробація.

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	9.11.2022	
2	Аналіз завдання, підбір літератури	9.11.22-11.11.22	
3	Аналіз літератури з досліджуваної проблеми	11.11.22-13.11.22	
4	Аналіз проблеми	13.11.22-14.11.22	
5	Дослідження методів	14.11.22-19.11.22	
6	Аналіз методів	19.11.22-25.11.22	
7	Оформлення пояснювальної записки	25.11.22-1.12.22	
8	Перевірка на плагіат	2.12.22	
9	Рецензування	7.12.22	
10	Підготовка презентації та доповіді	8.12.22-10.12.22	
11	Занесення роботи в електронний архів	11.12.22	
12	Попередній захист кваліфікаційної роботи	12.12.22	

Дата видачі завдання 9 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Кузьомін О.Я.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 60 с., 21 рис., 56 джерел.

ШТУЧНИЙ ІНТЕЛЕКТ, РЕНТГЕН, НЕЙРОННА МЕРЕЖА, ДИНАМІЧНЕ ПРОГРАМУВАННЯ, ВИСОКОЧАСТОТНІ ФІЛЬТРИ, ЗАХВОРЮВАННЯ ЛЕГЕНІВ.

Об'єктом дослідження є різні рентгеновські снимки грудної клітки.

Метою дослідження є розробка методів, які дозволяють розпізнавати захворювання легенів за ознаками зображення.

Використано методи обробки зображень. Проведен аналіз різноманітних методів перевірки рентгеновських та інших зображень на захворювання.

У результаті дослідження здійснена програмна реалізація системи для аналізу захворювань легенів.

ARTIFICIAL INTELLIGENCE, X-RAY, NEURAL NETWORK, DYNAMIC PROGRAMMING, HIGH-FREQUENCY FILTERS, LUNG DISEASES.

The object of the research is various x-ray images of the chest.

The purpose of the research is to develop methods that allow to recognize lung diseases based on image features.

Image processing methods are used. An analysis of various methods of checking X-ray and other images for diseases was carried out.

As a result of the research, the software implementation of the system for lung analysis was carried out.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1 Штучний інтелект та охорона здоров'я.....	8
1.1 Застосування штучного інтелекту у системах охорони здоров'я.....	8
1.2 Глибоке навчання для виявлення захворювань очей при діабеті (діабетична ретинопатія)	10
1.3 CNN для виявлення пухлин мозку.....	14
1.4 Використання CNN для діабетичної ретинопатії.....	15
1.5 Постановка задачі дослідження	16
2 Моделі навчання в галузі охорони здоров'я.....	17
2.1 Розробки моделі навчання.....	17
2.2 Інтегрована модель на основі LOG і RF	34
2.3 FCMIM-SVM	36
2.4 HMANN	37
2.5 SRL-RNN	39
2.6 Схема обробки із замкнутим циклом охорони здоров'я.....	41
2.7 GAN + RAE + DQN.....	43
2.8 HQLA.....	45
2.9 tVAE.....	46
3 Комп'ютерна модель фільтрації зображень.....	48
3.1 Обґрунтування вибору середовища програмної реалізації	48
3.2 Програмна реалізація	49
Висновки.....	54
Перелік джерел посилання	55

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект

ДР – діабетичної ретинопатії

ML – Machine Learning (машинне навчання)

DL – Deep Learning (глибоке навчання)

ANN – Artificial Neural Networks (штучні нейронні мережі)

CNN – Convolutional Neural Network (згорткова нейронна мережа)

ЗНТ – значення насиченості тону

TP – True Positive (істинно позитивний)

TN – True Negative (істинно негативний)

FP – False Positive (помилково позитивний)

FN – False Negative (помилково негативний)

TPR – True Positive Rate (справжній позитивний коефіцієнт)

TFR – True False Rate (справжній негативний коефіцієнт)

ККМ – коефіцієнт кореляції Метьюза

RMSE – Root-mean-square Error(середньоквадратична помилка)

ВСТУП

У наш час штучний інтелект використовується майже у всіх сферах, до яких причетна людина. Він допомагає створювати нові та вдосконалювати старі принципи. Його можна побачити і у повсякденному житті, наприклад пошук у мережі інтернет виконується за принципами глибокого навчання.

Штучний інтелект також присутній і у системах охорони здоров'я. Він використовується для завдань стандартизації, діагностики, систематики захворювань. Він допомагає лікарям зберегти дуже багато часу, що, у свою чергу, допомагає врятувати життя більшій кількості пацієнтів.

Глибоке навчання (DL) – одна з гілок штучного інтелекту, яка за останні роки демонструє експоненційне зростання. Наукове співтовариство зосередило свою увагу на DL завдяки його універсальності, високій продуктивності, високій здатності до узагальнення та міждисциплінарному використанню, серед багатьох інших якостей. Крім того, велика кількість медичних даних і розробка більш потужних комп'ютерів також сприяли інтересу до цієї галузі.

У діагностиці штучний інтелект використовують для знаходження різноманітних захворювань мозку, кісток, печінки та багатьох інших.

Системи штучного інтелекту також використовуються при знаходженні хвороб легенів та раку на ранніх стадіях.

Актуальність дослідження полягає у тому, що зараз лікарі все ще більше покладаються на свій зір, а не на системи у комп'ютерах. Якщо розробити таке програмне забезпечення, яке допоможе без людини швидко діагностувати різноманітні захворювання та впровадити його, це дозволить врятувати велику кількість життів. Також, якщо натренувати програму на визначення не лише однієї задачі, це допоможе ще більше спростити діагностування.

1 ШТУЧНИЙ ІНТЕЛЕКТ ТА ОХОРОНА ЗДОРОВ'Я

1.1 Застосування штучного інтелекту у системах охорони здоров'я

Штучний інтелект (ШІ) використовується у різних застосуваннях системи охорони здоров'я.

Основними областями ШІ є раннє виявлення, діагностика та лікування з наступним прогнозуванням та оцінкою прогнозу.

Прогноз будь-яких захворювань здійснюється за алгоритмами ANN та CNN. Алгоритми машинного навчання (ML), а саме регресійного аналізу, довели, що вони аналізують логістичний та довірчий інтервали захворювання. Алгоритмам машинного навчання в основному потрібні деякі дані для використання у якості навчального набору [1].

Для прогнозування та лікування аналіз захворювань проводили за допомогою ML [2]. ANN використовується для прогнозування захворювань з відомими та невідомими симптомами. У цьому дослідженні наведені результати дослідження діабетичної ретинопатії (ДР) з використанням Inception та CNN. Також було реалізовано приклад виявлення пухлини за допомогою CNN.

Методи DL були реалізовані для аналізу біосигналів та використанні більшого числа додатків для класифікації [3].

Методи DL використовуються у охороні здоров'я, промисловості та наукових дослідженнях [4] та використовуються для класифікації зображень, виявлення різних об'єктів, сегментації, реєстрації та інших завдань [5].

Дослідження включають області застосування, такі як нервова, сітчаточна, легенева, цифрова патології, молочної залози, серцевої, черепної порожнина та дослідження опорно-рухового апарату.

Для прогнозування серцево-судинних захворювань з використовувались дерева рішень SVM для логістичної регресії. Метод заснований на шести

модулях, включають попередню обробку, навчання, тестування на окремих моделях, правила застосування та порівняння результатів для прогнозування серцевих захворювань. База даних була зібрана з репозиторію Cleveland Heart Disease Dataset UCI з 13 ознаками, такими як вік, стать, тип болю в грудях, артеріальний тиск у спокої, рівень холестерину в сироватці в мг/дл, рівень цукру в крові натще більше 120 мг/дл, результат електрокардіографії у спокої, максимум досягнута частота серцевих скорочень, стенокардія, спричинена фізичним навантаженням, депресія сегмента ST, спричинена фізичною навантаженням, у порівнянні зі станом спокою, нахил пікового сегмента ST фізичного навантаження та кількість великих судин (0-3), пофарбовані при рентгеноскопії.

Структура показана на рисунку 1.1.

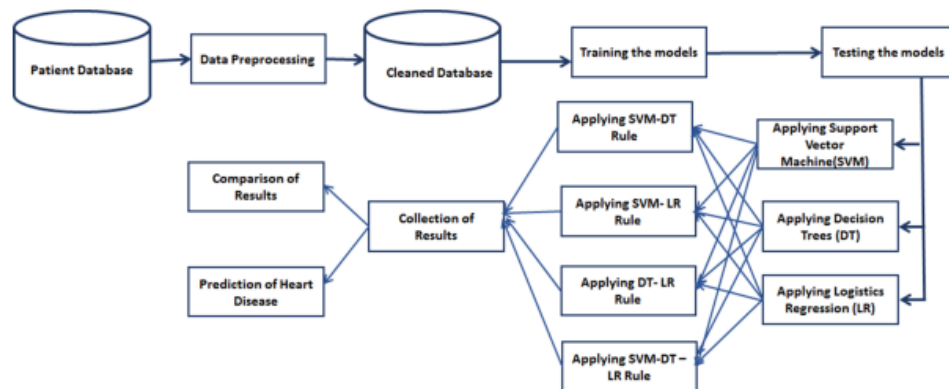


Рисунок 1.1 – Структура для прогнозування захворювань серця [6]

Всі ці функції допомагають прогнозувати захворювання та навчають необхідну модель, будується регресійна модель та модель дерева рішень. Результати показали, що існує потреба в більш складних моделях та комбінаціях моделей для покращення продуктивності та підвищення точності прогнозування серцево-судинних захворювань. Система відома як інтелектуальна з величезним обсягом даних, які завантажуються в системну базу даних.

1.2 Глибоке навчання для виявлення захворювань очей при діабеті (діабетична ретинопатія)

У дослідженні діабетичної ретинопатії (ДР) Варун Гульшан (2016) використовував алгоритм оптимізації зворотним поширенням, алгоритм глибокого навчання для виявлення ДР [7]. Ця хвороба відома тим, що є однією яка викликає найбільш швидке зростання сліпоти серед хворих на діабет. Майже 415 мільйонів діабетиків хворі у всьому світі. Це захворювання піддається лікуванню; інакше це також може призвести до необоротної сліпоти. Автори отримали 128 000 зображень із бази даних різних лікарень як набір даних, оцінено 3-7 офтальмологами (ОП) із групи з 54 ОП. Цей набір даних використовувався для навчання глибини алгоритма ШІ для виявлення ДР. Зображення наборів перевірки були оцінені ОП на наявність ДР, діабетичного набряку макули та якості зображення за допомогою інструмента анотації. За матеріалами Міжнародної шкали клінічної діабетичної ретинопатії, тяжкість ДР класифікувалася як відсутність, легка, помірна, важка або проліферативна, як показано на рисунку 1.2.

	E1	E2	E3	E4
No. of class	5	4	5	4
Img /class	500	500	800	800
Img radi (px)	200	200	500	500
Train steps	500	4000	500	500
Acc. of Training (%)	61	63	58	58
Acc. of Validation (%)	36	55	50	59
Final test acc. (%)	37.6	47.2	48.2	48.8

Рисунок 1.2 – Набір даних із кількістю доступних зображень [7]

ОП, сертифіковані радою США з найвищим рівнем самоузгодженості, були запрошено оцінити набори для клінічної перевірки *EyePACS-1* ($n = 8$) і *Messidor-2* ($n = 7$).

Нейронна мережа глибокого навчання (DL) обробляє велику математичну функцію з мільйонами параметрів для виконання заданого

завдання. Для навчання цієї функції потрібен великий набір зображень, для яких DL є ступінь тяжкості відомий як навчальний набір.

Кожне зображення надається функцією зі ступенем серйозності та порівнюється з відомим ступенем із навчального набору, а параметри функції змінюються, щоб зменшити помилку на цьому зображенні. Приклад діабетичної ретинопатії показаний на рисунку 1.3.



Рисунок 1.3 – Діабетична ретинопатія

Процес повторюється для кожного зображення в навчальному наборі багато разів, і функція вчиться точно обчислювати серйозність ДР на основі інтенсивності пікселів зображення для всіх зображень у навчальному наборі.

Безперервне число від 0 до 1 для ДР та інших класифікацій ДР генерується навченою нейронною мережою. Побудова робочих кривих приймача шляхом варіювання порогу спрацьовування та набору прояву та 2 робочих точки були обрані для алгоритму. Перша робоча точка оцінює специфічність ОР з виведення набору для виявлення реферабельного ДР (приблизно 98%), і виявлено, що продуктивність алгоритму дає кращі результати. Друга робоча точка відповідала чутливості 97% для виявлення реферабельної ДР, оскільки висока чутливість є необхідною умовою для потенційного інструменту скринінгу. Під час оцінювання алгоритм базується на глибокому машинному навчанні, що забезпечує високу чутливість і специфічність для виявлення реферального ДР.

В іншому прикладі [7] продуктивність згорткових нейронних мереж (CNN) була винятковою для розпізнавання різних реальних об'єктів. Попередньо навчені моделі виконують класифікацію об'єктів із зображення об'єкта, яке надається як вхідні дані.

Однак попередньо навчені моделі перенавчаються для набору класів; модель зберігає для вимірювання тяжкості ДР на зображеннях очей шкалу 0-5.

Класифікатор CNN, створений на основі мережі Inception V3, навчений для ImageNet, для 5-класу серйозності, як на рисунку 1.4, класифікація показала найкращі результати з точністю 48,2%.

Class	Number
0 (no DR)	25810
1 (mild)	2443
2 (average)	5292
3 (extreme)	873
4 (proliferate DR)	708

Рисунок 1.4 – Результати

Процедура складається з двох основних етапів, а саме попередньої обробки зображення та повторного навчання Inception V3, як на рисунку 1.5.

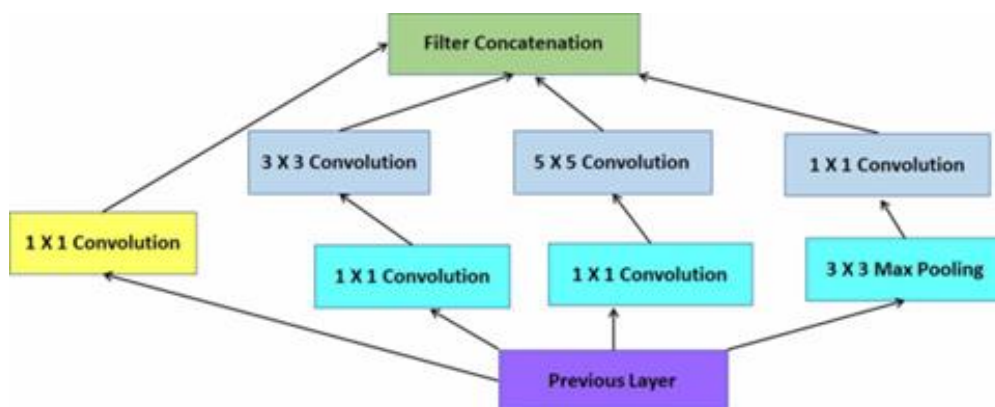


Рисунок 1.5 – Навчання Inception V3

Попередню обробку зображення можна виконати за допомогою трьох основних підзадач:

- зображення зменшуються до певного радіусу;
- видалення локального середнього кольору із зображень;
- межі зображення обрізані.

Модель Inception V3 використовується для досягнення результатів, як показано на рисунку 1.4.

Завдяки глибшій архітектурі та збільшеному розміру як по ширині, так і по висоті, його використовують для ДР.

Рисунок 1.6 описує рівні архітектури Inception із розміром виводу, параметрами та кроками.

Це є повним результатом початкової моделі з параметром.

Type	Pat_size Stride	O/P Size	Dep	1_1	3_3 Reduce	3_3	5_5 Reduce	5_5	Pool	Params	ops
Conv	7 X 7 / 2	112 X 112 X 64	1							27 K	34 M
Max- pool	3 X 3 / 2	56 X 56 X 64	0								
Conv	3 X 3 / 1	56 X 56 X 192	2		64	192				112 K	360 M
Max- pool	3 X 3 / 2	28 X 28 X 192	0								
Incep 3-a		28 X 28 X 256	2	64	96	128	16	32	32	159 K	128 M
Incep 3-b		28 X 28 X 480	2	128	128	192	32	96	64	380 K	304 M
Max- pool	3 X 3 / 2	14 X 14 X 480	0								
Incep 4-a		14 X 14 X 512	2	192	96	208	16	48	64	364 K	73 M
Incep 4-b		14 X 14 X 512	2	160	112	224	24	64	64	437 K	88 M
Incep 4-c		14 X 14 X 512	2	128	128	256	24	64	64	463 K	100 M
Incep 4-d		14 X 14 X 528	2	112	144	288	32	64	64	580 K	119 M
Incep 4-e		14 X 14 X 832	2	256	160	320	32	128	128	840 K	170 M
Max- pool	3 X 3 / 2	7 X 7 X 832	0								
Incep 5-a		7 X 7 X 832	2	256	160	320	32	128	128	1072 K	54 M
Incep 5-b		7 X 7 X 1024	2	384	192	384	48	128	128	1388 K	71 M
Avg- pool	7 X 7 / 1	1 X 1 X 1024	0								
Drop- 40%		1 X 1 X 1024	0								
Linear		1 X 1 X 1000	1							1000 K	1 M
Softmax		1 X 1 X 1000	0								

Рисунок 1.6 – Екземпляри GoogLeNet, отримані з архітектури Inception

1.3 CNN для виявлення пухлин мозку

Дослідження пухлини головного мозку можна провести на основі алгоритму CNN для виявлення об'єктів. Згенерована модель CNN показана на рисунку 1.7.

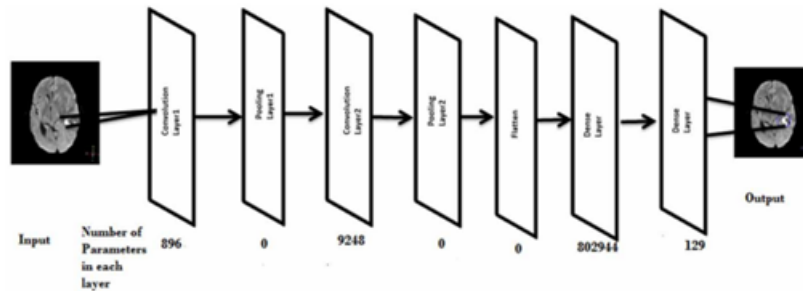


Рисунок 1.7 – Шари згорткової нейронної мережі для виявлення пухлини

Досягнуті результати складають 99,9%. Було використано 32 фільтри, розмір фільтра 3×3 , а шари conv2d, об'єднані, вирівняні та щільні. Розмір результату карти активації визначається як

$$(N + 2P - F) / \text{крок} + 1, \quad (1.1)$$

де N – розмір введення

P – відступ;

F – розмір фільтра та застосований крок на зображенні.

На рисунку 1.8 наведено деталі всіх параметрів, обчислених на кожному шарі.

Type	output shape	parameters
Conv	62 x 62x 32	896
Max-pool	31 x 31 x 32	0
Conv	29 x 29 x 32	9248
Max-pool	14 x 14x 32	0
Flatten	6272	0
Dense	128	802944
Dense	1	129
Total		813217

Рисунок 1.8 – Загальна кількість шарів CNN

Підсумок параметрів – це підсумовування параметрів усіх шарів дорівнює 896 плюс 0 плюс 9248 плюс 0 плюс 0 плюс 802944 та додання 129, що дорівнює 813217.

1.4 Використання CNN для діабетичної ретинопатії

Зображення отримуються з kaggle і для них застосовуються порогове значення насиченості тону (ЗНТ).

На рисунку 1.9 показано результати зображень сірого кольору, де зображення розмиті та розмиті межами, і це було виконано за допомогою коду Python.

Зображення в червоном-зеленим-синім кольором показують захворювання при ретинопатії та виділені на рисунку 1.9.

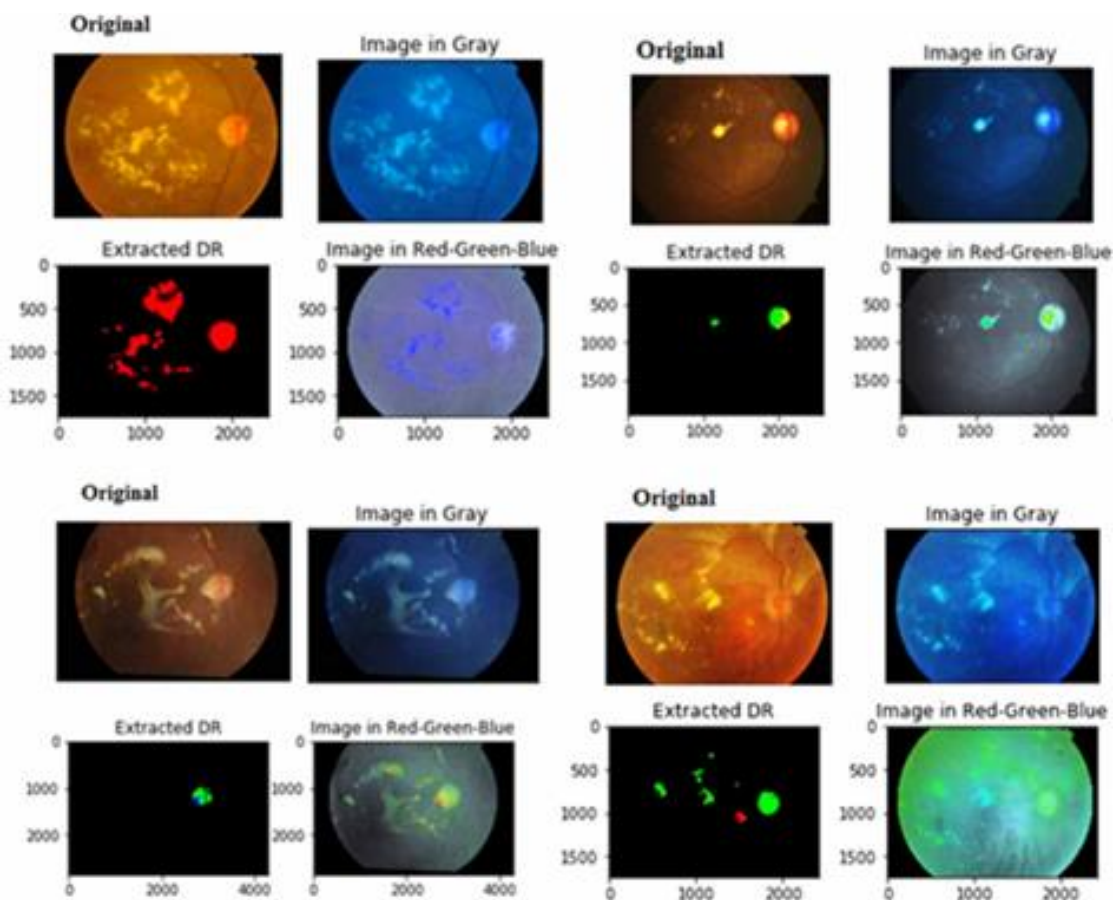


Рисунок 1.9 – Порогове виділення ЗНТ захворювання при ДР

Використовуючи колірний простір ЗНТ, можна отримати захворювання або пухлину, що допомагає виділити лише частину захворювання.

Що, у свою чергу, допомагає зрозуміти розмір і форму пухлини (захворювання).

1.5 Постановка задачі дослідження

Таким чином, застосування методів глибинного навчання у діагностиці різноманітних хвороб та уражень є актуальним. Тому ставиться завдання розробки алгоритму аналізу медичної діагностики рентгену грудної клітки застосовуючи методи глибокого навчання.

Об'єктом дослідження є різні рентгеновські знімки грудної клітки.

Метою дослідження є розробка методів, які дозволяють розпізнавати захворювання легенів за ознаками зображення.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів діагностування захворювань;
- провести аналіз існуючих моделей розпізнавання захворювань;
- розробити програмний застосунок розпізнавання захворювань.

2 МОДЕЛІ НАВЧАННЯ В ГАЛУЗІ ОХОРОНИ ЗДОРОВ'Я

2.1 Розробки моделі навчання

Для розробки моделі навчання в галузі охорони здоров'я можна розглянути п'ять основних етапів: визначення проблеми, набір даних, попередня обробка даних, розробка моделі ML та оцінка.

Розробляючи модель навчання в галузі охорони здоров'я, повинно спочатку відповісти на запитання: «Яка мета розробки цієї моделі навчання?» Щоб розробити корисну модель, першим кроком є визначення проблем і викликів у сфері охорони здоров'я.

Потрібно проаналізувати, як саме покращити медичні послуги за допомогою машинного навчання. На першому етапі ключовим моментом є перевірка наявності даних.

Це означає, що треба знати про існуючі джерела даних, оскільки даних має бути достатньо у доступі для розробки моделі навчання та оцінки цієї моделі.

У сфері охорони здоров'я брак даних може бути спричинений відсутністю цифрових даних, конфіденційністю пацієнтів, комерційними проблемами або рідкісними захворюваннями.

Під час розробки моделі навчання в галузі охорони здоров'я набори даних використовуються для навчання, перевірки та тестування.

Набори даних охорони здоров'я можуть включати демографічну інформацію, зображення, лабораторні результати, геномні дані та дані, отримані від датчиків [8, 9].

Для виробництва або збору цих даних використовуються різні платформи, наприклад, мережеві сервери, записи з електронної бази охорони здоров'я, дані про геноми, персональні комп'ютери, смартфони, мобільні програми та носимі пристрої [10, 11].

Перш ніж розробляти модель навчання в галузі охорони здоров'я, необхідно розробити відповідний механізм для оцінки моделі навчання. Моделі на основі ML орієнтовані на дані. Таким чином, можна зіткнутися з проблемою, яка називається надмірним або недостатнім обладнанням [12, 13]. Ефективна модель навчання повинна знаходити компроміс між надмірним і недостатнім оснащенням.

Це означає, що вона повинна мати відповідний ухил і правильну дисперсію. Недостатність виникає, коли ми розробляємо дуже просту модель навчання відносно складності проблеми та розміру набору даних. Ця навчальна модель має низьку продуктивність як для навчальних, так і для тестових наборів.

Це означає, що вона має велику упередженість. З іншого боку, переобладнання також виникає, коли модель навчання дуже складна і включає великі параметри щодо складності проблеми та розміру набору даних. У цьому випадку ця модель має хорошу продуктивність для навчального набору даних але має слабку продуктивність для тестового набору. У цьому випадку вона має високу дисперсію. Загалом, правильна модель навчання повинна мати низьку систематичну похибку та низьку дисперсію. Опис проблеми надмірного та недостатнього оснащення показаний на рисунку 2.1.

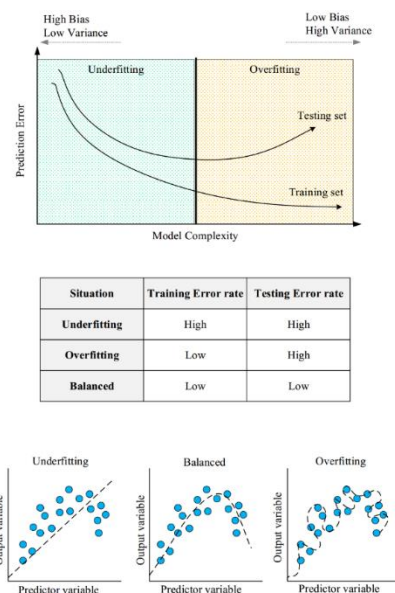


Рисунок 2.1– Розбір надмірного та недостатнього оснащення

Щоб запобігти переобладнанню, загальним рішенням є розділення набору даних на дві частини: набір для навчання та набір для тестування. «Набір для навчання» вказує на набір даних, який використовується для навчання моделі навчання та налаштування її параметрів. «Тестовий набір» також вказує на набір даних, який використовується для оцінки продуктивності моделі навчання. Зазвичай набір для навчання є більшим за набір для тестування, наприклад, з співвідношенням 70 до 30. Одним із рішень для вибору набору для навчання та набору для тестування є випадковий розподіл набору даних на дві частини. Іншим важливим моментом є те, що іноді набір даних невеликий. Таким чином, неможливо призначити частину набору даних лише для тестування. У цьому випадку використовується техніка K -кратної кросс валідації [14, 15]. У цій техніці набір даних ділиться на k секцій. Потім розділ використовується для тестування, а $k-1$ розділів використовується для навчання. Цей процес повторюється k разів, щоб на кожному кроці для тестування використовувався новий розділ. Потім ми повинні оцінити ефективність цієї моделі навчання на кожному кроці. Нарешті, загальна продуктивність моделі навчання дорівнює середній продуктивності за k кроків. K -кратна перехресна перевірка показана на рисунку 2.2.

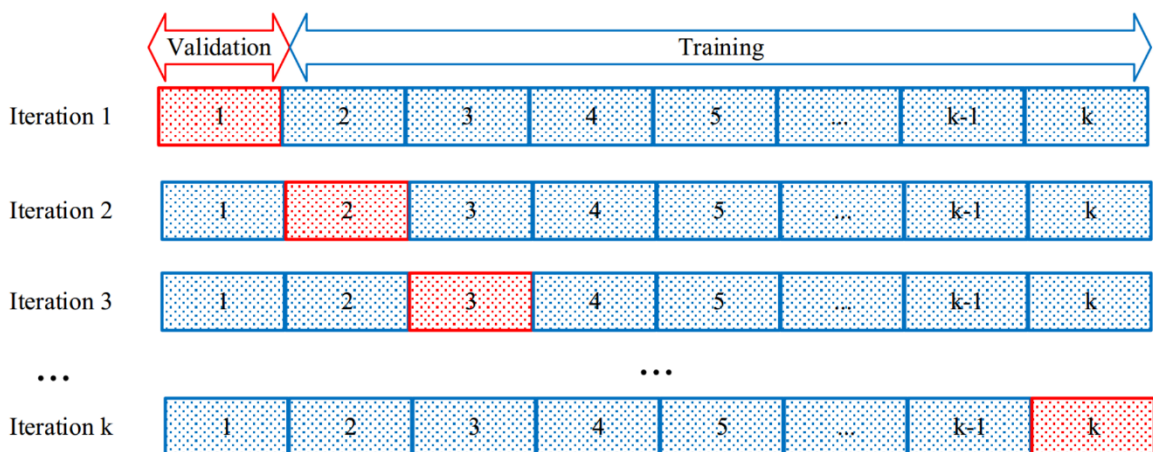


Рисунок 2.2 – K -кратна перехресна перевірка

Під час розробки моделі навчання в галузі охорони здоров'я однією з найскладніших проблем є попередня обробка даних, оскільки модель машинного навчання вимагає високоякісних даних для досягнення вищої якості в процесі навчання та більш відповідної продуктивності з точки зору точності.

Загалом, попередня обробка даних – це процес дослідження шумних даних, відсутніх значень, повторюваних даних і суперечливих даних.

Метою цього процесу є підвищення якості бази даних перед створенням моделі навчання. Тому під час попередньої обробки даних може знадобитися відфільтрувати викиди або оцінити відсутні значення.

Якщо дані також мають високі розміри, можна використовувати деякі методи зменшення даних, такі як вибір ознак [16, 17] або вилучення ознак [18]. Вибір функцій вибирає найкращу підмножину функцій. З іншого боку, вилучення ознак знаходить новий набір даних із меншими розмірами на основі початкового набору даних.

Розробляючи модель навчання в галузі охорони здоров'я, потрібно враховувати розмір бази даних, тип схеми навчання та час висновку моделі. Складність моделі навчання визначається на основі розміру бази даних, щоб уникнути надмірного або недостатнього оснащення.

Врахування часу навчання моделі є дуже важливим. Однак моделі навчання з більшою кількістю параметрів можуть дати більш точні результати.

Однак у цьому випадку ці моделі виконують більше обчислювальних операцій і потребують більшого часу для навчання. Як наслідок, їх не можна використовувати для програм реального часу. Таким чином, легкі архітектури є більш доречними для проектування нахиленої моделі. Розгляд типу схеми навчання також дуже важливий при розробці моделей ML [19, 20]. Загалом існує чотири основні методи навчання, включаючи навчання під наглядом, навчання без контролю, напівконтрольоване навчання та навчання з підкріпленням [21, 22].

Оцінка системи на основі машинного навчання означає виконання різних операцій для виявлення відмінностей між поточною поведінкою системи та очікуваною поведінкою [23].

Після розробки моделі навчання в охороні здоров'я слід провести необхідні оцінки, щоб визначити відповідь на запитання: «Чи ця модель має умови розгортання в реальних середовищах?»

У процесі оцінювання використовуються різні шкали для перевірки ефективності моделі навчання. Ця оцінка визначає її сильні та слабкі сторони.

Крім того, після розгортання моделі навчання в реальних середовищах потрібно переглянути продуктивність моделі навчання, щоб оцінити її поведінку під час взаємодії з реальними користувачами [24, 25]. Різні аспекти оцінки системи машинного навчання включають: оцінку даних, використаних для побудови кінцевої моделі навчання, оцінку алгоритмів навчання, які використовуються для розробки остаточної моделі, і оцінку продуктивності кінцевої моделі.

Під час розробки моделі навчання в галузі охорони здоров'я однією з найскладніших проблем є попередня обробка даних, оскільки модель машинного навчання вимагає високоякісних даних для досягнення вищої якості в процесі навчання та більш відповідної продуктивності з точки зору точності. Загалом, попередня обробка даних – це процес дослідження шумних даних, відсутніх значень, повторюваних даних і суперечливих даних.

Метою цього процесу є підвищення якості бази даних перед створенням моделі навчання.

Тому під час попередньої обробки даних може знадобитися відфільтрувати викиди або оцінити відсутні значення. Якщо дані також мають високі розміри, можна використовувати деякі методи зменшення даних, такі як вибір ознак [16, 17] або вилучення ознак [18].

Вибір функцій вибирає найкращу підмножину функцій. З іншого боку, вилучення ознак знаходить новий набір даних із меншими розмірами на основі початкового набору даних.

Оцінка даних, які використовуються для побудови кінцевої моделі навчання: продуктивність моделей навчання сильно залежить від даних.

Будь-яка помилка в даних може негативно вплинути на кінцеву модель і послабити її продуктивність. У процесі оцінки даних необхідно відповісти на різні запитання. Наприклад, чи достатньо даних для навчання та тестування моделі? Чи можна вважати існуючі дані репрезентативними для всіх реальних даних для конкретної території? Чи доступні дані збалансовані? Чи є в даних ворожа чи неправдива інформація?

Оцінка алгоритмів навчання, використаних для розробки остаточної моделі: на цьому кроці необхідно ретельно оцінити алгоритми навчання, які використовуються для створення остаточної моделі навчання, щоб визначити можливі помилки при проектуванні або виборі алгоритмів.

Наприклад, розробник повинен протестувати різні алгоритми навчання, щоб вибрати найбільш підходящий алгоритм для створення остаточної моделі. Якщо не виконається достатньо тестів для вибору належного алгоритму навчання, це може збільшити рівень помилок у кінцевій моделі навчання. Крім того, на цьому кроці ми можемо налаштувати різні параметри алгоритму навчання. Наприклад, параметри штучних нейронних мереж, такі як кількість нейронів у кожному шарі, кількість прихованих шарів і ваг мережі або параметри дерева рішень, включаючи кількість листів або його глибину.

Оцінка ефективності кінцевої моделі: після побудови та навчання остаточної моделі її продуктивність необхідно оцінити на основі таких факторів. Правильність: цей фактор оцінює, наскільки поточний результат системи навчання та очікувані результати близькі один до одного. У цій області існують шкали оцінювання.

Істинно позитивний (TP): кількість позитивних членів класу, які належним чином передбачені класифікатором і позначені як позитивний клас. Істинно негативний (TN): кількість негативних членів класу, які належним чином передбачені класифікатором і позначені як негативні класи.

Помилково позитивний результат (FP): кількість негативних членів класу, які помилково передбачені класифікатором і позначені як позитивний клас. Помилково негативний (FN): кількість позитивних членів класу, які помилково передбачені класифікатором і позначені як негативний клас. Чутливість: ця шкала визначається як ймовірність, щоб класифікатор дійсно передбачив результат як позитивний, коли відповідна базова правда також є позитивною. Інша назва цієї шкали – справжній позитивний коефіцієнт (TPR), і він розраховується як $TP / (TP + FN)$.

Специфічність: ця шкала визначається як ймовірність, щоб класифікатор справді передбачив результат як негативний, коли відповідна базова правда також є негативною. Інша назва специфічності – справжній негативний показник (TNR), і він розраховується як $TN / (TN + FP)$.

Позитивне прогнозоване значення (PPV): ця шкала визначається як ймовірність того, що класифікатор справді прогнозує результат як позитивний, коли результат тесту (вихід класифікатора) позитивний. Інша назва PPV – точність, і вона обчислюється як $TP / (TP + FP)$.

Негативне прогнозоване значення (NPV): ця шкала визначається як ймовірність, щоб класифікатор дійсно передбачив результат як негативний, коли результат тесту негативний. Ця шкала розраховується як $TN / (TN + FN)$.

Точність: цей масштаб дуже важливий. Зазвичай за цією шкалою оцінюють класифікатори. Він визначається як відсоток зразків, які дійсно були класифіковані класифікатором. Він розраховується як $(TP + TN) / (TP + TN + FP + FN)$.

Коефіцієнт кореляції Метьюза (КММ): він визначається як коефіцієнт кореляції між прогнозованим результатом і відповідною основною істинністю. Він має значення від +1 до -1. Якщо КММ=+1, то це означає, що класифікатор вірно прогнозує результат.

Якщо КММ=0, це означає, що класифікатор не може передбачити результат краще, ніж випадковим способом.

Якщо $KMM=-1$, то це означає, що існує повна суперечність між прогнозованим результатом і відповідною основною істинністю. Шкала KMM розраховується як

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN)}}. \quad (2.1)$$

Частота помилкових відкриттів (FDR): Ця шкала оцінює відношення зразків, які помилково передбачили як позитивні, до всіх зразків, які класифікуються як позитивні. Шкала FDR розраховується як $FDR = FP / (FP + TP)$.

AU-ROC: Ця шкала також є ще одним важливим критерієм, який використовується для оцінки класифікаторів. Він розраховується на основі площі під кривою робочих характеристик приймача (ROC). Ця шкала розраховується як $AU-ROC = \frac{1}{2} \left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP} \right)$.

F1-Score: ця шкала поєднує дві шкали, включаючи точність і чутливість. Вона визначається як їх середньозважене значення. Якщо $F1-Score=1$, це найкраще значення. Навпаки, коли $F1-Score=0$, це вважається найгіршим значенням. Ця шкала розраховується як $F1-Оцінка = 2 \times \left(\frac{Точність \times Чутливість}{Точність + Чутливість} \right)$.

Крива робочих характеристик приймача (ROC): ця крива є методом організації та вибору класифікаторів на основі їх продуктивності. ROC – це двовимірний граф. Його вертикальна вісь представляє чутливість, а горизонтальна вісь вказує на специфічність. Нова шкала визначається на основі ROC, яка називається площею під ROC (AUC), яка використовується для порівняння продуктивності класифікаторів. Вона має значення від 0,5 до одиниці. Якщо AUC близький до 0,5, класифікатор має слабку продуктивність.

Методи ML, наприклад, можна використовувати в програмах для автоматизації таких завдань, як сегментація медичних зображень. У цьому випадку для оцінки моделей машинного навчання можна використовувати інші шкали, такі як коефіцієнт Дайса та індекс Жаккара [26].

Параметр релевантності моделі використовується для оцінки невідповідності між моделлю та даними. Це відноситься до надмірної та недообладнаної. Якщо доступних даних недостатньо, це спричиняє невідповідність між даними та моделлю. Корисним рішенням для вирішення цієї проблеми є перехресна перевірка. Однак точно не можна визначити, скільки переобладнання допустимо для моделі навчання [27].

Ефективність – швидкість передбачення та швидкість навчання в моделі навчання. Проблема ефективності виникає, коли система на основі машинного навчання виконує процеси навчання або прогнозування дуже повільно. Як результат, слід враховувати час виконання алгоритмів навчання.

Іноді моделі навчання використовуються для прийняття рішення про медичне лікування. У результаті треба розуміти логіку та причину рішень, прийнятих цими моделями, щоб довіряти їхнім рішенням, щоб остаточні моделі були соціально прийнятними. Однак важко визначити інтерпретативність з точки зору математики.

Методи ML у сфері охорони здоров'я поділяються на дві основні категорії на основі схем попередньої обробки даних: схеми очищення даних, схеми зменшення даних.

Деякі методи на основі машинного навчання, представлені в охороні здоров'я, використовують методи очищення даних, щоб усунути протиріччя, наприклад відсутні дані або дані з шумом, оскільки такі проблеми є поширеними в наборах даних про здоров'я. Ці проблеми мають кілька причин: Пристрої для збору даних не є точними в галузі охорони здоров'я. У результаті деякі дані можуть бути втрачені через апаратні обмеження цих пристроїв або деякі дані можуть бути помилково записані; Деякі зразки даних виготовляються вручну лікарями або лікувальним персоналом. Таким чином, вони можуть бути неправильно записані через людські помилки; Деякі пацієнти ненавмисно або навмисно не висловлюють належної інформації про свою хворобу. Це спричиняє помилки під час запису даних. Загалом існує

кілька методів очищення даних, включаючи керування відсутніми значеннями, керування шумними даними та нормалізацію даних [18, 20].

Існує два основні підходи до керування відсутніми значеннями в галузі охорони здоров'я: видалення даних із відсутніми значеннями, оцінка відсутніх значень

Методи фільтрації використовуються для видалення шуму в наборах даних про здоров'я. Це підвищує точність моделі навчання. Однак виявити зашумлені дані нелегко. Рішенням є перевірка бази даних професіоналами та лікарями для покращення її якості. Це забезпечує більш точне моделювання та зменшує його похибку. Однак ця робота дорога і трудомістка.

Зазвичай дані про здоров'я виражаються в різних масштабах, наприклад (вік, стать тощо). Не можна порівнювати ці зразки даних між собою.

Для вирішення цієї проблеми відповідним рішенням є використання методів нормалізації даних, таких як метод Min-Max, щоб помістити дані в діапазон $[0,1]$.

Часто дані про здоров'я мають великі розміри. Це послаблює продуктивність алгоритмів машинного навчання, оскільки знижує якість процесу навчання та точність моделі навчання. Зменшення розмірності означає, що дані про стан здоров'я представлені в стислому вигляді. В результаті цей процес призводить до втрати деякої інформації. Відповідна схема зменшення розмірності в галузі охорони здоров'я повинна підтримувати корисні функції. Методи зменшення даних поділяються на дві основні категорії: вибір ознак і вилучення ознак.

У процесі вибору функцій з бази даних здоров'я вибирається підмножина функцій, яка буде використовуватися в процесі навчання. Процес вибору ознак виконується автоматично або напівавтоматично [16, 17]. Рішення про видалення або збереження функції базується на бажаній програмі. Загалом методи вибору функцій поділяються на три групи: методи оболонки, вбудовані методи, методи фільтрації.

У методах оболонки розглядається модель на основі ML як чорний ящик. Потім доповнюється ця модель різними підмножинами функцій. Далі оцінюється її продуктивність для кожного підмножини, щоб визначити її ефективність. Нарешті, вибирається найкраща підмножина функцій. Є два відповідних підходи до оболонки, включаючи пряме виділення та зворотне виділення. У процесі прямого спочатку розглядається порожня підмножина. Потім вибирається функція бази даних здоров'я та вставляємо її в підмножину. Далі оцінюється продуктивність моделі на основі ML. Якщо це зменшує системну помилку порівняно з іншими функціями, її додають до останнього піднабору. Цей процес триває до тих пір, поки рівень помилок не зменшиться. Методи зворотного відбору подібні до підходів прямого відбору. Однак різниця є. У цих схемах спочатку розглядається підмножина, що включає всі функції. Потім вибирається функція цієї підмножини на кожному кроці та видаляється ця функція з підмножини. Цей процес триває до тих пір, поки рівень помилок моделі навчання не зменшиться [16, 17].

У вбудованих методах процес вибору ознак є компонентом моделі навчання [16, 17].

Методи фільтрації розглядаються як незалежна частина моделі навчання. У цих методах перевірка пріоритетів виконується для кожної функції бази даних, тому ці функції ранжуються на основі певного критерію. Потім користувач вибирає кращі функції [16, 17].

Методи вилучення ознак використовуються для стиснення даних про здоров'я, які мають великі розміри [18]. Це зберігає основні функції бази даних і усуває її шум і кореляції. Це прискорить процес навчання та дасть точніші результати.

РСА є багатовимірним і неконтрольованим методом [18, 17]. РСА має завдання аналізувати дані для отримання корисної інформації. Потім він відображає цю інформацію як набір нових ортогональних змінних. Їх називають головними компонентами.

Лінійний дискримінантний аналіз (LDA) контрольований метод навчання [18, 17].

Його мета – знайти лінійну комбінацію ознак, які можна розділити на два або більше класів. Цей метод намагається максимізувати поділ між класами та точно генерувати лінійні дискримінантні функції.

Розкладання сингулярного значення (SVD) – це метод навчання без контролю [18, 17]. Близький до PCA.

По суті, SVD є узагальненим варіантом PCA. Він вважається методом матричної факторизації та є ефективною схемою для зменшення розмірності даних.

SVD дає оптимальне наближене представлення вихідної матриці з використанням матриці низького рангу.

У схемі контрольованого навчання є набір вхідних і вихідних даних (бази даних з мітками) [28]. Метою цієї методики навчання є виявлення зв'язку між входами та виходами в процесі навчання [29, 30]. Цей алгоритм створює функцію, яка зіставляє дані з мітками. Потім він використовується для передбачення мітки непозначених даних. Контрольоване навчання використовується, коли є результати (мітки) для навчального набору.

Наївний Байєс (NB) – це імовірнісний класифікатор, який виражає зв'язок між змінними (ознаками) і цільовою змінною (класом) як умовну ймовірність [30, 31].

NB – це проста схема, заснована на теорії Байєса. У цьому методі передбачається, що точки даних в одному класі розподіляються на основі певного розподілу ймовірностей.

У NB існує сильна гіпотеза, яка називається незалежністю ознак. Однак ця гіпотеза непрактична в реальному світі, оскільки більшість реальних наборів даних сильно корельовані.

Звичайно, сьогодні NB використовувався для вирішення багатьох реальних проблем і показав хорошу продуктивність.

Дерево рішень (DT) є методом навчання під наглядом. DT будує навчальну модель, використовуючи набір правил IF-THEN, отриманих із навчального набору, щоб передбачити вихідний клас [32, 33]. Ієрархічне дерево створюється на основі ознак у наборі даних. У дереві рішень є три типи вузлів: кореневий вузол (найвищий вузол у дереві рішень), внутрішній вузол (він вказує на експеримент (або порівняння) для кожної функції), кінцевий вузол (мітка класу або кінцевий результат) .

Штучна нейронна мережа (ШНМ) включає вхідні змінні, вихідні змінні та ваги. Поведінка мережі залежить від співвідношення між вхідними та вихідними змінними [34, 35]. ШНМ складаються з трьох рівнів. Кожен із цих рівнів включає ряд процесорних одиниць, які називаються нейронами. Перший рівень – це вхідний рівень, який отримує необроблені дані. Другий рівень також відомий як прихований шар, який виконує навчальне завдання. Зауважте, що деякі ШНМ можуть мати кілька прихованих рівнів. Третій шар також відомий як вихідний рівень. Вихідний рівень залежить від процесу навчання в прихованому шарі, а також від вагових коефіцієнтів, пов'язаних із вхідними одиницями та прихованими одиницями. Ця робота проводиться методом проб і помилок. Існує багато підходів до навчання ШНМ і модифікації ваг, щоб отримати найменшу помилку. Найпоширенішим методом є алгоритм зворотного поширення.

Ансамблеві системи навчання або системи множинної класифікації мають багато застосувань у широкому діапазоні питань [36, 37]. У ансамбльній системі різні методи навчання поєднуються один з одним для покращення результату прогнозування. Це допомагає нам розробити точну та надійну модель класифікації. Метою ансамблевого навчання є створення класифікаторів з відносно постійним упередженням. Крім того, ансамблеве навчання поєднує результати класифікаторів за допомогою схеми усереднення або інших методів для зменшення дисперсії та підвищення точності. Ансамблеві системи проектуються різними способами. Однак вони мають три основні частини. Різноманітність, це означає, що кожен учасник ансамблю

повинен бути навчений за різними методами, щоб покращити загальну ефективність системи навчання ансамблю. Одним із рішень є вибір різних наборів даних для навчання кожного учасника ансамблю. Підготовка учасників ансамблю, ця частина дуже важлива в кожній системі ансамблевого навчання. Існують різні схеми навчання членів, наприклад, bagging і boosting. Об'єднання учасників ансамблю. Це стосується правила комбінування для отримання остаточного рішення.

Випадковий ліс (RF) – цей класифікатор швидкий, точний і завадостійкий. RF – це методика ансамблевого навчання, яка класифікує дані за допомогою дерев рішень [35]. У цій схемі велика кількість незалежних дерев створюється з використанням початкового навчального набору, наприклад, матриці $N \times F$, де N є кількістю вибірок, а F вказує на кількість ознак. Після створення випадкового лісу він використовується для прогнозування міток.

Глибоке навчання (DL) – це схема навчання під наглядом. Це підмножина ШНМ. Інша назва DL – глибока нейронна мережа (DNN) [34]. У DL є кілька прихованих шарів між вхідним шаром і вихідним шаром. Таким чином, вона може витягувати функції високого рівня з набору даних. DL може працювати з позначеними та не позначеними наборами даних. Крім того, його можна навчити для досягнення кількох цілей [36].

Машина опорних векторів (SVM) – це контрольована бінарна схема навчання. SVM використовує позначений навчальний набір, щоб дізнатися різницю між двома класами, відображаючи вхідні дані в нелінійний результат.

У напівкеруваному методі навчання в процесі навчання використовуються як позначені, так і немарковані набори даних. Тому ця техніка вимагає навчання алгоритму навчання під наглядом на позначеному навчальному наборі. Крім того, слід використовувати алгоритм неконтрольованого навчання для отримання зразків даних з новими мітками [38, 39]. Ці зразки даних додаються до позначеного навчального набору для алгоритму навчання під наглядом.

Модель навчання з підкріпленням (RL) дозволяє машинам або агентам вивчати свою ідеальну поведінку в конкретній ситуації на основі попереднього досвіду [12, 24]. Модель, заснована на навчанні з підкріпленням, постійно вчиться через взаємодію з навколишнім середовищем і збирає інформацію для виконання своєї діяльності [40]. З часом були представлені різні методи вирішення проблеми навчання з підкріпленням. Наприклад, обчислювальні методи, такі як динамічне програмування (DP) до глибокого навчання з підкріпленням (DRL).

Динамічне програмування (DP) включає набір методів для розрахунку оптимальної політики повної моделі середовища.

На відміну від схем динамічного програмування, методи на основі Монте-Карло (MC) є вільною моделлю. Це означає, що їм не потрібна повна модель середовища і вони навчаються на основі досвіду (тобто вони вчаться за допомогою взаємодії з середовищем). MC може вирішити проблему навчання з підкріпленням шляхом усереднення результатів вибірки. Методи Монте-Карло гарантують наявність відповідних зразків, оскільки вони часто використовуються для епізодичних завдань. Це означає, що досвід потрібно розділити на епізоди. Таким чином, MC є поступовою схемою епізод за епізодом [12, 24].

Q -Learning відомий як відповідний і популярний алгоритм навчання з підкріпленням. Q -Learning допомагає агенту вивчити його найкращі дії. У цьому методі існує таблиця під назвою Q -Table. У цій таблиці зберігаються пари «дія-стан» і відповідні значення. Насправді пари дія-стан відомі як вхідні дані в цій таблиці, а Q -значення є її виходом. У Q -Learning метою є максимізація Q -Value [12, 24].

Стан-дія-нагорода-стан-дія (SARSA) – це метод навчання з підкріпленням. Його мета – вивчити політику MDP. SARSA і Q -Learning дуже схожі один на одного. Однак між SARSA та Q -Learning є одна головна відмінність. SARSA – це метод, який відповідає правилам. Навпаки, Q -Learning є методом поза. Виконання правил означає, що SARSA

дотримується існуючих правил для вибору дій. Потім він оновлює Q -Value у Q -Table. Тоді як схема поза правилами, як Q -Learning, не відповідає існуючим правилам. Він обирає дії, використовуючи жадібний спосіб, щоб максимізувати Q -Value в Q -Table [12, 24].

Глибоке навчання з підкріпленням (DRL) – це поєднання глибокого навчання та навчання з підкріпленням. Цю модель можна використовувати для вирішення багатьох складних завдань. Це допомагає агентам стати розумнішими. Це покращує їх здатність оптимізувати політику. Навчання з підкріпленням – це техніка машинного навчання, яка може працювати без будь-якої бази даних. Тому в DRL агенти можуть спочатку створити набір даних через взаємодію з середовищем. Потім ця база даних використовується для навчання глибоких мереж у DRL [12, 24].

Більшість моделей на основі машинного навчання, розроблених у сфері охорони здоров'я, використовують інструменти моделювання для оцінки ефективності, оскільки вони більш доступні, ніж практична реалізація. Вони також мають більшу гнучкість і знижують вартість. Щоб оцінити моделі на основі ML, необхідно змоделювати цю модель навчання за допомогою відповідних інструментів моделювання, таких як MATLAB, WEKA та R, щоб визначити її ефективність. Ці моделі навчання оцінюються на основі різних шкал оцінювання. Загалом критерії оцінювання поділяються на дві основні категорії.

Шкали дискримінації аналізують здатність моделі на основі машинного навчання ранжувати або розрізнити два класи. Найважливішими шкалами розрізнення є ROC, AU-ROC, F1-Score, чутливість і специфічність.

Калібрувальні шкали визначають, скільки прогнозованих результатів відповідає фактичним результатам. У реальному світі ці шкали дуже важливі, оскільки вони аналізують очікувані прибутки чи збитки. Наприклад, якщо ризик смерті, викликаний хірургічним втручанням, перевищує ризик смерті без операції, хірург може не виконувати цю операцію та відмовитися від неї.

Дуже важливо оцінювати моделі ML в охороні здоров'я, використовуючи їх практичну реалізацію, оскільки це дозволяє нам оцінювати та аналізувати моделі навчання в реальному середовищі. Однак це дуже дорого, оскільки зазвичай мається справа зі складністю апаратного забезпечення для розробки моделей навчання. Повторювати сценарії та проводити різні експерименти також дуже важко. У практичній реалізації потрібно оцінювати модель навчання в режимі реального часу та постійно оновлювати цю модель і повторно перевіряти її. Деякі важливі шкали під час практичного впровадження моделей навчання в охороні здоров'я включають їх узагальнення для нових даних, відгуки користувачів, довіру медичної спільноти до розробленої моделі, порівняння ефективності моделі з експертом у відповідній галузі та порівняння продуктивності моделі з іншими існуючими моделями.

Методи ML в охороні здоров'я поділяються на дві основні категорії на основі застосування: діагностика та лікування.

Діагностика – це дуже важливий етап у галузі медицини. Машинне навчання можна використовувати в цій сфері, щоб допомогти лікарям і виявити захворювання на ранніх стадіях, а також скоротити час виявлення. Наприклад, машинне навчання можна використовувати для покращення медичних зображень, аналізу лабораторних результатів, сегментації та ідентифікації елементів на зображеннях, виявлення захворювання, визначення ступеня захворювання, аналізу сигналів таких пристроїв, як електрокардіографія (ЕКГ) для виявлення серцевої недостатності або електроенцефалографії (ЕЕГ) для оцінки активності мозку.

Деякі методи на основі ML можуть допомогти в лікуванні захворювань. Наприклад, машинне навчання можна використовувати для діагностики відповідних доз, персоналізованої терапії, моніторингу процедури лікування та прогнозування прогресування захворювання. Ці методи зменшують витрати на лікування, зменшують витрати, пов'язані з виробництвом ліків, покращують процедуру лікування, економлять час на пошук відповідних ліків

і вирішують проблеми, викликані відсутністю лікарів-спеціалістів. Машинне навчання також може охоплювати хірургічні операції, щоб полегшити важкі операції високої складності, які навряд чи виконуються людьми.

2.2 Інтегрована модель на основі LOG і RF

Цинь [41] запропонував метод на основі ML для своєчасної діагностики хронічної хвороби нирок (ХНН). Спочатку автори використовували техніку імпутації KNN для оцінки відсутніх значень у базі даних. Вони також використовували оптимальну регресію підмножини та RF для зменшення розмірності та вибору найбільш підходящих характеристик у наборі даних. Потім модель навчання була розроблена з використанням різних класифікаторів. Хронічна хвороба нирок (ХНН) є серйозним захворюванням, яке може загрожувати загальному здоров'ю. Методи на основі ML можуть допомогти нам вчасно та точно діагностувати це захворювання. У реальному світі більшість медичних наборів даних мають багато відсутніх значень.

У [41] автори вважають, що існуючі методи діагностики ХНН мають низьку точність, або вони використовували обмежену та слабку техніку для оцінки відсутніх значень. Тому автори [41] запропонували модель діагностики ХНН на основі ML. Метою цього методу навчання є підвищення точності та покращення його застосування.

У [41] використовується база даних СКД, доступна в репозиторії машинного навчання Університету Каліфорнії Ірвайн (UCI). У цій базі даних є 400 точок даних. Ці точки даних мають 24 ознаки, включаючи 11 числових і 13 номінальних. Крім того, є дві кінцеві мітки, включаючи ХНН (у цьому наборі даних є 250 пацієнтів із ХНН) і НОТХН (у цьому наборі даних є 150 точок даних, які відомі як НОТХН). Зауважте, що цей набір даних відносно малий, і ця проблема обмежує продуктивність цього методу з точки зору можливості узагальнення.

У [41] для оцінки відсутніх значень у базі даних застосовано метод імпутації KNN. Цей метод вибирає k точок даних без пропущених значень. Ці точки даних мають бути найближчими до відсутніх значень. Шкала подібності – евклідова відстань. Тут є два випадки. Один випадок полягає в тому, що відсутнє значення є числовою змінною. У цьому випадку відсутнє значення оцінюється на основі медіани k точок даних. Другий випадок полягає в тому, що відсутнє значення є номінальною змінною. У цьому випадку вона виходить на основі більшості голосів. Крім того, ця навчальна модель використовує метод вибору ознак на основі регресії оптимальної підмножини та RF для вибору найвигідніших функцій.

У [41] схема навчання під наглядом використовується для прогнозування ХХН. У процесі класифікації досліджуються різні класифікатори. Мета полягає в тому, щоб класифікатори з найкращими характеристиками були обрані для розробки кінцевої моделі. Ці моделі навчання включають: логістичну регресію (LOG), випадковий ліс (RF), опорну векторну машину (SVM), K найближчого сусіда (KNN), наївний Байєс (NB), нейронну мережу прямого зв'язку (FNN). Потім вони оцінюють продуктивність різних моделей на основі кількох параметрів, таких як точність, кількість неправильних оцінок, час роботи тощо.

Нарешті, RF і LOG вибираються для створення остаточної моделі інтеграції.

Цей метод використовує оцінку на основі моделювання.

Для цього автори використовували програмне забезпечення R 3.5.2 для моделювання моделі прогнозування ХХН. Для оцінки моделі навчання використовується метод 4-Fold-Cross-Validation.

Нарешті, ця модель навчання була оцінена за різними критеріями, такими як точність, чутливість, специфічність і оцінка F1.

2.3 FCMIM-SVM

Лі [42] представив систему на основі ML для виявлення серцевої недостатності. Він запропонував метод вибору функцій під назвою FCMIM. Крім того, автори досліджували різні методи навчання, такі як штучні нейронні мережі (ANN), опорні векторні машини (SVM), дерево рішень (DT), наївний байес (NB), K найближчий сусід (KNN) і логістична регресія (LR). Для розробки кінцевої моделі навчання. Нарешті вони створили остаточну систему навчання під назвою FCMIM-SVM.

Серцева хвороба, як відомо, серйозна хвороба. Це може загрожувати життю багатьох людей у світі. Традиційні методи виявлення цього захворювання трудомісткі, дорогі та малоефективні. Таким чином, методи на основі ML можуть бути дуже ефективними, оскільки вони можуть виявляти серцеві захворювання за допомогою швидкої, точної та недорогої схеми. Крім того, продуктивність схеми на основі ML можна покращити, якщо використовувати збалансовану базу даних і ефективну схему вибору функцій. Що стосується згаданих питань, автори [42] надали метод на основі ML і підхід до вибору ознак для швидкого та точного виявлення захворювань серця.

FCMIM-SVM використовує набір даних про захворювання серця, пов'язаний із Клівлендом. Цей набір даних включає 303 точки даних. Кожна точка даних також має 75 характеристик. Є шість точок даних із відсутніми значеннями. У процесі попередньої обробки ці точки даних було видалено. Крім того, є два класи для кінцевої мітки: HD або Not-HD.

FCMIM-SVM застосовує різні методи попередньої обробки даних. Наприклад, він видаляє точки даних із відсутніми значеннями з набору даних. Він також виконує деякі операції нормалізації, такі як стандартний скаляр (SS) і мінімально-максимальний скаляр над набором даних. Крім того, FCMIM-SVM розробляє метод вибору функцій під назвою FCMIM для зменшення розмірності. Крім того, різні.

У [43] автори використали оцінку на основі моделювання. Вони використовували симулятор WEKA 3.9.1 для реалізації NE.

Для оцінки цих методів була використана техніка 3, 5, 10-кратної перехресної перевірки. NE оцінюються на основі різних критеріїв, включаючи точність, точність, відкликання, оцінку F1, ROC і час обробки.

2.4 HMANN

Ma [44] запропонував вдосконалену нейронну мережу під назвою HMANN. Ця схема використовується для виявлення, сегментації та ідентифікації хронічної ниркової недостатності.

HMANN реалізовано на платформі Internet of Medical Things (IoMT). Цей метод поєднує машину опорних векторів (SVM), багат шаровий перцептрон (MLP) і алгоритм зворотного поширення (BP).

При поганій роботі нирок ця проблема може загрожувати життю людини. Тому дуже важливо вчасно виявити камені в нирках.

Часто цифрові зображення мають низьку контрастність. Вони також дуже шумні. Тому дуже важко використовувати ці зображення для виявлення аномалій нирок. Штучні нейронні мережі є одним з найпоширеніших інструментів для вирішення цієї проблеми. Тому що вони відмовостійкі.

Їх також можна легко узагальнити.

Крім того, вони мають відповідну здатність до навчання. Тому в [44] була розроблена система на основі нейронної мережі.

Автори використовують зображення в наборі даних про хронічну хворобу нирок UCI для навчання та тестування HMANN.

У цьому методі немає пояснень щодо цієї бази даних. Автори не називають кількість зображень у наборі даних та їх тип.

Метод попередньої обробки даних. Як згадувалося раніше, цифрові зображення часто мають шум і низьку контрастність. Їх оцінка складна.

У HMANN автори зменшили шум за допомогою порогових вейвлет-коефіцієнтів. Загалом на цих зображеннях виконується процес попередньої обробки, щоб подолати низький контраст і шум.

Процес попередньої обробки даних включає три кроки: відновлення зображень за допомогою методу набору рівнів, посилення або згладжування за допомогою фільтра Габора, покращення контрастності за допомогою процесу вирівнювання гістограми.

Крім того, лікар-спеціаліст вручну виконує процес сегментації нормальних і аномальних цифрових зображень.

Потім HMANN використовує процес виділення ознак, який називається матрицею спільного виникнення сірого рівня (GLCM), у цих сегментованих областях, щоб виділити ознаки, пов'язані з цією хворобою.

Ці функції включають адаптивні функції, функції Хараліка та гістограми. Потім виконується процес вибору функції для вибору дев'яти функцій.

У [44] остаточна модель навчання побудована на основі трьох основних компонентів, включаючи SVM, MLP і BP.

Остаточна модель навчання називається HMANN. Метою HMANN є класифікація цифрових зображень, змінених на попередньому кроці, ідентифікація каменів у нирках і точне визначення їх розташування.

Оцінка. HMANN використовує оцінку на основі моделювання.

Цей метод моделюється та оцінюється за допомогою різноманітних експериментів, щоб визначити його ефективність.

Однак автори не пояснюють інструмент моделювання, набір для навчання, набір для тестування та інші параметри моделювання. HMANN оцінюється на основі різних критеріїв, таких як швидкість прогнозування, AUC, точність, час обчислення та ROC.

2.5 SRL-RNN

Ванг [45] запропонував модель на основі МН під назвою SRL-RNN. Ця схема використовує навчання з підкріпленням і рекурентну нейронну мережу (RNN). Метою SRL-RNN є вирішення проблеми динамічного режиму лікування (DTR). Основна ідея цього методу полягає в поєднанні двох сигналів, включаючи показник і оцінку одночасно.

Багато дослідників переглядали системи рекомендацій щодо ліків, щоб допомогти лікарям приймати кращі рішення.

Ці системи можуть бути розроблені з використанням керованих алгоритмів або алгоритмів навчання з підкріпленням.

Контрольовані системи використовують схожість між пацієнтами для надання рекомендацій.

Однак ці методи не можуть безпосередньо визначити зв'язок між хворобою та ліками. Ці методи залежать від істинності землі.

Однак немає відповіді на запитання: як створюється ця основоположна правда? У цьому випадку вони працюють за сигналом індикатора.

У той час як системи навчання з підкріпленням не мають цієї проблеми. Однак вони можуть надавати рекомендації щодо лікування, які сильно відрізняються від рецепту, рекомендованого лікарем. Це тому, що керівник їх не контролює.

Ця проблема може збільшити ризик лікування. Насправді вони працюють на основі сигналу оцінки. Тому автори [45] поєднують контрольоване навчання та навчання з підкріпленням, щоб створити нову модель під назвою SRL-RNN.

Цей метод дозволяє уникнути несанкціонованих ризиків і вивести оптимальне та динамічне лікування.

Використовується велика та доступна база даних під назвою MIMIC-3 v1.4 для оцінки SRL-RNN. Ця база містить інформацію про 43 000 пацієнтів у відділеннях інтенсивної терапії (ВІТ).

Ця інформація збиралася з 2001 по 2012 рік. Вона містить інформацію про 6695 конкретних захворювань і 4127 ліків.

У [45], коли точка даних має багато відсутніх значень, більше 10 ознак, тоді ця точка даних повинна бути видалена з бази даних. З іншого боку, коли точка даних має невелику кількість відсутніх значень, тоді ці відсутні значення оцінюються за допомогою методу KNN.

У [45] автори представили глибоку архітектуру під назвою SRL-RNN для управління DTR, включаючи кілька захворювань і різні рецепти. Мета полягає в тому, щоб вивчити директивну політику шляхом поєднання сигналу індексу та сигналу оцінки.

SRL-RNN включає три основні мережі: мережа акторів для виробництва ліків у часових змінах на основі динамічного стану пацієнтів. У цьому процесі рішення лікаря відіграють роль індикаторного сигналу. Це означає наявність супервізора для забезпечення безпечних дій і прискорення процесу навчання.

Критична мережа для оцінки дії, пов'язаної з мережею акторів, щоб винагородити або покарати рекомендоване лікування.

Мережа LSTM для розробки SRL-RNN для управління частково спостережуваним марковським процесом прийняття рішень (POMDP). Він узагальнює спостереження, щоб отримати більш повне спостереження.

LSTM є однією з найвідоміших рекурентних нейронних мереж (RNN). Він відомий як глибока нейронна мережа.

SRL-RNN використовує обидва методи оцінки, тобто на основі моделювання та на основі практичної реалізації.

При практичному впровадженні рецепти, виготовлені цим методом, оцінюються для двох пацієнтів у відділенні інтенсивної терапії.

Автори не згадують програмне забезпечення, яке використовується для моделювання цього методу. Набір даних поділено на три групи, включаючи набір для навчання (80% набору даних), набір перевірки (10% набору даних) і набір для тестування (10% набору даних). У [45] рівень смертності

розглядається як оцінна шкала для оцінки ефекту цього методу для зниження смертності.

Коефіцієнт Жаккара використовувався для вимірювання сумісності між рецептами, рекомендованими SRL-RNN, і рецептами, виписаними лікарем.

2.6 Схема обробки із замкнутим циклом охорони здоров'я

Дай та ін. [46] змоделювали людське тіло за допомогою глибоких нейронних мереж (DNN) і використали глибоке навчання з підкріпленням (DRL), щоб знайти відповідні схеми лікування для змодельованого тіла.

У цьому методі змодельоване тіло грає роль пацієнта, а DRL – лікаря.

В охороні здоров'я необхідно постійно спостерігати за організмом людини, щоб вчасно проводити відповідне лікування.

Проте проводити несанкціоновані тести на людському тілі невірно. Тому необхідно спроектувати віртуальне тіло людини.

Однак людський організм – це дуже складна система. Сьогодні сучасну науку супроводжує великий прогрес.

Однак він не може повністю імітувати тіло людини. Рішення полягає в тому, щоб розглядати тіло як чорний ящик для інтерпретації вихідних даних у відповідь на вхідні дані.

Це означає, що він заснований на методі, керованому даними.

DNN є корисним інструментом для моделювання людського тіла, оскільки він має можливість глобальної апроксимації. Тому в [46] DNN використовується для моделювання людського тіла.

У [46] автори використовують базу даних, яка містить 990 зображень язика. Ці зображення містять 9 різних структур для навчання глибокої нейронної мережі (DNN).

У цій моделі немає методу попередньої обробки.

Модель навчання, представлена в [46], включає два основні компоненти: змодельоване тіло та лікувальну частину. Змодельоване тіло складається з двох основних частин, включаючи мережу регулювання та мережу декодування.

Завдання регулюючої мережі – показати вплив лікування на стан здоров'я. Крім того, мережа декодування має завдання перетворити простір з низькими розмірами (тобто стан здоров'я) у простір з високими розмірами.

У [46] LSTM використовувався як метод глибокого навчання для моделювання людського тіла.

У [46] концептуальний автоматичний кодер глибинного вирівнювання (CADAЕ) використовувався як мережа декодування.

Другий компонент, тобто частина лікування, також відповідає за отримання спостережень і надання терапевтичних рекомендацій.

Цей компонент динамічно взаємодіє зі змодельованим тілом. Він складається з двох основних частин: діагностики захворювання та відповідних терапевтичних рекомендацій.

У [46] автор використав схему глибокого навчання з підкріпленням (DRL), щоб об'єднати ці дві частини.

У зв'язку з цим вони використовували глибоку Q -мережу (DQN) для дискретного простору та глибокий детермінований градієнт політики (DDPG) для безперервного простору.

Цей метод використовує оцінку на основі моделювання. Тому ця схема моделюється за допомогою TensorFlow, встановленого на Python. Змодельоване тіло тренується за допомогою CADAЕ.

Цей метод оцінюється з точки зору рівня конвергенції та рівня помилкової діагностики.

Цей метод представив експериментальні результати у формі графіка. У зв'язку з цим не надаються числові результати для цієї схеми.

2.7 GAN + RAE + DQN

Тсенг та ін. [47] представили глибоку схему навчання з підкріпленням для прийняття рішень щодо лікування.

Цей метод включає три компоненти: GAN для створення штучних даних на основі невеликого набору даних, перехідна DNN для побудови віртуального радіотерапевтичного середовища, DQN для визначення оптимальної дози опромінення для процесу променевої терапії.

Зазвичай лікарі вважають, що хірургічне втручання не підходить для лікування пацієнтів з недрібноклітинним раком легенів (НДРЛ), і їх краще лікувати за допомогою променевої терапії.

Однак ця технологія прогресує з кожним днем.

Однак результати його лікування незадовільні. Доцільним варіантом є збільшення дози опромінення при променевої терапії для посилення процесу лікування. Хоча це може збільшити запалення через радіацію та знизити якість життя пацієнтів.

Це дослідження намагається відповісти на таке запитання: «Чи можуть алгоритми машинного навчання визначити оптимальну дозу опромінення на основі особливостей пацієнтів для локального контролю пухлин і мінімізації запалення?».

В останні роки глибоке навчання з підкріпленням успішно використовується в різних областях.

Це пояснюється тим, що ця методика навчання може витягувати функції високого рівня безпосередньо з необроблених даних. Тому в роботі [47] DQN використовується для визначення дози опромінення при променевої терапії.

У цьому дослідженні використовується база даних, що включає 114 пацієнтів з НДКРЛ. Зверніть увагу, що кожна вибірка даних складається з 297 ознак.

У [47] автори використовують схему вибору ознак для вибору дев'яти важливих характеристик для моделювання середовища променевої терапії.

З цією метою використовується теорія графів байєсівської мережі, щоб ієрархічно визначити зв'язки між ознаками та бажаним результатом.

Ця схема намагається знайти мінімальні можливості для локального контролю пухлини та зменшення запалення через опромінення.

У [47] автори змодельовали середовище променевої терапії, щоб спроектувати середовище штучної променевої терапії. Для виконання цієї роботи поставлено завдання алгоритму переходу DNN.

Для цієї роботи вони використовували GAN разом з алгоритмом переходу DNN. Це тому, що доступна база даних дуже мала.

У результаті GAN, яка є глибокою нейронною мережею, може створювати штучні дані, дуже схожі на реальні.

Потім алгоритм DNN переходу навчається на основі реальних і штучних даних для моделювання середовища променевої терапії.

Далі DQN взаємодіє з цим змодельованим середовищем, щоб імітувати рішення лікаря та визначити дозу опромінення для кожного пацієнта.

Цей метод використовує оцінку на основі моделювання. Він застосовує програмне забезпечення MATLAB для процесу вибору функцій. У цьому випадку AUC розглядається як шкала оцінки.

Процес оцінювання використовує метод 10-кратної перехресної перевірки. Потім остаточна модель навчання реалізується в TensorFlow.

Як згадувалося раніше, у базі даних є 114 зразків даних. Потім GAN використовує цю базу даних для створення штучних даних.

Після виконання цього процесу створюється 4000 штучних зразків даних. У результаті кількість вибірок даних (реальних даних і штучних даних) дорівнює 4114.

Потім алгоритм DNN навчається відповідно до цієї нової бази даних. У цьому випадку критерієм оцінки є середня точність.

Потім алгоритм DQN виконується на 34 пацієнтах у протоколі UMCC.

2.8 HQLA

Хашемі [48] запропонував алгоритм навчання на основі підкріплення під назвою HQLA. Цей алгоритм використовує базу даних Квебеку для прогнозування поширеності коронавірусу. У цьому алгоритмі використовуються дві методики, включаючи навчання з підкріпленням і еволюційні алгоритми.

Моделювання та прогнозування епідемічного процесу COVID-19 може допомогти фахівцям у сфері охорони здоров'я завершити його поширеність. Однак передбачити поширеність COVID-19 дуже складно через його нечітку та складну природу.

Метаевристичні алгоритми дуже гнучкі та ефективні. Вони можуть вирішити багато проблем у сфері охорони здоров'я, оскільки вони зменшують обчислювальні витрати та складність часу. Вони також можуть ефективно досліджувати оптимальні відповіді.

Крім того, алгоритми навчання з підкріпленням можуть вирішити багато проблем у реальному світі, особливо в охороні здоров'я.

Квебек – одна з провінцій Канади. Набір даних включає зразки даних, пов'язані з COVID-19, і рівень смертності, зареєстрований з 25 червня по 19 липня 2020 року. Ця база даних містить 63713 зразків даних, пов'язаних із пацієнтами з COVID-19, і 5770 зразків даних, пов'язаних із померлими внаслідок COVID-19. .

Цей метод (HQLA) поєднує навчання з підкріпленням і еволюційні алгоритми. Ця схема дозволяє вирішити складні задачі оптимізації в короткостроковий період часу.

HQLA використовує різні еволюційні алгоритми, такі як GWO [49], SCA [50], MFO [51], PSO [52], WCA [53], SFS [54] для оновлення положення частинок у просторі відповіді. Q-Learning використовується для вибору найкращого оператора (еволюційного алгоритму) у процесі оптимізації для

отримання найкращої ефективності. *Q-Learning* починається з кількох випадкових операцій.

Потім він оцінює ефективність для кожного оператора на кожному кроці.

Це допомагає *Q-Learning* вивчати найкращі операції для отримання найкращої відповіді. Якщо оператор покращує якість кінцевої відповіді, *Q-Learning* винагороджує цього оператора. В іншому випадку він штрафує поточного оператора.

HQLA використовує оцінювання на основі моделювання. Зверніть увагу, що автори не згадують про програмне забезпечення, яке використовується для реалізації цього методу.

У процесі оцінки середня квадратична помилка розглядається як цільова функція. Його оптимальна кількість дорівнює $6,26 \times 10^{-6}$.

2.9 tVAE

Бакум та ін. [55] представив перехідні варіаційні автокодери (tVAE). Він намагається вивчити процедуру прогресування захворювання, щоб зіставити стан пацієнта з його наступним станом у наступний момент часу.

Навчання з підкріпленням (RL) є корисним інструментом для розробки індивідуального режиму лікування. З етичних причин агенти RL не можуть безпосередньо взаємодіяти з реальними пацієнтами.

Двома рішеннями цієї проблеми є: навчання моделі з використанням існуючого набору даних (Off-policy RL), вивчення моделі віртуального середовища з використанням доступного набору даних (On-policy RL).

Ця схема базується на техніці on-policy. tVAE прагне точно вивчити модель хвороби.

Використовувалося база даних MIMIC. У ній зберігається інформація про 2067 пацієнтів у реанімації. У цій базі даних такі параметри пацієнтів, як доза гепарину, вимірювалися щогодини.

База даних MIMIC містить відсутні значення. Для визначення відсутніх значень, пов'язаних із дозуванням гепарину, використовується метод інтерполяції вибірки та зберігання. Для оцінки відсутніх значень використовується штучна нейронна мережа.

Метод tVAE використовує стандартну структуру VAE для моделювання переходів між послідовними станами пацієнта.

tVAE навчає штучну нейронну мережу так, щоб вона отримувала безперервні вхідні дані та створювала вихідні дані.

Цей метод може розглядати безперервний простір захворювання та створювати випадковість у моделі.

tVAE підходить для медичних часових рядів. Після розробки віртуального середовища пацієнта для визначення найкращої дози гепарину використовується алгоритм навчання підкріплення.

У [55] tVAE використовує оцінку на основі моделювання. Цей метод моделюється в TensorFlow.

У процесі оцінки набір даних ділиться на дві частини: набір для навчання (85% вибірок даних) і набір для тестування (15% вибірок даних).

Крім того, критерієм оцінки є середня абсолютна похибка.

3 КОМП'ЮТЕРНА МОДЕЛЬ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ

3.1 Обґрунтування вибору середовища програмної реалізації

У рамках кваліфікаційної роботи був розроблений алгоритм для фільтрації зображень за допомогою високочастотних фільтрів.

Для створення мною було обрано мову Python та середовище Jupiter Notebook, яке дозволяє використовувати цю мову.

Побачити приклад інтерфейсу Jupiter Notebook можна на рисунку 3.1.



```

In [ ]: import pandas as pd

In [ ]: from fugue_notebook import setup
        setup()

In [ ]: df = pd.DataFrame({'a':[1,2,3,4], 'b':[1,2,3,4]})
        df.to_csv('df.csv', index=False)

In [ ]: %%fsql
        -- This SQL cell sees the dataframe defined in the previous cell
        SELECT *
        FROM df
        WHERE a > 2
        PRINT

In [ ]: %%fsql
        df2 = LOAD "/Users/kevinkho/Work/fugue/df.csv" (header=TRUE, infer_schema=TRUE)
        SELECT *
        FROM df2
        WHERE b < 2
        PRINT
        SAVE OVERWRITE "/Users/kevinkho/Work/fugue/df.csv"

In [ ]:

```

Рисунок 3.1 –Інтерфейс Jupyter Notebook

Додаток Jupyter Notebook – це серверно-клієнтська програма, яка дозволяє редагувати та запускати документи блокнота через веб-браузер.

Програму Jupyter Notebook можна запустити на локальному робочому столі, не потребуючи доступу до Інтернету.

Її також можна встановити на віддаленому сервері та отримати доступ через Інтернет.

Окрім відображення/редагування/запуску документів блокнота, у додатку Jupyter Notebook є «Інформаційна панель» (Notebook Dashboard), «панель керування».

Панель керування показує локальні файли та дозволяє відкривати документи блокнота або вимикати їх ядра.

3.2 Програмна реалізація

Спочатку, призводиться імпорт `pandas` і пакету під назвою `numpy` для числових маніпуляцій (рис. 3.2). Будемо взаємодіяти з рентгенівськими знімками грудної клітки, отриманими з загальнодоступного набору даних `ChestX-ray8`. Потім прочитаємо файл `csv` у фрейм даних (рис. 3.3) і роздрукуємо кілька перших рядків даних, щоб передивитися кількість пацієнтів (рис. 3.4).

```
In [1]: # Import necessary packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import os
import seaborn as sns
sns.set()
```

Рисунок 3.2 – Імпорт `pandas` і пакету `numpy`

```
In [2]: # Read csv file containing training data
train_df = pd.read_csv("nih/train-small.csv")
# Print first 5 rows
print(f'There are {train_df.shape[0]} rows and {train_df.shape[1]} columns in this data frame')
train_df.head()
```

Рисунок 3.3 – Читання `csv` у фрейм даних

Out[2]:	Image	Atelectasis	Cardiomegaly	Consolidation	Edema	Effusion	Emphysema	Fibrosis	Hernia	Infiltration	Mass	Nodule	PatientId	Pleural_Thickening	Pneumon
0	00008270_015.png	0	0	0	0	0	0	0	0	0	0	0	8270	0	
1	00029855_001.png	1	0	0	0	1	0	0	0	1	0	0	29855	0	
2	00001297_000.png	0	0	0	0	0	0	0	0	0	0	0	1297	1	
3	00012359_002.png	0	0	0	0	0	0	0	0	0	0	0	12359	0	
4	00017951_001.png	0	0	0	0	0	0	0	0	1	0	0	17951	0	

Рисунок 3.4 – Роздрук кількох перших рядків даних

Файл містить назви рентгенівських знімків грудної клітки (стовпець «Зображення»), а стовпці, заповнені одиницями та нулями, визначають, які діагнози було поставлено на основі кожного рентгенівського зображення.

Дослідимо типи даних у кожному стовпці та чи існують у даних нульові значення (рис. 3.5).

```
In [3]: # Look at the data type of each column and whether null values are present
train_df.info()

RangeIndex: 1000 entries, 0 to 999
Data columns (total 16 columns):
Image                1000 non-null object
Atelectasis          1000 non-null int64
Cardiomegaly         1000 non-null int64
Consolidation        1000 non-null int64
Edema                1000 non-null int64
Effusion             1000 non-null int64
Emphysema           1000 non-null int64
Fibrosis             1000 non-null int64
Hernia               1000 non-null int64
Infiltration         1000 non-null int64
Mass                 1000 non-null int64
Nodule               1000 non-null int64
PatientId            1000 non-null int64
Pleural_Thickening  1000 non-null int64
Pneumonia            1000 non-null int64
Pneumothorax        1000 non-null int64
dtypes: int64(15), object(1)
memory usage: 125.1+ KB
```

Рисунок 3.5 – Перевірка на нульові значення

«PatientId» має ідентифікаційний номер для кожного пацієнта. Потрібно перевірити чи відповідають данні до кожного пацієнта окремо та чи представляє кожне зображення різну людину (рис. 3.6).

```
In [4]: print(f"The total patient ids are {train_df['PatientId'].count()}, from those the unique ids are {train_df['PatientId'].value_counts().shape[0]} ")

The total patient ids are 1000, from those the unique ids are 928
```

Рисунок 3.6 – Перевірка на відповідність

Як можемо побачити, кількість унікальних пацієнтів у наборі даних менша за загальну кількість, тому має бути певне збігання. Для пацієнтів із кількома записами слід переконатися, що вони не з'являються як у навчальних, так і в тестових наборах, щоб уникнути витоків даних.

Створимо список імен кожного стану або захворювання пацієнта та видалимо непотрібні елементи та надрукуємо кількість позитивних міток для кожної умови для перевірки (рис. 3.7).

```
In [5]: columns = train_df.keys()
columns = list(columns)
print(columns)

['Image', 'Atelectasis', 'Cardiomegaly', 'Consolidation', 'Edema', 'Effusion', 'Emphysema', 'Fibrosis', 'Hernia', 'Infiltration', 'Mass', 'Nodule', 'PatientId', 'Pleural_Thickening', 'Pneumonia', 'Pneumothorax']
```

```
In [6]: # Remove unnecessary elements
columns.remove('Image')
columns.remove('PatientId')
# Get the total classes
print(f"There are {len(columns)} columns of labels for these conditions: {columns}")

There are 14 columns of labels for these conditions: ['Atelectasis', 'Cardiomegaly', 'Consolidation', 'Edema', 'Effusion', 'Emphysema', 'Fibrosis', 'Hernia', 'Infiltration', 'Mass', 'Nodule', 'Pleural_Thickening', 'Pneumonia', 'Pneumothorax']
```

```
In [7]: # Print out the number of positive labels for each class
for column in columns:
    print(f"The class {column} has {train_df[column].sum()} samples")

The class Atelectasis has 106 samples
The class Cardiomegaly has 20 samples
The class Consolidation has 33 samples
The class Edema has 16 samples
The class Effusion has 128 samples
The class Emphysema has 13 samples
The class Fibrosis has 14 samples
The class Hernia has 2 samples
The class Infiltration has 175 samples
The class Mass has 45 samples
The class Nodule has 54 samples
The class Pleural_Thickening has 21 samples
The class Pneumonia has 10 samples
The class Pneumothorax has 38 samples
```

Рисунок 3.7 – Перевірка на нульові значення

Використовуючи імена зображень, перелічені у файлі csv, отримаємо зображення, пов'язане з кожним рядком даних у кадрі даних та візуалізуємо випадковий вибір зображень із набору даних задля перегляду (рис. 3.8).

```
In [8]: # Extract numpy values from Image column in data frame
images = train_df['Image'].values

# Extract 9 random images from it
random_images = [np.random.choice(images) for i in range(9)]

# Location of the image dir
img_dir = 'nih/images-small/'

print('Display Random Images')

# Adjust the size of your images
plt.figure(figsize=(20,10))

# Iterate and plot random images
for i in range(9):
    plt.subplot(3, 3, i + 1)
    img = plt.imread(os.path.join(img_dir, random_images[i]))
    plt.imshow(img, cmap='gray')
    plt.axis('off')

# Adjust subplot parameters to give specified padding
plt.tight_layout()

Display Random Images
```

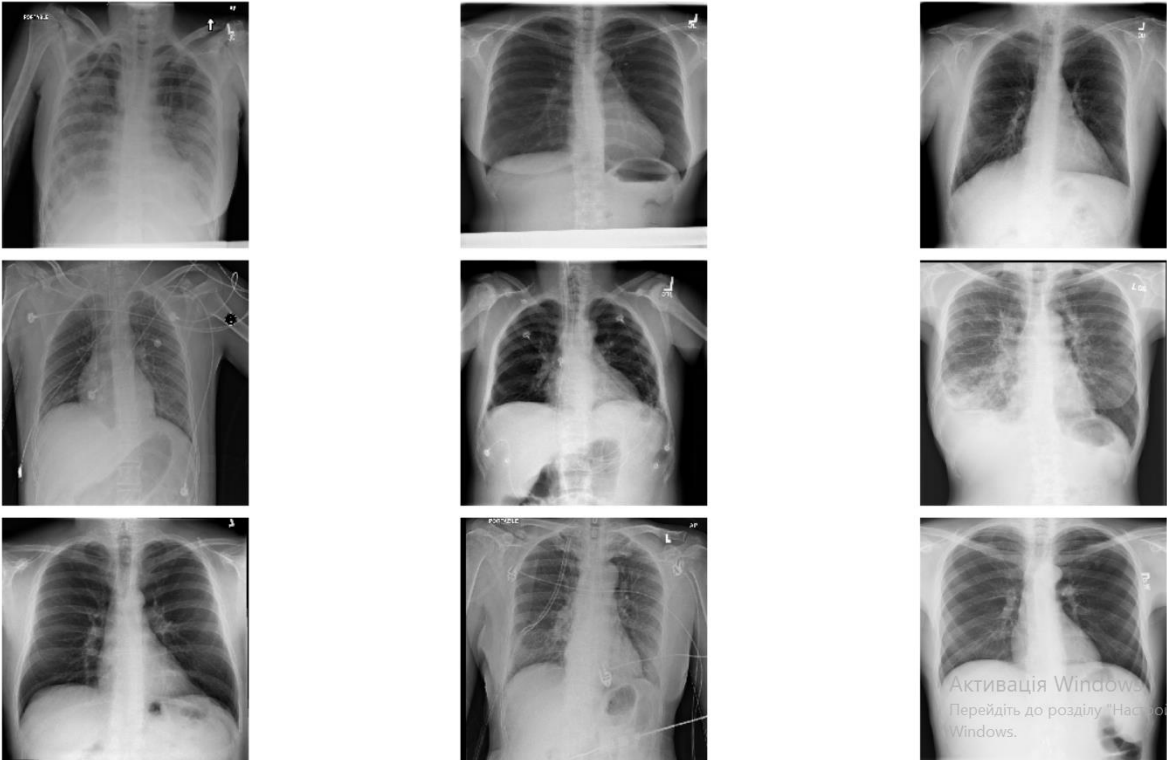


Рисунок 3.8 – Випадковий вибір зображень

Далі переглянемо перше зображення в наборі даних і роздрукуємо деякі деталі вмісту зображення (рис. 3.9).

```

sample_img = train_df.Image[0]
raw_image = plt.imread(os.path.join(img_dir, sample_img))
plt.imshow(raw_image, cmap='gray')
plt.colorbar()
plt.title('Raw Chest X Ray Image')
print(f"The dimensions of the image are {raw_image.shape[0]} pixels width and {raw_image.shape[1]} pixels height, one single color channel")
print(f"The maximum pixel value is {raw_image.max():.4f} and the minimum is {raw_image.min():.4f}")
print(f"The mean value of the pixels is {raw_image.mean():.4f} and the standard deviation is {raw_image.std():.4f}")

```

The dimensions of the image are 1024 pixels width and 1024 pixels height, one single color channel

The maximum pixel value is 0.9804 and the minimum is 0.0000

The mean value of the pixels is 0.4796 and the standard deviation is 0.2757

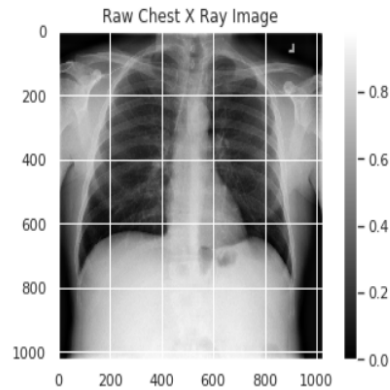


Рисунок 3.9 – Деталі вмісту зображення

Далі побудуємо графік розподілу значень пікселів на зображенні, яке було виведене на рисунку 3.6. Треба переглянути їх розподіл на цьому графіку (рис. 3.10).

```

In [10]: # Plot a histogram of the distribution of the pixels
sns.distplot(raw_image.ravel(),
             label=f'Pixel Mean {np.mean(raw_image):.4f} & Standard Deviation {np.std(raw_image):.4f}', kde=False)
plt.legend(loc='upper center')
plt.title('Distribution of Pixel Intensities in the Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('# Pixels in Image')

```

Out[10]: Text(0, 0.5, '# Pixels in Image')

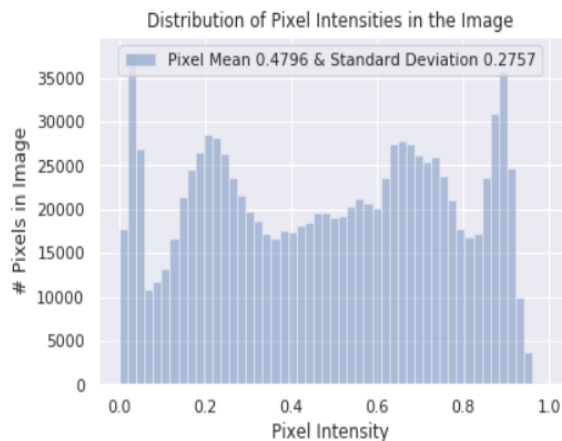


Рисунок 3.10 – Графік розподілу пікселів

ВИСНОВКИ

У рамках кваліфікаційної роботи було проаналізовано типи зображень у медицині й зазначено, що їх класифікують залежно від способу отримання.

В умовах захворювання на коронавірусну інфекцію, актуальною задачею є дослідження рентгенологічних зображень. Наведено загальну характеристику зображень в медицині, одержаних на основі діагностичних досліджень.

Проаналізовано алгоритми попереднього опрацювання зображень у медицині, які включають методи покращення зображення, сегментації та фільтрації кісткових тканин. Такі методи спрямовані на підвищення якості зображень з метою виявлення на них характерних об'єктів.

Досліджено характеристики об'єктів зображень у медицині, які за ознаками слід об'єднати у групи.

Методи ідентифікації об'єктів зображення складаються із кореляційних, ознакових, синтаксичних та нормалізаційних. Зазначено перспективність застосування ознакових алгоритмів за умов, коли немає чітких меж, розмірів інтервалів об'єктів.

Проаналізовано існуючі методи глибокого навчання та алгоритми попередньої обробки рентгенівських знімків. Зазначено, що найбільш ефективні для розв'язування задач розпізнавання образів є методи, які ґрунтуються на нейромережевому підході, зокрема, згорткові нейронні мережі.

Розроблено алгоритм розпізнавання рентгенологічних зображень, який включає процедуру попереднього опрацювання знімків

Результати дослідження апробовано у вигляді тез доповідей під час Міжнародного молодіжного форуму «Діджиталізація науки як виклик сьогодення» [56].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Granville, V. (2017). Difference between machine learning, data science, AI, deep learning, and statistics. *Data Science Central*..
2. Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., ... & Wang, Y. (2017). Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology*, 2(4).
3. Ganapathy, N., Swaminathan, R., & Deserno, T. M. (2018). Deep learning on 1-D biosignals: a taxonomy-based survey. *Yearbook of medical informatics*, 27(01), 098-109.
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
5. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60-88.
6. Mythili, T., Mukherji, D., Padalia, N., & Naidu, A. (2013). A heart disease prediction model using SVM-Decision Trees-Logistic Regression (SDL). *International Journal of Computer Applications*, 68(16).
7. Weng, C. H., Huang, T. C. K., & Han, R. P. (2016). Disease prediction with different types of neural network classifiers. *Telematics and Informatics*, 33(2), 277-292.
8. De Bruyne, S., Speeckaert, M. M., Van Biesen, W., & Delanghe, J. R. (2021). Recent evolutions of machine learning applications in clinical laboratory medicine. *Critical Reviews in Clinical Laboratory Sciences*, 58(2), 131-152.
9. Lee, S. W., Ali, S., Yousefpoor, M. S., Yousefpoor, E., Lalbakhsh, P., Javaheri, D., ... & Hosseinzadeh, M. (2021). An energy-aware and predictive fuzzy logic-based routing scheme in flying ad hoc networks (fanets). *IEEE Access*, 9, 129977-130005.
10. Tao, W., Concepcion, A. N., Vianen, M., Marijnissen, A. C., Lafeber, F. P., Radstake, T. R., & Pandit, A. (2021). Multiomics and machine learning

accurately predict clinical response to adalimumab and etanercept therapy in patients with rheumatoid arthritis. *Arthritis & Rheumatology*, 73(2), 212-222.

11. Alizadehsani, R., Roshanzamir, M., Abdar, M., Beykikhoshk, A., Khosravi, A., Panahiazar, M., ... & Sarrafzadegan, N. (2019). A database for using machine learning and data mining techniques for coronary artery disease diagnosis. *Scientific data*, 6(1), 1-13.

12. Golsorkhtabar, M., Nia, F. K., Hosseinzadeh, M., & Vejdandarast, Y. (2010, July). The novel energy adaptive protocol for heterogeneous wireless sensor networks. In *2010 3rd International Conference on Computer Science and Information Technology* (Vol. 2, pp. 178-182). IEEE.

13. Daradkeh, Y.I., Gorokhovatskyi, V., Tvoroshenko, I., Zeghid, M. (2022) Tools for Fast Metric Data Search in Structural Methods for Image Classification, *IEEE Access*, 10, pp. 124738–124746.

14. Nikravan, M., Movaghar, A., & Hosseinzadeh, M. (2018). A lightweight defense approach to mitigate version number and rank attacks in low-power and lossy networks. *Wireless Personal Communications*, 99(2), 1035-1059.

15. Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., ... & Nenadic, G. (2019). Machine learning methods for wind turbine condition monitoring: A review. *Renewable energy*, 133, 620-635.

16. Dhal, P., & Azad, C. (2021). A comprehensive survey on feature selection in the various fields of machine learning. *Applied Intelligence*, 1-39.

17. Tiwari, S. R., & Rana, K. K. (2021). Feature selection in big data: Trends and challenges. *Data Science and Intelligent Applications*, 83-98.

18. Guyon, I., & Elisseeff, A. (2006). An introduction to feature extraction. In *Feature extraction* (pp. 1-25). Springer, Berlin, Heidelberg.

19. Xiong, Z., Cui, Y., Liu, Z., Zhao, Y., Hu, M., & Hu, J. (2020). Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation. *Computational Materials Science*, 171, 109203.

20. Xu, Z., Qin, W., Tang, Q., & Jiang, D. (2014). Energy-efficient cognitive access approach to convergence communications. *Science China Information Sciences*, 57(4), 1-12.
21. Mandal, I. (2017). Machine learning algorithms for the creation of clinical healthcare enterprise systems. *Enterprise Information Systems*, 11(9), 1374-1400.
22. Feldman, K., Faust, L., Wu, X., Huang, C., & Chawla, N. V. (2017). Beyond volume: The impact of complex healthcare data on the machine learning pipeline. In *Towards integrative machine learning and knowledge extraction* (pp. 150-169). Springer, Cham.
23. Zhang, J. M., Harman, M., Ma, L., & Liu, Y. (2020). Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*.
24. Javaheri, D., Hosseinzadeh, M., & Rahmani, A. M. (2018). Detection and elimination of spyware and ransomware by intercepting kernel-level system routines. *IEEE Access*, 6, 78321-78332.
25. Mesbahi, M. R., Rahmani, A. M., & Hosseinzadeh, M. (2017). Highly reliable architecture using the 80/20 rule in cloud computing datacenters. *Future Generation Computer Systems*, 77, 77-86.
26. Setiawan, A. W. (2020, November). Image segmentation metrics in skin lesion: accuracy, sensitivity, specificity, dice coefficient, Jaccard index, and Matthews correlation coefficient. In *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)* (pp. 97-102). IEEE.
27. Werpachowski, R., György, A., & Szepesvári, C. (2019). Detecting overfitting via adversarial examples. *Advances in Neural Information Processing Systems*, 32.
28. Berry, M. W., Mohamed, A., & Yap, B. W. (Eds.). (2019). *Supervised and unsupervised learning for data science*. Springer Nature.

29. Caruana, R., & Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* (pp. 161-168).
30. alias Balamurugan, A., Rajaram, R., Pramala, S., Rajalakshmi, S., Jeyendran, C., & Prakash, J. D. S. (2011). NB+: an improved naive Bayesian algorithm. *Knowledge-Based Systems*, 24(5), 563-569.
31. Ballard, Z., Brown, C., Madni, A. M., & Ozcan, A. (2021). Machine learning and computation-enabled intelligent sensor design. *Nature Machine Intelligence*, 3(7), 556-565.
32. Miorelli, R., Kulakovskiy, A., Chapuis, B., D'almeida, O., & Mesnil, O. (2021). Supervised learning strategy for classification and regression tasks applied to aeronautical structural health monitoring problems. *Ultrasonics*, 113, 106372.
33. Dhasaradhan, K., Jaichandran, R., Shunmuganathan, K. L., Usha Kiruthika, S., & Rajaprakash, S. (2021). Hybrid machine learning model using decision tree and support vector machine for diabetes identification. In *Data Engineering and Intelligent Computing* (pp. 293-305). Springer, Singapore.
34. Shrestha, Y. R., Krishna, V., & von Krogh, G. (2021). Augmenting organizational decision-making with deep learning algorithms: Principles, promises, and challenges. *Journal of Business Research*, 123, 588-603.
35. Villarrubia, G., De Paz, J. F., Chamoso, P., & De la Prieta, F. (2018). Artificial neural networks used in optimization problems. *Neurocomputing*, 272, 10-16.
36. Zubair Hasan, K. M., & Hasan, Z. (2019). Performance evaluation of ensemble-based machine learning techniques for prediction of chronic kidney disease. In *Emerging research in computing, information, communication and applications* (pp. 415-426). Springer, Singapore.
37. Gottwald, G. A., & Reich, S. (2021). Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation. *Physica D: Nonlinear Phenomena*, 423, 132911.

38. Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
39. Piccialli, F., Di Somma, V., Giampaolo, F., Cuomo, S., & Fortino, G. (2021). A survey on deep learning in medicine: Why, how and when?. *Information Fusion*, 66, 111-137.
40. Xu, X., Zuo, L., & Huang, Z. (2014). Reinforcement learning algorithms with function approximation: Recent advances and applications. *Information Sciences*, 261, 1-31.
41. Qin, J., Chen, L., Liu, Y., Liu, C., Feng, C., & Chen, B. (2019). A machine learning methodology for diagnosing chronic kidney disease. *IEEE Access*, 8, 20991-21002.
42. Li, J. P., Haq, A. U., Din, S. U., Khan, J., Khan, A., & Saboor, A. (2020). Heart disease identification method using machine learning classification in e-healthcare. *IEEE Access*, 8, 107562-107582.
43. Abdar, M., Zomorodi-Moghadam, M., Zhou, X., Gururajan, R., Tao, X., Barua, P. D., & Gururajan, R. (2020). A new nested ensemble technique for automated diagnosis of breast cancer. *Pattern Recognition Letters*, 132, 123-131.
44. Ma, F., Sun, T., Liu, L., & Jing, H. (2020). Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network. *Future Generation Computer Systems*, 111, 17-26.
45. Wang, L., Zhang, W., He, X., & Zha, H. (2018, July). Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2447-2456).
46. Dai, Y., Wang, G., Muhammad, K., & Liu, S. (2020). A closed-loop healthcare processing approach based on deep reinforcement learning. *Multimedia Tools and Applications*, 1-23.
47. Tseng, H. H., Luo, Y., Cui, S., Chien, J. T., Ten Haken, R. K., & Naqa, I. E. (2017). Deep reinforcement learning for automated radiation adaptation in lung cancer. *Medical physics*, 44(12), 6690-6705.

48. Khalilpourazari, S., & Hashemi Doulabi, H. (2022). Designing a hybrid reinforcement learning based algorithm with application in prediction of the COVID-19 pandemic in Quebec. *Annals of Operations Research*, 312(2), 1261-1305.
49. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
50. Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96, 120-133.
51. Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 89, 228-249.
52. Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.
53. Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110, 151-166.
54. Salimi, H. (2015). Stochastic fractal search: a powerful metaheuristic algorithm. *Knowledge-Based Systems*, 75, 1-18.
55. Baucum, M., Khojandi, A., & Vasudevan, R. (2020). Improving deep reinforcement learning with transitional variational autoencoders: A healthcare application. *IEEE Journal of Biomedical and Health Informatics*, 25(6), 2273-2280.
56. Берковський, Д. (2022). Дослідження методів діагностики та лікування захворювання за допомогою глибокого навчання для системи охорони здоров'я, 122, 225.