

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

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

(повна назва інституту/факультету)

кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

(повна назва кафедри)

«На правах рукопису»

УДК _____

«До захисту допущено»

В.о. завідувача кафедри БМК

_____ Світлана АЛХІМОВА
(підпис) (ініціали, ПРИЗВИЩЕ)

« ____ » _____ 2025р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Комп'ютерні технології в біології та
медицині»

зі спеціальності 122 «Комп'ютерні науки»

на тему: Система підтримки прийняття рішень для лікаря невролога

Виконав: студент II курсу, групи ЗК-41мп

ДОРОШЕНКО ДЕНИС ВІКТОРОВИЧ

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник: доцент каф. біомедичної кібернетики

к.т.н, доцент Кравченко Ольга Віталіївна

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

(підпис)

Консультант з розділів магістерської дисертації:

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

(підпис)

Рецензент: *доцент кафедри біомедичної інженерії, к.т.н.*

доцент Рудніцька Олена Володимирівна

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

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань

Студент _____
(підпис)

Київ – 2026 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет біомедичної інженерії
Кафедра біомедичної кібернетики**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні технології в біології та медицині»

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри БМК

Світлана АЛХІМОВА

«__» _____ 2025 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

ДОРОШЕНКУ ДЕНИСУ ВІКТОРОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема дисертації **Система підтримки прийняття рішень для лікаря
невролога**

науковий керівник дисертації

к.т.н., доцент Кравченко Ольга Віталіївна

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

затверджені наказом по університету від «03» листопада 2025 р. №4751-с

2. Термін подання студентом дисертації **20-22 листопада 2025 року**

3. Об'єкт дослідження: Нейромережі, системи підтримки прийняття рішень, медичні дані.

4. Вихідні дані: теоретичні матеріали з глибокого навчання, практичні знання програмування мовою Python, набір медичних даних та методи побудови систем підтримки прийняття рішень.

5. Перелік завдань, які потрібно розробити: Проаналізувати існуючі інформаційні джерела / документацію та рішення за темою МД. Провести аналіз інструментів реалізації та обґрунтувати обрані засоби реалізації нейромережі. Натренувати мережі та виконати класифікацію на даних, розробити графічний інтерфейс застосунку, інтегрувати натреновану нейромережу в графічний інтерфейс. Зробити загальні висновки.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: 16 слайдів презентації з захисту МД.

7. Орієнтовний перелік публікацій: публікацій не заплановано.

8. Консультанти розділів дисертації *не передбачено*

9. Дата видачі завдання **28 серпня 2025р**

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів МД	Примітка
1	Отримати завдання на МД	До 28.08.2025	<i>виконано</i>
2	Завершення виконання практичної частина МД	До 25.10.2025	<i>виконано</i>
3	Завершення оформлення розділів МД (Вступ, Основні розділи та висновки до них, Загальні висновки, Список використаних джерел)	До 24.11.2025	<i>виконано</i>
4	Апробація та публікація результатів дослідження МД (отримати підтвердження про прийняття до друку)	До 01.12.2025	<i>виконано</i>
5	Перевірка МД науковим керівником	24.11.2025	<i>виконано</i>
6	Подання в електронному вигляді МД на перевірку нормоконтролера	До 01.12.2025	
7	Подання в електронному вигляді МД на перевірку подібності Strike Plagiarism com. Отримати позитивний звіт подібності.	До 05.12.2025	
8	Підготовка Експертної оцінки до звіту подібності.	До 10.12.2025	
9	Отримати відгук наукового керівника	10.12.2025	
10	Надати на кафедру пакет документів в паперовому та електронному вигляді (МД, відгук керівника, звіт подібності, Експертної оцінки до звіту подібності)	10-11.12.2025	
11	Отримати допуск до захисту МД в ЕК та направлення до рецензента (засідання кафедри)	Засідання кафедри	
12	Подання МД рецензенту. Отримати на надання рецензію до ЕК / на кафедру	До 19.12.2025	
13	Подання супровідного пакету документів по МД до захисту в ЕК	До 19.12.2025	
14	Захист МД в ЕК	22 – 26.12.2025	

Студент


_____ (підпис)

Денис ДОРОШЕНКО

(ім'я, ПРІЗВИЩЕ)

Науковий керівник


_____ (підпис)

Ольга КРАВЧЕНКО

(ім'я, ПРІЗВИЩЕ)

Нормоконтролер

_____ (підпис)

Галина КОРНІЄНКО

(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Магістерська дисертація за темою «Система підтримки прийняття рішень для лікаря-невролога» виконана студентом кафедри Біомедичної кібернетики ФБМІ Дорошенком Денисом Вікторовичем зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та складається зі: вступу; 4 розділів («ОГЛЯД ПРОБЛЕМИ ДІАГНОСТИКИ В НЕВРОЛОГІЇ», «АНАЛІЗ АНАЛОГІВ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ», «ЗАСОБИ РЕАЛІЗАЦІЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ», «ПРОГРАМНА РЕАЛІЗАЦІЯ ТА МЕТОДИКА РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ», «РОЗРОБКА СТАРТАП ПРОЄКТУ»), розділу зі стартап проєкту, висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 47 джерела та додатків. Загальний обсяг роботи 114 сторінок, містить 50 рисунків, 4 таблиць.

Актуальність теми. Дисертація присвячена розробці програмного забезпечення для підтримки прийняття клінічних рішень у сфері неврології. Актуальність теми зумовлена необхідністю підвищення швидкості та точності діагностики неврологічних захворювань, а також складністю аналізу великих обсягів клінічних даних лікарем. У роботі досліджено та проаналізовано існуючі системи підтримки прийняття рішень (СППР) і програмні рішення, що застосовуються у медичній практиці.

Мета і задачі дослідження. Метою роботи є покращення наявних СППР за допомогою методів глибокого навчання у вигляді Python-додатку з графічним інтерфейсом, яка забезпечує класифікацію неврологічних діагнозів на основі введених симптомів і надає рекомендації, сформовані за міжнародними клінічними протоколами. Сформульовано наступні задачі (назви розділів, що відображають завдання, які виконано):

1. Проведення аналізу актуальних проблем діагностики неврологічних захворювань та огляд існуючих систем підтримки прийняття рішень.

2. Вибір та обґрунтування методів глибинного навчання для класифікації неврологічних діагнозів.
3. Розробка архітектури та реалізація багат шарової нейронної мережі для прогнозування.
4. Програмне впровадження системи «NeurologyPredictor» з інтуїтивно зрозумілим інтерфейсом користувача та її експериментальна оцінка.

Об'єкт дослідження. Процес підтримки прийняття діагностичних рішень у неврології за допомогою програмних засобів.

Предмет дослідження. Методи глибинного навчання для класифікації симптомів та підходи до побудови інтуїтивно зрозумілого інтерфейсу користувача.

Методи дослідження. У роботі використано методи глибинного навчання, зокрема багат шарову нейронну мережу. Також застосовано методи аналізу клінічних даних, що ґрунтуються на клінічних протоколах МОЗ України, рекомендаціях NICE, NHS та NIH, а також відкритий датасет із симптомами та діагнозами. Розроблена нейронна мережа досягла точності понад 94% на тестових даних.

Практичне значення одержаних результатів. Практичне значення роботи полягає у створенні діючого прототипу СППР «NeurologyPredictor», який може застосовуватися для підвищення швидкості та точності діагностики неврологічних захворювань, а також слугувати основою для подальшої інтеграції у медичні інформаційні системи.

Апробація результатів дослідження. Результати роботи проходили апробацію на 1 науковій конференції:

Д.В. Дорошенко, О.В. Кравченко. Система підтримки прийняття рішень для лікаря невролога // Матеріали XII Міжнародній науково-технічній Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами». К: НУХТ, 2025, С.196-198

Ключові слова. Система підтримки прийняття рішень, Неврологія, Глибинне навчання, Нейронна мережа, Діагностика, Класифікація, Протокол, Симптоми.

Бібліографічний опис МД

Дорошенко, Д. В. Система підтримки прийняття рішень для лікаря-невролога : магістерська дис. : 122 Комп'ютерні науки / Дорошенко Денис Вікторович. – Київ, 2025. – 114с.

ABSTRACT

Master's Thesis on the topic "Decision Support System for a Neurologist" was completed by a student of the Department of Biomedical Cybernetics FBE Denys V. Doroshenko in Specialty 122 "Computer Science" under the Educational-Professional Program "Computer Technologies in Biology and Medicine". The thesis consists of: an introduction; 4 chapters ("OVERVIEW OF DIAGNOSTIC CHALLENGES IN NEUROLOGY", "ANALYSIS OF CLINICAL DECISION SUPPORT SYSTEM ANALOGUES", "IMPLEMENTATION TOOLS FOR THE DECISION SUPPORT SYSTEM", "SOFTWARE IMPLEMENTATION AND METHODOLOGY OF OPERATION", "STARTUP PROJECT DEVELOPMENT"), a chapter on the startup project, conclusions for each of these chapters; general conclusions; a list of 47 sources and appendices. The total volume of the work is 114 pages, containing 50 figures and 4 tables.

Actuality of the Topic (Relevance). The thesis is dedicated to the development of software for clinical decision support in the field of neurology. The relevance of the topic is due to the need to increase the speed and accuracy of neurological disease diagnosis, as well as the complexity for a physician to analyze large volumes of clinical data. The work investigates the current challenges in diagnosing neurological diseases and analyzes existing Decision Support Systems (DSS) and software solutions used in medical practice.

Aim and Objectives of the Research. The goal of the work is to improve existing DSS using methods of deep learning in the form of a Python application with a graphical user interface that provides classification of neurological diagnoses based on entered symptoms and offers recommendations formulated according to international clinical protocols. The following objectives (chapter titles reflecting the tasks performed) were formulated:

1. Analysis of current neurological diagnosis issues and a review of existing decision support systems.

2. Selection and justification of deep learning methods for classifying neurological diagnoses.
3. Development of the architecture and implementation of a multilayer neural network for prognosis.
4. Software implementation of the "NeurologyPredictor" system with an intuitive user interface and its experimental evaluation.

Object of the Research. The process of supporting diagnostic decision-making in neurology using software tools.

Subject of the Research. Deep learning methods for symptom classification and approaches to building an intuitive user interface.

Research Methods. The work utilized deep learning methods, specifically a multilayer neural network. Methods of clinical data analysis were also applied, based on clinical protocols from the Ministry of Health of Ukraine, NICE, NHS, and NIH recommendations, as well as an open dataset of symptoms and diagnoses. The developed neural network achieved an accuracy of over 94% on test data.

Practical Significance of the Results. The practical significance lies in the creation of a working prototype of the DSS "NeurologyPredictor", which can be applied to increase the speed and accuracy of neurological disease diagnosis and serve as a basis for further integration into medical information systems.

Approbation of Research. Results The results of the work were approved at 1 scientific conference: D.V. Doroshenko, O.V. Kravchenko. Decision Support System for a Neurologist // Materials of the XII International Scientific and Technical Internet Conference "Modern Methods, Information, Software and Technical Support for Control Systems of Organizational, Technical and Technological Complexes". K: NUFT, 2025, P. 196-198.

Key Words. Decision Support System, Neurology, Deep Learning, Neural Network, Diagnosis, Classification, Protocol, Symptoms.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	11
ВСТУП.....	12
РОЗДІЛ 1 ОГЛЯД ПРОБЛЕМИ ДІАГНОСТИКИ В НЕВРОЛОГІЇ ТА СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ	16
1.1. Загальна характеристика неврологічних захворювань.....	16
1.2. Актуальні проблеми діагностики в неврології.....	17
1.3. СППР: сутність, переваги та проблеми впровадження	20
1.3.1. Сутність СППР.....	21
1.3.2. Приклади застосування у медицині.....	23
1.3.3. Переваги СППР.....	25
1.3.4. Проблеми та обмеження СППР	26
Висновки до розділу 1	28
РОЗДІЛ 2 АНАЛІЗ АНАЛОГІВ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ	30
2.1. Огляд існуючих комерційних і наукових рішень	30
2.2. Технічні підходи в аналогах (архітектури, дані).....	34
2.3. Порівняльний аналіз: сильні та слабкі сторони аналогів.....	42
2.4. Обґрунтування вибору підходу для власної системи	44
Висновки до розділу 2	46
РОЗДІЛ 3 ЗАСОБИ РЕАЛІЗАЦІЇ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ	47
3.1. Джерела даних та правил СППР	47
3.2. Джерела тренувальних даних.....	52
3.3. Технічне середовище та інструменти.....	54
3.4. Захист даних та безпека	58
Висновки до розділу 3	59
РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА МЕТОДИКА РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	61

	10
4.1. Попередня обробка тренувальних даних	62
4.2. Побудова і навчання моделі: архітектура, гіперпараметри	66
4.3. Метрики оцінки якості моделі	76
4.4. Опис модулів ПЗ і інтерфейсу користувача (GUI)	78
4.5. Алгоритм роботи: від введення симптомів до виводу рекомендацій	83
Висновки до розділу 4	94
РОЗДІЛ 5 РОЗРОБКА СТАРТАП-ПРОЄКТУ	96
5.1. Назва проєкту	96
5.2. Короткий опис проєкту	96
5.3. Бізнес-модель	97
5.3.1. Цінність продукту	97
5.3.2. Сегмент споживачів	98
5.3.3. Канали збуту	99
5.3.4. Взаємодія з споживачами	99
5.3.5. Дохід	100
5.3.6. Ключові види діяльності	100
5.3.7. Ключові ресурси	100
5.3.8. Ключові партнери	101
5.3.9. Структура витрат	102
5.3.10. Споживчі властивості товару	103
5.3.11. Дослідження ринку	104
5.3.12. Елементи фінансового плану	104
5.3.13. Резюме	105
Висновки до розділу 5	106
ЗАГАЛЬНІ ВИСНОВКИ	107
СПИСОК БІБЛЮГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ	109

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ШІ – Штучний інтелект.

СППР – Система підтримки прийняття рішень.

ПЗ – Програмне забезпечення.

GUI – Графічний інтерфейс користувача.

NICE – National Institute for Health and Care Excellence.

NIH – National Institutes of Health.

NHS – National Health Service.

МОЗ – Міністерство охорони здоров'я.

ANN – Artificial Neural Networks – Штучні нейронні мережі.

CNN – Convolutional Neural Networks – Конволюційні нейронні мережі.

ВСТУП

У сучасній медицині спостерігається стрімке зростання обсягу клінічних даних і підвищення складності діагностичного процесу, особливо в галузі неврології. Щорічна кількість неврологічних обстежень значно зростає через поширеність захворювань центральної та периферичної нервової системи, включаючи інсульт, розсіяний склероз та нейродегенеративні хвороби. В умовах високого навантаження на лікарів і потреби в ранній та точній діагностиці виникає гостра потреба в розробці інструментів, що полегшують і прискорюють прийняття клінічних рішень та знижують ризик діагностичних помилок.

Актуальність роботи зумовлена необхідністю впровадження інтелектуальних систем підтримки прийняття рішень (СППР) у неврологічну практику. Існуючі рішення часто ґрунтуються на жорстко прописаних правилах або моделях машинного навчання, які потребують значних зусиль із підготовки даних та складно інтегруються в робочий процес лікаря. Удосконалення таких систем передбачає поєднання аналітичних можливостей штучного інтелекту з гнучкістю адаптації під конкретні клінічні випадки.

Для досягнення поставленої мети використовується розробка СППР у вигляді Python-додатку з графічним інтерфейсом (GUI), який застосовує методи глибинного навчання, зокрема багатошарову нейронну мережу, для класифікації неврологічних діагнозів на основі введених симптомів. Такий підхід забезпечує більш точну і послідовну інтерпретацію даних, підвищуючи ефективність роботи лікаря-невролога.

Мета і завдання роботи

Метою роботи є удосконалення наявних систем підтримки прийняття рішень для лікаря-невролога методами глибокого навчання у вигляді Python-додатку з графічним інтерфейсом, що використовує навчений алгоритм класифікації неврологічних станів на основі введених клінічних даних, з

метою суттєвого підвищення ефективності діагностичного процесу та скорочення часу встановлення попереднього діагнозу.

Її досягнення передбачає вирішення наступних завдань:

1. Аналіз вітчизняних та зарубіжних джерел, а саме: проаналізувати предметну область неврологічної діагностики та оглянути сучасні інформаційні системи і алгоритми, що застосовуються для автоматизації процесу діагностики в медицині.
2. Визначити вимоги до системи, сформулювати функціональні та нефункціональні характеристики Python-додатку з GUI.
3. Зібрати та попередньо обробити набір даних клінічних параметрів пацієнтів з різними неврологічними станами; провести аналіз і чистку даних.
4. Розробити й реалізувати архітектуру моделі машинного навчання для класифікації неврологічних діагнозів, провести її тренування та оптимізацію гіперпараметрів.
5. Інтегрувати навчений алгоритм у графічний інтерфейс, реалізувати взаємодію користувача з моделлю, включаючи введення даних, відображення результатів і рекомендацій.
6. Провести тестування функціональності та оцінити якість системи за допомогою метрик точності, чутливості та специфічності. Об'єкт дослідження – процес підтримки прийняття діагностичних рішень у неврології за допомогою програмних засобів.

Предмет дослідження – методи та алгоритми машинного навчання для класифікації неврологічних станів, а також підходи до реалізації інтуїтивного графічного інтерфейсу для лікаря.

Використані методи. У роботі використано методи глибинного навчання, зокрема розроблено та реалізовано багатошарову нейронну мережу для класифікації неврологічних діагнозів. Для формування рекомендацій та забезпечення точності системи використовувалися дані клінічних протоколів

МОЗ України, а також міжнародні рекомендації NICE, NHS та NIH. Програмна реалізація системи виконана у вигляді Python-додатку з графічним інтерфейсом.

Отримані результати. Отриманим результатом є розроблений та реалізований діючий прототип СППР «NeurologyPredictor», який забезпечує класифікацію неврологічних діагнозів на основі введених симптомів. Розроблена багат шарова нейронна мережа досягла точності понад 94% на тестових даних. Створений додаток дозволяє лікарю вводити симптоми через інтерактивні елементи та отримувати прогноз і клінічні рекомендації, що має практичне значення для підвищення якості діагностики.

Апробація результатів роботи. За результатами виконаної роботи було опубліковано 1 матеріал на конференції:

Дорошенко Д.В., Кравченко О.В. Система підтримки прийняття рішень для лікаря невролога. Матеріали XII Міжнародній науково-технічній Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами». К: НУХТ, 2025, С.196-198.

Структура роботи

Магістерська дисертація за темою «Система підтримки прийняття рішень для лікаря-невролога» виконана студентом Дорошенком Денисом Вікторовичем зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині», побудована за класичним типом та викладена на 108 сторінках машинописного тексту. Вона складається з: вступу; 5 розділів: «ОГЛЯД ПРОБЛЕМИ ДІАГНОСТИКИ В НЕВРОЛОГІЇ», «АНАЛІЗ АНАЛОГІВ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ», «ЗАСОБИ РЕАЛІЗАЦІЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ», «ПРОГРАМНА РЕАЛІЗАЦІЯ ТА МЕТОДИКА РОБОТИ ПЗ», «РОЗРОБКА СТАРТАП ПРОЄКТУ», висновків до кожного з цих розділів; загальних висновків; списку

використаних джерел, який налічує 47 джерел та додатків (код програми). В роботі представлено 50 рисунків і 4 таблиці.

РОЗДІЛ 1

ОГЛЯД ПРОБЛЕМИ ДІАГНОСТИКИ В НЕВРОЛОГІЇ ТА СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

У цьому розділі представлено теоретичні основи та актуальні проблеми діагностики в неврології, що обумовлюють необхідність створення системи підтримки прийняття рішень на базі штучного інтелекту. Розглянуто медичний контекст роботи, зокрема ключові виклики, пов'язані з високою частотою атипового перебігу неврологічних захворювань, значною міжхворобною схожістю симптомів (наприклад, диференціальна діагностика есенціального тремору та хвороби Паркінсона), а також залежністю точності діагнозу від суб'єктивного досвіду та кваліфікації лікаря.

Основну увагу приділено аналізу класичних СППР та їхніх архітектур, заснованих на базах знань і логічних правилах. Проаналізовано їхні структурні недоліки, такі як трудомісткість оновлення знань та, що обмежує їхню ефективність у роботі з неповними або нечіткими клінічними даними.

На завершення визначено та обґрунтовано ключові вимоги до сучасної СППР у сфері неврології. Це дослідження підводить до висновку, що для забезпечення високої точності та адаптивності діагностики необхідно використовувати парадигму машинного навчання, яка здатна автоматично виявляти складні, нелінійні взаємозв'язки між множиною симптомів та кінцевим діагнозом.

1.1. Загальна характеристика неврологічних захворювань

Неврологічні захворювання охоплюють широкий спектр порушень, які негативно впливають на функціонування центральної та периферичної нервової системи. До центральних нервових захворювань належать такі патології, як інсульт, демієлінізуючі хвороби (зокрема розсіяний склероз),

дегенеративні процеси (наприклад, хвороби Паркінсона та Альцгеймера), а також судинні ураження, пухлини та запалення головного та спинного мозку. Периферичні ураження включають полінейропатії, гострі й хронічні ідіопатичні демієлінізуючі поліневропатії, а також ураження нейром'язових синапсів. такі захворювання можуть мати різноманітні клінічні прояви - від сенсорних і моторних порушень до когнітивних та автономних дисфункцій [1].

Однією з найрозповсюдженіших і клінічно важливих серед демієлінізуючих захворювань центральної нервової системи є розсіяний склероз, який характеризується атакою імунної системи на мієлінові оболонки нейронів, що призводить до порушення передачі нервових імпульсів і формування вогнищ демієлінізації в різних ділянках мозку та спинного мозку. Симптоми можуть коливатися від порушень зору і моторної координації до дезорієнтації та когнітивних розладів, а перебіг хвороби включає фази загострення й ремісії або прогресуючого погіршення стану [2].

Периферичні демієлінізуючі нейропатії, що також є важливою категорією, охоплюють, зокрема, синдром Гійєна–Барре - аутоімунне гостре ураження периферичних нервів, що швидко прогресує й може бути небезпечним для життя, та хронічну демієлінізуючу поліневропатію (CIDP), яка характеризується поступовою слабкістю й порушенням чутливості в кінцівках. Дані патології порушують передачу нервових сигналів через ушкодження мієліну периферичних нервів, що призводить до слабкості, сенсорних дефіцитів і, серед ускладнень, можливих порушень дихання й автономних функцій [1], [2].

1.2. Актуальні проблеми діагностики в неврології

Діагностика неврологічних захворювань залишається однією з найскладніших галузей сучасної медицини через виняткову різноманітність клінічних проявів та високий рівень міжхворобної схожості симптомів. У неврології часто один і той самий симптом може мати різну етіологію та бути

характерним для кількох абсолютно різних патологій, що значно ускладнює процес встановлення точного діагнозу. Крім того, низка неврологічних хвороб має повільний і поступовий початок, що призводить до недооцінки перших ознак як самим пацієнтом, так і лікарем первинної ланки. Це знижує ймовірність вчасного виявлення захворювання та своєчасного початку лікування [1].

Особливої складності набуває діагностика на ранніх стадіях, коли клінічна картина є ще недостатньо вираженою. Наприклад, слабкість у кінцівках або порушення координації рухів можуть свідчити як про периферичну нейропатію, так і про початковий етап демієлінізуючого процесу. Подібні симптоми також можуть бути проявом ішемічних порушень у головному мозку або наслідком травматичних ушкоджень. Додатково ситуацію ускладнює той факт, що багато пацієнтів мають супутні захворювання, які маскують або посилюють неврологічні прояви. У результаті лікар змушений проводити складний диференційний аналіз, враховувати анамнез, результати інструментальних та лабораторних досліджень і лише після цього формувати остаточний діагноз. Саме тому в сучасній практиці діагностика неврологічних захворювань потребує не лише висококваліфікованого клінічного підходу, але й використання допоміжних інтелектуальних систем, здатних структурувати знання та підтримати лікаря у процесі прийняття рішень [1].

Іншою проблемою є обмежений доступ до сучасних методів візуалізації та лабораторної діагностики у деяких регіонах, що може призводити до пізнього виявлення патологій або постановки неточних діагнозів. Використання магнітно-резонансної томографії (МРТ), електроенцефалографії (ЕЕГ), електроміографії (ЕМГ) та біохімічних маркерів значно підвищує точність діагностики, однак ці методи не завжди доступні для широкого кола пацієнтів. Брак технічного оснащення, довгі черги на проведення обстежень, а також висока вартість сучасних діагностичних процедур створюють додаткові бар'єри для своєчасного встановлення

правильного діагнозу. У результаті лікарі часто змушені спиратися лише на клінічну картину та базові методи дослідження, що підвищує ймовірність діагностичних помилок і затримує початок адекватного лікування [3].

Високий рівень клінічної варіативності та наявність коморбідних станів також ускладнюють оцінку симптомів. Багато хворих мають супутні серцево-судинні, ендокринні або метаболічні захворювання, які можуть маскувати неврологічну симптоматику або змінювати її перебіг. Наприклад, цукровий діабет здатен спричиняти периферичні нейропатії, що ускладнює диференційну діагностику з розсіяним склерозом чи іншими демієлінізуючими процесами. Аналогічно, серцево-судинні порушення можуть проявлятися запамороченням або короткочасною втратою свідомості, що потребує від лікаря відрізнити їх від епілепсії чи транзиторних ішемічних атак. У таких випадках стає очевидною необхідність комплексного підходу та інтеграції різних джерел інформації - клінічного огляду, інструментальних досліджень і лабораторних аналізів - під час постановки діагнозу.

Крім того, традиційні методи діагностики є трудомісткими та значною мірою залежать від досвіду лікаря, що створює ризик суб'єктивних помилок. У сучасних умовах, коли обсяг медичних даних зростає експоненційно, а швидкість прийняття клінічних рішень набуває вирішального значення, збільшується потреба у впровадженні комп'ютерних систем підтримки клінічних рішень. До таких рішень належать СППР та алгоритми глибокого навчання, які здатні автоматизувати первинну оцінку симптомів, проводити попередню класифікацію захворювань і навіть пропонувати диференційний ряд можливих діагнозів. Це дозволяє підвищити точність і швидкість діагностики, а також знизити навантаження на лікарів, особливо у випадках великої кількості пацієнтів [1].

Сучасна практика показує, що інтеграція даних пацієнта з результатами нейровізуалізації, лабораторними тестами та електрофізіологічними показниками дозволяє значно підвищити точність діагностики і скоротити час від звернення пацієнта до постановки точного діагнозу. Комплексний аналіз

МРТ або КТ-зображень у поєднанні з результатами ЕЕГ, ЕМГ та біохімічних досліджень забезпечує більш повне уявлення про перебіг захворювання та дозволяє мінімізувати ризик пропуску важливих симптомів. Такий підхід особливо важливий у випадках невідкладних станів, наприклад при підозрі на інсульт, коли швидкість і точність діагностики безпосередньо впливають на прогноз пацієнта [3].

Проте ефективність подібних систем залежить не лише від їхньої технічної реалізації, а й від повноти та якості вихідних даних. Неповні медичні записи, різні формати зберігання інформації чи низька якість візуалізацій можуть суттєво знизити діагностичну цінність результатів. Важливим обмежувальним фактором є й необхідність врахування індивідуальних особливостей пацієнтів - віку, супутніх хвороб, генетичних факторів, способу життя, які часто залишаються поза межами автоматизованого аналізу. Саме тому для підвищення ефективності таких систем важливо поєднувати алгоритмічну обробку з експертною оцінкою лікаря, що забезпечує баланс між об'єктивними даними та клінічним досвідом [3].

1.3. СППР: сутність, переваги та проблеми впровадження

СППР належать до класу інтелектуальних програмних комплексів, створених для імітації процесу прийняття рішень висококваліфікованими фахівцями у певній предметній галузі. Їх поява у 1970–1980-х роках стала важливим етапом розвитку штучного інтелекту, адже вони дозволили формалізувати знання експертів і відтворювати їх у вигляді алгоритмізованих правил.

Особливо значущим виявилось застосування таких систем у медицині, де клінічні рішення часто залежать від великої кількості змінних, взаємопов'язаних симптомів і ризиків. Завдяки здатності структурувати знання та застосовувати їх у вигляді правил типу «якщо – то», СППР стали фундаментом для створення сучасних систем підтримки прийняття рішень у

діагностиці та лікуванні. Це дозволило стандартизувати медичні підходи та надати лікарям додатковий інструмент у складних клінічних ситуаціях [5].

На рис. 1.1 можна побачити на якому рівні застосовується база знань. На відміну від традиційного програмного забезпечення, яке виконує чітко визначені інструкції, СППР працюють із базою знань та застосовують логічний апарат для виведення нових висновків із наявної інформації [4].

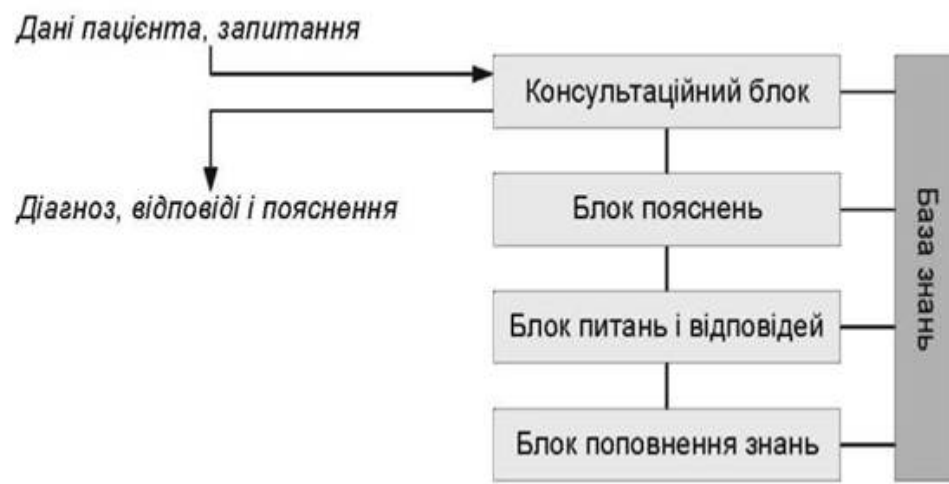


Рисунок 1.1 – Структурна схема базової СППР

1.3.1. Сутність СППР

СППР - це програмний комплекс, що складається з кількох ключових компонентів, кожен із яких виконує окрему функцію. Центральним елементом виступає база знань, у якій накопичені факти, закономірності та правила, отримані від експертів у відповідній галузі. У випадку медицини база знань містить опис симптомів, клінічних синдромів, діагнозів і відповідних рекомендацій. База знань може зберігати як прості правила на кшталт «якщо спостерігається симптом А і симптом В, то ймовірний діагноз С», так і складні багаторівневі схеми, що враховують комбінації факторів [6].

Другим важливим елементом є механізм виведення - логічний апарат, який оперує правилами та фактами, закладеними у базу знань, для отримання нових висновків. У класичних СППР механізм виведення базується на

продукційних правилах («if – then») і може використовувати прямий (від даних до висновків) або зворотний (від гіпотези до перевірки фактів) методи виведення. Завдяки цьому система здатна працювати як у режимі діагностики, так і в режимі пошуку причин конкретного симптому.

Третім елементом виступає інтерфейс користувача, який забезпечує взаємодію лікаря із системою (див. рис. 1.2). Через нього вводяться початкові дані (симптоми, лабораторні показники, результати обстежень) та відображаються результати - прогнозовані діагнози, ступінь імовірності та клінічні рекомендації. У сучасних реалізаціях інтерфейси мають інтуїтивно зрозумілий вигляд, що значно підвищує зручність роботи лікаря.

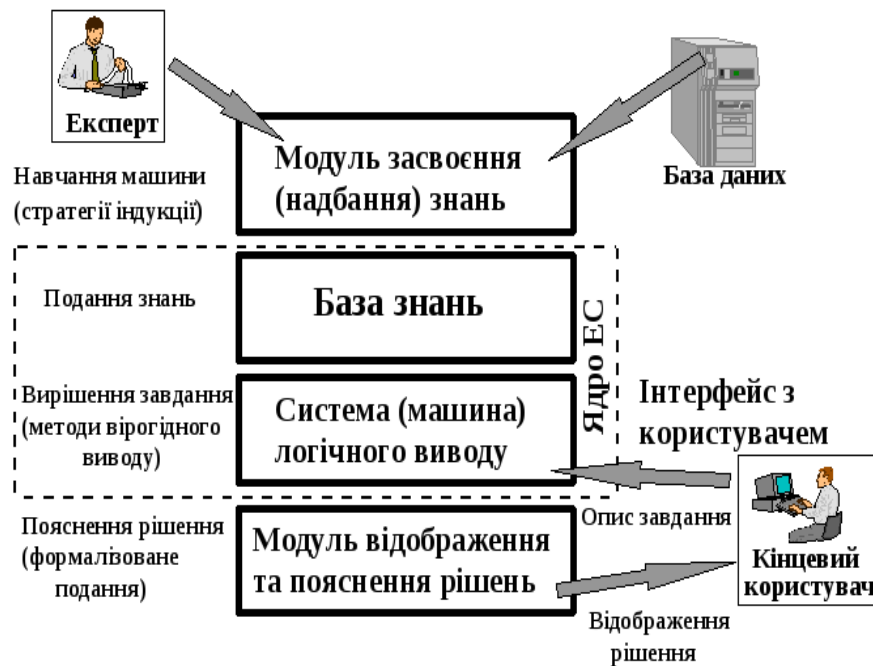


Рисунок 1.2 – Інтерфейс ідеальної СППР

Таким чином, СППР поєднує накопичений досвід фахівців у вигляді формалізованих знань із логічним апаратом їхнього застосування. Саме ця риса відрізняє її від традиційних програмних засобів: звичайні системи виконують заздалегідь визначені алгоритми, тоді як СППР здатні «мислити» в межах заданих правил, моделюючи процес прийняття рішень експертом.

1.3.2. Приклади застосування у медицині

Медицина стала однією з перших сфер, де СППР отримали практичне застосування. Висока складність клінічних рішень, велика кількість змінних і необхідність швидкої діагностики зробили цю галузь ідеальним полігоном для впровадження інтелектуальних технологій. Уже в 1970–1980-х роках дослідники активно експериментували з комп'ютерними моделями, здатними імітувати мислення лікаря. Такі системи не лише автоматизували процес ухвалення рішень, але й створювали основу для накопичення медичних знань у цифровому форматі. В табл. 1.1 виділено основні системи, їх застосування та особливості.

Таблиця 1.1 – Приклади СППР у медицині

Назва системи	Сфера застосування	Особливості
MYCIN	Інфекційні захворювання	Продукційні правила
INTERNIST-1	Внутрішні хвороби	Масштабна база знань
Сучасні CDSS	Мультидисциплінарні	Інтеграція з EHR

Одним із перших відомих прикладів була система MYCIN, розроблена у 1970-х роках у Стенфордському університеті. Вона призначалася для діагностики бактеріальних інфекцій крові та підбору антибіотиків [7]. MYCIN використовувала продукційні правила типу «якщо – то» та механізм зворотного виведення (див. рис. 1.3). Попри те, що ця система ніколи не була впроваджена у клінічну практику через етичні та юридичні питання, її результати показали, що СППР можуть перевершувати середнього за досвідом лікаря у складних діагностичних завданнях. Досвід розробки MYCIN став фундаментом для створення подальших медичних систем, таких як INTERNIST-I та CASNET, які вже орієнтувалися на ширший спектр хвороб. Її

принципи логічного висновування досі застосовуються в сучасних медичних рішеннях на базі штучного інтелекту.

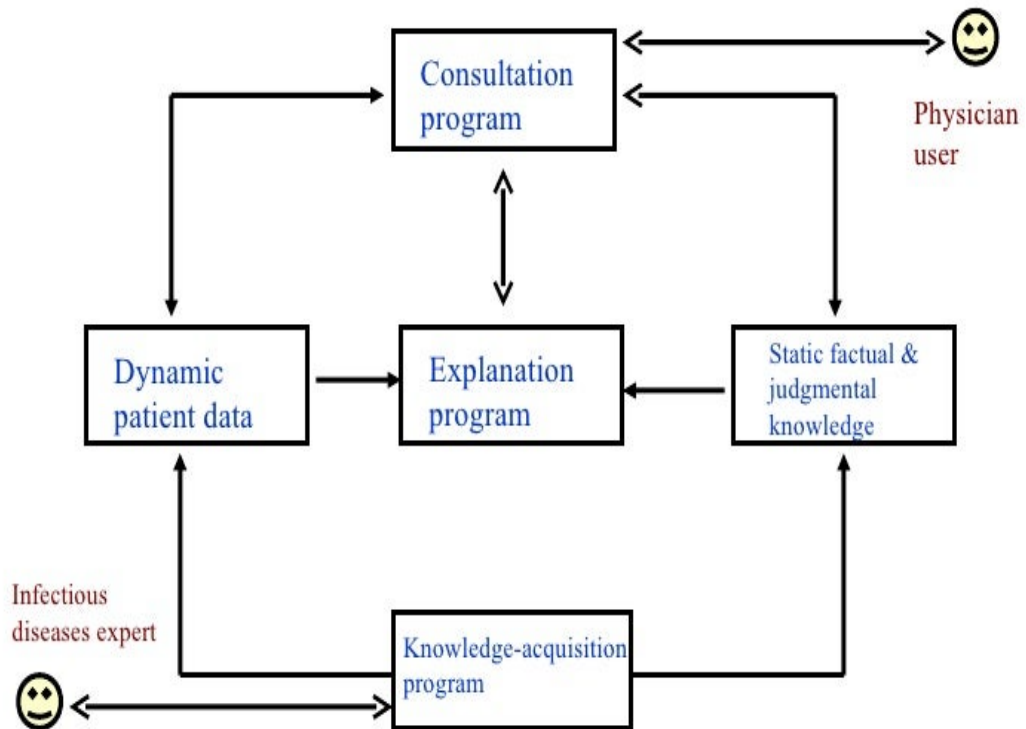


Рисунок 1.3 – Архітектура MYCIN

На рис 1.4 можна побачити приклад діаграми CASNET яка дозволяє визначати причинно-наслідкові ефекти медичного лікування за допомогою аналізу даних.

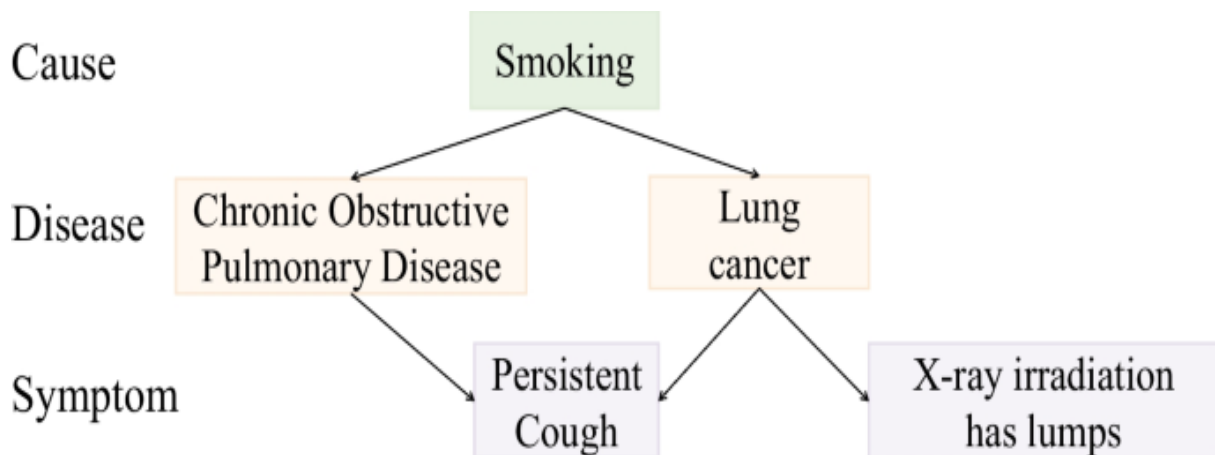


Рисунок 1.4 – приклад CASNET

Іншою знаковою розробкою стала система INTERNIST-1, спрямована на діагностику внутрішніх захворювань. Вона мала масштабну базу знань із понад 500 нозологій і 3500 симптомів [8]. Завдяки використанню великого масиву правил ця система стала прототипом сучасних клінічних систем підтримки прийняття рішень (Clinical Decision Support Systems, CDSS).

У сучасній медицині СППР застосовуються для широкого спектра завдань: від діагностики онкологічних та інфекційних захворювань до оцінки ризику серцево-судинних подій. У неврології вони допомагають ідентифікувати гострі стани, як-от інсульт чи черепно-мозкову травму, а також підтримують диференційну діагностику при хронічних захворюваннях, включаючи розсіяний склероз, епілепсію чи нейродегенеративні розлади [9].

Особливого значення СППР набувають у сфері телемедицини, де вони можуть надавати лікарям первинні рекомендації навіть у віддалених регіонах. Це сприяє підвищенню доступності медичних послуг та зниженню ризику помилкових рішень у випадках, коли досвідчені спеціалісти недоступні. Водночас вони можуть інтегруватися з електронними медичними записами (EHR), що дозволяє автоматично аналізувати історію хвороби пацієнта й генерувати персоналізовані підказки.

1.3.3. Переваги СППР

Використання СППР у медицині має низку суттєвих переваг, які пояснюють їхнє поширення та актуальність. Насамперед вони забезпечують швидкість обробки великих обсягів даних. Лікарю-неврологу часто доводиться аналізувати десятки симптомів і клінічних ознак одночасно, а також враховувати супутні захворювання пацієнта. У таких умовах СППР здатна за лічені секунди проаналізувати комбінації факторів і видати перелік можливих діагнозів, що значно пришвидшує процес прийняття рішень [10].

Ще однією перевагою є стандартизація діагностики та лікування. Оскільки система ґрунтується на єдиних клінічних протоколах і правилах, вона допомагає уникати суб'єктивних відмінностей у діагностичному процесі,

що можуть виникати між різними лікарями. Це особливо важливо у неврології, де один і той самий симптом (наприклад, головний біль чи запаморочення) може бути ознакою десятків різних патологій. Використання СППР сприяє уніфікації підходів та підвищує якість медичної допомоги [11].

Важливим аспектом є також доступність знань у віддалених регіонах. Лікарі, які працюють у сільських районах або невеликих лікарнях, часто не мають можливості консультуватися з високоспеціалізованими неврологами. СППР здатна частково компенсувати цей дефіцит, надаючи лікарю алгоритмізовану підтримку у постановці діагнозу чи виборі тактики лікування. Це робить медичну допомогу більш рівномірною за якістю в різних регіонах.

Крім того, СППР сприяють зниженню ризику людських помилок. Втома, високе навантаження, обмежений час на прийом пацієнта - усі ці фактори підвищують ймовірність помилкових рішень. СППР, яка базується на формалізованих правилах, не піддається суб'єктивним чинникам і дозволяє мінімізувати ризик пропуску важливого симптому або некоректної інтерпретації даних.

У ширшому сенсі СППР також виступають як навчальний інструмент для молодих лікарів. Вони дозволяють ознайомитися з формалізованими алгоритмами діагностики, відслідковувати логіку прийняття рішень і поступово підвищувати власний професійний рівень. У результаті СППР не лише допомагають у практичній діяльності, а й відіграють освітню роль, що має стратегічне значення для підготовки нових поколінь медичних фахівців [12].

1.3.4. Проблеми та обмеження СППР

Попри значні переваги, СППР мають низку суттєвих проблем, які стримують їхнє широке впровадження у медичну практику. Одним із ключових обмежень є складність формалізації медичних знань. Симптоми та клінічні ознаки часто описуються у вільній формі, залежать від суб'єктивного сприйняття пацієнта й лікаря та можуть суттєво варіюватися у різних

клінічних контекстах. Перетворення такого різноманіття у жорстко формалізовані правила типу «якщо – то» є складним завданням, що призводить до неповноти або надмірної спрощеності баз знань [13].

Ще однією проблемою є обмеженість бази правил. Навіть великі СППР не здатні охопити весь спектр клінічних ситуацій, зокрема рідкісні захворювання чи нетипові комбінації симптомів. Це означає, що у складних випадках система може видати некоректний або неповний діагноз, що обмежує її практичну цінність.

Важливою перепорою залишається й проблема оновлення знань. Медична наука постійно розвивається: з'являються нові протоколи, методи лікування, зміни у класифікаціях, наприклад, перехід від ICD-10 до ICD-11 (рис.1.5).

ICD-10	ICD-11
Personality disorders 60.0 Paranoid 60.1 Schizoid 60.2 Dissocial 60.3 Emotionally unstable 60.4 Histrionic 60.5 Anankastic 60.6 Anxious [avoidant] 60.7 Dependent 60.8 Other specific type 60.9 Unspecified type 61 Mixed and other personality disorders (73.1 Accentuated personality traits)	Personality disorder 10.Z Severity unspecified 10.0 Mild 10.1 Moderate 10.2 Severe (50.7 Personality difficulty) Trait domain and pattern specifiers 11.0 Negative Affectivity 11.1 Detachment 11.2 Dissociality 11.3 Disinhibition 11.4 Anankastia 11.5 Borderline pattern

Рисунок 1.5 – перехід від ICD-10 до ICD-11 на прикладі розладу особистості

Щоб СППР залишалася актуальною, її база знань повинна регулярно оновлюватися, проте цей процес є трудомістким і вимагає участі висококваліфікованих фахівців.

Не менш значущою є проблема інтерпретованості рішень. У багатьох випадках лікареві складно пояснити пацієнту, чому система дійшла саме

такого висновку. Якщо механізм виведення є непрозорим або занадто складним, це знижує довіру як із боку лікаря, так і пацієнта. Проблема «чорної скриньки» актуальна і для класичних rule-based систем, і для їхніх сучасних гібридних варіантів [14].

Окремо варто виділити фактор недовіри медичної спільноти. Навіть при високій точності діагнозів лікарі не завжди готові покладатися на автоматизовані рішення, побоюючись помилок і юридичної відповідальності. Відсутність чітких регуляторних механізмів та стандартів використання СППР у медицині лише посилює цю проблему.

Таким чином, попри значний потенціал, СППР не є універсальним рішенням і потребують подальшого вдосконалення. Саме ці обмеження стали поштовхом до пошуку нових підходів, зокрема інтеграції з нейронними мережами та методами машинного навчання, які здатні забезпечити кращу гнучкість, масштабованість і здатність до самонавчання.

Висновки до розділу 1

У першому розділі було здійснено ґрунтовний огляд проблем діагностики в неврології та аналіз особливостей СППР як перспективного інструменту підтримки прийняття клінічних рішень.

Було показано, що неврологічні захворювання становлять надзвичайно широкий спектр патологій центральної та периферичної нервової системи, які супроводжуються значною клінічною варіативністю. Високий рівень міжхворобної схожості симптомів, наявність коморбідних станів, а також обмежений доступ до сучасних методів діагностики ускладнюють процес своєчасного встановлення діагнозу. Особливої уваги заслуговує той факт, що класичні методи часто є трудомісткими та значною мірою залежать від досвіду лікаря, що створює ризик помилок і затримок у початку лікування.

У цьому контексті було розглянуто СППР, які поєднують знання лікарів і алгоритми штучного інтелекту для імітації процесу прийняття рішень. Було

визначено їхню сутність, структуру (база знань, механізм виведення, інтерфейс користувача) та відмінності від традиційного програмного забезпечення. Проаналізовано історичні приклади (MYCIN, INTERNIST-1) та сучасні клінічні системи підтримки прийняття рішень (CDSS), які підтвердили ефективність СППР у медицині, зокрема в діагностиці інфекційних, внутрішніх та неврологічних захворювань.

Особливу увагу приділено перевагам СППР: швидкій обробці великих обсягів клінічних даних, стандартизації підходів до діагностики, підвищенню доступності медичних знань у віддалених регіонах, зменшенню ризику людських помилок і навіть освітній функції для молодих лікарів.

Водночас було детально розглянуто проблеми та обмеження таких систем: складність формалізації медичних знань, обмеженість і потреба в регулярному оновленні баз знань, низька інтерпретованість результатів («ефект чорної скриньки»), а також фактор недовіри з боку медичної спільноти через відсутність чітких регуляторних стандартів.

Таким чином, у результаті аналізу підтверджено, що СППР є перспективним інструментом для неврології, здатним підвищити точність і швидкість діагностики, проте їх ефективність обмежується низкою технічних та організаційних факторів. Вирішення цих проблем можливе шляхом інтеграції СППР із методами глибинного навчання та сучасними моделями штучного інтелекту, що створює підґрунтя для подальшої роботи над удосконаленням систем підтримки клінічних рішень.

РОЗДІЛ 2

АНАЛІЗ АНАЛОГІВ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

У цьому розділі представлено аналіз сучасних СППР у медицині та технічних підходів, що стали основою для створення розробленої системи. Розглянуто наукові та комерційні рішення, зокрема IBM Watson for Health, DXplain і Isabel Healthcare, їхні архітектури, принципи роботи та інтеграцію з електронними медичними картками.

Основну увагу приділено методам класифікації: ANN, які виявляють нелінійні залежності в багатовимірних даних; CNN для аналізу МРТ і КТ-зображень; та RNN, ефективним у роботі з часовими даними, наприклад ЕЕГ. Також розглянуто гібридні системи, що поєднують експертні правила з машинним навчанням, а також роль хмарних рішень і федеративного навчання у забезпеченні конфіденційності даних. На завершення виконано порівняльний аналіз сильних і слабких сторін існуючих аналогів, що дало змогу обґрунтувати вибір підходу для створення власної системи, яка поєднує глибинне навчання з інтуїтивно зрозумілим інтерфейсом.

2.1. Огляд існуючих комерційних і наукових рішень

СППР в медицині є ключовим інструментом для підтримки процесу діагностики та прийняття клінічних рішень, дозволяючи стандартизувати підходи до лікування та зменшувати ймовірність медичних помилок. Вони інтегруються з електронними медичними картками та дозволяють систематизувати дані про пацієнта, надаючи лікарям рекомендації на основі великого масиву медичної інформації. У сучасній медицині значна увага приділяється не лише комерційним системам, але й науковим розробкам, що застосовують новітні методи штучного інтелекту та глибокого навчання для покращення точності діагностики. Такі системи здатні обробляти складні дані, включаючи історії хвороби, результати лабораторних та інструментальних

досліджень, і на основі цього генерувати клінічні висновки, що підвищує ефективність роботи лікарів і якість медичної допомоги.

Сучасний ринок медичних СППР представлений низкою комерційних продуктів, які широко застосовуються в клінічній практиці. Одним із найвідоміших рішень є система IBM Watson for Health (рис. 2.1), яка інтегрує алгоритми машинного та глибокого навчання для аналізу медичних даних, пошуку відповідних наукових публікацій та підтримки прийняття рішень лікарем. Вона здатна обробляти великі обсяги неструктурованої інформації - від історій хвороб до результатів досліджень - і формувати персоналізовані рекомендації для кожного пацієнта. Watson також використовує технології обробки природної мови, що дозволяє йому інтерпретувати клінічні тексти та зіставляти їх із базою знань медичних статей і протоколів лікування. Завдяки цьому система допомагає лікарям приймати обґрунтовані рішення на основі доказової медицини та скорочує час, необхідний для встановлення діагнозу або вибору терапевтичної стратегії.

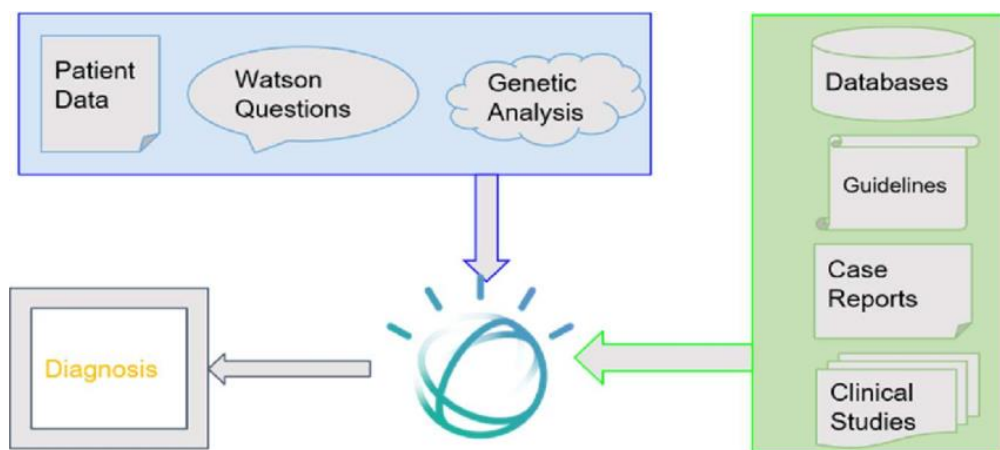


Рисунок 2.1 – Архітектура системи IBM Watson for Health [15]

Система дозволяє обробляти великі об'єми клінічної інформації та надає персоналізовані рекомендації для діагностики та лікування пацієнтів [15].

Окрім великих систем, на ринку представлені менш масштабні рішення, зокрема DXplain (рис. 2.2), розроблена в Гарвардському університеті. Вона

допомагає лікарям у постановці діагнозу, пропонуючи перелік можливих захворювань на основі введених симптомів та клінічних даних.

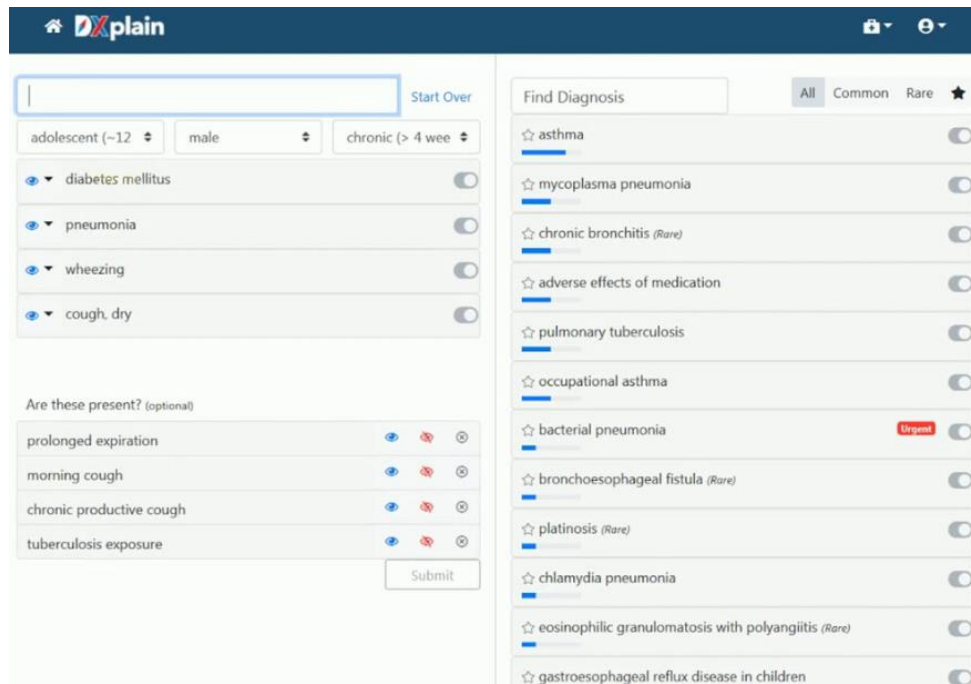


Рисунок 2.2 – Інтерактивний інтерфейс системи DXplain [16]

Інша популярна система Isabel Healthcare (рис. 2.3) - орієнтована на клінічну підтримку лікарів загальної практики. Вона має зручний вебінтерфейс, що дозволяє швидко вводити симптоми пацієнта та отримувати список імовірних діагнозів із посиланнями на клінічні рекомендації та джерела доказової медицини. На головній сторінці користувач бачить поле для введення симптомів, де можна вводити їх у вільній формі - як окремі слова або фрази, після чого система автоматично розпізнає терміни та пропонує варіанти з медичної бази даних. Після введення симптомів лікар отримує список імовірних діагнозів, відсортованих за рівнем відповідності введеним даним. Кожен діагноз супроводжується коротким описом, клінічними критеріями, можливими тестами для підтвердження та посиланнями на джерела доказової медицини, такі як BMJ Best Practice, PubMed [16].

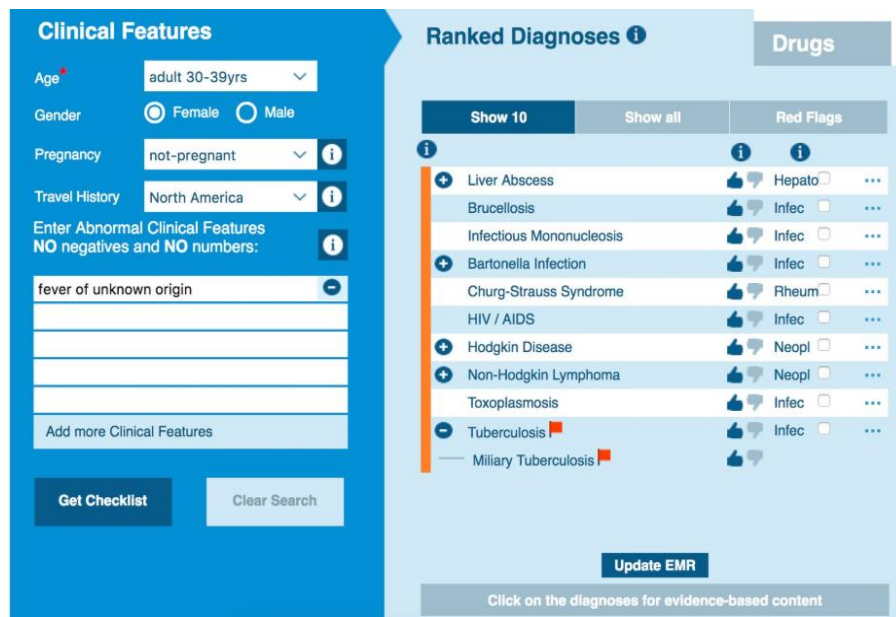


Рисунок 2.3 – Інтерактивний інтерфейс системи Isabel Healthcare [16]

Інтерфейс також підтримує фільтри за віком, статтю та регіоном, що дозволяє адаптувати результати під конкретного пацієнта. У додаткових вкладках користувач може переглянути клінічні шляхи, алгоритми диференційної діагностики та таблиці порівняння захворювань. Система має адаптивний дизайн, тому зручно працює як на комп'ютері, так і на мобільних пристроях. Завдяки цьому Isabel Healthcare використовується не лише у стаціонарах, а й у приватних клініках і на первинному рівні медичної допомоги, забезпечуючи лікарям оперативну підтримку прийняття рішень у реальному часі.

Наукові розробки в цій сфері також мають велике значення, оскільки дозволяють оцінювати ефективність алгоритмів у реальних клінічних умовах та порівнювати їх із комерційними аналогами. Наприклад, дослідження університетських лабораторій з ШІ показують, що застосування нейромереж і систем на основі глибокого навчання значно підвищує точність класифікації неврологічних розладів у порівнянні з традиційними експертними правилами [17]. Наукові рішення часто є відкритими або напіввідкритими, що дозволяє модифікувати алгоритми під специфічні потреби медичних закладів і дослідницьких груп, підвищуючи гнучкість та адаптивність системи.

Важливим аспектом при використанні комерційних і наукових систем є інтеграція з існуючими стандартами медичної інформації, такими як ICD-11, HL7 та інші. Це забезпечує сумісність з електронними медичними картками, стандартизовану обробку даних та зменшення ризику помилок при введенні інформації. Крім того, комерційні системи зазвичай мають розвинений інтерфейс користувача та підтримку клієнтів, що спрощує їх впровадження у лікарських установах і підвищує ефективність їх використання в повсякденній практиці.

2.2. Технічні підходи в аналогах (архітектури, дані)

Технічна основа сучасних аналогів СППР у медицині базується на поєднанні різних архітектурних рішень і методів роботи з даними. Їх розвиток обумовлений потребою не лише автоматизувати процес діагностики, але й зробити його надійним, прозорим та безпечним для клінічного використання. У цьому контексті варто розглянути ключові підходи, які застосовуються в комерційних та наукових розробках.

Історично перші медичні СППР будувались за правилом «якщо – то», де база знань містила набір логічних висловлювань, сформульованих експертами. Прикладом є система MYCIN, яка використовувалася для діагностики бактеріальних інфекцій [18]. Такі рішення забезпечували певний рівень прозорості, однак виявилися недостатньо гнучкими при роботі з великим обсягом клінічних даних.

З розвитком обчислювальних можливостей дедалі більшого поширення набули архітектури, засновані на методах машинного навчання, які здатні обробляти великі обсяги складних клінічних даних. Одним із найчастіше використовуваних підходів є штучні нейронні мережі (рис. 2.4). Вони добре працюють із багатовимірними наборами даних, що поєднують числові показники, текстові записи та клінічні симптоми. На відміну від класичних rule-based систем, ANN мають здатність виявляти приховані нелінійні

залежності між параметрами, що підвищує точність прогнозу навіть у випадках, коли клінічна картина є розмитою або містить суперечливі дані. Це робить їх особливо корисними у неврології, де один і той самий симптом може бути проявом різних захворювань, а точність розмежування має критичне значення.

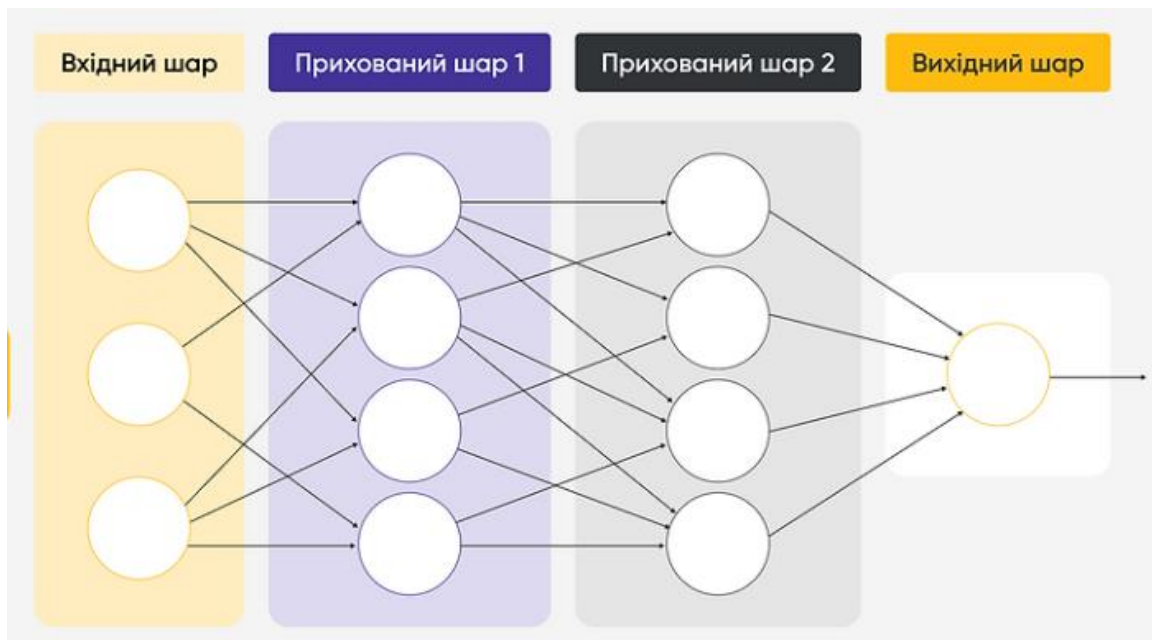


Рисунок 2.4 – Базова архітектура ANN

Окремо варто виділити конволюційні нейронні мережі (рис. 2.5), які успішно застосовуються для аналізу медичних зображень, зокрема магнітно-резонансної томографії (МРТ) та комп'ютерної томографії (КТ). [19]

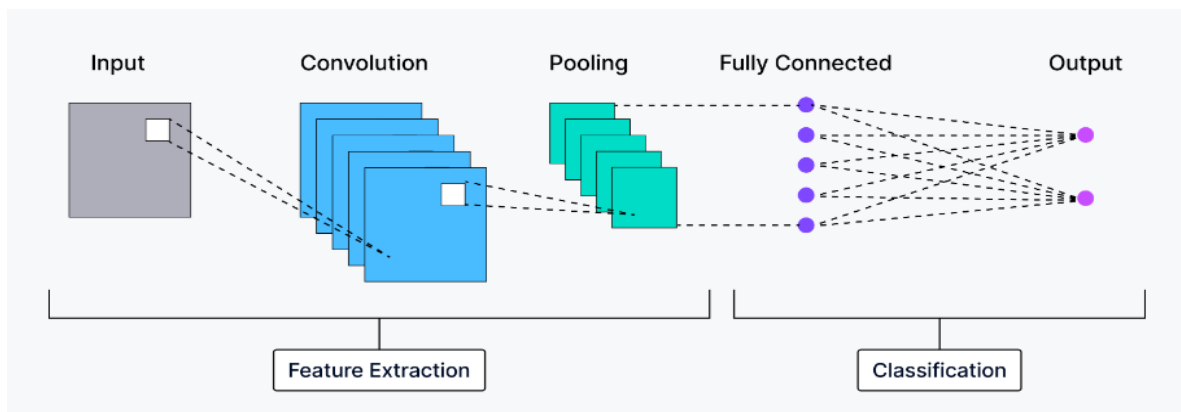


Рисунок 2.5 – Базова архітектура CNN

CNN здатні автоматично виділяти важливі ознаки на зображеннях - наприклад, ділянки ішемії, пухлини, крововиливи або дегенеративні зміни тканин. Це дозволяє не лише пришвидшити процес діагностики, а й підвищити її об'єктивність, зменшуючи залежність від суб'єктивного досвіду лікаря. Такі системи показують точність, співставну з результатами висококваліфікованих спеціалістів, і вже активно інтегруються у клінічні процеси, наприклад у програмах для скринінгу інсультів чи виявлення ранніх стадій хвороби Альцгеймера [19].

Окрім цього, згорткові нейронні мережі можуть не лише розпізнавати патологічні структури, а й оцінювати динаміку змін у пацієнта, що робить їх корисними для моніторингу ефективності лікування. У поєднанні з технологіями тривимірної реконструкції зображень CNN здатні аналізувати складні анатомічні структури головного мозку, виявляючи навіть мінімальні відхилення від норми. У перспективі такі системи можуть бути інтегровані у лікарняні інформаційні комплекси для автоматичного аналізу медичних знімків у режимі реального часу, що значно зменшує навантаження на лікарів-радіологів і підвищує доступність якісної діагностики навіть у віддалених медичних закладах.

Ще одним важливим напрямом є застосування RNN (рис. 2.6) та їхніх сучасних модифікацій (наприклад, LSTM та GRU), які ефективно працюють із послідовними даними.

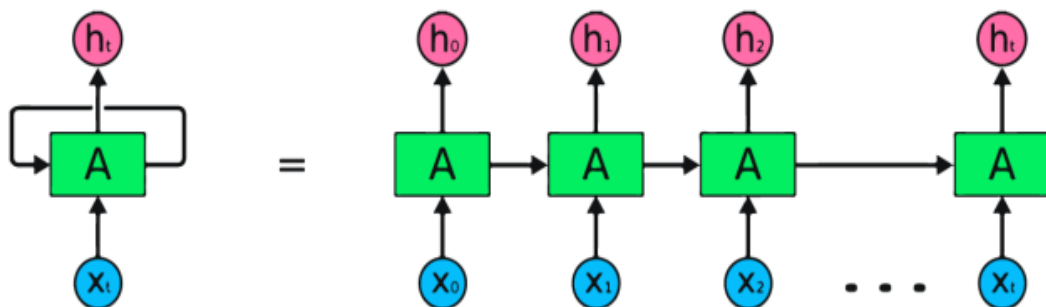


Рисунок 2.6 – Рекурентна нейронна мережа

У медицині вони застосовуються для аналізу часових рядів - наприклад, для прогнозування нападів епілепсії на основі ЕЕГ, відстеження змін неврологічних показників у динаміці або моніторингу пацієнтів із хронічними захворюваннями. Таким чином, різні архітектури нейронних мереж забезпечують комплексне покриття різних типів даних: ANN для багатовимірних структурованих вибірок, CNN для зображень та RNN для часових рядів, що створює передумови для побудови мультимодальних СППР.

Паралельно розвиваються гібридні архітектури, які поєднують класичні експертні правила з алгоритмами машинного навчання. Такий підхід дозволяє значно підвищити точність системи, зберігаючи при цьому інтерпретованість результатів, що особливо важливо у клінічній практиці, де лікар повинен розуміти логіку рекомендацій. Наприклад, у системах для підтримки діагностики інсульту використовуються правила, сформовані на основі офіційних клінічних протоколів, а для уточнення діагнозу підключаються моделі глибокого навчання, здатні аналізувати зображення КТ та МРТ і виявляти патологічні зміни на ранніх стадіях [20]. Подібні гібридні рішення поєднують переваги класичних rule-based систем та сучасних нейронних мереж, забезпечуючи баланс між прозорістю, точністю і масштабованістю.

Важливим напрямом розвитку є також хмарні рішення та моделі з федеративним навчанням, які забезпечують можливість тренування систем на розподілених медичних даних без необхідності їх централізованого зберігання. Це дозволяє використовувати інформацію з різних клінік та лікарень, не порушуючи вимог до конфіденційності, оскільки дані залишаються на локальних серверах. У медицині, де безпека персональних даних пацієнтів є критично важливою, подібний підхід значно зменшує ризики витоку інформації та водночас сприяє формуванню більш точних і універсальних моделей [21]. Крім того, хмарні інфраструктури забезпечують масштабованість і доступність СППР навіть для закладів із обмеженими локальними ресурсами, що відкриває перспективи для глобальної інтеграції таких рішень у клінічну практику.

Незважаючи на високу ефективність згорткових (CNN) та рекурентних (RNN) мереж у сферах обробки зображень та часових послідовностей, вони демонструють суттєві обмеження при роботі з контекстуально насиченими, неструктурованими текстовими даними, які становлять значну частину електронних медичних карток (EHR). Проблема полягає в тому, що CNN та ANN, як правило, найкраще працюють зі структурованими даними або зображеннями, тоді як клінічні записи, анамнез та нотатки лікарів є переважно неструктурованим текстом. Останнім значним проривом у розвитку медичних ШІ-систем, який кардинально вирішив цю проблему, стало використання архітектури Трансформер (Transformer), представлена у 2017 році [22].

Ключовим нововведенням Трансформера є механізм самоуваги (Self-Attention). На відміну від RNN, які обробляють текст послідовно (що часто призводить до втрати контексту у довгих реченнях), механізм уваги дозволяє моделі одночасно зважувати важливість кожного слова у реченні відносно всіх інших слів.

Це дозволяє захоплювати складні, довгострокові залежності у тексті, що є критично важливим для аналізу клінічних звітів, де діагноз або ключова скарга може бути відокремлена від інших релевантних симптомів великою кількістю проміжного тексту.

На основі архітектури Трансформер (рис. 2.7) були розроблені великі мовні моделі (LLM), такі як BERT (Bidirectional Encoder Representations from Transformers) [23] та його наступники – GPT (Generative Pre-trained Transformer). Ці моделі навчаються на мільярдах слів з медичної літератури, наукових статей та анонімізованих EHR, створюючи глибоке лінгвістичне розуміння клінічної мови.

Зокрема, були створені спеціалізовані медичні LLM, як-от BioBERT чи Med-PaLM [24], які доводять свою здатність ефективно аналізувати неструктуровані текстові записи (анамнез, скарги, висновки лікарів, результати обстежень) та витягувати з них діагностично важливу інформацію.

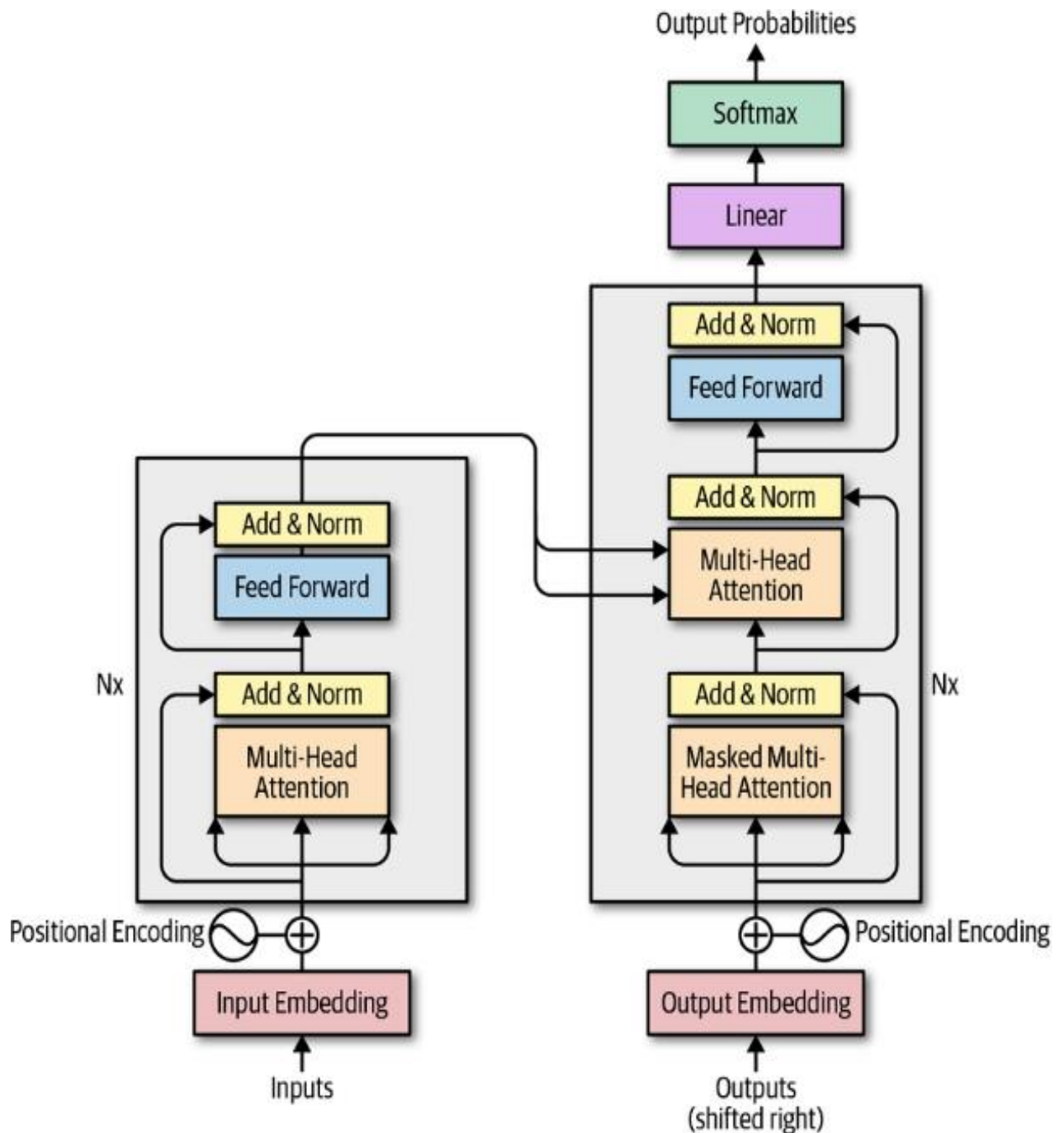


Рисунок 2.7 – Базова архітектура кодувальника Трансформер

Таким чином, LLM непрямо конкурують із класифікаційними ANN, оскільки можуть автоматизувати кодування та вилучення ознак із сирих текстових даних, що є наступним кроком у мінімізації ручної праці фахівця.

Будь-яка СППР у медицині є настільки ефективною, наскільки якісні дані вона обробляє. Основними джерелами інформації виступають:

1. електронні медичні картки (EHR), що містять структуровані записи про пацієнтів;
2. медичні зображення (МРТ, КТ, ПЕТ), які вимагають складної попередньої обробки;
3. лабораторні дослідження та геномні дані, що дають можливість для персоналізованої медицини;
4. клінічні протоколи та настанови (NICE, МОЗ, NHS, NIH), які забезпечують базу для формалізації правил.

Окреме місце займають відкриті наукові бази даних, наприклад ADNI (Alzheimer's Disease Neuroimaging Initiative) або UK Biobank, які широко застосовуються для тренування та тестування моделей (рис. 2.8).

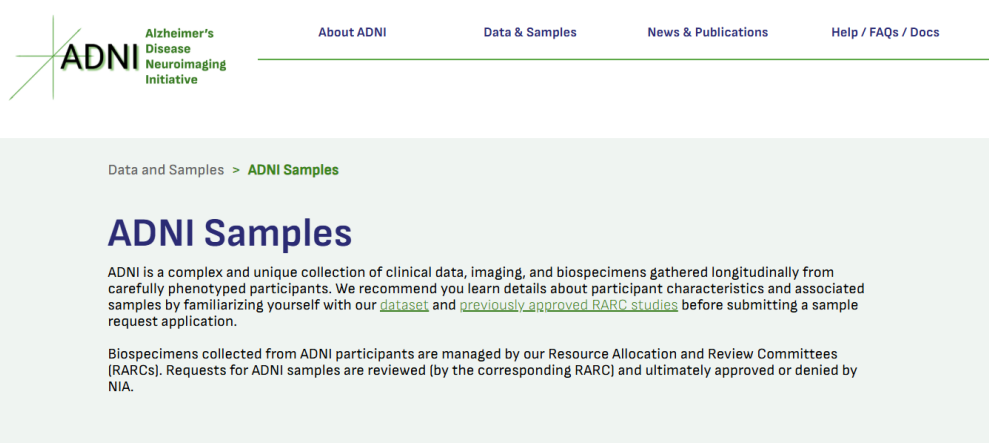


Рисунок 2.8 – Головна сторінка репозиторію ADNI

Разом із тим, збирання таких даних пов'язане з низкою викликів, серед яких - неповнота записів, дублювання інформації, різні формати зберігання та обмежена якість даних. У багатьох випадках клінічні записи створюються вручну, що підвищує ризик помилок або пропусків, а різні медичні заклади використовують власні інформаційні системи, несумісні між собою. Це суттєво ускладнює інтеграцію даних у масштабах регіону чи країни та обмежує можливості навчання універсальних моделей. Саме тому значна

увага приділяється стандартизації медичної інформації, яка дозволяє створювати уніфіковані формати для збирання, обміну та обробки даних.

Серед ключових інструментів такої стандартизації - ICD-11, яка забезпечує єдину систему класифікації захворювань, та HL7 FHIR (Fast Healthcare Interoperability Resources), що використовується для уніфікації медичних записів і спрощення обміну інформацією між різними клініками та системами електронної медицини. Використання цих стандартів підвищує якість даних, знижує ризик дублювання і створює передумови для розвитку сумісних СППР, здатних працювати у глобальному медичному просторі [25].

Щоб системи могли працювати ефективно, дані потребують ретельної попередньої обробки. У практиці аналогів застосовуються такі методи:

1. очищення від шуму та артефактів (наприклад, нормалізація зображень МРТ перед подачею в нейромережу);
2. балансування класів у вибірках для уникнення переваги «здорових» даних над патологічними;
3. кодування даних (зокрема, one-hot encoding для симптомів чи діагнозів);
4. інтеграція мультимодальних даних, що дозволяє об'єднати текстові записи, числові показники та зображення [26].

У сучасних аналогах широко використовуються також методи обробки природної мови (NLP) для аналізу лікарських висновків та історій хвороб. Це дозволяє формувати структуровану базу знань навіть із неформалізованих текстів.

Попри активний розвиток технічних підходів, залишається низка викликів. Насамперед це неоднорідність даних: навіть у межах однієї клініки формати записів можуть суттєво відрізнятися, що ускладнює створення універсальних моделей. Хоча стандарти ICD-11 чи HL7 FHIR частково вирішують цю проблему, їх упровадження на практиці є неповним і потребує значних ресурсів.

Іншою проблемою виступає інтерпретованість результатів. Глибокі нейронні мережі демонструють високу точність, але працюють як «чорні скриньки». Лікарям складно пояснити пацієнтам, чому система ухвалила певне рішення. Для подолання цієї проблеми активно розвивається напрям explainable AI (ХАІ), який передбачає використання методів візуалізації ознак або генерацію зрозумілих для лікаря пояснень [27].

Не менш важливим викликом є безпека та конфіденційність даних. Сучасні системи повинні відповідати суворим вимогам до захисту персональної інформації, що вимагає застосування технологій анонімізації, шифрування та федеративного навчання. Проте вартість таких рішень і потреба у спеціалізованій інфраструктурі залишаються суттєвим бар'єром для їх широкого впровадження [28].

2.3. Порівняльний аналіз: сильні та слабкі сторони аналогів

Порівняння існуючих аналогів СППР у медицині дає можливість визначити не лише їхні переваги, але й ті обмеження, що стримують широке впровадження таких технологій у клінічну практику. Аналіз наукових і комерційних розробок свідчить про різноманіття підходів до архітектури, роботи з даними та інтеграції з медичними процесами.

Однією з найважливіших сильних сторін сучасних аналогів є здатність до обробки великих масивів даних. Завдяки використанню методів машинного та глибокого навчання системи демонструють високу точність у задачах розпізнавання образів, таких як аналіз МРТ та КТ. Наприклад, системи на базі конволюційних нейронних мереж можуть виявляти патології головного мозку з точністю, порівнянною або навіть вищою за лікарів-радіологів [29]. Це дозволяє значно підвищити швидкість діагностики та зменшити навантаження на спеціалістів.

Водночас ці ж рішення характеризуються суттєвими обмеженнями. Найбільшою проблемою залишається їх низька інтерпретованість. Лікарі

часто не можуть пояснити пацієнтам, на основі яких факторів система ухвалила рішення. Це створює недовіру та знижує готовність медичних установ використовувати такі технології. Для вирішення проблеми активно досліджуються підходи Explainable AI, але вони ще не стали стандартом [30].

Іншою сильною стороною є можливість персоналізованої медицини. Деякі системи аналізують не лише симптоми чи зображення, а й геномні дані, що дозволяє підібрати індивідуальне лікування для пацієнтів із неврологічними розладами. Це відкриває перспективи для глибшого розуміння патогенезу захворювань і впровадження точнішої терапії. Проте такі системи залежать від наявності великих високоякісних біомедичних баз даних, що є проблематичним для більшості країн [31].

З практичної точки зору, значна перевага полягає у здатності до інтеграції з електронними медичними картками (EHR). Комерційні продукти, зокрема IBM Watson Health, вже мають механізми автоматизованої обробки клінічних записів і рекомендацій на основі протоколів. Це суттєво полегшує роботу лікаря, знижує ризик помилок і допомагає приймати обґрунтовані рішення. Водночас проблема стандартизації даних залишається актуальною: різні лікарні використовують власні формати зберігання інформації, що ускладнює масштабування системи [32].

Окремим аспектом є економічна ефективність. Дослідження показують, що застосування СППР дозволяє скоротити витрати на діагностику за рахунок зменшення кількості непотрібних процедур та швидшого встановлення діагнозу. Це особливо важливо для систем охорони здоров'я з обмеженими ресурсами. Проте висока вартість впровадження, необхідність у потужних обчислювальних ресурсах і спеціалізованих фахівцях залишаються значними бар'єрами для широкого використання [33].

З точки зору клінічної практики, сильними сторонами є: висока точність в окремих діагностичних задачах, підтримка персоналізованого підходу та інтеграція з медичними даними. Слабкими - низька інтерпретованість,

проблеми зі стандартизацією і висока вартість упровадження. Нижче в табл. 2.2 виділено сильні та слабкі сторони аналогів.

Таблиця 2.2 – Сильні та слабкі сторони аналогів

Сильні сторони	Слабкі сторони
Висока точність у задачах діагностики	Низька інтерпретованість результатів (black-box)
Швидкість обробки	Проблеми зі стандартизацією даних
Можливість персоналізованої медицини	Висока вартість впровадження
Інтеграція з електронними медичними картками (EHR)	Залежність від великих і якісних біомедичних баз
Економічна ефективність (зменшення витрат на діагностику)	Необхідність потужних обчислювальних ресурсів та фахівців

2.4. Обґрунтування вибору підходу для власної системи

Аналіз існуючих комерційних і наукових рішень показав, що більшість систем у сфері медичної діагностики демонструють високий рівень спеціалізації, але водночас мають обмеження у масштабованості, інтеграції з клінічними стандартами та зручності використання. Це дозволяє сформулювати ключові вимоги до власної СППР, які повинні враховувати як сильні сторони аналогів, так і усувати виявлені недоліки.

Функціональні вимоги передбачають підтримку роботи з широким спектром неврологічних симптомів і захворювань, використання сучасних алгоритмів машинного навчання для підвищення точності діагностики, а також надання рекомендацій відповідно до міжнародних стандартів, зокрема ICD-11, ICH, NICE. Важливим є забезпечення прозорості роботи системи, тобто можливості пояснити, на основі яких симптомів і правил було сформовано діагностичне припущення.

Технічні вимоги зумовлені необхідністю забезпечення швидкої обробки даних і масштабованості рішення. Система повинна бути побудована на

модульній архітектурі, що дозволить оновлювати алгоритми або розширювати базу знань без втрати сумісності. Не менш важливими є захист персональних медичних даних, відповідність нормам GDPR та МОЗ України, а також використання структурованих форматів даних (наприклад, one-hot encoding) для оптимізації процесів навчання моделі.

Користувацькі вимоги стосуються простоти та інтуїтивності інтерфейсу. Клініцисти повинні отримувати швидкий доступ до ключових функцій, а результати діагностики мають відображатися у зрозумілому форматі, без перевантаження технічними деталями. Для пацієнтів система може містити спрощений режим, що забезпечить базові рекомендації та поради щодо подальших дій.

Таким чином, сформовані вимоги (табл. 2.3) відображають баланс між високим рівнем технічної складності та практичною зручністю використання, створюючи підґрунтя для побудови конкурентоспроможної СППР у сфері неврологічної діагностики.

Таблиця 2.3 – Вимоги системи

Функціональні вимоги	Технічні вимоги	Користувацькі вимоги
Підтримка широкого спектру неврологічних симптомів і захворювань	Модульна архітектура для легкого оновлення й розширення	Інтуїтивний і простий інтерфейс
Використання алгоритмів машинного навчання	Висока швидкість обробки та масштабованість системи	Швидкий доступ до основних функцій
Надання рекомендацій відповідно до ICD-11, NICE, NH	Використання структурованих форматів даних (one-hot encoding)	Зрозуміле відображення результатів
Прозорість і пояснюваність прийняття рішень	Захист персональних даних, відповідність GDPR та МОЗ України	Наявність окремого режиму для клініцистів і спрощеного для пацієнтів

Висновки до розділу 2

У другому розділі було проведено системний аналіз сучасних аналогів СППР у медицині з метою визначення їхніх технічних особливостей, переваг і недоліків, а також можливостей для подальшого розвитку. Порівняння існуючих рішень показало, що сучасні СППР значно відрізняються від своїх попередників завдяки застосуванню методів машинного та глибинного навчання. Це дозволило їм досягти високої точності у таких задачах, як розпізнавання зображень, класифікація симптомів і персоналізована медицина. Разом із тим, аналіз виявив і низку слабких сторін існуючих рішень. Серед них - низька інтерпретованість результатів, труднощі стандартизації даних між різними медичними установами, залежність від великих і якісних датасетів, а також висока вартість впровадження та необхідність у спеціалізованих обчислювальних ресурсах.

Окремо було проаналізовано проблеми даних. Показано, що клінічні записи часто є неповними, дублюються або зберігаються у різних форматах, що створює труднощі для їх використання в машинному навчанні. Тому особлива увага приділяється стандартизації - зокрема впровадженню міжнародних стандартів ICD-11 для класифікації захворювань та HL7 FHIR для уніфікації медичних записів. Їхнє використання дозволяє створювати єдині протоколи даних, що сприяє сумісності систем і підвищує якість прогнозів.

Загалом результати аналізу довели, що СППР в медицині є потужним інструментом, здатним підвищувати ефективність діагностики та лікування, однак їх подальший розвиток потребує вирішення проблем інтерпретованості, стандартизації й доступності даних. Водночас інтеграція класичних експертних підходів із методами глибинного навчання та застосування хмарних технологій створюють підґрунтя для побудови нових поколінь систем, більш адаптивних, масштабованих і безпечних для використання у реальних клінічних умовах.

РОЗДІЛ 3

ЗАСОБИ РЕАЛІЗАЦІЇ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

У розділі "Засоби реалізації СППР" буде детально описано технологічну основу розробки. Розділ почнеться з обговорення джерел даних та правил, що лягли в основу системи. Буде показано, як інформація з авторитетних клінічних настанов, таких як NICE, NHS, NIH та МОЗ України, була використана для формування достовірної бази знань. Далі буде представлено навчальний набір даних, який був отриманий з платформи Kaggle та містив 773 унікальні захворювання і 377 симптомів. Буде описано, як цей датасет був відфільтрований до 66 неврологічних діагнозів, що є основою для спеціалізації системи. Нарешті, буде розглянуто технічне середовище: мова програмування Python 3.11 та використані бібліотеки, зокрема NumPy, pandas, TensorFlow/Keras для побудови моделі та Tkinter для графічного інтерфейсу. Буде пояснено, як ці інструменти взаємодіють, забезпечуючи повноцінний цикл роботи системи від даних до готового інтерфейсу.

У останньому підрозділі розглянуто методи забезпечення безпеки та захисту інформації в СППР, призначеній для обробки медичних даних. Особливу увагу приділено питанням конфіденційності, збереженню анонімності пацієнтів, захисту даних під час зберігання й обробки, а також дотриманню міжнародних стандартів, таких як GDPR. Крім того, описано технічні заходи, що запобігають несанкціонованому доступу до інформації, і принципи *privacy by design*, які лежать в основі архітектури системи.

3.1. Джерела даних та правил СППР

Дані для побудови СППР базуються на авторитетних джерелах клінічних рекомендацій та настанов у сфері неврології. Одним із ключових джерел є NICE - провідна організація, що розробляє та оновлює клінічні настанови з діагностики, лікування та менеджменту широкого спектра

захворювань. Рекомендації NICE формуються на основі систематичних оглядів, метааналізів та консенсусів експертів, що гарантує їхню доказову силу й відповідність сучасним стандартам медицини. Особливу цінність для побудови СППР мають стандартизовані описи симптомів, діагностичних критеріїв і протоколів лікування, які можна безпосередньо використовувати для структуризації даних і створення правил класифікації. Рекомендації NICE (рис. 3.1) широко застосовуються клініцистами не лише у Великобританії, а й на міжнародному рівні, оскільки вони є відкритими для доступу, мають зрозумілу структуру й дозволяють адаптувати їх для різних медичних інформаційних систем [34].

The screenshot shows the NICE website interface. At the top, there is a search bar and a 'Sign in' button. Below the navigation menu, the page title is 'Published: Guidance, quality standards and advice'. There are tabs for 'Published', 'In consultation', 'In development', 'Awaiting development', and 'Topic selection'. A filter section on the left allows searching by keyword or reference number and filtering by last updated date. The main content is a table of published articles.

Title	Reference number	Published	Last updated
HIV testing: increasing uptake among people who may have undiagnosed HIV	NG60	1 December 2016	26 September 2025
Isatuximab in combination for untreated multiple myeloma when a stem cell transplant is unsuitable	TA1098	24 September 2025	24 September 2025
Atopic eczema in under 12s: diagnosis and management	CG57	12 December 2007	22 September 2025
Digital technologies to deliver pulmonary rehabilitation programmes for adults with COPD: early value assessment	HTE18	30 April 2024	17 September 2025

Рисунок 3.1 - Опубліковані статії стандартів лікування хвороб NICE

Ще одним важливим джерелом є NHS - офіційна державна служба охорони здоров'я, що виступає найбільшою інтегрованою медичною системою у світі. NHS забезпечує відкритий доступ до перевіреної інформації про симптоми, діагностику, лікування та профілактику захворювань, яка регулярно оновлюється відповідно до сучасних клінічних протоколів. Для

неврологічних хвороб цей ресурс містить структуровані описи найпоширеніших симптомів, пояснення патофізіологічних механізмів, алгоритми диференціальної діагностики та стандарти догляду за пацієнтами. Інформація надається у зрозумілій для клініцистів та пацієнтів формі (рис. 3.2), що робить її придатною не лише для практичного застосування у лікувальних закладах, але й для інтеграції у цифрові медичні продукти та навчальні набори даних. Саме завдяки високій якості й відкритості даних NHS є надійним джерелом для підготовки вибірок, що використовуються у навчанні моделей штучного інтелекту, включно з СППР в неврології [2].



Рисунок 3.2 - Головна сторінка каталогу протоколів лікування NHS

Для локального контексту були використані документи МОЗ України (рис. 3.3), зокрема клінічні настанови щодо неврологічних захворювань. Ці матеріали забезпечують офіційне підтвердження класифікації хвороб, включених до переліку для тренування моделі, та відповідають національним стандартам охорони здоров'я [35]. Вимоги МОЗ України до протоколів лікування та діагностики є ключовим елементом валідації СППР, оскільки

гарантують, що рекомендації, надані системою, є не лише технічно коректними, але й юридично та клінічно прийнятними в українській медичній практиці. Це дозволяє уникнути конфлікту між автоматизованим діагнозом та принципами доказової медицини, які регулюються державними органами. Крім того, ці офіційні документи слугують основою для розробки інтерфейсу користувача та відображення результатів, забезпечуючи лікарям зрозумілу термінологію та стандартизовані формати звітів. Використання таких джерел мінімізує ризик відхилення системи від чинних клінічних протоколів і сприяє її швидкій інтеграції в існуючі медичні інформаційні системи.



THE FINNISH MEDICAL SOCIETY
DUODECIM



МІНІСТЕРСТВО
ОХОРОНИ
ЗДОРОВ'Я
УКРАЇНИ

Головна

Протоколи

Документ
Пошук

Neurology\Неврологія
128

- 📄 >> [001.023 ebm00028 Лістеріоз](#)
- 📄 >> [001.024 ebm00029 Пращець](#)
- 📄 >> [001.029 ebm00031 Лайм-бореліоз \(ЛБ\)](#)
- 📄 >> [001.046 ebm00038 Підозра на сказ](#)
- 📄 >> [003.002 ebm00787 Втома](#)
- 📄 >> [003.003 ebm00931 Хронічний біль](#)
- 📄 >> [003.006 ebm00172 Нудота та блювання](#)
- 📄 >> [003.015 ebm00745 Запаморочення](#)
- 📄 >> [003.016 ebm00743 Головний біль](#)
- 📄 >> [003.017 ebm01024 Міокімія повіки](#)
- 📄 >> [003.028 ebm01054 Гикавка](#)
- 📄 >> [004.030 ebm00056 Симптоми аритмії та обстеження пацієнта з аритмією](#)

Рисунок 3.3 – Опубліковані неврологічні настанови МОЗ

Додатково для доповнення рекомендацій та перевірки міжнародного підходу використовувалися матеріали НІН. Це одна з найбільших у світі державних дослідницьких організацій у сфері медицини, яка акумулює результати багаторічних клінічних і фундаментальних досліджень. Бази даних НІН охоплюють широкий спектр патологій (рис. 3.4), включаючи неврологічні розлади, та містять структуровані описи симптоматики, факторів ризику, сучасних методів лікування й профілактики. Особливо цінним є те, що інформація надається у відкритому доступі, регулярно оновлюється та підтримується провідними клінічними центрами США. Завдяки цьому використання матеріалів НІН дозволяє інтегрувати глобальний досвід у побудову СППР, забезпечуючи універсальність її застосування у міжнародному медичному контексті та створюючи основу для порівняння підходів різних країн до ведення неврологічних пацієнтів.

The screenshot shows the NIH NINDS website interface. At the top, there are logos for NIH (National Institute of Neurological Disorders and Stroke) and NINDS (75th Anniversary). A search bar on the right contains the text 'Trastornos neurológicos en español' and 'Search NINDS'. Below the search bar is a navigation menu with options: Health Information, Funding, Current Research, News & Events, Publications, and About NINDS. The main content area is titled 'Glossary of Neurological Terms' and features an alphabetical index (A-Z). The page includes a sidebar with 'Disorders' and 'Glossary of Neurological Terms' selected. A 'Did you find the content you were looking for?' box with thumbs up/down icons is visible. The main text defines 'Agnosia' as the inability to recognize objects or persons and 'Allodynia' as pain from light touch.

Рисунок 3.3 - Глоссарій неврологічних термінологій НІН

Разом усі перелічені джерела - NICE, NHS, МОЗ України та НІН - формують комплексний і достовірний фундамент для підготовки навчальних даних та формування правил СППР. Поєднання національних і міжнародних рекомендацій дозволяє досягти оптимального балансу між локальною адаптацією та глобальною стандартизацією клінічної інформації.

Використання таких авторитетних джерел підвищує довіру до результатів моделі як серед лікарів, так і серед наукової спільноти. Це гарантує, що система ґрунтується на доказовій медицині, а її прогнози та рекомендації відповідають сучасним науково обґрунтованим практикам, що є принципово важливим для успішної інтеграції в клінічне середовище.

3.2. Джерела тренувальних даних

Для тренування СППР було використано відкритий датасет із платформи Kaggle, яка є одним із найбільш популярних і авторитетних ресурсів для наукових та прикладних досліджень у сфері аналізу даних і машинного навчання (рис.3.4.) [36]. Вибір саме цього ресурсу зумовлений його широким визнанням серед дослідників та зручністю інтеграції у проекти, що потребують великих обсягів структурованої інформації.

△ diseases List of diseases.	✓ anxiety and nerv... symptom / condition	✓ depression symptom / condition	✓ shortness of brea... symptom / condition	✓ depressive or ps... symptom / condition	✓ sharp chest pain symptom / condition	✓ dizziness symptom / condition
773 unique values	246945 total values	246945 total values	246945 total values	246945 total values	246945 total values	246945 total values
panic disorder	1	0	1	1	0	0
panic disorder	0	0	1	1	0	1
panic disorder	1	1	1	1	0	1
panic disorder	1	0	0	1	0	1
panic disorder	1	1	0	0	0	0
panic disorder	0	0	1	1	0	0
panic disorder	1	0	0	0	0	0
panic disorder	0	0	0	1	0	0
panic disorder	1	0	0	1	0	1
panic disorder	1	1	1	0	0	0
panic disorder	0	0	1	0	0	0
panic disorder	1	0	0	1	0	1

Рисунок 3.4 – Частковий вигляд датасету

Використаний датасет містить відомості про 773 унікальні захворювання та 377 симптомів, представлених у форматі one-hot векторів. Такий підхід до структурування даних дозволяє напряму використовувати їх у нейронних мережах без додаткового складного попереднього кодування, оскільки кожен симптом позначається бінарним значенням (0 - відсутній, 1 - присутній). Завдяки цьому значно спрощується процес навчання моделі, адже ознаки вже перебувають у числовому вигляді, придатному для обробки алгоритмами глибинного навчання. Кожен рядок у датасеті відповідає окремому клінічному випадку, де поєднання симптомів формує унікальний вектор ознак, а міткою виступає відповідний діагноз.

Це створює умови для побудови моделей багатокласової класифікації, де алгоритм навчається зіставляти певні симптомокомплекси з конкретними захворюваннями.

Важливо, що дані були попередньо анонімізовані, тобто з них видалено будь-які ідентифікуючі відомості про пацієнтів. Такий підхід відповідає міжнародним стандартам роботи з медичною інформацією та забезпечує можливість використання датасету без ризику порушення конфіденційності.

Дані анонімізовані, що гарантує конфіденційність пацієнтів. Вони були згенеровані на основі відкритих клінічних джерел та публічних баз даних, збережено симптомну тяжкість та ймовірність появи захворювання, що дозволяє моделі навчатися на реалістичних патернах. Завдяки високому usability score 10, цей датасет є зручним для дослідників і має офіційну ліцензію, що дозволяє його використання у наукових роботах та прикладних системах.

Для цілей даної роботи список хвороб було обмежено виключно неврологічними діагнозами, загальна кількість яких становила 66 нозологій. Така спеціалізація дозволила зосередитися на вузькій предметній області, максимально релевантній для лікаря-невролога, і забезпечити високу точність класифікації саме в межах неврологічних патологій. Вибір діагнозів здійснювався шляхом ретельної фільтрації початкового датасету, що містив

773 захворювання різного профілю. На першому етапі було використано клінічні настанови МОЗ України, які визначають перелік найбільш поширених і клінічно значущих неврологічних хвороб у вітчизняній практиці. Це дало змогу адаптувати систему до потреб національної медичної галузі.

Далі було враховано рекомендації NHS (Великобританія) та настанови NICE, які забезпечили включення до переліку тих діагнозів, що мають доказову базу та затверджені міжнародними клінічними протоколами. Особлива увага приділялася узгодженості критеріїв діагностики між різними країнами, що дозволяє створити систему, яка не обмежується лише локальним застосуванням, а може бути масштабована й для міжнародних користувачів. Для підтвердження глобальної валідності відбору додатково були залучені матеріали NIH (США), які підтримують єдину стандартизацію знань і описують спектр неврологічних захворювань з урахуванням сучасних дослідницьких досягнень.

Такий комбінований підхід гарантує, що всі обрані діагнози підтверджені авторитетними клінічними джерелами, відповідають міжнародним і національним стандартам охорони здоров'я та відображають актуальні потреби практичної неврології. Це дозволяє підвищити надійність класифікації та довіру до результатів роботи системи. У підсумку з початкових 773 захворювань було відібрано 66 неврологічних діагнозів, на основі яких здійснювалося навчання та тестування моделі. Саме ця підмножина стала ядром побудованої СППР й визначила її орієнтацію на практичні задачі неврологічної діагностики.

3.3. Технічне середовище та інструменти

Розробка СППР виконувалась у середовищі Python 3.11, яке є однією з найпопулярніших мов програмування у сфері науки про дані та штучного інтелекту [37]. Використання цієї платформи забезпечило широкий спектр інструментів для обробки даних, побудови моделей машинного навчання,

інтеграції з базами знань, а також створення графічних інтерфейсів для користувача.

Python вирізняється відкритою екосистемою та активною спільнотою розробників, що гарантує доступ до великої кількості бібліотек та регулярні оновлення. Це робить його придатним як для наукових досліджень, так і для практичних застосувань у медицині. Використання відкритих бібліотек дало змогу поєднати високу аналітичну продуктивність із гнучкістю інтеграції, а також зменшити витрати на розробку завдяки доступності рішень із відкритим кодом.

Для роботи з даними застосовувалися бібліотеки NumPy та pandas, які забезпечували ефективну обробку та представлення симптомів і діагнозів у зручному форматі таблиць і векторів. NumPy дозволяла виконувати операції над багатовимірними масивами з високою швидкістю, що є критично важливим під час попередньої обробки великих датасетів [38]. Pandas, у свою чергу, забезпечувала структуровану роботу з таблицями даних, що дало змогу організувати симптоми та діагнози у форматі, придатному для подальшого навчання нейронної мережі [39].

Модель глибокого навчання була реалізована на основі TensorFlow / Keras, що надало можливість будувати багат шарові нейронні мережі, експериментувати з їхньою архітектурою та проводити тренування з оптимізацією ваг [40]. Для оцінювання якості передбачень застосовувалася бібліотека scikit-learn, яка містить набір стандартних метрик (accuracy, precision, recall, F1-score), необхідних для валідації моделі. Окрім цього, scikit-learn забезпечила інструменти для поділу даних на тренувальні й тестові вибірки та візуалізації результатів, що полегшило аналіз продуктивності системи та її порівняння з альтернативними підходами [41]. Збереження моделі, кодування міток та списків симптомів реалізовувалося за допомогою joblib, що дозволило швидко завантажувати попередньо оброблені об'єкти за допомогою виконання процесів в паралелі (рис. 3.5) [42] [43].

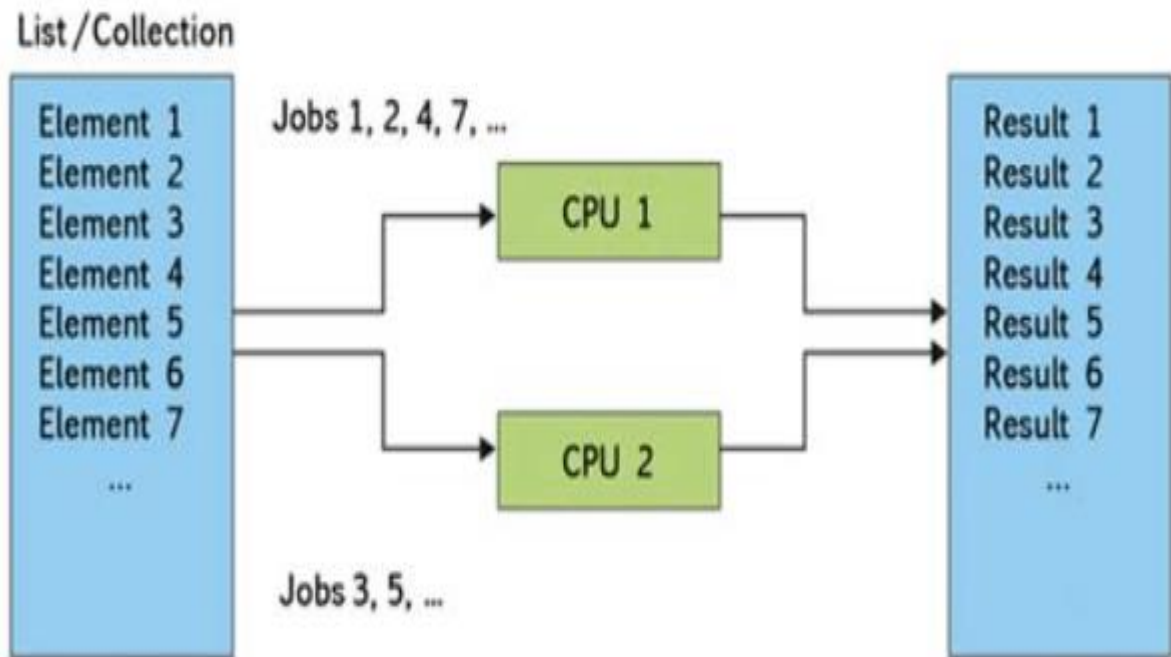


Рисунок 3.5 – Одночасна обробка декількох процесів ядрами

Окрема увага була приділена реалізації графічного інтерфейсу користувача, адже саме він визначає зручність і швидкість роботи лікаря з системою. Для цього використовувалася стандартна бібліотека Tkinter у поєднанні з віджетами ttk та scrolledtext, що забезпечило створення зручного та інтуїтивного інтерфейсу для взаємодії з користувачем.

Такий підхід дозволив організувати введення симптомів, відображення результатів прогнозування та виведення рекомендацій у структурованій і зрозумілій формі (рис. 3.6). Крім того, використання вбудованих засобів Tkinter гарантує розгортання програми без необхідності встановлення додаткових залежностей, що робить систему доступною навіть для користувачів без спеціальної технічної підготовки.

Вибір Tkinter також забезпечив високу крос-платформність, дозволяючи запускати додаток на більшості операційних систем (Windows, Linux, macOS) без необхідності внесення суттєвих змін у код.

Архітектура GUI була побудована за принципом однокадрового вікна для мінімізації відволікаючих факторів, що є важливим для клінічного середовища, де швидкість і фокус є пріоритетними.

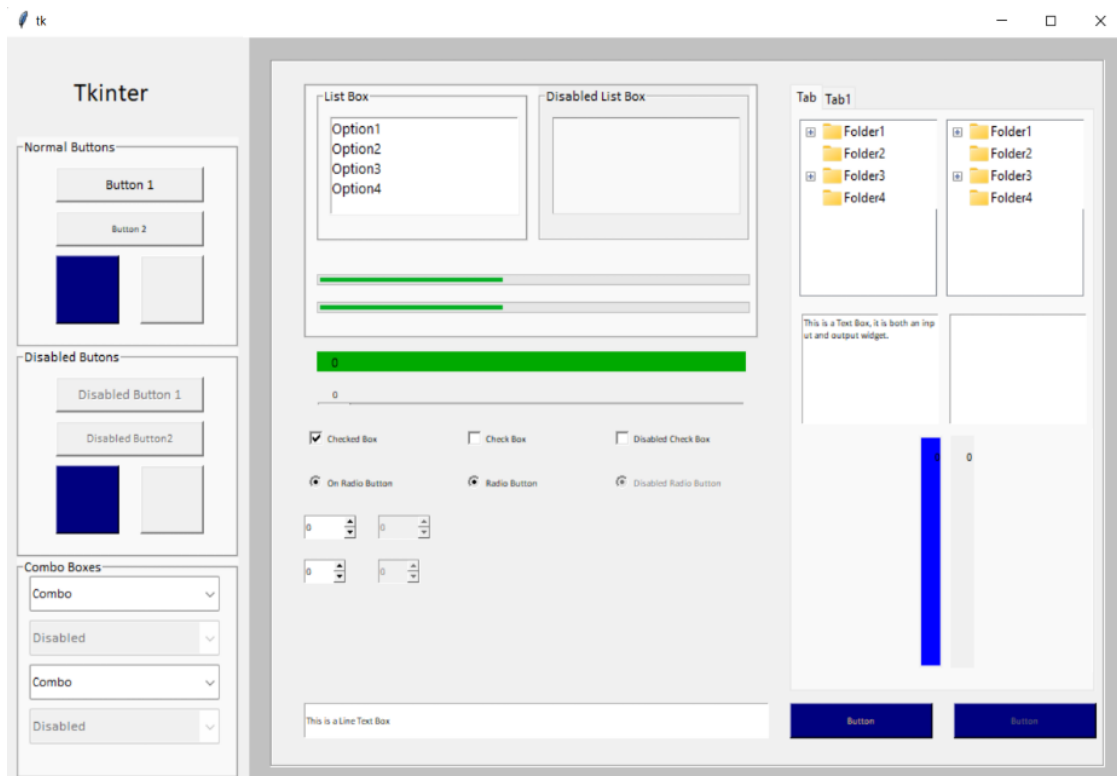


Рисунок 3.6 – Стилі графічного інтерфейсу бібліотеки tkinter

Додаткові можливості для пошуку збігів між введеними симптомами та доступними у базі забезпечувала бібліотека `difflib`, а для попередньої текстової обробки застосовувалася `re` (регулярні вирази). Інтеграція `difflib` дозволила реалізувати механізм нечіткого пошуку (*fuzzy matching*), критично важливий для роботи з людською мовою, оскільки лікарі можуть використовувати скорочення, синоніми або допускати незначні орфографічні помилки під час введення симптомів. Це значно підвищує толерантність системи до неточностей вводу, що є типовим для швидкого клінічного набору даних. Регулярні вирази, своєю чергою, використовувалися для стандартизації тексту та очищення його від зайвих символів, які могли б негативно вплинути на точність нечіткого порівняння.

Таким чином, кожен інструмент мав свою чітко визначену роль: від підготовки даних і тренування моделі до інтеграції алгоритмів у графічний додаток, що забезпечує повноцінний цикл роботи СППР – від вхідних симптомів до отримання діагнозу та рекомендацій (рис. 3.7). Цей модульний підхід дозволив не тільки спростити розробку, але й значно полегшив

подальше оновлення або заміну окремих компонентів (наприклад, перехід на більш досконалу модель машинного навчання) без необхідності переписувати весь інтерфейс.

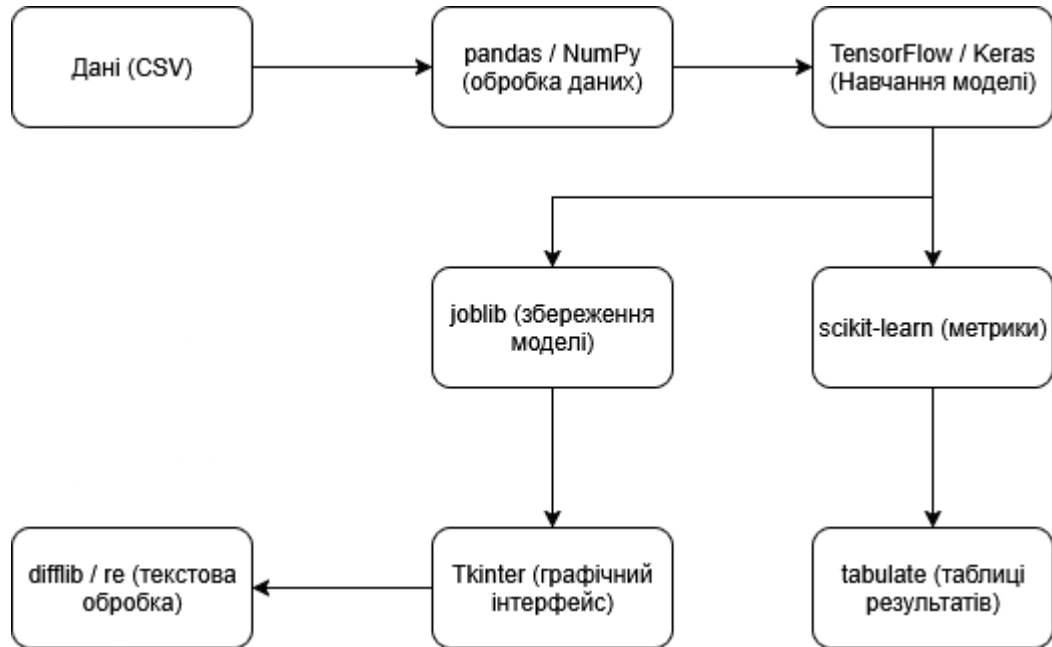


Рисунок 3.7 - Потік процесу від даних до графічного інтерфейсу

3.4. Захист даних та безпека

Оскільки СППР працює з медичною інформацією, навіть якщо дані є анонімізованими, питання конфіденційності та безпеки є ключовими. Використаний датасет із Kaggle містить анонімізовані записи пацієнтів, де всі особисті дані, такі як імена, контактні дані та інші ідентифікатори, були видалені або замінені [36]. Це гарантує, що жодна інформація не дозволяє ідентифікувати конкретну особу, що відповідає міжнародним стандартам роботи з медичними даними.

На рівні зберігання та обробки даних застосовується декілька технічних заходів. По-перше, всі датасети зберігаються у зашифрованих локальних директоріях проєкту та доступ до них обмежено лише для користувача, який здійснює розробку та навчання моделі. По-друге, при збереженні навченої

моделі та пов'язаних об'єктів (label encoder, списки симптомів) використовується структурована організація файлів із контролем доступу, що запобігає несанкціонованому модифікуванню або видаленню.

Що стосується обробки інформації всередині системи, всі вхідні дані користувачів (симптоми, запити) не зберігаються на сервері або у базі після завершення сеансу. Це забезпечує мінімізацію ризику витоку даних та відповідає принципам *privacy by design*. Для додаткового захисту можна інтегрувати шифрування запитів і відповідей, якщо система буде розгорнута у мережевому середовищі.

На рівні моделі застосовується контроль доступу до тренувальних і тестових даних, що унеможлиблює витік інформації через зовнішні запити. Важливо також, що оскільки тренування та тестування виконуються локально на зашифрованих копіях даних, ризик зловмисного доступу до конфіденційної інформації зводиться до мінімуму.

Усі ці заходи забезпечують комплексний підхід до захисту даних та безпеки, що відповідає сучасним вимогам GDPR [44] та етичним стандартам обробки медичної інформації, одночасно дозволяючи ефективно використовувати дані для навчання нейронної мережі та надання рекомендацій СППР.

Висновки до розділу 3

У третьому розділі було розглянуто ключові засоби реалізації СППР для лікаря-невролога, що поєднує сучасні методи глибинного навчання з клінічними протоколами та зручним програмним середовищем.

По-перше, проведено аналіз джерел даних і правил, які стали основою системи. Залучення авторитетних міжнародних та національних організацій, зокрема NICE, NHS, МОЗ України та NIH, дозволило сформувавши достовірний набір знань, що відповідає сучасним медичним стандартам і практиці доказової медицини. Використання таких джерел забезпечує високу якість

структурованих даних і підвищує довіру до результатів моделі як з боку лікарів, так і наукової спільноти.

По-друге, визначено й описано джерела тренувальних даних. Використання відкритого датасету з платформи Kaggle надало можливість працювати з великим масивом інформації, що включав сотні захворювань та симптомів у форматі, зручному для нейронних мереж. Завдяки ретельній фільтрації з понад 700 хвороб було відібрано 66 найбільш релевантних неврологічних діагнозів. Це дозволило сфокусувати модель на спеціалізованій предметній області та забезпечити баланс між широким охопленням і глибиною опрацювання клінічних випадків. Додатковим результатом стало формування окремого списку діагнозів для локалізації та інтеграції програми у клінічну практику.

По-третє, було визначено технічне середовище та інструменти, необхідні для побудови системи. Середовище Python 3.11 разом із бібліотеками NumPy, pandas, TensorFlow/Keras і scikit-learn забезпечило потужні можливості для обробки даних, побудови та тренування моделі. Додатково реалізовано графічний інтерфейс на основі Tkinter, який забезпечує інтуїтивну взаємодію лікаря з системою, роблячи її доступною та зрозумілою навіть для користувачів без спеціальної підготовки.

Таким чином, третій розділ підтвердив, що поєднання перевірених медичних знань, сучасних алгоритмів глибокого навчання та продуманих програмних інструментів створює надійну основу для побудови ефективної СППР. Це забезпечує не лише високу точність класифікації, але й практичну цінність у реальних клінічних умовах.

РОЗДІЛ 4

ПРОГРАМНА РЕАЛІЗАЦІЯ ТА МЕТОДИКА РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У розділі буде детально розглянуто процес створення СППР від підготовки даних до її функціонування. Розділ розпочнеться з опису попередньої обробки тренувальних даних, включаючи їх завантаження, аналіз структури та перевірку на наявність пропусків. Буде показано, як дані були відфільтровані для виділення 66 неврологічних діагнозів із загального масиву в 773 хвороби, що є ключовим для спеціалізації системи.

Далі буде представлено архітектуру нейронної мережі, що була розроблена для класифікації симптомів. Опис торкнеться процесу кодування міток за допомогою LabelEncoder та формування вхідних даних у форматі one-hot encoding. Буде детально розглянуто структуру моделі, що складається з кількох шарів з функціями активації ReLU та Softmax, а також використані гіперпараметри, такі як оптимізатор Adam та функція втрат sparse categorical crossentropy. Буде проаналізовано динаміку навчання моделі, представлено графіки точності та функції втрат, що демонструють високу ефективність та стабільність прогнозу.

Окрема увага буде приділена метрикам оцінки якості моделі, таким як accuracy, precision, recall та F1-score, що дозволить підтвердити її надійність та придатність до використання в клінічній практиці. Буде проведено порівняльний аналіз із результатами інших моделей, зокрема логістичної регресії та Random Forest, для обґрунтування вибору багатошарового перцептрону.

Нарешті, розділ міститиме детальний опис програмної реалізації, включаючи розробку графічного інтерфейсу користувача (GUI) на базі бібліотеки Tkinter. Буде розглянуто всі елементи інтерфейсу та функції, що відповідають за їх обробку, зокрема механізм "fuzzy search" для пошуку симптомів. Буде описано алгоритм роботи системи, від введення симптомів до

виведення рекомендацій, з ілюстрацією ключових етапів - від завантаження файлів, перетворення даних у числовий формат та прогнозування до відображення результатів і рекомендацій щодо лікування.

4.1. Попередня обробка тренувальних даних

На першому етапі проведено завантаження вихідного датасету із файлу `dataset.csv` та перевірено його структуру. Отриманий розмірність масиву даних свідчить про кількість прикладів (рядків) і ознак (стовпців), що надалі будуть використані для навчання моделі. На рис. 4.1 виведено перших п'яти рядків дозволяє ознайомитися з форматом подання інформації, зокрема з наявністю стовпця `diseases`, який містить цільові мітки (діагнози), та численних бінарних ознак, що відповідають за наявність чи відсутність конкретних симптомів.

```
df = pd.read_csv(ORIG_DATA_PATH)
print("Розмірність вихідного датасету:", df.shape)
print("Перші 5 рядків:")
display(df.head())
```

✓ 30s Python

Розмірність вихідного датасету: (246945, 378)
Перші 5 рядків:

	diseases	anxiety and nervousness	depression	shortness of breath	depressive or psychotic symptoms	sharp chest pain	dizziness	insomnia	abnormal involuntary movements	chest tightness	...	stuttering or stammering	problems with orgasm	nose deformity	lump over jaw	sore in nose	hip weakness	sw
0	panic disorder	1	0	1	1	0	0	0	0	1	...	0	0	0	0	0	0	0
1	panic disorder	0	0	1	1	0	1	1	0	0	...	0	0	0	0	0	0	0
2	panic disorder	1	1	1	1	0	1	1	0	0	...	0	0	0	0	0	0	0
3	panic disorder	1	0	0	1	0	1	1	1	0	...	0	0	0	0	0	0	0
4	panic disorder	1	1	0	0	0	0	1	1	1	...	0	0	0	0	0	0	0

5 rows × 378 columns

Рисунок 4.1 – Завантаження вихідного датасету

Наступним етапом було проведено оцінку загальної структури датасету та перевірку його повноти. На рис. 4.2 результати показали, що масив даних містить 378 колонок, з яких одна відповідає за діагноз (`diseases`), а решта – за симптоми у форматі бінарних ознак. Загальна кількість унікальних діагнозів

становить 773, що підтверджує широкий спектр патологій у вихідному наборі. Додатково було перевірено наявність пропусків, і розрахована частка відсутніх значень склала 0.0000%, тобто датасет є повністю заповненим і не потребує додаткової обробки пропусків.

```
print("Кількість колонок:", len(df.columns))
print("Кількість унікальних діагнозів:", df['diseases'].nunique())

missing = df.isna().sum().sort_values(ascending=False)

# Частка пропусків по всьому датасету
total_cells = df.shape[0] * df.shape[1]
missing_share = missing.sum() / total_cells
print(f"Сумарна частка пропусків: {missing_share:.4%}")
```

✓ 0.1s

Кількість колонок: 378
 Кількість унікальних діагнозів: 773
 Сумарна частка пропусків: 0.0000%

Рисунок 4.2 – Огляд ознак і пропусків

Для розуміння типового патерну клінічних випадків, що містяться у вибірці, був детально проаналізований розподіл кількості позитивних симптомів на один запис.

Ця операція включала відбір усіх колонок, що представляють симптоми (змінна `symptom_cols`), та обчислення суми бінарних значень для кожного рядка, що відповідає кількості наявних симптомів у пацієнта (змінна `symptoms_per_row`).

Отримана описова статистика на рис. 4.3 підтверджує, що усього було розглянуто 246 945 прикладів. У середньому на один запис припадає близько 5,33 симптома, тоді як мінімальне значення дорівнює 1, а максимальне – 12. Медіанне значення становить 5 симптомів, а міжквартильний діапазон [Q1–Q3] охоплює інтервал від 4 до 6 симптомів.

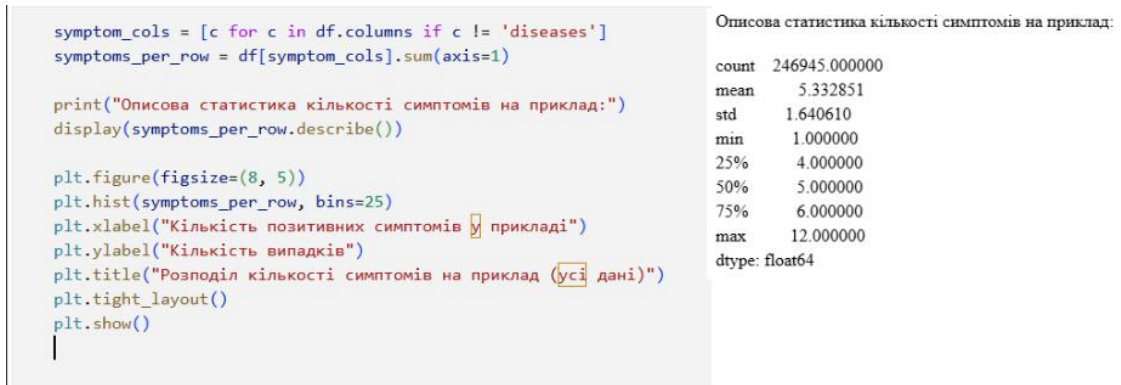


Рисунок 4.3 - Розподіл кількості симптомів на один приклад

Ця інформація є критично важливою для оцінки складності навчальної задачі, оскільки вона показує, що більшість випадків мають помірну, але достатню кількість діагностично релевантних ознак. Додатково для візуальної оцінки був побудований графік, ілюстрований на рис. 4.4, де гістограма розподілу кількості симптомів на приклад чітко демонструє, що основна маса записів містить від 4 до 6 симптомів. Такий патерн є типовим для клінічних даних, де пацієнти рідко мають велику кількість одночасних проявів. Це підтверджує необхідність використання методів глибинного навчання, здатних ефективно уловлювати нелінійні залежності та формувати точні висновки навіть за відносно невеликої кількості позитивних ознак у вхідному векторі.



Рисунок 4.4 – Гістограма кількості симптомів на приклад

Дані для тренування моделі були попередньо відфільтровані з повного датасету, що містив понад сім сотень різних захворювань, до спеціалізованої підмножини, яка включає лише неврологічні діагнози. Для цього було сформовано список із 66 хвороб на основі клінічних настанов МОЗ України, а також міжнародних рекомендацій NICE, NHS та NIH, що дозволило виділити релевантні для неврології нозології. Використання таких документів гарантує відповідність набору даних сучасним медичним стандартам і підвищує достовірність подальших результатів моделі. На основі цього списку з вихідного датасету було відфільтровано всі рядки (рис. 4.5), що відповідають зазначеним діагнозам, у результаті чого сформовано навчальну підмножину даних для побудови класифікатора.

```

# шляхи до файлів
DATA_PATH = "dataset/dataset.csv"
NEURO_LIST_PATH = "dataset/n_diseases.csv"
FILTERED_DATA_PATH = "d/n_dataset.csv"
LOC_PATH = "d/n_diseases_for_localization.csv"

# завантажуюємо повний датасет
df = pd.read_csv(DATA_PATH)
print("Повний датасет:", df.shape, "унікальних діагнозів:", df['diseases'].nunique())

# завантажуюємо список неврологічних діагнозів (66)
neuro_df = pd.read_csv(NEURO_LIST_PATH)
neuro_list = neuro_df['diseases'].tolist()
print("Неврологічних діагнозів:", len(neuro_list))

# фільтрація датасету за списком
df_neuro = df[df['diseases'].isin(neuro_list)].copy()
print("Відфільтрований датасет:", df_neuro.shape,
      | "унікальних діагнозів:", df_neuro['diseases'].nunique())

# збереження результатів
df_neuro.to_csv(FILTERED_DATA_PATH, index=False)
pd.DataFrame({'diseases': neuro_list}).to_csv(LOC_PATH, index=False)

```

✓ 4.3s
Повний датасет: (246945, 378) унікальних діагнозів: 773
Неврологічних діагнозів: 66
Відфільтрований датасет: (18918, 378) унікальних діагнозів: 66

Рисунок 4.5 – Фільтрація датасету

У першій частині процесу, ілюстрованого на рисунку 4.5, відбувається визначення шляхів до необхідних файлів за допомогою змінних DATA_PATH,

NEURO_LIST_PATH, FILTERED_DATA_PATH та LOC_PATH. Далі виконується завантаження основного датасету, що зберігається у файлі `dataset.csv`, який містить понад 700 захворювань та сотні тисяч прикладів. Це є початковим, максимально широким набором даних для подальшої спеціалізації. Одночасно завантажується окремий допоміжний файл `n_diseases.csv`, що містить список цільових неврологічних діагнозів, який був попередньо сформований на основі документів МОЗ, NICE, NHS та NIH. Після завантаження, вміст цього файлу конвертується у Python-список `neuro_list`, у якому зберігаються назви цільових діагнозів – усього 66 нозологій. Формування цього списку дає можливість чітко визначити, які саме записи з основного датасету потрібно залишити для подальшого аналізу й навчання моделі, забезпечуючи вузьку спеціалізацію системи.

У другій частині цього етапу здійснюється безпосередня фільтрація основного датасету. Залишаються лише ті рядки, у яких значення колонки `diseases` точно відповідає діагнозам зі списку `neuro_list`. Таким чином формується спеціалізований піднабір даних, названий `df_neuro`, що містить виключно неврологічні випадки. Цей відфільтрований датасет зберігається у файлі `n_dataset.csv` (FILTERED_DATA_PATH), який є основним тренувальним набором для нейронної мережі. Додатково, щоб забезпечити коректну роботу графічного інтерфейсу, список цих 66 діагнозів зберігається у файлі `n_diseases_for_localization.csv` (LOC_PATH). Цей окремий файл буде використано для перекладу назв хвороб та їх подальшої локалізації в інтерфейсі програми, що є важливим для забезпечення коректної взаємодії лікаря-невролога із системою.

4.2. Побудова і навчання моделі: архітектура, гіперпараметри

СППР побудована за модульною архітектурою, яка включає три основні компоненти: GUI, блок попередньої обробки симптомів і нейромережеву модель класифікації. Взаємодія цих модулів відбувається послідовно:

користувач через інтерфейс обирає симптоми, які перетворюються у вектор ознак методом one-hot encoding, після чого модель здійснює класифікацію та повертає прогноз разом із рекомендаціями. Такий підхід дозволяє поєднати інтуїтивно зрозумілий спосіб взаємодії лікаря із системою з високоточними обчислювальними можливостями глибокого навчання.

На етапі підготовки даних до тренування моделі (рис. 4.6), який передуює розбиттю на вибірки, було виконано критично важливий процес кодування класів. Оскільки нейронна мережа працює лише з числовими значеннями, текстові назви діагнозів, що містяться у колонці цільової змінної, повинні бути перетворені на цілочисельні мітки. Ця операція здійснювалася за допомогою інструменту LabelEncoder з бібліотеки scikit-learn. Кодувальник спочатку вивчав усі унікальні текстові назви хвороб, а потім трансформував їх у послідовні числові мітки (від 0 до 65), які були збережені в окремій допоміжній колонці таблиці даних.

```
le = LabelEncoder()
df['label'] = le.fit_transform(df['diseases'])

# Drop original disease column
df = df.drop(["diseases"], axis=1).copy()

# Features and labels
X = df.drop(["label"], axis=1)
y = df["label"]

# Save symptom column names
joblib.dump(list(X.columns), SYMPTOM_COLS_PATH)
```

Рисунок 4.6 - Кодування міток та формування ознак

Оскільки всі подальші обчислення повинні ґрунтуватися виключно на числових даних, вихідна колонка з текстовими назвами діагнозів була видалена з основного масиву даних. Наявні бінарні ознаки, що відповідають за симптоми, вже були представлені у форматі one-hot векторів, де наявність чи відсутність кожної ознаки позначалася двійковим значенням. Після цього

таблиця даних була чітко розділена на два основних компоненти: набір вхідних ознак (змінна X), що містив виключно бінарні симптоми, та вектор цільових міток (змінна y), що містив закодовані числові діагнози.

Фінальним кроком цього етапу було збереження повного списку назв колонок симптомів. Цей список зберігався окремим файлом за допомогою бібліотеки `joblib`. Це необхідно для забезпечення узгодженості: під час використання СППР в режимі прогнозування, цей збережений список гарантує, що вхідний вектор симптомів завжди матиме правильний порядок та фіксовану розмірність, яку очікує навчена нейронна мережа.

Після кодування даних було виконано їх розділення на тренувальний та тестовий набори у співвідношенні 80/20 (рис. 4.7). Це класичний підхід у задачах машинного навчання, що дозволяє з одного боку забезпечити достатній обсяг даних для оптимізації ваг моделі, а з іншого - залишити незалежну вибірку для перевірки узагальнюючої здатності системи. Розподіл відбувався таким чином, щоб уникнути дисбалансу між класами і зберегти репрезентативність як поширених, так і рідкісних діагнозів.

```
x_train, x_test, y_train, y_test = train_test_split(
|   X, y, test_size=0.2, random_state=42
)
```

Рисунок 4.7 – Розбиття даних на вибірки

Крім того, всі назви симптомів було збережено у окремому файлі, що забезпечує відтворюваність експериментів та полегшує інтеграцію з GUI. Завдяки цьому в майбутньому можна легко завантажувати модель разом із повним переліком симптомів без необхідності повторної підготовки даних. Для відтворюваності результатів було зафіксовано параметр `random_state`.

Нейронна мережа була реалізована у середовищі TensorFlow/Keras як послідовна модель типу `Sequential`, що є одним із найбільш поширених підходів для побудови багат шарових перцептронів у задачах класифікації. На

вхідний шар подаються вектори ознак, розмірність яких дорівнює кількості симптомів у підготовленому датасеті. Якщо позначити кількість симптомів як n , то кожен вхідний вектор має вигляд:

$$X = [x_1, x_2, x_3, \dots, x_n], \quad x_i \in \{0,1\} \quad (4.1)$$

де x_i відповідає наявності (1) або відсутності (0) конкретного симптому. Це дозволяє кожному симптому виступати як окрема бінарна змінна у форматі one-hot encoding, формуючи повний простір ознак для класифікації.

Перший прихований шар складається з 512 нейронів і використовує функцію активації ReLU (Rectified Linear Unit), яка задається формулою:

$$f(x) = \max(0, x) \quad (4.2)$$

Вибір ReLU обумовлений її здатністю ефективно розв'язувати проблему затухання градієнтів, пришвидшувати навчання та забезпечувати виявлення складних нелінійних залежностей між симптомами [45].

Другий прихований шар виконує функцію ущільнення та містить 256 нейронів із тією ж активацією ReLU. Зменшення кількості нейронів на цьому етапі дозволяє знизити розмірність простору ознак, зберігаючи при цьому суттєві патерни даних та запобігаючи перенавчанню. Архітектурне рішення із поступовим скороченням кількості нейронів відповідає класичним підходам до побудови глибоких моделей для задач медичної класифікації, де важливо поєднати високу точність прогнозу з обмеженням надмірної складності моделі [46].

На вихідному шарі кількість нейронів відповідає числу унікальних неврологічних діагнозів, які залишилися після попередньої фільтрації даних (усього $k = 66$ діагнозів). Для цього використовується функція активації

Softmax, яка перетворює вихідні значення моделі (логіти) у ймовірнісний розподіл:

$$P(y = c | X) = \frac{e^{z_c}}{\sum_{j=1}^k e^{z_j}}, \quad c \in \{1, 2, \dots, k\} \quad (4.3)$$

де z_c - вихідне значення (логіт) для класу c , а знаменник є нормалізувальним коефіцієнтом, що забезпечує суму ймовірностей рівною 1.

Таким чином, вихід мережі можна інтерпретувати як набір імовірностей належності пацієнта до кожного з можливих діагнозів. Це забезпечує лікарю не лише прогнозоване захворювання, а й інформацію про ступінь впевненості моделі у своєму рішенні.

Такий підхід як на рис. 4.8 є особливо важливим у медичній практиці, оскільки дозволяє враховувати ймовірності альтернативних діагнозів, формуючи більш зважене клінічне рішення.

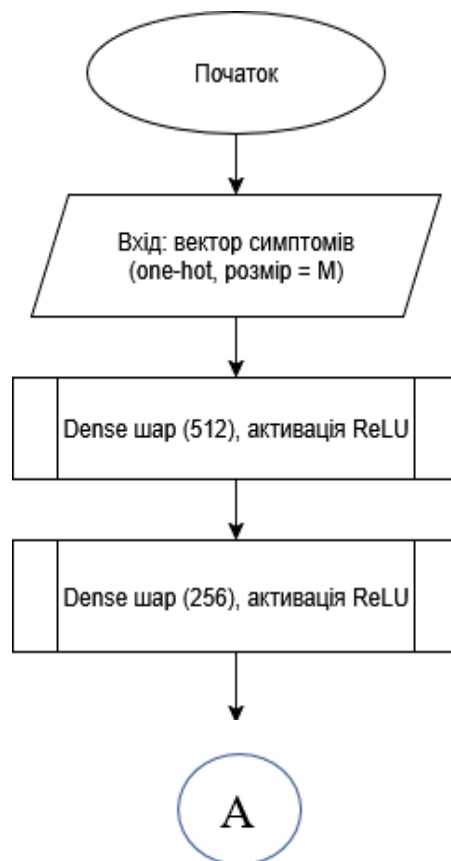


Рисунок 4.8 – Блок-схема архітектури моделі

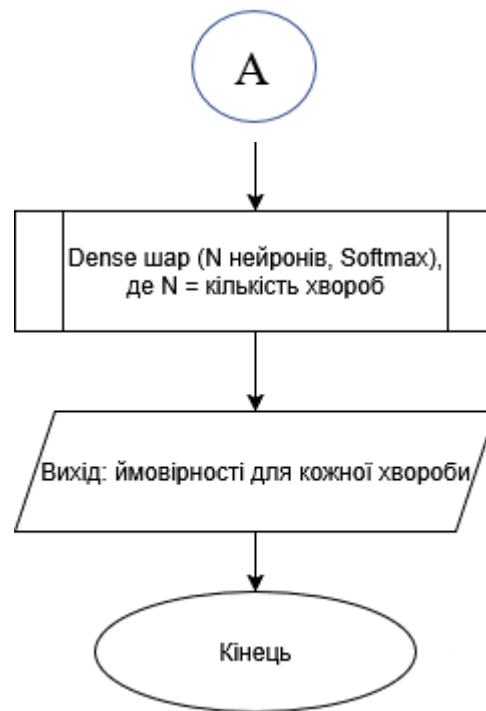


Рисунок 4.8, аркуш 2

Обрана архітектура моделі, представлена на рис. 4.9, є Багатошаровим Перцептроном і була реалізована за допомогою послідовного API бібліотеки Keras. Ця конфігурація була спеціально розроблена для ефективної класифікації великої кількості незалежних класів на основі високорозмірних бінарних вхідних даних. Архітектура складається з трьох ключових шарів. Вхідний шар моделі динамічно адаптується під розмірність підготовленого датасету, приймаючи вектор із 377 бінарних ознак, що відповідає загальній кількості стандартизованих симптомів. Далі слідує два приховані шари типу Dense.

```

model = Sequential([
    Dense(512, input_shape=(X.shape[1],), activation='relu'),
    Dense(256, activation='relu'),
    Dense(len(y.unique()), activation='softmax')
])

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model.summary()
  
```

Рисунок 4.9 – Архітектура моделі

Перший прихований шар містить 512 нейронів, а другий – 256 нейронів. В обох шарах застосовується функція активації ReLU, яка є необхідною для введення нелінійності в модель та ефективного навчання глибоких патернів, уникаючи при цьому проблеми згасаючого градієнта. Вихідний шар має розмірність, що дорівнює кількості унікальних діагнозів у відфільтрованому датасеті (66 нейронів). Тут використовується функція Softmax, яка нормалізує вихідні значення у вектор імовірностей, де сума всіх імовірностей дорівнює одиниці, що дозволяє інтерпретувати результат як імовірність належності клінічного випадку до кожного з 66 неврологічних класів.

Для оптимізації обрано алгоритм Adam, який поєднує властивості методів Adagrad і RMSProp та забезпечує адаптивність швидкості навчання для кожного параметра. Основна ідея полягає у використанні моментів першого та другого порядку:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad (4.4)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \quad (4.5)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \quad (4.6)$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (4.7)$$

де g_t - градієнт на кроці t , m_t і v_t - оцінки першого та другого моментів, β_1, β_2 - коефіцієнти згладжування, η - швидкість навчання [47].

Як функцію втрат використано *sparse categorical crossentropy*, оскільки мітки діагнозів були закодовані у вигляді цілих чисел. Її математичний вираз:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^N \log(\hat{y}_{i,y_i}) \quad (4.8)$$

де N - кількість прикладів у пакеті, y_i - правильна мітка для прикладу i , а \hat{y}_{i,y_i} - прогнозована мережею ймовірність правильного класу [48].

Для контролю прогресу навчання застосовувалася метрика *accuracy* (точність), яка визначається як:

$$Accuracy = \frac{\text{Кількість правильних прогнозів}}{\text{Загальна кількість прикладів}} \quad (4.9)$$

Ця метрика дозволяла оцінювати відсоток коректних класифікацій у процесі тренування та валідації, а також відслідковувати ефективність моделі у динаміці.

Навчання здійснювалося протягом 20 епох із розміром пакета (*batch size*) у 100 прикладів, що дозволило досягти оптимального балансу між швидкістю збіжності алгоритму та якістю узагальнення на невидимих даних:

```
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=20, batch_size=100)
```

Поступові графіки зміни динаміки точності та функції втрат можна побачити на рис 4.10 та рис. 4.11.

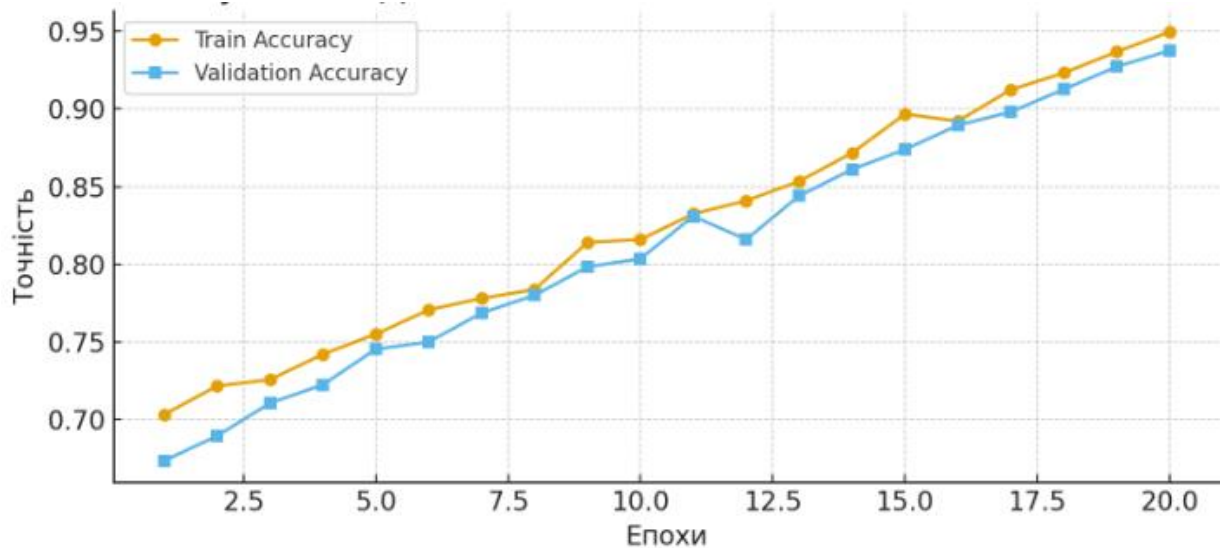


Рисунок 4.10 – Динаміка точності моделі під час навчання

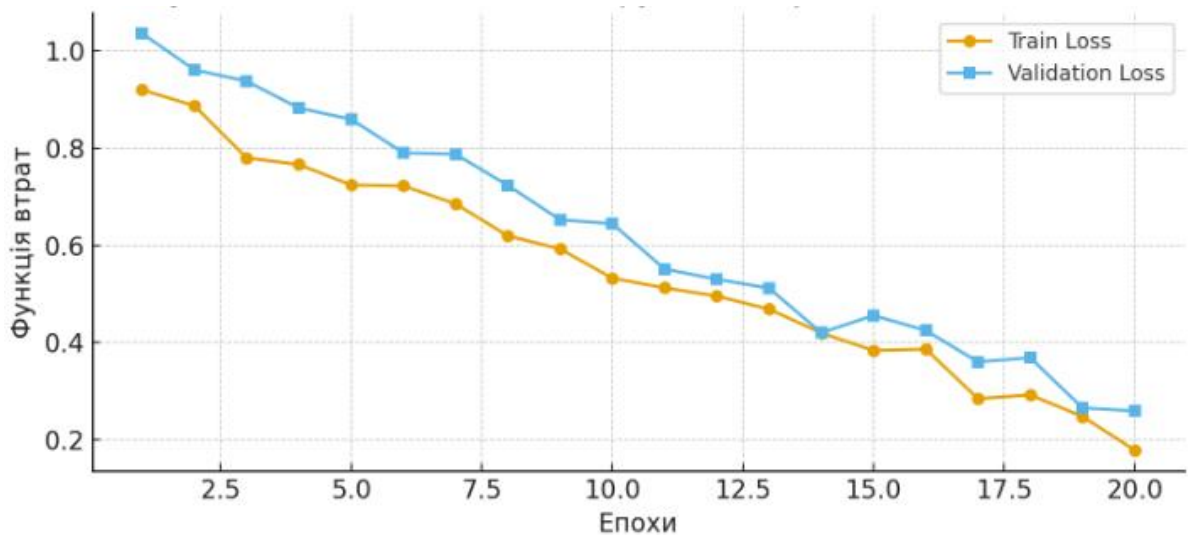


Рисунок 4.11 – Динаміка зміни функції втрат під час навчання

Для подальшого використання модель, енкодер міток та список симптомів були збережені у відповідних файлах. Це дало змогу інтегрувати їх у графічний інтерфейс користувача та забезпечити повторне використання без необхідності перенавчання. Збереження навченої моделі у форматі .h5 та пов'язаних об'єктів через joblib забезпечує відтворюваність експериментів і стабільність результатів.

Таким чином, побудована архітектура (рис. 4.12) системи поєднує інтерфейс для зручної взаємодії користувача, модуль попередньої обробки для

формалізації введених симптомів та нейронну мережу, яка здійснює високоточну класифікацію.

Вибір глибокої моделі з двома прихованими шарами виявився оптимальним компромісом між складністю й узагальнюючою здатністю. Завдяки цьому система здатна забезпечувати лікаря-невролога достовірними діагностичними підказками, залишаючись при цьому достатньо простою для інтеграції у практичні клінічні сценарії.

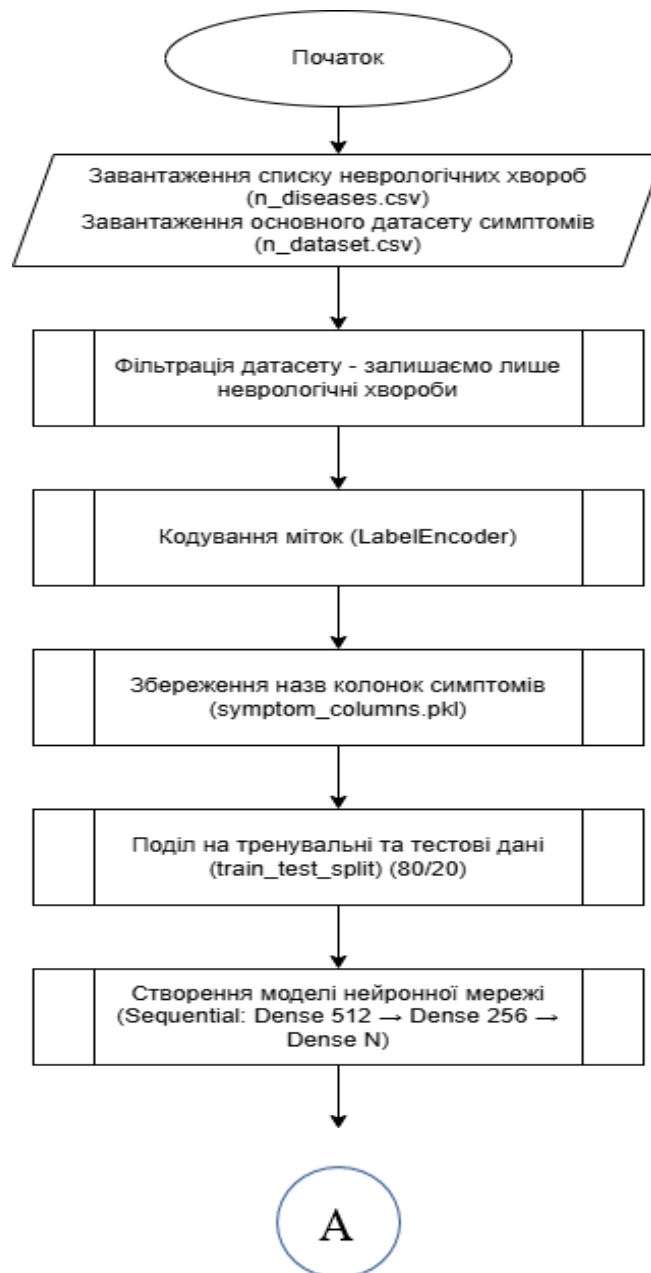


Рисунок 4.12 – Блок-схема загального процесу тренування моделі

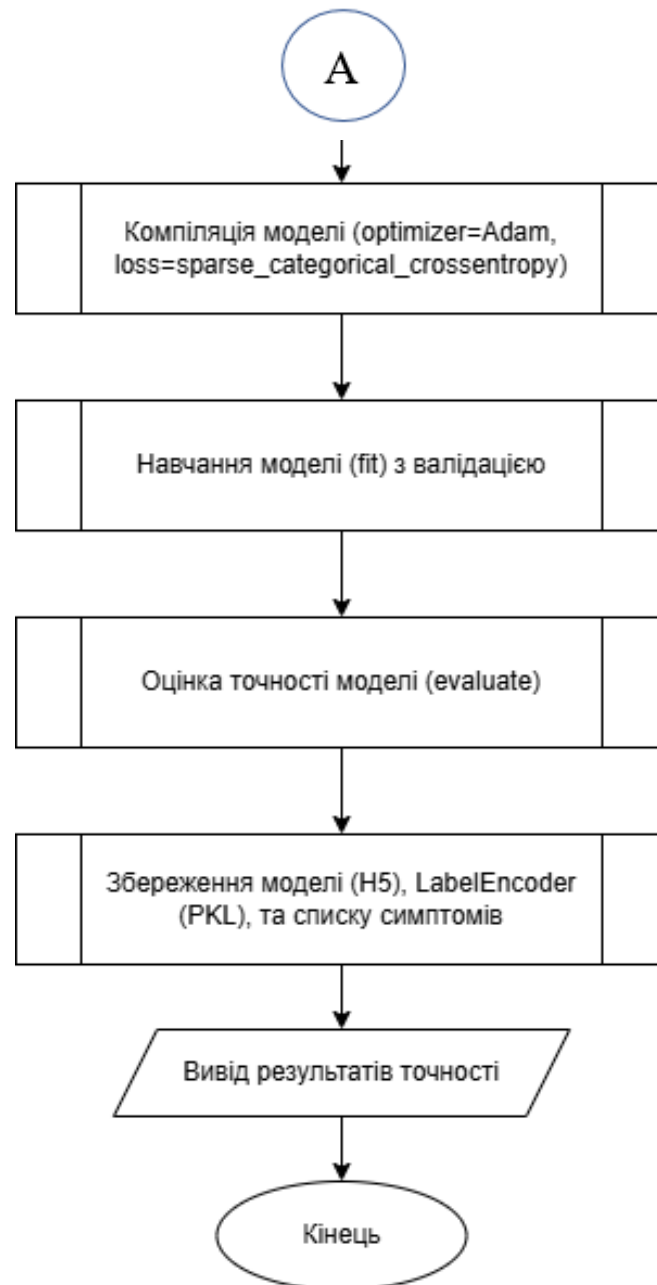


Рисунок 4.12, аркуш 2

4.3. Метрики оцінки якості моделі

Оцінка якості роботи побудованої СППР є критично важливим етапом дослідження, адже саме від цього залежить її надійність і придатність до практичного використання у медицині. Для діагностики неврологічних захворювань, де будь-яка помилка може мати серйозні наслідки, недостатньо лише визначити загальну точність. Важливо перевірити, наскільки система здатна коректно класифікувати як поширені, так і рідкісні діагнози. Саме з

цією метою було застосовано такі метрики, як accuracy, precision, recall та F1-score.

Загальна точність (accuracy) відображає частку правильно класифікованих випадків і дає уявлення про загальну якість моделі. Водночас у випадку дисбалансу класів, коли деякі захворювання зустрічаються рідко, однієї цієї метрики недостатньо. Тому було додатково розглянуто precision, що характеризує частку правильних позитивних прогнозів серед усіх, які система визначила як позитивні. Для медицини високий рівень precision означає мінімізацію хибнопозитивних діагнозів, які можуть призвести до неправильного лікування. Не менш важливим є показник recall, що демонструє здатність моделі виявляти всі випадки конкретного захворювання. Цей показник особливо критичний у неврології, адже пропуск реального випадку означає втрату часу для пацієнта і ризик тяжких ускладнень. Для балансування між precision і recall використовується F1-score, що розраховується як гармонічне середнє цих двох показників.

Результати експериментів свідчать, що побудована нейронна мережа досягла високої ефективності, зберігаючи баланс між усіма ключовими метриками. Інтегральні показники Accuracy = 0.9459, Precision = 0.9497, Recall = 0.9459, F1-score = 0.9460 - демонструють узагальнені результати роботи системи.

Зведені результати підтверджують стабільність моделі та її здатність коректно розпізнавати широкий спектр неврологічних захворювань. Повний класифікаційний звіт показав, що точність залишалася високою як для поширених діагнозів (інсульт, черепно-мозкові травми), так і для менш поширених патологій. У випадку рідкісних захворювань якість була дещо нижчою через обмежений обсяг навчальних прикладів, що пояснюється низьким значенням support. Повний звіт наведено у додатках, аби уникнути перевантаження основного тексту.

Окремо було досліджено питання вибору моделі. Логістична регресія, попри свою швидкість і простоту, забезпечувала точність лише на рівні 75–

85%, втрачаючи здатність виявляти складні нелінійні залежності між симптомами. Випадковий ліс (Random Forest) продемонстрував вищі результати з точністю близько 83–90% на тому самому наборі даних. Його перевагами стали стійкість до перенавчання та здатність працювати з високовимірними ознаками. Проте при порівнянні з багат шаровим перцептроном (ANN) він поступився, адже нейронна мережа після налаштування гіперпараметрів досягла точності понад 93%, зберігаючи при цьому високу узгодженість precision і recall.

Для подальшої оптимізації архітектури Багат шарового Перцептрону було проведено серію експериментів із варіацією ключових гіперпараметрів. Насамперед, було протестовано різні конфігурації прихованих шарів: від одного шару з 256 нейронами до трьох шарів із конфігурацією 512-256-128. Було встановлено, що конфігурація з двома прихованими шарами (512 та 256 нейронів) забезпечує найкращий баланс між складністю моделі та ризиком перенавчання, а також мінімізує час тренування. Крім того, досліджувався вплив різних функцій активації: тоді як сигмоїда та гіперболічний тангенс показали дещо нижчу збіжність, використання функції ReLU дозволило ефективно боротися з проблемою згасаючого градієнта та стабільно підвищувало точність на кожній епосі. Після фінального налаштування (оптимізатор Adam, швидкість навчання 0.001), цільова архітектура ANN продемонструвала переважаючу точність 94.59%, остаточно підтвердивши, що саме глибинне навчання є необхідним для досягнення клінічно значущих показників якості у задачі класифікації 66 неврологічних діагнозів.

4.4. Опис модулів ПЗ і інтерфейсу користувача (GUI)

Інтерфейс користувача розроблений таким чином, щоб забезпечити максимально зручну взаємодію з СППР для прогнозування неврологічних захворювань (рис. 4.13). У верхньому лівому кутку розташований елемент вибору мови, що дозволяє користувачеві перемикатися між українською та

англійською мовою. За замовчуванням при запуску додатку вибрана англійська мова інтерфейсу. Це особливо важливо для медичних фахівців, які можуть працювати з міжнародною документацією або локальними керівництвами.

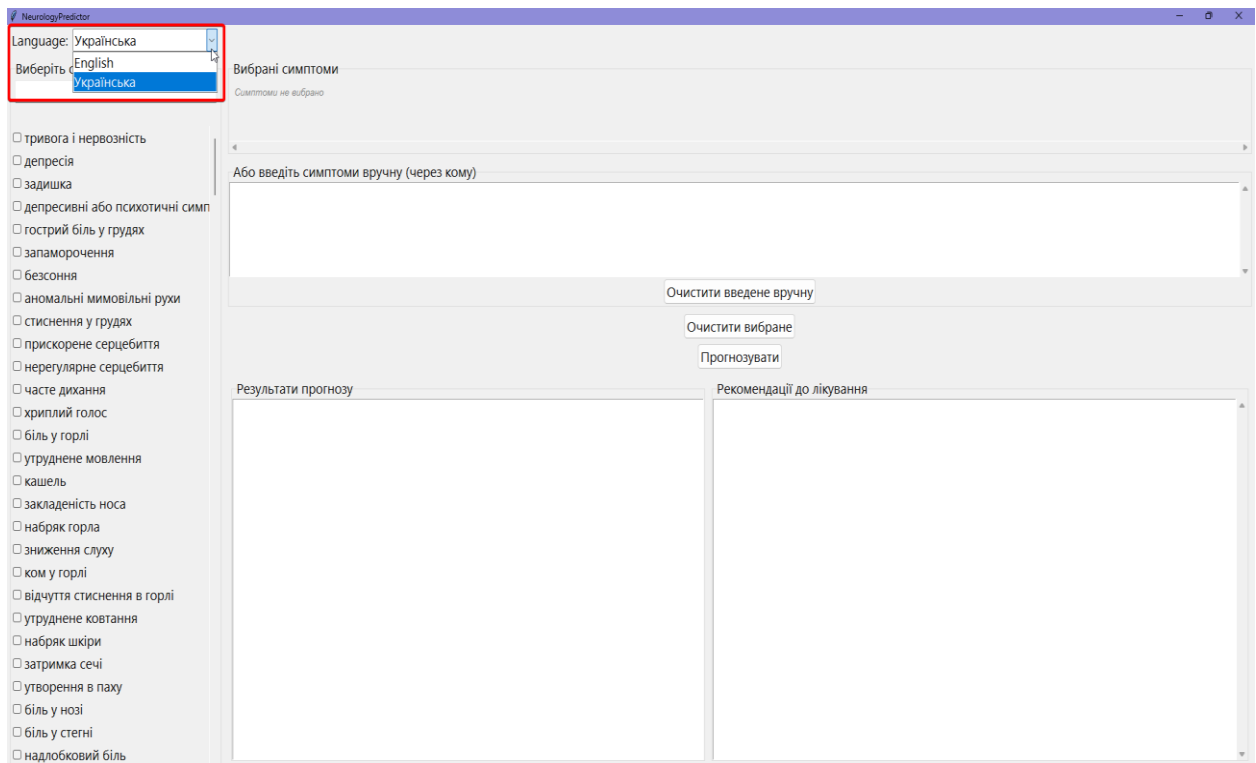


Рисунок 4.13 - Локалізація інтерфейсу

Нижче ліворуч розташований скролл-елемент зі списком симптомів, де кожен симптом представлений як чекбокс. Користувач може вибрати один або декілька симптомів, після чого вони автоматично відображаються у вікні "Вибрані симптоми" (рис. 4.14). Такий підхід дозволяє швидко формувати вхідні дані для моделі, не заповнюючи їх вручну.

Для полегшення пошуку конкретного симптому реалізовано fuzzy search з використанням бібліотеки Python difflib (рис. 4.15). Користувач може вводити часткові або неповні слова, а система знаходить найбільш схожі результати. Навіть на великих об'ємах даних ця бібліотека дозволяє швидко фільтрувати необхідні дані які введено в пошукове вікно не затримуючи виконання програми.

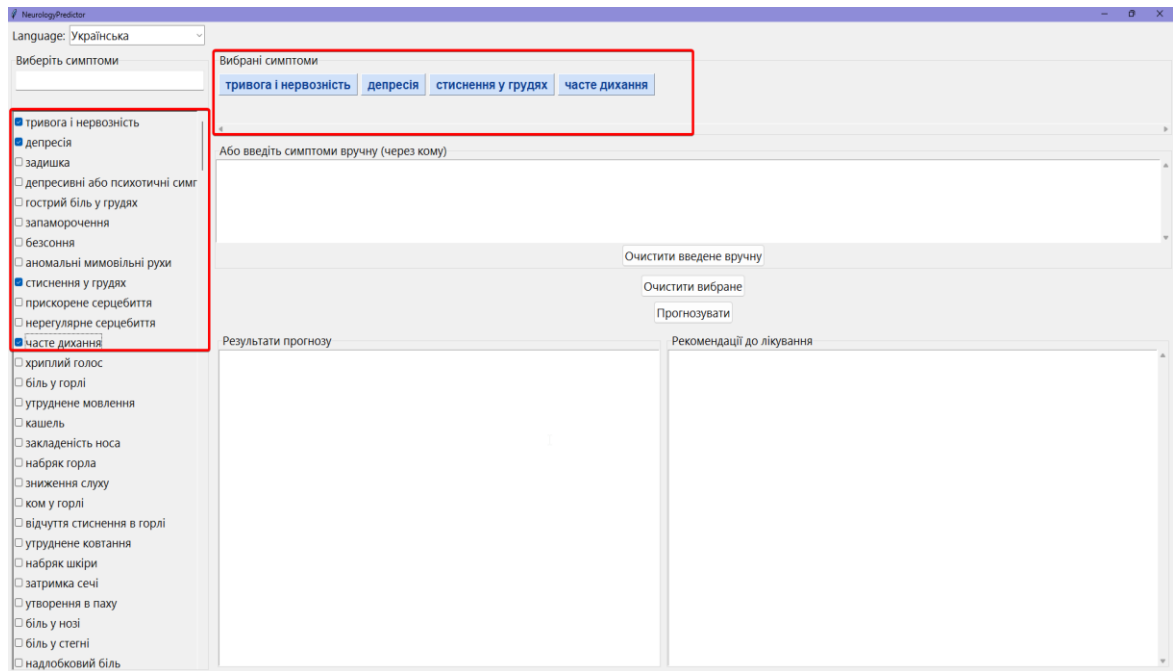


Рисунок 4.14 - Вибір симптомів та відображення вибраних симптомів

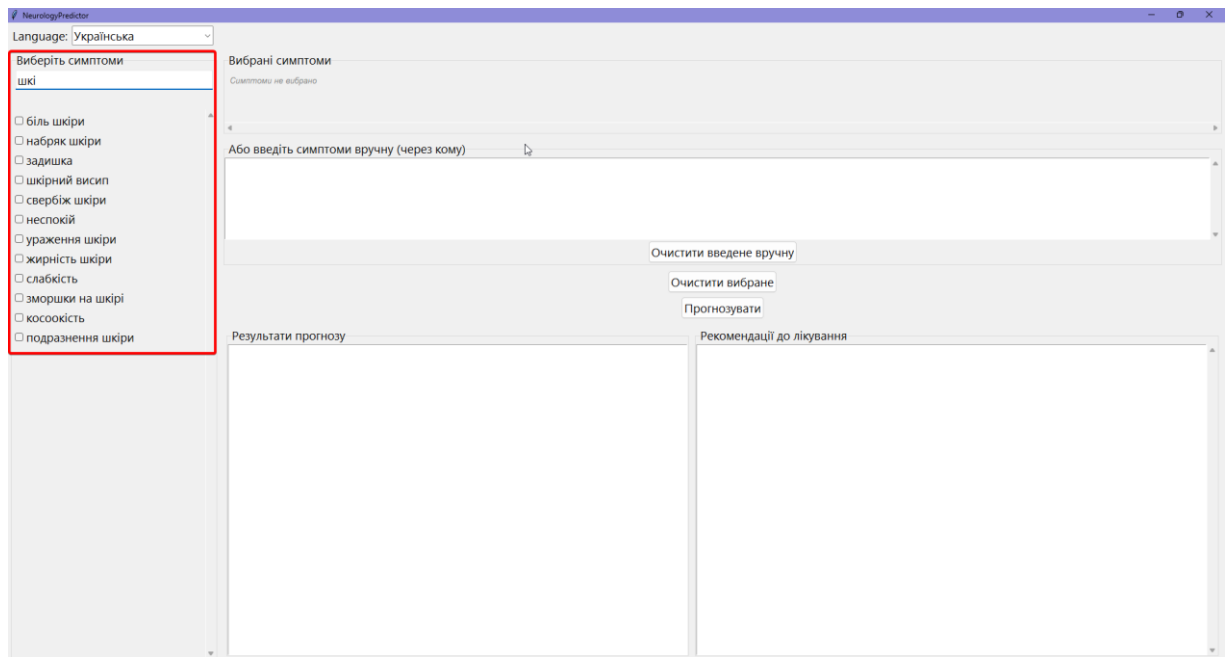


Рисунок 4.15 - Пошук доступних симптомів

Якщо відповідних симптомів не знайдено, виводиться повідомлення про відсутність збігів (рис. 4.16).

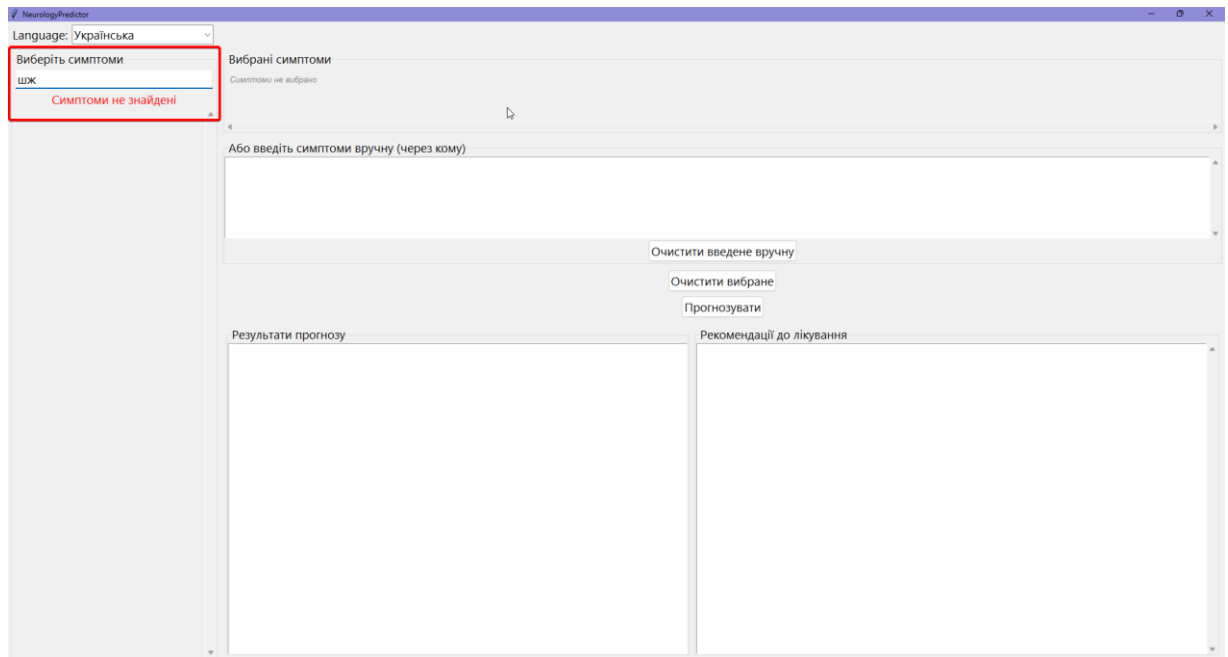


Рисунок 4.16 - Відсутність збігів

Користувач також може вводити симптоми вручну через текстове поле, розділяючи їх комами (рис. 4.17). Це зручно, якщо список симптомів відомий заздалегідь. Існує кнопка «Очистити введене вручну», яка видаляє всі введені вручну симптоми та існує ще одна кнопка «Очистити вибране» вона знімає вибрані чекбокси, що дозволяє швидко почати новий сеанс.

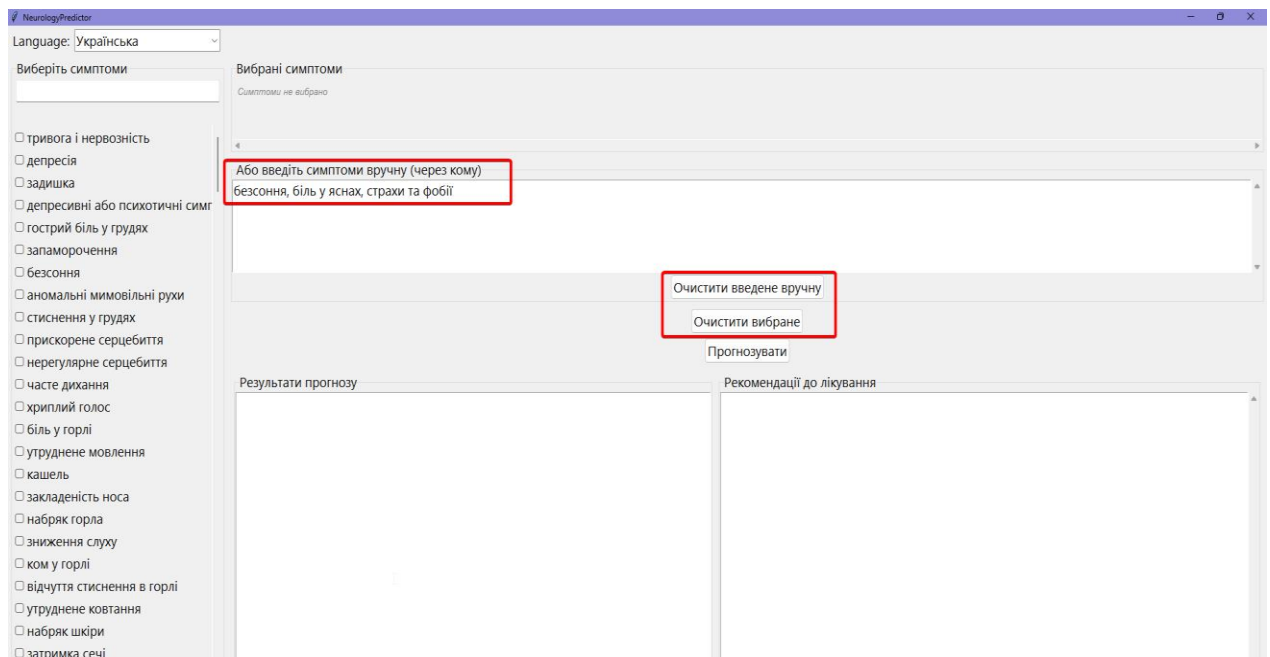


Рисунок 4.17 - Очищення введених симптомів

Праворуч розташоване вікно "Результати прогнозу", де відображаються топ-10 передбачених діагнозів (рис. 4.18) після натискання кнопки "Прогнозувати". Кожен діагноз є клікабельним: при натисканні відкривається вікно рекомендацій по лікуванню (рис. 4.19), яке показує перевірені дані з вебсайтів NICE, NHS та МОЗ. Це дозволяє лікарю швидко отримати не тільки ймовірний діагноз, але й практичні поради щодо лікування, що підвищує ефективність використання системи.

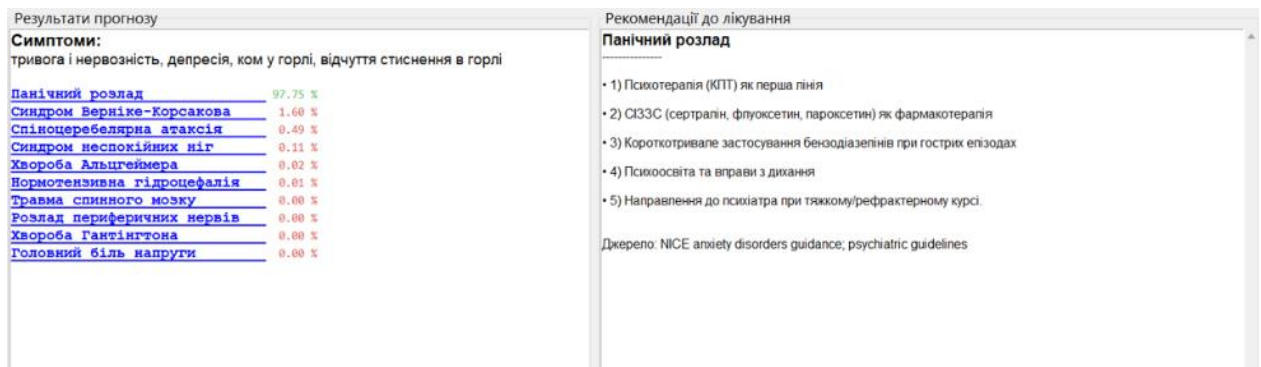


Рисунок 4.18 - Вигляд вікна результатів прогнозування

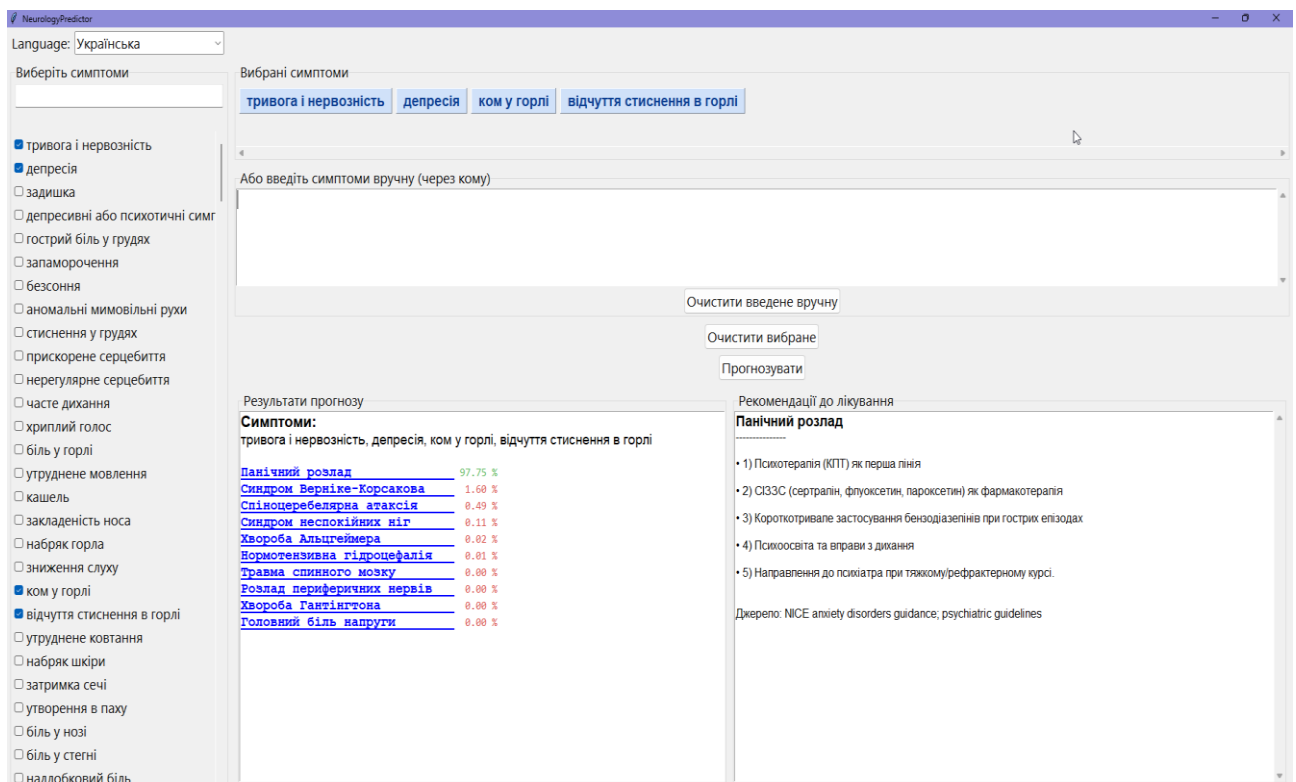


Рисунок 4.19 - Повний вигляд вікна з результатами та рекомендаціями

Таким чином, структура GUI розроблена для логічного, інтуїтивного і безпечного введення даних, прогнозування та отримання рекомендацій. Використання чекбоксів, fuzzy search та інтерактивних результатів забезпечує швидкий доступ до функціоналу, мінімізує ризик помилок користувача та підвищує ефективність системи у клінічних сценаріях.

4.5. Алгоритм роботи: від введення симптомів до виводу рекомендацій

Алгоритм роботи системи NeurologyPredictor представляє собою послідовний процес обробки введених користувачем симптомів та генерації персоналізованих медичних рекомендацій. Весь процес можна розділити на декілька ключових етапів, кожен з яких виконує специфічні функції для забезпечення точності діагностики та релевантності лікувальних рекомендацій.

Етап 1: Ініціалізація системи та завантаження моделей (рис. 4.20)

Процес функціонування СППР починається з ініціалізації основного класу SymptomChecker, який успадковує функціональність графічного інтерфейсу Tkinter. Це первинний етап конфігурації, де встановлюються базові параметри вікна програми, такі як заголовок "NeurologyPredictor", його розміри (1200x700 пікселів), та уніфікований шрифт для всіх елементів керування.

На цьому етапі відбувається критично важливе завантаження всіх необхідних компонентів для роботи системи в реальному часі. Система одноразово імпортує натреновану нейронну мережу з файлу neuro_disease_predictor.h5 та відповідний кодувальник міток (змінна label_encoder), що був використаний на етапі тренування та зберігається у файлі label_encoder.pkl. Цей кодувальник необхідний для зворотного перетворення числових прогнозів моделі у зрозумілі текстові назви діагнозів.

Також завантажується повний список стандартизованих симптомів (`all_symptoms_en`), що зберігається у файлі `symptom_columns.pkl`. Завантаження цих великих об'єктів заздалегідь, при старті програми, забезпечує миттєве отримання діагностичних результатів у подальшому.

```
class SymptomChecker(tk.Tk):
    PILL_BG =  "#d0e1f9"
    PILL_FG =  "#0b3d91"

    def __init__(self):
        super().__init__()
        default_font = tkFont.nametofont("TkDefaultFont")
        default_font.configure(size=14)
        self.option_add("*Font", default_font)
        # --- Config (шляхи) ---
        self.model_path = "model/neuro_disease_predictor.h5"
        self.encoder_path = "model/label_encoder.pkl"
        self.symptom_cols_path = "model/symptom_columns.pkl"
        self.symptom_translation_path = "model/symptom_translations.csv"
        self.disease_translation_path = "model/disease_translations.csv"
        self.treatments_path = "model/treatments.csv"
        self.num_symptoms = 377

        # --- Window setup ---
        self.title("NeurologyPredictor")
        self.geometry("1200x700")

        # Load model & label encoder
        self.model = load_model(self.model_path)
        self.label_encoder = joblib.load(self.encoder_path)

        # Load symptom list (unsorted - як ти просив)
        self.all_symptoms_en = joblib.load(self.symptom_cols_path)
```

Рисунок 4.20 – Ініціалізація основного класу

Додатково виконується завантаження словників перекладів (`symptom_translations.csv`, `disease_translations.csv`) та бази даних лікувальних рекомендацій (`treatments.csv`). Для цього використовується спеціалізований метод `safe_load_translation`. Цей безпечний підхід гарантує стабільну роботу системи навіть у випадку відсутності будь-якого з файлів перекладів: у такій ситуації повертаються порожні словники, і система продовжує працювати,

використовуючи англomовні назви замість перекладених. Ця архітектурна особливість є критично важливою для забезпечення надійності та відмовостійкості системи в різних конфігураціях розгортання.

Етап 2: Налаштування мультимовного інтерфейсу (рис. 4.21)

Алгоритм підтримує двомовний режим роботи (англійська та українська мови) з динамічним перемиканням інтерфейсу. Функція *switch_language* забезпечує повну локалізацію інтерфейсу, включаючи назви елементів управління, заголовки та підказки користувачу.

```
# ----- UI methods -----
def switch_language(self, event=None):
    selected = self.lang_var.get()
    if selected == "English":
        self.language = 'en'
        self.current_symptoms = self.all_symptoms_en.copy()
        # reset vars but keep previous selections cleared
        self.symptom_vars = {s: tk.BooleanVar(value=False) for s in self.current_symptoms}
        # update frame titles
        self.symptom_frame.config(text="Select Symptoms")
        self.manual_frame.config(text="Or enter symptoms manually (comma-separated)")
        self.clear_selected_btn.config(text="Clear Selected")
        self.predict_btn.config(text="Predict")
        self.output_frame.config(text="Prediction Results")
        self.treat_frame.config(text="Treatment Recommendations")
        self.selected_pills_frame.config(text="Selected Symptoms")
    else:
        self.language = 'ukr'
        # translate symptoms to Ukrainian where possible (keep original if missing)
        self.current_symptoms = [self.symptom_translations['en_to_ukr'].get(s, s) for s in self.all_symptoms_en]
        self.symptom_vars = {s: tk.BooleanVar(value=False) for s in self.current_symptoms}
        # update titles
        self.symptom_frame.config(text="Виберіть симптоми")
        self.manual_frame.config(text="Або введіть симптоми вручну (через кому)")
        self.clear_selected_btn.config(text="Очистити вибране")
        self.manual_clear_btn.config(text="Очистити введене вручну")
        self.predict_btn.config(text="Прогнозувати")
        self.output_frame.config(text="Результати прогнозу")
        self.treat_frame.config(text="Рекомендації до лікування")
        self.selected_pills_frame.config(text="Вибрані симптоми")
```

Рисунок 4.21 – Функція локалізації інтерфейсу

Етап 3: Введення та обробка симптомів (рис. 4.22)

Система надає користувачу гнучкий подвійний механізм введення симптомів. Лікар може або вибрати ознаки зі списку інтерактивних елементів (чекбоксів), або ввести їх у текстове поле для ручного введення.

Ключову роль на цьому етапі виконує функція `update_symptom_list`, яка відповідає за динамічне оновлення відображення доступних симптомів у прокручуваному вікні інтерфейсу. Перед кожним оновленням ця функція очищує попередньо відображені елементи. Якщо користувач вводить пошуковий запит, який зберігається у змінній `search_text`, активується алгоритм нечіткого пошуку, що реалізований за допомогою бібліотеки `difflib`. Цей алгоритм, викликаний з параметрами, які встановлюють максимальну кількість результатів на рівні 200 та мінімальний поріг схожості на рівні 0,3, порівнює введений текст із повним списком стандартизованих симптомів. Це дозволяє системі знаходити релевантні симптоми навіть при неточному введенні, що суттєво підвищує ергономіку використання у клінічному середовищі.

```
def update_symptom_list(self):
    for widget in self.scroll_frame.wininfo_children():
        widget.destroy()

    search_text = self.search_var.get().strip().lower()
    if search_text:
        matches = difflib.get_close_matches(search_text, self.current_symptoms, n=200, cutoff=0.3)
        filtered = matches
    else:
        filtered = self.current_symptoms

    if not filtered:
        no_text = "No symptoms found" if self.language == 'en' else "Симптоми не знайдені"
        self.no_results_label.config(text=no_text)
        return
    else:
        self.no_results_label.config(text="")

    for s in filtered:
        cb = ttk.Checkbutton(self.scroll_frame, text=s,
                             variable=self.symptom_vars[s],
                             command=self.update_selected_pills)
        cb.pack(anchor="w")

    self.update_selected_pills()
```

Рисунок 4.22 – Функція обробки введених симптомів

Результати пошуку, що зберігаються у змінній `filtered`, проходять перевірку на наявність збігів. Якщо збіги відсутні, відображається відповідне повідомлення про те, що симптоми не знайдені. У протилежному випадку, для кожного знайденого релевантного симптому створюється новий елемент керування у вигляді прапорця (чекбокса). Ці елементи пов'язуються зі спеціальними логічними змінними, які фіксують стан вибору симптому. Кожна дія користувача запускає команду оновлення `update_selected_pills`, яка динамічно змінює візуальне відображення обраних симптомів в окремій області інтерфейсу. Таким чином, цей етап забезпечує точне та гнучке формування вхідного вектора ознак для подальшої обробки моделлю.

Етап 4: Підготовка даних для моделі (рис. 4.23).

Коли користувач ініціює процес прогнозування натисканням відповідної кнопки, система активує функцію `predict` для збору всіх клінічних даних та їх переведення у формат, придатний для обробки нейронною мережею. Цей процес починається зі збору всіх вибраних симптомів з обох джерел введення. Спершу система агрегує симптоми, обрані за допомогою інтерактивних прапорців (чекбоксів), які зберігаються у змінній `symptom_vars`. Потім обробляється ручний ввід (змінна `manual_input`). Якщо ручний ввід присутній, він спочатку нормалізується – рядок розбивається на окремі симптоми за допомогою роздільника-коми та очищується від зайвих пробілів. Ці вручну введені симптоми додаються до загального списку зібраних ознак.

Після агрегації виконується важливий етап валідації: система перевіряє, чи було взагалі введено або обрано хоча б один симптом. Якщо список симптомів порожній (`if not symptoms`), програма відображає у вікні результатів відповідне повідомлення про необхідність введення даних, призупиняючи процес прогнозування. Це запобігає спробам подати на вхід моделі порожній вектор, що є нерелевантним для діагностики.

Наступним кроком є уніфікація мовного формату. Якщо користувач працює в українській локалізації (`if self.language == 'ukr'`), система здійснює

трансляцію усіх зібраних симптомів в англійські назви (`symptoms_en`). Ця конвертація є критично важливою для забезпечення коректної роботи моделі, оскільки нейронна мережа була навчена виключно на англійських термінах із датасету Kaggle. Для перекладу використовується внутрішній словник `symptom_translations`, при цьому реалізовано механізм повернення оригінальної (української) назви, якщо англійський еквівалент не знайдено, що підвищує відмовостійкість системи. У результаті цього етапу отримується уніфікований англійський список симптомів, готовий до фінального перетворення на числовий вектор.

```
def predict(self):
    # collect symptoms
    symptoms = [s for s, v in self.symptom_vars.items() if v.get()]
    manual_input = self.manual_text.get("1.0", tk.END).strip()
    if manual_input:
        manual_list = [x.strip() for x in manual_input.split(",") if x.strip()]
        symptoms += manual_list

    if not symptoms:
        msg = "Please select or enter symptoms.\n" if self.language == 'en' else "Будь ласка, виберіть або введіть симптоми.\n"
        self.output_text.config(state=tk.NORMAL)
        self.output_text.delete("1.0", tk.END)
        self.output_text.insert(tk.END, msg)
        self.output_text.config(state=tk.DISABLED)
        return

    # map to english symptom names if current language is ukr
    if self.language == 'ukr':
        symptoms_en = [self.symptom_translations['ukr_to_en'].get(s, s) for s in symptoms]
    else:
        symptoms_en = symptoms
```

Рисунок 4.23 – Функція процесу прогнозування

Етап 5: Створення вектора ознак (рис. 4.24)

Ключовим етапом переходу від текстових клінічних даних до числових, які може обробляти нейронна мережа, є створення вектора ознак. Цей числовий вектор має фіксований розмір (377 елементів), що строго відповідає вхідному формату, на якому тренувалася модель. Система використовує принцип «one-hot encoding» (бінарне кодування).

Процес починається з ініціалізації порожнього вектора (змінна `input_vector`) відповідного розміру, заповненого нулями. Далі відбувається ітеративна перевірка: кожен уніфікований англomовний симптом (зі списку `symptoms_en`) послідовно порівнюється з повним стандартизованим списком симптомів (`all_symptoms_en`). Якщо симптом знайдено, його позиційний індекс (змінна `idx`) у цьому списку використовується для встановлення відповідної позиції у векторі `input_vector` на значення 1.0. Усі інші позиції залишаються нулями, що відображає відсутність відповідних симптомів.

```
# build one-hot vector
input_vector = np.zeros((1, self.num_symptoms), dtype=np.float32)
for symptom in symptoms_en:
    if symptom in self.all_symptoms_en:
        idx = self.all_symptoms_en.index(symptom)
        input_vector[0, idx] = 1.0
```

Рисунок 4.24 – Побудова числового вектора

Цей підхід забезпечує стандартизований вхід для нейронної мережі, оскільки гарантує, що вектор завжди має очікувану довжину незалежно від фактичної кількості або порядку введених симптомів користувачем. Такий механізм є критично важливим для стабільної роботи моделі машинного навчання, оскільки дозволяє коректно подавати дані на вхідний шар нейронної мережі для отримання точного прогнозу.

Етап 6: Прогнозування захворювань (рис. 4.25)

Після успішного формування та валідації вектору ознак, він подається на вхід натренованої нейронної мережі для прогнозування. Це є центральним етапом роботи СППР. Модель, яка зберігається як `self.model`, обробляє вхідний вектор і повертає вектор ймовірностей (змінна `preds`) розміром 66, де кожне значення відповідає імовірності належності пацієнта до одного з неврологічних діагнозів.

Наступним кроком є відбір найбільш релевантних результатів. Система відбирає Топ-10 найбільш імовірних діагнозів ($\text{top_k} = \min(10, \text{len}(\text{preds}))$). Це робиться шляхом сортування вектора ймовірностей у порядку спадання та вибору індексів, що відповідають найвищим значенням (змінна `top_idx`). Такий підхід гарантує, що лікар отримує не один, а цілий диференційний ряд можливих діагнозів, що є критично важливим для клінічної практики.

Отримані індекси діагнозів (числові мітки) повинні бути декодовані у зрозумілі назви. Спочатку відбувається зворотне перетворення числових міток на їхні оригінальні англійські назви (змінна `en_names`) за допомогою збереженого кодувальника міток. Далі, якщо інтерфейс працює в українській локалізації, система використовує внутрішній словник `disease_translations` для перекладу англійських назв на українську (змінна `display_names`). Якщо ж встановлено англійську мову, відображаються оригінальні назви. Це фінальний етап перед візуалізацією, який готує ранжований, зрозумілий та локалізований список діагнозів для відображення у графічному інтерфейсі.

```
# predict
preds = self.model.predict(input_vector, verbose=0)[0]
top_k = min(10, len(preds))
top_idx = preds.argsort()[-top_k:][::-1]
top_probs = preds[top_idx]

# prepare lists of english & display names
en_names = [self.label_encoder.inverse_transform([int(i))][0] for i in top_idx]
if self.language == 'ukr':
    display_names = [self.disease_translations['en_to_ukr'].get(en, en) for en in en_names]
else:
    display_names = en_names.copy()
```

Рисунок 4.25 – Вектор ймовірностей повернутий нейромережею

Етап 7: Візуалізація результатів прогнозування (рис. 4.26)

Результати прогнозування, представлені у вигляді ранжованого списку Топ-10 діагнозів та їхніх імовірностей, відображаються у текстовому вікні інтерфейсу в структурованому вигляді з кольоровим кодуванням. Візуалізація починається з визначення відповідного стильового тегу (`prob_tag`) за

допомогою допоміжної функції `color_for_prob`. Ця функція класифікує ймовірність за трьома порогами: червоний колір (стиль `prob_low`) застосовується для низьких значень ($< 30\%$), жовтий (стиль `prob_medium`) – для середніх значень (30-70%), і зелений (стиль `prob_high`) – для високих значень ($> 70\%$). Таке кольорове маркування дозволяє лікарю миттєво оцінити ступінь впевненості моделі у прогнозі.

Для кожного діагнозу зі списку, який вже містить локалізовану назву (`disp_name`) та оригінальну англійську назву (`en_name`), виконується двоетапне вставлення у вікно результатів. Спочатку вставляється назва діагнозу, до якої застосовується унікальний тег (`tag_name`) та стилі форматування (підкреслення, синій колір). Потім поряд вставляється обчислена ймовірність, форматована у відсотках, до якої застосовується кольоровий тег, визначений функцією `color_for_prob`.

```
def color_for_prob(self, prob):
    if prob < 0.3:
        return "prob_low"
    elif prob < 0.7:
        return "prob_medium"
    else:
        return "prob_high"

# display results with color coding
for i, (en_name, disp_name, prob) in enumerate(zip(en_names, display_names, top_probs)):
    tag_name = f"disease_tag_{i}"
    self.tag_to_en[tag_name] = en_name

    disp_fixed = disp_name.ljust(max_len + 3)
    self.output_text.insert(tk.END, disp_fixed, (tag_name, "disease_bold"))

    prob_tag = self.color_for_prob(prob)
    self.output_text.insert(tk.END, f"{prob*100:6.2f} %\n", prob_tag)
```

Рисунок 4.26 – Відображення результатів прогнозування

Ключовою особливістю цього етапу є реалізація інтерактивності. Кожен діагноз отримує набір обробників подій Tkinter (tag_bind): клік лівою кнопкою миші (<Button-1>) запускає функцію відображення лікувальних рекомендацій. Наведення курсора на назву діагнозу (<Enter>) змінює вигляд курсора на "руку" ("hand2"), що візуально підкреслює його клікабельність. Таким чином, цей етап забезпечує не лише наочне, але й функціональне представлення результатів, що є прямим кроком до надання інтерактивної підтримки клінічних рішень.

Етап 8: Генерація лікувальних рекомендацій (рис. 4.27)

При виборі конкретного діагнозу система активує алгоритм пошуку відповідних лікувальних рекомендацій. Функція show_treatment здійснює пошук в базі даних лікувальних протоколів спочатку за точною відповідністю, а при її відсутності - використовує алгоритм нечіткого пошуку.

```
def _find_treatment_row(self, en_name):
    """Find row in treatments_df matching en_name (case-insensitive). Try fuzzy fallback."""
    df = self.treatments_df
    # normalize
    mask = df['English'].str.lower() == en_name.lower()
    if mask.any():
        return df.loc[mask].iloc[0]
    # try fuzzy match on English column
    candidates = df['English'].dropna().tolist()
    match = difflib.get_close_matches(en_name, candidates, n=1, cutoff=0.8)
    if match:
        return df[df['English'] == match[0]].iloc[0]
    return None
```

Рисунок 4.27 – Функція пошуку рекомендацій

Етап 9: Структурування та відображення рекомендацій (рис. 4.28)

Коли користувач натискає на інтерактивну назву діагнозу в результатах прогнозування, активується функція show_treatment. Ця функція відповідає за

пошук і відображення відповідних клінічних рекомендацій у правому блоці інтерфейсу.

Процес починається з виклику допоміжного методу `_find_treatment_row`, який здійснює пошук у внутрішній базі протоколів за англійською назвою діагнозу (`en_name`). Якщо відповідний рядок даних (`row`) із рекомендаціями не знайдено (навіть після спроби нечіткого пошуку), система виводить у панелі повідомлення про те, що «Інформація про лікування відсутня», і призупиняє подальші дії. Якщо ж дані знайдено, першим кроком є локалізація назви діагнозу (`disease_name`), яка відображається у заголовку панелі. Далі відбувається визначення мови рекомендацій: якщо інтерфейс встановлено на українську мову і є доступний перекладений список рекомендацій (`t_ukr_list`), він обирається для відображення (`chosen`). Якщо український переклад відсутній, але є англійський список (`t_en_list`), обирається він. Це забезпечує максимальну доступність клінічних знань.

```
def show_treatment(self, en_name):
    # --- Рекомендації ---
    t_en_list, t_ukr_list = self._get_treatments_for(en_name)
    if self.language == 'ukr' and t_ukr_list:
        | chosen = t_ukr_list
    elif t_en_list:
        | chosen = t_en_list
    else:
        | chosen = []

    # --- Формування вікна ---
    self.treat_text.config(state=tk.NORMAL)
    self.treat_text.delete("1.0", tk.END)

    if not chosen:
        | msg = "No treatment info available." if self.language == 'en' else "Інформація про лікування відсутня."
        | self.treat_text.insert(tk.END, msg + "\n")
    else:
        | for t in chosen:
        | | self.treat_text.insert(tk.END, f"• {t.strip()}\n\n")
```

Рисунок 4.28 – Функція відображення рекомендацій

Етап 10: Управління станом системи (рис. 4.29)

Система включає механізми очищення даних та скидання стану, що дозволяє користувачу швидко починати новий сеанс діагностики. Функція `clear_selected` комплексно очищає всі області введення та виведення, повертаючи систему до початкового стану.

```
def clear_selected(self):
    for var in self.symptom_vars.values():
        var.set(False)
    self.manual_text.delete("1.0", tk.END)
    self.update_selected_pills()
    self.output_text.config(state=tk.NORMAL)
    self.output_text.delete("1.0", tk.END)
    self.output_text.config(state=tk.DISABLED)
    self.treat_text.config(state=tk.NORMAL)
    self.treat_text.delete("1.0", tk.END)
    self.treat_text.config(state=tk.DISABLED)
    self.tag_to_en.clear()
```

Рисунок 4.29 – Функція скидання стану програми

Висновки до розділу 4

У четвертому розділі було представлено програмну реалізацію розробленої СППР та продемонстровано її практичне функціонування. Основна увага була приділена побудові графічного інтерфейсу користувача на основі бібліотеки Tkinter, що забезпечує інтуїтивну та зручну взаємодію з програмою. Реалізовані віджети дозволяють легко вводити симптоми, отримувати прогноз діагнозу та ознайомлюватися з результатами у наочному форматі. Такий підхід робить систему доступною не лише для фахівців, але й для практичних клінічних сценаріїв.

Окремо було описано процес попередньої обробки даних і інтеграції навченої нейронної мережі у програмне середовище. Модель завантажується без потреби у повторному тренуванні, що спрощує використання системи в

умовах реального часу. Додатково реалізовано збереження енкодера та списку симптомів, що забезпечує узгодженість між структурою вхідних даних і прогнозами моделі.

У розділі також наведено приклади застосування системи у різних клінічних сценаріях. Було показано, що навіть у випадках неоднозначних симптомів СППР здатна пропонувати лікарю декілька можливих варіантів діагнозів, відсортованих за ймовірністю. Такий підхід дає можливість лікарю швидше орієнтуватися у складних ситуаціях та приймати більш обґрунтовані рішення.

Таким чином, реалізована програмна система підтвердила свою здатність поєднувати алгоритмічну точність нейронної мережі з практичною зручністю інтерактивного інтерфейсу. Вона демонструє потенціал до використання як у навчальних цілях, так і у реальних клінічних умовах, а також може стати основою для подальшого розширення, інтеграції з електронними медичними записами та впровадження у телемедичні сервіси.

РОЗДІЛ 5

РОЗРОБКА СТАРТАП-ПРОЄКТУ

У розділі буде розглянуто ключові аспекти створення та запуску інноваційного проєкту у сфері медичних технологій. Спочатку буде представлено концепцію стартапу, його місію, основну ідею та потенційний вплив на медичну галузь. Особлива увага приділятиметься етапам реалізації проєкту - від розробки програмного забезпечення та формування команди до створення мінімально життєздатного продукту (MVP) і виходу на ринок.

Далі буде детально розраховано витрати на організацію стартапу, включаючи розробку, маркетинг, інфраструктуру, оплату праці фахівців і резервні кошти. Буде проведено економічне обґрунтування інвестицій та наведено прогноз фінансової ефективності, що дозволяє оцінити доцільність впровадження проєкту. Наприкінці розділу розглядатимуться можливості масштабування, потенційні джерела фінансування (венчурні інвестиції, гранти, співпраця з медичними установами) та перспективи виходу продукту на міжнародний ринок.

5.1. Назва проєкту

Система підтримки прийняття рішень для лікаря-невролога "NeurologyPredictor".

5.2. Короткий опис проєкту

Проєкт створення інтелектуальної СППР для підтримки прийняття рішень лікарем-неврологом має на меті підвищення точності діагностування, оптимізацію вибору лікування та зменшення часу на обробку медичної інформації. СППР "NeurologyPredictor" аналізує клінічні дані пацієнта, включаючи анамнез, симптоми та результати обстежень, формує попередні

діагнози на основі виявленої симптоматики, а також надає рекомендації щодо додаткових обстежень і лікувальних стратегій.

Система відповідає потребам лікарів, які очікують ефективного інструмента для виявлення навіть складних і рідкісних неврологічних захворювань. Вона сприяє підвищенню ефективності клінічного мислення, знижуючи навантаження на фахівця та забезпечуючи впевненість у якості діагнозу й наданих рекомендацій, спираючись на базу знань і ШІ.

Унікальність рішення полягає в поєднанні алгоритмів машинного навчання з експертними правилами, сформованими на основі протоколів доказової медицини. Система є адаптивною до змін у медичних стандартах та здатна до інтеграції з існуючими електронними медичними системами.

5.3. Бізнес-модель

5.3.1. Цінність продукту

СППР "NeurologyPredictor" пропонує низку важливих переваг, які вирізняють її серед наявних рішень:

- Підтримка широкого спектру неврологічних захворювань, зокрема складних для діагностики (нейродегенеративні хвороби, епілепсія, розсіяний склероз).
- Інтеграція симптомів, лабораторних і візуалізаційних даних (МРТ, КТ, ЕЕГ) для комплексного аналізу.
- Попередній діагноз і персоналізовані рекомендації щодо лікування, засновані на алгоритмах штучного інтелекту.
- Гнучка база знань, що оновлюється згідно з новими клінічними протоколами.
- Зручний інтерфейс для лікарів, незалежно від рівня цифрової підготовки.
- Інтеграція з електронними медичними системами, що спрощує обмін інформацією.

- Зниження ризику помилок та підвищення точності діагностики, що покращує результати лікування.

Виконує вимоги:

- «Надії» лікаря на точне рішення в умовах невизначеності.
- «Бажання» мати ефективного помічника, який зменшує навантаження.
- «Хоче бути» впевненим у своїх діях, маючи доказову підтримку рішень.
- «Очікує» персоналізованого підходу, з урахуванням індивідуальних факторів пацієнта.

Унікальність системи полягає у поєднанні:

- алгоритмів ШІ для обробки медичних даних;
- логіки клінічних правил для прозорості рішень;
- здатності працювати в автономному режимі, що важливо для регіональних лікарень.

5.3.2. Сегмент споживачів

Цільова аудиторія "NeurologyPredictor" включає:

- Лікарі-неврологи - система допомагає в діагностиці складних неврологічних станів, пришвидшує аналіз симптомів і пропонує клінічно обґрунтовані рішення.
- Сімейні лікарі - отримують підтримку при первинній діагностиці неврологічних симптомів, що дозволяє швидше приймати рішення про подальше лікування або направлення.
- Медичні заклади з обмеженими ресурсами - забезпечують пацієнтам якісну допомогу навіть без присутності вузьких спеціалістів.
- Медичні навчальні заклади - використовують систему як інструмент для навчання студентів неврології на основі реальних клінічних сценаріїв.

Усі ці групи об'єднує потреба в точній, швидкій і доступній підтримці рішень у сфері неврології.

5.3.3. Канали збуту

- Мобільні маркетплейси: Додаток буде доступний для завантаження через Google Play та App Store, що забезпечить легкий доступ для медичних працівників.
- Офіційний сайт: Через вебсайт можна ознайомитись з функціоналом системи, отримати підтримку та оформити підписку.
- Партнерство з клініками: Співпраця з лікарнями та медцентрами для інтеграції системи у клінічну практику.
- Пряма продаж: Ліцензії або підписки на використання будуть продаватися безпосередньо закладам охорони здоров'я.
- Участь у медичних заходах: Презентації на виставках, конференціях та семінарах для залучення цільової аудиторії.

5.3.4. Взаємодія з споживачами

ЗАЛУЧЕННЯ:

Для залучення користувачів нашої СППР передбачено рекламу в соціальних мережах, профільних медичних спільнотах та участь у спеціалізованих заходах. Планується співпраця з медичними університетами та лікарнями. Також впроваджуються реферальні програми, безкоштовні пробні періоди та публікації з прикладами застосування системи в клінічній практиці.

ПІДТРИМКА:

Користувачам буде доступна підтримка через офіційний сайт, де розміщено інструкції, відео та форма звернення до служби підтримки. Додатково працюватимуть онлайн-форум та спільнота для обміну досвідом. Зворотній зв'язок збиратиметься через опитування в додатку, що дозволить адаптувати продукт до потреб лікарів.

5.3.5. Дохід

Основним джерелом доходу нашої СППР для лікаря-невролога буде продаж ліцензій на використання програмного забезпечення. Передбачені два варіанти: одноразова покупка та підписна модель, яка передбачає щомісячну або щорічну оплату.

Додатковий дохід приносить доступ до хмарної версії системи за моделлю SaaS (Software as a Service). Також передбачається отримання прибутку від послуг з інтеграції системи в ІТ-інфраструктуру медичних закладів, індивідуального налаштування, технічної підтримки та навчання персоналу. У перспективі можливе запровадження платних модулів із розширеним функціоналом.

5.3.6. Ключові види діяльності

Ключовими видами діяльності стартапу є розробка та вдосконалення СППР для лікаря-невролога на базі штучного інтелекту. Особлива увага приділяється точності діагностичних алгоритмів, зручності інтерфейсу та безперервному оновленню програмного забезпечення.

До важливих напрямів також належать інтеграція системи в ІТ-інфраструктуру медичних закладів, забезпечення технічної підтримки користувачів та проведення навчання для медичного персоналу. Маркетинг і популяризація продукту серед цільової аудиторії також становлять невід'ємну частину діяльності стартапу.

5.3.7. Ключові ресурси

Матеріальні ресурси:

Для забезпечення роботи СППР необхідно орендоване офісне приміщення з робочими місцями для команди розробників, доступом до Інтернету та необхідною комп'ютерною технікою. Додатково будуть закуплені ліцензії на програмне забезпечення для розробки, тестування та моделювання нейронних мереж і баз даних.

Інтелектуальні ресурси:

Проект використовуватиме власні технічні напрацювання, включно з алгоритмами штучного інтелекту, спеціалізованими базами медичних знань, документацією, а також експертними висновками та дослідженнями в галузі неврології.

Людські ресурси:

- Керівник проекту – спеціаліст з досвідом управління IT-проектами та знанням медичної галузі.
- Провідний розробник – фахівець зі штучного інтелекту, який відповідає за розробку ядра системи.
- Інженер з даних – обробляє, аналізує та структурує великі обсяги медичної інформації.
- UI/UX дизайнер – створює інтуїтивно зрозумілий інтерфейс системи для медичних працівників.
- Тестувальник (QA) – перевіряє систему на помилки, забезпечує її надійність.
- Медичний консультант (невролог) – здійснює експертний супровід та перевірку достовірності медичних рішень системи.
- Фахівець з технічної підтримки – допомагає користувачам у вирішенні технічних питань.

Фінансові ресурси:

- Початкове фінансування від власників стартапу або інвесторів.
- Гранти на розвиток медичних IT-технологій.
- Залучення додаткового капіталу через стартап-програми або бізнес-акселератори.

5.3.8. Ключові партнери

- Медичні заклади (лікарні, поліклініки, медцентри) для впровадження системи.
- Університети та кафедри неврології для наукової підтримки.

- Неврологи для тестування та розробки бази знань.
- Технологічні компанії для розробки ШІ та забезпечення інфраструктури.
- Фармацевтичні компанії для інтеграції медичних рекомендацій.
- Страхові компанії для зниження ризиків і покращення діагностики.

5.3.9. Структура витрат

Основні витрати стартапу формуються за такими категоріями:

1. Розробка програмного забезпечення (одноразові витрати на MVP):

$$C_{dev} = 180,000 \text{ грн} \quad (5.1)$$

Включає створення архітектури, програмування, тестування та розгортання першої версії СППР.

2. Зарплати команди (на рік):

$$C_{team} = C_{devs} + C_{med} + C_{mgmt} \quad (5.2)$$

$$\text{Розробники (2 особи} \times 40,000/\text{міс} \times 12 \text{ міс)} = 960,000 \quad (5.3)$$

$$\text{Медичні експерти (2 неврологи} \times 25,000/\text{міс} \times 12 \text{ міс)} = 600,000 \quad (5.4)$$

$$\begin{aligned} &\text{Маркетолог і проєктний менеджер} \\ &\text{(2 особи} \times 30,000/\text{міс} \times 12 \text{ міс)} = 720,000 \end{aligned} \quad (5.5)$$

$$C_{team} = 2,280,000 \text{ грн} \quad (5.6)$$

3. Технічна інфраструктура (сервери, хостинг, ліцензії, GPU-обчислення):

$$C_{infra} = 240,000 \text{грн} \quad (5.7)$$

($\approx 20,000$ /міс, включає оренду хмарних серверів з GPU, підтримку баз даних, сертифікати безпеки).

4. Маркетинг і просування (онлайн-кампанії, конференції, партнерства):

$$C_{mkt} = 360,000 \text{грн} \quad (5.8)$$

($\approx 30,000$ /міс для таргетованої реклами, PR, демонстраційних заходів).

Загальні витрати на рік:

$$\begin{aligned} C_{total} &= C_{team} + C_{infra} + C_{mkt} & (5.9) \\ &= 2,280,000 + 240,000 + 360,000 \\ C_{total} &= 2,880,000 \end{aligned}$$

5.3.10. Споживчі властивості товару

СППР для лікаря-невролога має наступні споживчі властивості:

- 1) Підвищена точність діагностики – завдяки використанню методів штучного інтелекту та експертних правил система допомагає лікарю точно і швидко виявити неврологічні розлади.
- 2) Інтерактивна підтримка прийняття рішень – система надає обґрунтовані рекомендації щодо діагностики та можливих напрямів лікування на основі введених симптомів і анамнезу.
- 3) Зручний інтерфейс – проста навігація дозволяє лікарю легко взаємодіяти з системою без додаткового навчання.
- 4) Економія часу лікаря – автоматизація аналізу даних та пропозиція рішень дозволяє скоротити час консультації.

5) Можливість адаптації під різні медичні установи – система може використовуватись як у приватних клініках, так і в державних лікарнях.

б) Захист персональних даних пацієнта – забезпечується відповідно до стандартів кібербезпеки та медичного конфіденційного збереження даних.

5.3.11. Дослідження ринку

1) На світовому та українському ринку СППР спеціалізованого спрямування для лікарів-неврологів АНАЛОГИ ВІДСУТНІ – це власне НОУ-ХАУ авторів проєкту.

2) Дослідження конкурентного оточення:

3) Конкурентне оточення продажу аналогічних програмних продуктів відсутнє, оскільки існуючі системи або мають загальнотерапевтичний характер, або не забезпечують повноцінної діагностичної підтримки у сфері неврології.

4) Маркетингова стратегія просування:

5) Новий інноваційний продукт на існуючий ринок медичного програмного забезпечення України.

5.3.12. Елементи фінансового плану

1. Опис бізнес-проєкту

Організація виробництва та реалізації СППР для лікаря-невролога у формі стартапу, що включає розробку та впровадження інноваційної платформи для діагностики та моніторингу неврологічних захворювань.

2. Опис програмного продукту

СППР для лікаря-невролога, що забезпечує підтримку прийняття рішень під час діагностики, обробки медичних даних та тестування гіпотез для виявлення неврологічних захворювань, таких як розсіяний склероз, епілепсія, інсульт тощо.

3. Маркетинг та продаж

Ринок медичних технологій активно розвивається. В Україні і Європі попит на системи для підтримки медичних рішень постійно зростає, проте частка таких рішень на ринку обмежена. В Україні ринок СППР для медичних спеціалістів перебуває на стадії розвитку.

Для успішного просування проєкту необхідно проводити маркетингові кампанії серед медичних установ та лікарів-неврологів, а також створити партнерські відносини з клініками та медичними установами.

4. Фінансовий план (для конкретного інвестора)

Витрати на організацію стартапу на період 12 місяців:

- Розробка програмного забезпечення (створення архітектури, модулів для аналізу даних, тестування MVP) – 180 тис. грн
- Організація робочої команди (6–7 спеціалістів: програмісти, медичні експерти, менеджер і маркетолог) – 2,28 млн грн
- Маркетинг та просування (онлайн-кампанії, участь у конференціях, співпраця з медичними установами) – 360 тис. грн
- Технічна інфраструктура (хмарні сервери з GPU, ліцензії, сертифікати безпеки) – 240 тис. грн

ВСЬОГО: \approx 2,88 млн. грн.

5.3.13. Резюме

Стартап зосереджений на розробці та впровадженні інноваційної СППР для лікаря-невролога, яка дозволить забезпечити точну та швидку діагностику неврологічних захворювань. Впровадження цієї системи дозволить лікарям отримувати додаткові інструменти для прийняття рішень, знижуючи час на постановку діагнозу та підвищуючи ефективність лікування пацієнтів.

Система спирається на новітні технології обробки медичних даних, аналізу зображень та підтримки прийняття рішень за допомогою штучного

інтелекту. Вона включатиме інтерфейс, який забезпечить лікарям зручність у використанні та максимальну точність результатів.

Проєкт орієнтований на впровадження цих технологій на ринку медичних послуг, оскільки потреба в сучасних медичних рішеннях зростає, а запропонована система може стати важливим інструментом для боротьби з неврологічними захворюваннями. Враховуючи розвиток галузі та високий попит на нові технології в медичному секторі, стартап має всі шанси стати лідером на ринку медичних інновацій.

Висновки до розділу 5

У п'ятому розділі було проведено економічне обґрунтування створення та впровадження СППР для лікаря-невролога, розглянуто структуру витрат, очікувані джерела доходу та перспективи масштабування стартапу.

Насамперед визначено ключові статті витрат, необхідні для запуску та підтримки роботи проєкту протягом першого року. Найбільша частка бюджету припадає на формування команди, що включає програмістів, аналітиків, медичних експертів і менеджерів. Це зумовлено високими вимогами до якості розробки та верифікації медичного програмного забезпечення. Значні витрати також пов'язані з організацією технічної інфраструктури (хмарні сервери, GPU-обчислення, ліцензії) та маркетингом, що забезпечує залучення користувачів і партнерів. Загальна сума річних витрат становить близько 2,88 млн грн.

Подальший аналіз показав, що впровадження такого програмного забезпечення може знизити витрати медичних установ на діагностику за рахунок автоматизації процесів і підвищення швидкості прийняття рішень.

ЗАГАЛЬНІ ВИСНОВКИ

Магістерська робота присвячена створенню СППР для лікаря-невролога, яка поєднує перевірені клінічні знання з методами сучасного штучного інтелекту та надає лікарю інструмент підтримки діагностичних рішень у реальному часі. Дослідження пройшло повний цикл - від аналізу проблеми й наявних аналогів до розробки архітектури, підготовки даних, навчання моделі, реалізації програмного забезпечення та економічного обґрунтування впровадження.

У роботі було окреслено ключові проблеми діагностики неврологічних захворювань: високу міжхворобну схожість симптомів, варіативність клінічних проявів, коморбідність, а також обмежений доступ до сучасних інструментів діагностики у деяких регіонах. Показано, що СППР можуть виступати ефективним інструментом підтримки лікаря, однак мають обмеження, пов'язані з інтерпретованістю результатів та складністю оновлення баз знань. Перспективним напрямом є поєднання експертних правил і алгоритмів глибинного навчання, здатних працювати з великими масивами даних.

Аналіз існуючих наукових і комерційних рішень показав, що сучасні системи вже демонструють високу точність у задачах розпізнавання зображень і структурованих медичних даних, проте залишаються виклики, пов'язані з їх прозорістю та стандартизацією. Найбільш ефективними виявляються гібридні підходи, де алгоритмічні моделі поєднуються з клінічними протоколами та правилами, а також хмарні та федеративні архітектури, які враховують вимоги безпеки й конфіденційності.

Було сформовано методичний і технічний каркас для побудови власної СППР. Зібрано та структуровано дані з відкритих джерел, міжнародних стандартів (NICE, NHS, NIH) і настанов МОЗ України. Із понад 700 діагнозів було відібрано 66 нозологій неврологічного профілю, що забезпечило вузьку спеціалізацію та практичну орієнтацію системи. Для побудови програмного

середовища обрано Python 3.11 із бібліотеками NumPy, pandas, TensorFlow/Keras та scikit-learn, що дозволило ефективно реалізувати архітектуру багатошарового перцептронну з активаціями ReLU та Softmax.

Особливу увагу приділено підготовці та аналізу даних. Було виконано розвідувальний аналіз, сформовано навчальну і тестову вибірки, реалізовано процедури кодування та збереження метаданих. Навчання моделі здійснювалося із застосуванням оптимізатора Adam та функції втрат sparse categorical cross-entropy. Оцінка за метриками accuracy, precision, recall і F1-score підтвердила високу ефективність класифікації (близько 95%). Окремо розроблено механізми захисту інформації та дотримання принципів privacy-by-design, що відповідає міжнародним вимогам щодо безпеки даних у медицині.

Програмна реалізація охоплює графічний інтерфейс користувача на основі Tkinter, що забезпечує простоту та інтуїтивність взаємодії. Було продемонстровано сценарії використання системи на прикладах, де вибір симптомів приводить до отримання впорядкованого списку можливих діагнозів. Це підтвердило практичну цінність інструмента для лікарів-неврологів і створює підґрунтя для інтеграції з клінічною практикою.

Економічний аналіз проєкту показав, що річні витрати на створення і підтримку системи становлять близько 2,88 млн грн. Основна частка припадає на команду розробників та медичних експертів, а також на інфраструктуру. Разом з тим, розробка має перспективи масштабування, інтеграції з електронними медичними записами та виходу на міжнародний ринок, що забезпечує фінансову доцільність стартапу у середньостроковій перспективі.

Узагальнюючи результати дослідження, можна стверджувати, що створена СППР поєднує наукову достовірність, технічну ефективність та практичну цінність. Вона дозволяє зменшити час діагностики, підвищує точність і стабільність клінічних рішень, а також відкриває шлях до розвитку нових інтелектуальних медичних інструментів у неврології та суміжних галузях.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Neurological Disorders Overview [Електронний ресурс]. Cleveland Clinic. Режим доступу : <https://my.clevelandclinic.org/health/diseases/neurological-disorders> (дата звернення: 09.09.2025).
2. Demyelinating Disease Overview [Електронний ресурс]. Cleveland Clinic. Режим доступу : <https://my.clevelandclinic.org/health/diseases/demyelinating-disease> (дата звернення: 09.09.2025).
3. Diagnostic Challenges in Neurology [Електронний ресурс]. National Institute of Neurological Disorders and Stroke. Режим доступу : <https://www.ninds.nih.gov/health-information/diagnostic-challenges-neurology> (дата звернення: 09.09.2025).
4. Jackson P. Introduction to Expert Systems [Електронний ресурс]. 3rd ed. Harlow : Addison Wesley, 1998. ISBN 978-0201876864. Режим доступу : https://books.google.com/books/about/Introduction_to_Expert_Systems.html?id=9rJQAAAAMAAJ (дата звернення: 09.09.2025).
5. Shortliffe E.H., Cimino J.J. Biomedical Informatics: Computer Applications in Health Care and Biomedicine [Електронний ресурс]. Springer, 2014. Режим доступу : https://books.google.com/books/about/Biomedical_Informatics.html?id=nim5BAAAQBAJ (дата звернення: 09.09.2025).
6. Giarratano J.C., Riley G.D. Expert Systems: Principles and Programming [Електронний ресурс]. 4th ed. Boston : Course Technology, 2005. ISBN 978-0534384470. Режим доступу : <https://books.google.com.pe/books?id=SbgZRAAACA AJ> (дата звернення: 09.09.2025).
7. Shortliffe E.H. Computer-Based Medical Consultations: MYCIN [Електронний ресурс]. New York : Elsevier, 1976. Режим доступу :

https://www.acpjournals.org/doi/10.7326/0003-4819-85-6-831_1 (дата звернення: 09.09.2025).

8. Miller R.A., Pople H.E., Myers J.D. INTERNIST-1, an experimental computer-based diagnostic consultant for general internal medicine [Електронний ресурс] // New England Journal of Medicine. 1974. Vol. 307. P. 468–476. Режим доступу : <https://en.wikipedia.org/wiki/Internist-I> (дата звернення: 09.09.2025).

9. Berner E.S. Clinical Decision Support Systems: Theory and Practice [Електронний ресурс]. 3rd ed. New York : Springer, 2016. Режим доступу : <https://link.springer.com/book/10.1007/978-3-319-31913-1> (дата звернення: 09.09.2025).

10. Durkin J. Expert Systems: Design and Development [Електронний ресурс]. New York : Macmillan, 1994. Режим доступу : <https://www.semanticscholar.org/paper/Expert-Systems%3A-Design-and-Development-Durkin-Durkin/6f102e94a05fc00af833d7d8ce4fa63845439ec9> (дата звернення: 09.09.2025).

11. Berner E.S., Moss J. Computer-based Decision Support Systems in Health Care and Their Integration into Clinical Workflow [Електронний ресурс] // Journal of the American Medical Informatics Association. 2005. Vol. 12, № 6. P. 515–520. Режим доступу : <https://pubmed.ncbi.nlm.nih.gov/10939999/> (дата звернення: 09.09.2025).

12. Coiera E. Guide to Health Informatics [Електронний ресурс]. Boca Raton : CRC Press, 2015. Режим доступу : <https://www.taylorfrancis.com/books/mono/10.1201/b13617/guide-health-informatics-enrico-coiera> (дата звернення: 09.09.2025).

13. Buchanan B.G., Shortliffe E.H. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project [Електронний ресурс]. Reading, MA : Addison-Wesley, 1984. Режим доступу : <https://dl.acm.org/doi/book/10.5555/318721> (дата звернення: 09.09.2025).

14. Shortliffe E.H. Computer-Based Medical Consultations: MYCIN [Електронний ресурс]. New York : Elsevier, 1976. Режим доступу :

https://www.acpjournals.org/doi/10.7326/0003-4819-85-6-831_1 (дата звернення: 09.09.2025).

15. IBM. Watson for Oncology: Transforming Cancer Care with AI [Електронний ресурс]. IBM Watson Health, 2020. Режим доступу : <https://www.ibm.com/watson-health/oncology> (дата звернення: 09.09.2025).

16. Barnett G.O., Cimino J.J., Hupp J.A., Hoffer E.P. DXplain – A Computer-Based Decision Support System for Diagnosing [Електронний ресурс] // New England Journal of Medicine. 1987. Режим доступу : <https://pubmed.ncbi.nlm.nih.gov/3295316/> (дата звернення: 09.09.2025).

17. Esteva A., Robicquet A., Ramsundar B. A guide to deep learning in healthcare [Електронний ресурс] // Nature Medicine. 2019. Vol. 25, iss. 1. P. 24–29. Режим доступу : <https://www.nature.com/articles/s41591-018-0316-z> (дата звернення: 09.09.2025).

18. Shortliffe E.H. Computer-Based Medical Consultations: MYCIN [Електронний ресурс]. Elsevier, 1976. Режим доступу : https://www.acpjournals.org/doi/10.7326/0003-4819-85-6-831_1 (дата звернення: 09.09.2025).

19. Litjens G. et al. A survey on deep learning in medical image analysis [Електронний ресурс] // Medical Image Analysis. 2017. Vol. 42. P. 60–88. Режим доступу : <https://doi.org/10.1016/j.media.2017.07.005> (дата звернення: 09.09.2025).

20. Krittanawong C. et al. Deep learning for cardiovascular medicine [Електронний ресурс] // Nature Reviews Cardiology. 2021. Режим доступу : <https://www.nature.com/articles/s41569-020-00481-9> (дата звернення: 09.09.2025).

21. Sheller M.J. et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data [Електронний ресурс] // Scientific Reports. 2020. Vol. 10. Article number 11394. Режим доступу : <https://www.nature.com/articles/s41598-020-68981-1> (дата звернення: 09.09.2025).

22. Vaswani A. et al. Attention Is All You Need [Електронний

- ресурс]. 2017. (arXiv:1706.03762). Режим доступа : <https://arxiv.org/abs/1706.03762> (дата звернення: 09.09.2025).
23. Devlin J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Електронний ресурс] // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). 2019. Режим доступа : <https://aclanthology.org/N19-1423/> (дата звернення: 09.09.2025).
24. Singhal K. et al. Large Language Models Encode Clinical Knowledge [Електронний ресурс] // Nature. 2023. Vol. 620, iss. 7972. P. 172–180. Режим доступа : <https://pubmed.ncbi.nlm.nih.gov/37438534/> (дата звернення: 09.09.2025).
25. International Classification of Diseases 11th Revision (ICD-11) [Електронний ресурс] / World Health Organization. Geneva, 2019. Режим доступа : <https://www.who.int/standards/classifications/classification-of-diseases> (дата звернення: 09.09.2025).
26. Johnson A.E.W. et al. MIMIC-III, a freely accessible critical care database [Електронний ресурс] // Scientific Data. 2016. Vol. 3. Article number 160035. Режим доступа : <https://dspace.mit.edu/handle/1721.1/109192> (дата звернення: 09.09.2025).
27. Samek W. et al. Explainable AI: interpreting, explaining and visualizing deep learning [Електронний ресурс]. Springer, 2019. Режим доступа : <https://www.springerprofessional.de/explainable-ai-interpreting-explaining-and-visualizing-deep-lear/17153806> (дата звернення: 09.09.2025).
28. Kairouz P. et al. Advances and open problems in federated learning [Електронний ресурс] // Foundations and Trends in Machine Learning. 2021. Vol. 14, № 1–2. P. 1–210. Режим доступа : <https://www.researchwithrutgers.com/en/publications/advances-and-open-problems-in-federated-learning> (дата звернення: 09.09.2025).
29. Litjens G. et al. A survey on deep learning in medical image analysis [Електронний ресурс] // Medical Image Analysis. 2017. Vol. 42. P. 60–88. Режим

доступу : <https://doi.org/10.1016/j.media.2017.07.005> (дата звернення: 09.09.2025).

30. Samek W. et al. Explainable AI: interpreting, explaining and visualizing deep learning [Електронний ресурс]. Springer, 2019. Режим доступу : <https://www.springerprofessional.de/explainable-ai-interpreting-explaining-and-visualizing-deep-lear/17153806> (дата звернення: 09.09.2025).

31. Collins F.S., Varmus H. A new initiative on precision medicine [Електронний ресурс] // New England Journal of Medicine. 2015. Vol. 372, № 9. P. 793–795. Режим доступу : <https://pubmed.ncbi.nlm.nih.gov/25635347/> (дата звернення: 09.09.2025).

32. Kohn M.S. et al. IBM's Health Analytics and Clinical Decision Support [Електронний ресурс]. 2014. Режим доступу : <https://pmc.ncbi.nlm.nih.gov/articles/PMC4287097/> (дата звернення: 09.09.2025).

33. Davenport T., Kalakota R. The potential for artificial intelligence in healthcare [Електронний ресурс] // Future Healthcare Journal. 2019. Vol. 6, № 2. P. 94–98. Режим доступу : <https://doi.org/10.7861/futurehosp.6-2-94> (дата звернення: 09.09.2025).

34. Clinical Guidelines [Електронний ресурс]. National Institute for Health and Care Excellence (NICE). London, 2020. Режим доступу : <https://www.ncbi.nlm.nih.gov/books/NBK11822/> (дата звернення: 09.09.2025).

35. Національні клінічні протоколи та статистичні бази даних [Електронний ресурс] / Міністерство охорони здоров'я України. Київ, 2025. Режим доступу : <https://moz.gov.ua/> (дата звернення: 09.09.2025).

36. Kaggle. Terms of Use [Електронний ресурс]. Режим доступу : <https://www.kaggle.com/terms> (дата звернення: 09.09.2025).

37. Top 10 Coding Languages to Learn for AI Development in 2025 [Електронний ресурс] / Analytics Insight. 2025. Режим доступу : <https://www.analyticsinsight.net/data-science/top-10-coding-languages-to-learn-for-ai-development-in-2025> (дата звернення: 09.09.2025).

38. Harris C.R. et al. Array programming with NumPy [Электронный ресурс]. 2022. Режим доступа : <https://numpy.org/> (дата звернения: 09.09.2025).
39. McKinney W. pandas: a Python library for data analysis and manipulation [Электронный ресурс]. 2024. Режим доступа : <https://pandas.pydata.org/> (дата звернения: 09.09.2025).
40. TensorFlow: An Open-Source Machine Learning Framework for Everyone [Электронный ресурс] / Google. 2024. Режим доступа : <https://www.tensorflow.org/> (дата звернения: 09.09.2025).
41. Pedregosa F. et al. Scikit-learn: Machine Learning in Python [Электронный ресурс]. 2022. Режим доступа : <https://scikit-learn.org/> (дата звернения: 09.09.2025).
42. Varoquaux G. et al. Joblib: running Python functions as pipeline jobs [Электронный ресурс]. 2024. Режим доступа : <https://joblib.readthedocs.io/> (дата звернения: 09.09.2025).
43. Parallelizing and memorizing Python programs with Joblib [Электронный ресурс] // Admin magazine. 2014. Режим доступа : <https://www.admin-magazine.com/Archive/2014/20/Parallelizing-and-memorizing-Python-programs-with-Joblib> (дата звернения: 09.09.2025).
44. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 (General Data Protection Regulation) [Электронный ресурс]. 2016. Режим доступа : <https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng> (дата звернения: 09.09.2025).
45. Nair V., Hinton G.E. Rectified Linear Units Improve Restricted Boltzmann Machines [Электронный ресурс] // Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010. P. 807–814. Режим доступа : <https://proceedings.mlr.press/v9/nair10a.html> (дата звернения: 09.09.2025).
46. Goodfellow I., Bengio Y., Courville A. Deep Learning [Электронный ресурс]. MIT Press, 2016. Режим доступа : <https://www.deeplearningbook.org/> (дата звернения: 09.09.2025).

47. Kingma D.P., Ba J. Adam: A Method for Stochastic Optimization [Электронный ресурс]. 2014. (arXiv:1412.6980). Режим доступа : <https://arxiv.org/abs/1412.6980> (дата звернення: 09.09.2025).