

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

## МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Нейромережева інформаційна технологія медичного  
діагностування»

Виконав: студент 2-го курсу, групи ЗКП-22м  
спеціальності І22 «Комп'ютерні науки»

(номер і назва напрямку підготовки, спеціальності)



Пащенко О.О.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН



Паночинин І.О.М.

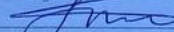
(прізвище та ініціали)

« 07 »

12

2023 р.

Оponent: к.т.н., професор каф. КСУ



Биков М. М.

(прізвище та ініціали)

« 07 »

12

2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

« 08 »

12

2023 р.

Вінниця ВНТУ - 2023 рік

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та  
автоматизації  
Кафедра комп'ютерних наук  
Рівень вищої освіти II-й (магістерський)  
Галузь знань – 12 «Інформаційні технології»  
Спеціальність – 122 «Комп'ютерні науки»  
Освітньо-професійна програма – «Системи штучного інтелекту»

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри КН  
Д.т.н., проф. Яровий А.А.

29.08 2023 року





### **ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Пашенку Олексію Олеговичу

(прізвище, ім'я, по батькові)

- Тема роботи Інформаційна технологія медичного діагностування на основі нейронної мережі Кохонена  
керівник роботи к.т.н., доцент кафедри КН Паночишин Ю.М  
затверджені наказом вищого навчального закладу від "18" 09 2023 року № 247
- Строк подання студентом роботи 13.11 2023 року
- Вихідні дані до роботи:  
Вхідні дані – дані, на основі яких робиться діагноз – не менше 12 параметрів аналізу крові; кількість діагностованих груп хвороб – не менше 10; обсяг навчальної вибірки – не менше 800, обсяг тестової вибірки – не менше 100, середовище програмування – об'єктно-орієнтоване.
- Зміст текстової частини:  
Вступ, аналіз предметної області діагностування за аналізом крові, розробка інформаційної технології нейромережевого діагностування за аналізом крові, програмна реалізація інформаційної технології нейромережевого діагностування за аналізом крові, економічна частина, висновки, перелік використаних джерел, додатки
- Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)  
Алгоритм роботи програми нейромережевого діагностування за аналізом крові, UML діаграма класів програми нейромережевого діагностування за аналізом крові, робочі вікна програми нейромережевого діагностування за аналізом крові, результати тестування програми нейромережевого діагностування за аналізом крові та програми-аналога.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Паночишин Ю.М., к.т.н., доц. каф. КН	 29.08.23	 10.11.23
5	Кавецький В. В., к.е.н., доц. каф. ЕПВМ	 8.10.23	 23.10.23

7. Дата видачі завдання 29.08. 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сучасного рівня інформаційних технологій діагностування за аналізом крові. Постановка задач дослідження	01.09.23 - 05.09.23	Розділ 1
2	Побудова моделей діагностування за аналізом крові на основі нейромереже та функціонування нейронної мережі	06.09.23 - 16.09.23	Розділ 2
3	Практичне застосування та оцінка ефективності розроблених моделей	17.09.23 - 07.10.23	Розділ 3,4
4	Підготовка економічної частини	08.10.23 - 23.10.23	Розділ 5
5	Апробація та/або впровадження результатів дослідження	24.10.23 - 01.11.23	Тези доповідей
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	02.11.23 - 10.11.23	Пояснювальна записка, графічний матеріал, презентація

Студент



(підпис)

Пащенко О.О.

Керівник роботи



(підпис)

Паночишин Ю.М.

## АНОТАЦІЯ

УДК 004.8

Пащенко О. О. Нейромережева інформаційна технологія медичного діагностування. Магістерська кваліфікаційна робота зі спеціальності 122 – «Комп'ютерні науки», освітня програма – «Системи штучного інтелекту». Вінниця: ВНТУ, 2023. 110 с.

На укр. мові. Бібліогр.: 23 назв; рис.: 20; табл. 14.

У даній магістерській кваліфікаційній роботі проаналізовано процес комп'ютеризованого медичного діагностування за аналізом крові, розглянуто параметри аналізу крові та їх зв'язок з відповідними групами хвороб. Було обґрунтовано вибір нейронної мережі Кохонена та процесу кластеризації для інформаційної технології медичного діагностування. Спроектовано структуру, математичну модель та порядок функціонування нейронної мережі Кохонена. Розроблено структуру процесів обробки інформації інформаційної технології. Програмна реалізація здійснена на мові C# у середовищі Visual Studio з використанням бібліотеки Accord.NET. Розроблена програма має вищу достовірність медичного діагностування пацієнтів (91%), ніж аналогічна програма (84%), а значить достовірність медичного діагностування пацієнтів покращена на 7%.

Графічна частина складається із 6 плакатів.

У економічному розділі доведено доцільність проведення науково-дослідної роботи за даною темою. Термін окупності становить 1,05 р., що свідчить про комерційну привабливість науково-технічної розробки.

Ключові слова: медичне діагностування, нейронна мережа Кохонена, аналіз крові.

## **ABSTRACT**

Paschenko O. O. Neural network information technology of medical diagnosis. Master's thesis in the specialty 122 - «Computer sciences», educational program – «Artificial intelligence systems». Vinnytsia: VNTU, 2023. 110 p.

In Ukrainian language. Bibliogr .: 23 titles; fig . 20; table 14.

In this master's qualification work, the process of computerized medical diagnosis based on blood analysis was analyzed, the parameters of blood analysis and their connection with the relevant groups of diseases were considered. The choice of the Kohonen neural network and the clustering process for the information technology of medical diagnosis was substantiated. The structure, mathematical model and order of functioning of Kohonen's neural network are designed. The structure of information processing processes of information technology has been developed. The program was implemented in the C# language in the Visual Studio environment using the Assord.NET library. The developed program has a higher reliability of medical diagnosis of patients (91%) than a similar program (84%), which means that the reliability of medical diagnosis of patients is improved by 7%.

The graphic part consists of 6 posters.

In the economic section, the expediency of conducting research work on this topic is proven. The payback period is 1.05 years, which indicates the commercial attractiveness of scientific and technical development.

**Key words:** medical diagnosis, Kohonen's neural network, blood analysis.

## ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ КОМП'ЮТЕРИЗОВАНОГО МЕДИЧНОГО ДІАГНОСТУВАННЯ.....	7
1.1 Процес комп'ютеризованого медичного діагностування за аналізом крові .....	7
1.2 Обґрунтування методу розв'язання задачі.....	12
1.3 Обґрунтування вибору аналогу до програмного засобу медичного діагностування за аналізом крові.....	15
1.4 Постановка задачі .....	17
1.5 Висновок до розділу 1 .....	19
2 ПРОЕКТУВАННЯ НЕЙРОМЕРЕЖЕВОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МЕДИЧНОГО ДІАГНОСТУВАННЯ.....	20
2.1 Обґрунтування вибору теоретичної основи нейромережевої інформаційної технології медичного діагностування .....	20
2.2 Аналіз процесу кластеризації.....	26
2.3 Обґрунтування вибору методу кластеризації.....	27
2.4 Структура, математична модель та порядок функціонування нейромережі Кохонена.....	31
2.5 Алгоритм нейромережевого медичного діагностування за аналізом крові .....	38
2.6 Висновок до розділу 2 .....	44
3 ПРОГРАМНА РЕАЛІЗАЦІЯ НЕЙРОМЕРЕЖЕВОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МЕДИЧНОГО ДІАГНОСТУВАННЯ.....	45
3.1 Обґрунтування вибору мови та середовища програмування .....	45
3.2 Реалізація основних компонентів програми медичного діагностування на основі нейронної мережі Кохонена .....	47
3.3 Висновок до розділу 3 .....	54
4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ МЕДИЧНОГО ДІАГНОСТУВАННЯ НА ОСНОВІ НЕЙРОННОЇ МЕРЕЖІ КОХОНЕНА.....	55

4.1 Тестування програми медичного діагностування на основі нейронної мережі Кохонена .....	55
4.2 Аналіз результатів роботи програми медичного діагностування на основі нейронної мережі Кохонена .....	58
4.3 Висновок до розділу 4 .....	60
5 ЕКОНОМІЧНА ЧАСТИНА .....	61
5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки.....	61
5.2 Розрахунок узагальненого коефіцієнта якості розробки.....	65
5.3 Розрахунок витрат на проведення науково-дослідної роботи .....	67
5.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором .....	80
5.5 Висновок до розділу 5 .....	84
ВИСНОВКИ .....	85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	87
Додаток А (обов'язковий) ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	90
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ.....	90
Додаток Б (обов'язковий) Лістинг програми.....	91
Додаток В (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА .....	101
Додаток Г (довідниковий) Інструкція користувача .....	108

## ВСТУП

**Актуальність.** У сучасному світі спостерігається бурхливий розвиток науки і техніки в усіх галузях. Медичній галузі приділяється особлива увага. Автоматизація та комп'ютеризація процесів медичної діагностики дозволяє забезпечити високу точність постановки діагнозу пацієнтів та спростити напружену роботу лікарів.

Відсутність точних, загальнодоступних та швидких програмних засобів діагностування є причиною невірного визначення хвороб пацієнтів у лікарнях, витрат часу на очікування у чергах, реєстрацію та проведення непотрібних аналізів. У такій ситуації бажаним є створення програмного засобу для діагностування пацієнтів, що зможе забезпечити високу достовірність та швидкодію діагностування різноманітних хвороб, який не потребує спеціального апаратного забезпечення для свого функціонування.

Для створення заданої інформаційної технології, призначеної для визначення діагнозу пацієнтів, запропоновано використати підхід до діагностування, заснований на дослідженні показників загального аналізу крові, який дозволить підвищити достовірність визначення діагнозу пацієнтів. Основою методу реалізації даного підходу буде кластеризація даних за допомогою нейронної мережі Кохонена.

**Зв'язок роботи з науковими програмами, планами, темами.** Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем» та плану наукової та навчально-методичної роботи кафедри.

**Мета і завдання досліджень.** Метою магістерської кваліфікаційної роботи є підвищення достовірності медичного діагностування за рахунок використання нейромережевих інтелектуальних технологій.

Для досягнення мети розробки необхідно виконати такі задачі:



- провести аналіз предметної області медичного діагностування;
- розглянути існуючі методи медичного діагностування та обрати й обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- розробити інформаційну технологію нейромережевого медичного діагностування;
- обґрунтувати вибір типу нейромережі та розробити її структуру;
- виконати програмну реалізацію запропонованої інформаційної технології;
- провести тестування програмного продукту та виконати аналіз отриманих результатів.

**Об’єкт дослідження** – процеси комп’ютеризованого медичного діагностування із застосуванням нейромережевих інтелектуальних технологій.

**Предмет дослідження** – інформаційна технологія та програмні засоби нейромережевого медичного діагностування та достовірність діагностування.

**Методи дослідження.** У роботі використані наступні методи наукових досліджень: системного аналізу, теорії нейронних мереж для реалізації інформаційної технології, методи математичної статистики для розробки процесу розв’язання задачі нейромережевого діагностування за аналізом крові та обрахунків результатів експериментів із програмним засобом, об’єктно-орієнтованого програмування.

#### **Наукова новизна одержаних результатів.**

1. Набула подальшого розвитку інформаційна технологія нейромережевого медичного діагностування, яка відрізняється використанням штучної нейронної мережі Кохонена, що дозволило підвищити достовірність програмних засобів медичного діагностування.

**Практичне значення** одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення нейромережевого медичного діагностування.

Запропонована інформаційна технологія сприяє підвищенню достовірності програмних засобів нейромережевого медичного діагностування, зокрема:

- розроблено алгоритм роботи програмного забезпечення нейромережевого медичного діагностування;
- розроблено програмні засоби для нейромережевого медичного діагностування.

**Достовірність теоретичних положень** магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології нейромережевого медичного діагностування. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

**Особистий внесок здобувача.** Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз процесу нейромережевого діагностування за аналізом крові та методів підвищення достовірності програмних засобів діагностування за аналізом крові [1].

**Апробація результатів роботи.** Результати роботи були апробовані на Міжнародній науково-практичній Інтернет-конференції студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2024)» (м. Вінниця, Україна, 2023-2024) [1].

**Публікації.** За результатами досліджень опубліковано одні тези доповіді на науково-технічній конференції [1].

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ КОМП'ЮТЕРИЗОВАНОГО МЕДИЧНОГО ДІАГНОСТУВАННЯ

## 1.1 Процес комп'ютеризованого медичного діагностування за аналізом крові

У нашому житті глобальний процес автоматизації досяг майже усіх галузей людської діяльності, включно з медициною. В медицині виділяють такі напрямки її комп'ютеризації та подальшого удосконалення:

- створення БД пацієнтів лікарень;
- системи автоматизованої реєстрації пацієнтів;
- системи діагностування захворювань;
- системи телемедицини (телекомунікаційні та електронні інформаційні технології для надання медичної допомоги та послуг у галузі охорони здоров'я в потрібному місці) [2, 3].

Традиційний процес медичного діагностування пацієнтів у лікарнях є досить тривалим та багатоетапним – пацієнт має зареєструватись, пройти консультацію у сімейного лікаря, отримати направлення до лікаря-спеціаліста, який може визначити хибно з деякою ймовірністю реальну хворобу пацієнта. Для того, щоб підтвердити або спростувати діагноз треба пройти низку аналізів, що може призвести до загострення хвороби через довготривалість цього процесу або поставлення хибного діагнозу.

З огляду на те, що потенційними споживачами розроблюваної системи будуть працівники медзакладів та їх клієнти, а їх потребою буде якісне медичне обслуговування, то на сучасному етапі це забезпечується комп'ютеризацією медичної галузі. Отже, на сьогоднішній день, коли більшість людей має мобільні гаджети, раціональним є використання їх функціональних можливостей та широке застосування мобільних інформаційних додатків населенням, тобто покращення загального процесу

медичного діагностування шляхом застосування телемедицини. Це дозволить пацієнтам отримувати діагноз без необхідності відвідувати лікарню.

Наявні автоматизовані методи медичної діагностики є недостатньо ефективними – більшість систем орієнтована на визначення конкретної хвороби, що через специфіку медичної галузі, має високу імовірність виявитись хибним; мають велику складність алгоритмів та їх програмної реалізації; можуть використовуватись лише фахівцями у медицині [4].

Таким чином, існує проблема недосконалості сучасних методів та засобів медичного діагностування, які не забезпечують достатню точність виявлення захворювань, швидкодію (оперативність) та зручність використання із застосуванням персональних мобільних гаджетів (телефони, планшети, смартфони та ін.), не можуть використовуватись пацієнтами дистанційно без потреби у відвідуванні лікарні.

З метою покращення процесу медичного діагностування пацієнтів доцільно було б створити програмний засіб, який характеризується сумісністю з довільним програмним, апаратним забезпеченням та має дружній та інтуїтивно зрозумілий інтерфейс, який буде використовувати інтелектуальні алгоритми постановки діагнозу пацієнта, забезпечувати його швидкість та достовірність.

Для такого автономного медичного діагностування доцільно використовувати підхід, заснований на застосуванні параметрів загального аналізу крові, що проводиться при більшості захворювань і при профілактичних обстеженнях, і містить в собі 15 показників, кожен з яких при наявності відхилень від визначених нормальних значень цих показників здорової людини передбачає імовірність наявності певних хвороб. Медичне діагностування на основі обробки отриманих параметрів аналізу крові допоможе виявити вид захворювання пацієнта та надати рекомендації щодо терапії [5].

Загальний аналіз крові – це лабораторне дослідження крові пацієнта, що включає підрахунок всіх видів клітин крові (еритроцитів, лейкоцитів,

тромбоцитів), визначення їх параметрів, лейкоцитарної формули, вимірювання рівня гемоглобіну, підрахунок співвідношення клітинної маси до плазми (гематокрит). Загальний аналіз крові призначається при більшості захворювань і профілактичних обстеженнях і містить в собі 15 показників, кожен з яких при відхиленні від норми сигналізує про імовірну наявність певних хвороб [6].

Розглянемо показники загального аналізу крові та відповідні їм хвороби при відхиленні від норми:

1) Еритроцити — червоні кров'яні тільця, які втратили ядро та містять гемоглобін і транспортують кисень по організму.

Підвищення норми еритроцитів може свідчити про поліцитемію (хвороба кісткового мозку) і зневоднення, а зниження — про анемію внаслідок втрати крові, нестачі в організмі заліза та вітамінів.

2) Гемоглобін — основний компонент еритроцитів, який забарвлює кров у червоний колір і доправляє кисень до органів та тканин.

Підвищений гемоглобін може свідчити про поліцитемію, надмірне фізичне навантаження, зневоднення, згущення крові. Знижений гемоглобін може свідчити про анемію.

3) Колірний показник визначається кількісним співвідношенням гемоглобіну в еритроцитах.

Цей показник використовується для визначення виду анемії. Перевищення норми може свідчити про сфероцитоз, зниження норми — про залізодефіцитну анемію.

4) Гематокрит - це відсоткове співвідношення формених елементів крові до її загального об'єму. Цей показник допомагає лікарю розділяти з чим пов'язана анемія: з втратою еритроцитів, що свідчить про захворювання, або з надмірним розрідженням крові.

5) Тромбоцити - це кров'яні тільця, що утворюються з клітин кісткового мозку. Завдяки цим клітинам кров має властивість згортатися.

Підвищення тромбоцитів вказує на можливий запальний процес, поліцитемію, а також може бути наслідком хірургічних операцій. Зниження кількості тромбоцитів може свідчити про системні аутоімунні захворювання, апластичну анемію, гемолітичну анемію, гемолітичну хворобу, ізоімунізацію по резус-факторам і групам крові.

6) Ретикулоцити — це молоді, незрілі еритроцити, у яких наявні залишки ядра. У кров потрапляє лише незначна частина таких еритроцитів, а основна частина міститься у кістковому мозку.

Надлишковий вміст ретикулоцитів в крові свідчить про анемію, крововтрату. Зниження кількості ретикулоцитів може бути ознакою захворювань нирок, порушень еритроцитного обміну, апластичної анемії.

7) Середній об'єм еритроцитів (норма - 80...95 для чоловіків і 80...100 для жінок). Об'єм еритроцитів зменшується при залізодефіцитній анемії, а підвищується – при В12 дефіцитній, при гепатитах, при зниженні функції щитовидної залози.

8) Середній вміст гемоглобіну в еритроциті. Цей показник підвищується рідко, а ось зниження — ознака анемії або зменшення функції щитовидної залози.

9) Середня концентрація гемоглобіну в еритроцитах. Підвищення значень майже завжди говорить про апаратну помилку, а зниження – про наявність залізодефіцитної анемії.

ШОЕ — це аббревіатура, що розшифровується, як швидкість осідання еритроцитів. Відхилення ШОЕ від норми може означати наявність запального або патологічного процесу, який протікає в організмі. ШОЕ може підвищуватися при інфекційному захворюванні, вагітності, анемії, запальному процесі, утворенні пухлини.

10) Лейкоцити — це білі кров'яні тільця. Їх основна функція - захищати організм від мікробів і чужорідних речовин.

Підвищення кількості лейкоцитів може свідчити про лейкози, запальні або інфекційні процеси, алергії, крововтрати, аутоімунні захворювання.

Зниження кількості лейкоцитів може свідчити про деякі інфекції (краснуха, грип, кір тощо), генетичні аномалії імунітету, патології кісткового мозку, підвищення функції селезінки. Лейкоцитарна формула — це відсоткове співвідношення різних видів лейкоцитів.

11) Нейтрофіли — забезпечують боротьбу організму з бактерійною інфекцією і відповідають за видалення клітин, загиблих у результаті цієї боротьби. У крові наявні два види нейтрофілів:

Підвищення нейтрофілів говорить про наявність бактеріальних, грибкових та деяких інших інфекцій, наявність запальних процесів внаслідок травматизації тканин, при артрозах, артритих та ін.

Нейтрофіли можуть також підвищуватися внаслідок температурних перепадів, фізичних навантажень, під час вагітності. Зменшення нейтрофілів може виникнути внаслідок виснаження організму, внаслідок тривалих хронічних захворювань, захворювань щитовидної залози.

12) Еозинофіли — беруть участь у боротьбі з алергією, паразитарними інвазіями. Кількість еозинофілів збільшується при паразитарних інвазіях і бронхіальній астмі, алергічних реакціях, а також при наявності злоякісних пухлин. Зменшується при пологах, гнійних інфекціях, шоку, оперативних втручаннях.

13) Лімфоцити — ведуть боротьбу з чужорідними клітинами й білками, вірусними інфекціями, виділяють у кров антитіла і блокують антигени.

Кількість лімфоцитів збільшується при інфекційному монокульозі, туберкульозі, гепатиті, лейкозах, а також вірусних інфекціях. Зменшується при гострій інфекції, раку, аутоімунних захворюваннях, імунодефіциті.

14) Базофіли — беруть участь в алергічних реакціях негайного виду.

Кількість базофілів збільшується при захворюваннях крові, вітрянці, виразковому коліті, непереносимості ліків і харчових продуктів.

Зменшується при гіпертиреозі, вагітності, овуляції, гострих інфекціях, стресі та підвищеному виробленні гормонів наднирниками.

15) Моноцити — знищують чужорідні білки і клітини в тканинах організму. Кількість моноцитів збільшується після гострих інфекцій, при сифілісі, туберкульозі, ревматичних хворобах. Зменшується при ураженні кісткового мозку.

## 1.2 Обґрунтування методу розв'язання задачі

Для визначення медичного діагнозу пацієнта необхідно знати нормальні значення кожного з параметрів загального аналізу крові (табл.1.1), а також показники, характерні для кожної хвороби, яку буде діагностувати програмний засіб (табл. 1.2).

Таблиця 1.1 – Діапазони нормальних значень параметрів загального аналізу крові

№	Параметри аналізу крові	позначення	Норма	
			чоловіки	жінки
1	Лейкоцити —white blood cells	WBC	4...9	
2	Еритроцити —red blood cells	RBC	4...5,1	3,7...4,7
3	Гемоглобін— hemoglobin	HGB	130...160	120...140
4	Гематокрит — hematocrit	HTC	0,38...0,50	
5	Середній об'єм еритроцитів— mean corpuscular volume	MCV	80...100	
6	Середній вміст гемоглобіну в еритроциті — mean concentration hemoglobin	MCH	24...34	
7	Середня концентрація гемоглобіну в еритроциті — mean corpuscular hemoglobin concentration	MCHC	32...38	
8	Індекс розподілу еритроцитів — red cell distribution	RDW	4,5...14,5	
9	Тромбоциты — platelets	PLT	150...4000	
10	Тромбокрит — Thrombocrit	PTC	1...10	
11	Нейтрофіли (абс.) — Neutrophil	NEU	2...6	
12	Лімфоцити (абс.) — Lymphocyte	LYMP	1...3	
13	Моноцити (абс.) — Monocytes	MONO	0...0,5	
14	Еозінофіли (абс.)— Eosinophils	EO	0...0,5	
15	Базофіли (абс.) — Basophil	Baso	0...0,1	

Таким чином, загальний аналіз крові може забезпечити всебічний аналіз пацієнта, групу захворювань та навіть його приблизний (орієнтовний) діагноз.



Для реалізації цієї методики доречно використовувати кластеризацію даних, яка зможе забезпечити високу достовірність їх обробки і при цьому мати нескладну програмну реалізацію.

Таблиця 1.2 – Значення параметрів загального аналізу крові для визначених хвороб, які може діагностувати програмний додаток

№	хвороба	гемоглобін		еритроцити		колірний показник	ретикулоцити	тромбоцити	ШОЕ		лейкоцити	паліткоюдерні	сегментоядерні	еозинофіли	базофіли	лімфоцити	моноцити
		ч	ж	ч	ж				ч	ж							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	відсутня	145	135	4,55	4,2	1	0,7	250	5,5	8,5	6,5	3,5	59,5	2,5	0,5	29	5,5
2	анемія (легка)	114	104	3,44	3,1	1	1,8	250	13,6	22,6	6,5	3,5	59,5	2,5	2,1	29	12,6
3	анемія (середня)	90	80	2,8	2,8	1	2,3	250	20	29	6,5	3,5	59,5	2,5	2,1	29	12,6
4	анемія (тяжка)	79	69	1,7	1,6	1	3,3	250	31	42	6,5	3,5	59,5	2,5	2,1	29	12,6
5	загальне запалення організму	145	135	5,66	5,3	1	-0,3	461	20	29	17	12	98	11	0,5	29	17
6	вірусна інфекція	145	135	4,55	4,2	1	0,7	39	20	29	1,4	12	98	11	-1,1	63	-6
7	отруєння	145	135	2,8	2,8	1	0,7	250	20	29	17	12	98	11	2,1	29	17
8	алергія	145	135	2,8	2,8	1	-1,9	461	20	29	15	3,5	59,5	11	0,5	63	5,5
9	глисти	145	135	4,55	4,2	1	0,7	250	20	29	6,5	3,5	59,5	11	2,1	29	5,5
10	злоякісні захворювання	114	109	4,55	4,2	0,54	0,7	39	20	29	6,5	12	98	2,5	0,5	29	5,5
11	онкологічні захворювання	145	135	3,44	3,1	1,46	-0,3	601	20	29	6,5	12	98	2,5	0,5	-5	17
12	зневоднення організму	191	161	6,3	5,8	1,76	3,3	250	-8	-12	-7	3,5	59,5	2,5	0,5	29	5,5
14	гемолітична анемія	99	99	1,7	1,6	1,31	2,3	39	20	29	6,5	3,5	59,5	2,5	2,1	63	12,6
15	злоякісні новоутворення	114	109	5,66	5,3	1,46	1,8	601	31	42	17	17	123	11	0,5	-5	17
16	тромбоз	145	135	4,55	4,2	1	0,7	461	5,5	8,5	6,5	3,5	59,5	2,5	2,1	29	17
17	лейкоз	145	135	4,55	4,2	1	0,7	461	5,5	8,5	15	3,5	59,5	2,5	3,1	63	22
18	бактеріальна інфекція	145	135	4,55	4,2	1	0,7	250	20	29	17	17	123	7,6	-1,1	-5	5,5
19	захворювання суглобів	145	135	4,55	4,2	1	0,7	250	20	29	-2	12	98	2,5	0,5	-5	17
20	зниження імунітету	145	135	4,55	4,2	1	0,7	109	5,5	8,5	-2	3,5	59,5	2,5	0,5	6	-6
21	захворювання нирок	114	109	5,66	5,3	1	-0,3	250	20	29	10,6	3,5	59,5	2,5	-1,1	29	5,5
22	захворювання кісткового мозку	145	135	3,44	3,1	1	1,8	250	5,5	8,5	10,6	3,5	59,5	2,5	0,5	-5	-6
23	гіпергідратація	99	99	1,7	1,6	1	-1,9	250	5,5	8,5	6,5	3,5	59,5	2,5	0,5	29	5,5
24	залізодефіцитна анемія	69	79	1,7	1,6	0,24	1,8	39	-8	-12	-7	3,5	59,5	2,5	0,5	29	12,6
25	захворювання шлунку	145	135	4,55	4,2	1,46	0,7	250	5,5	8,5	6,5	3,5	59,5	2,5	0,5	29	5,5
26	поліцитемія	191	161	5,66	5,3	1	0,7	601	-8	-12	10,6	3,5	59,5	11	0,5	29	5,5
27	мієлолейкоз	145	135	4,55	4,2	1	0,7	391	5,5	8,5	-2	3,5	59,5	2,5	0,5	29	5,5
28	захворювання селезінки	221	207	4,55	4,2	1	0,7	461	5,5	8,5	-2	3,5	59,5	2,5	-1,1	29	5,5

Параметри загального аналізу крові у вигляді вхідного числового вектору даних будуть порівнюватись з векторами визначених хвороб. Вектор, найближчий до вхідного веазу нає хворобу пацієнта. Якщо у хворого кілька захворювань, то можна провести аналогічне порівняння, але з вилученням вже діагностованої хвороби [7, 8]. Кластеризація даних, яка буде використовуватись для визначення медичного діагнозу пацієнтів, є досить ефективною при застосуванні запропонованого підходу, так як вхідні дані являють собою вектори чисел.

Медичне діагностування пацієнтів на сьогоднішній день не так часто проводиться із використанням програмних засобів. Зазвичай застосовують апаратні засоби, що орієнтовані на визначення конкретно однієї хвороби. Медичне діагностування без спеціалізованого інструментарію часто веде до визначення невірної діагнозу, характеризується недостатньою швидкістю опрацювання даних.

Медичне діагностування за загальним аналізом крові без використання програмних додатків у сучасних умовах глобальної зміни клімату, виникнення нових захворювань, забруднення повітря, еволюцією захворювань, пристосування організму людини до певних захворювань залежно від її місцезнаходження, оточуючого середовища та навіть їжі, веде до поступової зміни нормальних значень параметрів загального аналізу крові людей, які проживають на визначеній території [9]. Тому можуть виникнути розбіжності при постановці діагнозу під час простого порівняння лікарем аналізів крові пацієнта та нормальних значень загального аналізу крові. Це викликає недостатню достовірність визначення медичного діагнозу пацієнта за аналізом крові без застосування інтелектуальних програмних додатків, які можуть опрацьовувати інформацію з урахуванням територіальних та вікових змін у показниках загального аналізу крові [10].

### **1.3 Обґрунтування вибору аналогу до програмного засобу медичного діагностування за аналізом крові**

Системи онлайн діагностування на даний момент мають недостатню ефективність, а саме - не забезпечують задовільну швидкодiю та високу достовiрнiсть визначення медичного діагнозу пацієнта.

Найпопулярнiша на сьогодні система онлайн діагностування YourDiagnosis [11], базується на класичних медичних алгоритмах, якi на даний момент вже не є новаторськими. Вона має середню точнiсть діагностування 70%. Процес медичного діагностування мiстить в собі занесення у вiдповiднi комiрки значень показникiв загального аналізу кровi. Це є досить просто для користувача, але одночасно i повiльно. Отже, якщо потрiбно визначити діагнози для кiлькох пацієнтiв, то ця система не є достатньо зручною. Неможливе збереження завчасно введених даних, тому, якщо на сайтi системи або з Інтернетз'єднанням виникла помилка, то це призведе до того, що всi показники потрiбно знов перенабирати. Швидкодiя медичного діагностування може тривати до двох хвилин.

Інтерактивна система діагностування Diagnos.com [12] має найбільшу в світі базу даних захворювань (близько 240) та заснована на методах нечіткої логіки. Процес діагностування цієї системи дуже повільний. Пацієнт має переглянути всі симптоми імовірних хвороб та при наявності відмітити їх. Процес обробки даних після відмітки всіх симптомів може тривати 10 хвилин. Достовірність медичного діагностування даної системи у середньому дає 67%, оскільки визначення медичного діагнозу лише за симптомами хвороби не є надійним. Медичне діагностування за аналізом крові на даному сайті не пристосоване до мобільних гаджетів. Достовірність медичного діагностування складає у середньому 79%, а швидкодiя – до 5 хвилин.



Рисунок 1.1 – Вид головного вікна веб-сервісу YourDiagnosis

Отже, популярні сучасні системи медичного діагностування мають такі недоліки:

- в основі мають недостатньо ефективні алгоритми;
- мають недостатньо високу достовірність діагностування;
- пристосовані для визначення лише одної або невеликої кількості хвороб;
- час визначення діагнозу є досить великим через незручний процес введення даних та складні обрахунки;
- незручність процесу визначення діагнозу одночасно для кількох пацієнтів;
- неможливість застосування для телемедицини;

Таким чином, розробка нейромережевої інформаційної технології для медичного діагностування пацієнтів з використанням визначених параметрів загального аналізу крові та методу кластеризації даних є перспективним для попереднього формування діагнозу хворого, що зможе забезпечити

достовірність постановки діагнозу пацієнта, високу швидкість роботи, зручність та легкість для освоєння та використання, а також завдяки реалізації алгоритму кластеризації даних не буде потребувати особливих програмних та апаратних засобів, що дозволить провести впровадження системи кластеризації пацієнтів за аналізом крові в сучасні лікарні та замінить велику частину паперових записів, а також призведе до спрощення організації ресурсів лікарні для постановки одразу декількох діагнозів пацієнта та подальшої рекомендації його спеціалізованим лікарям.

#### **1.4 Постановка задачі**

Даний програмний додаток передбачає постановку попереднього медичного діагнозу пацієнту на основі аналізу крові як надійного, швидкого та простого засобу виявлення хвороб, а також кластеризації даних, що передбачає розподіл параметрів хвороб на кластери (групи), які визначають хворобу пацієнта, чиї показники крові підлягають обробці та, відповідно, дозволяють рекомендувати спеціалізованого лікаря відповідно до визначеної хвороби.

Враховуючи сучасний фінансовий стан лікарень, розроблюване програмне забезпечення має бути дешевим, нескладним у програмній реалізації, легко впроваджуваним та мати невеликий об'єм на диску для можливого використання програми на малопотужних комп'ютерах, нескладних мобільних гаджетах або через Інтернет. Програма не повинна потребувати додаткового програмного або апаратного забезпечення, причому підхід до реалізації повинен мати точні обрахунки та видавати безпомилкові результати. Тому дана програма має поєднувати раціональність, невимогливість до обладнання, вірність розрахунків, а також дружній інтуїтивний інтерфейс. Цього потребують користувачі – пацієнти і працівники лікарень.

Отже, розроблювана програма має забезпечувати такі властивості:

1) Висока швидкість та достовірність роботи. Завдяки ефективного підходу та методу його реалізації процес обрахунків та обробки інформації суттєво пришвидшиться та забезпечить високу достовірність формулювання діагнозу пацієнта.

2) Простий та зручний доступ до необхідної інформації та функцій. Програма повинна характеризуватися дружнім інтерфейсом та нескладною роботою з нею, оскільки орієнтована на користувачів невисокого рівня досвідченості у роботі з комп'ютером.

3) Можливість роботи з великими об'ємами даних. Застосований в програмному додатку метод кластеризації даних дозволяє виконувати обчислення та подальшу обробку результатів для великих обсягів інформації, внесеної до системи. При цьому не повинні зменшуватися достовірність постановки діагнозу та зручність введення даних для їх подальшої обробки.

4) Невибагливість до програмного та апаратного забезпечення. Беручи до уваги обмеженість ресурсів для впровадження будь-яких систем у сучасні лікарні та для подальшого використання пацієнтами, програма буде розроблена так, щоб витрати на її впровадження були найменшими, але не за рахунок функціональних можливостей, параметрів та загальної функціональності.

5) Безвідмовність роботи. Програма завдяки порівняній простоті реалізації майже не потребує втручання спеціалістів з технічного обслуговування та не викликає фатальних збоїв у системі.

6) Безпека та конфіденційність. Параметри загального аналізу крові та визначений діагноз пацієнта, що використовує програма, не мають бути доступними іншим користувачам.

Програма медичного діагностування пацієнтів створюється для автоматизації процесу постановки діагнозу пацієнтів згідно з параметрами їх загального аналізу крові. Його застосування оптимізує цей процес за рахунок підвищення достовірності діагностування та шляхом використання підходу,

заснованого на аналізах крові та методи їх програмної реалізації – кластеризація даних.

Для розв'язання задачі медичного діагностування цим програмним засобом користувачу необхідно мати результати загального аналізу крові, які можна отримати в будь-якій державній лікарні або приватній лабораторії. Виміряні значення параметрів аналізу крові заносяться у програму для подальшої обробки.

Вихідними даними програми є визначений попередній медичний діагноз. Попередній означає те, що формується не точна назва хвороби, а група хвороб з тим, щоб пацієнт у подальшому звернувся до відповідного лікаря-фахівця для уточнення діагнозу.

### **1.5 Висновок до розділу 1**

У розділі було проаналізовано процес комп'ютеризованого медичного діагностування за аналізом крові, розглянуто параметри аналізу крові та їх зв'язок з відповідними групами хвороб, виявлені проблеми та недоліки, пов'язані із недосконалістю сучасного діагностування. Було визначено актуальність проблеми та обґрунтовано метод її розв'язання – розробка інформаційної технології медичного діагностування на основі нейронної мережі Кохонена, що дозволить провести автоматизацію визначення попереднього діагнозу пацієнта та підвищить його достовірність. Були розглянуті основні програмні засоби, які виконують медичну діагностику, та визначені їх недоліки.

## **2 ПРОЕКТУВАННЯ НЕЙРОМЕРЕЖЕВОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МЕДИЧНОГО ДІАГНОСТУВАННЯ**

### **2.1 Обґрунтування вибору теоретичної основи нейромережевої інформаційної технології медичного діагностування**

Для реалізації комп'ютеризованого медичного діагностування з використанням показників аналізу крові перспективним видається застосування методів штучного інтелекту, які розв'язують задачу визначення діагнозу швидше та точніше, ніж людина. Основними методами штучного інтелекту є:

- системи логічного висновку,
- системи нечіткої логіки,
- експертні системи,
- імітаційне моделювання,
- генетичні алгоритми
- нейронні мережі.

Проаналізуємо переваги та недоліки цих методів та оберемо один з них як теоретичну основу для реалізації інформаційної технології медичного діагностування.

Системи логічного висновку [13] засновані на моделюванні міркувань. Теоретичним базисом є логіка. Програмні засоби, які використовують логічний висновок, являють собою набори правил та фактів виду «Якщо (умова), то (дія)». Найчастіше алгоритм, як послідовність дій, що приводять до необхідного результату, не задається.

Перевагами таких систем є відносна простота застосування, висока модульність, наочність, простота схеми логічного виведення, легкість внесення змін та доповнень у систему, що є зручним для розробників таких систем та їх користувачів. Організація виведення є єдиною формальною



процедурою доведення певного твердження для задоволення умовам правил, що занесені до системи. Являючись найбільшою перевагою, це одночасно є і недоліком складності застосування при побудові евристик, які відображають специфіку задачі. Незважаючи на те, що предметній області медицини підходить такий спосіб подання, оскільки є умови, що являють собою показники аналізу крові, та дії, що являють собою діагнози. Дані, що обробляються системою, мають великий об'єм та часто змінюються, що складно забезпечити за допомогою даної форми представлення. Існує також проблема виключення протиріч при представленні за допомогою логічного висновку, які виникають при визначенні складного діагнозу, що містить дві або більше хвороб одночасно.

Системи, які використовують нечітку логіку [13] оперують нечіткими поняттями, які є більш звичними для мислення людини, ніж формальні та чіткі математичні формули. Ці системи дають можливість оперувати вхідними даними, що задані нечітко: наприклад, неперервно змінювані в часі значення (динамічні), значення, щякію неможливо задати однозначно (результати статистичних опитувань, рекламних компаній, тощо); можливість нечіткої формалізації критеріїв оцінки і порівняння: оперування критеріями «більшість», «можливо», «переважно», тощо; можливість проведення якісних оцінок як вхідних даних, так і виведених результатів: оперування не тільки власне значеннями даних, але їхнім ступенем імовірності і розподілом; можливість швидкого моделювання складних динамічних систем і їхній порівняльний аналіз із заданою точністю.

Предметна область медичного діагностування може бути реалізована на основі нечіткої логіки, використовуючи симптоми хвороб як інформацію для неї або використовуючи нечіткі значення показників аналізу крові, наприклад, «рівень гемоглобіну трохи нищий за нормальний». Але при використанні такого методу діагностики виникає складність у формулюванні нечітких правил для бази знань системи, надлишковість бази знань, постійне збільшення її об'єму пропорційно кількості нових захворювань та їх ознак.

Експертні системи [13] використовують алгоритми успішних рішень у певній предметній області, містять аналітичні здатності та знання одного або декількох експертів щодо даної галузі, і здатні робити логічні висновки на основі цих знань, забезпечуючи вирішення специфічних завдань (консультування, діагностування, навчання, проектування, тестування тощо) без участі експерта (фахівця в конкретній галузі). Також визначаються як системи, що використовують базу знань для вирішення завдань (видачі рекомендацій) у деякій предметній галузі. Експертні системи моделюють механізм мислення людини, їх застосовують для операцій, які потребують великого досвіду, накопиченого людьми, що є важливим для реалізації інформаційної технології медичного діагностування, яка має виконувати завдання замість людини-експерта.

Експертні системи, перш за все, орієнтовані на користувачів, а не на експертів та розробників у визначеній галузі знань, що задовольняє вимоги пацієнтів щодо використання інформаційної технології, яка розробляється. Але при цьому часто виникає проблема використання таких систем, яка полягає у труднощах адаптації недосвідчених користувачів. Більшість експертних систем доступні лише тим експертам, які створювали їх бази знань. Тому потрібно розробляти паралельно відповідний користувацький інтерфейс, який забезпечив би кінцевому споживачу властивий йому режим роботи, що потребує окремих витрат та досліджень бажаних споживчих характеристик розробки. Після отримання нових знань експертів система не завжди покращує свою роботу, що є вагомим недоліком – система повинна часто оновлюватись для того, щоб відстежувати останні зміни у медичній галузі (поява нових хвороб чи зміна значень показників аналізу крові).

Імітаційне моделювання [13] – це метод, що дозволяє будувати моделі процесів, які описують, як ці процеси проходили б насправді. Таку модель можна «програвати» у часі як для одного випробування, так і для заданої їх кількості. При цьому визначатимуться результати випадковим характером процесів. За цими даними отримують доволі стійку статистику.

Системи, які використовують імітаційне моделювання, відрізняються високою достовірністю, але при цьому для розробки імітаційної моделі за предметною областю медичного діагностування потрібна велика кількість даних, якими система буде оперувати. Розробка такої системи забирає багато часу та вимагає великих грошових ресурсів. Для отримання результатів потрібні також і великі витрати часу.

Генетичні алгоритми [13] – це еволюційні алгоритми пошуку, які використовуються для розв'язання задач моделювання і оптимізації шляхом послідовного комбінування, підбору і варіації шуканих параметрів з використанням механізмів, які нагадують еволюцію в біології. Особливістю генетичного алгоритму є використання оператора «схрещування», який виконує рекомбінацію рішень-кандидатів, роль якої аналогічна ролі схрещення у живій природі.

Завдання медичного діагностування у випадку використання генетичного алгоритму може бути сформульована як пошук оптимального значення (виду хвороби), де значення – це складна функція, яка залежить від визначених вхідних параметрів (напр., показників аналізу крові). Потрібно знайти такі значення параметрів, при яких досягається найкраще точне значення функції. В інших випадках точний оптимум не потрібний - рішенням може вважатися будь-яке значення, яке є кращим за певну задану величину. Перевага генетичного алгоритму полягає в його здатності маніпулювати одночасно багатьма параметрами, що дозволяє ефективно використовувати залежність вихідного результату від 13 змінних як для безпосередньо діагностування, так і для передбачення розвитку хвороби за симптомами. При даних перевагах генетичні алгоритми не завжди знаходять оптимальне рішення та часто потребують багато часу для вирішення задачі.

Штучні нейронні мережі [14, 15] – це математичні моделі, побудовані за принципами функціонування біологічних нейромереж — мереж нервових клітин живого організму. Архітектура і принцип дії базуються на аналогії з

мозком живих істот. Ключовим елементом цих систем є штучний нейрон як імітаційна модель нервової клітини мозку – біологічного нейрона.

Нейронні мережі мають переваги у розв'язанні задач при невідомих закономірностях, коли залежність між вхідними та вихідними даними невідома, чого не можуть забезпечити традиційні математичні методи та експертні системи. Така особливість допомагає виявити нові хвороби та їх розвиток. Нейромережі швидко адаптуються до змін у режимі реального часу. Із зростанням адаптивної здатності системи, заснованої на нейромережі, підвищується її стійкість у нестационарному середовищі, яким для предметної області медичного діагностування є набір хвороб та показники аналізу крові, які можуть змінюватись. Окремо зазначимо потенційно високу швидкодію завдяки паралелізму обробки інформації, що дозволяє діагностувати одночасно кількох пацієнтів, а також високу достовірність, яка є найважливішим фактором для даної предметної області. Головним недоліком застосування нейромережі для реалізації інформаційної технології медичного діагностування є потреба у великому обсязі навчальної вибірки.

Отже, оглянувши та проаналізувавши основні методи штучного інтелекту, можна зробити висновок, що для задачі медичного діагностування за аналізом крові за умови потреби у високій достовірності та швидкодії обробки інформації найкращим варіантом основи інформаційної технології буде застосування нейромереж. Системи логічного висновку та нечіткої логіки потребують занадто багато інформації та утворюють надлишковість у базах знань, що пояснюється специфікою предметної області, коли існує багато варіантів значень показників аналізу крові для однієї хвороби. Експертні системи не спроможні реагувати на зміни так, щоб якість діагностування одразу покращувалась із додаванням нових знань у базу, системам логічного висновку та генетичним алгоритмам складно визначити діагноз пацієнта вірно за умови наявності у нього кількох хвороб одночасно, що потребує додавання окремих алгоритмів для подолання даних недоліків. Штучні нейромережі забезпечують, на відміну від систем нечіткої логіки та

генетичних алгоритмів, високу достовірність обробки інформації, на відміну від імітаційного моделювання та генетичних алгоритмів, – швидкість роботи, а також потребують значно меншої кількості даних, аніж експертні системи та системи, що застосовують імітаційне моделювання.

Приймаючи до уваги те, що найважливішими параметрами для розроблюваної технології є достовірність та швидкість, можна однозначно стверджувати, що застосування нейромереж як теоретичної основи для реалізації інформаційної технології медичного діагностування є найоптимальнішим варіантом для реалізації медичного діагностування за аналізом крові.

Як основа методу обробки інформації буде використана кластеризація даних [15], - розбиття заданої вибірки об'єктів на підмножини, що називаються кластерами, так, щоб кожен кластер складався зі схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися. Специфіка предметної області дозволяє використовувати кластеризацію даних, оскільки для розв'язання задачі визначення медичного діагнозу за аналізом крові в наявності є інформація про показники аналізу крові, що виступають об'єктами кластеризації, та діагнози, що виступають кластерами. Розв'язанням задачі буде правильне віднесення об'єкту, вхідного вектору показників аналізу крові, до кластеру можливого діагнозу.

Класифікація – це процес віднесення невідомого об'єкта до одного з наперед заданих класів. Для вирішення задачі класифікації виявляються ознаки, які характеризують групи об'єктів досліджуваного набору даних - класи; за цими ознаками новий об'єкт можна віднести до того чи іншого класу.

Кластеризація - це автоматичне розбиття елементів деякої множини на групи схожих між собою об'єктів. Синонімами терміну "кластеризація" є "автоматична класифікація", "навчання без учителя" та "таксономія".

Завдання кластеризації подібне до завдання класифікації, є її логічним продовженням, але її відмінність полягає в тому, що класи досліджуваного

набору даних заздалегідь не відомі. Таким чином, кластеризація призначена для розбиття сукупності об'єктів на однорідні групи (кластери або класи). Якщо дані вибірки представити як точки в просторі ознак, то завдання кластеризації зводиться до визначення "локалізації точок". Метою кластеризації є пошук існуючих структур даних. Кластеризація є описовою процедурою, вона не робить ніяких статистичних висновків, але дає можливість провести ознаковий аналіз та визначити структуру даних.

У штучних нейронних мережах під поняттям кластер розуміється підмножина «близьких один до одного» (схожих між собою) об'єктів із множини векторів-характеристик. Отже, кластер характеризує групу об'єктів, що мають загальні властивості [15].

## 2.2 Аналіз процесу кластеризації

Ефективність та якість процесу кластеризації залежить від обраного методу кластеризації, а отримані результати потребують інтерпретації, дослідження властивостей і вивчення характеристик об'єктів для можливого точного опису отриманих кластерів.

Кластеризація даних складається з таких етапів:

### 1) Визначення властивостей (характеристик).

Спочатку вибираються властивості, які характеризують дані об'єкти. Ними можуть бути кількісні характеристики (координати, інтервали та ін.), якісні характеристики (колір, статус, звання та ін.) та ін. Потім застосовують спроби зменшення розмірності простору векторів, тобто виділення найважливіших властивостей об'єктів. Зменшення розмірності пришвидшує процес кластеризації, але у деяких випадках дозволяє візуально оцінювати результат. Потім виділені характеристики нормалізуються. Далі всі об'єкти подаються у виді характеристичних векторів. В даній роботі використовується ототожнення об'єкту та його характеристичного вектора .

### 2) Вибір метрики.

Другим етапом кластеризації є вибір метрики, по якій буде визначатись подібність об'єктів. Метрика обирається залежно від:

- розташування об'єктів у просторі;
- неявних характеристик кластерів.

Якщо всі координати об'єкта дійсні та неперервні, а кластери являють собою щось подібне до гіперсфери, то використовується класична евклідова метрика, математична формула якої подається такою залежністю (2.1):

$$d_2(x_i, x_j) = (\sum_{k=1}^d (x_{i,k} - x_{j,k})^2)^{\frac{1}{2}} = \|x_i - x_j\|_2 \quad (2.1)$$

### 3) Представлення результатів.

Результати кластеризації повинні бути представлені в зручному для обробки вигляді, щоб здійснити оцінку якості кластеризації. Зазвичай використовується один з таких способів:

- подання кластерів набором характерних точок;
- подання кластерів центроїдами;
- подання кластерів їх обмеженнями.

Різні методи кластеризації можуть утворювати різні кластери, і це є нормальним явищем. Однак, створення майже схожих кластерів різними методами говорить про правильність кластеризації [14,15].

## 2.3 Обґрунтування вибору методу кластеризації

У результаті застосування різних методів кластеризації даних можуть бути отримані кластери даних різної конфігурації. Наприклад, можливо отримати кластери ланцюгового типу. Це коли кластери представлено довгими "ланцюгами", кластери мають подовжену форму та інші. Деякі методи кластеризації можуть утворювати кластери довільної форми, що задається користувачем. Різні методи кластеризації можуть прагнути до

створення кластерів певних розмірів (наприклад, тільки великих чи тільки малих), або дозволяти наявність у наборі даних кластерів різної величини. Деякі методи особливо чутливі до викидів або шумів, інші – менш чутливі. У результаті використання різних методів кластеризації можуть бути отримані неоднакові результати по кількості та розмірах кластерів і це є нормальним явищем та особливістю роботи певного алгоритму. Всі ці особливості потрібно враховувати при виборі методу кластеризації даних. На теперішній час розроблено більше сотні різних алгоритмів кластеризації [15,16].

При виборі методу кластеризації для даної задачі було розглянуто такі відомі методи:

- алгоритм k-means та його похідні;
- ієрархічні алгоритми;
- дерево мінімального покриття;
- нечітка кластеризація;
- метод найближчого сусіда;
- генетичні алгоритми.
- Нейромережевий метод;

Для вирішення завдань кластеризації все частіше використовують саме штучні нейронні мережі.

У такому нейромережевому підході є ряд переваг та особливостей:

- штучні нейромережі в силу своєї природи легко працюють в розподілених системах з великою паралелізацією;
- оскільки штучні нейромережі налаштовують свої вагові коефіцієнти, ґрунтуючись на вхідних даних, то це допомагає зробити вибір вагомих характеристик менш суб'єктивним.

Існує велика кількість парадигм штучних нейромереж, наприклад, перцептрон, LVQ – мережі, радіально-базисні мережі, саморганізувальні мапи Кохонена, які можна використовувати для реалізації процесу кластеризації. Але найкраще себе зарекомендувала саме мережа під назвою карта Кохонена,



що саморганізується.

Всі зазначені вище алгоритми мають свої і переваги, і недоліки, на основі яких буде зроблений подальший вибір для реалізації методу діагностування із використанням показників загального аналізу крові.

У методу *k-means* основним недоліком є те, що потрібно задавати заздалегідь *k* - кількість кластерів та еталонів, що не завжди є можливим зробити раціонально. Метод дуже чутливий до початкових наближень значень центрів кластерів, при визначенні середнього значення кластера можливі похибки, для точного визначення результату може бути недостатньо заданої кількості ітерацій у випадку, якщо стабілізація центрів кластерів відбувається повільно, тому швидкодія алгоритму досить невелика через суттєву кількість розрахунків.

Ієрархічні алгоритми іноді ще іменують структурним чи геометричним підходом, який володіє більшою формалізацією поняття близькості. Всі вони спираються на початкову «гіпотезу компактності»: всі близькі об'єкти у просторі об'єктів повинні відноситися до одного кластера, а всі різні об'єкти відповідно повинні ставитися в різні кластери, що може призвести до значної похибки при кластеризації даних.

Метод найближчого сусіда швидко виконується, простий у реалізації, але, як і інші «жадібні» алгоритми, може приводити до неоптимальних рішень.

Метод дерева мінімального покриття може виділяти кластери довільної форми, але має громіздку структуру, що ускладнює процес програмної реалізації та подальшого імовірного застосування програмного засобу у галузі телемедицини.

Нечіткі алгоритми мають велику обчислювальну складність та можуть не підійти для нашої задачі, якщо наперед невідома кількість кластерів або необхідно однозначно віднести кожен об'єкт до певного кластера, що суттєво ускладнює процес діагностування у одного пацієнта кількох хвороб. При

цьому виникає проблема невизначеності об'єкту, який суттєво віддалений від центрів усіх кластерів.

Для реалізації процесу кластеризації даних нейромережею Кохонена, передбачається навчання нейромережі без учителя та самоорганізація нейромережі. Існує значна простота в реалізації та гарантоване одержання відповіді після проходження даних по шарах мережі. Недоліками методу є робота тільки з числовими даними, мінімізація обсягу мережі та необхідність задання числа кластерів, що суттєво не впливають на якість реалізації запропонованого підходу.

Генетичні методи відрізняються від традиційних алгоритмів, оскільки обробляють не числові значення параметрів самого завдання, а їх закодовану форму, здійснюють пошук рішення, виходячи не з єдиної початкової точки, а з деякої їх популяції, застосовують цільову функцію, а не її похідні або іншу додаткову інформацію, використовують ймовірнісні, а не детерміновані правила вибору. Крім того, вони мають значні недоліки - не гарантують знаходження глобального рішення за прийнятний час, не гарантують, що виявлене рішення буде оптимальним і мають переваги над іншими методами лише при великих розмірах завдань і при відсутності упорядкованості у вхідних даних, коли альтернативою є алгоритм повного перебору варіантів [15].

Розроблювана програма медичного діагностування пацієнтів має займати невеликий об'єм, забезпечувати велику швидкодію роботи та точність обробки числових даних. Отже, обраний метод повинен мати негроміздку структуру, забезпечувати високу достовірність медичного діагностування при програмній реалізації. Найкращим варіантом у даній ситуації є використання нейромереж, а саме - нейромережі Кохонена.

## 2.4 Структура, математична модель та порядок функціонування нейромережі Кохонена

Штучна нейронна мережа Кохонена має неконтрольоване навчання та навчальна множина складається лише із значень вхідних змінних (без відповідних значень вихідних змінних). Нейромережа виділяє кластери в навчальних даних і розподіляє дані по відповідних кластерах. Якщо в подальшому нейромережа стикається з набором даних, який не схожий на жоден із вивчених зразків, вона відносить його до нового кластеру. Якщо дані містять мітки класів, то нейромережа спроможна вирішувати завдання класифікації. Нейромережі Кохонена можна також використовувати і у задачах, де класи відомі. Тоді перевага буде у спроможності нейромережі виявляти схожість між різними класами [16, 17].

Нейромережа Кохонена має лише два шари: вхідний і вихідний. Її також називають самоорганізованою мапою. Елементи самоорганізованої мапи розташовуються в певному просторі, як правило, двовимірному (рис. 2.1).

Нейромережа Кохонена навчається методом послідовних наближень. Почавши з обраного випадковим чином початкового розташування центрів, мережа поступово покращується у завданні кластеризації навчальних даних.

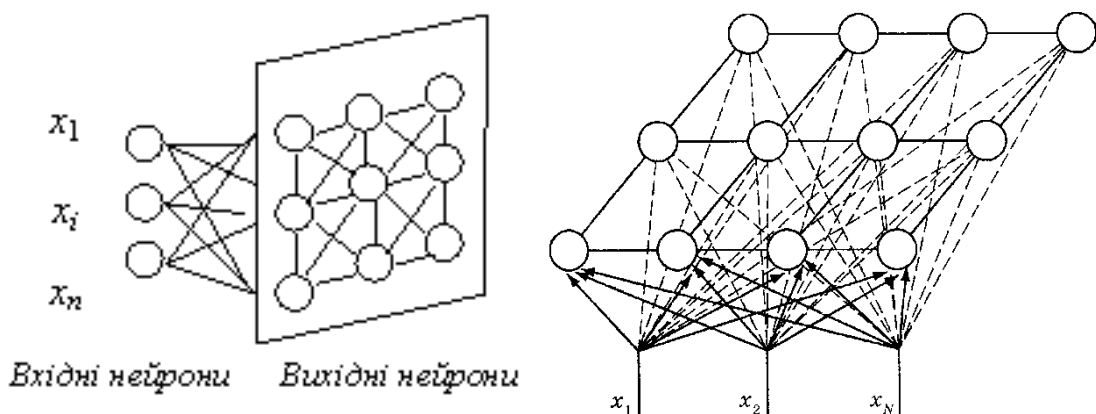


Рисунок 2.1 – Двовимірна структура нейромережі Кохонена

Основний ітераційний алгоритм Кохонена послідовно проходить через низку епох, на кожній з яких опрацьовується один навчальний приклад. Вхідними сигналами нейромережі є вектори дійсних чисел, які послідовно пред'являються нейромережі. Бажані вихідні сигнали наперед не визначаються. Після пред'явлення мережі достатнього числа вхідних векторів, синаптичні ваги нейронів мережі відповідають кластерам. Крім того, ваги організовуються таким чином, що топологічно близькі нейрони є чутливими до подібних вхідних сигналів.

Для реалізації алгоритму потрібно визначити окіл нейрона-переможця (зону сусідства нейронів). На рис. 2.2 показано зони топологічного сусідства нейронів мережі на карті ознак у різні моменти часу.  $NE_j(t)$  - множина нейронів, які вважаються сусідами нейрона  $j$  у момент часу  $t$ . Зони сусідства з часом зменшуються.

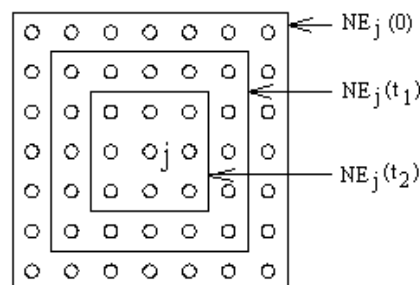


Рис. 2.2 - Зони топологічного сусідства нейронів на карті ознак у різні моменти часу

функціями сусідства нейронів можуть бути потенційно визначені такі:

а) функція «мексиканський капелюх», що обчислюється за формулою (рис. 2.3);

$$f_{gauss2}(d, \sigma) = \left(1 - \left(\frac{d}{\sigma}\right)^2\right) \cdot e^{-\left(\frac{d}{\sigma}\right)^2}, \quad (2.2)$$

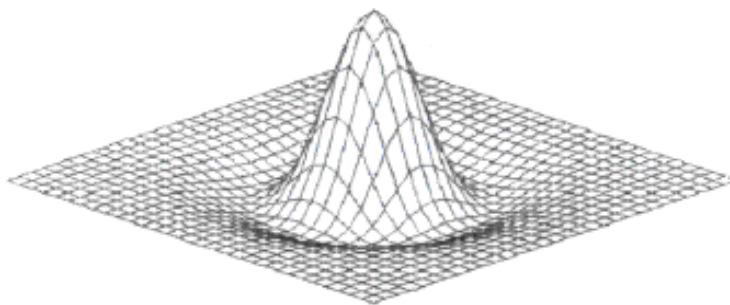


Рисунок 2.3 – Функція «мексиканский капелюх»

б) дзвоноподібна функція Гауса, що обчислюється за формулою (рис. 2.4);

$$f_{gauss1}(d, \sigma) = e^{-\frac{d^2}{2\sigma^2}}, \quad (2.3)$$

де  $\sigma^2$  - дисперсія відхилення;

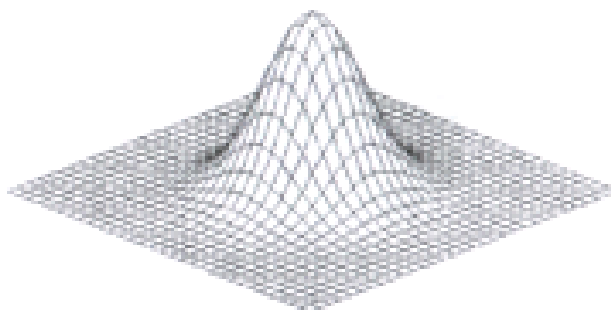


Рисунок 2.4 – Дзвоноподібна функція Гауса

в) косинусоїдна функція, що обчислюється за формулою (рис. 2.5);

$$f_{\cos}(d, \sigma) = \begin{cases} \cos\left(\frac{d\pi}{2\sigma}\right), & d < \sigma; \\ 0, & d \geq \sigma; \end{cases} \quad (2.4)$$

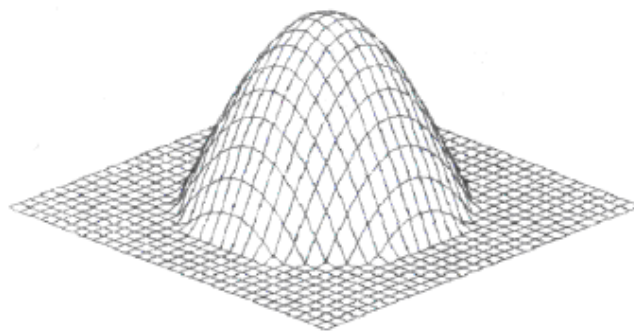


Рисунок 2.5 – Косинусоїдна функція

г) циліндрична функція, що обчислюється за формулою (рис. 2.6);

$$f_{cylinder}(d, \sigma) = \begin{cases} 1, & d < \sigma; \\ 0, & d \geq \sigma; \end{cases} \quad (2.5)$$

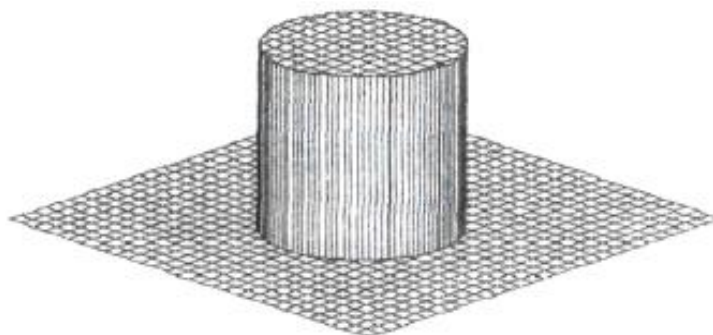


Рисунок 2.6 – Циліндрична функція

д) конусоподібна функція, що обчислюється за формулою (рис. 2.7);

$$f_{cone}(d, \sigma) = \begin{cases} 1 - \frac{d}{\sigma}, & d < \sigma; \\ 0, & d \geq \sigma; \end{cases} \quad (2.6)$$

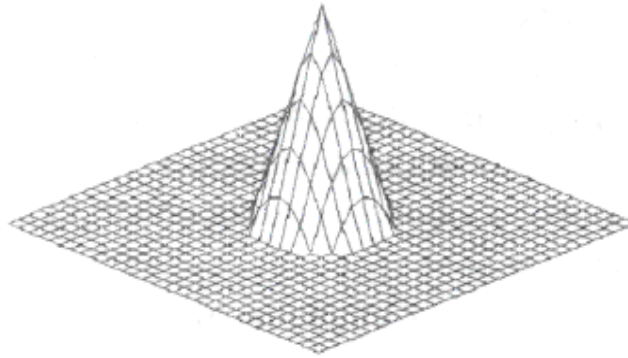


Рисунок 2.7 – Конусоподібна функція

е) функція «французький капелюх», що обчислюється за формулою (рис. 2.8);

$$h(p) = \begin{cases} 1, & |p| \leq \alpha \\ -\frac{1}{3}, & \alpha < |p| \leq 3\alpha \\ 0, & |p| > 3\alpha \end{cases} \quad (2.7)$$

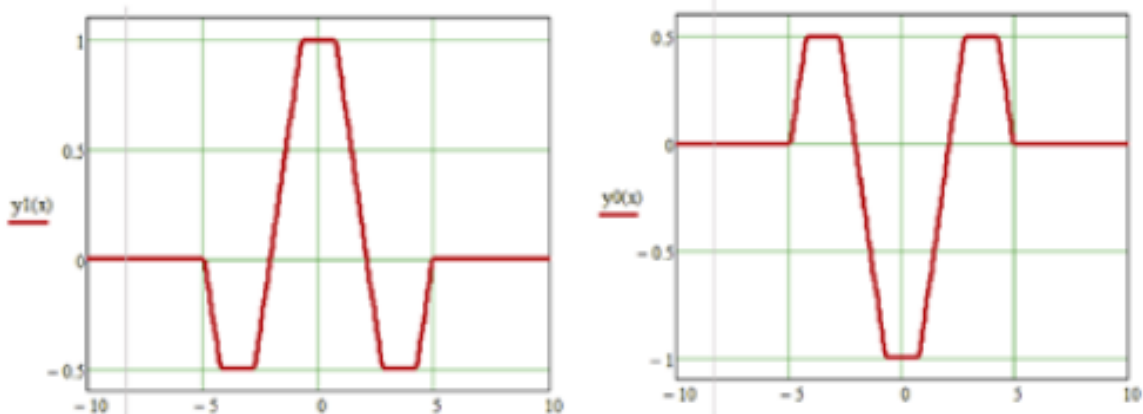


Рисунок 2.8 – Функція «французький капелюх»

Загальний алгоритм функціонування нейромережі Кохонена:

1. Ініціалізація (створення) нейромережі. Ваговим коефіцієнтам нейронів мережі присвоюються малі випадкові значення.
2. Пред'явлення нейромережі нового вхідного вектору.

3. Обчислення відстані від вхідного вектора до ваг всіх нейронів мережі:

Відстані  $d_j$  від вхідного вектору до ваг кожного нейрона  $j$  обраховуються за формулою :

$$d_j = \sum_{i=1}^N (x_i(t) \cdot w_{ij}(t))^2 \quad (2.8)$$

де  $x_i$  -  $i$ -ий елемент вхідного сигналу в момент часу  $t$ ,  $w_{ij}(t)$  - вага зв'язку від  $i$ -го елемента вхідного сигналу до  $j$ -го нейрона у момент часу  $t$ .

4. Вибір нейрона з мінімальною відстанню (рис. 2.9):

Обирається нейрон-переможець  $j^*$ , для якого відстань  $d_j$  найменша.

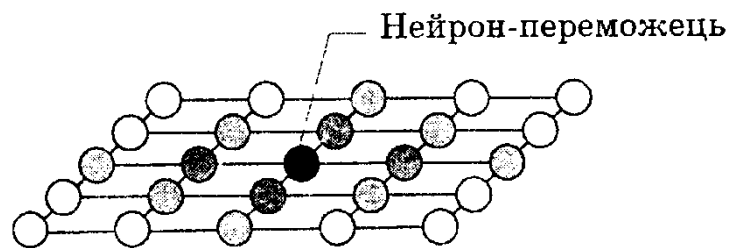


Рисунок 2.9 – Нейрон-переможець у шарі мапи Кохонена

5. Налаштування ваг нейрона  $j^*$  та його сусідів (рис. 2.10):

Робиться налаштування ваг нейрона  $j^*$  і всіх нейронів його околу. Нові значення ваг нейронів обраховуються за формулою .

$$w_{ij}(t+1) = w_{ij}(t) + r(t)(x_i(t) - w_{ij}(t)) \quad (2.9)$$

де  $r(t)$  – коефіцієнт швидкості навчання, яке зменшується з часом (додатне число в діапазоні від 0 до одиниці).



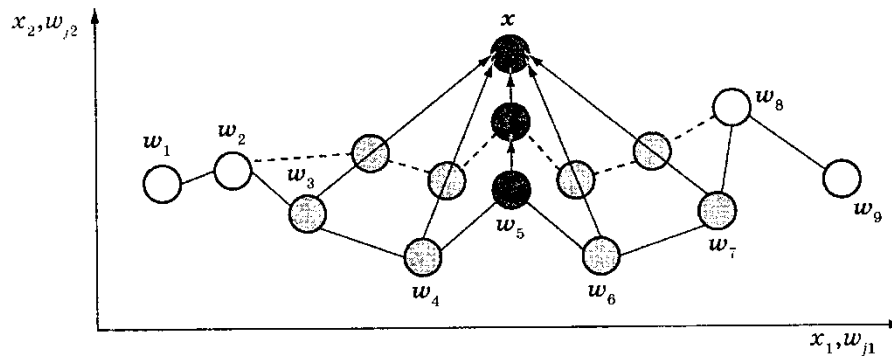


Рисунок 2.10 – Принцип зміни ваг мапи Кохонена при навчанні

### 6. Повернення до кроку 2.

В алгоритмі нейромережі Кохонена застосовується коефіцієнт швидкості навчання, який зменшується поступово для точнішої корекції на новій епосі навчання. У результаті позиція центру кластеру встановлюється в такій позиції, яка задовільним чином кластеризує навчальні приклади, для яких даний нейрон опиняється переможцем.

Топологічна впорядкованість досягається в алгоритмі за допомогою застосування поняття околу. Окіл (інакше - радіус сусідства) - це кілька нейронів, які оточують нейрон-переможець. Відповідно до динаміки швидкості навчання, розмір околу зменшується поступово так, що спочатку до нього входить достатньо велике число нейронів (можливо вся мапа), а на останніх етапах окіл стає нульовим і складається лише з одного нейрона-переможця. В алгоритмі навчання мережі корекція використовується не тільки до нейрона-переможця, але і до всіх нейронів його поточного околу. Як результат такої зміни околу початкові доволі великі ділянки нейромережі мігрують у бік навчальних прикладів. Нейромережа формує укрупнену структуру топологічного порядку, при якій подібні між собою приклади активують групи нейронів, які знаходяться поруч на топологічній мапі. З кожною новою епохою навчання параметр швидкості навчання і радіус околу зменшуються, тим самим всередині ділянок мапи відбуваються більш тонкі розходження, що, врешті рещт, призводить до точнішого налаштування ваг

кожного нейрона. Часто процес навчання зумисне розбивають на два етапи: більш короткий, з великим значенням швидкості навчання і великих околів, і більш тривалий з малим значенням швидкості навчання і нульовими або майже нульовими околами [16, 17].

## 2.5 Алгоритм нейромережевого медичного діагностування за аналізом крові

Загальний алгоритм нейромережевого медичного діагностування за аналізом крові на основі нейромережі Кохонена наведено на рис. 2.11.

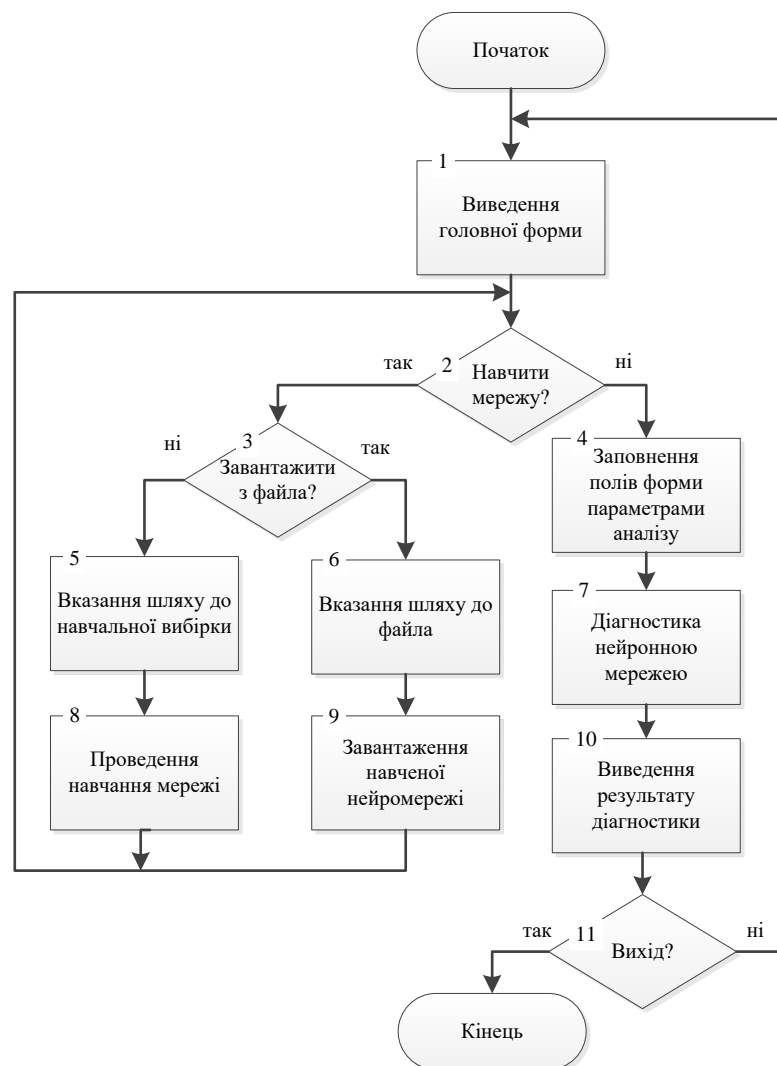


Рисунок 2.11 – Загальний алгоритм нейромережевого медичного діагностування за аналізом крові на основі нейронної мережі Кохонена

Архітектура розробленої програми медичного діагностування за аналізом крові на основі нейромережі Кохонена складається з двох шарів нейронів (рис. 2.12). Перший шар не має фактичних нейронів, а він тільки отримує вхідні сигнали і передає їх на другий шар. Кожен штучний нейрон другого шару має свій вектор ваг, розмірність якого відповідає розмірності вхідного шару. Нейрони з'єднуються латеральними зв'язками з сусідніми нейронами, що визначає структуру і топологію мапи Кохонена. Нейрони вихідного шару ще називаються кластерними елементами, а їх число визначає максимальне число груп (кластерів), на які система може розділити вхідні дані. Збільшуючи число нейронів другого шару, можна збільшувати ступінь деталізації результатів кластеризації. Оскільки планується, що програма буде визначати 12 груп хвороб для чоловіків та 12 для жінок, то доцільним буде обрання кількості нейронів шару Кохонена не менше 24 (по одному нейрону на кожен кластер). А оскільки для надійної роботи нейромережі треба передбачити якусь кількість «запасних» нейронів, то оберемо число нейронів шару Кохонена 30.

Оскільки стандартна реалізація нейромережі Кохонена не передбачає визначеного числа кластерів, та початкові центри кластерів набувають випадкові малі значення. Було проведено модифікацію даної нейромережі, яка полягала у попередньому визначенні числа кластерів та їх центрів, які відповідають нормальним та відхиленням від норми значенням параметрів загального аналізу крові (табл. 1.1 та 1.2).

Нейромережа має два шари: вхідний шар та вихідний (шар Кохонена) (рис.2.12). Шар Кохонена складається з деякого числа  $n$  паралельно діючих лінійних нейроелементів. Всі вони мають однакове число входів  $m$  та отримують на свої входи один і той самий вектор вхідних сигналів  $x=(x_1\dots x_m)$ . На виході  $j$ -го лінійного нейроелемента отримаємо сигнал, який розраховується за такою формулою:

$$y_j = w_{j0} \sum_{i=1}^m w_{ji} x_i \quad (2.10)$$

де  $w_{ji}$  — ваговий коефіцієнт  $i$ -го входу  $j$ -го нейрону,  $w_{j0}$  — пороговий коефіцієнт.

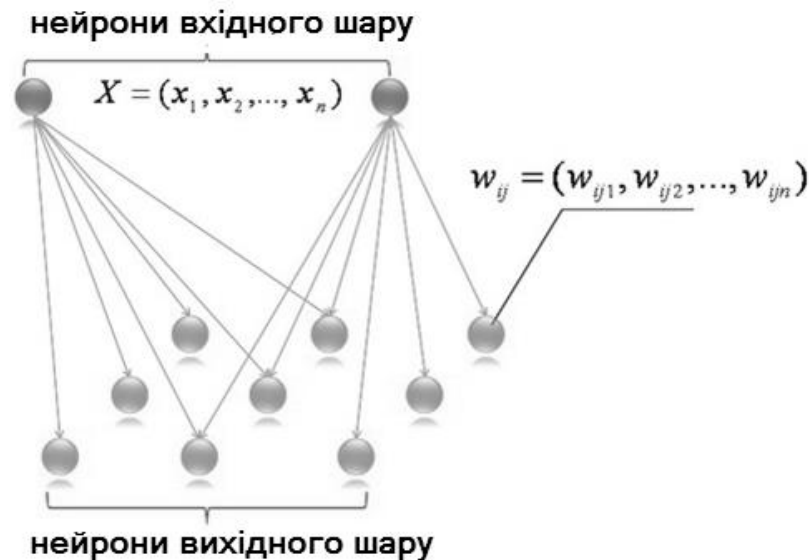


Рисунок 2.12 – Структура нейромережі Кохонена розроблюваної програми

Після проходження шару лінійних нейроелементів сигнали поступають на обробку за правилом «переможець забирає все» (WTA): серед вихідних сигналів  $y_i$  знаходиться максимальний; його номер  $j_{\max} = \arg \max_j \{y_j\}$ . Остаточно, на виході мережі сигнал з номером  $j_{\max}$  дорівнює одиниці, а всі інші — нулю. Якщо максимум одночасно досягається для кількох  $j_{\max}$ , то всі відповідні сигнали приймають значення одиниці.

Навчання розроблюваної нейромережі відбувається таким чином. На початку роботи визначається число кластерів та їх центри. Дана інформація вже визначена наперед: число кластерів становить 30, початкові центри кластерів відповідають наборам параметрів хвороб для визначених груп хвороб. Після цього певний вхідний вектор із набору навчальних обирається

та встановлюється на входах нейромережі. Після цього відмінності між вхідним вектором та всіма векторами ваг мережі обчислюються за формулою:

$$D_{ij} = |X^l - W_{ij}| = \sqrt{(x_1 - w_{ij1})^2 + \dots + (x_n - w_{ijn})^2} \quad (2.11)$$

де  $i$  та  $j$  – координати нейронів у вихідному шарі. Після цього нейромережа обирає нейрона-переможця із переліку визначених центрів кластерів, тобто такого, щоб його ваговий вектор був подібний на вхідний за формулою :

$$D(k_1, k_2) = \min_{i,j} D_{i,j} \quad (2.12)$$

де  $k_1$  та  $k_2$  – координати нейрона-переможця. Після цього відбувається корекція вагових векторів переможця та сусідніх з ним нейронів. Близні до переможця нейрони визначаються функцією топологічного сусідства «Мексиканський капелюх», яка розраховується за формулою :

$$h(p, t) = \exp\left(-\frac{p^2}{\sigma^2(t)}\right) \left(1 - \frac{2}{\sigma^2(t)} p^2\right) \quad (2.13)$$

де  $p$  – відстань до ваг нейрона переможця, що знаходиться за формулою :

$$p = \sqrt{(k_1 - i)^2 + (k_2 - j)^2} \quad (2.14)$$

де  $\sigma$  – функція, яка визначає радіус сусідства. На початку функціонування програми радіус сусідства включає весь простір сенсорного поля (гратки), але з часом його значення зменшується.

Після обчислення топологічної функції сусідства ваги усіх нейронів переобраховуються за формулою (2.15).

$$W_{ij}(t+1) = W_{ij}(t) + \alpha(t)h(p,t)(X^l(t) - W_{ij}(t)) \quad (2.15)$$

де  $\alpha(t)$  – функція швидкості навчання, що змінюється з часом також.

Якщо нейрон є переможцем або його сусідом, то його ваговий вектор оновлюється або інакше лишається незмінним. На кожному кроці нейромережа визначає той нейрон, чий ваговий вектор найбільш схожий на вхідний, та коригує його ваги та ваги нейронів-сусідів, щоб наблизити їх значення до вхідного вектора (рис. 2.13).

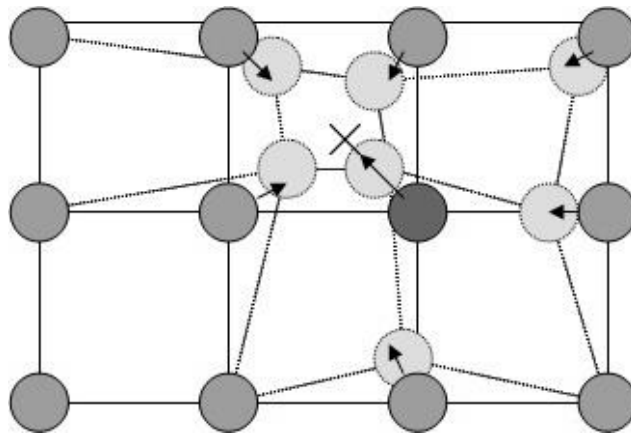


Рисунок 2.13 – Оновлення ваг нейрона-переможця та його сусідів та «підштовхування» їх в сторону вхідного вектора, що позначений на рисунку «x». Суцільні та пунктирні лінії відображують ситуацію до та після оновлення, відповідно.

Кожний вхідний вектор з навчальної вибірки подається на нейромережу і навчання триває або деяку фіксовану кількість циклів, або доки різниця між вхідним і ваговим векторами досягне наперед заданого малого значення  $\varepsilon$ . Різниця ваг сусідніх нейронів зменшується з плином часу, і, отже, вони організовуються у кластери (групи), які відповідають одному з класів векторів навчальної множини.

Для оптимальних розрахунків параметрів запропонованої нейромережі, яка використовується при розробці програми, було введено також процедуру попередньої обробки вхідної інформації, при якій числові величини ознак, що утворюють вхідний вектор, приводяться до деякого заданого діапазону. Тут необхідна нормалізація, оскільки вхідні величини ознак змінюються в достатньо широкому діапазоні, тому існує імовірність некоректної роботи нейромережі з такими даними. Так, у одному вхідному векторі можуть знаходитися значення, які відрізняються один від одного на декілька порядків: колірний показник – 0,85-1,15%, стандартний рівень гемоглобіну – 120-180г/л. Такий дисбаланс між величинами показників може викликати нестійкість роботи мережі, погіршувати результати навчання і уповільнювати процес навчання. Після нормалізації всі величини вхідних ознак будуть приведені до певного вузького діапазону, а саме -  $[0 \dots 1]$ , що дозволяє нейромережі працювати з даними коректніше [17].

Для реалізації запропонованого підходу до медичного діагностування була використана нормалізація вхідних даних, яка виконується за формулою :

$$y = \frac{(x - x_{\min})(d_2 - d_1)}{x_{\max} - x_{\min}} + d_1 \quad (2.16)$$

де:  $x$  - значення, які підлягають нормалізації;

$[d_1, d_2]$ - інтервал значень  $x$ ;

$[x_{\min}, x_{\max}]$ - інтервал, до якого буде зведено значення  $x$ .

Таким чином, було розроблено алгоритм, за яким буде у подальшому працювати програма медичного діагностування на основі нейронної мережі Кохонена.

## **2.6 Висновок до розділу 2**

У розділі було обґрунтовано вибір нейронної мережі Кохонена та процесу кластеризації для інформаційної технології медичного діагностування. Спроектовано структуру математичну модель та порядок функціонування нейронної мережі Кохонена. Розроблено структуру процесів обробки інформації інформаційної технології. Розроблено алгоритм роботи нейромережевого медичного діагностування за аналізом крові.



## **3 ПРОГРАМНА РЕАЛІЗАЦІЯ НЕЙРОМЕРЕЖЕВОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МЕДИЧНОГО ДІАГНОСТУВАННЯ**

### **3.1 Обґрунтування вибору мови та середовища програмування**

На теперішній час існує багато програмних середовищ, які підтримують можливість реалізації нейронних мереж, основними з яких є середовища та мови програмування MatLAB, Deductor, C++, Java, C#, кожна з яких має свої переваги та недоліки.

Програмне середовище MatLAB підтримує у повній мірі за допомогою вбудованих бібліотек створення нейромереж, але має досить негнучку структуру, обмежені функціональні можливості для використання та створення графічного інтерфейсу для написаної програми. Додатки, створені за допомогою цього середовища, зазвичай, є складними у редагуванні та при запуску на різних платформах.

Програмне середовище Deductor передбачає вбудовану реалізацію нейромереж, число яких в даній програмі обмежено. Дане середовище призначене, насамперед, для користувацьких цілей та може використовуватись, зокрема, як програми для тестування написаного коду нейромережі.

Мова програмування Java є досить простою, безпечною та передбачає запуск на довільних платформах. Вона має багато бібліотек, що дозволяють повністю реалізувати будь-яку нейромережу, але у той самий час програми, написані цією мовою, є громіздкими та потребують багато ресурсів та часу для виконання, що може бути некорисним при створенні програмного забезпечення, орієнтованого на звичайного користувача.

Мови програмування C++ та C# мають подібні до мови Java переваги, але бібліотеки цих мов є більш функціональними саме для створення нейромереж. Програми, написані на C# та C++ виграють у швидкості та

запуску на різних платформах, а також легко можуть бути модифікованими за бажанням.

Всі наведені вище програмні середовища мають можливості бути використаними для створення розроблюваного програмного забезпечення. У той самий час програмне середовище, яке обирається, має бути, по-перше, кросплатформним, по-друге, створені за його допомогою програми повинні легко редагуватись та модифікуватись за бажанням користувача, та, по-третє, повинні мати порівняно велику швидкість виконання програм [14, 17, 18].

Таким чином, взявши до уваги зазначені вимоги до мови програмування та оцінивши переваги та недоліки розглянутих програмних середовищ, було обрано мову програмування C# та середовище розробки Visual Studio. Оскільки мова програмування C# має більше можливостей для реалізації нейромереж, ніж C++, а також для неї було створено багато фреймворків та бібліотек нейромереж, незважаючи на незначну перевагу по швидкості C++, саме мову C# було обрано для реалізації програмного забезпечення медичного діагностування пацієнтів на основі нейромережі Кохонена.

Також була використана спеціалізована бібліотека Accord.NET [19] — це фреймворк для наукових застосувань в .NET. Фреймворк включає набір бібліотек, доступних у вихідному коді, а також через виконувані установники та пакети NuGet. Основні сфери застосування : чисельна лінійна алгебра, числова оптимізація, статистика, машинне навчання, штучні нейромережі, обробка сигналів і зображень та допоміжні бібліотеки (для побудови графіків і візуалізації). Спочатку фреймворк був створений для розширення можливостей AForge.NET Framework, але з тих пір він поглинув у собі AForge.NET. Нова версія об'єднала обидві платформи під ім'ям Accord.NET.

### 3.2 Реалізація основних компонентів програми медичного діагностування на основі нейронної мережі Кохонена

Опишемо оператори, функції та класи, що використовуються для представлення нейромереж за допомогою мови С#.

Приклад класів для представлення елементів мережі:

```
// Вхід
public class Input
{
    // Зв'язки з нейронами
    public Link[] OutgoingLinks;
}
// Зв'язок входу з нейроном
public class Link
{
    // Нейрон
    public Neuron Neuron;
    // Вага зв'язку
    public double Weight;
}
public class Neuron
{
    //Всі входи нейрону
    public Link[] IncomingLinks;
    // Накопичуваний нейроном заряд
    public double Power { get; set; }
}
```

Для нейромережі Кохонена приклад визначення масивів вхідних векторів та ваг нейронів:

```
public class KohonenNetwork
{
    private readonly Input[] _inputs;
    private readonly Neuron[] _neurons;
    ...
}
```

Приклад розрахунку ваг нейронів:

```
// Надати вектор на вхід нейронної мережі
public int Handle(int[] input)
{
    for (var i = 0; i < _inputs.Length; i++)
    {
        var inputNeuron = _inputs[i];
        foreach (var outgoingLink in inputNeuron.OutgoingLinks)
        {
            outgoingLink.Neuron.Power += outgoingLink.Weight * input[i];
        }
    }
    var maxIndex = 0;
    for (var i = 1; i < _neurons.Length; i++)
    {
        if (_neurons[i].Power > _neurons[maxIndex].Power)
            maxIndex = i;
    }
}
```

```

//Зняти імпульс з усіх нейронів
foreach (var outputNeuron in _neurons)
{
    outputNeuron.Power = 0;
}
return maxIndex;
}

```

Приклад процедури навчання:

```

public void Study(int[] input, int correctAnswer)
{
    var neuron = _neurons[correctAnswer];
    for (var i = 0; i < neuron.IncomingLinks.Length; i++)
    {
        var incomingLink = neuron.IncomingLinks[i];
        incomingLink.Weight = incomingLink.Weight + 0.5 * (input[i] -
incomingLink.Weight);
    }
}

```

Основними класами програми медичного діагностування пацієнтів на основі нейромережі Кохонена є `Neuron` та `NeuralNetwork`, які представлені на рисунку 3.1. Наведену діаграму класів побудовано у середовищі `Rational Rose`. Вона демонструє поля (Fields) кожного класу, його атрибути (Properties), методи (Methods) та події (Events). Ці класи знаходяться у відношенні залежності, оскільки клас `NeuronNetwork`, оголошує клас `Neuron` та створює 86 екземплярів його, тобто існування класу `Neuron` залежить від існування класу `NeuralNetwork`.

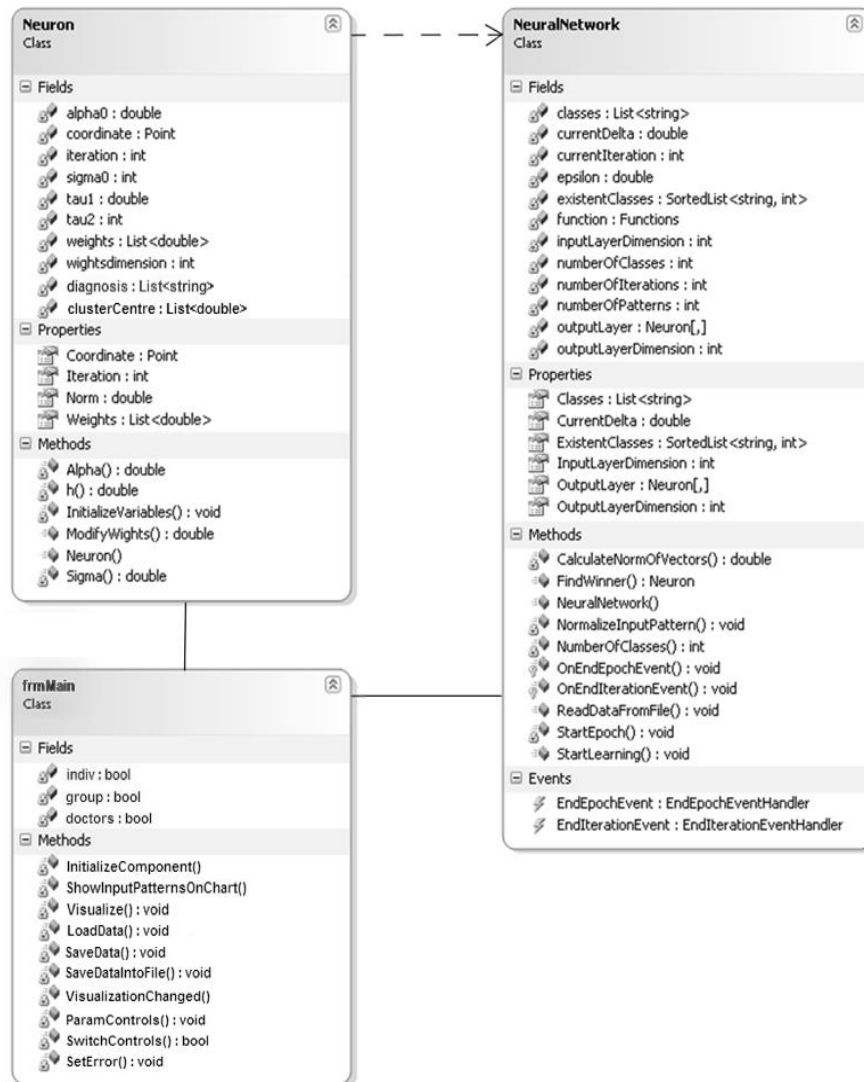


Рисунок 3.1 – Діаграма класів SOFM.Neuron та SOFM.NeuralNetwork

Кожен клас Neuron має поле Coordinate, яке показує позицію нейрона в двовимірній ґратці. Поле Weights – вектор ваг виду generic, яке містить значення типу double. Клас Neuron може виконувати операцію ModifyWeights, яка переобчислює вектор ваг нейрону відповідно до обраної функції сусідства «Мексиканський капелюх» (h), формулу якої занесено у код програми, відстань до нейрона-переможця або значення епсілон, яке за замовченням дорівнює  $10^{-4}$ .

Клас NeuralNetwork містить двовимірну ґратку нейронів та може виконувати певний набір операцій:

```

public NeuralNetwork(int 30, int numberOfIterations, double epsilon, Functions f)
// Створює нейронну мережу, що містить 30 нейронів на другому шарі, та
// визначену кількість ітерацій, епсілон та функцію топологічного сусідства
«Мексиканський капелюх».
public void ReadDataFromFile(string inputDataFileName)
// Читає інформацію вхідних даних з певного файлу
public void StartLearning()
// Починає навчати мережу.

```

Обчислення функції «Мексиканський капелюх» відбувається за такою процедурою:

Function.MexicanHat:

```

{
    distance = Math.Sqrt(Math.Pow((winnerCoordinate.X - coordinate.X), 2) +
Math.Pow((winnerCoordinate.Y - coordinate.Y), 2));
    result = Math.Exp(-(distance * distance) / Math.Pow(Sigma(iteration), 2)) * (1 -
(2 / Math.Pow(Sigma(iteration), 2)) * (distance * distance));
    break;
}

```

Всі шаблони з навчальної множини продовжують подаватися доти, доки вектор ваг не буде переобчислюватись на невелику величину, яка не відрізняється від початкової ( $currentEpsilon \leq Epsilon$ ).

```

public void StartLearning()
{
    int iterations = 0;
    while (currentEpsilon > epsilon)
    {

```

```

List<List<double>>patternsToLearn = new
List<List<double>>(numberOfPatterns);
foreach (List<double> pArray in patterns)
    patternsToLearn.Add(pArray);
Random randomPattern = new Random();
List<double> pattern = new List<double>(inputLayerDimension);
for (int i = 0; i < numberOfPatterns; i++)
{
    pattern = patternsToLearn[randomPattern.Next(numberOfPatterns - i)];

    StartEpoch(pattern);

    patternsToLearn.Remove(pattern);
}
iterations++;
OnEndIterationEvent(new EventArgs());
}
}

```

Метод типу `private StartEpoch()` передбачає визначення нейрона-переможця та початок процесу перед обрахуванням ваг нейронів на всіх зовнішніх шарах `outputLayer`.

```

private void StartEpoch(List<double> pattern)
{
    Neuron Winner = this.FindWinner(pattern);
    currentEpsilon = 0;
    for (int i = 0; i < outputLayerDimension; i++)
        for (int j = 0; j < outputLayerDimension; j++)
        {
            currentEpsilon +=
                outputLayer[i, j].ModifyWeights(pattern, Winner.Coordinate,

```



```

        currentIteration, function);
    }
    currentIteration++;
    currentEpsilon = Math.Abs(currentEpsilon /
        (outputLayerDimension * outputLayerDimension));
    EndEpochEventArgs e = new EndEpochEventArgs();
    OnEndEpochEvent(e);
}

```

Таким чином, епоха навчання визначена як процес подання одного з шаблонів на вхід нейромережі, а ітерація представляє собою набір епох, коли всі шаблони із навчальної вибірки представлені на входи нейромережі. Кожного разу, коли епохи або ітерації закінчуються, відбуваються відповідні події [19, 20].

```

protected virtual void OnEndEpochEvent(EndEpochEventArgs e)
{
    if (EndEpochEvent != null)
        EndEpochEvent(this, e);
}

protected virtual void OnEndIterationEvent(EventArgs e)
{
    if (EndIterationEvent != null)
        EndIterationEvent(this, e);
}

```

Таким чином, у даному розділі були виявлені переваги обраної мови програмування, яка була використана для програмної реалізації

запропонованого методу, визначені класи програмного забезпечення та представлені елементи коду програми, що відображають основні виконувані операції.

### **3.3 Висновок до розділу 3**

У розділі було обгрунтовано вибір мови програмування C# та середовища програмування Visual Studio. Також була використана бібліотека Accord.NET. Було описано реалізацію основних компонентів програми медичного діагностування на основі нейронної мережі Кохонена, розроблено діаграму класів програми.

## 4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ МЕДИЧНОГО ДІАГНОСТУВАННЯ НА ОСНОВІ НЕЙРОННОЇ МЕРЕЖІ КОХОНЕНА

### 4.1 Тестування програми медичного діагностування на основі нейронної мережі Кохонена

Метою роботи розробленої програми є визначення попереднього медичного діагнозу пацієнта за показниками його загального аналізу крові.

Після запуску програми відкриється головне вікно (див. рис.4.1).

Попереднє діагностування захворювань за аналізом крові

Діагностування **Навчання**

Стать:  Ч  Ж      Вік:  До 15  15-20  50+

Введіть результати загального аналізу крові:

Лейкоцити WBC	<input type="text"/>	Тромбоцити PLT	<input type="text"/>
Еритроцити RBC	<input type="text"/>	Швидкість осідання еритроцитів - ШОЕ (ESR),	<input type="text"/>
Гемоглобін HGB	<input type="text"/>	Формула крові	
Гематокрит HCT	<input type="text"/>	Нейтрофіли NEU	<input type="text"/>
Середній об'єм еритроцитів MCV	<input type="text"/>	Лімфоцити LYMPH	<input type="text"/>
Середній вміст гемоглобіну в еритроциті MCH	<input type="text"/>	Моноцити MONO	<input type="text"/>
Середня концентрація гемоглобіну в еритроциті MCHC	<input type="text"/>	Еозінофіли EO	<input type="text"/>
Індекс розподілу еритроцитів RDW-CV	<input type="text"/>	Базофіли BASO	<input type="text"/>

Рисунок 4.1 – Початкове вікно програми

Якщо нейронна мережа Кохонена ще не була навчена, то необхідно вибрати вкладку «Навчання». Після цього з'явиться вікно, зображене на рис.

4.2. У цьому вікні можна обрати одну із двох опцій:

- 1) завантажити вже навчену нейронну мережу Кохонена із файла;

2) провести навчання нейронної мережі Кохонена на нових даних.

У обох випадках з'являється додаткове вікно навігатора для вибору місця знаходження необхідного файлу на диску, як зображено на рис. 4.3.

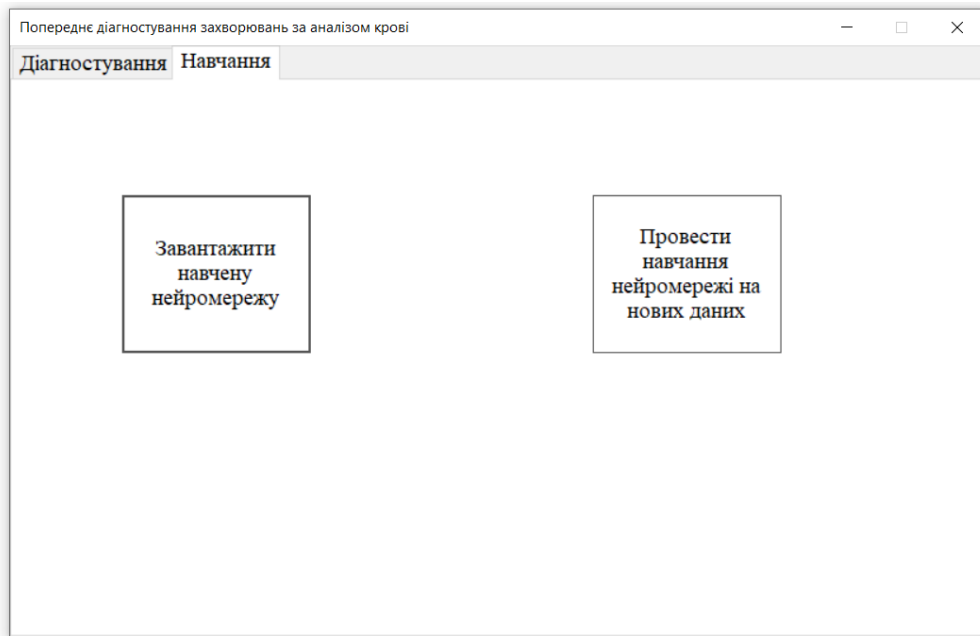


Рисунок 4.2 – Вкладка «Навчання» програми медичного діагностування

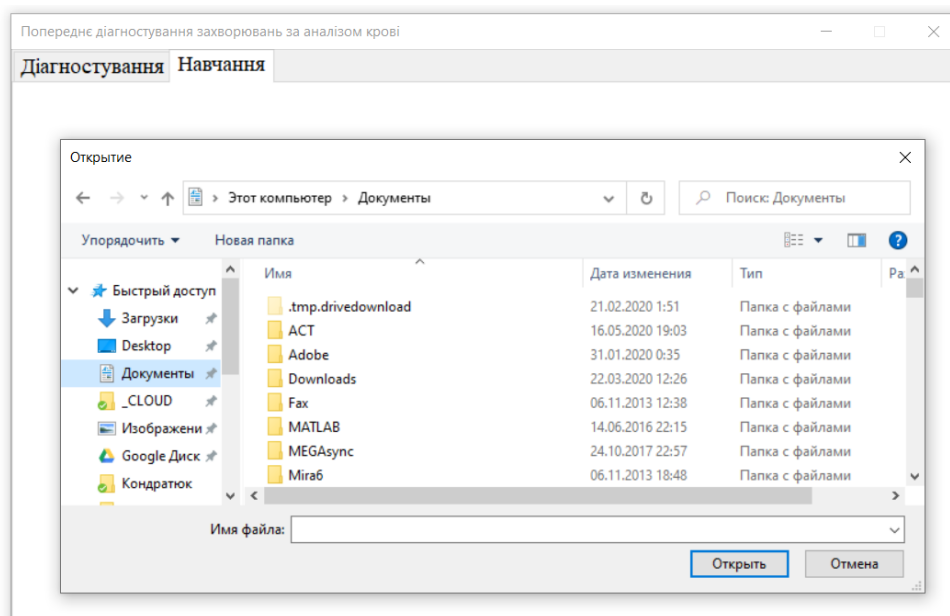


Рисунок 4.3 – Додаткове вікно навігатора для вибору місця знаходження необхідного файлу на диску

Інформація про навчену нейронну мережу зберігається у вигляді файлу формату .txt, який представляє собою масив чисел (матриця ваг зв'язків нейронів мережі) розміром  $m \times n$ , де  $m=17$  (де 17 – кількість входів нейронної мережі, на які подають 15 параметрів загального аналізу крові, параметри статі і віку пацієнта), а  $n$  - число нейронів у мережі Кохонена.

Дані навчальної вибірки також зберігається у вигляді файлу формату .txt, який являє собою масив чисел розміром  $V \times (m+2)$ , де до  $m$  додається 2 параметри: ідентифікатор пацієнта та ідентифікатор класу хвороб, а  $V$  – об'єм навчальної вибірки, тобто число пацієнтів з відомими групами хвороб, на прикладах яких навчається нейронна мережа Кохонена.

Після проведення навчання нейромережа Кохонена буде готова до використання з метою попереднього діагностування пацієнтів і тому можна перейти до вкладки «Діагностування» та заповнити усі поля параметрами пацієнта (його стать, вік) та його аналізу крові (як це наведено на рис. 4.4).

Попереднє діагностування захворювань за аналізом крові			
Діагностування		Навчання	
Стать: <input checked="" type="radio"/> Ч <input type="radio"/> Ж	Вік: <input type="radio"/> До 15 <input checked="" type="radio"/> 15-20 <input type="radio"/> 50+		
Введіть результати загального аналізу крові:			
Лейкоцити WBC	<input type="text" value="5"/>	Тромбоцити PLT	<input type="text" value="210"/>
Еритроцити RBC	<input type="text" value="4.2"/>	Швидкість осідання еритроцитів - ШОЕ (ESR), Формула крові	<input type="text" value="9"/>
Гемоглобін HGB	<input type="text" value="132"/>	Нейтрофіли NEU	<input type="text" value="4"/>
Гематокрит HCT	<input type="text" value="0.44"/>	Лімфоцити LYMPH	<input type="text" value="2"/>
Середній об'єм еритроцитів MCV	<input type="text" value="87"/>	Моноцити MONO	<input type="text" value="0.3"/>
Середній вміст гемоглобіну в еритроциті MCH	<input type="text" value="29"/>	Еозінофіли EO	<input type="text" value="0.4"/>
Середня концентрація гемоглобіну в еритроциті MCHC	<input type="text" value="36"/>	Базофіли BASO	<input type="text" value="0.06"/>
Індекс розподілу еритроцитів RDW-CV	<input type="text" value="7.2"/>	<input type="button" value="Продіагностувати"/>	

Рисунок 4.4 – Заповнення на вкладці «Діагностування» всіх полів із параметрами пацієнта (стать, вік) та аналізу крові.

Далі необхідно натиснути на кнопку «Продіагностувати» і з'явиться вікно з результатом попереднього медичного діагностування пацієнта (див. рис. 4.5).

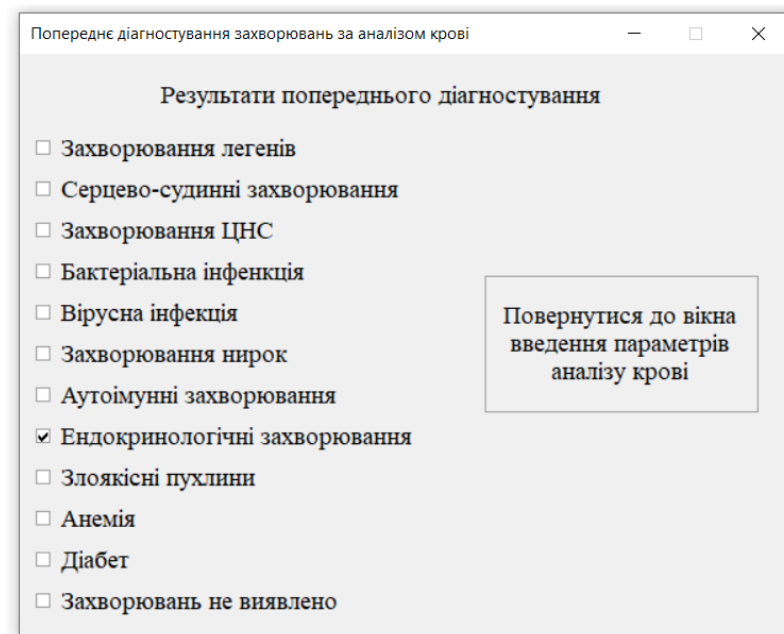


Рисунок 4.5 – Вікно програми з результатом попереднього медичного діагностування пацієнта

Як видно з рис. 4.5, пацієнт має хворобу із групи «Ендокринологічні захворювання». Із цього вікна є можливість повернутися до вікна згідно рис.4.4, натиснувши кнопку «Повернутися до вікна введення параметрів аналізу крові».

Тестування розробленої програми показало її надійну роботу у різних режимах та повну відповідність поставленому завданню.

#### **4.2 Аналіз результатів роботи програми медичного діагностування на основі нейронної мережі Кохонена**

Проведемо експериментальні дослідження програми медичного діагностування на основі нейронної мережі Кохонена з метою перевірки

правильності її роботи та визначення достовірності медичного діагностування у порівнянні з програмою-аналогом. Для проведення експериментів була використані безкоштовні загальнодоступні бази даних показників аналізів крові та відповідних їм хвороб.

Перевіримо достовірність роботи програми попереднього медичного діагностування пацієнтів на основі нейронної мережі Кохонена у порівнянні с програмою-аналогом. Для порівняння оберемо популярну систему онлайн медичного діагностування YourDiagnosis [11], яка базується на медичних алгоритмах. Для діагностування було взято 100 наборів вхідних значень та перевіримо чи вірно вони були розпізнані. Результати тестування подано у табл. 4.1.

Таблиця 4.1 – Результати тестування розробленої програми та програми-аналога YourDiagnosis

Програмний засіб	Кількість пацієнтів у тестовій вибірці	Кількість правильно діагностованих пацієнтів	Кількість невірно діагностованих пацієнтів	Достовірність діагностування пацієнтів
YourDiagnosis	100	84	16	84 %
Розроблена програма	100	91	9	91 %

Із табл. 4.1 видно, що розроблена програма має вищу достовірність медичного діагностування пацієнтів (91%), ніж аналогічна програма (84%), а значить достовірність медичного діагностування пацієнтів покращена на 7%, тобто мета роботи досягнута.

Тестування розробленої програми медичного діагностування пацієнтів на основі нейронної мережі Кохонена показало її надійну роботу. Програма повністю відповідає завданню. Інструкцію користувача наведено у Додатку Г.

### 4.3 Висновок до розділу 4

У четвертому розділі в результаті тестування програми медичного діагностування пацієнтів на основі нейронної мережі Кохонена було доведено її повну працездатність та відповідність поставленому завданню. Розроблена програма має вищу достовірність медичного діагностування пацієнтів (91%), ніж аналогічна програма (84%), а значить достовірність медичного діагностування пацієнтів покращена на 7%, Тобто мета роботи досягнута – достовірність медичного діагностування підвищена.



## 5 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота на тему «Нейромережева інформаційна технологія медичного діагностування» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

### **5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки**

Метою проведення комерційного і технологічного аудиту дослідження за темою «Нейромережева інформаційна технологія медичного

діагностування» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 5.1 [21].

Таблиця 5.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження таблиці 5.1

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 5.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	5	5
2. Ринкові переваги (наявність аналогів)	3	4	3
3. Ринкові переваги (ціна продукту)	3	4	3
4. Ринкові переваги (технічні властивості)	3	3	2
5. Ринкові переваги (експлуатаційні витрати)	3	3	3
6. Ринкові перспективи (розмір ринку)	4	3	3
7. Ринкові перспективи (конкуренція)	3	2	2
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	4	5	5
11. Практична здійсненність (термін реалізації)	5	4	5
12. Практична здійсненність (розробка документів)	5	5	4
Сума балів	45	46	42
Середньоарифметична сума балів $СБ_c$	44,3		

За результатами розрахунків, наведених в таблиці 5.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 5.3 [21].

Таблиця 5.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів $СБ$ розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Нейромережева інформаційна технологія медичного діагностування» становить 44,3 бала, що, відповідно до таблиці 5.3, свідчить

про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

## 5.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення розраховуємо за формулою [22]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (5.1)$$

де  $k$  – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

$\alpha_i$  – коефіцієнт, який враховує питому вагу  $i$ -го технічного показника в загальній якості розробки. Коефіцієнт  $\alpha_i$  визначається експертним шляхом і

при цьому має виконуватись умова  $\sum_{i=1}^k \alpha_i = 1$  ;

$\beta_i$  – відносне значення  $i$ -го технічного показника якості нової розробки.

Відносні значення  $\beta_i$  для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (5.2)$$

де  $I_{ni}$  та  $I_{na}$  – чисельні значення конкретного  $i$ -го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ni}} ; \quad (5.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 5.4.

Таблиця 5.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Достовірність діагностування пацієнтів	%	84	91	1,08	0,35
Швидкодія діагностування	мкс	20	5	4	0,3
Потреба в оперативній пам'яті для функціонування	МВ	6	4	1,5	0,1
Місце на диску	МВ	700	500	1,4	0,1
Доступність інтерфейсу	бал	6	8	1,33	0,15

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 1,08 \cdot 0,35 + 4 \cdot 0,3 + 1,5 \cdot 0,1 + 1,4 \cdot 0,1 + 1,33 \cdot 0,15 = 2,07.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,07 рази.

### 5.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Нейромережева інформаційна технологія медичного діагностування», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

#### 5.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

#### Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою [21]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.4)$$

де  $k$  – кількість посад дослідників залучених до процесу досліджень;

$M_{ni}$  – місячний посадовий оклад конкретного дослідника, грн;

$t_i$  – число днів роботи конкретного дослідника, дн.;

$T_p$  – середнє число робочих днів в місяці,  $T_p=22$  дні.

$$Z_o = 18300,00 \cdot 50 / 22 = 41590,91 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту з розробки і дослідження нейромережевої інформаційної технології	18300,00	831,82	50	41590,91
Консультант (лікар-терапевт вищої категорії)	12000,00	545,45	10	5454,55
Інженер-дослідник (програміст 1-ї категорії)	16500,00	750,00	48	36000,00
Технік-лаборант	7150,00	325,00	32	10400,00
Всього				93445,45

#### Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ( $Z_p$ ) за відповідними найменуваннями робіт НДР на тему «Нейромережева інформаційна технологія медичного діагностування» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (5.5)$$

де  $C_i$  – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_i$  – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду  $C_i$  можна визначити за формулою:



$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (5.6)$$

де  $M_M$  – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo  $M_M=6700,00$  грн;

$K_i$  – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [21];

$K_c$  – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$T_p$  – середнє число робочих днів в місяці, приблизно  $T_p = 22$  дн;

$t_{зм}$  – тривалість зміни, год.

$$C_l = 6700,00 \cdot 1,10 \cdot 1,35 / (22 \cdot 8) = 56,53 \text{ грн.}$$

$$З_{pl} = 56,53 \cdot 3,50 = 197,86 \text{ грн.}$$

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{доод} = (З_o + З_p) \cdot \frac{H_{доод}}{100\%}, \quad (5.7)$$

де  $H_{доод}$  – норма нарахування додаткової заробітної плати. Приймемо 11%.

$$З_{доод} = (93445,45 + 4406,35) \cdot 11 / 100\% = 10763,70 \text{ грн.}$$

Таблиця 5.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Встановлення офісного обладнання робочого місця розробника програмного забезпечення	3,50	2	1,10	56,53	197,86
Інсталяція програмного забезпечення медичного діагностування	7,35	4	1,50	77,09	566,60
Формування дослідної бази даних початкового діагностування	14,30	3	1,35	69,38	992,12
Налагодження програмних блоків формування нейромережі	3,50	5	1,70	87,37	305,78
Монтаж серверного обладнання	1,50	4	1,50	77,09	115,63
Встановлення інтерфейсних блоків зв'язку засобів діагностування	1,60	3	1,35	69,38	111,01
Тренування нейромережі	24,00	3	1,35	69,38	1665,10
Тестування	8,00	2	1,10	56,53	452,25
Всього					4406,35

### 5.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{дод}) \cdot \frac{H_{zn}}{100\%} \quad (5.8)$$

де  $H_{zn}$  – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (93445,45 + 4406,35 + 10763,70) \cdot 22 / 100\% = 23895,41 \text{ грн.}$$

### 5.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Нейромережева інформаційна технологія медичного діагностування».

Витрати на матеріали ( $M$ ), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej}, \quad (5.9)$$

де  $H_j$  – норма витрат матеріалу  $j$ -го найменування, кг;

$n$  – кількість видів матеріалів;

$C_j$  – вартість матеріалу  $j$ -го найменування, грн/кг;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$B_j$  – маса відходів  $j$ -го найменування, кг;

$C_{ej}$  – вартість відходів  $j$ -го найменування, грн/кг.

$$M_1 = 3,0 \cdot 172,00 \cdot 1,05 - 0 \cdot 0 = 541,80 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.7.

### 5.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі ( $K_e$ ), які використовують при проведенні НДР на тему «Нейромережева інформаційна технологія медичного діагностування» відсутні.

### 5.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування

необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Таблиця 5.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
ПАПІР MAESTRO A4 STANDARD+ (PAPER_MS80/MS.A4.80.ST)	172,00	3,0	0	0	541,80
Папір DoubleA A5, 80 г, 500 арк. Premium, клас А (149366)	130,00	4,0	0	0	546,00
Настільний набір Buromax 16 items, black (BM.6302-01)	240,00	3,0	0	0	756,00
Органайзер настільний металевий, 22x14x13 см, чорний H-Tone (JJ41220)	260,00	4,0	0	0	1092,00
Картридж Canon 725 Black (3484B002)	902,00	2,0	0	0	1894,20
Диск DVD Verbatim 4.7Gb 16X SlimBox 1шт MatteSilv AZO (43547-1disk)	42,00	2,0	0	0	88,20
USB накопичувач (флешка) MICRODRIVE 128GB Silver (L)	210,00	1,0	0	0	220,50
Всього					5138,70

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.}i} \cdot K_i, \quad (5.10)$$

де  $C_i$  – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.}i}$  – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань устаткування.

$$B_{\text{снец}} = 2700,00 \cdot 1 \cdot 1,1 = 2970,00 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 5.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ПРИСТРІЙ БЕЗПЕРЕБІЙНОГО ЖИВЛЕННЯ POWERWALKER VI 600 STL	1	2700,00	2970,00
МАРШРУТИЗАТОР XIAOMI MI WIFI ROUTER 4C GLOBAL (DVB4231GL)	1	820,00	902,00
Всього			3872,00

### 5.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прог}} = \sum_{i=1}^k C_{\text{прог}} \cdot C_{\text{прог},i} \cdot K_i, \quad (5.11)$$

де  $C_{\text{прог}}$  – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прог},i}$  – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань програмних засобів.

$$B_{прз} = 1250,00 \cdot 1 \cdot 1,05 = 1312,50 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 5.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Програмне забезпечення для розробки Python	1	1250,00	1312,50
Навчальна база нейромережі даних медичних карток	1	8320,00	8736,00
Всього			10048,50

### 5.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_г} \cdot \frac{t_{вик}}{12}, \quad (5.12)$$

де  $Ц_б$  – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$  – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_г$  – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (31899,00 \cdot 2) / (3 \cdot 12) = 1772,17 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.10.

Таблиця 5.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер розробника ПЗ (Компютер ARTLINE X39 v67 (X39v67) Intel Core i5-11400F / RAM 16ГБ / SSD 1ТБ / nVidia GeForce RTX 3060 12ГБ)	31899,00	3	2	1772,17
Робоче місце інженера-розробника інформаційної технології медичного діагностування	8120,00	7	2	193,33
Робоче місце керівника проекту	7100,00	7	2	169,05
Серверне обладнання підтримки нейромережі	35499,00	3	2	1972,17
Засоби передачі даних	7460,00	4	2	310,83
Офісна оргтехніка	10100,00	5	2	336,67
ОС Windows 11	6210,00	3	2	345,00
Прикладний пакет Microsoft Office 2021 Professional Plus	5890,00	3	2	327,22
Приміщення лабораторії розробки	356000,00	25	2	2373,33
Всього				7799,77

### 5.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ( $B_e$ ) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (5.13)$$

де  $W_{yi}$  – встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  – тривалість роботи обладнання на етапі дослідження, год;

$C_e$  – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo  $C_e = 7,50$  грн;

$K_{eni}$  – коефіцієнт, що враховує використання потужності,  $K_{eni} < 1$ ;

$\eta_i$  – коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$$B_e = 0,32 \cdot 390,0 \cdot 7,50 \cdot 0,95 / 0,97 = 936,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.11.

### 5.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Нейромережева інформаційна технологія медичного діагностування» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:



$$B_{cb} = (Z_o + Z_p) \cdot \frac{H_{cb}}{100\%}, \quad (4.14)$$

де  $H_{cb}$  – норма нарахування за статтею «Службові відрядження», приймемо  $H_{cb} = 20\%$ .

$$B_{cb} = (93445,45 + 4406,35) \cdot 20 / 100\% = 19570,36 \text{ грн.}$$

Таблиця 5.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника ПЗ (Компютер ARTLINE X39 v67 (X39v67) Intel Core i5-11400F / RAM 16ГБ / SSD 1ТБ / nVidia GeForce RTX 3060 12ГБ)	0,32	390,0	936,00
Робоче місце інженера-розробника інформаційної технології медичного діагностування	0,10	390,0	292,50
Робоче місце керівника проекту	0,05	390,0	146,25
Серверне обладнання підтримки нейромережі	0,40	300,0	900,00
Засоби передачі даних	0,04	300,0	90,00
Офісна оргтехніка	0,55	4,8	19,80
ПРИСТРІЙ БЕЗПЕРЕБІЙНОГО ЖИВЛЕННЯ POWERWALKER VI 600 STL	0,08	150,0	90,00
МАРШРУТИЗАТОР XIAOMI MI WIFI ROUTER 4C GLOBAL (DVB4231GL)	0,02	390,00	58,50
Всього			2533,05

5.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (5.15)$$

де  $H_{cn}$  – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo  $H_{cn} = 30\%$ .

$$B_{cn} = (93445,45 + 4406,35) \cdot 30 / 100\% = 29355,54 \text{ грн.}$$

### 5.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_s = (Z_o + Z_p) \cdot \frac{H_{is}}{100\%}, \quad (5.16)$$

де  $H_{is}$  – норма нарахування за статтею «Інші витрати», прийmemo  $H_{is} = 60\%$ .

$$I_s = (93445,45 + 4406,35) \cdot 60 / 100\% = 58711,09 \text{ грн.}$$

### 5.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.17)$$

де  $H_{нзв}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo  $H_{нзв} = 115\%$ .

$$B_{нзв} = (93445,45 + 4406,35) \cdot 115 / 100\% = 112529,58 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Нейромережева інформаційна технологія медичного діагностування» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{од} + Z_n + M + K_e + B_{стц} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сп} + I_e + B_{нзв}. \quad (5.18)$$

$$B_{заг} = 93445,45 + 4406,35 + 10763,70 + 23895,41 + 5138,70 + 0,00 + 3872,00 + 10048,50 + 7799,77 + 2533,05 + 19570,36 + 29355,54 + 58711,09 + 112529,58 = 382069,51 \text{ грн.}$$

Загальні витрати  $ZB$  на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{заг}}{\eta}, \quad (5.19)$$

де  $\eta$  - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo  $\eta=0,9$ .

$$ZB = 382069,51 / 0,9 = 424521,68 \text{ грн.}$$

#### 5.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Нейромережева інформаційна технологія медичного діагностування» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

$\Delta N$  – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	200	500	750	450

$N$  – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 1500 осіб;

$C_o$  – вартість послуги у році до впровадження інформаційної системи, прийmemo 25000,00 грн;

$\pm \Delta C_o$  – зміна вартості послуги від впровадження результатів, прийmemo 687,50 грн.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta \Pi_i$  для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [21]:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\mathcal{G}}{100}\right), \quad (5.20)$$

де  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo  $\rho = 40\%$ ;

$\mathcal{G}$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році  $\mathcal{G} = 18\%$ ;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (687,50 \cdot 1500,00 + 25687,50 \cdot 200) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1679380,50 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (687,50 \cdot 1500,00 + 25687,50 \cdot 700) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 5175963,00 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (687,50 \cdot 1500,00 + 25687,50 \cdot 1450) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 10420836,75$$

грн.

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (687,50 \cdot 1500,00 + 25687,50 \cdot 1900) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 13567761,00$$

грн.

Приведена вартість збільшення всіх чистих прибутків  $\Pi\Pi$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^i}, \quad (5.21)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

$T$  – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,35$ ;

$t$  – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} III &= 1679380,50/(1+0,35)^1 + 5175963,00/(1+0,35)^2 + 10420836,75/(1+0,35)^3 + \\ &+ 13567761,00/(1+0,35)^4 = 1243985,56 + 2840034,57 + 4235466,85 + 4084821,76 = \\ &= 12404308,73 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (5.22)$$

де  $k_{инв}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо  $k_{инв} = 2$ ;

$3B$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 424521,68 грн.

$$PV = k_{инв} \cdot 3B = 2 \cdot 424521,68 = 849043,35 \text{ грн.}$$

Абсолютний економічний ефект  $E_{абс}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = III - PV \quad (5.23)$$

де  $III$  – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 12404308,73 грн;

$PV$  – теперішня вартість початкових інвестицій, 849043,35 грн.

$$E_{abc} = III - PV = 12404308,73 - 849043,35 = 11555265,38 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій  $E_g$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_g = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.24)$$

де  $E_{abc}$  – абсолютний економічний ефект вкладених інвестицій, 11555265,38 грн;

$PV$  – теперішня вартість початкових інвестицій, 849043,35 грн;

$T_{жс}$  – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_g = \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 11555265,38/849043,35)^{1/4} = 0,96.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій  $\tau_{min}$ :

$$\tau_{min} = d + f, \quad (5.25)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні  $d = 0,12$ ;

$f$  – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,4.

$\tau_{min} = 0,12 + 0,4 = 0,52 < 0,96$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_g$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну

роботу за темою «Нейромережева інформаційна технологія медичного діагностування» доцільно.

Період окупності інвестицій  $T_{ок}$  які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (5.26)$$

де  $E_g$  – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,96 = 1,05 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

### 5.5 Висновок до розділу 5

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Нейромережева інформаційна технологія медичного діагностування» становить 44,3 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий). При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,07 рази. Також термін окупності становить 1,05 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок. Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Нейромережева інформаційна технологія медичного діагностування».



## ВИСНОВКИ

У роботі було проаналізовано процес комп'ютеризованого медичного діагностування за аналізом крові, розглянуто параметри аналізу крові та їх зв'язок з відповідними групами хвороб, виявлені проблеми та недоліки, пов'язані із недосконалістю сучасного діагностування. Було визначено актуальність проблеми та обґрунтовано метод її розв'язання – розробка інформаційної технології медичного діагностування на основі нейронної мережі Кохонена, що дозволить провести автоматизацію визначення попереднього діагнозу пацієнта та підвищить його достовірність. Були розглянуті основні програмні засоби, які виконують медичну діагностику, та визначені їх недоліки.

У другому розділі було обґрунтовано вибір нейронної мережі Кохонена та процесу кластеризації для інформаційної технології медичного діагностування. Спроектовано структуру, математичну модель та порядок функціонування нейронної мережі Кохонена. Розроблено структуру процесів обробки інформації інформаційної технології. Розроблено алгоритм роботи нейромережевого медичного діагностування за аналізом крові.

У третьому розділі було обґрунтовано вибір мови програмування C# та середовища програмування Visual Studio. Також була використана бібліотека Accord.NET. Було описано реалізацію основних компонентів програми медичного діагностування на основі нейронної мережі Кохонена, розроблено діаграму класів програми.

У четвертому розділі в результаті тестування програми медичного діагностування пацієнтів на основі нейронної мережі Кохонена було доведено її повну працездатність та відповідність поставленому завданню. Розроблена програма має вищу достовірність медичного діагностування пацієнтів (91%), ніж аналогічна програма (84%), а значить достовірність медичного діагностування пацієнтів покращена на 7%, Тобто мета роботи досягнута – достовірність медичного діагностування підвищена.

У п'ятому розділі було визначено рівень комерційного потенціалу розробки за темою «Нейромережева інформаційна технологія медичного діагностування» становить 44,3 балів, що вказує на комерційну вагомість проведення даних досліджень (рівень комерційного потенціалу розробки середній). При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,07 рази. Також термін окупності становить 1,05 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок. Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Нейромережева інформаційна технологія медичного діагностування».

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. О. О. Пащенко, Ю. М. Паночишин Нейромережева інформаційна технологія медичного діагностування / в Матеріали Міжнародна науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2024)», Вінниця, 2024, [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/viewFile/19773/16373>
2. Автоматизація медичних установ [Електронний ресурс]. Режим доступу: <http://www.aksimed.com/clients/> - Назва з екрану
3. Телемедицина [Електронний ресурс]. Режим доступу: <http://uk.wikipedia.org/wiki/Телемедицина> - Назва з екрану
4. Алгоритми діагностування пацієнтів [Електронний ресурс]. Режим доступу: <http://www.vidal.com/vracham/Informatsiya-dlya-spetsialistov/Nevrologiya-psihiatriya/Algoritmy-diagnostiki-i-vedeniya-patsientov/> - Назва з екрану
5. Аналізи крові [Електронний ресурс]. Режим доступу: <http://medportal.com/enc/analysis/blood/> - Назва з екрану
6. Загальний аналіз крові [Електронний ресурс]. Режим доступу: <http://ukrhealth.net/zagalnij-analiz-krovi/> - Назва з екрану
7. Загальний аналіз крові: розшифровка та норми [Електронний ресурс]. Режим доступу: <http://rheumatology.org.ua/blog/articles/1076/> - Назва з екрану
8. Симахін В.А. Кластеризація медичних даних за допомогою нейромереж / В.А. Симахін, А.В. Кашинцев [Електронний ресурс]. Режим доступу: <http://neurocomp.com/klasterizaciya-medicinskix-dannyx-s-pomoshhyu-nejrosetej/> - Назва з екрану
9. Автоматизація загального аналізу крові [Електронний ресурс]. Режим доступу: <http://www.myshared.com/slide/114542/> - Назва з екрану
10. Автоматизація - хімічний аналіз [Електронний ресурс]. Режим доступу: <http://www.ngpedia.com/id226550p1.html> - Назва з екрану

11. Your Diagnosis. Your Free Online Medical Diagnosis & Symptoms Analysis Tool [Електронний ресурс]. Режим доступу: <http://www.yourdiagnosis.com/> - Назва з екрану.

12. Інноваційний медичний центр [Електронний ресурс]. Режим доступу: <http://www.diagnos.com/> - Назва з екрану.

13. Руденко О.В. Штучні нейронні мережі: Навчальний посібник / О.В.Руденко, Є.В.Бодянський. - Харків : ТОВ «Компанія СМІТ», 2006. — 404 с. - ISBN 966-8630-73-X.

14. Кононова К. Ю. Машинне навчання: методи та моделі: підручник для бакалаврів, магістрів та докторів філософії спеціальності 051 «Економіка» / К. Ю. Кононова. – Харків: ХНУ імені В. Н. Каразіна, 2020. – 301 с.

15. Dostal P. Cluster analysis and neural network / P. Dostal, P.Pokorny [Електронний ресурс]. Режим доступу: [http://dsp.vscht.cz/konference\\_matlab/MATLAB08/prispevky/025\\_dostal.pdf](http://dsp.vscht.cz/konference_matlab/MATLAB08/prispevky/025_dostal.pdf)

16. Нейронні мережі: теорія та практика [Електронний ресурс]. Режим доступу: <https://intalent.pro/article/neyronnye-setiteoriya-i-praktika.html>

17. Дмитрієва Т.В. Самоорганізуючі карти Кохонена / Т.В. Дмитрієва [Електронний ресурс]. Режим доступу: [http://storage.library.opu.ua/online/conference/tez\\_45/1-8.pdf](http://storage.library.opu.ua/online/conference/tez_45/1-8.pdf)

18. Нейронні мережі на C#. Мережа Кохонена [Електронний ресурс]. Режим доступу: <http://www.pvsm.com/net/7339> - Назва з екрану.

19. Нейронні мережі на C#. Мережа Кохонена [Електронний ресурс]. Режим доступу: <http://habrahabr.com/post/143668/> – Назва з екрану.

20. Албахарс Д. C# 5.0. Довідник. Повний опис мови / Джозеф Алхабарі, Бен Алхабарі — М. : «Вільямс», 2013. — 1008 с.

21. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

22. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка – Вінниця : ВНТУ, 2016. – 113 с.

23. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. К. Колесницький. – Вінниця : ВНТУ, 2023. – (58 с.)

## Додаток А (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬНазва роботи: Нейромережева інформаційна технологія медичного діагностуванняТип роботи: магістерська кваліфікаційна робота  
(БДР, МКР)Підрозділ кафедра комп'ютерних наук, ФІТА  
(кафедра, факультет)

## Показники звіту подібності Unicheck

Оригінальність 81,4% Схожість 18,6%

## Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи  Пашченко О.О.Керівник роботи  Паночишин Ю.М.

## Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захистуОсоба, відповідальна за перевірку  Озеранський В.С.

## Додаток Б (обов'язковий)

### Лістинг програми

#### Фрагмент лістингу

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace SOFMTTest
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new frmMain());
        }
    }
}
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <sectionGroup name="applicationSettings"
type="System.Configuration.ApplicationSettingsGroup, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" >
      <section name="SOFMTTest.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
    </sectionGroup>
    <sectionGroup name="userSettings"
type="System.Configuration.UserSettingsGroup, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" >
      <section name="SOFMTTest.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
allowExeDefinition="MachineToLocalUser" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <applicationSettings>
    <SOFMTTest.Properties.Settings>
      <setting name="MaximalNumberOfNeurons" serializeAs="String">
        <value>1000</value>
      </setting>
      <setting name="MaximalNumberOfIterations" serializeAs="String">
        <value>10000</value>
      </setting>
    </SOFMTTest.Properties.Settings>
  </applicationSettings>
  <userSettings>
    <SOFMTTest.Properties.Settings>
      <setting name="MinimalEpsilon" serializeAs="String">
        <value>1E-09</value>
      </setting>
      <setting name="MaximalEpsilon" serializeAs="String">
        <value>0.1</value>
      </setting>
    </SOFMTTest.Properties.Settings>
  </userSettings>
</configuration>
using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using SOFM;

        Application.DoEvents();
    }
    private void nn_EndIterationEvent(object sender, EventArgs e)
    {
        if (pbStatus.Value < pbStatus.Maximum) pbStatus.Value++;
    }

    private void lbPatterns_SelectedIndexChanged(object sender, EventArgs e)
    {
        inputDataChart.Series[1].Clear();
        outputDataChart.Series[1].Clear();
        inputDataChart.Series[1].Add(nn.Patterns[lbPatterns.SelectedIndex][0],
nn.Patterns[lbPatterns.SelectedIndex][1]);
        Neuron Winner = nn.FindWinner(nn.Patterns[lbPatterns.SelectedIndex]);
        outputDataChart.Series[1].Add(Winner.Weights[0], Winner.Weights[1]);
        sofmVisualizer.LightUpThePixel(Winner.Coordinate.X, Winner.Coordinate.Y);
    }

    private void lblIONTLink_Click(object sender, EventArgs e)
    {
        System.Diagnostics.Process.Start("http://www.iont.ru");
    }

    private void chbVisualization_CheckedChanged(object sender, EventArgs e)
    {
        this.nn_EndEpochEvent(this, new EndEpochEventArgs());
    }

    private void AddPatternsToListBox()
    {
        lbPatterns.Items.Clear();
        string patternString;
        for (int i = 0; i < nn.Patterns.Count; i++)
        {
            patternString = "";
            patternString += nn.Classes[i]+" ";
            for (int j = 0; j < nn.InputLayerDimension; j++)
                patternString += nn.Patterns[i][j].ToString("g2") + " ";
            lbPatterns.Items.Add(patternString);
        }
    }

    private void SetError(TextBox tb, string s)
    {
        errorProvider.SetError(tb, s);

        bool errorOnOtherTextBoxControl = false;
        foreach (Control c in this.gbInputParams.Controls)
        {
            if (c is TextBox)
            {
                if (errorProvider.GetError(c).Length!=0)
                    errorOnOtherTextBoxControl = true;
            }
        }
        if (!errorOnOtherTextBoxControl)
            btnLoadDataAndCreateNetwork.Enabled = true;
        else
            btnLoadDataAndCreateNetwork.Enabled = false;
    }

    private void tb_Validating(object sender, CancelEventArgs e)
    {

```



```

TextBox tb = (TextBox)sender;
if (tb.Text.Length != 0)
{
    bool notValidSymbol = false;
    char[] cArray = tb.Text.ToCharArray();
    foreach (char c in cArray)
    {
        if (tb.Name == "tbEpsilon")
        {
            if (!Char.IsDigit(c) && c != '.') notValidSymbol = true;
        }
        else
        {
            if (!Char.IsDigit(c)) notValidSymbol = true;
        }
    }
    if (!notValidSymbol)
    {
        switch (tb.Name)
        {
            case "tbNumberOfCards":
            {
                if (Convert.ToInt32(tb.Text) <=
Properties.Settings.Default.MaximalNumberOfNeurons) SetError(tb, "");
                else SetError(tb, "Maximal number of neurons is - " +
Properties.Settings.Default.MaximalNumberOfNeurons.ToString());
                break;
            }
            case "tbIterationsNumber":
            {
                if (Convert.ToInt32(tb.Text) <=
Properties.Settings.Default.MaximalNumberOfIterations) SetError(tb, "");
                else SetError(tb, "Maximal number of iterations - " +
Properties.Settings.Default.MaximalNumberOfIterations.ToString());
                break;
            }
            case "tbEpsilon":
            {
                double epsilon = Convert.ToDouble(tb.Text);
                if (epsilon <=
Properties.Settings.Default.MaximalEpsilon && epsilon >=
Properties.Settings.Default.MinimalEpsilon) SetError(tb, "");
                else SetError(tb, "Epsilon must lie between " +
Properties.Settings.Default.MinimalEpsilon.ToString() + "and " +
Properties.Settings.Default.MaximalEpsilon.ToString());
                break;
            }
        }
    }
    else
    {
        SetError(tb, "Incorrect input. Please try again.");
    }
}
}

namespace SOFM
{
    public enum Functions
    {
        Discrete,
        Gaus,
        MexicanHat,
        FrenchHat
    }
}
using System;
using System.Collections.Generic;

```

```

using System.Text;
using System.IO;

namespace SOFM
{
    public delegate void EndEpochEventHandler(object sender, EndEpochEventArgs e);
    public delegate void EndIterationEventHandler(object sender, EventArgs e);

    public class NeuralNetwork
    {
        private Neuron[,] outputLayer;

        public Neuron[,] OutputLayer
        {
            get { return outputLayer; }
            set { outputLayer = value; }
        }
        private int inputLayerDimension;
        private int outputLayerDimension;
        private int numberOfPatterns;
        private List<List<double>> patterns;
        private List<string> classes;
        private SortedList<string, int> existentClasses;
        private List<System.Drawing.Color> usedColors;
        private bool normalize;
        private int numberOfIterations;
        private int currentIteration;

        private Functions function;
        private double epsilon;
        private double currentEpsilon;

        private double CalculateNormOfVectors(List<double> vector1, List<double>
vector2)
        {
            double value = 0;
            for(int i=0; i<vector1.Count; i++)
                value += Math.Pow((vector1[i] - vector2[i]), 2);
            value = Math.Sqrt(value);
            return value;
        }

        private void NormalizeInputPattern(List<double> pattern)
        {
            double nn = 0;
            for (int i = 0; i < inputLayerDimension; i++)
            {
                nn += (pattern[i] * pattern[i]);
            }
            nn = Math.Sqrt(nn);
            for (int i = 0; i < inputLayerDimension; i++)
            {
                pattern[i] /= nn;
            }
        }

        private void StartEpoch(List<double> pattern)
        {
            Neuron Winner = this.FindWinner(pattern);
            currentEpsilon = 0;
            for (int i = 0; i < outputLayerDimension; i++)
                for (int j = 0; j < outputLayerDimension; j++)
                {
                    currentEpsilon += outputLayer[i, j].ModifyWights(pattern,
Winner.Coordinate, currentIteration, function);
                }
            currentIteration++;
            currentEpsilon = Math.Abs(currentEpsilon / (outputLayerDimension *
outputLayerDimension));
            EndEpochEventArgs e = new EndEpochEventArgs();

```

```

        OnEndEpochEvent(e);
    }

    public bool Normalize
    {
        get { return normalize; }
        set { normalize = value; }
    }

    public List<List<double>> Patterns
    {
        get { return patterns; }
    }

    public List<string> Classes
    {
        get { return classes; }
    }

    public int InputLayerDimension
    {
        get { return inputLayerDimension; }
    }

    public int OutputLayerDimension
    {
        get { return outputLayerDimension; }
    }

    public double CurrentDelta
    {
        get { return currentEpsilon; }
    }

    public SortedList<string, int> ExistentClasses
    {
        get { return existentClasses; }
    }

    public List<System.Drawing.Color> UsedColors
    {
        get { return usedColors; }
    }

    private int NumberOfClasses()
    {
        existentClasses = new SortedList<string, int>();
        existentClasses.Add(classes[0], 1);
        int k = 0;
        int d = 2;
        for (int i = 1; i < classes.Count; i++)
        {
            k=0;
            for (int j = 0; j < existentClasses.Count; j++)
            {
                if (existentClasses.IndexOfKey(classes[i])!=-1) k++;
                if (k == 0)
                {
                    existentClasses.Add(classes[i],d);
                    d++;
                }
            }
        }
        return existentClasses.Count;
    }

    public System.Drawing.Color[,] ColorSOFM()
    {
        System.Drawing.Color[,] colorMatrix = new
System.Drawing.Color[outputLayerDimension, outputLayerDimension];
        int numOfClasses = NumberOfClasses();
        List<System.Drawing.Color> goodColors = new List<System.Drawing.Color>();
    }

```

```

goodColors.Add(System.Drawing.Color.Black);
goodColors.Add(System.Drawing.Color.Red);
goodColors.Add(System.Drawing.Color.Navy);
goodColors.Add(System.Drawing.Color.Green);
goodColors.Add(System.Drawing.Color.Yellow);
usedColors = new List<System.Drawing.Color>(numOfClasses);
usedColors.Add(goodColors[0]);
int k = 0;
int randomColor = 0;
Random r = new Random();
while (usedColors.Count != numOfClasses)
{
    k = 0;
    randomColor = r.Next(goodColors.Count);
    foreach (System.Drawing.Color cl in usedColors)
        if (cl == goodColors[randomColor]) k++;
    if (k == 0) usedColors.Add(goodColors[randomColor]);
}
for (int i = 0; i < outputLayerDimension; i++)
    for (int j = 0; j < outputLayerDimension; j++)
        colorMatrix[i, j] =
System.Drawing.Color.FromKnownColor(System.Drawing.KnownColor.ButtonFace);

    for (int i = 0; i < patterns.Count; i++)
    {
        Neuron n = FindWinner(patterns[i]);
        colorMatrix[n.Coordinate.X,n.Coordinate.Y] =
usedColors[existentClasses[classes[i]]-1];
    }
    return colorMatrix;
}

public NeuralNetwork(int m, int numberOfIterations, double epsilon, Functions
f)
{
    outputLayerDimension = m;
    currentIteration = 1;
    this.numberOfIterations = numberOfIterations;
    function = f;
    this.epsilon = epsilon;
    currentEpsilon = 100;
}

public Neuron FindWinner(List<double> pattern)
{
    List<double> norms = new List<double>(outputLayerDimension *
outputLayerDimension);
    double D = 0;
    Neuron Winner = outputLayer[0, 0];
    double min = CalculateNormOfVectors(pattern, outputLayer[0, 0].Weights);
    for (int i = 0; i < outputLayerDimension; i++)
        for (int j = 0; j < outputLayerDimension; j++)
        {
            D = CalculateNormOfVectors(pattern, outputLayer[i, j].Weights);
            if (D < min)
            {
                min = D;
                Winner = outputLayer[i, j];
            }
        }
    return Winner;
}

public void StartLearning()
{
    int iterations = 0;
    while (iterations<=numberOfIterations && currentEpsilon > epsilon)
    {
        List<List<double>> patternsToLearn = new
List<List<double>>(numberOfPatterns);

```

```

        foreach (List<double> pArray in patterns)
            patternsToLearn.Add(pArray);
        Random randomPattern = new Random();
        List<double> pattern = new List<double>(inputLayerDimension);
        for (int i = 0; i < numberOfPatterns; i++)
        {
            pattern = patternsToLearn[randomPattern.Next(numberOfPatterns -
i)];

            StartEpoch(pattern);

            patternsToLearn.Remove(pattern);
        }
        iterations++;
        OnEndIterationEvent(new EventArgs());
    }
}

public void ReadDataFromFile(string inputDataFileName)
{
    StreamReader sr = new StreamReader(inputDataFileName);
    string line = sr.ReadLine();
    int k = 0;
    for (int i = 0; i < line.Length; i++)
    {
        if (line[i] == ' ') k++;
    }

    inputLayerDimension = k;
    int sigma0 = outputLayerDimension;

    outputLayer = new Neuron[outputLayerDimension, outputLayerDimension];
    Random r = new Random();
    for (int i = 0; i < outputLayerDimension; i++)
        for (int j = 0; j < outputLayerDimension; j++)
        {
            outputLayer[i, j] = new Neuron(i, j, sigma0);
            outputLayer[i, j].Weights = new List<double>(inputLayerDimension);
            for (k = 0; k < inputLayerDimension; k++)
            {
                outputLayer[i, j].Weights.Add(r.NextDouble());
            }
        }

    k = 0;
    while (line != null)
    {
        line = sr.ReadLine();
        k++;
    }
    patterns = new List<List<double>>(k);
    classes = new List<string>(k);
    numberOfPatterns = k;

    List<double> pattern;

    sr = new StreamReader(inputDataFileName);
    line = sr.ReadLine();

    while (line != null)
    {
        int startPos = 0;
        int endPos = 0;
        int j = 0;
        pattern = new List<double>(inputLayerDimension);
        for (int ind = 0; ind < line.Length; ind++)
        {
            if (line[ind] == ' ' && j != inputLayerDimension)
            {
                endPos = ind;
            }
        }
    }
}

```

```

        pattern.Add(Convert.ToDouble(line.Substring(startPos, endPos -
startPos)));
        startPos = ind + 1;
        j++;
    }
    if (j > inputLayerDimension) throw new InvalidDataException("Wrong
file format. Check input data file, and try again");
    }
    if (normalize) this.NormalizeInputPattern(pattern);
    patterns.Add(pattern);
    startPos = line.LastIndexOf(' ');
    classes.Add(line.Substring(startPos));
    line = sr.ReadLine();
    }
}

public event EndEpochEventHandler EndEpochEvent;
public event EndIterationEventHandler EndIterationEvent;

protected virtual void OnEndEpochEvent(EndEpochEventArgs e)
{
    if (EndEpochEvent != null)
        EndEpochEvent(this, e);
}

protected virtual void OnEndIterationEvent(EventArgs e)
{
    if (EndIterationEvent != null)
        EndIterationEvent(this, e);
}
}
} using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;

namespace SOFM
{
    public class Neuron
    {
        private List<double> weights;
        private Point coordinate;
        private int iteration;
        private int wightsdimension;
        private int sigma0;
        private double alpha0 = 0.1;
        private double tau1;
        private int tau2 = 1000;

        private double h(Point winnerCoordinate, Functions f)
        {
            double result = 0;
            double distance = 0;
            switch (f)
            {
                case Functions.Discrete:
                    {
                        distance = Math.Abs(this.Coordinate.X - winnerCoordinate.X) +
Math.Abs(this.Coordinate.Y - winnerCoordinate.Y);
                        switch ((int)distance)
                        {
                            case 0:
                                result = 1;
                                break;
                            case 1:
                                result = 0.5f;
                                break;
                            case 2:
                                result = 0.25f;
                                break;
                        }
                    }
                }
            }
        }
    }
}

```

```

        case 3:
            result = 0.125f;
            break;
        }
        break;
    }
    case Functions.Gaus:
    {
        distance = Math.Sqrt(Math.Pow((winnerCoordinate.X -
coordinate.X), 2) + Math.Pow((winnerCoordinate.Y - coordinate.Y), 2));
        result = Math.Exp(-(distance * distance) /
(Math.Pow(Sigma(iteration), 2)));
        break;
    }
    case Functions.MexicanHat:
    {
        distance = Math.Sqrt(Math.Pow((winnerCoordinate.X -
coordinate.X), 2) + Math.Pow((winnerCoordinate.Y - coordinate.Y), 2));
        result = Math.Exp(-(distance * distance) /
Math.Pow(Sigma(iteration), 2)) * (1 - (2 / Math.Pow(Sigma(iteration), 2)) * (distance
* distance));
        break;
    }
    case Functions.FrenchHat:
    {
        int a = 2;
        distance = Math.Abs(this.Coordinate.X - winnerCoordinate.X) +
Math.Abs(this.Coordinate.Y - winnerCoordinate.Y);
        if (distance <= a) result = 1;
        else
            if (distance < a && distance <= 3 * a) result = -1 / 3;
            else
                if (distance > 3 * a) result = 0;
        break;
    }
    }
    return result;
}

private void InitializeVariables(int sigma0)
{
    iteration = 1;
    this.sigma0 = sigma0;
    tau1 = 1000 / Math.Log(sigma0);
}

private double Alpha(int t)
{
    double value = alpha0 * Math.Exp(-t/tau2);
    return value;
}

private double Sigma(int t)
{
    double value = sigma0 * Math.Exp(-t/tau1);
    return value;
}

public List<double> Weights
{
    get { return weights; }
    set
    {
        weights = value;
        wightsdimension = weights.Capacity;
    }
}

public Point Coordinate
{

```

```

    get { return coordinate; }
    set { coordinate = value; }
}

public int Iteration
{
    get { return iteration; }
}

public Neuron(int x, int y, int sigma0)
{
    coordinate.X = x;
    coordinate.Y = y;
    InitializeVariables(sigma0);
}

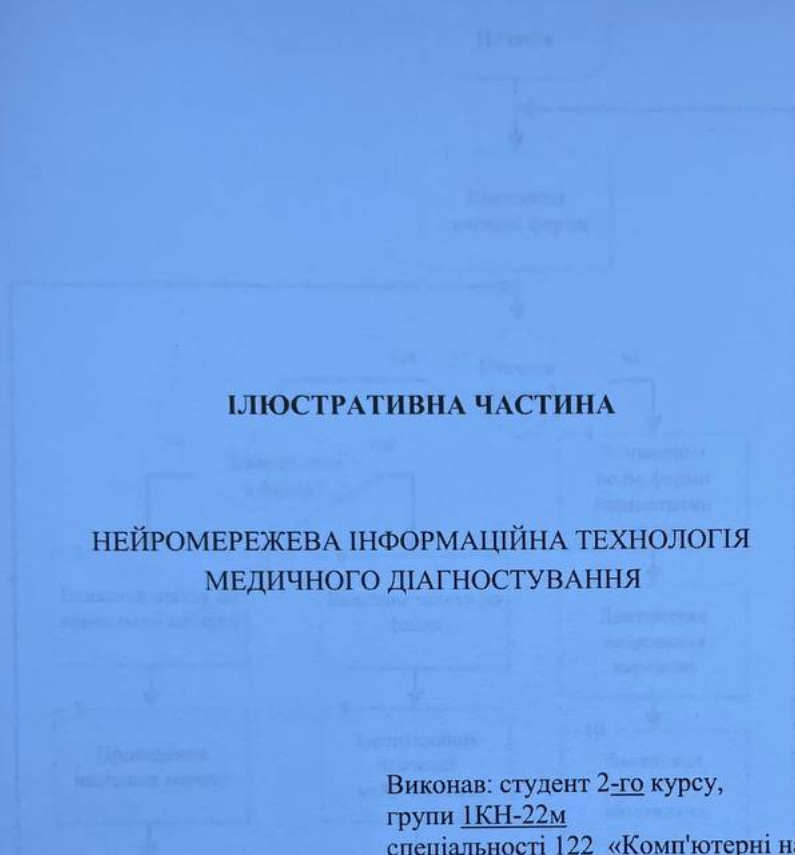
public Neuron(Point coordinate, int sigma0)
{
    this.coordinate = coordinate;
    InitializeVariables(sigma0);
}

public double ModifyWights(List<double> pattern, Point winnerCoordinate, int
iteration, Functions f)
{
    double avgDelta = 0;
    double modificationValue =0;
    for (int i = 0; i < wightsdimension; i++)
    {
        modificationValue = Alpha(iteration) * h(winnerCoordinate, f) *
(pattern[i] - weights[i]);
        weights[i] += modificationValue;
        avgDelta += modificationValue;
    }
    avgDelta = avgDelta / wightsdimension;
    return avgDelta;
}

public double Norm
{
    get
    {
        double norm = 0;
        foreach (double d in weights)
        {
            norm += d;
        }
        norm = norm / this.wightsdimension;
        return norm}}}}

```





**ІЛЮСТРАТИВНА ЧАСТИНА**

**НЕЙРОМЕРЕЖЕВА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ  
МЕДИЧНОГО ДІАГНОСТУВАННЯ**

Виконав: студент 2-го курсу,  
групи ІКН-22м  
спеціальності 122 «Комп'ютерні науки»  
(шифр і назва напрямку підготовки, спеціальності)  
Пашенко О.О.  
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН  
Паночишин Ю.М.  
(прізвище та ініціали)  
« 07 » 12 2023 р.

Рис. 1.1 – Структурна діаграма нейронної мережі для медичного діагностування

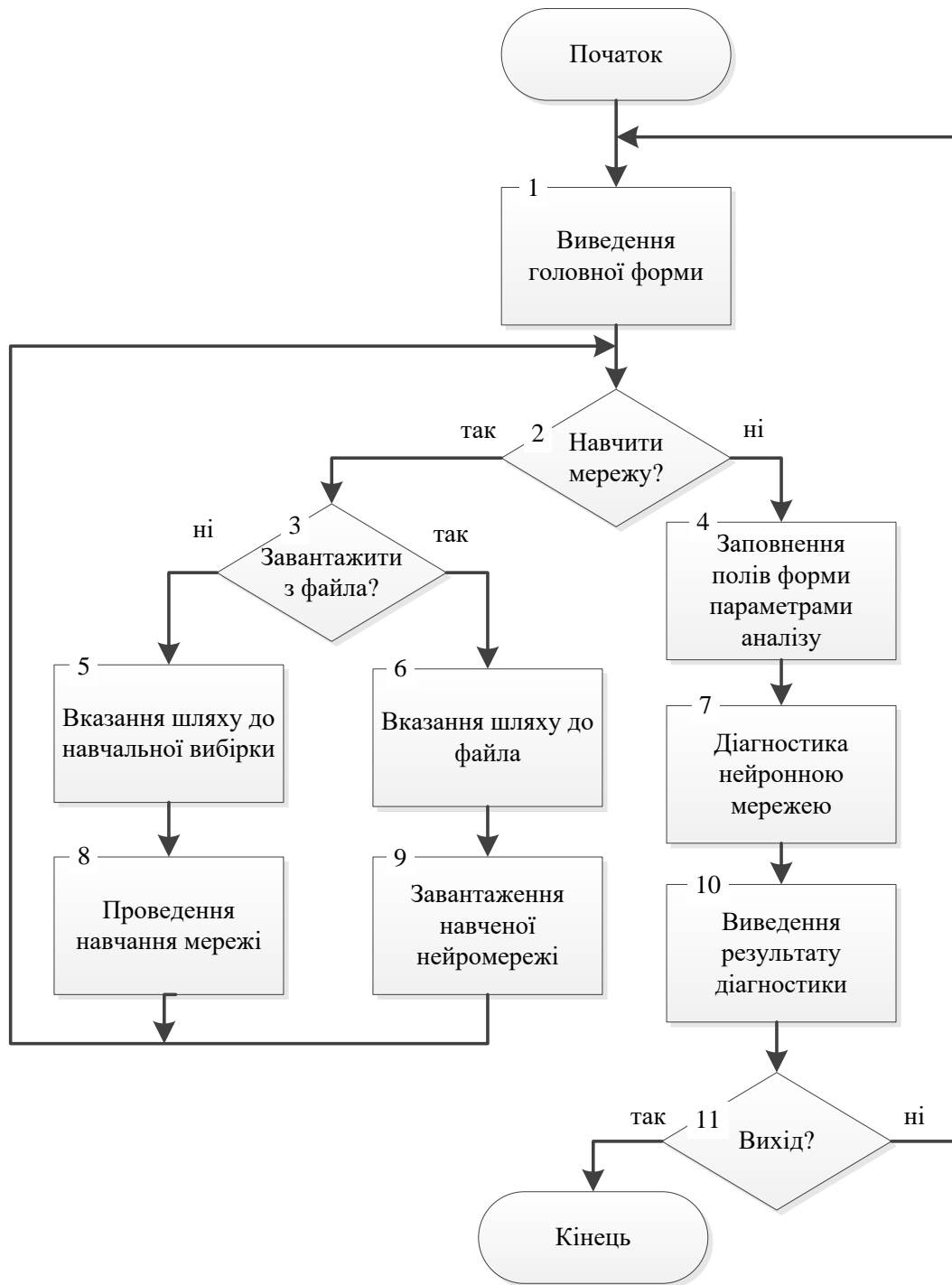


Рисунок В.1 – Граф-схема загального алгоритму роботи програми медичного діагностування

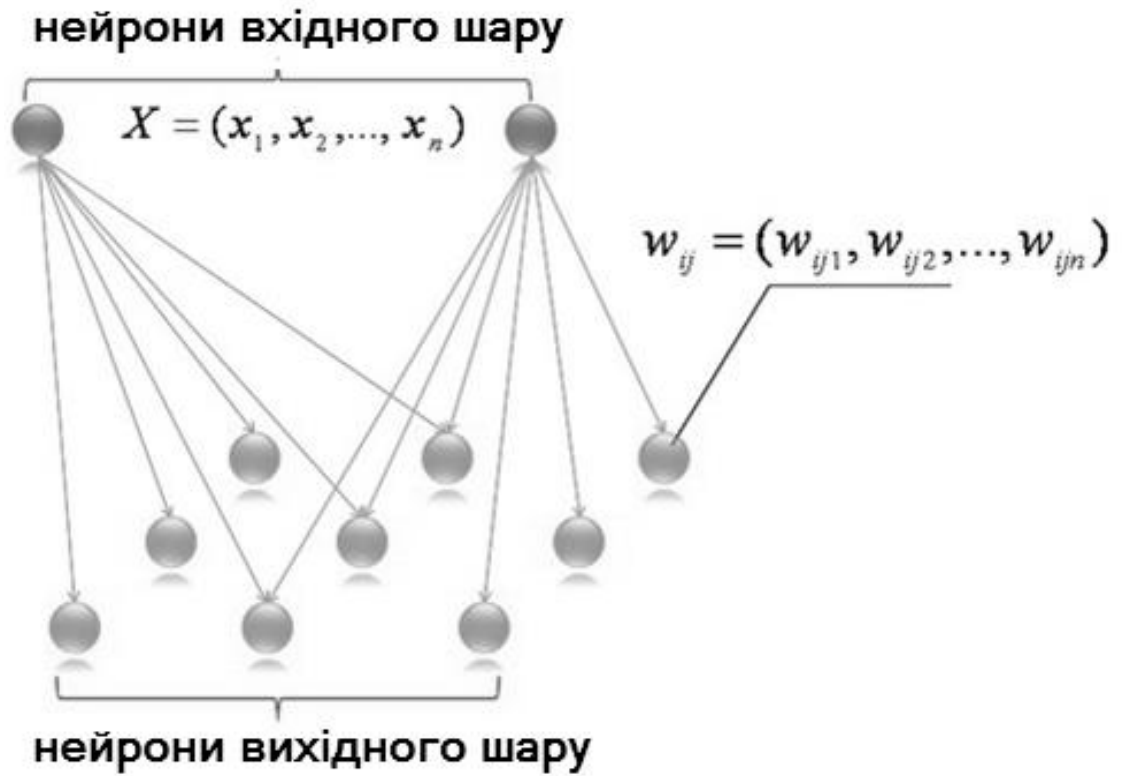


Рисунок В.2 – Структура нейромережі Кохонена розроблюваної програми

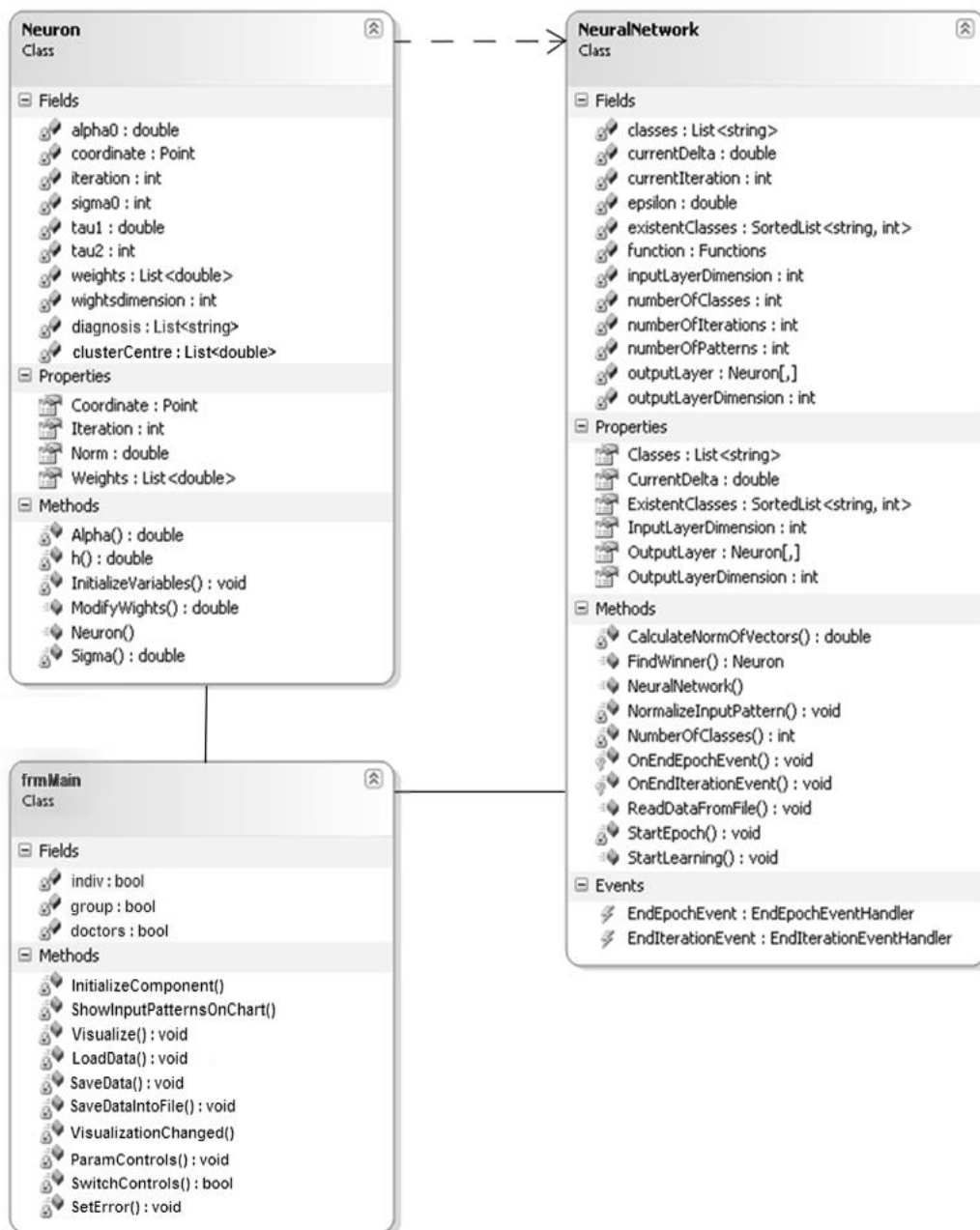


Рисунок В.3 – Діаграма класів програми медичного діагностування

Попереднє діагностування захворювань за аналізом крові

Діагностування **Навчання**

Стать:  Ч  Ж      Вік:  До 15  15-20  50+

Введіть результати загального аналізу крові:

Лейкоцити WBC	<input type="text"/>	Тромбоцити PLT	<input type="text"/>
Еритроцити RBC	<input type="text"/>	Швидкість осідання еритроцитів - ШОЕ (ESR), Формула крові	<input type="text"/>
Гемоглобін HGB	<input type="text"/>	Нейтрофіли NEU	<input type="text"/>
Гематокрит HCT	<input type="text"/>	Лімфоцити LYMPH	<input type="text"/>
Середній об'єм еритроцитів MCV	<input type="text"/>	Моноцити MONO	<input type="text"/>
Середній вміст гемоглобіну в еритроциті MCH	<input type="text"/>	Еозінофіли EO	<input type="text"/>
Середня концентрація гемоглобіну в еритроциті MCHC	<input type="text"/>	Базофіли BASO	<input type="text"/>
Індекс розподілу еритроцитів RDW-CV	<input type="text"/>	<input type="button" value="Продіагностувати"/>	

Попереднє діагностування захворювань за аналізом крові

Діагностування **Навчання**

Завантажити  
навчену  
нейромережу

Провести  
навчання  
нейромережі на  
нових даних

Рисунок В.4 – Стартові вікна програми медичного діагностування

Попереднє діагностування захворювань за аналізом крові

Діагностування **Навчання**

Стать:  Ч  Ж      Вік:  До 15  15-20  50+

Введіть результати загального аналізу крові:

Лейкоцити WBC	<input type="text" value="5"/>	Тромбоцити PLT	<input type="text" value="210"/>
Еритроцити RBC	<input type="text" value="4.2"/>	Швидкість осідання еритроцитів - ШОЕ (ESR), Формула крові	<input type="text" value="9"/>
Гемоглобін HGB	<input type="text" value="132"/>	Нейтрофіли NEU	<input type="text" value="4"/>
Гематокрит HCT	<input type="text" value="0.44"/>	Лімфоцити LYMPH	<input type="text" value="2"/>
Середній об'єм еритроцитів MCV	<input type="text" value="87"/>	Моноцити MONO	<input type="text" value="0.3"/>
Середній вміст гемоглобіну в еритроциті MCH	<input type="text" value="29"/>	Еозінофіли EO	<input type="text" value="0.4"/>
Середня концентрація гемоглобіну в еритроциті MCHC	<input type="text" value="36"/>	Базофіли BASO	<input type="text" value="0.06"/>
Індекс розподілу еритроцитів RDW-CV	<input type="text" value="7.2"/>	<input type="button" value="Продіагностувати"/>	

Попереднє діагностування захворювань за аналізом крові

**Результати попереднього діагностування**

- Захворювання легенів
- Серцево-судинні захворювання
- Захворювання ЦНС
- Бактеріальна інфекція
- Вірусна інфекція
- Захворювання нирок
- Аутоімунні захворювання
- Ендокринологічні захворювання
- Злоякісні пухлини
- Анемія
- Діабет
- Захворювань не виявлено

**Повернутися до вікна введення параметрів аналізу крові**

Рисунок В.5 – Приклад результату роботи програми медичного діагностування

Програмний засіб	Кількість пацієнтів у тестовій вибірці	Кількість правильно діагностованих пацієнтів	Кількість невірно діагностованих пацієнтів	Достовірність діагностування пацієнтів
YourDiagnosis	100	84	16	84 %
Розроблена програма	100	91	9	91 %

Рисунок В.6 – Результати тестування розробленої програми та програми-аналога YourDiagnosis

## Додаток Г (довідниковий)

### Інструкція користувача

Після запуску програми відкриється головне вікно (рис.Г.1).

Рисунок Г.1 – Початкове вікно програми

Якщо нейронна мережа ще не навчена, то потрібно обрати вкладку «Навчання», після чого з'явиться вікно, представлене на рис. Г.2.

Рисунок Г.2 – Вкладка «Навчання» програми



Там можна обрати одну із 2 опцій:

- 1) завантажити навчену нейронну мережу із файла;
- 2) провести навчання нейромережі на нових даних.

У обох випадках з'явиться додаткове вікно навігатора для вказання місця знаходження потрібного файла на диску, як показано на рис. Г.3.

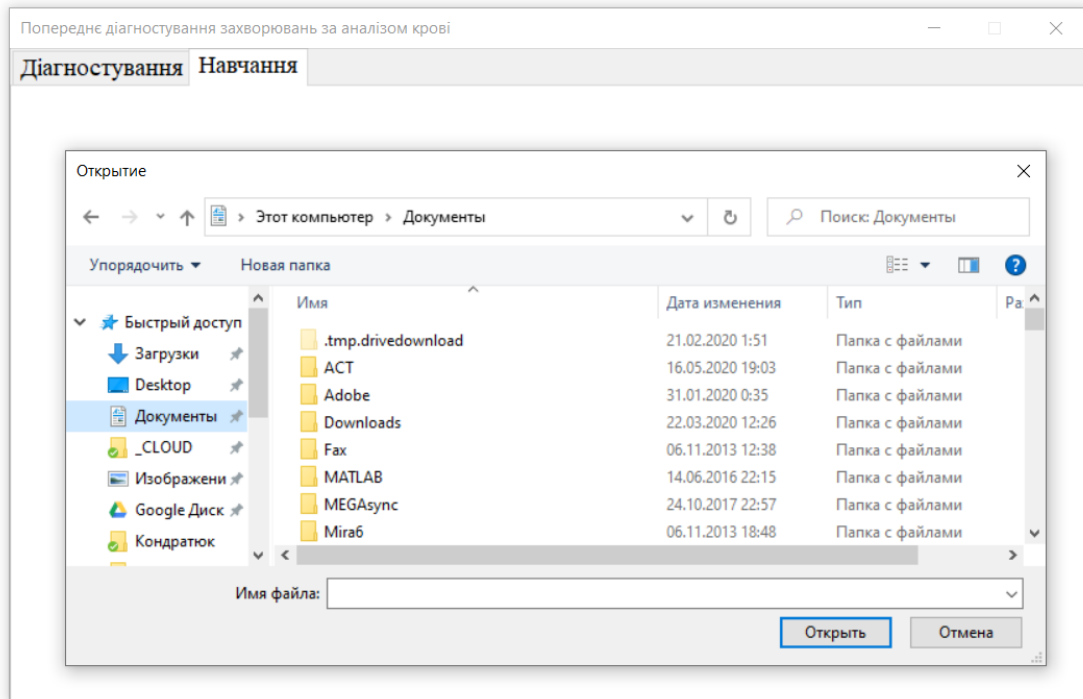


Рисунок Г.3 – Додаткове вікно навігатора для вказання місця знаходження потрібного файла на диску

Після проведення навчання мережа Кохонена готова до використання для попереднього діагностування пацієнтів і тому можна перейти на вкладку «Діагностування» та заповнити всі поля параметрами пацієнта (стать, вік) та його аналізу крові (як це показано на рис. Г.4).

Попереднє діагностування захворювань за аналізом крові

Діагностування Навчання

Стать:  Ч  Ж Вік:  До 15  15-20  50+

Введіть результати загального аналізу крові:

Лейкоцити WBC	<input type="text" value="5"/>	Тромбоцити PLT	<input type="text" value="210"/>
Еритроцити RBC	<input type="text" value="4.2"/>	Швидкість осідання еритроцитів - ШОЕ (ESR), Формула крові	<input type="text" value="9"/>
Гемоглобін HGB	<input type="text" value="132"/>	Нейтрофіли NEU	<input type="text" value="4"/>
Гематокрит HCT	<input type="text" value="0.44"/>	Лімфоцити LYMPH	<input type="text" value="2"/>
Середній об'єм еритроцитів MCV	<input type="text" value="87"/>	Моноцити MONO	<input type="text" value="0.3"/>
Середній вміст гемоглобіну в еритроциті MCH	<input type="text" value="29"/>	Еозинофіли EO	<input type="text" value="0.4"/>
Середня концентрація гемоглобіну в еритроциті MCHC	<input type="text" value="36"/>	Базофіли BASO	<input type="text" value="0.06"/>
Індекс розподілу еритроцитів RDW-CV	<input type="text" value="7.2"/>	<input type="button" value="Продіагностувати"/>	

Рисунок Г.4 – Заповнення на вкладці «Діагностування» всіх полів параметрів пацієнта (стать, вік) та його аналізу крові.

Далі потрібно натиснути на кнопку «Продіагностувати» і з'явиться вікно з результатом попереднього діагностування пацієнта (див. рис. Г.5).

Попереднє діагностування захворювань за аналізом крові

Результати попереднього діагностування

- Захворювання легенів
- Серцево-судинні захворювання
- Захворювання ЦНС
- Бактеріальна інфекція
- Вірусна інфекція
- Захворювання нирок
- Аутоімунні захворювання
- Ендокринологічні захворювання
- Злоякісні пухлини
- Анемія
- Діабет
- Захворювань не виявлено

Повернутися до вікна введення параметрів аналізу крові

Рисунок Г.5 – Вікно програми з результатом попереднього діагностування пацієнта

Із рис. Г.5 видно, що пацієнт має хворобу із групи «Ендокринологічні захворювання». Із цього вікна можна повернутись до вікна по рис. Г.4, натиснувши кнопку «Повернутися до вікна введення параметрів аналізу крові».