

Хмельницький національний університет  
Факультет програмування  
та комп'ютерних і телекомунікаційних систем  
Кафедра телекомунікацій, медійних та інтелектуальних технологій

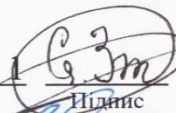
ДИПЛОМНА РОБОТА МАГІСТРА

Модель оптимізації робочого процесу сімейних лікарів на основі штучної  
нейронної мережі

Галузь знань: \_\_\_\_\_ 11 – Математика та статистика \_\_\_\_\_

Спеціальність: \_\_\_\_\_ 113 – Прикладна математика \_\_\_\_\_

Шифр ДРПМ. 15/022.01.15.00

Виконав: студент 2 курсу, група ПМм 19 – 1  \_\_\_\_\_  
Ініціали, прізвище

Керівник к.ф. -м.н., доцент \_\_\_\_\_  
Ініціали, прізвище

До захисту допускаю:  
Зав. кафедри ТМІТ, д.т.н., доцент \_\_\_\_\_  
Ініціали, прізвище

\_\_\_\_\_ 2020 р.

Хмельницький, 2020

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Програмування та комп'ютерних і телекомунікаційних систем

Кафедра телекомунікацій, медійних та інтелектуальних технологій

Освітній рівень магістр

Галузь знань 11 Математика та статистика

Спеціальність 113 Прикладна математика

Освітня програма освітньо-професійна програма підготовки магістра



ЗАТВЕРДЖУЮ

Зав. кафедри С.К.Підченко

“3.” 09 2020р.

**ЗАВДАННЯ**

**НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)**

Зембіцький Сергій Павлович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Модель оптимізації робочого процесу сімейних лікарів на основі штучної нейронної мережі

Керівник проекту (роботи) Грипинська Надія Василівна

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.09.2020 р. № 118

2. Строк подання студентом проекту (роботи) на кафедру 01.12.2020 р.

3. Вихідні дані до проекту (роботи) Модель оптимізації обробки звернення пацієнта (графічна, типу блок схема), модель інтерпретації відібраного анамнезу в вхідний вектор, модель бази знань алгоритму машинного навчання, модель бази знань планів лікування захворювань, формула обрахунку загального часу обробки звернення, web застосунок обробки звернення пацієнта.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області, формування проблематики та невирішених завдань, аналіз та вибір інструментарію для вирішення поставлених завдань, розробка моделей, розробка web застосунку, проведення практичної апробації, аналіз ефективності розробленого оптимізаційного рішення.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Графічні моделі (блок схеми), графічні об'єкти лістингу програмного коду, графічні об'єкти інтерфейсу користувача web застосунку, графічні об'єкти типу screenshots, графічні об'єкти скан - копій документів, матеріали для презентації.

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, д
		завдання видає
Перший	Козяр О.В. сімейний лікар	05.09.2020р

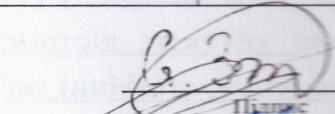
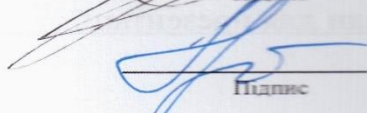
7. Дата видачі завдання « 05 » вересня 2020р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)
1. Аналіз предметної області	з 05.09.2020р. по 20.09.2020р.
2. Формування проблематики та невирішених завдань	з 21.09.2020р. по 01.10.2020р.
3. Аналіз та вибір інструментарію для вирішення завдань	з 02.10.2020р. по 12.10.2020р.
4. Розробка моделей та web застосунку	з 13.10.2020р. по 02.11.2020р.
5. Проведення практичної апробації	з 03.11.2020р. по 05.11.2020р.
6. Аналіз ефективності розробленого оптимізаційного рішення	з 06.11.2020р. по 07.11.2020р.
7. Написання фінальної версії текстової замітки	з 08.11.2020р. по 15.11.2020р.
8. Представлення роботи до захисту	з 04.12.2020р. по 15.12.2020р.

Студент

Керівник проекту (роботи)

  
 Підпис  
 С.П.  
 Івн  
  
 Підпис  
 Н.В.  
 Івн

## АНОТАЦІЯ

Тема дипломної роботи: Модель оптимізації робочого процесу сімейних лікарів на основі штучної нейронної мережі

Автор роботи: Зембіцький Сергій Павлович

Керівник роботи: Грипинська Надія Василівна

Загальний обсяг роботи: 136 сторінок, 31 рисунок, 10 таблиць, 8 додатків, 27 посилань.

АЛГОРИТМ НАЇВНОГО БАССА, ВЕБ-СЕРВІС ДІАГНОСТИКИ РЕСПІРАТОРНИХ ЗАХВОРЮВАНЬ, МЕТОД ОПОРНИХ ВЕКТОРІВ, МОДЕЛЬ ОБРОБКИ ЗВЕРНЕНЬ, ШТУЧНИЙ ІНТЕЛЕКТ.

Метою роботи є розробка моделі оптимізації обробки звернення пацієнта сімейним лікарем на основі алгоритмів машинного навчання.

В даній дипломній роботі було створено комплекс оптимізаційних моделей та програмне рішення для пришвидшення обробки звернення пацієнта з респіраторним захворюванням. Проведено практичну апробацію, отримано високу оптимізаційну ефективність та позитивний відгук від сімейного лікаря.

## ANNOTATION

a master's degree work of Serhii Zembitskyi  
entitled «Model of optimization work process of family doctors on the basis an artificial neural network».

Mentor: Nadiia Hrypynska

Total volume of work: 136 pages, 31 figures, 10 tables, 8 appendices, 27 references.

APPEAL PROCESSING MODEL, ARTIFICIAL INTELLIGENCE, NAIVE BAYES ALGORITHM, SUPPORT VECTOR MACHINE, WEB SERVICE OF DIAGNOSTIC RESPIRATORY DISEASE.

The aim of the work is to develop a model for optimizing the processing of patient treatment by a family doctor, based on machine learning algorithms.

In this diplomas work was created a set of optimizing models and software for speed up processing appeal of patients with respiratory diagnose. Done practical approbation, was get high optimization efficiency and positive feedback from family doctor.

2.12.20

Дата / Date



Підпис студента / Signature

## ЗМІСТ

СКРОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ .....	6
ВСТУП.....	7
1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧІ.....	12
1.1 Аналіз предметної області.....	12
1.2 Аналітичний огляд результатів обробки джерел інформації .....	23
1.3 Постановка задачі та завдань дослідження .....	30
2 ВИБІР ТА ОБГРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАРІЮ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ.....	33
2.1 Програмний комплекс та мова програмування.....	33
2.2 Методи та підходи представлення даних .....	37
2.3 Бази знань.....	42
2.4 Машинне навчання та алгоритми класифікації .....	44
3 РОЗРОБКА ОПТИМІЗАЦІЙНИХ МОДЕЛЕЙ ТА WEB ЗАСТОСУНКУ. ПРОВЕДЕННЯ ПРАКТИЧНОЇ АПРОБАЦІЇ ТА ТЕСТУВАННЯ. АНАЛІЗ ЕФЕКТИВНОСТІ РОЗРОБЛЕНИХ МОДЕЛЕЙ ТА WEB ЗАСТОСУНКУ .....	57
3.1 Розробка моделі оптимізації, моделі архітектури функціонування хмарного середовища та допоміжних моделей.....	57
3.2 Розробка програмного рішення .....	72
3.3 Проведення практичної апробації та аналіз ефективності моделей та їх програмної реалізації .....	83
ВИСНОВКИ.....	89
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	91
ДОДАТОК А ТАБЛИЧНО - ВЕКТОРНА ІНФОРМАЦІЯ ФАКТОРІВ ВПЛИВУ ТА КЛІНІКИ ЗАХВОРЮВАНЬ .....	94

ДОДАТОК Б SCREENSHOTS МОДУЛІВ WEB ЗАСТОСУНКУ.....	97
ДОДАТОК В ПРАКТИЧНА АПРОБАЦІЯ.....	105
ДОДАТОК Г ПОСІБНИК КОРИСТУВАЧА RSD:ML .....	107
ДОДАТОК Д ВІДГУК СІМЕЙНОГО ЛІКАРЯ.....	116
ДОДАТОК Е НАУКОВА СТАТТЯ. ВІСНИК ХНУ. ТЕХНІЧНІ НАУКИ. ....	117
ДОДАТОК Ж РЕЗУЛЬТАТ ПЕРЕВІРКИ НА АНТИПЛАГІАТ .....	121
ДОДАТОК К МАТЕРІАЛИ ПРЕЗЕНТАЦІЇ ДЛЯ ЗАХИСТУ ДРМ .....	122

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

ЦПМСД — центр первинної медико – санітарної допомоги,

МІС — медична інформаційна система,

ЕМК — електронна медична карта,

НСЗУ — національна служба здоров'я України,

СКБД — система керування базою даних,

НБА — наївний баєсовський алгоритм,

ОЗП — оперативний запам'ятовуючий пристрій,

АК — амбулаторна карта пацієнта,

SPA (Single Page Application) — односторінковий застосунок,

API (Application programming interface) — програмний інтерфейс застосунку,

САПР — система автоматизації та прийняття рішень,

АНБ — алгоритм наївного Баєса,

МОВ — метод опорних векторів,

к – бс — метод к – ближніх сусідів,

лс — локальний сервер,

хс — хост – сервер.

## ВСТУП

**Актуальність теми.** У зв'язку з проведенням в Україні медичної реформи, а саме реформи первинної ланки медицини, сімейні лікарі стали витрачати багато часу на обробку анамнезу, встановлення діагнозу та розробку плану лікування. Таким чином постає проблема щодо зменшення витрати часу. Тому виникає актуальність проведення даного дослідження та розробки моделі оптимізації для реалізації програмного рішення, яке надалі дасть змогу вирішити дану проблему.

Використання математичного апарату, комп'ютерних систем та методики алгоритмів машинного навчання міцно увійшло в медицину, тому на даний момент існують напрацювання в галузі математичної медицини. В наукових роботах представленні методи та підходи до інтелектуальної обробки анамнезу. Основна проблематика, яку вирішують науковці це збір, обробка та аналіз анамнезу пацієнтів, прогнозування діагнозів, обробка даних електрокардіограми, класифікація медичних записів. Науковець з Каліфорнійського університету Вільям Бакст зробив вагомий внесок в діагностування інфаркту міокарда. За допомогою штучної нейронної мережі, використавши 20 параметрів, він розробив систему, яка діагностує інфаркт міокарда з успішністю 92%. Компанія Google на даний час проводить розробку системи класифікації медичних записів під назвою DeepMind Health, також компанія MadyMatch Technology розробила штучну нейронну мережу для діагностування інсульту. Компанія NEVLABS розробила програмне рішення діагностики респіраторних захворювань.

**Стисла оцінка сучасного стану проблеми.** В ході дослідження з'ясувалося, що сімейний лікар найбільше часу витрачає на обробку анамнезу пацієнтів, розробку плану лікування та фіксацію звернення пацієнта в амбулаторну карту. Тому в подальшому дослідженні набуває актуальності потреба використання алгоритмів машинного навчання для обробки анамнезу пацієнта, прогнозування діагнозу, розробки плану лікування та створення запису до амбулаторної карти пацієнта.

Галузь для якої буде проводитися дослідження є первинна ланка медицини. Вхідними даними слугують проведені дослідження з експертною групою сімейних лікарів на базі практики в кабінеті медичної практики сімейної медицини ФОП Козяр О.В. та протоколи доказової медицини, розроблені світовими організаціями охорони здоров'я. Вихідними даними є прогнозований або встановлений діагноз, план лікування пацієнта та запис до амбулаторної карти пацієнта.

Розроблена інформаційно — аналітична система, в подальшому розвитку галузі, допоможе швидше опрацьовувати вхідну інформацію звернення пацієнта та пришвидшить обробку анамнезу пацієнта. Інформаційно — аналітична система також зможе слугувати навчально — інформаційним засобом для лікарів інтернів.

Проаналізувавши ситуацію, яка склалася в даній галузі, можна зробити висновки, що представленні рішення не повністю автоматизують та пришвидшують обробку анамнезу пацієнта. Тому основною задачею даного наукового дослідження є розробка моделі оптимізації обробки звернення пацієнта та розробка програмного забезпечення для реалізації даної моделі з використанням алгоритмів машинного навчання та подальшого розгортання на серверах з технологією хмарного обчислення.

**Зв'язок з напрямками наукових досліджень випускної кафедри.** Кафедра телекомунікацій, медійних та інтелектуальних технологій вивчає методи штучного інтелекту та підходи до машинного навчання. Тематика даної наукової роботи охоплює використання алгоритмів машинного навчання для вирішення поставлених задач. Тому присутній прямий зв'язок даної наукової роботи з кафедрою телекомунікацій, медійних та інтелектуальних технологій.

**Мета дослідження.** Розробка моделі оптимізації обробки звернення пацієнта, сімейним лікарем, на основі алгоритмів машинного навчання.

**Гіпотеза дослідження.** Використання алгоритмів машинного навчання зменшить витрату часу на обробку анамнезу респіраторного захворювання пацієнта, розробку плану лікування та фіксацію звернення пацієнта до сімейного лікаря в амбулаторну карту пацієнта.

**Об'єкт дослідження.** Мінімізація часу обробки епізоду звернення пацієнта.

**Предмет дослідження.** Модель оптимізації обробки анамнезу пацієнта.

**Завдання дослідження наступні:**

- розробити модель оптимізації обробки анамнезу та розробки плану лікування на основі алгоритмів машинного навчання;
- розробити програмне рішення практичної реалізації моделі оптимізації;
- провести аналіз отриманих результатів практичної апробації програмного рішення;
- зробити висновки про успішність проведення наукової роботи.

**Стислий огляд інструментарію дослідження.** В даній науковій роботі використовувалися теоретичний та емпіричний методи дослідження. Саме ці методи в сукупності дали змогу сформулювати основні поняття про базові процеси, які відбуваються в сімейного лікаря при зверненні пацієнта. За допомогою теоретичного методу було сформовано основний алгоритм роботи сімейного лікаря при зверненні пацієнта. Опрацьовано методики та підходи до встановлення діагнозу для створення бази знань класифікації діагнозу та розробки плану лікування пацієнта. Опрацьовано доказові клінічні протоколи щодо проведення лікування пацієнта на основі яких було розроблено базу знань препаратів для лікування захворювання. Також було опрацьовано ключові моменти реалізації вузлів моделі обробки анамнезу для побудови та навчання алгоритму штучного інтелекту. За допомогою емпіричного методу дослідження було проведено пошук оптимального алгоритму машинного навчання, його тестування та конфігурування, подальшу практичну апробацію програмної реалізації моделі. Засобами наукового дослідження є програмно апаратний комплекс. В якості середовища розробки використовувалися програмний комплекс Microsoft Visual Code, мова програмування PHP, мова розмітки HTML, мова стилізації вебсторінок CSS та скриптова мова веб сценаріїв JavaScript. Апаратну частину забезпечував локальний комп'ютер web – сервер для відладки програмного продукту та хостинг – сервер для практичної апробації програмного продукту.

**Наукова новизна.** Запропонована модель, яка структурує, формує чіткий алгоритм дій сімейного лікаря, створює шаблон обробки звернення пацієнта та

пришвидшує його. Також науковою новизною є модульність архітектури побудови середовища обробки анамнезу. Зі сторони розробки модульність виражається в можливості легкої та швидкої інтеграції нового розробленого модулю, а зі сторони користувача модульність виражається в можливості використання незалежно один від одного модулів встановлення діагнозу, розробки плану лікування та створення фіксації звернення пацієнта в амбулаторну карту. Інноваційний підхід до мінімізації вхідного вектору анамнезу звернення пацієнта. Формування плану лікування з семплів інформації про препарат та його застосування.

**Практична значущість.** Практичне значення розробленої моделі виражається в можливості програмної реалізації хмарного обчислювального середовища для обробки анамнезу пацієнта. Програмна реалізація допоможе сімейному лікарю оптимізувати робочий процес, зменшити витрату часу на звернення пацієнта та обробку його анамнезу захворювання, встановити загальний шаблон алгоритму та порядок дій проведення та обробки прийому пацієнта.

**Структура основної частини.** Складається з наступних розділів:

Перший розділ — Стан питання та постановка задачі.

Другий розділ — Вибір та обґрунтування вибору інструментарію для вирішення поставленої задачі.

Третій розділ — Розробка моделі оптимізації та web застосунку. Проведення практичної апробації та тестування. Аналіз ефективності розроблених моделей та web застосунку.

В першому розділі проведено ознайомлення з галуззю дослідження та особливостями роботи сімейного лікаря. Проаналізовано існуючі рішення серед МІС, рекомендованих eHealth, та наукову літературу. Визначено проблематику та актуальність наукового дослідження. Висунуто гіпотезу, сформовано задачу та завдання дослідження.

В другому розділі досліджено та обрано існуючий інструментарій для вирішення поставленої задачі. Вивчено математичний апарат та логіку роботи алгоритмів класифікації з машинним навчанням, а саме алгоритм наївного Баеса, метод опорних векторів та метод  $k$  – ближніх сусідів.

В третьому розділі представлено розробку моделей та ключові моменти розробки web застосунку. Проведено модульне тестування та практичну апробацію в реальних умовах застосунку розміщеного на хостинг сервері. Також протестовано коректність роботи алгоритму машинного навчання. Проведено аналіз ефективності використання розробленого оптимізаційного рішення для сімейних лікарів.

**Публікації та апробація результатів дослідження.** Розроблений хмарний аналітичний комплекс за представленою моделлю було розгорнуто на хостинг сервері та апробовано спільнотою сімейних лікарів, відгук представлено у додатку Д на рисунку Д.1. Також було опубліковано статтю в науковому журналі «Вісник ХНУ технічні науки» у розділі системний аналіз та кібербезпека на тему «Проектування архітектури хмарного обчислювального середовища для обробки анамнезу пацієнтів з захворюванням дихальної системи на основі алгоритмів машинного навчання».

## 1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Аналіз предметної області

В даному розділі проводиться аналіз предметної області загальної медичної практики. Відбувається ознайомлення з основними структурними елементами функціонування ЦПМСД також проводиться аналіз функціональної, організаційної та технічної структури. Проводиться ознайомлення з основними ключовими моментами реформи сімейної медицини. Вивчається реальний стан справ на момент проведення наукової роботи, та формуються проблемні аспекти даної галузі.

Трансформація системи охорони здоров'я стосується кожного. Її мета — забезпечити всім громадянам України рівний доступ до якісних медичних послуг та перебудувати систему охорони здоров'я так, щоб у її центрі був пацієнт [1]. Однією з основних змін закладених в реформі медицини є запровадження системи eHealth. Застаріла паперова звітність піде в небуття. Всю медичну документацію буде переведено в електронний вигляд. Це розвантажить лікарів, дозволить їм більш якісно та оперативно надавати медичні послуги пацієнтам та унеможливить ситуацію втрати медичних даних пацієнтів. Також це дозволить збирати дані щодо необхідних послуг в окремих районах, більш точно розраховувати тарифи та контролювати якість медичної допомоги. Вже у 2019 році планується запуск нових електронних інструментів — електронної медичної картки, електронного направлення, електронного рецепта на “Доступні ліки” та ін [1].

Нижче розглянуто основні структурні елементи сімейної медицини.

ЦПМСД — лікувально — профілактичний заклад, який працює на принципах сімейної медицини та надає можливість проводити діяльність сімейному лікарю. Основними завданнями ЦПМСД є забезпечення проведення діагностики, надання консультацій та лікування пацієнтів.

WONCA – світова організація сімейних лікарів. Заснована в 1972 році членами – організаціями у 18 країнах світу. На даний момент WONCA складається зі 118 членів – організацій в 131 країні світу і територіями що мають близько 500000 сімейних лікарів та понад 90% населення світу. Також до складу організації входять 21 член академічних кафедр сімейної медицини. Основною місією WONCA є покращення якості життя людей шляхом визначення та пропагування їх цінностей, включаючи повагу до прав людини, включаючи гендерну справедливість та створення високих стандартів надання первинної медико – санітарної допомоги сімейними лікарями. Члени організації WONCA дотримуються наступних принципів:

- турбота про пацієнта та його сім'ю в контексті суспільства та громади;
- надання медичних послуг пацієнту незалежно від його статусу та приналежності до соціальних груп;
- створення навчальних академічних закладів для підвищення рівня кваліфікації членів організації;
- створення підтримки сімейних лікарів та створення форумів для організації обміну знаннями та досвідом між членами організації;
- представлення політики, освітньої та дослідницької діяльності іншим світовим спільнотам лікарів [2].

НСЗУ – Національна служба здоров'я України є центральним органом виконавчої влади, який реалізує державну політику у сфері державних фінансових гарантій медичного обслуговування населення. Діяльність НСЗУ спрямовується і координується Кабінетом Міністрів України через Міністра охорони здоров'я. Основні завдання НСЗУ:

- реалізація державної політики у сфері державних фінансових гарантій медичного обслуговування населення за програмою державних гарантій медичного обслуговування населення (програма медичних гарантій);
- виконання функцій замовника медичних послуг та лікарських засобів за програмою медичних гарантій;

- внесення на розгляд Міністра охорони здоров'я пропозицій щодо забезпечення формування державної політики у сфері державних фінансових гарантій медичного обслуговування населення [3].

Сімейна медицина це комплекс теоретичних і практичних підходів для організації сімейної лікувальної справи. Сімейна лікувальна практика — це практика, яка забезпечує тривалий догляд за станом здоров'я пацієнта та всіх членів його сім'ї. Загальна практика має широкий погляд на клінічні проблеми пацієнта. Сімейна медицина використовує наступні принципи:

- довготривалість спостереження;
- багатопрофільність первинно — медичної допомоги;
- координація медичної допомоги;
- відношення до сім'ї, як до одиниці медичного обслуговування [4].

Також одним із важливих підходів є профілактична спрямованість. Саме за допомогою профілактики можна уникнути ризику появи різних захворювань.

Сімейний лікар — лікар загальної практики, який отримав підготовку з надання первинної медико – санітарної допомоги та проводить діяльність у сфері сімейної медицини. Сімейний лікар працює за наступними основними вимогами WONCA:

- надання первинної медичної допомоги — вміння проводити первинне обстеження пацієнта та подальша координація діяльності вузьких фахівців, залучених до допомоги пацієнту;
- особлива увага до інтересів пацієнта — здатність приділяти увагу до інтересів пацієнта, вміння встановлення продуктивних відносин з пацієнтом і використання цих навичок при роботі з хворим, забезпечення безперервності надання меддопомоги тривалий час;
- вміння вирішувати певні проблеми пацієнта — прийняття рішення на основі знань про захворюваність та поширеність даної патології серед даної групи населення, лікування хвороб на ранніх стадіях та негайне надання допомоги за нагальної необхідності;

- комплексний підхід — одночасне лікування як гострих, так і хронічних станів пацієнта та організація для пацієнта необхідних оздоровчих і лікувальних заходів;
- соціальна спрямованість — узгодження потреб окремих пацієнтів з потребами всієї групи, що обслуговується та ресурсами, що наявні;
- цілісне уявлення про захворювання — вміння застосовувати біо та психо—соціальну модель з урахуванням культурних та екзистенціальних аспектів життя людей.

Пацієнт — людина, яка отримує медичну допомогу.

Анамнез — основний суб'єктивний метод дослідження пацієнта, який полягає в отриманні інформації про нього, виникнення і перебіг захворювання. Дані одержують під час валеологічного, психологічного, медичного обстеження людини шляхом розпитування. Збір анамнезу передуює об'єктивному дослідженню. Уміло зібраний анамнез дає змогу встановити контакт із пацієнтом, що важливо для подальшого встановлення діагнозу і прогнозу, правильного тлумачення даних об'єктивних досліджень, а також вибору оптимальних методів лікування та профілактики. Методика збирання анамнезу полягає в опитуванні самого хворого або осіб, що оточують його. Лікар збирає анамнез за традиційною схемою. Потім, встановлюючи діагноз і прогноз, призначаючи лікувально—профілактичні заходи, зіставляє анамнез із даними клінічного обстеження пацієнта [5].

План лікування — розроблений сімейним лікарем алгоритм дій для лікування пацієнта, який містить в собі інформацію про торгову назву препарату необхідного для застосування, дози діючої речовини, кількості разів на день, кількості днів та особливостей вживання препарату.

Протоколи доказової медицини — чіткий алгоритм дій лікаря, який сприяє узгодженості надання медичної допомоги пацієнтам на усіх рівнях, а також гарантує, що кожен пацієнт отримає високий рівень надання медичної допомоги. Міжнародні протоколи лікування орієнтовані на потреби пацієнта, тому їх впровадження сприятиме покращенню їх стану здоров'я. Протоколи лікування пропонують чіткі рекомендації для лікарів, а також підвищують якість клінічних рішень, у тому числі й серед тих фахівців, які звикли до застарілої медичної

практики. Для системи охорони здоров'я важливість впровадження клінічних протоколів пов'язана з підвищенням ефективності роботи (часто шляхом стандартизації надання медичної допомоги пацієнтам) та оптимізації вартості лікування [6].

База протоколів доказової медицини – інформаційне сховище, яке містить в собі протоколи доказової медицини. Доступ до сховища забезпечується виключно фахівцям медичної галузі.

МКХ 10 — документ, який використовується як провідна статистична та класифікаційна основа в системі Охорони здоров'я. Метою МКХ є створення умов для систематизованої реєстрації, аналізу, інтерпретації та порівняння даних про смертність та захворюваність, отриманих в різних країнах або регіонах та в різний час. МКХ використовується для перетворення словесного формулювання діагнозів хвороб та інших проблем, пов'язаних зі здоров'ям, у коди, які забезпечують зручність збереження, збору та аналізу даних. МКХ стала міжнародною стандартною діагностичною класифікацією для всіх загальних епідеміологічних цілей та багатьох цілей, пов'язаних з управлінням системою охорони здоров'я. Ці цілі включають аналіз загальної ситуації зі здоров'ям груп населення, а також підрахунок частоти і поширеності хвороб та інших проблем, пов'язаних зі здоров'ям, в їхньому взаємозв'язку з різними факторами [7].

Система шифрування ІСРС 2 — міжнародна класифікація первинної медичної допомоги (International Classification for Primary Care — ІСРС) розроблена Всесвітньою організацією сімейних лікарів WONCA для ефективного ведення пацієнтів на первинній ланці. Класифікатор аспектів сімейної медицини, таких як причини звернення та проблеми зі здоров'ям. Класифікація ІСРС—2 використовується як для організації і зберігання медичних даних пацієнтів, так і для потреб статистики. Для медичних карток потрібно якомога більше конкретних деталей, тоді як для статистики необхідні згруповані дані залежно від їх частоти або їх значення для формування статистики [8].

Гострі респіраторні захворювання — збірна група захворювань, спричинених широким спектром інфекційних збудників і поєднаних спільними особливостями

епідеміології та патогенезу ураження дихальних шляхів, розвитком респіраторного синдрому і загальноінтоксикаційних проявів різного ступеня вираженості. При ГРЗ респіраторний синдром, на відміну від багатьох інфекційних (кір, краснуха, ротавірусний гастроентерит тощо) та неінфекційних хвороб (хронічний фарингіт курців, поліпозна риносинусопатія, алергічний ларингіт тощо), за яких ураження дихальних шляхів є лише однією з багатьох ознак клінічної картини, завжди є провідним. ГРЗ — поняття, що включає респіраторні хвороби вірусного та бактерійного походження, в тому числі й такі, що проявляються синдромом локального ураження дихальних шляхів — ларингіт, тонзиліт, фарингіт, бронхіт тощо [9].

Хронічне респіраторне захворювання — це хронічне захворювання дихальних шляхів та інших легеневих структур.

Також в ході ознайомлення з областю дослідження було сформовано основні поняття клініки розпоширених респіраторних захворювань для формування бази знань алгоритму машинного навчання. Клініку респіраторних захворювань представлено нижче:

Синусит — запалення слизової оболонки верхніх дихальних шляхів. Клініка: гарячка, головний біль, втомлюваність, закладеність вух, температура до 37,5, порушення нюху, виділення слизу та гною, неприємний запах з роту, зубний біль, кашель присутній, порушення носового дихання.

Фарингіт — запальний процес у слизовій, лімфоїдній тканинах і в гландах глотки. Клініка: температура 37, виділення слизу та гною, біль у горлі, біль при ковтанні, почервоніння горла, виразка,

Тонзиліт — гостре інфекційно - алергічне захворювання, що характеризується гострим запаленням піднебінних мигдаликів та інших елементів лімфаденоїдного глоткового кільця. Клініка: головний біль, ломота в суглобах, озноб, температура 37.5, збільшені лімфатичні вузли, біль у горлі, гній на мигдаликах.

Тонзилофарингіт — гостре інфекційне захворювання глотки, або піднебінних мигдаликів. Клініка: гарячка, температура 38, гній на мигдаликах, запалення мигдаликів.

Ларингіт — гостре запалення слизової оболонки гортані та голосових складок. Клініка: погіршений загальний стан, температура 37.5, кашель сухий або вологий, біль при ковтанні, хриплість голосу, утруднене дихання.

Трахеїт — запальний процес слизової оболонки в хрящових трубах в нижньому відділі дихальних шляхів на шляху між бронхами та гортанню. Клініка: температура 38, біль у горлі, біль у глотці, біль при ковтанні, почервоніння горла.

Бронхіт — запалення бронхів. Клініка: головний біль, температура до 38, кашель вологий, біль у горлі, важкість дихання, хрипи.

Плеврит — запальний процес вісцерального та парієтального листа. Клініка: погіршення загального стану, біль у грудях, температура 38, присутній кашель, прослуховується шум, присутня задишка.

Пневмонія — інфекційне запалення легень. Клініка: погіршення загального стану, біль у грудях, слабкість, температура до 40, кашель сухий, присутнє виділення гнійного мокротиння, присутня задишка, важке дихання, вологі хрипи.

Поліпи в носі — аномальне утворення сформоване над слизовою оболонкою. Клініка: головний біль, поганий сон, закладеність вух, температура 36.6, знижене відчуття запахів, порушення носового дихання,

Емфізема легень — патологічний стан легеневої тканини, що характеризується підвищеним вмістом повітря. Клініка: температура 36.6, присутнє покашлювання, задишка, ортопноє, відчуття нехватки кисню, послаблене дихання.

Бронхіальна астма (легка, важка форма) — патологічний стан дихальних шляхів запального характеру, що супроводжується обструктивним синдромом. Клініка: температура 36.6, періодичне виникнення сухого кашлю, важкість дихання, свистяче дихання, задишка, приступи.

Абсцес легені — патологічний процес в наслідок якого формується порожнина в легеневій тканині. Клініка: погіршений загальний стан, біль у грудях,

температура 37, неприємний запах з роту, виділення гнійного мокротиння, бронхіальне дихання, вологі хрипи.

Дихальна недостатність — патологічний стан в наслідок якого відбувається порушення газового складу в крові в бік зниження оксигенації або підвищення кількості вуглекислого газу. Клініка: сірий відтінок шкіри, температура 36.6, задишка, відчуття нехватки кисню, збільшена частота дихання [10].

Далі розглянуто основні процеси та взаємозв'язки між процесами або елементами, які відбуваються в ЦПМСД.

Відбір анамнезу – процес при якому сімейний лікар методом опитування пацієнта збирає потрібну інформацію для подальшого встановлення діагнозу. Анамнез відбирається за наступною схемою:

- паспортна частина: проводиться відбір особових даних, таких як прізвище, ім'я, по батькові, вік, адреса, контактний телефон, місце роботи, посаду, наявність інвалідності, причину встановлення інвалідності, протипоказання;
- скарги хворого: проводиться відбір інформації про скарги пацієнта, наявність фонових і супутніх захворювань;
- анамнез хвороби - відомості про виникнення та перебіг: із якого часу пацієнт вважає себе хворим, як починалося захворювання (раптово, поступово) із чим пов'язує його виникнення, як перебігало до моменту звернення до лікарні, чи звертався за медичною допомогою, яке лікування проводилося та його ефективність;
- загальний анамнез - суб'єктивна характеристика пацієнта функціонального стану найважливіших органів і систем його організму, серед яких — не враховані до основного патологічного процесу;
- анамнез життя - значна частина анамнезу, потрібна для вивчення фізичного, психічного і соціального розвитку обстежуваного.

Деякі етапи відбору анамнезу за потреби сімейного лікаря можуть бути виключені зі схеми.

Аускультация — вислуховування звукових феноменів, пов'язаних із діяльністю внутрішніх органів, застосовуване для оцінки цієї діяльності та

діагностики її порушень. Акти дихання, скорочення серця, моторики шлунка і кишок викликають у тканинних структурах пружні коливання, частина яких досягає поверхні тіла. Ці коливання зазвичай нечутні на відстані, але можуть прослухуватися, якщо прикласти вухо до тіла хворого або через прилад для аускультативної діагностики. Аускультативні ознаки, які супроводжують діяльність різних органів, являють собою шуми різного тембру й тривалості. За акустикою та клінікою можна розрізняти аускультативні ознаки низько—, середньо— і високочастотні з характерним діапазоном частот. У процесі аускультативної діагностики необхідно звертати увагу на вид дихання, на співвідношення дихальних шумів із фазами вдиху і видиху. Під час аускультативної діагностики необхідно дотримуватися повної тиші у приміщенні, в якому обстежується хворий [11].

Пальпація — процес заснований на дотикальному відчутті руки, що виникає при русі і тиску пальців та долоні. За допомогою пальпації визначають властивості тканин та органів: їхній стан, величину, форму, консистенцію, рухливість, топографічні співвідношення, а також болючість досліджуваного органа. Розрізняють поверхневу і глибоку пальпацію. Поверхневу пальпацію проводять однією або обома долонями, покладеними плазом на досліджувану область. Глибоку пальпацію здійснюють спеціальними методами при дослідженні внутрішніх органів.

Обробка анамнезу — процес при виконанні якого сімейний лікар користуючись здобутими теоретичними та практичними знаннями проводить встановлення діагнозу згідно з відібраним анамнезом пацієнта, який звернувся за допомогою.

Розробка плану лікування — процес при якому згідно з встановленим діагнозом сімейний лікар проводить формування плану лікування пацієнта.

Звернення пацієнта — процес звернення пацієнта до сімейного лікаря про надання первинної медико — санітарної допомоги. Даний процес поділяється на наступні етапи:

- відбір анамнезу;
- обробка анамнезу;

- встановлення діагнозу;
- розробка плану лікування згідно з встановленим діагнозом;
- фіксація звернення пацієнта.

Результатом виконання даного процесу є встановлений діагноз, розроблений план лікування та зафіксоване в амбулаторній карті звернення пацієнта

Фіксація звернення пацієнта — процес шифрування, фіксації звернення пацієнта за системою ІСРС 2 та запис в амбулаторну карту пацієнта або ЕКП.

Амбулаторна карта пацієнта або Форма 025/о – первинний медичний документ хворого, який лікується амбулаторно або вдома. Містить в собі персональну інформацію про пацієнта та зафіксовані звернення пацієнта до лікаря.

МІС — комплексний програмний продукт, головне призначення якого автоматизація всіх основних процесів, пов'язаних із роботою медичних установ загальної і вузької спеціалізації. Автоматизовані медичні інформаційні системи дозволяють швидко й ефективно налагодити електронний документообіг, гнучко вибудовувати роботу з пацієнтами, вести оперативний облік роботи адміністративного персоналу, контролювати всі організаційні і фінансові питання [12].

Електронний кабінет пацієнта — програмний комплекс для доступу пацієнта до електронної медичної інформації. Також використовується для обміну медичною інформацією між сімейним лікарем та пацієнтом.

ЕМК — цифрова версія амбулаторної карти пацієнта, яка містить персональну інформацію про пацієнта та зафіксовані звернення пацієнта до лікаря. Зберігається в базах даних в локальних або хмарних дата центрах.

eHealth — електронна система охорони здоров'я, що забезпечує обмін медичною інформацією та реалізацію програми медичних гарантій населення.

Система eHealth складається з:

- центральної бази даних — ЦБД (адміністратор ДП “Електронне здоров'я”);
- електронних медичних інформаційних систем — МІС (системи, які дають змогу автоматизувати роботу медзакладів з ЦБД) [13].

Міністерство охорони здоров'я регулює впровадження eHealth на нормативно—правовому рівні. Національна служба здоров'я України укладає договори з постачальниками медичних послуг, аналізує та використовує дані з метою прогнозування потреб населення у медичних послугах, розробки програми медичних гарантій, встановлення тарифів, здійснення моніторингу договорів, здійснює оплати за медичні послуги на умовах договору.

Після ознайомлення з основними елементами та процесами функціонування сімейної медицини та ЦПМСД було проведено спостереження за виконанням робочого процесу сімейного лікаря. Дослідивши та проаналізувавши роботу сімейного лікаря було визначено, що найчастіше сімейний лікар зустрічається з випадками респіраторного захворювання, а найбільше часу сімейний лікар витрачає на обробку звернення пацієнтів та роботу з пацієнтами. Дослідження проводилося на базі переддипломної практики, в кабінеті медичної практики сімейної медицини ФОП Козяр О.В., разом з сімейним лікарем. Метою дослідження було:

- визначення загального алгоритму дій при роботі з пацієнтом з респіраторним захворюванням;
- фіксація часу проведення всіх етапів звернення пацієнта.

Проаналізувавши логування спостереження було сформовано загальну картину обробки звернення пацієнта. Таким чином звернення пацієнта поділяється на наступні етапи з наступною приблизною тривалістю:

- відбір первинної інформації про пацієнта — до 3 хвилин;
- визначення стану самопочуття пацієнта — від 5 до 10 хвилин;
- огляд носа пацієнта — менше 1 хвилини;
- огляд ротової порожнини пацієнта — менше 1 хвилини;
- огляд горла пацієнта — менше 1 хвилини;
- визначення характеристик кашлю в пацієнта — менше 1 хвилини;
- аускультация дихальної системи пацієнта — від 5 до 10 хвилин;
- розробка плану лікування пацієнта — від 5 до 10 хвилин;
- фіксація звернення пацієнта — від 10 до 15 хвилин.

Як видно з вище описаного, в середньому звернення одного пацієнта з респіраторним захворюванням та робота з ним може тривати від 30 до 50 хвилин. При тому, що рекомендований МОЗ час обробки звернення пацієнта від 15 до 20 хвилин. В основному більшість часу витрачається на формування та запис плану лікування та фіксацію звернення пацієнта. Також даний факт можна підтвердити прочитавши різні тематичні лікарські видання, одним із яких є Всеукраїнська медична газета «Ваше здоров'я». У виданні представлено текстову версію коментаря сімейного лікаря про спрощення та оптимізацію обробки звернення пацієнта. «У який спосіб можна було б спростити роботу сімейного лікаря? — Передусім через скорочення часу на діяльність, яку можна автоматизувати й оцифрувати. До речі, дуже багато часу «з'їдає» саме виписування рецептів, особливо за програмою «Доступні ліки». Трапляється, за один день ми надаємо по 30 таких документів — усі вони мусять бути в електронному вигляді, але такої можливості в нас поки немає. Так само певний час доводиться витрачати на заповнення картки хворого. Аби все це було в цифровому вигляді (як і має бути), наша робота суттєво спростилася б, а зекономлені кілька хвилин можна було б присвятити пацієнту. Бо на практиці під час прийому ми витрачаємо більше часу на писанину, аніж на хворого.» [14].

Таким чином проаналізувавши вище описану ситуацію можна зробити висновок, що основним вагомим ресурсом для сімейного лікаря є час і найбільше часу витрачається на формування плану лікування та фіксацію звернення пацієнта. В наступному підрозділі буде проаналізовано наявні на даний момент підходи до вирішення вище описаної задачі.

## 1.2 Аналітичний огляд результатів обробки джерел інформації

В даному підрозділі проводиться огляд та аналіз джерел інформації. Проводиться аналіз наявності програмних реалізацій для вирішення задачі. Також

проводиться аналіз наукових підходів. Огляд та аналіз здійснювався серед програмних рішень МІС, які рекомендують eHealth. Проаналізувавши наявні МІС було виявлено ряд недоліків, про які викладено нижче.

Не всі МІС мають можливість запуску на хмарних платформах, таким чином кінцевому користувачу потрібно думати про обчислювальний ресурс пристрою на якому проводиться робота з МІС.

Не всі МІС мають можливість працювати під мобільною платформою. Відбувається розтрата трафіку та збільшується час завантаження МІС, якщо користувач використовує мобільну платформу.

Повна відсутність автоматизованого визначення діагнозу. Відсутня можливість автоматизованої допомоги сімейному лікарю при визначенні діагнозу. Реалізувавши алгоритм машинного навчання можна було б допомогти сімейному лікарю швидше визначати діагноз. З кожним новим випадком система проводила б перенавчання та допомагала точніше встановлювати респіраторне захворювання. Таким чином сімейний лікар, надавши системі вхідну інформацію в вигляді анамнезу звернення пацієнта, зможе отримати прогнозований діагноз. Також введений на попередньому етапі анамнез пацієнта можна використати для формування повного та детального запису звернення пацієнта до амбулаторної карти.

Відсутність змоги працювати з різними системами класифікації та шифрування звернення пацієнта. У зв'язку з реформуванням сімейної медицини, сімейні лікарі змушені використовувати класифікатор ІСРС 2, оскільки всі сімейні лікарі раніше працювали за МКХ 10, то вони вимушені вивчати новий класифікатор ІСРС 2 для того, щоб обрати необхідний діагноз та надалі проводити відповідні дії, що не є зручно. Зі сторони зручності користувачу потрібно надати змогу одночасно працювати з різними системами шифрування діагнозу такими як МКХ 10 та ІСРС 2. Також потрібно надати можливість конвертації шифру між різними системами класифікації. Тоді користувач зможе з легкістю знайти діагноз за МКХ 10, а блок шифрування сам проведе класифікацію за системою ІСРС 2. Використовуючи дану

можливість користувач зможе менше витратити часу на пошук діагнозу за новим класифікатором ІСРС2.

Відсутність автоматизованого шифрування звернення пацієнта згідно з класифікатором ІСРС 2. Користувачу потрібно самостійно обрати код шифрування згідно з класифікатором ІСРС 2. Як і в попередньому випадку реалізація можливості автоматичного шифрування звернення пацієнта згідно з класифікатором ІСРС 2 може пришвидшити обробку звернення пацієнта.

Відсутність можливості автоматизованого формування електронного плану лікування пацієнта згідно з встановленим діагнозом. Сімейний лікар змушений витратити час на формування плану лікування пацієнта навіть в МІС. Автоматизувавши даний процес за допомогою алгоритмів машинного навчання можна створити систему яка після встановлення діагнозу сформує план лікування а лікарю залишиться переглянути його і за потреби відредагувати. Таким чином реалізація даного функціоналу також може привести до зменшення витрати часу на обробку звернення пацієнта.

Відсутність можливості друку розробленого плану лікування. Не у всіх пацієнтів є змога встановити електронний кабінет пацієнта, щоб в ньому знайти та побачити план лікування, також не у всіх МІС є розроблений електронний кабінет пацієнта. В такій ситуації сімейний лікар змушений переписувати план лікування на звичайний лист, що займає багато часу, тому реалізувавши можливість друку можна автоматизувати даний процес та зменшити час роботи з пацієнтом.

Відсутність можливості друку електронної медичної карти. Електронні медичні карти не повністю введені в дію, тому сімейний лікар вимушений фіксувати звернення в електронному вигляді в МІС та паралельно фіксувати в звичайній амбулаторній карті. При відсутності можливості друку сімейний лікар переписуючи інформацію з екрану в амбулаторну карту пацієнта буде витратити багато часу на ту роботу, яку можна автоматизувати.

Відсутність можливості формування детального опису звернення пацієнта, який включає клініку (анамнез) звернення, встановлений діагноз, та розроблений план лікування. Встановлене використання класифікатора ІСРС 2 реформою

сімейної медицини призвело до мінімізації запису звернення пацієнта, що є суттєвим недоліком. По — перше замість звичного запису до амбулаторної карти пацієнта потрібно формувати зашифрований запис звернення в вигляді RXXRXXRXX(опис відібраного анамнезу пацієнта) та RXX встановлений діагноз. Також потрібно витратити час на пошук відповідних значень кодів в таблицях класифікатора ІСРС 2 при шифруванні запису і при розшифруванні того самого запису при наступному зверненні пацієнта або іншим лікарем з вторинної ланки. По — друге зі сторони збереження даних мінімізація запису згідно з ІСРС 2 дійсно є оптимізаційним кроком тому, що відбувається економія місця на сервері. Але зі сторони сімейного лікаря при шифруванні та подальшій мінімізації відбувається повна втрата детальності опису анамнезу пацієнта, що є суттєвим недоліком для сімейного лікаря та лікарів вторинної ланки тому, що при наступних зверненнях пацієнта лікарі не зможуть ознайомитися з попередніми зверненнями та призначеннями. Тому реалізація одночасного збереження детального опису звернення пацієнта та збереження зашифрованого за ІСРС 2 опису звернення пацієнта призведе до правильності дій при наступних зверненнях пацієнта або зверненнях до лікарів вторинної ланки та полегшення роботи з МІС.

Також було досліджено програмний продукт «Здоровое дыхание» розроблений компанією Nevlabs. Основною задачею даного рішення є прогнозування діагнозу респіраторного захворювання на основі відібраного анамнезу пацієнта. Коли пацієнт знаходиться на прийомі, лікар ставить йому запитання (задані в програмі) і фіксує відповіді. Після заповнення відповідей програма, використовуючи штучний інтелект, визначає найбільш ймовірний діагноз (він вважається основним). При бажанні лікар може подивитися всі варіанти діагнозів з вагами, які їм призначила нейронна мережа [15]. Проаналізувавши опис даного рішення на офіційному сайті розробника, було сформовано позитивні та негативні сторони програмного продукту. Позитивною стороною є використання технології віддаленого обчислення, тобто обробка анамнезу на стороні сервера. До негативної сторони відноситься:

Відсутність крос—платформовості, тобто неможливості запуску додатку на різних пристроях з під різних операційних систем, що не дає змоги збільшити цільову аудиторію користувачів та не приносить зручність у користуванні сімейним лікарям.

Відсутність автоматичного формування плану лікування. Сімейний лікар змушений сам в ручному режимі формувати план лікування на, що витрачається багато часу.

Відсутність динамічного оновлення бази знань. Дана область дослідження вимагає динамічного оновлення бази знань для алгоритмів машинного навчання тому, що можуть відбуватися зміни в клініці респіраторного захворювання і використовуючи динамічні бази знань можна досягти більш точного прогнозування діагнозу. Також використовуючи динамічні бази знань можна застосувати кластеризацію для поділу пацієнтів на групи та більш точного прогнозування діагнозу в окремих групах тому, що існують певні діагнози, які виникають тільки в окремих групах пацієнтів.

Відсутність можливості перенавчання моделі на ходу. При використанні динамічного оновлення бази знань обов'язково потрібно перенавчати модель машинного навчання для отримання надалі актуальних прогнозів згідно з доданими новими випадками захворювань.

Відсутність можливості автоматичного формування запису до амбулаторної карти. При наявному відібраному анамнезі (відповіді на запитання сімейного лікаря) не розроблена можливість автоматичного формування запису до амбулаторної карти, таким чином сімейний лікар змушений витратити час на запис інформації в амбулаторну карту пацієнта в ручному режимі.

Відсутність можливості роботи з класифікатором ІСРС 2. Використовуючи даний додаток сімейний лікар змушений витратити час на самостійне шифрування звернення пацієнта за класифікатором ІСРС 2.

Також було проведено пошук наукових робіт в даній галузі. Пошук дав наступні результати:

Наукова робота на тему «Розробка системи підтримки прийняття рішень для багатопрофільної медичної допомоги». В даній науковій роботі «Розглянуто принципи підтримки прийняття рішень в системі сімейної медицини. Досліджено потоки даних, які використовуються для діагностики та профілактики захворювань. Запропоновано класифікацію ознак на основі аналізу даних електронної медичної картки пацієнта. Розроблено модель ідентифікації медико—біологічних параметрів на основі даних електронної медичної картки. Розроблено системну архітектуру та структуру бази даних клінічного моніторингу.» [16]. Ознайомившись та проаналізувавши дану наукову роботу можна зробити висновки, що представлена модель функціонування системи підтримки прийняття рішення в основному фокусується на обробці всіх електронних медичних карт наявних на центральному сервері для подальшого проведення кластеризації пацієнтів з різними захворюваннями та виявленням відхилення від нормального стану об'єктів певного кластера. Таким чином можна допустити, що результат який запропонує система підтримки прийняття рішення має профілактичний характер і не зможе задовольнити потребу миттєвого прогнозу діагнозу згідно з відібраним анамнезом та миттєвого формування плану лікування. Тому дана модель частково вирішує проблематику оптимізації та пришвидшення обробки звернення пацієнта.

Проаналізувавши основні ключові моменти реформи сімейної медицини, основні структурні елементи функціонування ЦПМСД та сімейного лікаря, функціонування, організаційної та технічної структури. Також провівши спостереження разом з сімейним лікарем було виявлено та сформовано проблемні аспекти галузі сімейної медицини. Основними проблемними аспектами сімейної медицини зі сторони оптимізації та пришвидшення обробки звернення пацієнта є відсутність максимальної автоматизації обробки звернення пацієнта, в наслідок чого сімейний лікар витрачає на це дуже багато часу. Також було виявлено, що сімейний лікар виконує одну й ту саму роботу двічі, а саме проводить фіксацію звернення пацієнта за допомогою МІС в центральній базі даних з мінімізованим описом та проводить фіксацію звернення в амбулаторну карту пацієнта з детальним описом звернення.

Проаналізувавши рекомендовані до використання eHealth МІС було з'ясовано, що МІС не спроможні в повному обсязі надати автоматизацію обробки звернення пацієнта, а деякі з них взагалі ускладнюють процес роботи сімейного лікаря з МІС. Більшість МІС надають базовий функціонал затверджений eHealth, серед якого є:

- можливість роботи з електронними деклараціями між пацієнтами та сімейними лікарями;
- формування мінімізованих електронних медичних записів згідно з класифікатором ІСРС 2;
- виписування е—рецепту за програмою «Доступні ліки»;
- створення електронних направлень.

В даній ситуації при умовах, які склалися сімейний лікар змушений наряду з роботою в МІС проводити записи в ручному режимі на звичайному листі або фіксувати звернення в ручному режимі у звичайну амбулаторну карту пацієнта на, що витрачається у двічі більше часу. Серед програмних рішень, які можна було б використовувати в якості допоміжних було знайдено додаток «Здоровое дыхание» компанії Nevlabs. Але даний додаток пришвидшить тільки процес визначення захворювання. Процес автоматизації розробки та формування плану лікування та фіксацію детального звернення пацієнта розробник в даному додатку не передбачив. В науковому середовищі ситуація склалася наступна. Представлено модель функціонування системи підтримки прийняття рішень для обробки електронних медичних карт для кластеризації пацієнтів та подальшого встановлення груп ризику, що не повністю вирішує поставлену задачу. Таким чином проаналізувавши ситуацію, яка склалася можна зробити наступні висновки. У зв'язку з реформуванням сімейної медицини сімейні лікарі вимушені використовувати рекомендовані eHealth МІС та проводити на ряду з роботою в МІС роботу з паперовою документацією для забезпечення повного деталізованого опису звернення пацієнта та витратити на це вдвічі більше часу. Так як представлені рішення не в повному обсязі вирішують поставлену задачу можна вважати, що дана тематика розкрита частково. Таким чином виникає актуальність

проведення дослідження в галузі сімейної медицини та математичної медицини, у напрямку оптимізації обробки медичної інформації за допомогою алгоритмів машинного навчання, а саме оптимізації обробки анамнезу звернення, плану лікування та звернення пацієнта. В наступному підрозділі буде сформовано постановку задачі та завдання дослідження.

### 1.3 Постановка задачі та завдань дослідження

Дано галузь сімейної медицини, проведена в ній реформа, відсутність максимальної автоматизації робочого процесу, а саме обробки звернення пацієнта, наявність базового функціонала МІС який не в повному обсязі вирішує проблематику мінімізації часу обробки звернення пацієнта, наявність додаткових рішень, які частково вирішують проблематику. В наслідок чого виникнення труднощів при роботі з МІС та збільшення часу обробки, звернення пацієнта, сімейним лікарем.

Розробити оптимізаційне рішення, яке пришвидшить обробку звернення пацієнта з респіраторним захворюванням, надасть змогу сімейним лікарям за найменшу кількість дій, в автоматичному режимі, встановити діагноз згідно з відібраним анамнезом, сформувавши план лікування та створити детальний опис звернення пацієнта.

Таким чином згідно з поставленою задачею був розроблений наступний план завдань даного дослідження:

- ознайомитися з алгоритмами машинного навчання за допомогою яких можна вирішити задачу прогнозування діагнозу за відібраним анамнезом;
  - розробити формулу обрахунку загального часу обробки звернення пацієнта.
- Основний показник – витрачений час на обробку звернення пацієнта;

- розробити модель інтерпретації текстової інформації відібраного анамнезу звернення пацієнта в вхідний вектор анамнезу для подальшої обробки. Основний показник оцінки – витрачений час на інтерпретацію анамнезу в вхідний вектор;
- розробити моделі баз знань для зберігання дамів необхідної інформації для навчання моделі алгоритму машинного навчання. Основні показники оцінки — витрачений час на отримання зліпку дампу необхідної інформації для навчання моделі та витрачений простір для збереження баз знань;
- визначити оптимальний алгоритм машинного навчання, який задовольнить умову мінімізації часу обробки вхідного потоку даних (відібраний анамнез звернення пацієнта), прогнозування або класифікації діагнозу та перенавчання моделі. Основний показник оцінки – витрачений час на обробку вхідного потоку даних;
- розробити модель оптимізації, пришвидшення або зменшення часу, обробки звернення пацієнта;
- на основі розроблених моделей розробити web застосунок для практичної апробації. Основні показники оцінки – витрачений час на завантаження web застосунку з врахуванням платформи кінцевого користувача, об'єм завантажуваного web застосунку на кінцевий пристрій;
- провести тестування web застосунку для перевірки якості навчання алгоритму машинного навчання. Основний показник оцінки – точність прогнозованого або класифікованого діагнозу;
- провести заміри та фіксацію витраченого часу на обробку звернення пацієнта з використанням розробленого web застосунку. Основні показники оцінки – витрата часу на проведення етапів обробки звернення пацієнта, витрата часу на обробку запитів сервером та надсилання відповідей клієнту, витрата часу на автоматизоване формування плану лікування та фіксації звернення пацієнта;
- провести аналіз ефективності розробленого оптимізаційного рішення. Основний показник оцінки – зменшення часу обробки звернення пацієнта, виражений у відсотках;
- зробити висновки про проведення наукового дослідження.

Сформувавши задачу та завдання дослідження можна сформувати гіпотезу даного наукового дослідження. Гіпотеза дослідження наступна: використання алгоритмів машинного навчання та інформаційних технологій зменшить витрату часу на обробку звернення пацієнта з респіраторним захворюванням, розробку плану лікування та фіксацію звернення пацієнта до сімейного лікаря в амбулаторну карту пацієнта. В цьому підрозділі було сформовано задачу та завдання дослідження. Виконавши поставленні завдання, розробивши модель оптимізації обробки звернення пацієнта та розробивши програмне рішення за моделлю можна зменшити час обробки звернення пацієнта, структурувати та шаблонізувати процес звернення пацієнтів та роботи з ними, в наслідок чого зменшиться загальний час обробки звернення пацієнта та пришвидшиться просування черги пацієнтів. Таким чином сімейний лікар зможе за робочий день надати медичну допомогу більшій кількості пацієнтів.

В даному розділі було проведено ознайомлення з предметною областю загальної медичної практики, основними структурними елементами функціонування ЦПМСД та сімейного лікаря також проведено аналіз функціональної, організаційної та технічної структури. Досліджено основні ключові моменти реформи сімейної медицини. Оцінено реальний стан справ на момент проведення даного наукового дослідження. Визначено загальний алгоритм дій сімейного лікаря при роботі з пацієнтом та обробці анамнезу звернення пацієнта, визначено час проведення кожного етапу та в цілому звернення пацієнта. Виявлено основні недоліки запропонованих eHealth MIS та сформовано загальну картину стану справ в галузі сімейної медицини на момент проведення дослідження. На основі вище описаного доведено актуальність проведення даного наукового дослідження. Сформовано задачу та завдання дослідження та висунуто гіпотезу наукового дослідження.

В наступному розділі буде проведено огляд та ознайомлення з інструментарієм виконання поставленої задачі. Опис алгоритмів для виконання задачі. Аналіз відібраного інструментарію для виконання задачі, та обґрунтування вибору.

## 2 ВИБІР ТА ОБГРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАРІЮ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

### 2.1 Програмний комплекс та мова програмування

В даному розділі представлено огляд інструментарію для вирішення поставленої задачі. Також проводиться аналіз та обґрунтування використання обраного інструментарію, підходів та методів для вирішення поставленої задачі. Для розробки програмної реалізації оптимізаційного рішення було використано наступні інструменти, методи та підходи: Apache, PHP, PHP – ML, HTML, CSS, Material design, jQuery, AJAX, MVC, SPA, База знань типу csv, Структуризований об'єкт JSON. Також для виконання поставленого завдання в дослідженні було використано наступний математичний апарат для побудови алгоритмів машинного навчання: алгоритм наївного Баєса, метод опорних векторів, метод  $k$  – ближніх сусідів. Нижче проведено детальне ознайомлення з кожною методикою використаною в даному науковому дослідженні та обґрунтування використання даної методики.

Apache — це відкрите програмне забезпечення для створення веб сервера (HTTP—сервера). Його головна функція — швидка та надійна доставка контенту в мережі інтернет. Веб сервер приймає запити від клієнтів через веб переглядач за протоколом HTTP / HTTPS. У відповідь Apache відправляє браузеру контент у вигляді статичних HTML сторінок. Основні переваги викладено нижче.

Доступність. Це програмне забезпечення з відкритим вихідним кодом. Значить, його може безкоштовно використовувати або модифікувати будь—який охочий. Розробники по всьому світу створюють конфігурації та модулі веб—сервера для своїх специфічних потреб. З цієї ж причини Apache регулярно отримує корисні додатки, що розширюють його базовий функціонал.

Зручність і гнучкість налаштування. Додаток легко налаштовується через текстові конфігураційні файли. Apache здатний обробляти великий обсяг трафіку і швидко масштабуватися, навіть без складного додаткового конфігурування.

Функціональність. У Apache динамічна модульна структура. Можна швидко підключати додатковий функціонал у вигляді завантажуваних модулів, навіть без звернення до зовнішніх джерел. Це дозволяє вирішувати цілий комплекс найважливіших завдань в галузі безпеки, кешування, редагування URL, розподілу навантаження. Завдяки гібридним модулям MPM, Apache може однаково успішно обслуговувати статичний і динамічний контент. Є можливість оперативно відключати непотрібні модулі та прискорювати роботу веб—сервера

Крос—платформовість. Сервер працює як на Windows і MacOS, так і на всіх Unix—подібних системах. Система адміністрування не має серйозних відмінностей на різних ОС.

Сумісність. Apache працює на базі скриптових або веб—орієнтованих мов програмування (PHP, Python, Tcl, Ruby, Perl, ASP), що робить його сумісним з найбільшим спектром баз даних і серверного ПЗ. Багато веб—додатків та інструментів відразу виходять із засобами запуску з—під Apache у вигляді PHP—модуля. Веб—сервер, підтримує технології FastCGI та CGI, що дозволяють користуватися програмними продуктами на об'єктно—орієнтованих мовах Java, sh, C, C ++.

Масштабованість. Підходить для веб—ресурсів будь—якого масштабу. Apache добре працює як на односторінковому сайті, так і на багатосторінковому сайті з щоденною аудиторією в десятки тисяч відвідувачів.

Підтримка користувачів. Apache утримує першість популярності серед веб—серверів з 1996 року. За минулий час для нього створена широка база документації — як офіційної, так і створеної сторонніми розробниками.

Основними недоліками є:

Трафік впливає на продуктивність. На сайтах з великою відвідуваністю продуктивність веб—сервера може значно знижуватися, що уповільнює роботу

самого ресурсу. Це пов'язано з тим, що робота Apache заснована на моделі «процесів» і обробляє кожен запит користувача окремо.

Складна конфігурація підвищує вразливість. Можливість підключати модулі в Apache це не завжди перевага. Чим більше модулів, тим складніше стають налаштування. Відповідно, більше шансів допустити помилки в контурі безпеки.

Незручне редагування. В операційних системах сімейства Unix / Linux конфігураційні файли Apache доводиться редагувати вручну. Це пов'язано з відсутністю вбудованого графічного інтерфейсу для налаштування. Рішення проблеми — безкоштовний інструмент Apache GUI, що дозволяє налаштовувати функції веб—сервера прямо з браузера.

Зайвий функціонал. Навіть без додаткових модулів Apache надає користувачам масу можливостей. Правда, більшість використовує лише невелику частину базового функціоналу програми. Тому часто після установки доводиться витратити час на відключення «зайвих» модулів.

Apache складається з ядра і динамічної модульної системи. Параметри системи змінюються за допомогою конфігураційних файлів. Для запуску на сервері декількох веб—проектів одночасно використовується механізм віртуальних хостів. Ядро Apache розроблено Apache Software Foundation на мові C. Основні функції ядра — обробка конфігураційних файлів, протокол HTTP / HTTPS і завантаження модулів. Ядро може працювати без модулів, але буде мати обмежений функціонал. Модуль — окремий файл, підключення якого розширює основний функціонал ядра. Модулі можуть включатися до складу сервера при першій інсталяції або завантажуватися пізніше через зміну конфігураційного файлу. Даний тип сервера було обрано тому, що він є крос — платформовим, гнучким в налаштуваннях та в більшості компаній хостерів він використовується як основний тип сервера [17].

RНР — це поширена мова програмування загального призначення з відкритим вихідним кодом. RНР спеціально сконструйований для веб—розробок та його код може інтегруватися безпосередньо в HTML. RНР відрізняється від JavaScript тим, що RНР—скрипти виконуються на сервері і генерують HTML, який надсилається клієнту. RНР надсилає тільки відповідь клієнту, код скрипта клієнт

прочитати не в змозі тому, що скрипт виконується на сервері. Присутня можливість налаштування сервера таким чином, щоб звичайні HTML—файли оброблялися процесором PHP, так що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML—файл або результат виконання скрипта [18]. Використовується даний інструмент для розробки бекенд ( серверна частина) логіки тому, що основною його перевагою є безпека і легкість використання. Скрипти написані на PHP виконуються на сервері таким чином їх зміст зацікавлена особа прочитати не в змозі тому зменшується ризик взлому розробленого рішення на PHP.

PHP ML — фреймворк машинного навчання розроблений на PHP. Використовуючи даний фреймворк можна з легкістю інтегрувати алгоритми машинного навчання на серверну частину надавши можливість користувачу передавати вхідну інформацію анамнезу звернення пацієнта та отримувати оброблену вихідну інформацію в вигляді прогнозованого діагнозу, сформованого плану лікування. За рахунок виконання алгоритмів машинного навчання на стороні сервера сторона користувача може використовувати пристрої з невеликим обчислювальним ресурсом та працювати з реалізованим оптимізаційним рішенням. Фреймворком передбачено збереження моделей навчання та їх вагових коефіцієнтів, тому зі сторони розробника потрібно реалізувати можливість збереження вхідних даних, а модель обрахунку вхідних даних при кожному перенавчанні зберігається в спеціалізовані файлові контейнери. Виконуючи функцію прогнозування алгоритм отримує зліпок попередньої моделі та вагових коефіцієнтів та перевіряє актуальність зліпку, якщо зліпок не актуальний проводиться перенавчання на «ходу» та відбувається прогноз. Також даний фреймворк вимагає чіткої структуризації інформації яку він опрацьовує.

HTML (Hyper Text Markup Language) — це мова гіпертекстової розмітки сторінки. Використовується для того, щоб встановити браузеру правила згідно з якими потрібно відображати завантажену сторінку. На даний момент актуальна версія HTML5. HTML5 визначає новий алгоритм парсингу для створення структури DOM. Додано нові елементи і теги video, audio і ряд інших та перевизначені правила і семантика вже існуючих елементів HTML. Після реалізації

нових функцій HTML5 став не просто новою версією мови розмітки для створення веб—сторінок, але і фактично платформою для створення додатків, а область його використання вийшла далеко за межі веб—середовища інтернет: HTML5 застосовується також для створення мобільних додатків під Android, iOS, Windows Mobile і навіть для створення десктопних додатків для звичайних комп'ютерів. На даний момент більшість нових версій web браузерів підтримують стандарт HTML5 [19]. В даній науковій роботі HTML5 використовується для створення каркасу web сторінки для реалізації оптимізаційного рішення.

## 2.2 Методи та підходи представлення даних

CSS (Cascading Stylesheets) — правила, які використовуються для стилізації сторінки. При отриманні відповіді зі сервера web браузер користувача аналізує CSS правила та стилізує відображення HTML каркасу сторінки згідно з правилами вказаними в CSS ресурсі. Правило CSS складається з наступних основних елементів:

- селектор. Назва елемента HTML на початку правила. Селектор вибирає елемент чи елементи, які будуть стилізовані;
- визначення (декларації). Одне або декілька правил на зразок «color: red;» вказує, яку властивість елемента потрібно стилізувати;
- псевдокласи. Ключові слова, які додаються до селектора для визначення його стану. Псевдокласи дають можливість стилізувати елемент на основі не тільки на основі DOM — дерева, але керуючись зовнішніми факторами, такими як історія відвідувань (наприклад, visited), стан вмісту (на кшталт: checked у деяких елементів форми) або позиції курсора миші (наприклад, : hover визначає, чи знаходиться курсор миші над елементом).

Даний стандарт використовувався для стилізації HTML каркасу на пристрої кінцевого користувача.

Material design — це мова візуальних образів, яка створена корпорацією Google для уніфікації інтерфейсів всіх її продуктів і сервісів. Всі елементи даного напрямку в дизайні, постійно оновлюються та доповнюються, але зберігають основи незмінними. Material design базується на тактильній реальності паперу і чорнила. Material design базується на наступних основних принципах:

- тактильні поверхні. Всі елементи інтерфейсу — це шари цифрового паперу. Вони розташовуються на різній висоті і відкидають тіні;
- поліграфічний дизайн. На цифровому папері потрібно писати цифровим чорнилом. Все, що зображено і написано на шарах—елементах, підпорядковується законам друкованого дизайну;
- анімація. Всі елементи, які є на екрані, не можуть просто так з'являтися і зникати, адже в реальному житті так не буває;
- адаптивний дизайн. Все це повинно працювати однаково на будь—яких пристроях.

Основна ідея анімації в Material Design – зробити інтерфейс користувача виразним і простим у використанні. Для цього анімація повинна відповідати трьом принципам. Інформативність — анімація показує просторові і ієрархічні зв'язки між елементами. Орієнтованість — анімація фокусує увагу на тому, що важливо, і не відволікає від основних дій. Виразність — анімація висловлює характер, індивідуальність і стиль кожного продукту. Також немаловажним аспектом є визначення ієрархії для забезпечення концентрування уваги користувача на потрібній інформації. Таким чином даний підхід проектування дизайну web застосунку використовувався в першу чергу для забезпечення максимального концентрування уваги сімейного лікаря на основній потрібній інформації. Використання філософії даного інтерфейсу забезпечило можливість відображати web застосунок однакового на всіх пристроях з адаптацією під них.

jQuery – багатофункціональна кросбраузерна бібліотека JavaScript невеликих розмірів. Приносить полегшення та оптимізацію в маніпулюванні документами HTML, обробці подій, створює новий підхід для реалізації анімації на web сторінці.

Створює полегшений підхід формування AJAX запитів та отримання відповіді від сервера завдяки простому у використанні API, який працює у багатьох браузерях.

AJAX – це аббревіатура Asynchronous Javascript and XML, що означає асинхронний Javascript та XML. При використанні AJAX оновлення сторінки, для відправки запиту на сервер та отримання відповіді з нього, не потрібне. Тому, що оновлюється лише той модуль в якому потрібно оновити інформацію. Дана методика надає змогу зменшити об'єм завантаження відповіді з сервера, таким чином відбувається економія трафіку та пришвидшення отримання відповіді з сервера. Пришвидження відбувається за рахунок разового завантаження основної структури сторінки і багаторазової відправки запиту та отримання відповіді з сервера в вигляді об'єкту JSON (в більшості випадків) або XML. AJAX використовує асинхронну передачу даних. Поки відбувається передача даних, користувач може здійснювати інші, необхідні йому дії. Під час «спілкування» з сервером потрібно сповістити користувача про те, що виконується обмін інформацією з сервером, в іншому випадку користувач може виконувати багаторазово одні й ті самі запити в результаті чого він не отримає відповідь або сервер відхилить можливість обробки запитів даного користувача з міркувань логіки безпеки функціонування сервера та web додатку. Основними перевагами використання AJAX є:

- створення зручного інтерфейсу web додатку;
- активне, динамічне «спілкування з сервером»;
- уникнення повного перезавантаження сторінки;
- зручність використання у роботі;
- зменшення використання трафіку.

jQuery та методика AJAX використовується для забезпечення функціонування моделі web додатку типу SPA, пришвидшення роботи клієнта з сервером, зменшення навантаження на сервер за рахунок відправки та отримання об'єкта JSON, оптимізації додатку при завантаженні на мобільних платформах, динамічного оновлення інформації на сторінці без перезавантаження сторінки.

MVC (model-view-controller) – модель – вигляд – контролер. Підхід, який використовує блоки, що відповідають за вирішення різних завдань. Один блок відповідає за дані додатка, інший відповідає за зовнішній вигляд, а третій контролює роботу додатка. Основними блоками є:

- модель — це компонент, який відповідає за дані, а також визначає структуру програми;
- вигляд — це компонент, який відповідає за взаємодію з користувачем. Тобто код компонента `view` визначає зовнішній вигляд програми і способи його використання;
- контролер — цей компонент відповідає за зв'язок між `model` і `view`. Код компонента `controller` визначає, як сайт реагує на дії користувача. Це ядро MVC-додатку.

Використовується для оптимізації виконання клієнтської та серверної частини web додатку методом поділу на модулі моделі, вигляду та контролера. Використання MVC надасть можливість забезпечити функціонування web додатку типу SPA та створить зручність написання програмного коду та подальший його рефакторинг.

SPA (Single Page Application) - тип web-додатків, в яких завантаження та виконання необхідних модулів функціонування відбувається на одній сторінці. Це дозволяє зменшити час повторного завантаження одних і тих самих модулів. Особливість архітектури SPA полягає в тому, що всі елементи, необхідні для роботи web додатку знаходяться на одній сторінці. Вони завантажуються при ініціалізації. Також даний вид додатків завантажує додаткові модулі після запиту від користувача. Вся маршрутизація проходить фіксацію активності користувача для подальшої зручної взаємодії користувача з web додатком. При завантаженні нових модулів в SPA контент на них оновлюється тільки частково, тому, що немає необхідності повторно завантажувати незмінні елементи. Це збільшує швидкість відповіді та зменшує обсяг відправленої та отриманої інформації між користувачем та сервером. Даний вид web додатку за способом взаємодії з користувачем

найбільше схожий на роботу прикладних додатків, але на сервері. Переваги використання SPA:

- доступність. Можна отримати миттєвий доступ до функціонала з будь-якого типу пристрою без проблем з сумісністю та не зважаючи на обчислювальний ресурс пристрою;
- універсальність. Використовувати web додаток можна практично з будь-якого пристрою, якщо на ньому є доступ до інтернету. Якщо при розробці інтерфейсу враховувалися адаптивність, то використовувати SPA однаково зручно з різних пристроїв;
- можливість використовувати великі обсяги даних. Розмір програми та об'єм даних «спілкування» з сервером не обмежений обчислювальним ресурсом пристрою;
- швидкість. Одна сторінка зі всіма необхідними елементами та модулями не тільки економить час на повторне завантаження даних, а й підвищує продуктивність роботи;
- можливості розробки. Розробникам доступні фреймворки, які спрощують створення архітектури проєкту і надають чимало готових елементів для роботи.

Недоліками використання SPA є:

- необхідність інтернет-з'єднання. Без доступу до мережі використовувати web додаток неможливо;
- труднощі з SEO. Особливості SPA ускладнюють або унеможливають процес індексації пошуковими системами всіх модулів програми. Це може викликати труднощі з оптимізацією;
- відсутність підтримки виконання JS скриптів на стороні клієнта. Багато користувачів відключають відображення JS-елементів у себе в браузерах, а Single Page Application використовує їх в роботі, тому в даному випадку web додаток може не працювати [20].

Обрано даний тип web додатку тому, що загальна концепція роботи користувача з SPA додатком не відрізняється від роботи з прикладним додатком. Також користувач не змушений завантажувати додаток на пристрій та займатися

його встановлення та налаштування. Сімейному лікарю лише потрібно перейти за посиланням на веб-сторінку в мережі для того, щоб розпочати свою роботу. Підхід до розробки SPA додатків містить в собі адаптивність сторінок, тому розробленим додатком можна користуватися на пристроях з різною роздільною здатністю.

### 2.3 Бази знань

Інформаційне сховище типу csv (comma-separated values) – підхід до збереження інформації розділеної за допомогою спецсимволу. Найчастіше в якості роздільника використовується символ «кома». Даний формат дозволяє зберігати інформацію без розмітки для представлення, таким чином відбувається мінімізація об'єму файлу в якому збережена інформація. Також інформацію з легкістю можна представити в табличному вигляді. Підлягає швидкому індексуванню завдяки можливості інтерпретації в матричний вигляд або багатомірний масив. Також не потребує спеціалізованого програмного забезпечення для редагування. З легкістю можна зберігати бази знань та моделі з ваговими коефіцієнтами для алгоритмів машинного навчання. В даному дослідженні використовується для збереження бази знань векторів клініки та анамнезу респіраторних захворювань. Даний підхід було обрано через відносно невеликий час отримання зліпку бази знань на опрацювання, в порівнянні з СКБД MySQL, Maria DB, PostgreSQL. При використанні СКБД відбувається велика витрата часу на відправку запиту його обробку та отримання відповіді. Також використовуючи даний метод збереження інформації можна швидко та з легкістю отримати значення потрібного елемента звернувшись до нього за індексом та на «льоту» змінювати вміст бази знань шляхом додавання нових векторів клініки захворювань та анамнезу пацієнтів. При потребі даний тип організації збереження інформації дозволяє з легкістю, без руйнування, модифікувати архітектуру бази знань. Тому підхід до реалізації бази знань згідно з стандартом csv є оптимальним варіантом в даному випадку.

JSON (JavaScript Object Notation) - простий формат обміну даними, зручний для читання і написання як людиною, так і комп'ютером. Він заснований на підмножині мови програмування JavaScript, визначеного в стандарті ECMA-262 3rd Edition - December 1999. JSON - текстовий формат, повністю незалежний від мови реалізації, але він використовує підходи, знайомі програмістам С-подібних мов, таких як С, С ++, С # , Java, JavaScript, Perl, Python і багатьох інших. Ці властивості роблять JSON ідеальною мовою для обміну даними. JSON заснований на двох структурах даних:

- колекція пар ключ / значення. У різних мовах, ця концепція реалізована як об'єкт, запис, структура, словник, хеш, іменований список або асоціативний масив;
- упорядкований список значень. У більшості мов це реалізовано як масив, вектор, список або послідовність.

Це універсальні структури даних. Майже всі сучасні мови програмування підтримують їх в будь-якій формі. Тому цей формат даних, незалежний від мови програмування, повинен бути заснований на цих структурах. В записі формат JSON виглядає наступним чином:

- об'єкт - невпорядкований набір пар ключ / значення. Об'єкт починається з «{» відкритої фігурної дужки та закінчується «}» закритою фігурною дужкою. Кожне ім'я супроводжується: двокрапкою, пари ключ / значення, значення розділяються, спецсимволом «кома»;
- масив - впорядкована колекція значень. Масив починається з «[» відкритої квадратної дужки та закінчується «]» закритою квадратною дужкою. значення розділяються, спецсимволом «кома»;
- величина значення може бути рядком в подвійних лапках, числом, true, false, null, об'єктом або масивом. Також дана підструктура підтримує вкладення;
- рядок – колекція з нуля або більше символів Unicode, укладена в подвійні лапки, використовуючи \ зворотній слеш в якості символу екранування. Символ представляється як односимвольний рядок. Схожий синтаксис використовується в С і Java;

- число представляється так само, як в С або Java. Для представлення використовується десяткова система;
- пробіли можуть використовуватися між будь-якими лексемами [21].

Плани лікування та шифри ICPC2, МКХ – 10 потребують організації складної структури даних, тому для реалізації даної потреби було обрано формат JSON. Використовуючи даний формат можна сформувавши складну модель збереження та представлення даних та інтерпретувати дані в об'єкти, що допоможе швидко та легко отримати доступ до об'єкта даних та отримати значення обраного об'єкта згідно з заданими параметрами запиту.

## 2.4 Машинне навчання та алгоритми класифікації

Для виконання поставлених завдань даного наукового дослідження потрібно використовувати алгоритм машинного навчання. За допомогою машинного навчання можливо буде реалізувати класифікацію та прогнозування діагнозу та подальше формування плану лікування, що значно пришвидшить час обробки звернення пацієнта. Для визначення оптимального алгоритму машинного навчання було обрано наступні алгоритми: алгоритм наївного Баєса, метод опорних векторів, метод k – ближніх сусідів. Нижче буде розглянуто загальні принципи машинного навчання та принципи роботи вище згаданих алгоритмів.

Машинне навчання – набір алгоритмів, які здатні проаналізувати набір вхідних даних без необхідності писати специфічний для даної проблеми код. Замість написання коду відбувається передача даних в загальний алгоритм, і він будує свою власну логіку на їх основі. Машинне навчання поділяється на наступні типи алгоритмів:

- машинне навчання з вчителем (supervised learning);
- машинне навчання без вчителя (unsupervised learning);

- машинне навчання з частковим використанням вчителя (semi – supervised learning);
- машинне навчання з підкріпленням (reinforcement learning).

Машинне навчання з вчителем (supervised learning) тип алгоритмів, які навчаються на розміненій вибірці даних. Даний підхід вимагає наявності повного набору розмічених даних для тренування моделі на всіх етапах її розробки. Наявність повного розміненого датасету означає, що кожному прикладу в навчальному наборі є відповідна відповідь, яку алгоритм має отримати. Таким чином розмінений датасет який складається з інформації про клініку респіраторного захворювання та анамнезу пацієнтів «навчить» алгоритм розпізнавати те чи інше респіраторне захворювання. При подачі нової інформації алгоритм зрівняє її з наявною в базі знань проведе прогноз відповіді. В більшості випадків машинне навчання з вчителем використовують для класифікації і регресії. В задачах класифікації алгоритм прогнозує дискретні значення, які відповідають номерам класів до яких належить той чи інший об'єкт. В датасеті (базі знань) з інформацією про клініку захворювання кожний запис буде мати відповідну мітку з діагнозом (напр. синусит чи тонзиліт). Ефективність роботи алгоритму оцінюється в правильності класифікації вхідної інформації відповідно до бази знань. Задачі регресії аналізують неперервні значення. Наприклад лінійна регресія обраховує значення змінної «у» враховуючи значення «х». Алгоритми машинного навчання з вчителем більш дієві якщо існує датасет великих об'ємів в якому детально описана інформація про об'єкти аналізу [22].

Машинне навчання без вчителя підхід до реалізації алгоритмів для відшукування невідомих результуючих значень. Реалізація даного алгоритму передбачає наявність нерозміченого датасету в якому немає явних результатів відповідно до описаних об'єктів. Таким чином задача алгоритму машинного навчання без вчителя відшукати значення та провести прогнозування вхідної інформації згідно з побудованою моделлю. При побудові моделі алгоритм самостійно відшукує кореляції між інформацією в датасеті та формує залежності та властивості. Даний алгоритм використовується для задач: кластеризації,

відшукування аномалій в датасетах, утворенні асоціативних списків та асоціацій. Ефективність таких алгоритмів обрахувати важко тому, що відсутня розмітка в датасеті для зрівняння правильності прогнозованого значення [22].

Машинне навчання з частковим використанням вчителя (semi – supervised learning) використовує частково розмічений датасет. Використовується при важкості відшукування кореляції, важливих даних та взаємозв'язків. Таким чином часткова розмітка в даному підході слугує корегувальним елементом для побудови моделі. Даний метод набув популярності в галузі обробки біомедичних зображень. Ефективність алгоритму вимірюється в точності виконаного результату.

Машинне навчання з підкріпленням (reinforcement learning) ітераційно – генеративний процес при виконанні якого алгоритм віднаходить оптимальний шлях вирішення поставленої задачі. Датасет в такому випадку будується самим алгоритмом і з кожною ітерацією алгоритм перенавчається для отримання актуальної моделі. Розмітка датасету відбувається самим алгоритмом в більшості випадків результуюча відмітка приймає значення типу вірно або хибно. Найчастіше використовується в іграх при навчанні поведінки персонажів або формуванні трафіку автомобілів. Також використовується для створення прогнозування поведінки різних об'єктів. При реалізації даного алгоритму немаловажну роль відіграє кількість ітерацій навчання або генерацій популяції. Чим більша кількість ітерацій, тим чіткіше буде виконаний результат. Ефективність роботи таких алгоритмів визначається точністю виконаного результату та швидкістю проведення навчання [22].

Алгоритм наївного Баєса - алгоритм класифікації, заснований на теоремі Байєса з припущенням про незалежність ознак. НБА передбачає, що наявність будь якої ознаки в класі не пов'язано з наявністю будь якої іншої ознаки [23]. Наприклад, захворювання синусит може вважатися синуситом, якщо вектор анамнезу містить в собі значення «гарячка» та «порушення носового дихання». Навіть якщо ці ознаки залежать один від одного або від інших ознак, в будь-якому випадку вони вносять незалежний внесок у ймовірність того, що цей діагноз є синусит. У зв'язку з таким припущенням алгоритм називається «наївним» [23].

Моделі на основі НБА досить прості і вкрай корисні при роботі з дуже великими наборами даних. При своїй простоті НБА потужніший за деякі складні алгоритми класифікації. Теорема Баєса дозволяє розрахувати апостеріорну ймовірність  $P(c|x)$  на основі  $P(c), P(x)$  та  $P(x|c)$  (2.1).

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (2.1)$$

де,  $P(c|x)$  апостеріорна ймовірність даного класу  $c$  (тобто даного значення цільової змінної) при даному значенні ознаки  $x$ ,

$P(c)$  апіорна ймовірність даного класу,

$P(x|c)$  правдоподібність, тобто ймовірність даного значення властивості даного класу,

$P(x)$  апіорна ймовірність даного значення властивості.

Нижче представлено алгоритм та спрощену модель роботи найвішого байєсовського алгоритму. Першим кроком є перетворення набору даних в частотну матрицю. Вхідні дані наведено у таблиці 2.1, частотна матриця представлена у таблиці 2.2.

Таблиця 2.1 – Тестова база знань виникнення захворювання синусит.

Анамнез	Фіксація виникнення
Головний біль	Не виникло
Гарячка	Не виникло
Головний біль	Не виникло
Головний біль	Не виникло
Гарячка	Виникло
Порушення нюху	Виникло
Головний біль	Виникло
Головний біль	Виникло
Порушення нюху	Виникло

Продовження таблиці 2.1

Порушення носового дихання	Виникло
Порушення нюху	Виникло
Порушення носового дихання	Виникло
Порушення нюху	Не виникло
Порушення носового дихання	Не виникло
Порушення носового дихання	Виникло

Таблиця 2.2 – Частотна таблиця виникнення захворювання синусит.

Анамнез	Виникнення	Не виникнення
Головний біль	2	3
Гарячка	1	1
Порушення нюху	3	1
Порушення носового дихання	3	1

Наступним кроком є створення таблиці правдоподібності. Створюється така таблиця за допомогою розрахунку відповідних ймовірностей. Результати представлені у таблиці 2.3.

Таблиця 2.3 – Розрахункова таблиця апостеріорної ймовірності виникнення захворювання синусит згідно з відібраним анамнезом.

Анамнез	Виникнення	Не виникнення	Розрахунок	Результат
Головний біль	2	3	5/15	0,33
Гарячка	1	1	2/15	0,13
Порушення нюху	3	1	4/15	0,26
Порушення носового дихання	3	1	4/15	0,26
Всього	9	6		
Розрахунок	9/15	6/15		
Результат	0,6	0,4		

Скориставшись теоремою Баєса можна провести розрахунок апостеріорної ймовірності для кожного класу при умовах які склалися. Відповідно клас з найбільшою апостеріорною ймовірністю буде прогнозом. Тому проаналізувавши таблицю 2.3 можна зробити висновки, що на виникнення захворювання синусит найбільше впливає властивість анамнезу порушення нюху та порушення носового дихання, а найменше впливає гарячка. Таким чином використовуючи вище описану методику як базовий функціонал, можна розробити архітектуру функціонування алгоритму машинного навчання для класифікації вхідного анамнезу та прогнозування діагнозу, попередньо створивши базу знань, яка містить в собі клініку захворювань та анамнез попередніх звернень пацієнтів. Даний алгоритм дає змогу виділити фактори, які найбільше або найменше впливають на формування захворювання. Цією властивістю можна також скористатися при проведенні подальшого аналізу даних для виявлення нових невідомих залежностей та отримання корисних результатів аналізу. Також на основі НБА можна проводити прогнозування декількох різних класів за допомогою множинних властивостей. Даний алгоритм машинного навчання в основному використовується для мультикласифікації.

Позитивні сторони НБА:

- класифікація, в тому числі мультикласова, виконується легко і швидко;
- коли припущення про незалежність виконується, НБА перевершує інші алгоритми, такі як логістична регресія (logistic regression), і при цьому вимагає менший обсяг навчальних даних;
- НБА краще працює з ознаками типу категорії, ніж з безперервними. Для безперервних ознак передбачається нормальний розподіл, що є досить сильним допущенням.

Негативні сторони НБА:

- якщо в тестовому наборі даних є певне значення категорійної ознаки, яке не зустрічалось в навчальному наборі даних, тоді модель присвоїть нульову ймовірність цьому значенню відповідно і не зможе зробити прогноз. Це явище відоме під назвою «нульова частота» (zero frequency). Дану проблему можна

вирішити за допомогою згладжування. Одним з найпростіших методів є згладжування по Лапласа (Laplace smoothing);

- хоча НБА є хорошим класифікатором, значення прогнозованих ймовірностей не завжди є достатньо точними;
- ще одним обмеженням НБА є припущення про незалежність ознак. В реальності набори повністю незалежних ознак зустрічаються рідко [23].

Таким чином використовуючи даний алгоритм класифікації можна з легкістю проводити мультикласову класифікацію та класифікацію на «льоту», що дозволяє використовувати даний алгоритм на сервері з невеликим обчислювальним ресурсом і в потрібний момент для користувача проводити перенавчання моделі та прогнозування діагнозу.

Метод опорних векторів – алгоритм логіка якого полягає в відшуванні гіперплощини в  $n$  – вимірному просторі для розділення об’єктів навчальної вибірки представлених в вигляді точок на два класи та побудові апроксимації вхідного вектору для відшування приналежності до першого або другого класу. Таким чином виникає задача бінарної класифікації в, якій об’єктам з  $X = R^n$  відповідає один з двох класів  $Y = \{-1, +1\}$ . Тому, є заданий розмічений датасет  $T^l = (\vec{x}_i, y_i)_{i=1}^l$ , потрібно створити алгоритм класифікації  $\alpha(\vec{x}): X \rightarrow Y$ .

Гіперплощина для поділу. В просторі  $R^n$  рівняння  $\langle \vec{w}, \vec{x} \rangle - b = 0$  при заданих  $\vec{w}$  та  $b$  визначає гіперплощину множини векторів  $\vec{x} = (x_1, \dots, x_n)$  які належать до простору меншої розмірності  $R^{n-1}$ . Наприклад для  $R^1$  гіперплощиною є точка, для  $R^2$  пряма, для  $R^3$  площина. Параметр  $\vec{w}$  визначає вектор нормалі для гіперплощини, а через  $\frac{b}{\|\vec{w}\|}$  виражається відстань від гіперплощини до початку координат. Гіперплощина розділяє на два півпростори  $\langle \vec{w}, \vec{x} \rangle - b > 0$  та  $\langle \vec{w}, \vec{x} \rangle - b < 0$ . Гіперплощина розділяє два класи  $C_1$  і  $C_2$ , якщо об’єкти цих класів лежать по різних сторонах від гіперплощини, то тоді виконуються наступні умови (2.2) та (2.3):

$$\begin{cases} \langle \vec{w}, \vec{x} \rangle - b > 0, & \forall x \in C_1 \\ \langle \vec{w}, \vec{x} \rangle - b < 0, & \forall x \in C_2 \end{cases} \quad (2.2)$$

або

$$\begin{cases} \langle \vec{w}, \vec{x} \rangle - b < 0, & \forall x \in C_1 \\ \langle \vec{w}, \vec{x} \rangle - b > 0, & \forall x \in C_2 \end{cases} \quad (2.3)$$

Лінійно роздільна вибірка. Нехай вибірка лінійно роздільна, тобто існує деяка гіперплощина, що розділяє класи -1 і +1. Тоді як алгоритм класифікації можна використовувати лінійний пороговий класифікатор (2.4):

$$\alpha(\vec{x}) = \text{sign}(\langle \vec{w}, \vec{x} \rangle - b) = \text{sign}\left(\sum_{i=1}^l w_i x_i - b\right) \quad (2.4)$$

де,  $\vec{x} = (x_1, \dots, x_n)$  – вектор значень властивостей об'єкта,

$\vec{w} = (w_1, \dots, w_n) \in R^n$  і  $b \in R$  – параметри гіперплощини.

Для двох лінійно роздільних класів можливі різні варіанти побудови гіперплощини. Метод опорних векторів вибирає ту гіперплощину, яка максимізує відступ між класами. Відступ - характеристика, яка оцінює, наскільки об'єкт "занурений" у свій клас, наскільки типовим представником класу він є. Чим менше значення відступу  $M_i$ , тим ближче об'єкт  $\vec{x}_i$  і підходить до границі класів і тим вище стає ймовірність помилки. Відступ  $M_i$  негативний тоді і тільки тоді, коли алгоритм  $\alpha(\vec{x})$  допускає помилку на об'єкті  $\vec{x}_i$ . Для лінійного класифікатора відступ визначається рівнянням (2.5):

$$M_i(\vec{w}, b) = y_i(\langle \vec{w}, \vec{x}_i \rangle - b) \quad (2.5)$$

Якщо вибірка лінійно роздільна, то існує така гіперплощина, відступ від якої до кожного об'єкта позитивний (2.6):

$$\exists \vec{w}, b: M_i(\vec{w}, b) = y_i(\langle \vec{w}, \vec{x}_i \rangle - b) > 0, i = 1 \dots l \quad (2.6)$$

Потрібно побудувати таку розділяючу гіперплощину, щоб об'єкти датасету перебували на найбільшій відстані від неї. Тому при множенні  $\vec{w}$  і  $b$  на константу  $c \neq 0$  рівняння (2.7) визначає ту ж саму гіперплощину, що і  $\langle \vec{w}, \vec{x} \rangle - b = 0$ .

$$\langle c\vec{w}, \vec{x} \rangle - cb = 0 \quad (2.7)$$

Для зручності при нормуванні: вибирається константа, щоб  $\min M_i(\vec{w}, b) = 1$ . При цьому в кожному з двох класів знайдеться хоча б один "граничний" об'єкт з датасету, відступ якого дорівнює цьому мінімуму: інакше можна було б змістити гіперплощину в сторону класу з великим відступом, тим самим збільшивши мінімальну відстань від гіперплощини до об'єктів датасету. Нормування дозволяє обмежити розділяючу смугу між класами:  $\{x: -1 < \langle \vec{w}, \vec{x}_i \rangle - b < 1\}$ . Всередині неї не може лежати жоден об'єкт з датасету. Ширину розділяючої смуги можна виразити як проекцію вектору  $\vec{x}_+ - \vec{x}_-$  на нормаль до гіперплощини  $\vec{w}$ . Щоб розділяюча гіперплощина знаходилася на найбільшій відстані від точок вибірки, ширина смуги повинна бути максимальною. Таким чином даний алгоритм приводить до постановки задачі оптимізації в межах квадратичного програмування (2.8).

$$\begin{cases} \|\vec{w}\|^2 \rightarrow \min_{w,b} \\ M_i(\vec{w}, b) \geq 1, i = 1, \dots, l \end{cases} \quad (2.8)$$

Лінійно не роздільна вибірка. В практиці лінійно роздільні вибірки не зустрічаються: в датасетах можливі викиди і нечіткі межі між класами. В такому випадку поставлене вище завдання не має рішень, і необхідно послабити обмеження, дозволивши деяким об'єктам потрапляти на "територію" іншого класу. Після послаблень обмеження розділяючої полоси поставлена задача зводиться до

задачі (2.9). Точність класифікації в даній умові визначається зміною обмежень, а самі обмеження визначаються емпіричним шляхом. Переваги методу опорних векторів:

- задача квадратичного програмування добре вивчена і має єдине рішення;
- метод опорних векторів еквівалентний двошаровій нейронній мережі, де число нейронів на прихованому шарі визначається автоматично як число опорних векторів;
- принцип оптимальної розділяючої гіперплощини призводить до максимізації ширини розділяючої смуги.

Недоліки методу опорних векторів:

- нестійкість до шуму: викиди в вихідних даних стають опорними об'єктами і безпосередньо впливають на побудову розділяючої гіперплощини;
- відсутність опису загальних методів побудови ядер і направляючих просторів, найбільш придатних для конкретного завдання;
- відсутність можливості мультикласифікації. Потреба у використанні моделей типу «Один проти всіх» або «Один проти одного» для мультикласифікації [24].

Використання даного алгоритму машинного навчання потребує практичної перевірки на реальному датасеті в реальних умовах для визначення ефективності відпрацювання алгоритму. Також для даного алгоритму потрібно використовувати модель типу «Один проти всіх» або «Один проти одного» для мультикласифікації, щоб виконати поставлену задачу прогнозування діагнозу, що може призвести до збільшення витрати обчислювального ресурсу хост – сервера хмарного обчислювального середовища, а відповідно і до зменшення кількості опрацювання запитів.

Метод  $k$  ближніх сусідів – бінарний алгоритм класифікації логіка, якого працює на визначення подібних властивостей  $k$  ближніх сусідів з класів датасету до вхідного вектору та віднесення вхідного вектору до подібного за властивостями класу. Нехай є  $m$  спостережень, кожному з яких відповідає запис в датасеті. Всі записи належать якомусь класу. Необхідно визначити клас для нового запису. На першому кроці алгоритму задається число  $k$  - кількість найближчих сусідів. На

другому кроці знаходяться  $k$  записів з мінімальною відстанню до вектору ознак нового об'єкта. Функція для розрахунку відстані повинна відповідати наступним правилам:  $d(x, y) \geq 0$ ,  $d(x, y) = 0$  тоді і тільки тоді коли  $x = y$ ,  $d(x, y) = d(y, x)$ ,  $d(x, z) \leq d(x, y) + d(y, z)$ , при умові, що точки  $x, y, z$  не лежать на одній прямій. Параметри  $x, y, z$  – вектори властивостей об'єктів порівняння. Для впорядкованих значень атрибутів знаходиться Евклідова відстань (2.9):

$$D_E = \sqrt{\sum_i^n (x_i - y_i)^2} \quad (2.9)$$

де  $n$  – кількість атрибутів.

Також інколи використовується нормалізація перед розрахунком відстані. мінімаксна нормалізація (2.10), нормалізація за допомогою стандартного відхилу (2.11)

$$X = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (2.10)$$

$$X = \frac{X - X_{\text{ср}}}{\sigma_x} \quad (2.11)$$

де  $\sigma_x$  – стандартний відхил,

$X_{\text{ср}}$  – середнє значення

На наступному кроці коли віднайдені всі відстані потрібно визначити вплив вхідного вектору на клас. Для цього використовується адитивна функція типу просте незважене голосування. Принцип дії даної функції наступний:

- задати число  $k$ ;
- визначити кількість векторів, які мають право голосу при визначенні класу;
- визначити  $k$  векторів в яких найменша відстань до вхідного вектору;
- кожний вектор голосує за клас до якого належить;

— вхідному вектору присвоюється клас який набрав найбільше голосів.

В випадку якщо декілька класів набрали однакову кількість голосів використовується підхід зваженого голосування. Даний підхід базується на формулі (2.12)

$$votes(class) = \sum_{i=1}^k \frac{1}{d^2(X, Y_i)} \quad (2.12)$$

де  $d^2(X, Y_i)$  – квадрат відстані від відомого вектору до вхідного вектору,

$n$  – кількість відомих векторів

Новий вектор класифікується згідно з найбільшою кількістю набраних голосів.

Перевагами використання даного алгоритму є:

- стійкість до аномальних викидів в датасеті;
- проста програмна реалізація;
- можливість модифікації алгоритму.

Основним недоліком є відсутність можливості відділення даних від алгоритму, тобто створення певної моделі проміжних коефіцієнтів для подальшого збереження та роботи з нею при нових класифікаціях [25].

В даному розділі було проведено ознайомлення та обґрунтування використання методів та підходів до виконання наукового дослідження. Тому, так як кінцевим результатом є оптимізаційне рішення для пришвидшення обробки звернення пацієнта створене за розробленою моделлю, потрібно розробити web додаток який буде працювати за розробленою моделлю. Для створення web додатку будуть використовуватися наступні технології та інструменти: Visual Code – для написання програмного коду, Apache – сервер обробки запитів користувача, PHP та PHP – ML серверна мова програмування та фреймворк для реалізації машинного навчання. HTML, CSS, Material design, jQuery, AJAX додатковий інструментарій для реалізації моделі MVC та SPA. База знань типу csv та JSON – інформаційні бази для збереження векторів клініки, анамнезу

захворювань та планів лікування пацієнтів. Також модель оптимізації вимагає використання машинного навчання тому на огляд та подальше визначення, емпіричним методом, оптимального алгоритму машинного навчання було обрано наступні алгоритми класифікації: алгоритм наївного Баеса, метод опорних векторів та метод  $k$  – ближніх сусідів.

В наступному розділі представлено процес вирішення поставленої задачі та завдань наукового дослідження, розробка моделі оптимізації та розробка оптимізаційного рішення для проведення практичної апробації та визначення ефективності моделі оптимізації.

### **3 РОЗРОБКА ОПТИМІЗАЦІЙНИХ МОДЕЛЕЙ ТА WEB ЗАСТОСУНКУ. ПРОВЕДЕННЯ ПРАКТИЧНОЇ АПРОБАЦІЇ ТА ТЕСТУВАННЯ. АНАЛІЗ ЕФЕКТИВНОСТІ РОЗРОБЛЕНИХ МОДЕЛЕЙ ТА WEB ЗАСТОСУНКУ**

3.1 Розробка моделі оптимізації, моделі архітектури функціонування хмарного середовища та допоміжних моделей

В даному розділі представлено процес вирішення поставлених завдань для пришвидшення обробки звернення пацієнта з респіраторним захворюванням, а саме розробка моделей: обробки звернення пацієнта, вектору клініки та анамнезу, функціонування хмарного сервісу обробки звернення пацієнта, інформаційних сховищ та баз знань. Також представлено процес розробки web додатку для проведення практичної апробації розроблених моделей. Проведено практичну апробацію та тестування web додатку для визначення ефективності розроблених моделей обробки звернення пацієнта.

Першим завданням наукового дослідження було створення формули обрахунку загального часу обробки звернення пацієнта. Проаналізувавши логування проведених, на базі практики в кабінеті сімейного лікаря, спостережень звернень пацієнтів можна сформулювати загальну картину роботи сімейного лікаря з пацієнтом. Тому процес обробки звернення пацієнта з респіраторним захворюванням складається з наступних дій сімейного лікаря:

- відбір персональної інформації про пацієнта;
- відбір анамнезу звернення пацієнта;
- огляд пацієнта;
- розробка плану лікування;
- фіксація звернення пацієнта в амбулаторній карті (форма №025/о) та МІС.

Виділивши основні дії, які виконує сімейний лікар та проаналізувавши їх було розроблено формулу (3.1) обрахунку загального часу проведення та обробки звернення пацієнта.

$$p = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 + t_7 + t_8 + t_9 \quad (3.1)$$

де  $t_1$  витрачений час на відбір первинної інформації про пацієнта

$t_2$  витрачений час на визначення стану самопочуття пацієнта

$t_3$  витрачений час на огляд носа пацієнта

$t_4$  витрачений час на огляд ротової порожнини пацієнта

$t_5$  витрачений час на огляд горла пацієнта

$t_6$  витрачений час на визначення характеристик кашлю в пацієнта

$t_7$  витрачений час на аускультатию дихальної системи пацієнта

$t_8$  витрачений час на розробку плану лікування пацієнта

$t_9$  витрачений час на фіксацію звернення пацієнта

Таким чином використання формули (3.1) при проектуванні оптимізаційного рішення призведе до деталізації, поділу на етапи, структурування та приведення до загального шаблону, дій сімейного лікаря. За допомогою формули (3.1) можна встановити загальну структуру модулів, функціональних елементів та взаємозв'язок між ними для подальшої розробки оптимізаційного рішення. Також формула (3.1) дає змогу обрахувати загальний час витрачений на обробку звернення пацієнта. За рахунок поділу процесу обробки звернення пацієнта на етапи можна провести аналіз витрат часу на обробку на кожен з етапів та відповідно прийняти рішення щодо проведення дій для пришвидшення часу обробки на обраному етапі.

Розробка моделі інтерпретації текстової інформації відібраного анамнезу звернення пацієнта в вхідний вектор анамнезу для подальшої обробки.

Алгоритми машинного навчання обраховують та навчаються на інформації представлений в цифрах, але в даному випадку на вході існує тільки текстова інформація в вигляді відібраного анамнезу звернення пацієнта. Тому виникає

потреба розробити модель за допомогою якої можливо буде інтерпретувати вхідний анамнез в вигляді текстової інформації в вектор значень для подальшої його обробки. Підхід до реалізації завдання наступний:

- анамнез (вхідна інформація) поділити на категорії;
- кожному фактору впливу на виникнення захворювання (напр. головний біль, біль у горлі, вологий кашель) присвоїти унікальне цифрове значення;
- застосувати коефіцієнт зсуву та коригування для отримання більшого проміжку між результуючими показниками діагнозів. Збільшення проміжку призведе до збільшення точності визначення діагнозу. Наявність в вхідному векторі значення температури вимагає використання коефіцієнту коригування тому, що даний показник в кожного пацієнта різний і при відсутності коригування вектору можна отримати близькі або однакові за результуючим значенням в базі знань вектори для алгоритму машинного навчання.

Враховавши всі особливості для проведення інтерпретації анамнезу в вектор було створено модель інтерпретації (3.2)

$$i = \sum_0^n id * k \quad (3.2)$$

де  $n$  – кількість факторів впливу на виникнення захворювання (в машинному представленні кількість ітерацій),

$id$  – цифрове значення унікального номеру фактору впливу,

$k$  – коефіцієнт коригування та зсуву (обирається самостійно архітектором системи машинного навчання).

Використовуючи раніше розроблену формулу (3.1) процесу проведення звернення пацієнта було зроблено висновок, що модель інтерпретації (3.2) потрібно використовувати на кожному з етапів відбору анамнезу  $t_2, t_3, t_4, t_5, t_6, t_7$ . Таким чином відібраний анамнез можна повністю інтерпретувати в числовий вхідний вектор для його подальшої обробки. Для формування результуючого значення

вектору навчання, який буде вказувати на вірний діагноз було створено модель (3.3)

$$r = \sum_0^s \sum_0^n id * k \quad (3.3)$$

де  $s$  – кількість категорій згідно з якими проводиться відбір анамнезу,

$n$  – кількість факторів впливу на виникнення захворювання (в машинному представленні кількість ітерацій),

$id$  – цифрове значення унікального номеру фактору впливу,

$k$  – коефіцієнт коригування та зсуву (обирається самостійно архітектором системи машинного навчання).

В результаті роботи моделей (3.2) та (3.3) можна отримати вхідний вектор анамнезу (3.4)

$$\vec{v}_{learning} = \langle a, b, c, d, e, f, g, y \rangle \quad (3.4)$$

де  $a$  – результат роботи моделі інтерпретації (3.2) для категорії самопочуття,

$b$  – результат роботи моделі інтерпретації (3.2) для категорії температура,

$c$  – результат роботи моделі інтерпретації (3.2) для категорії огляд носа,

$d$  – результат роботи моделі інтерпретації (3.2) для категорії огляд ротової порожнини,

$e$  – результат роботи моделі інтерпретації (3.2) для категорії огляд горла,

$f$  – результат роботи моделі інтерпретації (3.2) для категорії кашель,

$g$  – результат роботи моделі інтерпретації (3.2) для категорії аускультация дихальної системи,

$y$  - результат роботи моделі інтерпретації (3.3) для утворення повної навчальної пари.

Далі представлено, на прикладі респіраторного захворювання синусит, процес інтерпретації анамнезу в вигляді текстової інформації в вхідний вектор.

Унікальні значення факторів впливу на виникнення захворювання наведено у таблиці А.1. Нижче описано клініку синуситу та унікальні значення факторів впливу.:

- самопочуття: гарячка - 6, головний біль - 9, втомлюваність - 1, закладеність вух – 10;
- температура: до  $37.5C^{\circ}$  - до 375;
- ніс: порушення нюху - 13, виділення слизу та гною – 17;
- рот: неприємний запах з роту - 18, зубний біль – 19;
- кашель: присутній – 23;
- горло: фактори впливу відсутні – 0;
- дихання: порушення носового дихання – 36;

Підставивши значення в формулу (3.2) було отримано наступні значення:

- самопочуття –  $(6 * 2) + (9 * 2) + (1 * 2) + (10 * 2) = 52$
- температура – 375
- ніс –  $(13 * 2) + (17 * 2) = 60$
- рот –  $(18 * 2) + (19 * 2) = 74$
- кашель –  $23 * 2 = 46$
- горло – 0
- дихання -  $36 * 2 = 72$

Також для утворення результуючого значення було використано формулу (3.3) підставивши в неї значення отримані формулою (3.2),  $52 + 375 + 60 + 74 + 46 + 0 + 72 = 679$ .

Таким чином обрахувавши всі значення можна отримати вектор (3.5) анамнезу або клініки захворювання.

$$\vec{v}_{learning} = \langle 52, 375, 60, 74, 46, 0, 72, 679 \rangle \quad (3.5)$$

Вектор (3.4) можна використовувати для запису в базу знань клініки виникнення захворювання, знеособленого анамнезу пацієнтів та для формування вектору – запиту для створення прогнозу або класифікації діагнозу. Основною

позитивною характеристикою використання даних моделей (3.2), (3.3) є мінімізація вектору. В даному науковому дослідженні відібрані респіраторні захворювання формують множину з 52-х факторів впливу тому використовуючи вже існуючі підходи до формування вектору потрібно було б формувати вектор який містить в собі 52 значення, більшість з яких приймали б значення 0. Даний підхід до формування мінімізованого вектору є інноваційним. Саме використання мінімізованого вектору в порівнянні з звичайним розгорнутим вектором призведе до зменшення використання трафіку при «спілкуванні» з сервером. Також призведе до зменшення обчислювального ресурсу внаслідок зменшення навантаження на обчислення та зменшення часу на перенавчання алгоритму штучного інтелекту. А також зменшення часу на отримання актуального зліпку бази знань для проведення перенавчання та зменшення об'єму бази знань.

Розробити моделі баз знань для зберігання дамтів необхідної інформації.

Для повного забезпечення функціонування моделі оптимізації обробки звернення пацієнтів з респіраторним захворюванням використовуючи алгоритми машинного навчання потрібно використовувати два види інформаційного сховища а саме:

- база знань для збереження клініки респіраторних захворювань та знеособленого анамнезу звернень пацієнтів;
- інформаційне сховище для збереження препаратів лікування діагностованих захворювань та планів лікування.

Основними вимогами для реалізації бази знань є невеликий вихідний розмір, можливість швидко отримати актуальний зліпок для чергового перенавчання моделі на льоту, легке та швидке редагування бази знань. Також виникає вимога редагування архітектури бази знань без руйнування існуючої на випадок перенавчання алгоритму з новими респіраторними захворюваннями. Тому врахувавши всі вимоги було прийнято рішення взяти за основу базу знань типу «.csv». Архітектуру бази знань розроблено на основі матричної моделі (3.6).

$$K_{db} = \begin{bmatrix} a_{11} & b_{12} & c_{13} & d_{14} & e_{15} & f_{16} & g_{17} & y_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{ij} & b_{ij} & c_{ij} & d_{ij} & e_{ij} & f_{ij} & g_{ij} & y_i \end{bmatrix} \quad (3.6)$$

де  $a$  – значення клініки або анамнезу для категорії самопочуття,

$b$  – значення температури,

$c$  – значення клініки або анамнезу для категорії огляд носа,

$d$  – значення клініки або анамнезу для категорії огляд ротової порожнини,

$e$  – значення клініки або анамнезу для категорії огляд горла,

$f$  – значення клініки або анамнезу для категорії кашель,

$g$  – значення клініки або анамнезу для категорії аускультация дихальної системи,

$y$  – результуюче значення для встановлення вірного варіанту захворювання, яке відповідає клініці чи анамнезу та формування повної навчальної пари.

Повний приклад бази знань, яка містить в собі клініку респіраторних захворювань, наведено у таблиці А.2. Матричну модель (3.6) можна з легкістю модифікувати без руйнування вже існуючої структури. Дана властивість модульності архітектури виникає внаслідок поділу анамнезу на категорії. Розмір вихідного файлу бази знань, який містить в собі вектори клініки відібраних респіраторних захворювань для проведення даного наукового дослідження, складає близько 360 байт. Розмір вихідного файлу бази знань, який містить в собі один вектор клініки або анамнезу складає близько 24 байт. Таким чином проаналізувавши отримані результати можна зробити висновок, що створена архітектура бази знань за використаною моделлю (3.6) повністю задовольняє всі вимоги.

Для створення інформаційного сховища з планами лікування були висунуті наступні вимоги: чітка структуризація інформації, яка зберігається та можливість представлення інформації в вигляді об'єкту для проведення індексації значень та пришвидшення пошуку необхідної інформації.

Тому в даному випадку оптимальний формат інформаційного сховища є JSON об'єкт. Модель інформаційного сховища представлено на рисунку 3.1. Дана

модель працює методом семплування інформації. Тобто база даних містить в собі наперед заготовлені частини інформації плану лікування і в потрібний момент на запит з алгоритму машинного навчання або користувача видає відповідь в вигляді пулу інформації з планом лікування.

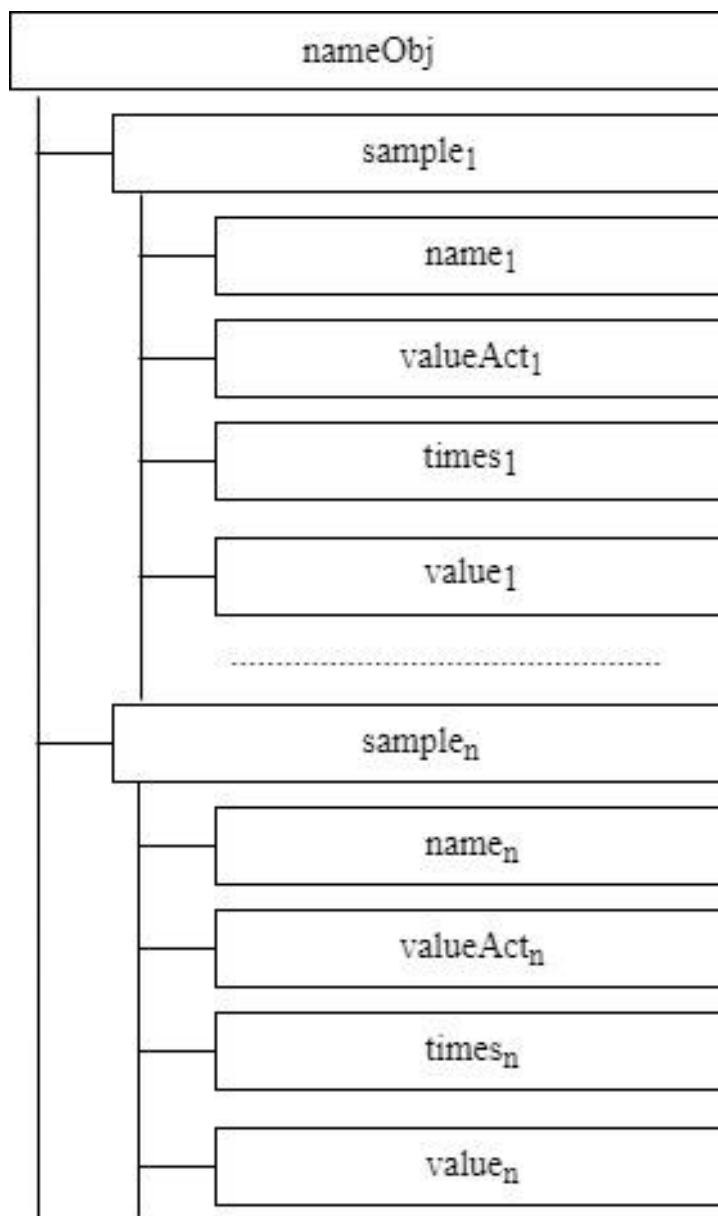


Рисунок 3.1 – Ієрархічна схема архітектури інформаційного сховища планів лікування.

Структура схеми наступна:

- nameObj – ім'я плану лікування (можна використовувати шифри діагнозу);
- sample<sub>n</sub> – назва та порядковий номер семплу плану лікування,

- name – торгова назва препарату;
- valueAct – доза діючої речовини в препараті;
- times – кількість разів прийому препарату на день;
- value – кількість днів прийому препарату.

Така архітектура має відносно неглибокі вкладення, до кінцевого об'єкту 2 вкладення, тому можна провести швидко індексацію об'єктів та швидко створювати вибірки семплів інформації для формування плану лікування за заданими критеріями. Тому використовуючи модель (3.6) та модель зображену на рисунку 3.1 можна створити базу знань для навчання алгоритму штучного інтелекту та інформаційне сховище для планів лікування, в яких буде:

- невеликий час для отримання актуальних зліпків бази знань або семплів інформації плану лікування;
- можливість швидкої, легкої модифікації архітектури бази знань та інформаційного сховища;
- невеликий розмір вихідних файлів;
- змога швидкого опрацювання вмісту інформації сервером, внаслідок чого відбудеться зменшення часу отримання відповіді клієнтом.

Визначити оптимальний алгоритм машинного навчання, який задовільнить умову мінімізації часу обробки вхідного потоку даних (відібраний анамнез звернення пацієнта), прогнозування або класифікації діагнозу та перенавчання моделі.

Для визначення оптимального алгоритму машинного навчання було взято наступні алгоритми: алгоритм наївного Баеса, метод опорних векторів та метод k – ближніх сусідів. Визначення оптимального алгоритму проводилося на локальному пристрої та на хостинг сервері. Обчислювальні характеристики пристроїв наведено у таблиці 3.1.

Основним критерієм вибору оптимального алгоритму був критерій мінімізації часу на обробку вхідного вектору та використання пам'яті для обробки. Заміри часу проводилися за наступним алгоритмом:

- провести захват відрізка обробки вхідного вектору;

- визначити загальний час обчислення по категорії виконання скриптів;
- визначити загальний час по активності дії;
- визначити час на обробку запиту сервером;
- визначити використану пам'ять для проведення обчислення;
- провести аналіз відібраної інформації.

Таблиця 3.1 – Обчислювальні характеристики пристроїв та мережі для визначення оптимального алгоритму машинного навчання.

Локальний пристрій	
Центральний процесор	Intel core i7 тактова частота 3.4 GHz
ОЗП	16 GB, 2400 Mhz
Сервер	Apache 2.4, PHP 7.1
База даних	«csv» та об'єкти JSON
Хостинг сервер	
Центральний процесор	Intel core i7 тактова частота 3.0 GHz
ОЗП	8 GB, 1600 Mhz
Сервер	Apache 2.4, PHP 7.4
База даних	«csv» та об'єкти JSON
Швидкість мережевого підключення (вхідна)	7.10 Mbit
Швидкість мережевого підключення(вихідна)	0.44 Mbit
Пінг мережевого підключення	51 с

Характеристики витрати обчислювального часу та пам'яті обробки вхідного вектору для проведення аналізу для локального пристрою та хост сервера наведено у таблиці 3.2. Кількість векторів в базі знань дорівнює кількості векторів клініки обраних для даного наукового дослідження респіраторних захворювань, це 16 векторів. Час для відбору активності було встановлено 5 секунд. Інструментом

відбору даних для аналізу були модулі Network та Performance застосунку Google chrome console.

Таблиця 3.2 – Характеристики обробки вхідного вектору на локальному пристрої та хостинг сервері.

	АНБ	МОВ	k – бс	АНБ	МОВ	k – бс
	лс	лс	лс	хс	хс	хс
Загальний час обчислення по категорії виконання скриптів (мс)	89.0	115.0	112.0	138.0	178.0	172.5
Загальний час по активності дії (мс)	8.9	15.1	12.3	8.6	14.5	11.9
Час на обробку запиту сервером (мс)	2.0	8.2	5.4	1.7	6.9	4.6
Використана пам'ять для проведення обчислення (мб)	2.6	3.1	2.9	4.5	5.5	5.2
Кількість векторів в базі знань	16	16	16	16	16	16
Час відбору активності (сек.)	5.0	5.0	5.0	5.0	5.0	5.0

Проаналізувавши дані в таблиці 3.2 можна зробити висновки, що метод опорних векторів, внаслідок того, що це бінарний класифікатор і необхідно використовувати модель надбудову для мультикласифікації, виконується найдовше. Також на локальному пристрої та хостинг сервері метод опорних векторів показав низькі результати використовуючи найбільше пам'яті для обчислення. Метод k – ближніх сусідів показав кращі результати ніж метод опорних векторів зменшивши час виконання на 2.8 мс. Також даний метод менше використовує пам'ять для обчислення. Алгоритм наївного Баєса показав найкращі результати таким чином на хостинг – сервері даний алгоритм з базою знань 16

векторів виконується 1.7 мс. і використовує 4.5 мб. пам'яті для обчислення. Можна зробити припущення, що при збільшені кількості векторів в базі знань буде збільшуватися час обробки та пам'ять для обчислення. Тому важливо використовувати алгоритм з найменшими значеннями витрати часу та використання ОЗП. Також алгоритм наївного Баєса підтримує мультикласифікацію без додаткових моделей, а його недолік в потребі використання великого об'єму бази знань з легкістю можна перетворити в перевагу наступним чином. При кожному зверненні пацієнтів база знань буде оновлюватися новими векторами анамнезу і тому при збільшені векторів алгоритм зможе більш точно класифікувати діагноз. Таким чином оптимальним алгоритмом машинного навчання для проведення класифікації діагнозу згідно з відібраним анамнезом є алгоритм наївного Баєса тому, що він задовольняє всі поставлені умови.

Згідно з розробленим комплексом моделей та формул було розроблено модель оптимізації, для пришвидшення, обробки звернення пацієнта. Модель представлена у вигляді графічної моделі (блок - схема). Використовуючи структуру формули (3.1) було сформовано наступні вимоги до функціонування хмарного аналітичного середовища:

- модульність архітектури, можливість функціонування модулів незалежно один від одного;
- забезпечення процесу перенавчання штучного інтелекту на льоту;
- автоматизоване визначення діагнозу за допомогою алгоритму наївного Баєса згідно з відібраним анамнезом;
- автоматизоване формування плану лікування;
- можливість автоматизованого формування детального опису звернення пацієнта;
- шифрування звернення пацієнта за системами ІСРС 2 та МКХ - 10;
- друк документації, запис до амбулаторної картки форма №025/о та плану лікування.

Враховавши всі вимоги було розроблено модель функціонування хмарного аналітичного середовища. Модель представлена на рисунку 3.2.

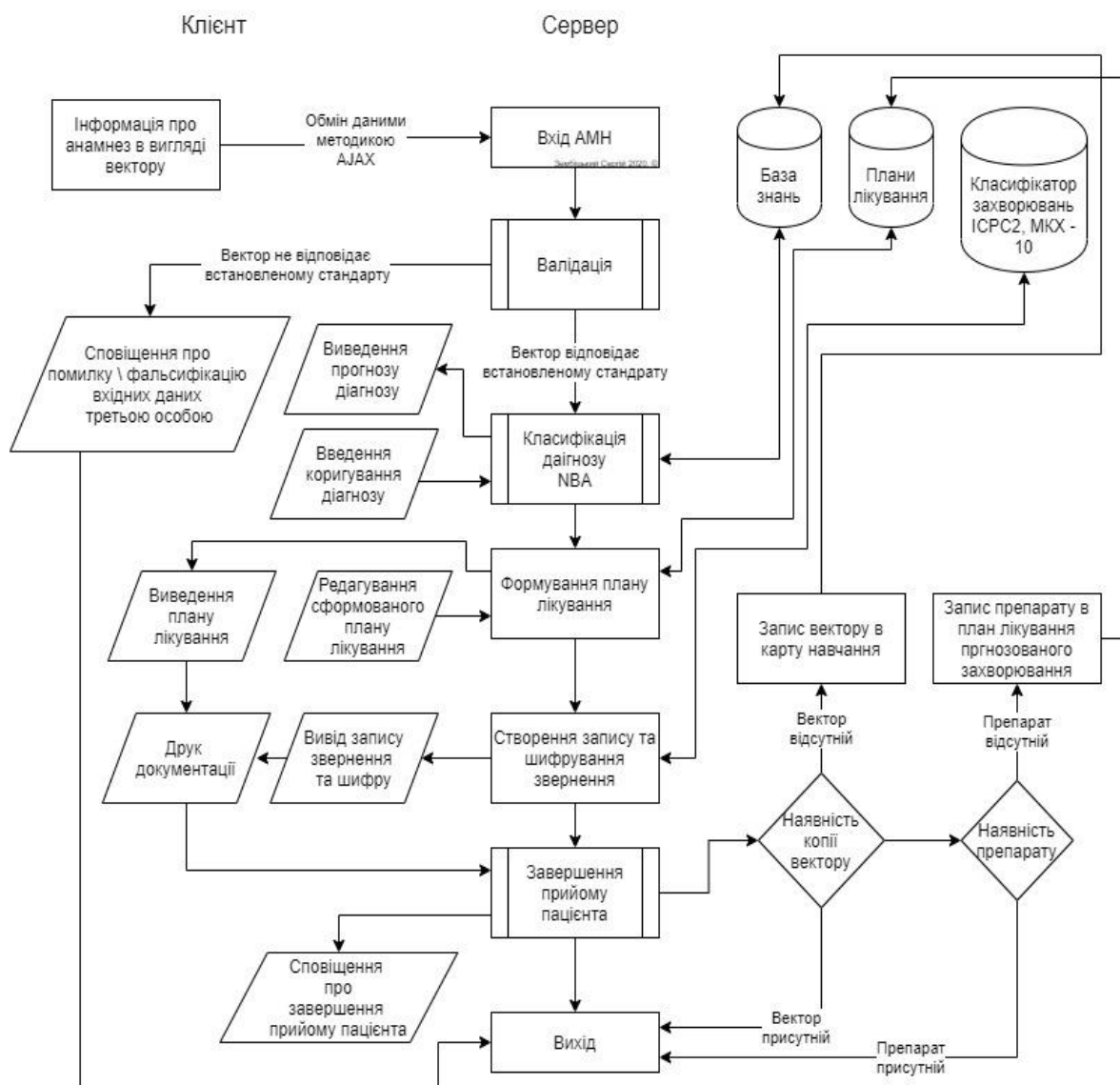


Рисунок 3.2 – Графічна модель оптимізації (блок - схема) обробки звернення пацієнта.

Середовище функціонує за наступним алгоритмом: після введення анамнезу пацієнта на web сторінці на вхід подається методом AJAX вектор анамнезу пацієнта. Вектор проходить валідацію, при невідповідності вектора до встановленої структури користувачу виводиться на екран сповіщення про помилку в системі або подробиці даних з боку третіх осіб і рекомендується перезавантажити сторінку, якщо вектор задовольняє всі умови та відповідає встановленій структурі він передається на вхід обчислювального ядра алгоритму наївного Баєса. На етапі класифікації ядро АНБ отримує актуальний зліпок бази знань з карти навчання, проводить класифікацію вхідного вектору, передає методом AJAX користувачу

інтерпретований вихідний вектор, який містить в собі назву діагнозу відповідно до введеного анамнезу, та виводить його на екран. Задля збереження людського фактору в системі розроблена можливість змінювати діагноз якщо на думку користувача (сімейного лікаря) він не відповідає тому, що встановив НБА. На етапі формування плану лікування пацієнта на вхід методом AJAX подається вектор діагнозу, ядро відповідальне за формування плану лікування звертається до JSON структурованого об'єкту, який містить плани лікування та формує, методом семпсування інформації, план лікування згідно з заданими параметрами. Сформований план лікування передається користувачу методом AJAX, пулом даних та виводиться на екран. За потреби користувач може провести редагування плану лікування. Далі процес виконання передається в модуль автоматизованого формування запису фіксації звернення пацієнта. Після формування запису звернення пацієнта проводиться шифрування епізоду звернення пацієнта. На даному етапі ядро модулю шифрування звертається до JSON структурованого об'єкту, який містить шифри ICPC2 та МКХ – 10 та за вхідним вектором кінцевого діагнозу, переданим методом AJAX, проводить шифрування епізоду за системами ICPC2 та МКХ – 10. Модуль шифрування епізоду звернення повертає значення шифру методом AJAX та виводить на екран користувача. Після отримання користувачем сформованого плану лікування та запису до амбулаторної карти пацієнта (форма №025/о) модуль виводу інформації заповнює шаблони друку документів отриманою інформацією та передає на ядро друку документи, які друкуються за допомогою принтера. На завершальному етапі проводиться процес підготовка до вивантаження пулу даних до баз знань. Спершу проходить перевірка на наявність точної копії в базі знань введеного користувачем вектору анамнезу. При відсутності точної копії вектору анамнезу модуль завершення епізоду проводить перевірку на відповідність структури вектору та вносить новий вектор в базу знань векторів. Якщо точна копія вектору наявна в базі знань модуль завершення епізоду переходить до аналізу плану лікування. Аналізатор плану лікування проводить пошук в структурованому об'єкті JSON бази знань препарати наявні в плані лікування. При відсутності, препарати вносяться в базу знань, при

наявності ігноруються. Після внесення пулу інформації до баз знань клієнту виводиться повідомлення про успішність завершення прийому пацієнта [26].

Таким чином використовуючи дану модель функціонування можна розробити програмне рішення на основі web технологій та технологій машинного навчання для пришвидшення проведення прийому пацієнтів з респіраторним захворюванням і полегшення ведення документації сімейним лікарем. В даній моделі передбачена модульність архітектури, що дає змогу незалежного використання модулів. Наприклад якщо сімейний лікар не бажає вводити анамнез для визначення респіраторного захворювання, а йому потрібно лише сформуванню плану лікування, він може обрати діагноз та передавши вектор обраного діагнозу мережа автоматично сформує план лікування. Також модульність зберігає за собою вплив людського фактору на кінцевий результат на випадок якщо сімейний лікар вважає, що захворювання в пацієнта відрізняється від поставленого мережею, лікар зможе змінити діагноз та продовжити формування плану лікування для обраного захворювання. Пришвидження обробки звернення пацієнта відбувається на етапі шифрування звернення пацієнта за системами ІСРС 2 та МКХ – 10, використовуючи модуль автоматизованого шифрування лікарю не потрібно в ручному режимі проводити пошук кодів діагнозів в довідниках – класифікаторах, всю роботу за нього виконає модуль формування запису до амбулаторної карти пацієнта та його шифрування. Повний процес навчання забезпечується встановленням зв'язку з базами знань, при проведенні завершення звернення, для подальшого їх поповнення новими векторами анамнезу прийомів пацієнтів. Таким чином при завершенні прийому пацієнта в бази знань записуються, при умові відсутності, вектори анамнезу звернення пацієнта та при наступному зверненні іншого пацієнта ядро алгоритму наївного Баєса отримує актуальний зліпок та проводить перенавчання на льоту для класифікації діагнозу. Так як виконання даного алгоритму на хостинг сервері займає лише 1.7 мс., з'являється можливість виконувати перенавчання моделі дійсно на льоту і відповідно швидко отримати результат діагнозу. Для забезпечення повного проведення та обробки звернення пацієнта в моделі передбачено використання модулю друку. За допомогою даного

модулю сформовані документи друкуються і сімейний лікар не витрачає часу на подвійну роботу в МІС та на папері. Тому далі буде розроблено web застосунок для проведення практичної апробації використовуючи модель зображену на рисунку 3.2 та проведено оцінку ефективності всіх розроблених моделей.

### 3.2 Розробка програмного рішення

Розробивши архітектуру функціонування хмарного середовища та моделі інтерпретації і обробки вхідної інформації, було перейдено до етапу розробки програмного рішення для проведення практичної апробації ефективності моделей. Для покриття більшої кількості пристроїв кінцевих користувачів було прийнято рішення розробки кросплатформового додатку. В даному науковому дослідженні було розроблено web застосунок з адаптивним позиціонуванням об'єктів та елементів користувача. Також в основі графічного дизайну та філософії представлення інформації для користувача було використано Material Design. Функціонування клієнтської частини забезпечує HTML, CSS, JavaScript, jQuery, функціонування серверної частини та алгоритму наївного Баєса забезпечує PHP. Проаналізувавши всі доступні та рекомендовані eHealth МІС було сформовано функціональну частину web застосунку, яка дасть змогу повністю забезпечити та пришвидшити обробку прийому пацієнта з респіраторним захворюванням. Функціонал web застосунку представлено нижче:

- пошук пацієнтів;
- модуль представлення інформації про пацієнта;
- модуль відбору анамнезу в пацієнта, складається з категорій самопочуття, огляд носа, огляд ротової порожнини, огляд горла, кашель, аускультация дихальної системи;
- модуль прогнозування діагнозу та формування плану лікування з використанням алгоритмів машинного навчання;

- модуль формування запису до амбулаторної картки пацієнта.

Клієнтську частину було спроектовано за структурою описаною в формулі (3.1), та моделі оптимізації обробки звернення пацієнтів (рисунок 3.2). Проектування web застосунку відбувалося використовуючи підходи до реалізації SPA додатків та модель MVC для забезпечення модульної структури логіки, представлення та контролерів. Далі представлено структуру клієнтської частини.

Модуль пошуку пацієнта відповідає за встановлення зв'язку з сервером методом AJAX, відправку пошукового запиту та отримання відповіді пулу інформації про пацієнта. Складається з елементів користувача типу «panel», «select» та «button». Зовнішній вигляд модулю представлено на рисунку Б.1.

Модуль представлення інформації про пацієнта виконує заповнення HTML шаблону який забезпечує структуроване відображення отриманої інформації про пацієнта з модулю пошук пацієнта. Інформація подається в вигляд таблиці, яка містить в собі 2 стовпця та 15 рядків. Структура таблиці наступна: Прізвище, Ім'я, По батькові, Дата народження, Вік, Стать, Контактний номер, Протипоказання, Країна, Індекс, Область, Район, Населений пункт, Вулиця, Будинок/ корпус/ квартира. Таблиця адаптується під різні розширення екранів, тому на мобільних пристроях таблиця відображає всі дані в одному стовпці перемістивши інформацію з другого стовпця в рядки. Модуль містить елементи користувача типу «panel», «table», «text». Зовнішній вигляд модулю представлено на рисунку Б.2.

Модуль відбору анамнезу відповідає за збір текстового варіанту анамнезу та формування, структурування і підготовку анамнезу до інтерпретації в вхідний вектор. Даний модуль включає в себе фактори впливу на виникнення захворювання із 6 встановлених категорій: самопочуття, огляд носа, огляд ротової порожнини, огляд горла, кашель, аускультация дихальної системи. Також було передбачено можливість створення приміток до відібраного анамнезу. Модуль містить елементи користувача типу «panel», «img», «text», «input:checkbox», «input:number» та «input:text». Використовуючи елементи «input:checkbox» провівши ініціалізацію та параметризацію «input:checkbox>value» значеннями розрахованими за формулою (3.2) можна в подальшому з легкістю інтерпретувати анамнез у вхідний вектор.

Елементи «input:number» та «input: text» використовуються для отримання значення температури та приміток до анамнезу. Параметризація елементів зображена на рисунку 3.3. Зовнішній вигляд модулю представлено на рисунках Б.3 — Б.8.

```

1 | <form method="post">
2 |   <div class="workPanel">
3 |     <div class="sohi"></div>
4 |     <div class="textDescription">Самопочуття</div>
5 |     <hr>
6 |     <div class="cbWrapper">
7 |       <input type="checkbox" id="cb01"
8 |         name="sohicb[]" value="1" class="input-checkbox">
9 |       <label class="cbLabel" for="cb01">Втомлюваність</label>
10|     </div>
11| .....
12|     <div class="group">
13|       <input type="number" class="itextPatient" id="t"
14|         value="36.6" min="35.0" max="42.0" step="0.1" required>
15|       <span style="width: 98%;" class="bar"></span>
16|       <label class="tbLabel">Температура</label>
17|     </div>
18|     <div class="spacer"></div>
19|   </div>
20| </form>

```

Рисунок 3.3 – Ініціалізація та параметризація елементів користувача для відбору анамнезу пацієнта.

Модуль прогнозування діагнозу та формування плану лікування відповідає за формування вхідного вектору анамнезу, відправки вектору на сервер та отримання результату з сервера в вигляді прогнозованого діагнозу. Також за допомогою даного модулю проводиться формування та друк плану лікування. Встановлення зв'язку з сервером забезпечує підхід AJAX. Складається даний модуль з елементів типу «panel», «button», «select», «table» та «img». Зовнішній вигляд представлено на рисунку Б.9. Модуль працює наступним чином. Після введення анамнезу в модулі відбору анамнезу, користувач натиснувши на елемент «ДІАГНОЗ» ініціює подію, яка на стороні клієнта формує вхідний вектор та відправляє його на сервер, в свою чергу сервер після отримання вектору проводить

прогнозування та відправляє результат клієнту. Функція обробки прогнозу представлено на рисунку 3.4.

```

1 | $('#diagnosis').click(function(){
2 |     $.ajax({
3 |         url: "core/c755c6100",
4 |         method: "POST",
5 |         dataType: "json",
6 |         data: {"sohi": na(), "temperature": ta(), "nose": nb(),
7 |             "mouth": nc(), "cough": nd(), "throat": ne(), "breath": nf()},
8 |         success: function (msg) {
9 |             $('#selectDiag').val(msg);
10|             $('.serverL').fadeOut();
11|         }
12|     });
13| });

```

Рисунок 3.4 – Функція відправки вхідного вектору на сервер та обробки прогнозу діагнозу.

На стороні клієнта отриманий результат відображається в елементі «select». Представлення результату відбувається в елементі «select» для збереження людського фактору. На випадок якщо сімейний лікар вважає, що реальний діагноз відрізняється від прогнозованого, тоді він з легкістю зможе змінити діагноз і продовжити роботу з новим, на його думку вірним, діагнозом. При завершенні прийому пацієнта вектор з відібраним анамнезом і значенням вірного діагнозу відправиться на сервер в базу знань. Таким чином відбудеться процес утворення нових знань штучного інтелекту. Якщо ж сімейний лікар згідний з прогнозованим діагнозом, для отримання плану лікування він натиснувши на елемент «ПЛАН» ініціює процес відправки, на сервер, значення діагнозу яке приймається з елемента «select». Сервер після отримання значення формує план лікування та передає його об'єктом JSON, клієнту. Клієнт розбирає дані згідно структури та відображає їх в елементі «table». Функція обробки плану лікування зображена на рисунку 3.5. При потребі користувач може редагувати план лікування. Друк плану лікування відбувається на стороні клієнта після натискання користувачем на елемент «ДРУК». Процес виконання супроводжується викликом модального вікна

системного друку в браузері та передає в якості аргументів html структуру таблиці. Для зручності та економії паперового ресурсу друк плану лікування проводиться на папері формату А5. Функцію друку плану лікування зображено на рисунку 3.6. Приклад плану лікування наведено на рисунку Б.13.

```

1 | $('#medicaments').click(function(){
2 |     $('#tableTh').find("tr:gt(0)").remove();
3 |     $.ajax({
4 |         url: "core/pt9b54155a",
5 |         method: "POST",
6 |         dataType: "json",
7 |         data: {"diag": $('#selectDiag').val()},
8 |         success: function (msg) {
9 |             for (let index = 0; index < msg.length; index++) {
10|                 $('#tableTh')
11|                 .append('<tr><td><input type="text" class="m-text" id="m-name" value="'
12|                 + msg.pot[index].name + '></td>' +
13|                 '<td><input type="text" class="m-text" id="m-valueAct" value="'
14|                 + msg.pot[index].valueAct + '></td>' +
15|                 '<td><input type="text" class="m-text" id="m-times" value="'
16|                 + msg.pot[index].times + '></td>' +
17|                 '<td><input type="text" class="m-text" id="m-value" value="'
18|                 + msg.pot[index].value + '></td>' +
19|                 '<td><input type="button" id="deleteRow" name="prog"
20|                 class="buttonText" value="ВИДАЛИТИ"></td></tr>');
21|             }
22|         }
23|     });
24| });

```

Рисунок 3.5 – Функція відправки запиту та обробки відповіді з планами лікування.

Пристрої, які працюють на мобільних операційних системах, документація для друку формується в залежності від системних налаштувань ядра друку. Пристрої на базі Android OS сформований документ завантажують на кінцевий пристрій з подальшим запитом на друк. Пристрої на базі iOS та iPadOS використовують для друку технологію AirPrint.

```

1 | $(function(){
2 |     $('#printPot').click(function(){
3 |         var html_to_print = " ";
4 |         $('#tableTh > tbody > tr > td > .m-text').each(function(){
5 |             html_to_print += $(this).val();
6 |         });
7 |         var iframe=$('#<iframe id="print_frame">');
8 |         $('body').append(iframe);
9 |         var doc = $('#print_frame')[0].contentDocument || $('#print_frame')[0].contentWindow.document;
10|         var win = $('#print_frame')[0].contentWindow || $('#print_frame')[0];
11|         doc.getElementsByTagName('body')[0].innerHTML=html_to_print;
12|         win.print();
13|         $('#iframe').remove();
14|     });
15| });

```

Рисунок 3.6 – Функція друку плану лікування.

Модуль формування запису до амбулаторної карти пацієнта виконує автоматизоване формування запису до амбулаторної карти звернення пацієнта та шифрує звернення за класифікатором ІСРС 2 і МКХ – 10. Модуль складається з елементів типу «panel», «text», «button», «input:textarea» та «img». Зовнішній вигляд представлено на рисунку Б.10. Модуль формування запису проводить збір і структурування інформації з модулів розміщених на сторінці та виводить в елемент «input:textarea» інформацію згідно з розробленим шаблоном. Також після формування запису проводиться шифрування звернення. Для шифрування встановлюється з'єднання з сервером, відправляється на сервер значення діагнозу та приймається відповідь в вигляді об'єкту JSON для подальшого відображення в елементах «text». Лістинг функції формування запису звернення зображено на рисунку 3.7. При потребі сімейний лікар натиснувши на елемент «ДРУК» може провести друк запису звернення для подальшого прикріплення в амбулаторну карту пацієнта форма №025/о. Приклад запису наведено на рисунку Б.14. Згідно структурних вимог форми №025/о друк відбувається на папері формату А5.

Модуль сайд меню та хедер. Принципи побудови структури сторінки відповідно Material вимагають обов'язкового використання елемента сайд меню для забезпечення зручного та швидкого доступу до основних об'єктів та елементів користувача. Модуль складається з елементів типу «panel», «text», «button» та «img». Зовнішній вигляд представлено на рисунку Б.11. В сайд меню розміщено

елементи для відображення основної інформації про профіль користувача, робочу дату та елементи швидкого доступу до функціоналу. Логіка виконання функцій швидкого доступу побудована на основі click – функцій. При ініціюванні події click – функція імітує натискання елемента «button» попередньо задавши в якості аргументу функції селектор елемента користувача для якого потрібно імітувати подію натискання. Лістинг click – функції представлено на рисунку 3.8. При відображенні сайд меню відбувається фокусування на батьківському елементі та затемнюється задній фон. Викликається за натисканням елемента типу «button» розташованим в хедері. Хедер (головний блок) використовується для відображення основної інформації та швидкого доступу до неї. Використовуючи структурний шаблон сайд меню та хедер було створено хот кейси для швидкого доступу до функціоналу з будь якого місця сторінки.

```

1 | $("#writeb").click(function(){
2 |     $.ajax({
3 |         url: "core/c355b7338",
4 |         method: "POST",
5 |         dataType: "json",
6 |         data: {"diag": $('#selectDiag').val()},
7 |         success: function (msg) {
8 |             $('#icpc2').text('ICPC 2: ' + msg.icpc2);
9 |             $('#icd10').text('МКХ - 10: ' + msg.icd10);
10|         }
11|     });
12|     $('#ambulatoryCard').val('');
13|     $('#ambulatoryCard').html('');
14|     $('#ambulatoryCard').val($.trim($('#ambulatoryCard').val()) + '\n' + 'Клініка епізоду: ');
15|     $('#ambulatoryCard').html($.trim($('#ambulatoryCard').val()) + '\n' + 'Клініка епізоду: ');
16|
17| .....
18|
19|     $('#ambulatoryCard').val($.trim($('#ambulatoryCard').val()) + '\n' + 'План лікування: ');
20|     $('#ambulatoryCard').html($.trim($('#ambulatoryCard').val()));
21|     $('#tableTh > tbody > tr > td > .m-text').each(function(){
22|         $('#ambulatoryCard').val($.trim($('#ambulatoryCard').val()) + ' ' + $(this).val() + ', ');
23|         $('#ambulatoryCard').html($.trim($('#ambulatoryCard').val()) + ' ' + $(this).html() + ', ');
24|     });
25| });

```

Рисунок 3.7 – Функція формування запису до амбулаторної карти пацієнта.

```

1 | function dcl(){
2 |     document.querySelector('#diagnosis').click();
3 | }

```

Рисунок 3.8 – Лістинг click – функції швидкого доступу прогнозування діагнозу.

Також в розробленому web – застосунку використовуються додаткові елементи користувача типу «panel» з властивістю модального вікна для відображення додаткової інформації. Всі елементи користувача інтерактивні та взаємодіють з користувачем. Використана тінь та підсвітка елементів для виділення потрібної інформації. В заголовках елементів типу «panel» використовується відповідні зображення для формування інтуїтивно зрозумілої функціональності модулю. Кожний модуль виділено окремо елементом користувача типу «panel». Кожний функціональний елемент «panel» має заголовок в якому розміщено зображення функціональності модулю, назва модулю та за потреби елементи керування модулем типу «button». Адаптація web застосунку під екрани різного формату відбувається за допомоги відсоткового співвідношення розміру всіх елементів сторінки. Зовнішній вигляд допоміжних елементів користувача представлено на рисунку Б.12.

Серверна частина розроблялася відповідно до моделі функціонування архітектури хмарного обчислювального середовища для визначення респіраторних захворювань (рисунок 3.2). Основні функції серверної логіки полягають в забезпеченні прогнозування діагнозу, формування плану лікування та шифруванні звернення пацієнта.

Прогнозування діагнозу забезпечує виконання ядра алгоритму найвнього Баєса. Лістинг коду виконання представлено на рисунку 3.9. Після отримання сервером запиту від клієнта відбувається приведення вектору до структури JSON. Отримання вхідного вектору від клієнта забезпечує метод POST з захищеним протоколом передачі даних HTTPS. В include директивах відбувається підключення ядра алгоритму найвнього Баєса та процесора обробки csv формату баз знань. Після успішного підключення директив процесор обробки csv отримує актуальний зліпок бази знань, створює динамічну змінну та розміщує в ній, відповідну до встановленої структури, інформацію з бази знань. Перенавчання моделі відбувається на льоту після утворення екземпляру об'єкту класу алгоритму найвнього Баєса. При класифікації діагнозу функція прогнозування параметризується аргументами значень вхідного вектору. Отримавши результат

класифікації, відбувається приведення результату до структури JSON та відправка клієнту раніше встановленим підключенням.

```

1 | <?php
2 | //Serhii Zembitskyi. All rights reserved. 06.10.2020
3 | $data_json = json_encode($_POST);
4 | $anamnesis = json_decode($data_json);
5 |
6 | require "../vendor/autoload.php";
7 | use Phpml\Classification\NaiveBayes;
8 | use Phpml\Dataset\CsvDataset;
9 |
10| $dataset = new CsvDataset('../dataset/data.csv' , 7, false);
11| $classifier = new NaiveBayes();
12| $classifier->train($dataset->getSamples(),$dataset->getTargets());
13|
14| echo json_encode($classifier->predict([
15|     $anamnesis->sohi,
16|     $anamnesis->temperature,
17|     $anamnesis->nose,
18|     $anamnesis->mouth,
19|     $anamnesis->cough,
20|     $anamnesis->throat,
21|     $anamnesis->breath
22| ]));
23| ?>

```

Рисунок 3.9 – Лістинг блоку коду прогнозування респіраторного захворювання.

Формування плану лікування відбувається після отримання значення діагнозу від клієнта. Зв'язок між клієнтом та сервером встановлюється методом POST по захищеному протоколу з'єднання HTTPS. Після нормалізації вхідного значення та приведення до структури об'єкту JSON модуль формування плану лікування проводить пошук серед файлів бази знань семпли плану лікування, згідно з заданими критеріями, збирає в один об'єкт всі семпли та проводить нормалізацію структури у відповідність до JSON об'єкту. Семпли планів лікування зберігаються в файлах з JSON структурою. Назва файлів складається з шифру діагнозу за МКХ – 10 та розширення «.medic». Використовуючи шифри діагнозів в іменах файлів можна швидко здійснити пошук файлу бази знань з планом лікування потрібного захворювання. Загальний вигляд файлу бази знань плану лікування зображено на рисунку 3.10. Зібраний та структурований об'єкт JSON з

планом лікування відправляється клієнту раніше встановленим каналом зв'язку. Частковий лістинг коду функціонування формування плану лікування представлено на рисунку 3.11.

```

1 | {
2 |     "pot":[
3 |         {
4 |             "name": "Левифлоксацин",
5 |             "valueAct": "750 мг",
6 |             "times": "1р/д",
7 |             "value": "5 днів"
8 |         },
9 |     ]
10| }
```

Рисунок 3.10 – Лістинг коду об'єкту семплу плану лікування.

```

1 | <?php
2 | //Serhii Zembitskyi. All rights reserved. 06.10.2020
3 | $data_json = json_encode($_POST);
4 | $in = json_decode($data_json);
5 | $pot = null;
6 |
7 | switch ($in->diag) {
8 |     case 'Синусит':
9 |         $pot = file_get_contents("../receipts/j01.medic");
10|         break;
11|
12| .....
13|
14| }
15|
16| $potDec = json_decode($pot, true);
17| $preout = $potDec;
18| $out = json_encode($preout);
19| echo $out;
20|
21| ?>
```

Рисунок 3.11 – Лістинг блоку коду формування плану лікування.

Шифрування звернення пацієнта виконується внаслідок ініціювання події, на клієнті, створення запису до амбулаторної карти пацієнта. З'єднання між клієнтом та сервером встановлюється методом POST використовуючи захищений протокол передачі даних HTTPS. В свою чергу після отримання вхідного значення

актуального діагнозу звернення, модуль шифрування звернення приводить вхідне значення до структури об'єкту JSON. Після нормалізації вхідного значення проводиться отримання зліпку файлу бази знань з вмістом шифр – кодів за системами ICPC 2 та МКХ – 10. Зліпок розміщується в змінну, приводиться до структури JSON та індексується для проведення швидкого пошуку значення шифру. Результат формується та розміщується в змінні \$out. Сформована відповідь значення шифру звернення пацієнта структурування у відповідність до об'єкту JSON та відправляється на клієнт раніше встановленим каналом зв'язку. Лістинг блоку коду роботи модулю шифрування звернення наведено на рис. 3.12.

```

1 | <?php
2 | //Serhii Zembitskyi. All rights reserved. 06.10.2020
3 | $data_json = json_encode($_POST);
4 | $in = json_decode($data_json);
5 | $pot = file_get_contents("../receipts/encrypt.medic");
6 | $potDec = json_decode($pot);
7 | $icpc2 = null;
8 | $icd10 = null;
9 |
10| switch ($in->diag) {
11|     case 'Синусит':
12|         $icpc2 = $potDec->icpc2->Синусит;
13|         $icd10 = $potDec->icd10->Синусит;
14|         break;
15|
16| .....
17|
18| }
19|
20| $out = json_encode($encrypt = array(
21|     'icpc2' => $icpc2,
22|     'icd10' => $icd10
23| ));
24| echo $out;
25| ?>

```

Рисунок 3.12 – Лістинг блоку коду шифрування звернення пацієнта.

Результатом виконання завдання розробки програмного рішення є розроблений web застосунок для пришвидшення обробки звернення пацієнтів з респіраторним захворюванням. Даний застосунок отримав назву RSD:ML, що є аббревіатурою respiratory system diagnostic with machine learning в перекладі з

англійської діагностика дихальної системи за допомогою машинного навчання. Застосунок RSD:ML розміщено на хостинг сервері провайдера 000webhost за web адресою [rsdml.000webhostapp.com](http://rsdml.000webhostapp.com) [27]. QR код посилання наведено на рисунку В.1. Також було проведено базову SEO аналітику для виведення застосунку в пошукову систему Google. Застосунок RSD:ML можна знайти в пошуковій системі Google за запитом `rsdml`, результат представлено на рисунку В.2. Для формування навичок користування застосунком у кінцевого користувача було розроблено «Посібник користувача RSD:ML». Даний посібник розміщено у додатку Г. Далі буде проведено тестування застосунку, відібрано заміри витраченого часу проведення та обробки прийому пацієнта з використанням розробленого застосунку RSD:ML та визначено ефективність розробленого оптимізаційного рішення.

### 3.3 Проведення практичної апробації та аналіз ефективності моделей та їх програмної реалізації

Після завершення розробки web застосунку було проведено його тестування. Тестування проводилося з метою визначення швидкості завантаження застосунку на кінцевий пристрій, об'єму завантажуваного застосунку на кінцевий пристрій, визначення швидкості обробки запитів сервером, стабільності роботи застосунку та визначення точності прогнозованого діагнозу. За основу типу тестування було взято модульне тестування (unit testing). В якості інструментарію було обрано інструментарій web розробника від Google. Стабільність застосунку визначалася за статусом виконання запиту на сервері. Також при тестуванні була створена імітація умов низької пропускної здатності мережевого з'єднання. Така імітація проводилася для перевірки стабільності роботи застосунку в користувачів, які підключаються до мережі варіантами відмінними від PON мережі або використовують мобільну мережу для роботи з застосунком. Швидкість мережевого з'єднання (вхідний/вихідний трафік) була встановлена близько 2

Mbit/s. Також було проведено тестування при високо швидкісному підключенню до мережі. Швидкість підключення близько 100 Mbit/s. Логування та результати проведення тестування представлено в таблиці 3.3 та таблиці 3.4. Реєстрація та фіксація подій, які відбувалися, проводилася за допомогою інструменту Performance console від Google. Час рендерингу не обмежувався та повністю залежав від часу і швидкості виконання запитів та обробки подій. При редагуванні плану лікування та підготовці до друку об'єм завантаження не враховувався тому, що дані події виконуються на клієнті і на момент обробки подій потрібна інформація була вже отримана з сервера та прокешована.

Таблиця 3.3 – Результати проведення модульного тестування (unit testing) RSD:ML.

Властивість, швидкість підключення (2/100 Mbit/s)	Статус обробки	Швидкість (ms)	Об'єм завантаження (b/kb/mb)
Завантаження сторінки (2)	200	5500.0 ms	1.4 mb
Завантаження сторінки (100)	200	3520.0 ms	1.4 mb
Профілювання пацієнта (2)	200	697.0 ms	514 b
Профілювання пацієнта (100)	200	187.0 ms	514 b
Модуль «План лікування»			
Визначення діагнозу (2)	200	205.0 ms	234 b
Визначення діагнозу (100)	200	191.0 ms	234 b
Отримання плану лікування (2)	200	735.0 ms	423 b
Отримання плану лікування (100)	200	685.0 ms	423 b
Додати медикамент	200	45.3 ms	-
Видалити медикамент	200	45.0 ms	-
Формування друку плану лікування	200	3962.0 ms	-
Модуль «Запис до амбулаторної карти пацієнта»			
Створити запис (2)	200	190.0 ms	237 b
Створити запис (100)	200	168.0 ms	237 b
Формування друку запису до АК	200	4125.6 ms	-

Таблиця 3.4 – Результати точності прогнозування/класифікації захворювання пацієнта.

Діагноз	Статус обробки	Визначено (+), не визначено (-)
Синусит	200	+
Фарингіт	200	+
Тонзиліт	200	+
Тонзилофарингіт	200	+
Ларингіт	200	+
Трахеїт	200	+
Пневмонія	200	+
Поліпи в носі	200	+
Бронхіт	200	+
Емфізема легені	200	+
Бронхіальна астма легка стадія	200	+
Бронхіальна астма важка стадія	200	+
Абсцес легені	200	+
Легенева недостатність	200	+
Плеврит	200	+

Також було проведено стрес тест на навантаження системи. В стрес тесті прийняло участь 45 сімейних лікарів. Загальна кількість користувачів, які використовували застосунок протягом одного місяця 200 користувачів. Облік та фіксація користувачів проводилася за допомогою умовно - безкоштовного web модулю Visitor Counter від компанії Eflsight Apps. При проведенні стрес тесту відхилень або аномалій в поведінці обробки вхідної інформації та отриманні результату виявлено не було. Хостинг сервер опрацьовував запити стабільно без виникнення HTTP помилок. Кількість учасників стрес тесту та загальну кількість користувачів зображено на рисунку В.4 та рисунку В.3. Проаналізувавши отриману інформацію з проведення модульного тесту можна зробити наступні висновки.

Кінцевий розмір завантаженого додатку близько 1.4 мегабайта, час завантаження від 4 до 6 секунд, що є оптимальним для даної структури та контенту застосунку. За рахунок використання технології AJAX відбувається роздільне завантаження інформації та обробка запитів без перезавантаження сторінки. Таким чином розмір інформації при отриманні результатів, таких як діагноз та план лікування, не перевищує 1 кілобайту. Використовуючи інформацію про статус обробки запитів можна стверджувати, що стабільність роботи застосунку визначена як висока. Всі подані запити на сервер обробилися та повернулася відповідь зі статусом обробки HTTP 200. Швидкість обробки запиту висока, при модуль тестуванні затримок та замороження не виявлено. При відмінності в швидкості мережевого з'єднання швидкість обробки запитів суттєво не відрізнялася. Алгоритм наївного Баєса відпрацьовує стабільно та класифікує вхідний вектор анамнезу звернення у відповідності до клініки захворювань існуючих в базі знань. Таким чином розроблений застосунок RSD:ML відповідає всім вимогам SPA застосунку та оптимізований для використання на мобільних платформах.

Наступним етапом було проведення практичної апробації застосунку в реальних умовах на прийомі пацієнтів з респіраторним захворюванням для визначення ефективності розроблених моделей та програмного рішення пришвидшення прийому та обробки звернення пацієнтів. При проведенні прийому пацієнтів відбувалася фіксація витрати часу кожного етапу обробки звернення пацієнта згідно з формулою (3.1) з використанням та без використання застосунку RSD:ML. Підставивши отримані значення при проведенні практичної апробації в формулу (3.1) та обрахувавши її було отримано наступні результати. При використанні МІС рекомендованих eHealth сімейний лікар витрачає на прийом пацієнта та обробку звернення 42 хвилин, використовуючи web застосунок RSD:ML витрачає 27 хвилин. Після проведення практичної апробації, сімейний лікар який виступав в ролі експерта, надав відгук про використання застосунку RSD:ML в своїй роботі, відгук прикріплено в додатку Д. Результати тестування наведено у таблиці 3.5

Таблиця 3.5 – Результати тестування в реальних умовах з використанням та без використання застосунку RSD:ML.

Етап проведення звернення пацієнта	Витрачено часу без використання RSD:ML (хв)	Витрачено часу з використанням RSD:ML (хв)
Відбір первинної інформації про пацієнта	2	1
Визначення стану самопочуття пацієнта	8	8
Огляд носа пацієнта	1	1
Огляд ротової порожнини пацієнта	1	1
Огляд горла пацієнта	1	1
Визначення характеристик кашлю в пацієнта	1	1
Аускультация дихальної системи пацієнта	10	10
Розробка плану лікування пацієнта	8	3
Фіксація звернення пацієнта	10	1
Всього витрачено часу	42	27

Таким чином спроектувавши математичні та графічні моделі, розробивши застосунок на основі отриманих моделей, провівши тестування та практичну апробацію можна перейти до аналізу ефективності розробленого оптимізаційного рішення. Використовуючи дані наведені у таблиці 3.6 можна виявити, що на етапі формування плану лікування, за допомогою RSD:ML сімейний лікар замість 8 хвилин витрачає 3 хвилини, що на 5 хвилин менше. Пришвидшення обробки даного етапу складає 62.5%. На етапі фіксації звернення пацієнта, сімейний лікар використовуючи RSD:ML витрачає замість 10 хвилин часу лише 1 хвилину. Пришвидшення етапу фіксації звернення пацієнта складає 90.0%. В загальному витрачено часу на проведення та обробку звернення пацієнта з використанням RSD:ML 27 хвилин. Звичними методами сімейний лікар змушений витратити 42 хвилин свого часу на прийом та обробку звернення пацієнта. Витрачений час зменшується на 35.7%. Тому сімейний лікар використовуючи оптимізаційне рішення RSD:ML може зменшити час роботи з пацієнтом на 15 хвилин, що в подальшому призведе до збільшення кількості прийнятих пацієнтів за робочий

день в наслідок чого і збільшення фінансування від НСЗУ. Також використання RSD:ML призводить до структуризації і встановлення єдиного шаблону дій сімейного лікаря при прийомі пацієнта з респіраторним зверненням та автоматизує роботу з формуванням планів лікування і записів до форми №025/о.

Проаналізувавши отримані результати ефективності розробленого оптимізаційного рішення, моделей та web – застосунку можна підтвердити раніше висунуту гіпотезу. Дійсно використання алгоритмів машинного навчання та інформаційних технологій в змозі суттєво зменшити витрату часу на обробку звернення пацієнта з респіраторним захворювання, розробку плану лікування та фіксацію звернення пацієнта до сімейного лікаря в амбулаторну карту пацієнта.

## ВИСНОВКИ

При дослідженні процесу прийому пацієнтів з респіраторним захворюванням було встановлено, що сімейний лікар витрачає, багато часу на цей процес, близько 50 хвилин. Таким чином постає проблематика зменшення витрати часу на обробку звернення пацієнта. Проаналізувавши наявні наукові підходи та рішення, медичні інформаційні системи рекомендовані eHealth, було сформовано наступну картину. Існуючі підходи та рішення вирішують проблему не в повному обсязі та не можуть зменшити час обробки звернення пацієнта. Також деякі з них унеможливають пришвидшити час обробки в наслідок відсутності потрібного функціоналу. В результаті проведення дослідження та його аналізу було сформовано актуальність, мету, об'єкт, предмет, гіпотезу та завдання дослідження. Визначивши проблематику та сформувавши завдання дослідження було проведено аналіз існуючого інструментарію для вирішення поставлених завдань. Проаналізувавши інструментарій було обрано оптимальні підходи та методи до вирішення поставленої задачі. Також було з'ясовано, що оптимізаційне рішення буде представлене у вигляді web застосунку типу SPA. Для обробки вхідного вектору анамнезу використовується алгоритм наївного Баєса.

За результатами виконаних теоретичних та практичних досліджень у дипломній роботі розроблено:

- формулу обрахунку загального часу обробки звернення пацієнта, формула (3.1),
- моделі інтерпретації текстового опису анамнезу в вхідний вектор, формули (3.2) та (3.3),
- модель бази знань, алгоритму машинного навчання, для збереження векторів клініки захворювання та векторів знеособленого анамнезу пацієнтів, формула (3.6),
- графічну модель, у вигляді блок схеми, для збереження інформаційних семплів плану лікування, рисунок 3.1,

- графічну модель оптимізації (пришвидшення) обробки звернення пацієнта з респіраторним захворюванням, рисунок 3.2,
- web застосунок на основі, вище описаних моделей, для проведення практичної апробації та визначення ефективності моделей,
- проведено unit тестування web застосунку, практичну апробацію та визначення ефективності застосунку для пришвидшення обробки звернення пацієнта.

Практична апробація відбувалася в реальних умовах сімейним лікарем на прийомі пацієнтів з респіраторним захворюванням. Проведено стрес тест web застосунку в результаті чого було встановлено, що робота застосунку, при навантаженні, стабільна. Також було отримано позитивний відгук, від сімейного лікаря, про роботу web застосунку. Отримані результати практичної апробації наступні, сімейний лікар використовуючи розроблений web застосунок RSD:ML зменшує час обробки звернення пацієнта на 15 хвилин або на 35%. Таким чином зменшення часу обробки звернення збільшує кількість пацієнтів в наслідок чого виникає фінансова вигода для сімейного лікаря.

Розроблені в дипломній роботі моделі в повному обсязі виконують поставлену задачу та вирішують проблематику зменшення часу обробки звернення. Тому дані моделі можна рекомендувати як основу математико – логічної та функціональної бази для розробки оптимізаційного рішення, а саме медичної інформаційної системи для сімейного лікаря. Розроблений web застосунок RSD:ML можна використовувати як базове рішення для створення МІС, використовуючи вже розроблений функціонал або як модуль до існуючої МІС. Взявши за основу розроблені моделі можна в подальшому проводити наукові дослідження за напрямком використання штучного інтелекту для формування плану лікування та рекомендації направлень до вузькопрофільного спеціаліста. Web застосунок в подальшому можна вдосконалити підтримкою API для забезпечення створення єдиної інтелектуальної САПР сімейного лікаря. Таким чином розроблене в даній дипломній роботі оптимізаційне рішення повністю вирішує поставлену задачу про, що свідчать отримані результати.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Реформа системи охорони здоров'я. URL: <https://www.kmu.gov.ua/diyalnist/reformi/rozvitok-lyudskogo-kapitalu/reforma-sistemi-ohoroni-zdorovya> (дата звернення 05.09.2020)
2. About WONCA. URL: <https://www.globalfamilydoctor.com/AboutWonca/brief.aspx> (дата звернення 05.09.2020)
3. Про Національну службу здоров'я України. URL: <https://nszu.gov.ua/pro-nszu> (дата звернення 05.09.2020)
4. Хто такий сімейний лікар? URL: <https://lrcpmsd.lic.org.ua/novyny/hto-takuj-simejnyj-likar-i-chym-vin-vidriznyayetsya-vid-dilnychogo-terapevta/> (дата звернення 08.09.2020)
5. Велика українська енциклопедія. Анамнез. URL: <https://vue.gov.ua/Анамнез> (дата звернення 08.09.2020)
6. Міністерство охорони здоров'я. Протоколи доказової медицини. URL: <https://moz.gov.ua/article/protocols/onlajn-platforma-z-protokolami-na-zasadah-dokazovoi-medicini> (дата звернення 08.09.2020)
7. Міжнародний класифікатор хвороб. Версія 10. URL: [https://www.wikiwand.com/uk/Міжнародний\\_класифікатор\\_хвороб](https://www.wikiwand.com/uk/Міжнародний_класифікатор_хвороб) (дата звернення 08.09.2020)
8. Класифікація ІСРС 2. Як нею користуватися? URL: <https://www.umj.com.ua/article/122883/klasifikatsiya-isrs-2-yak-neyu-koristuvatisya-materiali-treningu> (дата звернення 11.09.2020)
9. Гострі респіраторні вірусні захворювання. URL: <https://www.umj.com.ua/article/6986/gostri-respiratorni-zaxvoryuvannya-pitannya-klinichnoi-diagnostiki-ta-likuvannya-lekciya> (дата звернення 11.09.2020)

10. Колесник П.О., Рудакова С.О. Кишеньковий довідник сімейного лікаря: видання п'яте, доповнене під редакцією проф. Чопея І.В. / Колесник П.О., Рудакова С.О. – Ужгород, 2019. – 8с. – 70с.

11. Фармацевтична енциклопедія. URL: <https://www.pharmencyclopedia.com.ua/article/2921/auskultaciya> (дата звернення 11.09.2020)

12. Медичні інформаційні системи: огляд можливостей і приклад використання. URL: <https://evergreens.com.ua/ua/articles/medical-information-systems.html> (дата звернення 15.09.2020)

13. eHealth. Електронне здоров'я. URL: <https://ehealth.gov.ua/> (дата звернення 15.09.2020)

14. Всеукраїнська медична газета «Ваше здоров'я». Інна Білевич: Ми витрачаємо більше часу на писанину, аніж на хворого. URL: <https://www.vz.kiev.ua/inna-bilevych-my-vytrachayemo-bilshe-chasu-na-pysanynu-anizh-na-hvorogo/> (дата звернення 15.09.2020)

15. Nevlabs. Діагностика захворювань дихательних шляхів з допомогою нейросети. URL: <https://nevlabs.ru/software-development/portfolio/izomed-sppr/> (дата звернення 15.09.2020)

16. Стаття сапр Сітнікова О. А., Почебут М. В., розробка системи підтримки прийняття рішень для багатопрофільної медичної допомоги./ О. А. Сітнікова, М. В. Почебут // Вісник національного технічного університету «ХПІ». Харків – 2015.

17. Что такое Apache? URL: <https://eternalhost.net/blog/hosting/web-server-apache> (дата звернення 02.10.2020)

18. Что такое PHP? URL: <https://www.php.net/manual/ru/intro-what-is.php> (дата звернення 05.10.2020)

19. Введение в HTML 5. URL: <https://metanit.com/web/html5/1.1.php> (дата звернення 05.10.2020)

20. Что такое SPA приложения? URL: <https://www.instagram.com/p/CHgxK1FnDnk/> (дата звернення 10.10.2020)

21 Введение в JSON. URL: <https://www.json.org/json-ru.html> (дата звернення 10.10.202)

22 Обучение нейросети с учителем, без учителя, с подкреплением — в чем отличие? Какой алгоритм лучше? URL: <https://neurohive.io/ru/osnovy-data-science/obuchenie-s-uchitelem-bez-uchitelja-s-podkrepleniem/> (дата звернення 12.10.2020)

23 Шесть простых шагов для освоения наивного байесовского алгоритма. URL: <http://datareview.info/article/6-prostyih-shagov-dlya-osvoeniya-naivnogo-bayesovskogo-algoritma-s-primerom-koda-na-python/> (дата звернення 12.10.2020)

24 Метод опорных векторов (SVM). URL: [https://neerc.ifmo.ru/wiki/index.php?title=Метод\\_опорных\\_векторов\\_\(SVM\)](https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_(SVM)) (дата звернення 12.10.2020)

25 Алгоритм ближайшего соседа. URL: <https://basegroup.ru/community/articles/knn> (дата звернення 12.10.2020)

26 Зембіцький С. П. Проектування архітектури хмарного обчислювального середовища для обробки анамнезу пацієнтів із захворюванням дихальної системи на основі алгоритмів машинного навчання / С. П. Зембіцький, Н. В. Грипинська // Вісник Хмельницького національного університету. Технічні науки. – 2020. – № 3. – С. 88-91.

27 Діагностика респіраторних захворювань RSD:ML. URL: <https://rsdml.000webhostapp.com/> (дата звернення 02.11.2020)

**ДОДАТОК А**  
**(ОБОВ'ЯЗКОВИЙ)**  
**ТАБЛИЧНО - ВЕКТОРНА ІНФОРМАЦІЯ ФАКТОРІВ ВПЛИВУ ТА**  
**КЛІНІКИ ЗАХВОРЮВАНЬ**

Таблиця А.1 – Зведена таблиця факторів впливу

Фактор впливу	Порядковий номер	Результуюче значення
Втомлюваність	1	1
Погіршений загальний стан	2	6
Слабкість	3	9
Ломота в суглобах	4	12
Поганий сон	5	15
Гарячка	6	18
Озноб	7	21
Потовиділення	8	24
Головний біль	9	27
Закладеність вух	10	30
Біль у грудях	11	33
Сірий відтінок шкіри	12	36
Температура	350 - 420	350 – 420
Порушення нюху	13	39
Знижене відчуття запахів	14	42
Важке носове дихання	15	45
Виділення з носу	16	48
Виділення слизу та гною	17	51
Неприємний запах з рота	18	54

Продовження таблиці А.1

Зубний біль	19	57
Виділення мокротиння з кров'ю	20	60
Збільшені лімфатичні вузли	21	63
Почервоніння горла	22	66
Біль при ковтанні	23	69
Біль у горлі	24	72
Біль у глотці	25	75
Хриплість голосу	26	78
Гній на мигдаликах	27	81
Запалення мигдаликів	28	84
Виразка горла	29	87
Відсутній кашель	30	90
Присутній кашель	31	93
Покашлювання	32	96
Сухий кашель	33	99
Вологий кашель	34	102
Присутнє виділення гнійного мокротиння	35	105
Порушення носового дихання	36	108
Затруднене дихання	37	111
Ортопное	38	114
Задишка	39	117
Приступи задихання	40	120
Збільшена частота дихання	41	123
Важкість дихання	42	126
Послаблене дихання	43	129
Гостра дихальна недостатність	44	132
Відчуття нехватки кисню	45	135
Зменшена глибина дихання	46	138

Кінець таблиці А.1

Хрипи	47	141
Вологі хрипи	48	144
Хрипи нижче лопаток	49	147
Свистяче дихання	50	150
Прослуховується шум	51	153

Таблиця А.2 – База знань векторів клініки респіраторних захворювань

Самовідчуття	Температура	Ніс	Рот	Кашель	Горло	Дихання	Діагноз
$t_2$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	r
0	0	0	0	0	0	0	0
78	375	90	111	69	0	108	831
0	370	51	0	0	363	0	784
60	373	0	63	0	90	0	586
18	380	0	0	66	195	0	659
6	374	0	0	153	180	111	824
0	373	0	0	0	366	0	739
39	370	0	54	81	0	270	814
0	366	0	0	75	0	396	837
0	366	0	0	75	0	648	1089
36	366	0	0	0	0	513	915
0	366	0	0	72	0	642	1080
39	380	0	0	69	0	273	761
99	400	0	0	156	0	387	1042
72	366	42	0	0	0	108	588
27	380	0	0	78	90	267	842

**ДОДАТОК Б**  
**(ОБОВ'ЯЗКОВИЙ)**  
**SCREENSHOTS МОДУЛІВ WEB ЗАСТОСУНКУ**

Q +380681111111 Кравченко Марта Олексіївна ▼

Рисунок Б.1 – Модуль пошуку пацієнта.

👤 Загальна інформація

Прізвище	Кравченко
Ім'я	Марта
По батькові	Олексіївна
Дата народження	15.06.1992
Вік	28 років
Стать	Жіноча
Контактний номер	+380681111111
Протипоказання	Відсутні
Країна	УКРАЇНА
Індекс	29000
Область	ХМЕЛЬНИЦЬКА
Район	ХМЕЛЬНИЦЬКИЙ
Населений пункт	М.ХМЕЛЬНИЦЬКИЙ
Вулиця	ПР. МИРУ
Буд. корп. кв.	2А

Рисунок Б.2 – Модуль відображення інформації про пацієнта.



## Самопочуття

---

- Втомлюваність
- Погіршений загальний стан
- Слабкість
- Ломота в суглобах
- Поганий сон
- Гарячка
- Озноб
- Потовиділення
- Головний біль
- Закладеність вух
- Біль у грудях
- Сірий відтінок шкіри

Температура

**37,5**

---

Рисунок Б.3 – Модуль відбору анамнезу, самопочуття.

## Огляд носа

- Порушення нюху
- Знижене відчуття запахів
- Важке носове дихання
- Виділення з носа
- Виділення слизу та гною

Рисунок Б.4 – Модуль відбору анамнезу, огляд носа.

## Огляд ротової порожнини


- Неприємний запах з рота
- Зубний біль
- Виділення мокротиння з кров'ю
- Збільшені лімфатичні вузли

Рисунок Б.5 – Модуль відбору анамнезу, огляд ротової порожнини.

## Огляд горла

- Почервоніння горла
- Біль при ковтанні
- Біль у горлі
- Біль у глотці
- Хриплість голосу
- Гній на мигдаликах
- Запалення мигдаликів
- Виразка горла

Рисунок Б.6 – Модуль відбору анамнезу, огляд горла.

 **Кашель**

- Відсутній кашель
- Присутній кашель
- Покашлювання
- Сухий кашель
- Вологий кашель
- Присутнє виділення гнійного мокротиння

Рисунок Б.7 – Модуль відбору анамнезу, кашель.

**Аускультация дихальної системи**

- Порушення носового дихання
- Затруднене дихання
- Ортопноє
- Задишка
- Приступи задихання

Рисунок Б.8 – Модуль відбору анамнезу, аускультация дихальної системи.

Діагноз

Синусит

Назва ліків	Доза	Разів на день	Днів прийому	Виконати
Левофлоксацин	750 мг	1р/д	5 днів	ВИДАЛИТИ
Доксициклін	100 мг	2р/д	5 днів	ВИДАЛИТИ
Цефподоксим	200 мг	2р/д	5 днів	ВИДАЛИТИ
Амосицилін / Клавуланс	1000 мг	2р/д	5 днів	ВИДАЛИТИ

Рисунок Б.9 – Модуль автоматизованого формування плану лікування.



## Запис №1 від 05.11.2020 15.49

Клініка епізоду: Втомлюваність, Гарячка, Головний біль, Порушення нюху, Виділення слизу та гною, Неприємний запах з рота, Біль у горлі, Присутній кашель, Порушення носового дихання

Температура: 37.5

Примітки до епізоду:

Діагноз: Синусит

План лікування: Левофлоксацин, 750 мг, 1р/д, 5 днів, Доксициклін, 100 мг, 2р/д, 5 днів, Цефподоксим, 200 мг, 2р/д, 5 днів, Амосицилін / Клавуланат, 1000 мг, 2р/д, 5 днів,

ICPC 2: R75  
МКХ - 10: J01

Рисунок Б.10 – Модуль автоматизованого формування запису до амбулаторної карти пацієнта.

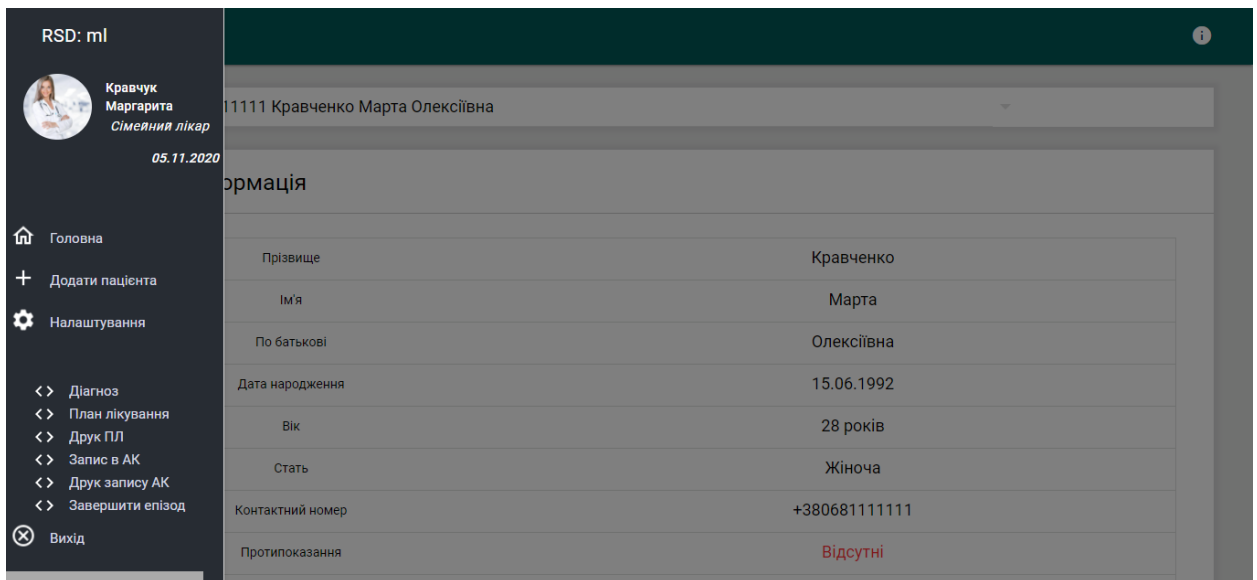


Рисунок Б.11 – Модуль side menu та header.



Respiratory system diagnosis machine learning (далі RSD:ML, сервіс) - web додаток створений для прогнозування, діагностування захворювань дихальної системи та розробки плану лікування за допомогою методів машинного навчання. Даний сервіс являється результатом практичної апробації науково - дослідної роботи. Сервіс розроблений для прогнозування та діагностування наступних захворювань дихальної системи: синусит, фарингіт, тонзиліт, тонзилофарингіт, ларингіт, трахеїт, пневмонія, поліпи в носі, бронхіт, відтік легені, бронхіальна астма (легка, важка стадія), абсцес легені, дихальна недостатність, плеврит.

Персональна інформація представлена для демонстрації роботи сервісу та не стосуються реальних людей.

Розробник RSD:ML не несе відповідальності за дії кінцевого користувача.

Посібник користувача розміщено за наступним [посиланням](#)

© Зембіцький Сергій. 2020.

Рисунок Б.12 – Модальна панель для відображення додаткової інформації.

10.11.2020	Діагностика респіраторних захворювань RSDML		
Левофлоксацин 750 мг		1р/д	5 днів
Доксициклін 100 мг		2р/д	5 днів
Цефподоксим 200 мг		2р/д	5 днів
Амосицилін / Клавуланат 1000 мг		2р/д	5 днів

<https://rsdml.000webhostapp.com> 1/1

Рисунок Б.13 – Приклад друкованого плану лікування.

10.11.2020 Діагностика респіраторних захворювань RSDML

Запис №1 від 10.11.2020 20:47

Клініка епізоду: Втомиваність, Гарячка, Головний біль, Порушення нюху, Виділення слизу та гною, Неприємний запах з рота, Біль у горлі, Присутній кашель, Порушення носового дихання  
Температура: 37.5  
Привідки до епізоду:  
Діагноз: Синусит  
План лікування: Левофлоксацин, 750 мг, 1р/д, 5 днів, Доксциклін, 100 мг, 2р/д, 5 днів, Цефподоксим, 200 мг, 2р/д, 5 днів, Амосицилін / Клавуланат, 1000 мг, 2р/д, 5 днів,,

ICPC 2: R75  
МКХ - 10: J01

<https://rsdml.000webhostapp.com> 1/1

Рисунок Б.14 — Приклад друкованого запису до амбулаторної карти пацієнта

**ДОДАТОК В**  
**(ОБОВ'ЯЗКОВИЙ)**  
**ПРАКТИЧНА АПРОБАЦІЯ**



Рисунок В.1 – QR код доступу до web застосунку.

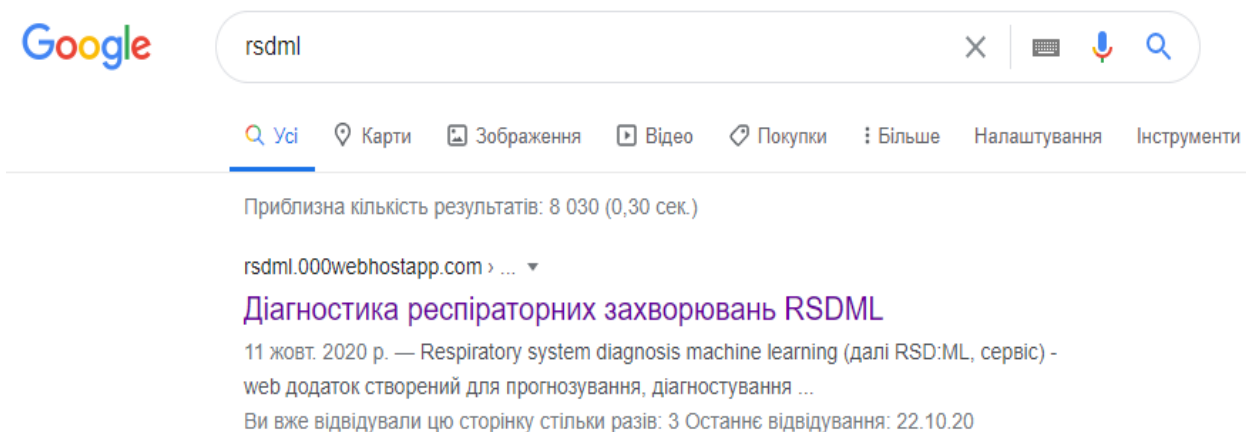


Рисунок В.2 – Результат проведення SEO аналітики.

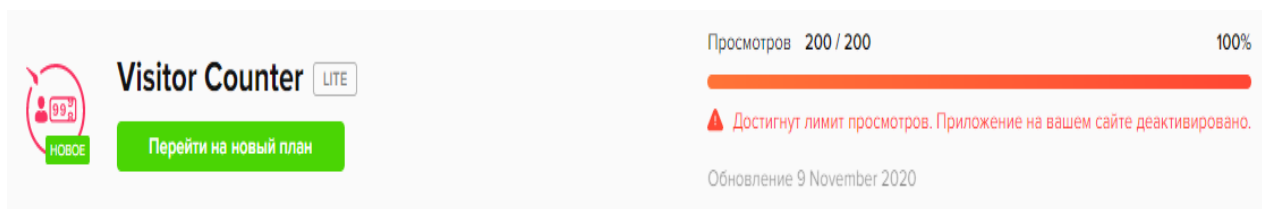


Рисунок В.3 – Кількість користувачів web застосунком за місяць.

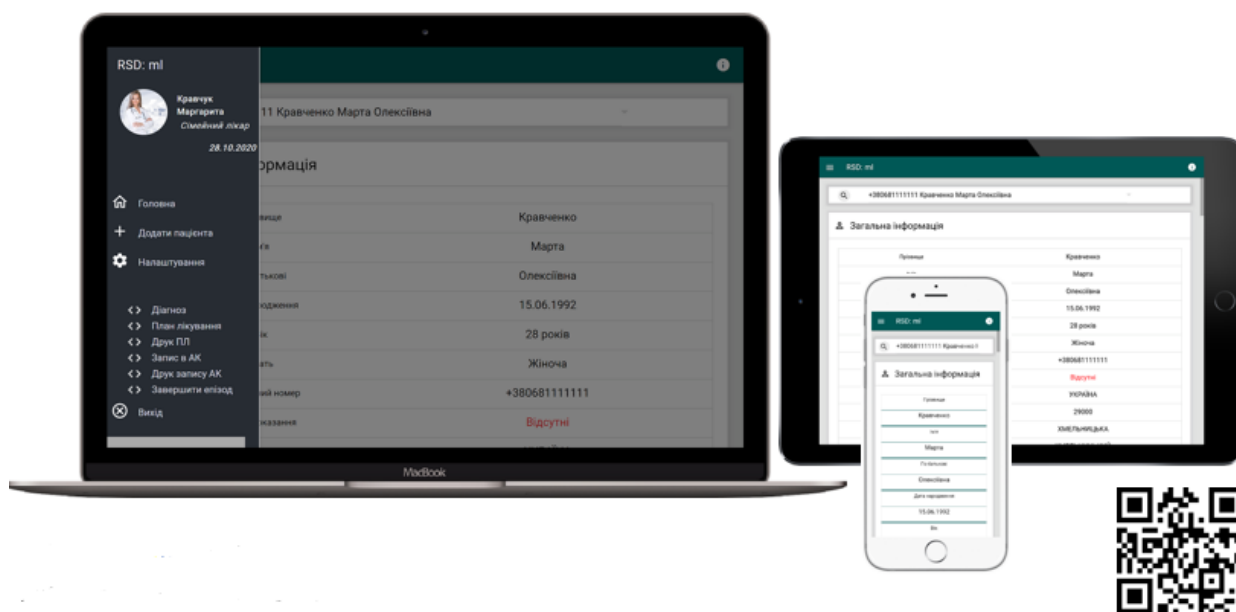
The screenshot displays a medical software interface. On the left is a dark sidebar with a user profile for 'Кравчук Маргарита Сімейний лікар' (Family Doctor) and a date '26.10.2020'. The sidebar contains navigation options: 'Головна', 'Додати пацієнта', 'Налаштування', and a list of actions: 'Діагноз', 'План лікування', 'Друк ПЛ', 'Запис в АК', 'Друк запису АК', and 'Завершити епізод'. The main content area shows a patient's name 'RSD: ml' and a list of symptoms: 'Порушення нюху, Виділення слизу та гною, Неприємний запах з роту, Біль у горлі, Покашлювання, Порушення носового'. Below this, a treatment plan is listed: 'вофлоксацин, 750 мг, 1р/д, 5 днів, Доксциклін, 100 мг, 2р/д, 5 днів, Цефподоксим, 200 мг, 2р/д, 5 днів, зуланат, 1000 мг, 2р/д, 5 днів'. At the bottom right, there is a date 'СЬОГОДНІ 45' and a 'Free Visitor Counter widget' notification.

Рисунок В.4. – Результати стрес тесту.

ДОДАТОК Г  
(ОБОВ'ЯЗКОВИЙ)  
ПОСІБНИК КОРИСТУВАЧА RSD:ML

# ПОСІБНИК КОРИСТУВАЧА RSD:ML

:швидкий старт



Respiratory system diagnosis machine learning (далі RSD:ML, сервіс)

- web додаток створений для прогнозування, діагностування захворювань дихальної системи та розробки плану лікування за допомогою штучного інтелекту. Даний сервіс є результатом практичної апробації науково - дослідної роботи. Сервіс розроблений для прогнозування та діагностування наступних захворювань дихальної системи:

- + синусит,
- + фарингіт,
- + тонзиліт,
- + тонзилофарингіт,
- + ларингіт,
- + трахеїт,
- + пневмонія,
- + поліпи в носі,
- + бронхіт,
- + емфізема легені,
- + бронхіальна астма (легка, важка стадія),
- + абсцес легені,
- + легенева недостатність,
- + плеврит.

Знайдіть пацієнта в пошуковому рядку:


+380681111111 Кравченко Марта Олексіївна

Натисніть  для відображення детальної інформації про пацієнта:

👤 Загальна інформація	
Прізвище	Кравченко
Ім'я	Марта
По батькові	Олексіївна
Дата народження	15.06.1992
Вік	28 років
Стать	Жіноча
Контактний номер	+380681111111
Протипоказання	Відсутні

Проводьте відбір анамнезу та обрийте прапорці на сторінці одночасно (для пришвидшення відбору анамнезу рекомендовано дотримуватися встановленого порядку категорій анамнезу):

Самопочуття



### Самопочуття


- Втомлюваність
- Погіршений загальний стан
- Слабкість
- Ломота в суглобах
- Поганий сон
- Гарячка
- Озноб
- Потовиділення
- Головний біль
- Закладеність вух
- Біль у грудях
- Сірий відтінок шкіри

## Температура

Температура  
37,5|

---


## Огляд носа

 **Огляд носа**

---

- Порушення нюху**
- Знижене відчуття запахів
- Важке носове дихання
- Виділення з носу
- Виділення слизу та гною**


## Огляд ротової порожнини

 **Огляд ротової порожнини**

---

- Неприємний запах з роту**
- Зубний біль
- Виділення мокротиння з кров'ю
- Збільшені лімфатичні вузли

## Огляд горла




## Огляд горла

---

- Почервоніння горла
- Біль при ковтанні
- Біль у горлі
- Біль у глотці

## Визначення характеристик кашлю




## Кашель

---

- Відсутній
- Присутній
- Покашлювання

## Результати аускультатії дихальної системи



## Аускультатія дихальної системи

---

- Порушення носового дихання
- Затруднене дихання
- Ортопноє
- Задишка
- Приступи задихання
- Збільшена частота дихання

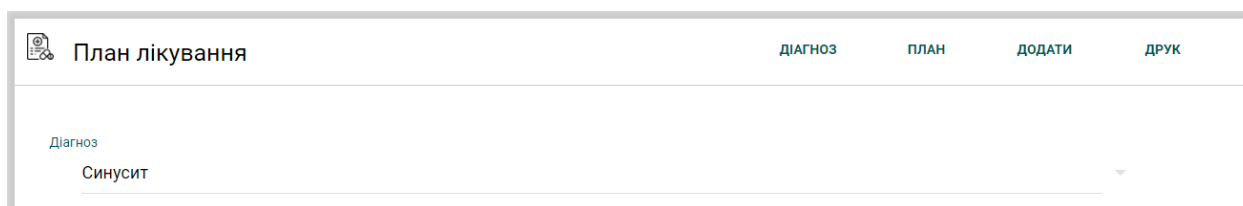
За потреби додайте примітки до анамнезу

Примітки до епізоду

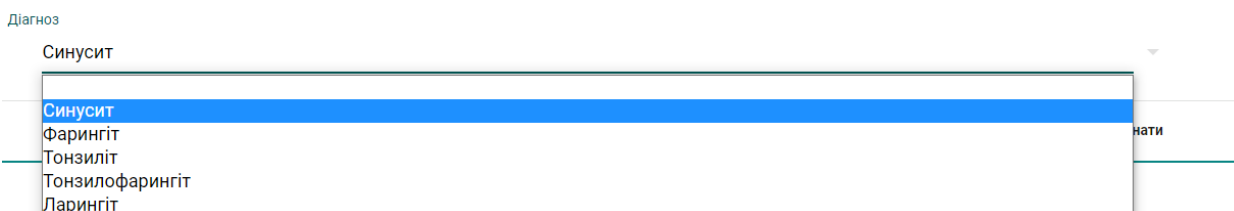
|

**ДІАГНОЗ**

Натисніть **ДІАГНОЗ** для прогнозування діагнозу (система використовуючи штучний інтелект сама встановить діагноз):



Якщо діагноз відрізняється від, вірного на Вашу думку, змініть його натиснувши на назву діагнозу та з випадного списку оберіть вірний (штучний інтелект проведе перенавчання):



**ПЛАН**

Встановивши діагноз натисніть **ПЛАН** для автоматизованого формування плану лікування:

Назва ліків	Доза	Разів на день	Днів прийому	Виконати
Левофлоксацин	750 мг	1р/д	5 днів	<b>ВИДАЛИТИ</b>
Доксициклін	100 мг	2р/д	5 днів	<b>ВИДАЛИТИ</b>
Цефподоксим	200 мг	2р/д	5 днів	<b>ВИДАЛИТИ</b>
Амосицилін / Клавулат	1000 мг	2р/д	5 днів	<b>ВИДАЛИТИ</b>

При потребі можна провести редагування плану лікування:

для видалення натисніть **ВИДАЛИТИ**,

для редагування таблиці встановіть курсор в текстове поле яке потрібно

змінити

Левифлоксацін

для створення нового рядка таблиці натисніть

**ДОДАТИ**

Назва ліків	Доза	Разів на день	Днів прийому	Виконати
Левифлоксацін	750 мг	1р/д	5 днів	<b>ВИДАЛИТИ</b>
Доксициклін	100 мг	2р/д	5 днів	<b>ВИДАЛИТИ</b>
Цефподоксим	200 мг	2р/д	5 днів	<b>ВИДАЛИТИ</b>
Амосикцилін / Клавулат	1000 мг	2р/д	5 днів	<b>ВИДАЛИТИ</b>
				<b>ВИДАЛИТИ</b>

Натисніть

**ДРУК**

для друку сформованого плану лікування:

31.10.2020

Діагностика респіраторних захворювань RSDML

Левифлоксацін	750 мг	1р/д	5 днів
Доксициклін	100 мг	2р/д	5 днів
Цефподоксим	200 мг	2р/д	5 днів
Амосикцилін / Клавулат	1000 мг	2р/д	5 днів

Передайте роздрукований план лікування пацієнту

Натисніть **СТВОРИТИ** для формування запису до амбулаторної карти пацієнта:

**Запис №1 від 31.10.2020 20.18**

Клініка епізоду: Втомлюваність, Гарячка, Головний біль, Порушення нюху, Виділення слизу та гною, Неприємний запах з роту, Біль у горлі, Присутній, Порушення носового дихання  
Температура: 37  
Примітки до епізоду:  
Діагноз: Синусит  
План лікування: Левофлоксацин, 750 мг, 1р/д, 5 днів, Доксциклін, 100 мг, 2р/д, 5 днів, Цефподоксим, 200 мг, 2р/д, 5 днів, Амосицилін / Клавуланат, 1000 мг, 2р/д, 5 днів,

ICPC 2: R75  
МКХ - 10: J01

Натисніть **СТВОРИТИ** **ДРУК** **ЗАВЕРШИТИ** для друку запису в амбулаторну карту пацієнта:

31.10.2020      Діагностика респіраторних захворювань RSDML

**Запис №1 від 31.10.2020 20.18**

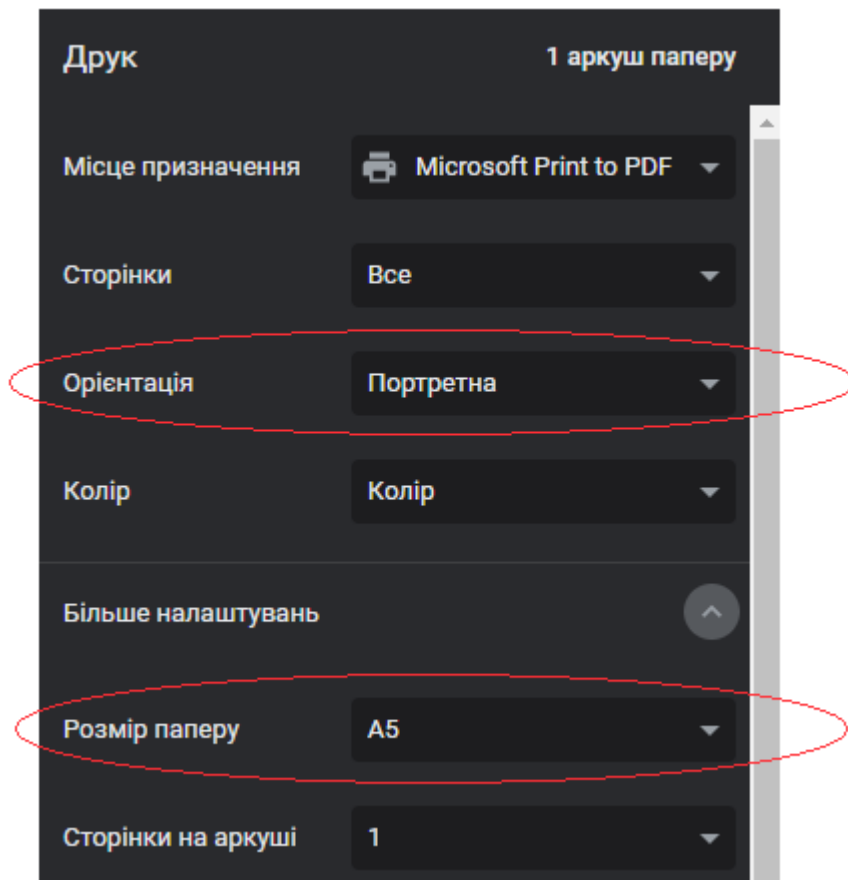
Клініка епізоду: Втомлюваність, Гарячка, Головний біль, Порушення нюху, Виділення слизу та гною, Неприємний запах з роту, Біль у горлі, Присутній, Порушення носового дихання  
Температура: 37  
Примітки до епізоду:  
Діагноз: Синусит  
План лікування: Левофлоксацин, 750 мг, 1р/д, 5 днів, Доксциклін, 100 мг, 2р/д, 5 днів, Цефподоксим, 200 мг, 2р/д, 5 днів, Амосицилін / Клавуланат, 1000 мг, 2р/д, 5 днів,,

ICPC 2: R75  
МКХ - 10: J01

Для завершення епізоду натисніть

**ЗАВЕРШИТИ**

Для друку документації згідно з встановленими стандартами оберіть, в модулі друку браузера, портретну орієнтацію сторінки та розмір паперу А5.



### УВАГА!

Персональна інформація представлена для демонстрації роботи сервісу та не стосуються реальних людей.

Розробник RSD:ML не несе відповідальності за дії кінцевого користувача.

**ДОДАТОК Д**  
**(ООБОВ'ЯЗКОВИЙ)**  
**ВІДГУК СІМЕЙНОГО ЛІКАРЯ**

Відгук про web додаток RSD:ml.

Я Козяр О.В. власник та засновник кабінету медичної практики сімейної медицини, сімейний лікар. Я використовую в своїй роботі web додаток RSD:ml розроблений Зембіцьким С.П. Додаток зручний у роботі, має приємний зовнішній вигляд. Інформація подається в структурованому вигляді, що призводить до полегшення роботи з додатком. Користуючись даним додатком можна привести до єдиного шаблону свої дії при роботі з пацієнтом. Після введення анамнезу пацієнта сервіс сам створює прогноз діагнозу. Також є можливість формування плану лікування та запису звернення пацієнта в автоматичному режимі. Присутня можливість зміни діагнозу звернення пацієнта та редагування плану лікування таким чином відбувається збереження «людського фактору» тому, що машини теж можуть помилятися. Повну автоматизацію обробки звернення приносить можливість друку документації такої як план лікування пацієнта та запис до амбулаторної карти форма №025/о. Також даним додатком можна користуватися з різних пристроїв, в кабінеті я можу використовувати ноутбук, на виїзді можна використовувати смартфон.

Таким чином я як сімейний лікар використовуючи даний додаток не витрачаю час на написання паперової документації. Діагноз звернення пацієнта встановлюється автоматично, згідно з введеним анамнезом, зразу можна сформуванати та роздрукувати для подальшого прикріплення до амбулаторної карти пацієнта запис до амбулаторної карти та план лікування для видачі пацієнту. Тому я набагато менше витрачаю часу на обробку звернення пацієнта використовуючи даний додаток та можу прийняти більше пацієнтів за робочий день. На мою думку даний додаток можна рекомендувати для користування також іншим сімейним лікарям.








Рисунок Д.1 — Скан копія відгуку, про застосунок RSD:ML, від сімейного лікаря

## ДОДАТОК Е (ОБОВ'ЯЗКОВИЙ)

НАУКОВА СТАТТЯ. ВІСНИК ХНУ. ТЕХНІЧНІ НАУКИ.

Technical sciences

ISSN 2307-5732

УДК 004.738.5

DOI 10.31891/2307-5732-2020-285-3-13

С. П. ЗЕМБІЦЬКИЙ, Н. В. ГРИПІНСЬКА  
Хмельницький національний університет

### ПРОЕКТУВАННЯ АРХІТЕКТУРИ ХМАРНОГО ОБЧИСЛЮВАЛЬНОГО СЕРЕДОВИЩА ДЛЯ ОБРОБКИ АНАМНЕЗУ ПАЦІЄНТІВ ІЗ ЗАХВОРЮВАННЯМ ДИХАЛЬНОЇ СИСТЕМИ НА ОСНОВІ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

*У статті представлено проектування архітектури хмарного середовища для обробки анамнезу захворювання дихальної системи та подальшого прогнозування діагнозу. Під час проведення наукової роботи було опрацьовано попередні публікації та наукові дослідження в галузі сімейної медицини. Проаналізувавши попередні дослідження було сформовано основне невирішене питання, яке полягає в розробці архітектури обробки епізоду звернення пацієнта з використанням методів машинного навчання та з подальшим розгортанням на серверах - хостерах з використанням хмарної обчислювальної технології. Також було сформовано основні цілі проведення даного наукового дослідження. Розглянуто основні характеристики для побудови архітектури та обрано їх оптимальні значення. Обґрунтовано доцільність використання технології передачі даних AJAX. Також було обґрунтовано доцільність використання баз знань формату csv та структурованого об'єкту JSON. Розглянуто алгоритми класифікації з машинним навчанням та обрано оптимальний алгоритм класифікації. Результатом дослідження є розроблена архітектура, в вигляді блок – схеми, функціонування хмарного обчислювального середовища.*

*Ключові слова: алгоритми класифікації, машинне навчання, обробка анамнезу*

S. ZEMBITSKYI, N. HRYPYNSKA  
Khmelnytskyi National University

### CREATING OF ARCHITECTURE OF CLOUD COMPUTING ENVIRONMENT FOR ANALYZING ANAMNESIS OF PATIENTS WITH RESPIRATORY SYSTEM DISEASE ON BASE MACHINE LEARNING

*Algorithms of machine learning are widely used in small galuzu, did not omit medical practice. The machine of modernization in the medical field has known the storage in different directions, such as the processing of biomedical images, the analysis of computer tomography, the analysis of electrocardiograms, the development of licenses, the pathology, the detection. Algorithms of machine learning are also used for the administration of the plant, as it is not necessary for the practical practice, but for the medical galusa. For example, private clientele vikoristovuyut systems and piece neural framing for the establishment and processing of routing of patients in the clinic, keruvannya chergami in the clinic, the establishment of interactive knowledge bases for detailed information to the medical staff. But it's a pity in the family of medicine, for the present moment, it is not possible to see in the implementation of algorithms of machine technology. In the out-of-town medical practice of testing systems of piece neural fences, to optimize the working process of the family doctor, to change the vitrate for an hour at the time of the patient's reception, and to set up the backward pattern of anamnesis of the patient's anamnesis. Therefore, the main task of this scientific advancement is to develop the architecture of a gloomy numerical center for reviewing the anamnesis of patients with mental illness on the basis of machine learning algorithms, so that we can give the values of the robotic field of recognition.*

*This work present process of developing architecture cloud processing environment for analyzing anamnesis of patients with respiratory system disease. During doing this scientific work was processed previous publications and scientific experiments in family medicine direction. Having analyzed previous experiments was created main problem this scientific work. Main problem is creating of architecture processing episode patient treatment with using machine learning and future integration on servers - hosters with cloud processing technologies. Also was created main targets this scientific work. Reviewed main features for creating of architecture and was selected them optimal correct values. Justified expediency using technologic of transmit data such as AJAX. Also was justified expediency using knowledge database that using csv format and using structured object such as JSON. Reviewed machine learning algorithms of classifications and was selected optimal algorithm. Result this experiment is created architecture in view block – schema, function cloud processing environment.*

*Keywords: algorithm classification, anamnesis processing, machine learning*

**Постановка задачі.** Алгоритми машинного навчання широко використовуються в різних галузях, не оминуло це і медичну практику. Машинне навчання в медичній галузі знайшло застосування в різних напрямках, таких як обробка біомедичних зображень, аналіз комп'ютерної томографії, аналіз електрокардіограми, розробка ліків, радіологія, виявлення патологій. Також алгоритми машинного навчання застосовуються для виконання завдань, які не стосуються лікувальної практики, але відносяться до медичної галузі. Наприклад, приватні клініки використовують системи штучних нейронних мереж для створення та обробки маршрутизації пацієнтів в клініці, керування чергами в клініці, створення інтерактивних баз знань для вдосконалення досвіду медичного персоналу. Але на жаль в сімейній медицині на даний момент не відбувається впровадження алгоритмів машинного навчання. В загальній медичній практиці використання систем штучних нейронних мереж створить оптимізацію робочого процесу сімейного лікаря, зменшить витрату часу на прийом пацієнта та обробку його анамнезу захворювання, встановить загальний шаблон алгоритму та порядок дій проведення прийому пацієнта та обробки анамнезу. Тому основною задачею даного наукового дослідження являється розробка архітектури хмарного обчислювального середовища для обробки анамнезу пацієнтів з захворюванням дихальної системи на основі алгоритмів машинного навчання, що надасть значні полегшення в роботі сімейного лікаря.

**Аналіз досліджень та публікацій.** На даний момент наукова спільнота проводить дослідження та розробки в сфері діагностики респіраторних захворювань за допомогою алгоритмів машинного навчання. Одне із таких досліджень викладене в дисертації Порєвої Ганни Сергіївни «Методи аналізу звуків легень

для оцінки стану дихальної системи людини». Дана дисертаційна робота присвячена актуальній темі обробки та аналізу звуків легень людини з метою отримання діагностично цінних параметрів для використання їх як в якості самостійних критеріїв оцінювання стану дихальної системи людини і як вхідних аргументів класифікаторів для автоматизації прийняття рішень щодо певних захворювань [1]. Проводиться розробка нових структур баз знань та модифікації алгоритмів машинного навчання [2]. Також проводяться розробки в області аналізу біомедичних зображень, а саме аналіз флюорографічних знімків легень та виявлення аномалій в знімках.

**Виділення невирішених частин.** Проаналізувавши існуючі наукові дослідження та публікації, можна зробити висновок, що основною невирішеною проблемою є відсутність реалізації алгоритмів машинного навчання для аналізу анамнезу пацієнта, розгорнутих в хмарному обчислювальному середовищі. Саме використання систем машинного навчання в хмарних технологіях надасть можливість кінцевому користувачу використовувати менш ресурсоємні пристрої за рахунок проведення всіх обчислень на стороні сервера та відправки опрацьованого пулу даних, клієнту. Також середовище хмарного обчислення допоможе в проектуванні кросплатформового застосунку, що збільшить кількість потенційних користувачів, за рахунок виконання додатка на web платформі, та зменшить витрати ресурсів на розробку кінцевого додатка.

**Формулювання цілей.** В ході проведення даного наукового дослідження було сформульовано наступні цілі:

- ознайомитися з типами хостингових серверів та визначити оптимальний тип та конфігурацію сервера;
- визначити тип web застосунку;
- визначити метод «спілкування» клієнта з сервером;
- визначити тип баз даних для збереження карт навчання та іншої інформації, яка забезпечує функціонування середовища;
- визначити алгоритм машинного навчання для обробки та прогнозування або класифікації діагнозу відповідно до відібраного анамнезу;
- розробити архітектуру функціонування хмарного обчислювального середовища.

**Виклад основного матеріалу.** Для реалізації хмарної архітектури першим основним кроком було визначення типу та конфігурації сервера хостингу. Існують наступні типи web хостингів:

- віртуальний хостинг (Shared hosting) розширений тип хостингу, який задовольняє більшість користувачів та власників невеликих сайтів. Плюсами даного типу сервера є простота в використанні та наявність універсальної панелі керування. Недоліками є відсутність доступу до сервера по SSH протоколу для встановлення розширень та налаштування сервера, невеликий об'єм виділеного місця, кількісне обмеження створення таблиць та баз даних, присутність «сусідів»;
- віртуальний виділений сервер (Virtual dedicated server) виділена частина сервера, яка містить для кожного клієнта окрему віртуальну машину. Використовується для середніх проектів. Плюсами даного типу сервера є можливість встановлення потрібної операційної системи та програмного забезпечення для функціонування сервісу, конфігурування обчислювальних ресурсів та наявність доступу до сервера по SSH протоколу. Основним недоліком є присутність «сусідів»;
- виділений сервер (Dedicated server) окремий обчислювальний пристрій з окремим виділеним каналом зв'язку. Використовується для великих проектів. Наявний повний контроль над програмною частиною сервера з повним доступом до всіх налаштувань сервера. Апаратну частину адмініструє хостинг-провайдер;
- колокейшн (Colocation) надання фізичного місця та виділеного каналу зв'язку в дата центрі хостинг – провайдера;
- хмарний сервер (Cloud hosting) віртуальний виділений сервер з гнучкою системою розподілення обчислювального ресурсу для виконання задач.

Таким чином для реалізації архітектури хмарного обчислення з використанням алгоритмів машинного навчання оптимальним варіантом сервера є віртуальний виділений сервер, виділений сервер або хмарний сервер.

Тип web застосунку – SPA (Single Page Application) односторінковий web додаток. Порівняно з звичайним багатосторінковим web додатком SPA оптимізований для зменшення використання інтернет трафіку за рахунок разового завантаження та кешування файлів ресурсів на пристрій кінцевого користувача, таким чином відбувається обмін лише інформацією для обробки між клієнтом та сервером в вигляді пулу даних. Також використовуючи дану технологію можна використовувати методику шаблонізації, що значно зменшить об'єм кінцевого web додатку.

Спілкування з сервером відбувається за методикою AJAX, що дозволить кінцевому користувачу працювати без примусового перезавантаження web сторінки при відправці та отриманні пулу даних. Також використання даної методики роботи з сервером зменшує використання інтернет трафіку та зменшує навантаження на сервер.

База знань організована у вигляді структурованих векторів клініки респіраторних захворювань та анамнезу пацієнтів. Об'єкт бази знань представлений в вигляді файлу з типом розширення “.csv”. Порівняно з звичайною базою даних MySQL організація структури бази знань в файлі з типом розширення “.csv” дає змогу системі алгоритму машинного навчання з мінімальною затримкою на вибірку інформації отримати повний зліпок бази знань для чергового перенавчання, побудови нової моделі навчання та створення

класифікації або прогнозу відповідно до нової моделі. Також використовуючи об'єкт файлу з типом розширення ".csv" для організації бази знань з легкістю можна вносити зміни в структуру самої бази знань не руйнуючи вже існуючу базу знань. Плани лікування та шифри ІСРС2, МКХ – 10 потребують організації складної структури даних, тому для реалізації даної потреби було обрано формат JSON. Використовуючи даний формат можна сформувати складну модель збереження та представлення даних та інтерпретувати дані в об'єкти, що допоможе швидко та легко отримати доступ до об'єкту даних та отримати значення обраного об'єкту.

Для реалізації задачі класифікації було обрано наступні алгоритми машинного навчання: алгоритм найвнього Баєса (далі АНБ), метод опорних векторів, метод К-ближніх сусідів. Алгоритм найвнього Баєса – це ймовірнісний алгоритм класифікації в основі якого лежить теорема Баєса про наївні припущення незалежності змінних. Цей алгоритм є мультикласифікатором і здатний опрацювати великі об'єми даних. Основною вимогою даного алгоритму являється деталізована база знань, яка будується на основі статистичних вибірок, в деяких випадках ця вимога вважається недоліком через неможливість відбору великої статистичної вибірки. На відміну від алгоритму найвнього Баєса метод опорних векторів є бінарним класифікатором. Задача класифікації методом опорних векторів полягає в відшуванні гіперплощини в  $n$ -вимірному просторі для розділення об'єктів навчальної вибірки представлених в вигляді точок на два класи та побудові апроксимації вхідного вектору для відшування приналежності до першого або другого класу. Через потребу мультикласифікації метод опорних векторів як самостійний окремий алгоритм використовувати неможливо тому для створення мультикласифікатора на основі даного методу використовується модель – надбудова «Один проти всіх» або «Один проти одного». Для методу опорних векторів наявність великої бази знань не обов'язкова, достатньо лише загальні характеристики для проведення класифікації. Метод К-ближніх сусідів – це метод класифікації, який використовує властивості евклідових відстаней для визначення приналежності до певного класу. Цей метод є бінарним і для багатокласової класифікації потрібно також використовувати модель надбудову «Один проти всіх» або «Один проти одного», що призводить до збільшення витрати обчислювального ресурсу та збільшення часу на обробку. Таким чином для проведення класифікації було обрано алгоритм найвнього Баєса тому, що даний алгоритм є мультикласифікатором і відсутня необхідність реалізації моделі – надбудови. Також алгоритм використовує мало обчислювальних ресурсів для проведення класифікації та швидко відпрацьовує на великих об'ємах бази знань. Деталізація бази знань буде використовуватися для більш чіткої класифікації.

Проаналізувавши описані вимоги розроблено архітектуру функціонування хмарного обчислювального середовища для обробки анамнезу пацієнтів з захворюванням дихальної системи на основі алгоритму машинного навчання (рис. 1).

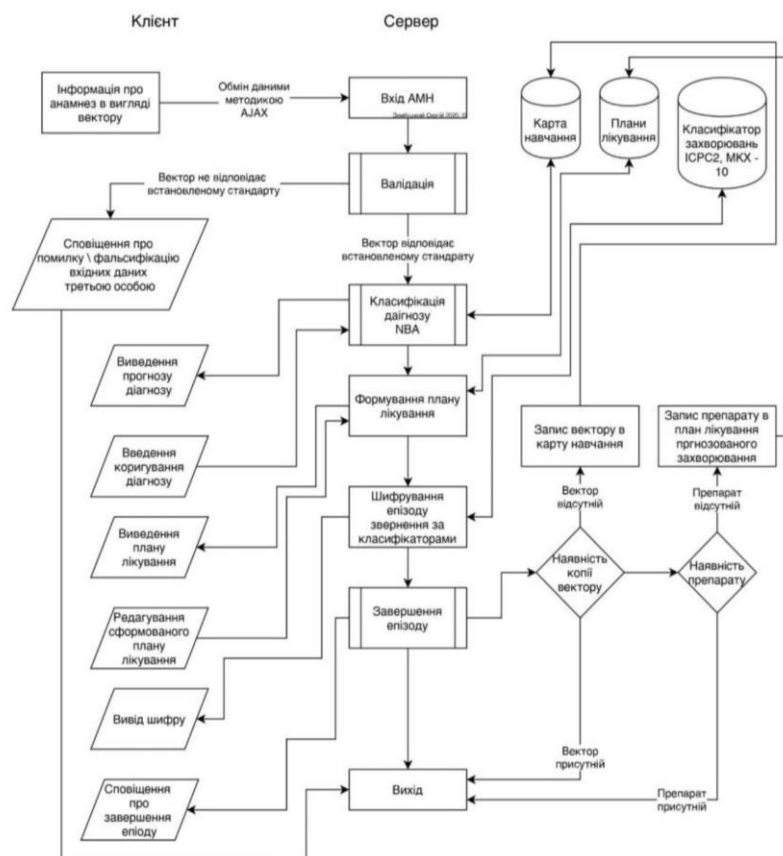


Рис. 1. Блок-схема архітектури хмарного обчислювального середовища

Середовище функціонує за наступним алгоритмом: після введення анамнезу пацієнта на web сторінці на вхід подається методом AJAX вектор анамнезу пацієнта. Вектор проходить валідацію, за невідповідності вектора до встановленої структури користувачу виводиться на екран сповіщення про помилку в системі або підробку даних з боку третіх осіб і рекомендується перезавантажити сторінку, якщо вектор задовольняє всі умови та відповідає встановленій структурі він передається на вхід обчислювального ядра алгоритму наївного Баеса. На етапі класифікації ядро АНБ отримує актуальний зліпок бази знань з карти навчання, проводить класифікацію вхідного вектору, передає методом AJAX користувачу інтерпретований вихідний вектор, який містить в собі назву діагнозу відповідно до введеного анамнезу, та виводить його на екран. Задля збереження людського фактора в системі розроблена можливість змінювати діагноз якщо на думку користувача (сімейного лікаря) він не відповідає тому, що встановив АНБ. На етапі формування плану лікування пацієнта на вхід методом AJAX подається вектор діагнозу, ядро відповідально за формування плану лікування звертається до JSON структурованого об'єкту, який містить плани лікування та формує план лікування згідно вказаних умов. Сформований план лікування передається користувачу методом AJAX, пулом даних та виводиться на екран. За потреби користувач може редагувати план лікування. Далі процес виконання передається в модуль шифрування епізоду звернення пацієнта. На даному етапі ядро модулю шифрування епізоду звертається до JSON структурованого об'єкту, який містить шифри ICPC2 та МКХ – 10 та за вхідним вектором кінцевого діагнозу, переданим методом AJAX, проводить шифрування епізоду за системами ICPC2 та МКХ – 10. Модуль шифрування епізоду звернення повертає значення шифру методом AJAX та виводить на екран користувача. На завершальному етапі проводиться підготовка до вивантаження пулу даних до баз знань. Спершу проходить перевірка на наявність точної копії в базі знань введеного користувачем вектору анамнезу. За відсутності точної копії вектору анамнезу модуль завершення епізоду проводить перевірку на відповідність структури вектору та вносить новий вектор в базу знань векторів. Якщо точна копія вектору наявна в базі знань модуль завершення епізоду переходить до аналізу плану лікування. Аналізатор плану лікування проводить пошук в структурованому об'єкті JSON бази знань препарати наявні в плані лікування. За відсутності препаратів вносяться в базу знань, за наявності – ігноруються. Після внесення пулу інформації до баз знань клієнту виводиться повідомлення про успішність завершення епізоду.

**Висновки.** На основі поданого матеріалу можна зробити висновок, що реалізований, за даною архітектурою, кінцевий програмний продукт допоможе сімейному лікарю оптимізувати робочий процес, зменшити витрату часу на прийом пацієнта та обробку його анамнезу захворювання, встановити загальний шаблон алгоритму та порядок дій проведення та обробки прийому пацієнта. Дана архітектура має можливість масштабування та розгортання на різних типах хмарних обчислювальних середовищ. Завдяки використанню алгоритму наївного Баеса відповідь від сервера на запит користувач отримує швидко, також даний алгоритм дає можливість обробляти великі навчальні вибірки та швидко проводити класифікацію діагнозу. Основною позитивною властивістю являється модульність архітектури. Зі сторони розробки модульність виражається в можливості легкої та швидкої інтеграції нового розробленого модулю, а зі сторони користувача модульність виражається в можливості використання незалежно один від одного модулів встановлення діагнозу, розробки плану лікування та шифрування звернення пацієнта.

### Література

1. Порєва Г.С. Методи аналізу звуків легень для оцінки стану дихальної системи людини / Порєва Г.С. / Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», 2020. – 2 с.

2. Шахгельдян К.И., Гельцер Б.И., Курпатов И.А. Методы машинного обучения для дифференциальной диагностики болезней органов дыхания / Шахгельдян К.И., Гельцер Б.И., Курпатов И.А. / Владивостокский государственный университет экономики и сервиса, 2019. – С. 10–13.

### References

1. Porieva H.S. Metody analizu zvukiv lehen dlia otsinky stanu dykhalnoi systemy liudyny / Porieva H.S. / Natsionalnyi tekhnichnyi universytet Ukrainy «Kyivskiy politekhnichnyi instytut imeni Ihoria Sikorskoho», 2020. – 2 s.

2. Shahgeldyan K.I., Gelcer B.I., Kurpatov I.A. Metody mashinnogo obucheniya dlya differentsialnoj diagnostiki boleznej organov dyhaniya / Shahgeldyan K.I., Gelcer B.I., Kurpatov I.A. / Vladivostokskij gosudarstvennyj universitet ekonomiki i servisa, 2019. – S. 10–13.

Надійшла / Paper received: 27.04.2020

Надрукована / Paper Printed : 05.06.2020

**ДОДАТОК Ж**  
**(ОБОВ'ЯЗКОВИЙ)**  
**РЕЗУЛЬТАТ ПЕРЕВІРКИ НА АНТИПЛАГІАТ**

Mon Nov 23 12:48:26 EET 2020, Стецюк Віктор Іванович, Хмельницький національний університет, ХНУ

## Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

Словари проверки: en\_US, ru\_RU, ua\_UA. Ошибок в документах: 9%

ID: 80925 Название: Модель оптимізації робочого процесу сімейних лікарів на основі штучної нейронної мережі Добавлено в БД: 2020-11-23 Авторы: Зембіцький Сергій Павлович Руководители: Гришинська Надія Василівна Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	139568	1203	5255 (4%)	58 (5%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

**ДОДАТОК К**  
**(ОБОВ'ЯЗКОВИЙ)**

**МАТЕРІАЛИ ПРЕЗЕНТАЦІЇ ДЛЯ ЗАХИСТУ ДРМ**



Модель оптимізації робочого  
процесу сімейних лікарів на  
основі штучної  
нейронної мережі

**ЗЕМБІЦЬКИЙ СЕРГІЙ  
ПАВЛОВИЧ**

ПМм 19 - 1

**ГРИПИНСЬКА НАДІЯ  
ВАСИЛИВНА**

к.ф.-м.н., доцент

## ОБ'ЄКТ, ПРЕДМЕТ, МЕТА, ГІПОТЕЗА ТА НОВИЗНА

**Об'єкт:** мінімізація часу обробки епізоду пацієнта.

**Предмет:** модель оптимізації обробки анамнезу пацієнта .











**Мета:** розробка моделі оптимізації обробки звернення пацієнта, сімейним лікарем, на основі алгоритмів машинного навчання.

**Гіпотеза:** використання алгоритмів машинного навчання зменшить витрату часу на обробку анамнезу респіраторного захворювання пацієнта, розробку плану лікування та фіксацію звернення пацієнта до сімейного лікаря в амбулаторну карту пацієнта.

**Наукова новизна:**

- + **Модель звернення пацієнта**
- + **Модульність** моделі обробки анамнезу пацієнта (графічна модель)
- + Інноваційний підхід до мінімізації вхідного вектору анамнезу
- + **Формування плану лікування з семплів інформації.**

## ЗАВДАННЯ

-  Ознайомитися з алгоритмами машинного навчання для прогнозування діагнозу
-  Розробити формулу обрахунку загального часу обробки звернення пацієнта
-  Розробити модель інтерпретації текстової інформації відібраного анамнезу в вектор
-  Розробити моделі баз знань для зберігання дамів необхідної інформації
-  Визначити оптимальний алгоритм машинного навчання для прогнозування діагнозу
-  Розробити модель оптимізації, пришвидшення або зменшення часу, обробки звернення пацієнта
-  На основі розроблених моделей розробити web застосунок для практичної апробації
-  Провести тестування web застосунку
-  Провести заміри та фіксацію витраченого часу на обробку звернення пацієнта з використанням розробленого web застосунку
-  Провести аналіз ефективності розробленого оптимізаційного рішення
-  Зробити висновки про проведення наукового дослідження

## ФОРМУЛА ОБРАХУНКУ ЗАГАЛЬНОГО ЧАСУ ОБРОБКИ ЗВЕРНЕННЯ ПАЦІЄНТА

$$p = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 + t_7 + t_8 + t_9$$

$t_1$  витрачений час на відбір первинної інформації про пацієнта

$t_2$  витрачений час на визначення стану самопочуття пацієнта

$t_3$  витрачений час на огляд носа пацієнта

$t_4$  витрачений час на огляд ротової порожнини пацієнта

$t_5$  витрачений час на огляд горла пацієнта

$t_6$  витрачений час на визначення характеристик кашлю в пацієнта

$t_7$  витрачений час на аускультацию дихальної системи пацієнта

$t_8$  витрачений час на розробку плану лікування пацієнта

$t_9$  витрачений час на фіксацію звернення пацієнта

Приводить до:

- + поділу звернення на етапи
- + структурування та утворення загального шаблону дій

Встановлює:

- + структуру модулів
- + структуру функціональних елементів
- + взаємозв'язок між компонентами

Дозволяє:

- + обрахувати загальний час обробки звернення
- + провести аналіз витрати часу на кожний етап обробки звернення для проведення дій щодо зменшення витрати часу

## МОДЕЛЬ ІНТЕРПРЕТАЦІЇ ВІДБІРАННОГО АНАМНЕЗУ У ВЕКТОР

$$r = \sum_0^s \sum_0^n id * k$$

s – кількість категорій згідно з якими проводиться

відбір анамнезу,

n – кількість факторів впливу на виникнення

захворювання (в машинному представленні

кількість ітерацій),

id – цифрове значення унікального номеру

фактору впливу,

k – коефіцієнт коригування та зсуву (обирається

самостійно архітектором системи машинного

навчання).

$$\vec{v}_{learning} = (52, 375, 60, 74, 46, 0, 72, 679)$$

Основна позитивна характеристика – інноваційний підхід до мінімізації вхідного вектору.

Взамін **52-х** факторів формується **тільки 7** значень і навчальна пара (права частина)

Надає:

- + можливість зменшити використання трафіку під час «спілкування з сервером»
- + можливість зменшення обчислювального ресурсу
- + можливість зменшення об'єму бази знань клініки та анамнезу

## БАЗА ЗНАНЬ КЛІНІКИ ТА АНАМНЕЗУ

$$K_{db} = \begin{bmatrix} a_{11} & b_{12} & c_{13} & d_{14} & e_{15} & f_{16} & g_{17} & y_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{ij} & b_{ij} & c_{ij} & d_{ij} & e_{ij} & f_{ij} & g_{ij} & y_i \end{bmatrix}$$

a – значення клініки або анамнезу для категорії самопочуття,

b – значення температури,

c – значення клініки або анамнезу для категорії огляд носа,

d – значення клініки або анамнезу для категорії огляд ротової порожнини,

e – значення клініки або анамнезу для категорії огляд горла,

f – значення клініки або анамнезу для категорії кашель,

g – значення клініки або анамнезу для категорії аускультация дихальної системи,

y – результуюче значення

0 0 0 0 0 0 0 0  
 78 375 90 111 69 0 108 831  
 0 370 51 0 0 363 0 784  
 60 373 0 63 0 90 0 586  
 18 380 0 0 66 195 0 659

Відсутній анамнез

Синусит

Фарингіт

Тонзиліт

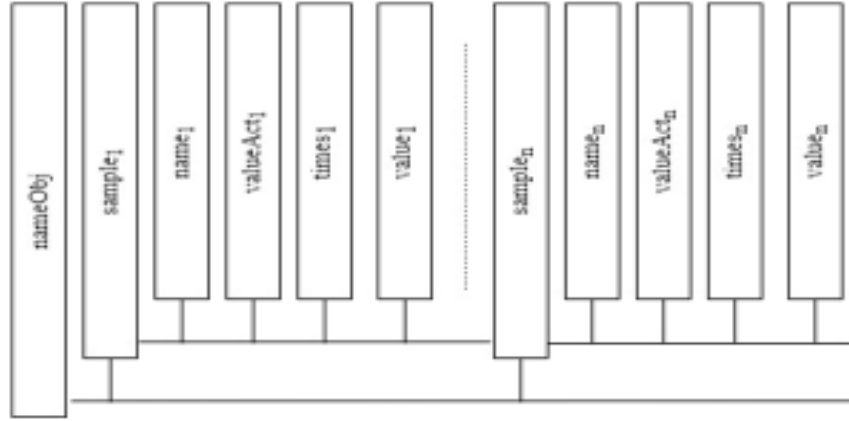
Тонзилофарингіт

Можливість модифікації бази знань без руйнування існуючої архітектури

Відносно невеликий розмір вихідного файлу бази знань. Розмір файлу з векторами клініки близько 360 байт.

Невеликий розмір вихідного вектору близько 24 байт.

## БАЗА ЗНАНЬ СЕМПЛІВ ПЛАНУ ЛІКУВАННЯ ЗАХВОРЮВАННЯ



```

"pot": [
  {
    "name": "Леводоноксацин",
    "valueAct": "750 мг",
    "times": "1р/д",
    "value": "5 днів"
  },
  {
    "name": "Доксициклін",
    "valueAct": "100 мг",
    "times": "2р/д",
    "value": "5 днів"
  },
  {
    "name": "Цефподоксим",
    "valueAct": "200 мг",
    "times": "2р/д",
    "value": "5 днів"
  }
]
  
```

Присутня можливість:

- + формування неглибоких вкладень
- + проведення швидкої індексації
- + швидкого створення вибірок семплів інформації
- + швидкої модифікації архітектури

Відносно невеликий час отримання актуальних зліпків семплів з бази знань

Відносно невеликий розмір файлу, в середньому близько 1 kb

## ВИЗНАЧЕННЯ ОПТИМАЛЬНОГО АМН

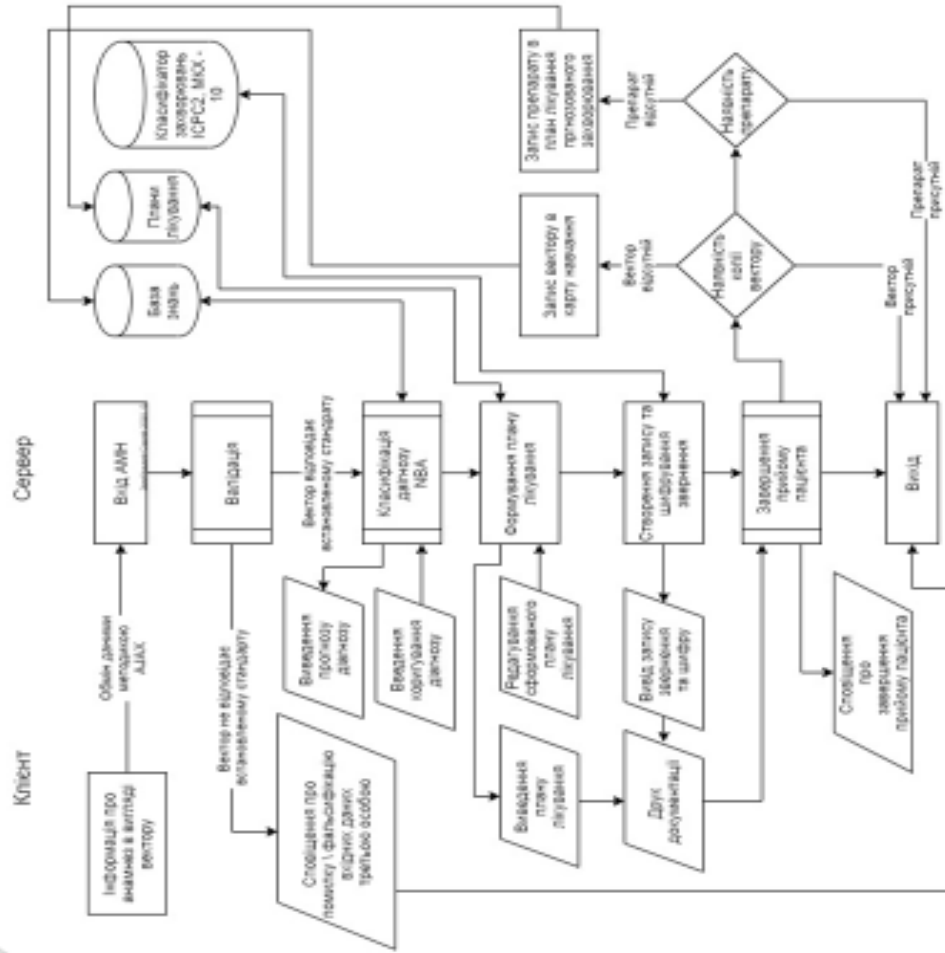
### Локальний сервер

	Алгоритм найвного		Метод опорних		Метод k – ближніх	
	Байєса	векторів	векторів	сусідів	сусідів	сусідів
Загальний час обчислення по категорії виконання скриптів (мс)	89.0	<b>115.0</b>			112.0	
Загальний час по активності дії (мс)	8.9	<b>15.1</b>			12.3	
Час на обробку запиту сервером (мс)	2.0	<b>8.2</b>			5.4	
Використана пам'ять для проведення обчислення (мб)	2.6	<b>3.1</b>			2.9	
Кількість векторів в базі знань	16	<b>16</b>			16	
Час відбору активності (сек.)	5.0	<b>5.0</b>			5.0	

### Хост сервер

	Алгоритм найвного		Метод опорних		Метод k – ближніх	
	Байєса	векторів	векторів	сусідів	сусідів	сусідів
Загальний час обчислення по категорії виконання скриптів (мс)	138.0		<b>178.0</b>		172.5	
Загальний час по активності дії (мс)	8.6		<b>14.5</b>		11.9	
Час на обробку запиту сервером (мс)	1.7		<b>6.9</b>		4.6	
Використана пам'ять для проведення обчислення (мб)	4.5		<b>5.5</b>		5.2	
Кількість векторів в базі знань	16		<b>16</b>		16	
Час відбору активності (сек.)	5.0		<b>5.0</b>		5.0	

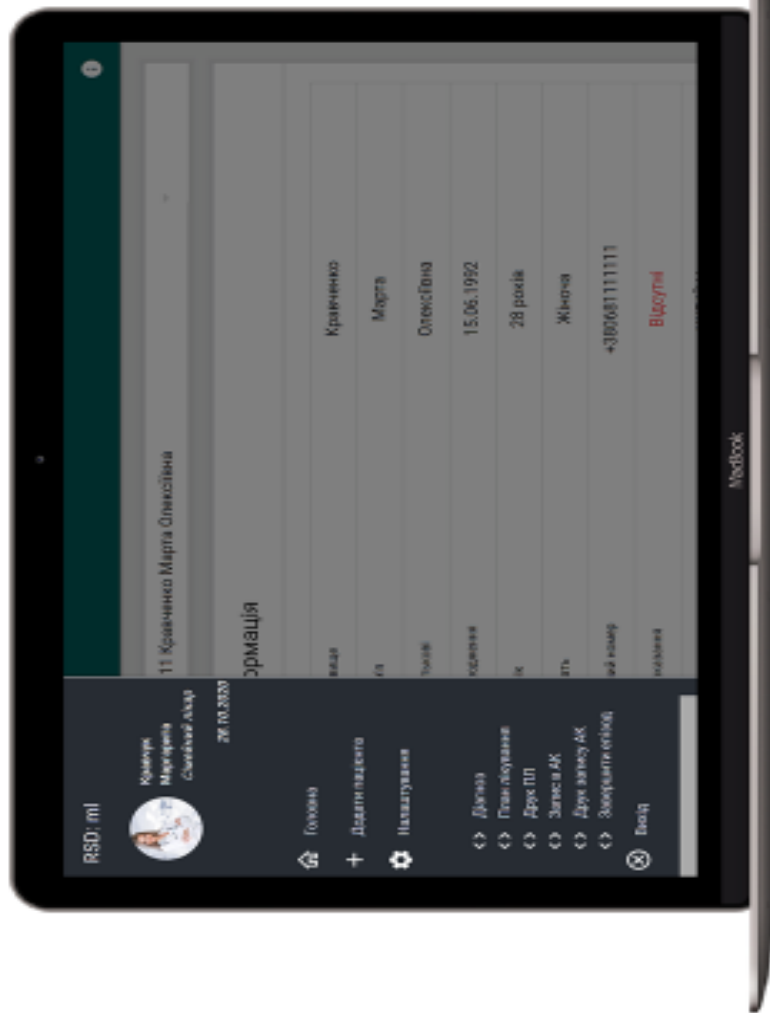
## МОДЕЛЬ ОПТИМІЗАЦІЇ ОБРОБКИ ЗВЕРНЕННЯ ПАЦІЄНТА



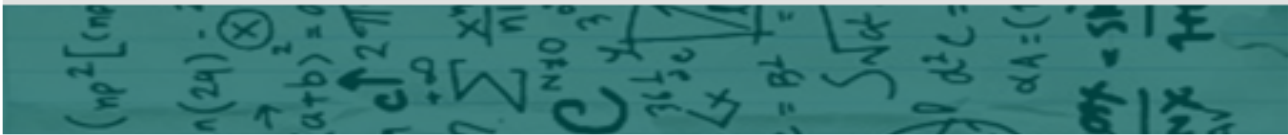
Концептуальною моделлю передбачено:

- + **модульність архітектури,**
- + **процес валідації вхідного вектору,**
- + **спілкування між клієнтом та сервером методом AJAX,**
- + **перенавчання на «льоту»,**
- + **зміну за потреби алгоритму машинного навчання,**
- + **роботу з формуванням документів для друку,**
- + **збереження людського фактору,**
- + **повний процес поповнення новими знаннями.**

# WEB ЗАСТОСУНОК



rsdml in **Google** search  
rsdml.000webhostapp.com



## UNIT TESTING

Властивість, швидкість підключення (2/100 Mbit/s)	Статус обробки	Швидкість (ms)	Об'єм завантаження (b/kb/mb)
Завантаження сторінки (2)	200	5500.0 ms	1.4 mb
Завантаження сторінки (100)	200	3520.0 ms	1.4 mb
Профілювання пацієнта (2)	200	697.0 ms	514 b
Профілювання пацієнта (100)	200	187.0 ms	514 b
Модуль «План лікування»			
Визначення діагнозу (2)	200	205.0 ms	234 b
Визначення діагнозу (100)	200	191.0 ms	234 b
Отримання плану лікування (2)	200	735.0 ms	423 b
Отримання плану лікування (100)	200	685.0 ms	423 b
Додати медикамент	200	45.3 ms	-
Видалити медикамент	200	45.0 ms	-
Формування друку плану лікування	200	3962.0 ms	-
Модуль «Запис до амбулаторної карти пацієнта»			
Створити запис (2)	200	190.0 ms	237 b
Створити запис (100)	200	168.0 ms	237 b
Формування друку запису до АК	200	4125.6 ms	-

Створено імітацію низької пропускної здатності та PON мережі

Отримано результати:

- + Відносно невеликий розмір кінцевого застосунку
- + Відносно невеликий розмір інформації обміну між клієнтом та сервером
- + Невеликий час обробки запиту
- + Висока стабільність роботи web – застосунку
- + Стабільність виконання алгоритму машинного навчання
- + Класифікація вхідного вектору у повній відповідності до існуючих захворювань в базі знань

## ЕФЕКТИВНІСТЬ

Етап проведення звернення пацієнта	Без використання RSD:ML (хв)	RSD:ML (хв)	Заощаджено (хв)	Ефективність (%)
Відбір пораненої інформації про пацієнта	2	1	1	50.0
Визначення стану самопочуття пацієнта	8	8	0	0.0
Огляд носа пацієнта	1	1	0	0.0
Огляд ротової порожнини пацієнта	1	1	0	0.0
Огляд горла пацієнта	1	1	0	0.0
Визначення характеристик кашлю в пацієнта	1	1	0	0.0
Аускультация дихальної системи пацієнта	10	10	0	0.0
Розробка плану лікування пацієнта	8	3	5	62.5
Фіксація звернення пацієнта	10	1	9	90.0
Всього витрачено часу	42	27	15	35.7

При умові 8 годинного робочого дня приблизна кількість прийнятих пацієнтів :

Без використання RSD:ML  
близько 11 пацієнтів

Використовуючи RSD:ML:  
близько 17 пацієнтів

Приріст 35.3%.

Структурування та приведення до єдиного шаблону дій сімейного лікаря.

Автоматизація формування плану лікування та запису до форми №025/о

Фінансова вигода

## ВИСНОВКИ

### Визначено:

- + обробка звернення займає **до 50 хв**,
- + проблематику зменшення часу обробки,
- + існуючі підходи та рішення **не зменшують час обробки** звернення,
- + **відсутність функціоналу** для вирішення проблематики.

### Розроблено:

- + формулу обрахунку загального часу обробки звернення пацієнта,
- + моделі інтерпретації текстового опису анамнезу в вхідний вектор,
- + модель бази знань,
- + графічну модель збереження інформаційних семплів плану лікування,
- + графічну модель оптимізації обробки звернення пацієнта з респіраторним захворюванням,
- + web застосунок на основі вище описаних моделей,
- + проведено unit тестування web застосунку, практичну апробацію та визначення ефективності застосунку.

### Сформовано:

- + актуальність,
- + мету,
- + об'єкт,
- + предмет,
- + гіпотезу,
- + завдання
- Дослідження,
- + оптимальний SDK

## ПРАКТИЧНА АПРОБАЦІЯ

### Проведено:

Практичну апробацію сімейним лікарем на прийомі пацієнтів.

### Отримано:

Позитивний відгук від сімейного лікаря про застосунок RSD:ML.

### Можна рекомендувати:

Моделі як основу математико – логічної та функціональної бази для розробки оптимізаційного рішення

### Впроваджується:

Web застосунок RSD:ML в масове використання з під'єднанням до ЦБД eHealth (НСЗУ)

### Публікація в науковому журналі

С. П. Зембіцький, Н. В. Грипинська Проектування архітектури хмарного обчислювального середовища для обробки анамнезу пацієнтів із захворюванням дихальної системи на основі алгоритмів машинного навчання/ С.П. Зембіцький, Н.В. Грипинська// Вісник Хмельницького національного університету. Технічні науки. – Хмельницький. – 2020. -№3. –С. 88 – 92.

### Майбутні дослідження:

+ на основі розроблених моделей проводити наукові дослідження за напрямком використання штучного інтелекту для формування плану лікування та рекомендації направлень до вузькопрофільного спеціаліста,

+ web застосунок в подальшому можна вдосконалити підтримкою API для забезпечення створення єдиної інтелектуальної САПР сімейного лікаря.

## ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

## РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник Зембіцький Сергій ПавловичТема Модель оптимізації робочого процесу сімейних лікарів на основі штучної нейронної мережіОсвітній рівень: магістрГалузь знань: 11 Математика та статистикаСпеціальність: 113 Прикладна математика

## Обсяг дипломної роботи

Кількість сторінок записки без додатків 93Кількість сторінок додатків 29

## Характеристика дипломної роботи магістра:

1. Короткий зміст ДР та прийнятих рішень Метою роботи є розробка моделі оптимізації обробки звернення пацієнта сімейним лікарем на основі алгоритмів машинного навчання.Створено комплекс оптимізаційних моделей для пришвидшення обробки звернення пацієнта з респіраторним захворюванням;розроблено програмне рішення практичної реалізації моделі оптимізації;проведено аналіз отриманих результатів практичної апробації програмного рішення.2. Висновок про відповідність ДР завданню на ДР Робота повністю відповідає завданню

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки, техніки і передових методів роботи: Робота складається з вступу, 3 розділів, висновків, додатків та переліку посилань. У вступі висвітлена актуальність поставленої задачі та мета роботи. Перший розділ присвячений дослідженню стану предметної області і розкривається постановка задачі для створення програмного забезпечення. У другому розділі описаний процес вибору та обґрунтування процесу вибору інструментарію для вирішення поставленої задачі. У третьому розділі описано процес розробки оптимізаційних моделей та web-застосунку, проведено практичну апробацію та тестування. У висновках зроблено підсумок виконаної роботи. Перелік посилань складає 22 джерела інформації. Робота виконана з використанням сучасних підходів до моделювання та комп'ютерних технологій

4. Позитивні сторони роботи Посадано теоретичні дослідження із практичним застосуванням. Розроблено web-застосунок проведення практичної апробації розробленої моделі із адаптивним позиціонуванням об'єктів та елементів користувача

5. Негативні сторони роботи Залежність програмного продукту від програмного забезпечення сторонніх розробників

6. Оцінка графічного оформлення та пояснювальної записки роботи Текстова частина виконана акуратно і згідно діючих стандартів. Формули, таблиці, рисунки оформлені відповідно до вимог.

7. Відгук про роботу в цілому Робота виконана на високому теоретичному рівні, має практичне застосування.

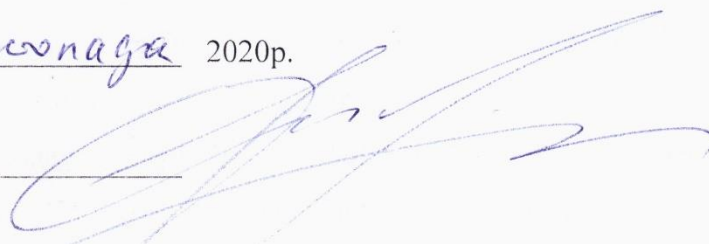
8. Інші зауваження Помилки, що впливають на кінцевий результат роботи не знайдено.

9. Оцінка дипломної роботи Дипломна робота магістра Зембіцького С.П. заслуговує на оцінку «відмінно».

РЕЦЕНЗЕНТ Міхалевський Віталій Цезарійович, кандидат фізико-математичних наук, доцент кафедри КНІТ Хмельницького національного університету

Прізвище, ім'я, по батькові, посада, місце роботи

27 листопада 2020р.



Підпис

Завідувачу кафедри ТМІТ  
д-р.техн.наук Підченку С.К.

Зембоцький Сергій Павлович  
ПІБ здобувача вищої освіти

ФПКТС, 2 курсу, групи ПМм-19-1

### ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів(Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

30.11.2020р.

дата

С.Зем  
підпис

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ**  
**КАФЕДРИ ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Модель оптимізації робочого процесу сімейних лікарів на основі штучної нейронної мережі \_\_\_\_\_

Автор: \_\_\_\_\_ Зембіцький Сергій Павлович \_\_\_\_\_

Спеціальність: \_\_\_\_\_ 113 – прикладна математика \_\_\_\_\_

Освітня програма: \_\_\_\_\_ освітньо-професійна \_\_\_\_\_

Науковий керівник: \_\_\_\_\_ Грипинська Надія Василівна, к.ф.-м.н доцент \_\_\_\_\_

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	+
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

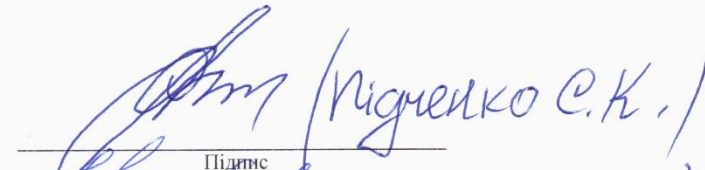
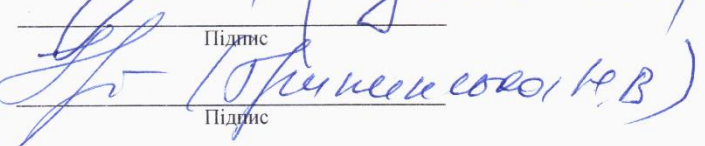
Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) відсоток співпадінь з кожним із документів не перевищує 1,01%.

3. 12. 20

Дата

  
 Підпис  
  
 Підпис

## Відгук про web додаток RSD:ml.

Я Козяр О.В. власник та засновник кабінету медичної практики сімейної медицини, сімейний лікар. Я використовую в своїй роботі web додаток RSD:ml розроблений Зембіцьким С.П. Додаток зручний у роботі, має приємний зовнішній вигляд. Інформація подається в структурованому вигляді, що призводить до полегшення роботи з додатком. Користуючись даним додатком можна привести до єдиного шаблону свої дії при роботі з пацієнтом. Після введення анамнезу пацієнта сервіс сам створює прогноз діагнозу. Також є можливість формування плану лікування та запису звернення пацієнта в автоматичному режимі. Присутня можливість зміни діагнозу звернення пацієнта та редагування плану лікування таким чином відбувається збереження «людського фактору» тому, що машини теж можуть помилятися. Повну автоматизацію обробки звернення приносить можливість друку документації такої як план лікування пацієнта та запис до амбулаторної карти форма №025/о. Також даним додатком можна користуватися з різних пристроїв, в кабінеті я можу використовувати ноутбук, на виїзді можна використовувати смартфон.

Таким чином я як сімейний лікар використовуючи даний додаток не витрачаю час на написання паперової документації. Діагноз звернення пацієнта встановлюється автоматично згідно з введеним анамнезом, зразу можна сформулювати та роздрукувати для подальшого прикріплення до амбулаторної карти пацієнта запис до амбулаторної карти та план лікування для видачі пацієнту. Тому я набагато менше витрачаю часу на обробку звернення пацієнта використовуючи даний додаток та можу прийняти більше пацієнтів за робочий день. На мою думку даний додаток можна рекомендувати для користування також іншим сімейним лікарям.

*Козяр О.В.*

